

**CS403 Midterm Mcqs**  
**Spring 2022 #Mid-Term**  
**WITH PROVE ANSWER**  
**ORANGE MONKEY TEAM**

**Important NOTE:-**

Maybe toppers are successful in school or college carrier but backbenchers are successful in their life because toppers run according to books while backbenchers don't follow rules and regulation that is why backbenchers are successful. For life, there are no books and another reason why toppers can't succeed because they fear failure and don't try anything new. Backbenchers don't fear failure as they have faced the failure in their school or college carrier so that is the reason they try new things and at last gain success.

As Abdul Kalam nicely quoted, "The best brains of the nation may be found on the last benches of the classroom"

**Translation into Urdu:**

---

**ORANGE MONKEY TEAM**

ہوسکتا ہے کہ ٹاپر اسکول یا کالج کیریئر میں کامیاب ہوں لیکن بیک بینچر اپنی زندگی میں کامیاب ہوتے ہیں کیونکہ ٹاپر کتابوں کے مطابق چلتے ہیں جبکہ بیک بینچر قواعد و ضوابط پر عمل نہیں کرتے ہیں اسی وجہ سے بیک بینچر کامیاب ہوتے ہیں۔ زندگی کے لیے، کوئی کتابیں نہیں ہیں اور ایک اور وجہ ہے کہ ٹاپرز کامیاب نہیں ہو پاتے کیونکہ وہ ناکامی سے ڈرتے ہیں اور کچھ نیا کرنے کی کوشش نہیں کرتے۔ بیک بینچر ناکامی سے نہیں ڈرتے کیونکہ انہوں نے اپنے اسکول یا کالج کیریئر میں ناکامی کا سامنا کیا ہے یہی وجہ ہے کہ وہ نئی چیزیں آزما رہے ہیں اور آخر کار کامیابی حاصل کرتے ہیں۔

جیسا کہ عبدالکلام نے اچھی طرح سے نقل کیا ہے، "قوم کے بہترین دماغ کلاس روم کے آخری بینچوں پر مل سکتے ہیں۔"



1. In cross reference matrix attributes are specified on \_\_\_\_\_.

Y axis....confirm

55. In **cross reference matrix** transitions are specified on \_\_\_\_\_ on \_\_\_\_\_.
- a. Y axis, X axis
  - b. X axis, Y axis
  - c. X axis, X axis
  - d. **Y axis, Y axis**

2. Which model operates at the lowest level of abstraction?

Physical.....confirm

Which model operates at the **lowest** level of abstraction?

Select correct option:

conceptual

internal

external

**physical**

3. Entities enrolled in a relationship are called its \_\_\_\_\_.

Participants...confirm

4. A \_\_\_\_\_ is an abstraction of a relationship

Relationship type....confirm

A *relationship type* is an abstraction of a relationship

5. In which phase of database development process specific database management system (DBMS) is chosen?

Logical data base design.....confirm

#### Logical Database Design

This is the next phase of database design, in which appropriate data model is **chosen**,

6. DBMS select a specific file organization for the storage of data on disk, to implement that specific file system the DBMS needs to create specific \_\_\_\_\_.

Indexes.....confirm

storage of data on disk, to implement that specific file system the DBMS needs to **create specific** indexes. Now whenever the DBMS will

7. Data inconsistency leads to a number of problems such as \_\_\_\_\_.

Loss of information and incorrect result.....confirm

specific data item. **Data inconsistency** leads to a number of problems, including loss of information and incorrect results. In database approach it is controlled because data is

8. Which of the following change can cause problem in database levels?

Deleting an attribute from the database structure.....confirm

inconsistency in the **database levels**. The changes which can be done transparently may include the following:

- Adding a file to the database
- Adding a new field in a file
- Changing the type of a specific field

9. In data based development process which of the following is related to requirement gathering phase.

Detailed study of system....confirm

flows between those sections. **Detailed study of** the system is started to find out the requirements of each section. This phase is the **detailed study of** the system and its functionality decisions made at this stage decide the overall activity of the organization. Requirements of one section of the organization are fulfilled in such a way that all the sections in the organization are supporting each other, for example we can say that the results produced by the processing taking place at one section are used as input for another section. All the users of the systems are interviewed and observed to pinpoint and precisely define the activities taking place in the different section of the organization.

## Database Development Process

10. Which of the following is correct about database modeling?

Process of creating the logical structure of the database.....confirm

The process of creating the logical structure of the database is called **database modeling**.

11. A relationship is an association between two entities or between an entity and itself. The diamond symbol for a relationship is \_\_\_\_\_ if one of the participants is dependent on the other.

Doubled.....confirm

Diamond is doubled if **one of the** participant is dependent on the other

12. Which of the following is incorrect related to secondary key?

Selected key must be unique identifier.....confirm

13. In database system the description and definition of data is stored in \_\_\_\_\_.

Schema....confirm

A database generally stores its schema in a data dictionary.

Although a schema is defined in text database language, the

14. A candidate key that does not have a null value and is selected to uniquely identify all other attribute values in any given row is called a \_\_\_\_\_.

Primary key....confirm

A certain value that may be associated with any attribute is NULL, that means "not given" or "not defined". A major characteristic of the PK is that it cannot have the NULL value. If PK is a composite, then none of the attributes included in the PK can have the NULL, for example, if we are using "name, fName" as PK of entity type STUDENT,

15. Which of the following is Not related to the management of data in the database?

How data will be transformed.....confirm

- Management of Data in the Database
- Management of Users associated with the database.

Management of the data means to specify that how data will be stored, structured and accessed in the database.

16. In entity-relationship diagram the entity type which has existence dependency constant is classified as:

Weak entity.....confirm

Que. In entity-relationship, the entity type which has existence dependency constraint is classified as

- a. single entity
- b. foreign entity
- c. weak entity
- d. strong entity

Answer: weak entity

17. Which of the following to the semantic Data Model?

E-R data model.....confirm

Developed in a semantic data model  
(generally E-R data model)

18. Detailed data flow diagram is created for only those processes from the \_\_\_\_\_ for which we want to show the details.

Level 0 diagram.....confirm

DFDs. The special features associated with this diagram are that, one, it is optional, that is, it is created for only those processes from the level 0 diagram for which we want to show the details. For a small sized system we may not need to develop even a single

19. The total participation by entities is represented in E-R diagram as:

Double line....confirm

20. Which of the following is the major drawback of file processing system?

## Non-sharing of data

Another major drawback in the traditional file system environment is the non-sharing of data. It means if different systems of an organization are using some common data then rather than storing it once and sharing it, each system stores data in separate files. This

21. A relationship that involves three entities called.

**Ternary relationship....confirm**

- **Ternary Relationship**

A Ternary relationship is the one that involves three entities e.g.

22. The entity relationship model consists of the following major constructs

\_\_\_\_\_  
**Entity, Attribute and relationship.....confirm**

### Constructs in E-R Data Model

The E-R data model supports following major constructs:

- Entity
- Attribute
- Relationship

23. The DBMS uses the \_\_\_\_\_ to access the database at each layer or model.

**Data Dictionary.....confirm**

The DBMS uses the data dictionary to access the database at each layer or model, f

24. Which of the following is the best description of the data security in DBMS?

**Efficient Check on all users access.....confirm**

- **Better Data Security:**

All application programs access data through DBMS, So DBMS can very efficiently check that which user is performing which action and accessing which part of data, So A DBMS is the most effectively control and maintain security of Data stored in a database.

25. Consider the following tables

EMP(empId, empName, qual, depld)

DEPT(depld, depName, numEmp)

The primary key of EMP table is \_\_\_\_\_ whereas foreign key of EMP table is \_\_\_\_\_.

Empld, depld

26. Which is not correct according to Data Flow Diagram?

DFDs provide us a way expressing decision points....confirm

**Data Flow Diagrams:**

The most common tool used for designing database systems is **Data Flow Diagram**. It is used to design systems graphically and expresses different system detail in different DFD levels.

DFDs show the flow of data between different processes of a specific system.

DFDs are simple, and hide complexities.

DFDs are Descriptive and links between processes describe the information flow.

27. Which of the following Greek letter is used to denote the select operation?



( ) Greek sigma symbol.....confirm

3. Which of the following is used to denote the selection operation in relational algebra?

- a) Pi (Greek)
- b) Sigma (Greek)
- c) Lambda (Greek)
- d) Omega (Greek)

View Answer

Answer: b

28. In relational data model, a record or an entity instance is represented as \_\_\_\_\_ in a relation.

Row...confirm

table consists of rows and columns. Rows of a table are also called tuples. A row or tuple of a table represents a record or an entity instance, where as the columns of the

29. While mapping many-to-many relationship \_\_\_\_\_ table(s) are created.

Three

30. Select and project are the examples of \_\_\_\_\_.

Unary Operation.....confirm

: "Select" and "project" are the examples of  
Select correct option:

Unary operations  
Binary operations

31. In natural join the common attribute in the output table appears \_\_\_\_\_.

Once.....confirm

In natural join the common attribute in the output table appears \_\_\_\_\_.

- Twice
- Three times
- Four times
- **Once**

32. RDM is simple there is just one structure and that is \_\_\_\_\_.

a Relation or a table....confirm

The RDM is simple, why, there is just one structure and that is a relation or a table.

33. RDM stands for \_\_\_\_\_.

Relational Data model....confirm

relational data model (RDM),

34. Which of the following constant states that if a foreign key exists in a relation either the foreign key value must match the primary key value of some tuple in its home relation or the foreign key value must be completely null?

Referential Integrity Constraint....confirm

**Referential Integrity Constraint:**

This constraint is applied to foreign keys. Foreign key is an attribute or attribute combination of a relation that is the primary key of another relation. This constraint states that if a foreign key exists in a relation, either the foreign key value must match the primary key value of some tuple in its home relation or the foreign key value must be completely null.

35. Which constraint is generally used for efficiency purpose in the data entry process.

**Default value.....confirm**

**Default Value:**

This constraint means that if we do not give any value to any particular attribute, it will be given a certain (default) value. This constraint is generally used for the efficiency purpose in the data entry process. Sometimes an attribute has a certain

36. The relationship between department to employees is .

**One to many....confirm**

In the second part of the Figure-1 we see a relationship between the Employee and project entities, the relationship describes one to many association between the project and the employees. It shows that there can be one project having a number of employees,

37. A row of a table represents a \_\_\_\_\_ whereas the column of the table represents the \_\_\_\_\_.

**Record, Attribute....confirm**

A row of a table represents a \_\_\_\_\_, whereas the column of the table represents the

\_\_\_\_\_.

Answer: Record, attribute

38. Which is not the function of Database Management System?

**Failure Response Service**

39. The conceptual database design can be transformed into any \_\_\_\_\_ .

**Data model**

40. Which of the following is the second stage in the database development process?

**Requirement Analysis**

41. Functional dependencies under interference rule reflexivity are called dependencies.

Trivial

42. If many instances of the “Book” entity are associated with the many instances of the “Student” entity, then the relationship between them is:

Many to many

43. Which of the following constraints enforces entity integrity?

PRIMARY KEY

44. Unary operations involves

Only one relation

45. A \_\_\_\_\_ entity has a primary key that is partially or totally derived from the parent entity in the relationship.

Weak

46. Business \_\_\_\_\_

Relationship

47. In \_\_\_\_\_ appropriate data model is chosen.

Logical Database Design

48. Which of the following is drawback of standardization?

Lack of uniqueness

49. Recursive relationship is also called \_\_\_\_\_ .

Unary

50. The foreign key attribute, which is present as a primary key in another relation is called as \_\_\_\_\_

\_\_\_\_\_ of the foreign key attribute.

**51. Binary relation**

**Home relation**

**52. DFD does not provide us a way to represent**

**Decisions**

**53. A \_\_\_\_\_ is used to maintain a connection between the users of the database system.**

**File server...confirm**

**54. Which of the following is an advantage of using the Traditional File Organization?**

**Simplicity**

**55. Which of the following is the input to Analysis Phase?**

**DFD**

**56. Name and \_\_\_\_\_ are part of the definitions of an attribute.**

**Domain**

**57. Which of the following most certainly implies the need for an entire table to implement?**

**A ternary relationship**

**58. Core of the Database Architecture is**

**Internal level**

**59. \_\_\_\_\_ is same as equi-join with a slight difference.**

**Natural join**

**60. Which of the following database environment causes a network overhead?**

**File server.....confirm**

desired operation on the desired data wants to write back the data on the database he will have to send the whole file back to the server, thus causing a lot of network overhead. The Good thing

61. Conceptual database design is implemented using a \_\_\_\_\_ .

Semantic data model....confirm

**Conceptual database design**

This design is implemented using a semantic data model,

62. Which constraint is generally used for efficiency purposes in the data entry process?

Default value

63. A 3 NF relation is converted to BCNF by

Dependent attributes of overlapping composite keys are put in a separate relation

64. Each \_\_\_\_\_ of a table contains atomic/single value.

Cell

65. A relation is said to be in BCNF when

It has no overlapping composite keys which have related attributes

66. Which one of the following is not a part of three-level architecture?

Physical level

67. Which of the following is INCORRECT about naming entities?

Always use abbreviations as they are convenient

68. In entity relationship diagram an instance is

Any particular entity

69. Which of the following is an example of metadata

Student

70. Level of data at which entities or objects exist in reality is called

Real world data

71. There are \_\_\_\_\_ basic properties of the database relations.

Six

72. The population of the data of the organization for which the database is created is called the \_\_\_\_\_ of the database.

Extension...confirm

62. The population of the data of the organization for which the database is created is called the \_\_\_\_\_ of the database.

- a. Extension
- b. Intension

73. A key which consist of two or more attributes is called\_.

Composite key....confirm

or it could be composite which consists of two or more attributes.

74. How many types of dependencies are?

Three...(existence dependency, Identifier dependency, referential dependency).....confirm

75. A mapping is one to one if each entity in X is associated with at most \_\_\_\_\_ entity in Y and vice versa.

One....confirm

A mapping R from X to Y is one-to-one if each entity in X is associated with at most one entity in Y and vice versa.

76. In \_\_\_\_\_ join, only selected rows of a relation are made cross product with second relation.

Theta

77. Which of the following is not a key?

Integrity constraint key....confirm

**ORANGE MONKEY TEAM**

- Super Key
- Candidate Key
- **Primary Key**
- Alternate Key
- Secondary Key
- Foreign Key

78. A collection of concepts that can be used to describe the structure of a database

**Data model**

79. Any combination of attributes with super key is

**Super key**

80. Binary relationships are those, which are established between

**Two entity type**

81. When data stored onto a magnetic media it is stored in \_\_\_\_\_-.

**Binary format.....confirm**

data when stored onto a **magnetic media** is stored in binary format,

82. Which type of Data flow diagram (DFD) explains the functionality of the processes?

**Context level diagram**

This is **Data Flow Diagram** (DFD)  
 I have a context diagram of the s  
 in the examination system, these  
 her entities.

83. In the employee table, we may store the name, father name, phone but age can be computed from the data of birth. SO age is the best example of the \_\_\_\_\_ attribute.

**Derived.....confirm**

stored in the database or obtained some other way. For example, we may store the name, father name, address of employees, but age **can be computed** from date of birth. The

© Copyright Virtual University of Pakistan

77

Database Management System (CS403)

VU

advantage of declaring age as derived attribute is that whenever we will access the age,

84. \_\_\_\_\_ are called participants, when enrolled in a relationship.

**Entities**

85. In case of Context-level Diagram, the system is represented by

**One process only**

86. Collection of raw facts is called \_\_\_\_\_ .

**Data**

87. A software package designed to store and manage database

**DBMS**

88. In \_\_\_\_\_ join common attributes appear twice in the output with selected rows.

**Equi join**

89. Within a table, each primary key value \_\_\_\_\_ .

**Must be unique.....confirm**

Within a table, **each primary key** value \_\_\_\_\_.

Select correct option:

is a minimal superkey

is always the first field in each table

must be numeric

**must be unique**

90. **Must be numeric**

**Must be unique**

91. \_\_\_\_\_ is not an operation of relational algebra.

**Addition**

92. Candidate keys which are not selected as primary key are called

**Alternate keys**

93. Internal schema/view defines the \_\_\_\_ of the data.

**Structures...confirm**

**Internal schema** or view defines the structures of the data ;

94. Major benefits of data sharing is:

**Data of different application places at same place**

95. Functional dependency between two attributes or sets of attributes i.e. Q and P is represented as .

$Q \rightarrow P$

96. Which of the following is not the component of Relational Data Model?

**Manipulation Language**

97. The attribute that is calculated from other attribute is called .

**Derived attribute**

98. Consider the two attributes “CNIC\_NO” and “Registration\_NO”. Both the attributes can be used for the unique identification. Now if the user selects the “CNIC\_NO” as primary key, then “Registration\_NO” is a \_\_\_\_\_ key.

**Alternative**

99. Which of the following constraint is included in the definition of the table?

Null

100. The database deals with large amount of data. This property is called

Scalability

101. Which one of the following is NOT a characteristic of meta data?

Includes user data.....confirm

102. Theta join is denoted mathematically as \_\_\_\_\_ .

$R \times \Theta S$

103. In cross reference matrix transitions are specified on \_\_\_\_\_ and database objects are specified on \_\_\_\_\_ .

Y axis, Y axis

104. Each component of DBMS performs \_\_\_\_\_ functions.

Different

105. Which of the following data model is not record based?

Hierarchical Data Model

106. Network Data model

Semantic Data model

107. The basic purpose of diamond in ER diagram is to show the \_\_\_\_\_ .

Relationships

108. Which of the following tool is used to design conceptual database design?

This model is independent of any tool

109. In “one to many” cardinality one instance of a relation is mapped.

Many instances of second entity type

110. Cross reference matrix is developed at the \_\_\_\_\_ stage of the database.

Designing

111. User rights information is stored in

Catalog

112. Which of the following is incorrect about attributes?

One entity type may have two attributes with same name.....confirm

An attribute is identified by a name allocated to it and that has to be unique with respect to that entity type. It means **one entity type** cannot have two attributes with the same name. However, different entity types may have attributes with the same name. The guidelines for naming an attribute are similar to those of entity types. However, one

113. Logical data independency provides independency to change \_  
model.

Conceptual

114. In natural join the common attribute in the output table appears

Once

115. Which levels are mostly used for detailed DFD?

Level-0, Level-1.....confirm

116. By default, a non-key attribute in a relation can have \_\_\_\_\_ value.

Null

117. The three levels architecture is useful for

Hiding the details of internal systems

118. Which of the following is called a replacement of the manual file system?

File processing system.....confirm

computer-based approach of handling the commercial or business applications. That is why it is also called a replacement of the manual file system. Before the use computers,

119. In three level architecture of the database which level is responsible for storage of data on some storage media?

Internal/Physical view....confirm

120. With the help of \_\_\_\_\_ technique data values with the smaller sized codes can further reduce the space needed by the data for storage in the database.

Compression

121. If an entity is linked with itself, than it is called \_\_\_\_\_ relationship.

Recursive....confirm

An ENTITY TYPE linked with itself, also called recursive relationship.

122. Once the data has been transported to the physical level it is then managed by the \_\_\_\_\_.

Operating system.....confirm

enforced at this level. Once the data has been transported to the physical level it is then managed by the operating system. Operating

123. Duplication of data in controlled redundancy is controlled and \_\_\_\_\_.

Deliberate.....confirm

124. Teleprocessing is a sub-type of \_\_\_\_\_.

Multi-user database

125. In an ERD, the focus is on the \_\_\_\_\_ and the relationships between them.

Entities

126. The most widely used conceptual model is the \_\_\_\_ model.

ER

127. Student and book are two entities, what is the cardinality between the two?

Many to many

128. RDM is a simple there is just one structure and that is .

A relation or a table

129. A description on a particular collection of data using the given data model.

Database

130. Consider thy key “CNIC, Name” and choose the best option that describes the key most effectively.

The given key is a super key and candidate key as well.

131. Normalization is a process of restricting a relation to

Minimize duplication of data in a database

132. Unary relationship involves a single entity it is also called \_ .

Recursive relationship....confirm

#### Unary Relationship

These are the relationships, which involve a **single entity**. These are also called recursive relationships. Unary relationships may have one to one, one to many and

133. Which of the following is a part of a DFD but not included in the ERD?

The external entity having a single instance

134. The database management system (DBMS) is used to

All of above

135. In cross reference matrix attributes are mention on \_\_\_\_\_ and reports are mention on

X axis, Y axis

136. A table can be logically connected to another table by defining a \_\_\_\_\_.

**Primary key.....confirm**

Explanation : A table can be logically connected to another table by defining a **Primary Key**.

137. In the relation model, the number of attributes and number of tuples in a relation are termed as \_\_\_\_\_ and \_\_\_\_\_ respectively.

**degree, cardinality**

138. The function that an entity plays in a relationship is called that entity's \_\_\_\_\_.

**Role**

139. A database system is designed to meet the information needs of \_\_\_\_\_ in an organization.

**Multiple users**

140. Which of the following is the example of a strong entity?

**Ahmad account of ABC bank has no existence if the ABC bank doesn't exist anymore**

141. Which of the following is a completeness constraint?

**Partial Completeness Constraint**

142. In an employee entity which of the following attribute will work as a primary key?

**Employee id**

143. \_\_\_\_\_ is the first computer-based approach for handing the commercial or business applications

**Database Systems**

144. Which of the following is NOT the classification of entity type?

Dependent entity type

145. Which of the following is NOT the type of key?

Derived key

146. Data Redundancy means

Duplication of data

147. Which of the following is a type of dependency?

Referential

148. In your opinion, why relational database is widely acceptable?

Due to its dependencies

149. \_\_\_\_\_ is the first comprehensive complete database design.

Conceptual database design

150. How data security can be assured in the database?

Implementing encryption algorithms

151. The attributes whose value is not stored in a database is known as \_\_\_\_\_ attribute.

Derived....confirm

152. Logical database design, like conceptual database design is our

Database design

153. If a single instance of “student” is associated with the many instances of the “subjects” entity, then the relationship between them is:

One to many

154. The conceptual database design is drawn through .  
ER model
155. A relational database is  
One that consists of two or more tables that are joined in some way .
156. Data, information  
Information, data
157. In which stage of the database development process logical design is created?  
Implementation
158. Which of the following is NOT an entity?  
Playground
159. After moving the determined columns from the original table to the new table while converting to 2NF, the determinate becomes the of the new table.  
Primary key
160. Binary operator  
Project operator
161. Free standing data dictionary is created by  
CASE tools
162. A patient can be an outdoor or indoor. Analyze this statement and tell what type of constraint will be applied on patient?  
Total completeness
163. If  $A \rightarrow B$  and  $A \rightarrow C$ , then  $A \rightarrow BC$ . The inference Rule applies is:  
Union
164. Which of the following is the correct abbreviation of ERD?  
Entity relationship diagram

165. Database relation is also represented in a two dimensional structure called  
**Table**

166. Which of the following specifies the number of instances of one entity that can or must be associated with each instance of another entity?

**Cardinality view.....confirm**

Recalling from the previous lecture we can say that that cardinality is just an expression which tells us about the number of instances of one entity which can be present in the second relation. Maximum cardinality tells us that how many instance of an entity can be placed in the second relation at most. Now we move onto discuss that what the minimum

167. Data about data is metadata

**True**

168. Which of the following describes the job of a database administrator?

**Monitoring and controlling database security and authorization: setting up controls to ensure the quality and integrity of data....confirm**

169. Each table must have a \_\_\_\_\_ key.

**Primary**

170. A relational database system is based on the concept(s) of:

**Tables, row and columns**

171.  $A \rightarrow BC$  and  $A \rightarrow B$ , then  $A \rightarrow C$  follow \_\_\_\_\_ inference rule.

**Projectivity**

172. Suppose, we have two entities “shape” and “triangle”, then which of the following is a correct statement?

Shape is a super type entity and triangle is a sub type entity

173. The entity that is NOT dependent on any other entity for its existence is known as entity.

Strong

174. In cross reference matrix attributes are specified on

X axis

175. Which of the following clarify the semantics of a relationship?

Roles

176. How many types of cardinality is specified about relationship?

Four

177. Which of the following is NOT a component of a DFD?

Relationship between external entities

178. Which of the following provides us the facility to add properties of one entity to another entity automatically?

Inheritance

179. If there is more than one relationship between two entities and we mentioned role on the relationship link, then it is called:

Multiple relationships

180. Which one of the following is not an advantage of DBMS

Recovery procedure is difficult

181. In many to many relationship a third table is created for the relationship, which is also called as \_

**Associative entity type**

182. When one entity instance of another entity for its existence, then it is called

**Existence dependency**

183. Relational data model is widely accepted due to its mathematically proven foundation and its

**Simplicity**

184. If an entity “motor cycle” is associated with at most one “driver”, then the relationship between these two entities is:

**One to one**

185. Database application is a program which is used for performing certain operations such as

**Insertion of data, Extraction of data, Updating the data....All of above**

186. Creating the relationships between new tables and their predecessors through the use of foreign keys is general requirements of

**Second normal form**

187. In the domains of attributes of a relation are atomic, that is they consist of single units that cannot be broken down further.

**First normal form**

188. Atomicity is a feature of .

**1NF**

189. The \_\_\_\_\_ can see entire information structure of the database.

**DBA**

190. The attribute whose value is not stored in a database is known as attribute.

**Derived**

191. \_\_\_\_\_ are the structure defined for placing data in the attributes while designing physical design of database.

Data types

192. Suppose a relation has five columns and ten rows, select the correct option to identify cardinality of relation.

Ten....confirm

130. Suppose a relation has five columns and ten rows, select the correct option to identify cardinality of relation.  
a) Five  
b) Ten

193. Which feature of database provides conversion from inconsistent state of DB to a consistent state ensuring minimum data loss?

Recovery service

194. Consider an employee is allotted an organization vehicle, which might solely be driven by that employee. What kind of relationship exists between employee and company?

Many to one

195. Which of the following is a feature of CHECK constraint?  
used to enforce referential integrity

196. Which of the following semantic model is used for the graphical representation to the conceptual database design?

Entity Relationship (ER)

197. Suppose we have two entities department and University. Which type of dependency exists between these two entities?

Referential (Not sure)

198. \_\_\_\_\_ is the next stage of conceptual database design.  
logical database design

199. The \_\_\_\_\_ model is independent of any tool or data model.

**Entity Relationship (ER) (page 122)**

**200. While converting to 2NF all the columns are deleted from the original table except for the determinate which will serve as a \_\_\_\_\_ key.**

**Primary key**

**201. In relational algebra, which of the following operator is used to find tuples that are present in one relation but are not in another?**

**Union**

**202. Subject may be the prerequisite of another subject, is the example of \_\_\_\_\_ relationship.**

**Unary (google)**

**203. The maximum PL/SQL size of data type “VARCHAR2” in DBMS is bytes a)**

**32767**

**204. \_\_\_\_\_ is state of database where a “course” cannot be inserted in the table, because this course has not been registered to any “student”.**

**Insertion anomaly**

**205. The maximum PL/SQL size of data type “BLOB” in DBMS is**

**4 gigabytes**

**206. \_\_\_\_\_ states that in a relation no attributes of a primary key (PK) can have a null value.**

**Entity integrity constraint**

**207. A database state where deletion of the information about the student record deletes the course information as well is called \_\_\_\_\_.**

**Deletion anomaly**

**208. An entity type is**

**A coherent set of similar objects that we want to store data on (e.g. STUDENTS, COURSE, CAR)**

**209. External scheme evolves as user needs are \_\_\_\_\_ over the time.**

**Modified**

210. An attribute that is the collection of some other attribute(s) is known as \_ attributes.

**Composite**

211. Which of the following is correct regarding Dataflow diagram?

**Created at increasing levels of detail**

212. Used to represent the relationships among the external entities are those attributes that are a combination of two or more than two attributes.

**Composite attributes**

213. Controlling redundancy in a database management system DOES NOT help to

**avoid unauthorized access to data**

214. Which of the following concepts is applicable with respect to 3NF?

**Transitive dependency**

215. Identify the operation which is NOT one of the parts of the five basic set operations in relational algebra?

**Join**

216. Making a change to the conceptual schema of a database but not affecting the existing external schemas is an example of

**Logical data independence.**

217. For a third normal form we concentrate on relations with one \_\_\_\_\_ key, and we eliminate transitive dependencies

**Candidate**

218. The entity relationship diagram is used to graphically represent the database model.

**Conceptual**

219. The \_\_\_\_\_ determines that the link between two entities is optional or mandatory.

Minimum cardinality

220. Select the most appropriate statement. Which of the following data model is used to design logical database design?

Relational

221. \_\_\_\_\_ are used by database administrator in situations where we have large

number of records and wastage of small amount of space in each record can lead to loss of huge amount of data storage space.

Codes

222. An attribute in a relation is foreign key If the \_\_\_\_\_ key from one relation is used as an attribute in that relation.

Primary....confirm

The foreign key attribute, which is present as a primary key in another relation is called as home relation of foreign key attribute, so in EMP table the deptId is foreign key and its home relation is DEPT.

The foreign key attribute and the one present in another relation as primary key can

223. in data flow diagram which of the following operator is used, when data flows from a source process to one of the mentioned sinks?

Ring Sum Operator...confirm

o Ring Sum Operator:

This operator is used when data from a source process can flow to one of the mentioned sinks. For this purpose the symbol used is displayed in Figure: 11a and its presentation in

224. in data flow diagram which of the following operator is used, when data flows from a source process to all the connected sinks?

AND operator....confirm

o AND Operator:

This operator is used when data from a source process must flow to all the connected sinks. For this purpose the symbol used is displayed in Figure: 12a and its presentation in

225. The \_\_\_\_\_ rule specifies that an entity can be a member of only one subtype at a time.

disjoint page no 108

226. In relational algebra, which of the following Greek is used to represent the project operator?

$\Pi$  (PI) (page no 150)

227. In three level architecture of the database, which level is helpful to display data which is not actually placed in the database?

External Level....confirm

**External View** (Level, Schema or Model):

External views are also helpful when we want to display the data which is not place in the database or not stored at all.

228. Which of the following normal forms deals multivalued dependency?

4NF

229. While mapping \_\_\_\_\_ relationship, primary key of the entity type against one side of relationship will be included in the entity type on the many side of the relationship as foreign key.

one to many

230. In data flow diagram \_\_\_\_\_ symbol is used to represents data flows.

arrow (google)

231. In the below entity type which attribute can work as an alternate key?

Employee (Reg# Primary key,Name,CNIC,designation,address,phone)

CNIC

232. From the following who is responsible for proper working of the database and DBMS?

Database administrator

233. Suppose that “Program” is an entity which represents different courses are being offered by an institute, like MCS. BCS etc. Following are the major attributes of this entity. Identify the attribute which can be used as primary key.

program\_ Code

234. The following is an of \_\_\_\_\_ inference rule.

If stId → prName and prName → credits then stId → credits

Transitivity

235. Which of the following is the example of a weak entity?

The tire can exist without being attached to the car

236. In Relational Data Model, the \_\_\_\_\_ of a relation is the number of columns in that relation

Degree

237. Which of the following normal form deals with possible loss less decompositions?

5NF

238. Suppose we have a list of bank customers which availed loan facility from different branches of a bank. If we want to get a list of all customers and the amount of the loans excluding branch name. Which of the following operation allows us to generate this relation?

Union

239. Which of the following provides us data independency?

3-Level Architecture

240. Cross reference matrix is a tool available in the data dictionary, which helps us in finding \_\_\_\_\_.

Entities of the database and their associations

241. Conceptual database design, expressed in E-R data model can be implemented using any \_\_\_\_\_.

Database Management System (DBMS)

242. Which of the following might be represented with a single-valued attribute?

Computer's processor speed

243. Which of the following is a Unary operation?

Project

193. Which of the following data model has the following components?

1. Construct (Tables)
2. Manipulation language (SQL)
3. integrity constraints?

**Relational**

244. Which of the following Greek is used to represent the select operation?

$\sigma$  (page 149)

245. Data about data is called metadata.

True (page no 10)

246. Normalization of database is used to \_\_\_\_\_.

Eliminate redundancy

247. In student entity which of the following is the best example of multivalued attribute?

Email address

248. Theta join is denoted mathematically as

$R \bowtie_{\theta} S$

249. In which normal form, we convert composite attributes into individual attributes?

1NF

250. Which of the following tool is used to graphically represent the components and design of the system?

DFD

251. Data type “BLOB” in database management system stands for \_\_\_\_\_.

Binary Large Object (page no 185)

252. In a three level DBMS architecture, the \_\_\_\_\_ level interacts directly with the users.

External

253. In the Database System \_\_\_\_\_ provides many features to ensure the data integrity.

Database Management System (DBMS)

basis of information produced from the database and the wrong information leads to wrong decisions, and business collapse. In the database system environment, DBMS provides many features to ensure the data integrity, hence provides more reliable data

254. The process of defining one or more subtypes of a super and forming relationships is known as

Generalization

255. External scheme evolves, as user needs are \_\_\_\_\_ over the time.

Switch

256. \_\_\_\_\_ of the Database is most difficult to change when a database is created.  
Extension

257. In Database development process, which phase involves designing of any new application for the enhancement of the system?  
Maintenance phase....confirm

258. An attribute of one entity is associated with another attribute of the same entity is called:

Recursive relationship....confirm

**Recursive Relationship:**

This is the situation when any attribute of one entity is associated with another attribute of the same entity. Such a link initiates from one entity and terminates on the same entity.

259. Relational Data model is created in the \_\_\_\_\_ phase.  
Analysis

260. Database system saves lots of efforts and finances, this advantage provides \_\_\_\_\_.  
Time saving structure

261. Data inconsistency leads to a number of a problems such as \_\_\_\_\_.  
loss of information and incorrect results

262. Every primary key (PK) will be a candidate key and every candidate key will be a \_\_\_\_\_.  
Super key...confirm

also holds between candidate and primary keys, that is, every primary key (PK) is a candidate key and every candidate key is a super key.

263. If we want to include attributes from two different relations, which of the following operations will be used?  
Union

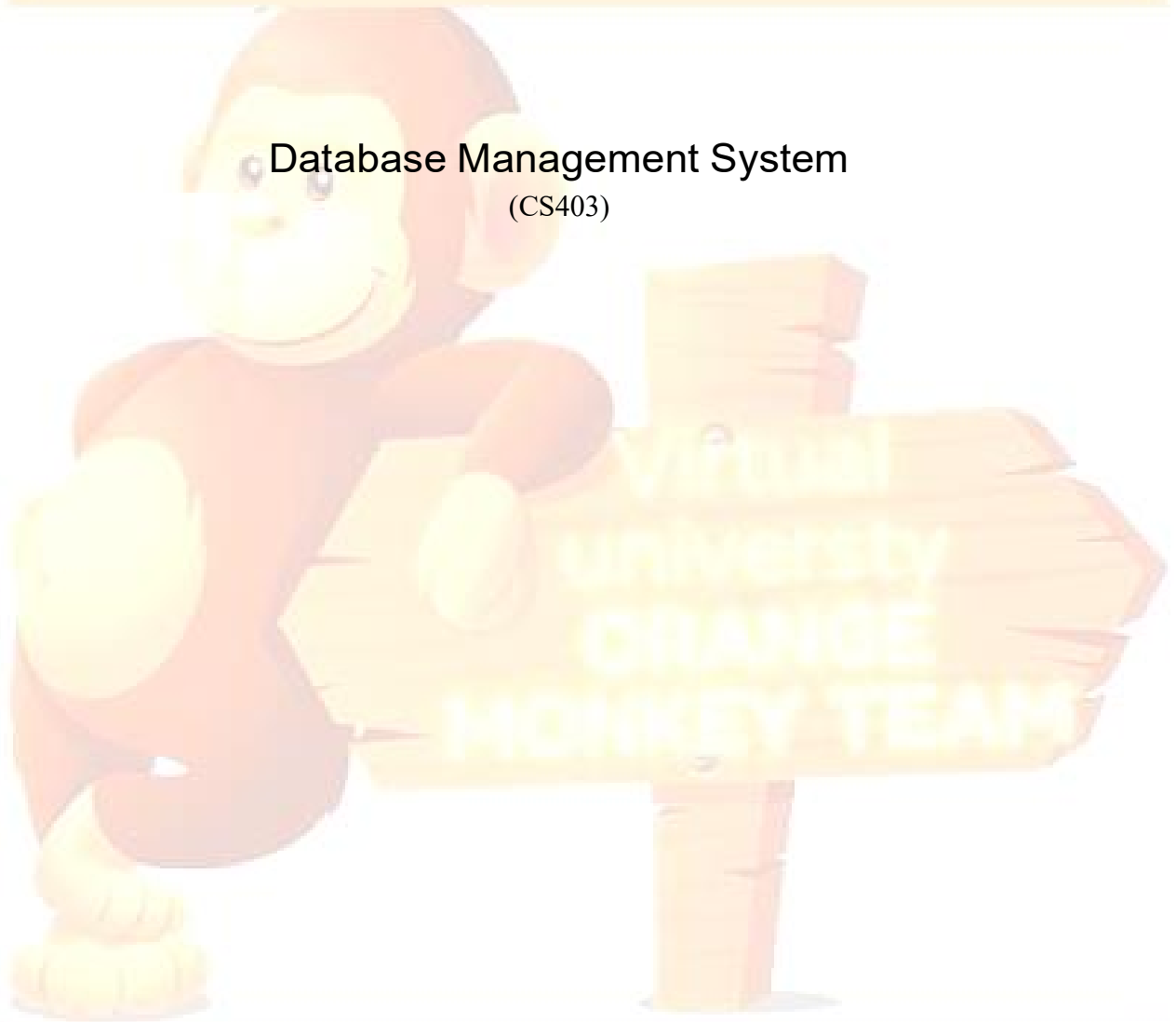
264. When have two entities "Person" and "Doctor. Which of all is best suitable type of relationship between the mentioned entities?  
Inheritance.....confirm

We have two **Raise For Success** entities "Person" and "Doctor". Which of the following is best **suitable type of relationship** between the mentioned entities?

• **Inheritance Correct**



Database Management System  
(CS403)



DATABASE MANAGEMENT SYSTEM .....	1
LECTURE NO. 01.....	9
Reading Material.....	9
Overview of Lecture.....	9
Introduction to the course.....	9
Database definitions.....	10
Importance of the Databases.....	12
Databases and Traditional File Processing Systems.....	12
Advantages of Databases.....	15
LECTURE NO. 02.....	17
Reading Material.....	17
Overview of Lecture.....	17
Difference between Data and Information.....	17
Further Advantages of Database Systems.....	19
Cost Involved.....	21
Importance of Data.....	22
Levels of Data.....	22
Users of Database Systems.....	24
LECTURE NO. 03.....	31
Reading Material.....	31
Overview of Lecture.....	31
Database Architecture.....	31
The Architecture.....	33
External View (Level, Schema or Model).....	35
Conceptual or Logical View.....	37
LECTURE NO. 04.....	40
Reading Material.....	40
Overview of Lecture.....	40
Internal or Physical View / Schema.....	40
Data Independence.....	43
Functions of DBMS.....	45
LECTURE NO. 05.....	50
Reading Material.....	50
Overview of Lecture.....	50
Database Development Process.....	51
Preliminary Study.....	51
Database Development Process: Approach 2.....	54
Tools Used for Database System Development.....	56
Data Flow Diagrams.....	56
Types of DFD.....	60
LECTURE NO. 06.....	63
Reading Material.....	63
Overview of Lecture.....	63
Detailed Data Flow Diagram.....	63
Data Dictionary.....	64
Database Design Phase.....	67

Data Model .....	68
Types of Data Models .....	68
Types of Database Design .....	69
LECTURE NO. 07 .....	70
Reading Material .....	70
Overview of Lecture .....	70
Entity-Relationship Data Model .....	70
The Entity .....	71
Classification of entity types .....	74
Attribute .....	75
Types of Attributes .....	77
Summary .....	79
Exercises .....	79
LECTURE NO. 08 .....	80
Reading Material .....	80
Overview of Lecture .....	80
Attributes .....	80
The Keys .....	80
LECTURE NO. 09 .....	85
Reading Material .....	85
Overview of Lecture .....	85
Relationships .....	85
Types of Relationships .....	87
LECTURE NO. 10 .....	91
Reading Material .....	91
Overview of Lecture .....	91
Roles in Relationships .....	95
Dependencies .....	97
Enhancements in E-R Data Model .....	98
Super-type and Subtypes .....	98
Summary .....	99
LECTURE NO. 11 .....	100
Reading Material .....	100
Overview of Lecture .....	100
Inheritance Is .....	100
Super types and Subtypes .....	101
Specifying Constraints .....	103
Completeness Constraint .....	103
Disjointness Constraint .....	104
Subtype Discriminator .....	108
LECTURE NO. 12 .....	110
Reading Material .....	110
Overview of Lecture .....	110
Steps in the Study of system .....	110
LECTURE NO. 13 .....	118
Reading Material .....	118
Overview of Lecture .....	118

Identification of Entity Types of the Examination System .....	118
Relationships and Cardinalities in between Entities .....	120
Conceptual Database Design.....	122
Logical Database Design.....	122
Conclusion .....	123
LECTURE NO. 14.....	124
Reading Material.....	124
Overview of Lecture.....	124
Relational Data Model.....	125
Introduction to the Relational Data model .....	126
Mathematical Relations.....	129
Database Relations.....	130
Summary.....	130
Exercise .....	131
LECTURE NO. 15.....	132
Reading Material.....	132
Overview of Lecture.....	132
Database and Math Relations .....	132
Degree of a Relation .....	133
LECTURE NO. 16.....	140
Reading Material.....	140
Overview of Lecture.....	140
Mapping Relationships.....	140
Binary Relationships .....	140
Unary Relationship.....	144
Data Manipulation Languages.....	146
Relational Algebra .....	147
Exercise .....	147
LECTURE NO. 17.....	148
Reading Material.....	148
Overview of Lecture.....	148
The Project Operator.....	150
LECTURE NO. 18.....	157
Reading Material.....	157
Overview of Lecture.....	157
Types of Joins .....	157
Theta Join .....	157
Equi-Join .....	159
Natural Join.....	159
Outer Join .....	161
Semi Join .....	161
Relational Calculus .....	162
Tuple Oriented Relational Calculus .....	162
Domain Oriented Relational Calculus.....	162
Normalization.....	162
LECTURE NO. 19.....	164
Reading Material.....	164
Overview of Lecture.....	164

Functional Dependency .....	164
Inference Rules .....	166
Normal Forms.....	166
Summary.....	167
Exercise .....	168
LECTURE NO. 20.....	169
Reading Material.....	169
Overview of Lecture.....	169
Second Normal Form.....	169
Third Normal Form.....	171
Boyce - Codd Normal Form.....	173
Higher Normal Forms.....	175
Summary.....	175
Exercise .....	175
LECTURE NO. 21.....	176
Reading Material.....	176
Overview of Lecture.....	176
Normalization Summary .....	176
Normalization Example .....	177
Physical Database Design.....	181
Summary.....	182
LECTURE NO. 22.....	183
Overview of Lecture.....	183
The Physical Database Design Considerations and Implementation.....	183
DESIGNING FIELDS .....	184
CODING AND COMPRESSION TECHNIQUES .....	185
LECTURE NO. 23.....	187
Reading Material.....	187
Overview of Lecture.....	187
Physical Record and De-normalization .....	187
Partitioning.....	187
Physical Record and Denormalization.....	187
Denormalization Situation 1.....	188
Partitioning.....	189
LECTURE NO. 24.....	191
Reading Material.....	191
Overview of Lecture.....	191
Vertical Partitioning .....	191
Replication .....	192
Reduced training cost .....	194
MS SQL Server .....	194
LECTURE NO. 25.....	196
Reading Material.....	196
Overview of Lecture.....	196
Rules of SQL Format.....	196
Data Types in SQL Server.....	197
Summary.....	200
Exercise .....	200

LECTURE NO. 26.....	201
Reading Material.....	201
Overview of Lecture.....	201
Categories of SQL Commands.....	201
Summary.....	205
Exercise.....	205
LECTURE NO. 27.....	206
Reading Material.....	206
Overview of Lecture.....	206
Alter Table Statement.....	206
LECTURE NO. 28.....	210
Reading Material.....	210
Select Statement.....	211
Attribute Alias.....	213
LECTURE NO. 29.....	215
Reading Material.....	215
Overview of Lecture.....	215
Data Manipulation Language.....	215
LECTURE NO. 30.....	220
Reading Material.....	220
Overview of Lecture.....	220
ORDER BY Clause.....	220
Functions in SQL.....	221
GROUP BY Clause.....	222
HAVING Clause.....	223
Cartesian Product.....	224
Summary.....	225
LECTURE NO. 31.....	226
Reading Material.....	226
Overview of Lecture.....	226
Inner Join.....	226
Outer Join.....	228
Semi Join.....	230
Self Join.....	231
Subquery.....	232
Summary.....	236
Exercise.....	237
LECTURE NO. 32.....	238
Reading Material.....	238
Overview of Lecture.....	238
Application Programs.....	238
User Interface.....	239
Forms.....	240
Tips for User Friendly Interface.....	243
LECTURE NO. 33.....	246
Reading Material.....	246
Overview of Lecture.....	246

LECTURE NO. 34.....	255
Reading Material.....	255
Overview of Lecture.....	255
LECTURE NO. 35.....	260
Reading Material.....	260
Overview of Lecture.....	260
File Organizations .....	260
LECTURE NO. 36.....	265
Reading Material.....	265
Overview of Lecture.....	265
Hashing .....	265
Hash Functions.....	266
Hashed Access Characteristics .....	266
Mapping functions .....	266
Open addressing.....	269
LECTURE NO. 37.....	270
Reading Material.....	270
Overview of Lecture.....	270
Index.....	270
Index Classification .....	272
Summary.....	274
LECTURE NO. 38.....	275
Reading Material.....	275
Overview of Lecture.....	275
Ordered Indices.....	275
Clustered Indexes .....	275
Non-clustered Indexes.....	276
Dense and Sparse Indices .....	276
Multi-Level Indices.....	277
LECTURE NO. 39 AND 40 .....	280
Reading Material.....	280
Overview of Lecture.....	280
Views.....	280
To Focus on Specific Data .....	280
Characteristics /Types of Views.....	283
Characteristics of Views .....	286
LECTURE NO. 41.....	288
Reading Material.....	288
Overview of Lecture.....	288
Updating Multiple Tables .....	288
Materialized Views .....	289
Transaction Management.....	291
LECTURE NO. 42.....	293
Reading Material.....	293
Overview of Lecture.....	293
The Concept of a Transaction.....	293
Transactions and Schedules .....	294

Concurrent Execution of Transactions .....	295
Serializability.....	296
Lock-Based Concurrency Control .....	297
Deadlocks .....	299
LECTURE NO. 43.....	302
Reading Material.....	302
Overview of Lecture.....	302
Incremental Log with Deferred Updates.....	302
Incremental Log with Immediate Updates.....	305
Concurrency Control.....	307
Summary.....	308
LECTURE NO. 44.....	310
Reading Material.....	310
Overview of Lecture.....	310
Uncommitted Update Problem .....	310
Inconsistent Analysis .....	311
Serial Execution .....	312
Serializability.....	315
Locking.....	315
Summary.....	316
LECTURE NO. 45.....	318
Reading Material.....	318
Overview of Lecture.....	318
Locking Idea .....	319
DeadLock.....	320
DeadLock Handling.....	320
Wait – for Graph.....	320
Deadlock Resolution .....	323
Timestamping rules.....	324



## Lecture No. 11

### Reading Material

“Database Systems Principles, Design and Implementation” written by Catherine Ricardo, Maxwell Macmillan.	
--	--

### Overview of Lecture

- Inheritance
- Super type
- Subtypes
- Constraints
- Completeness
- Disjointness
- Subtype Discrimination

According to the Microsoft Dictionary of Computing

#### Inheritance Is

The transfer of the characteristics of a class in object-oriented programming to other classes derived from it. For example, if “vegetable” is a class, the classes “legume” and “root” can be derived from it, and each will inherit the properties of the “vegetable” class: name, growing season, and so on<sup>2</sup>. Transfer of certain properties such as open files, from a parent program or process to another program or process that the parent causes to run.

Inheritance in the paradigm of database systems we mean the transfer of properties of one entity to some derived entities, which have been derived from the same entities.

## Super types and Subtypes

Subtypes hold all the properties of their corresponding super-types. Means all those subtypes which are connected to a specific supertype will have all the properties of their supertype.

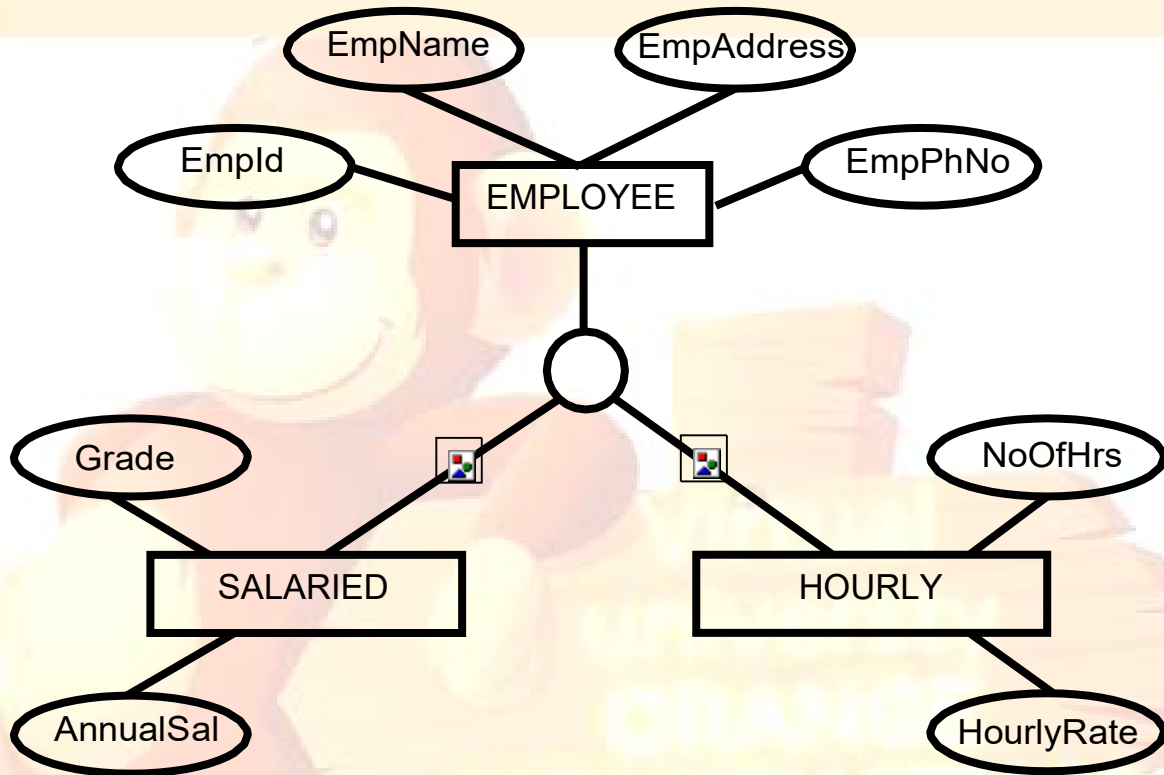


Fig-1 a

The Figure:1 above shows that the supertype and subtype relation between the SALARIED and HOURLY employees with the supertype entity EMPLOYEE, we can see that the attributes which are specific to the subtype entities are not shown with the supertype entity. Only those attributes are shown on the supertype entity which are to be inherited to the subtypes and are common to all the subtype entities associated with this supertype.

The example shows that there is a major entity or entity supertype name EMPLOYEE and has a number of attributes. Now that in a certain organization there can be a number of employees being paid on different payment criteria.

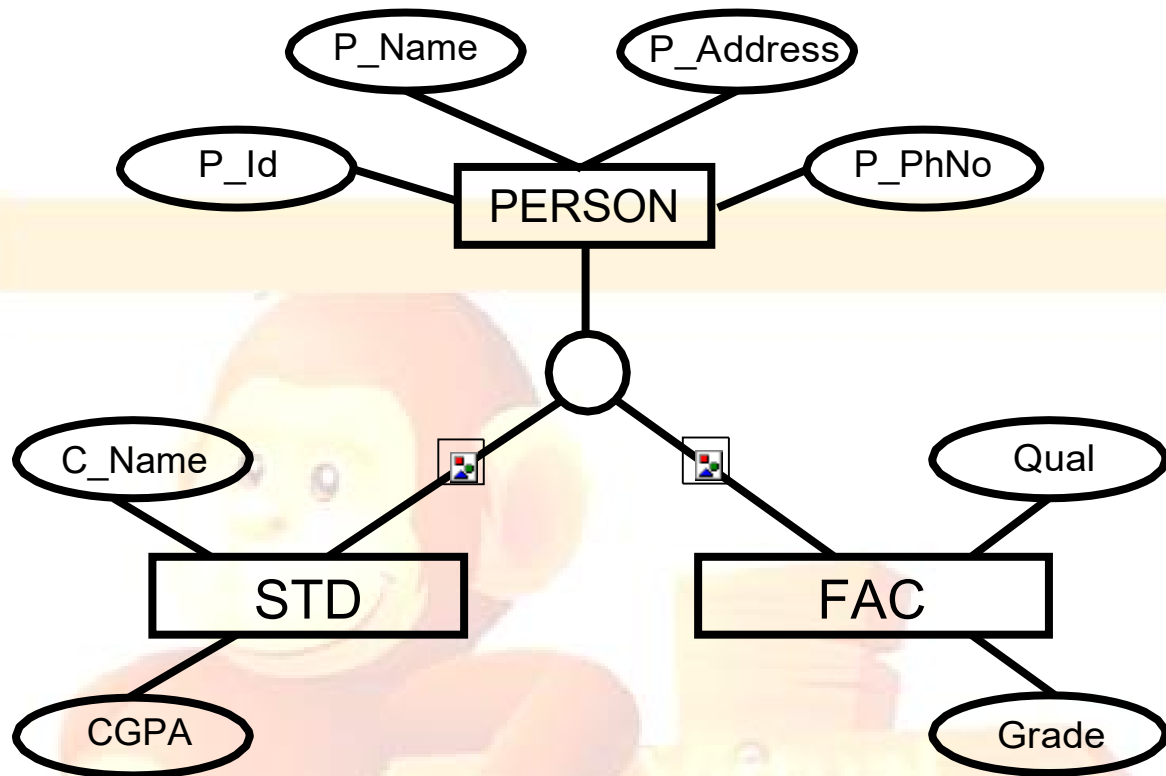


Fig – 1 b

The second example is that of student and the Faculty members who are at the super level same type of entities. Both the entities at the super level belong to the same entity of type Person. The distinct attributes of the student and faculty members are added later to the sub entities student and fac.

#### Supertype / subtype Relationship:

The use of supertype and subtype for the entities is very useful because it allows us to create hierarchy of the entities according to the attributes they have and we need not to write all the attributes again and again. We can group similar types of entities and the attributes associated with those entities at certain levels.

This also adds clarity to the definitions of the entities as it is not necessary to write the attribute again and again for all the entities.

Moreover it also eases the operation of removing or adding attributes from the entities, here it is worth noting that adding an attribute at the super entity level will add the

attribute to the below listed or derived sub entities and removing the attribute will remove the attribute from the entities at sublevels in the same way.

The process of identifying supertype and creating different type of sub entities is supported by the general knowledge of the designer about the organization and also based of the attributes of the entities which are entities existing in the system..

### Specifying Constraints

Once there has been established a super/sub entity relationship there are a number of constraints which can be specified for this relationship for specifying further restrictions on the relationship.

### Completeness Constraint

There are two types of completeness constraints, partial completeness constraints and total completeness constraints.

#### Total Completeness:

Total Completeness constraint exist only if we have a super type and some subtypes associated with that supertype, and the following situation exists between the super type and subtype.

All the instances of the supertype entity must be present in at one of the subtype entities, i.e.—there should be not instance of the supertype entity which does not belong to any of the subtype entity.

This is a specific situation when the supertype entities are very carefully analyzed for their associated subtype entities and no sub type entity is ignored when deriving sub entities from the supertype entity.

#### Partial Completeness Constraint:

This type of completeness constraint exists when it is not necessary for any supertype entity to have its entire instance set to be associated with any of the subtype entity.

This type of situation exists when we do not identify all subtype entities associated with a supertype entity, or ignore any subtype entity due to less importance or least usage in a specific scenario.

## Disjointness Constraint

This rule or constraint defines the existence of a supertype entity in a subtype entity. There exist two types of disjoint rules.

- Disjointness rule
- Overlap rule

### Disjoint constraint:

This constraint restricts the existence of one instance of any supertype entity to exactly one instance of any of the subtype entities.

Considering the example given in Fig 1a it is seen that there can be two types of employees, one which are fixed salary employees and the others are hourly paid employees. Now the disjoint rule tells that at a certain type an employee will be either hourly paid employee or salaried employee, he can not be placed in both the categories in parallel.

### Overlap Rule:

This rule is in contrast with the disjoint rule, and tells that for one instance of any supertype entity there can be multiple instances existences of the of the instance for more than one subtype entities. Again taking the same example of the employee in an organization we can say that one employee who is working in an organization can be allowed to work for the company at hourly rates also once he has completed his duty as a salaried employee. In such a situation the employee instance record for this employee will be stored in both the sub entity types.

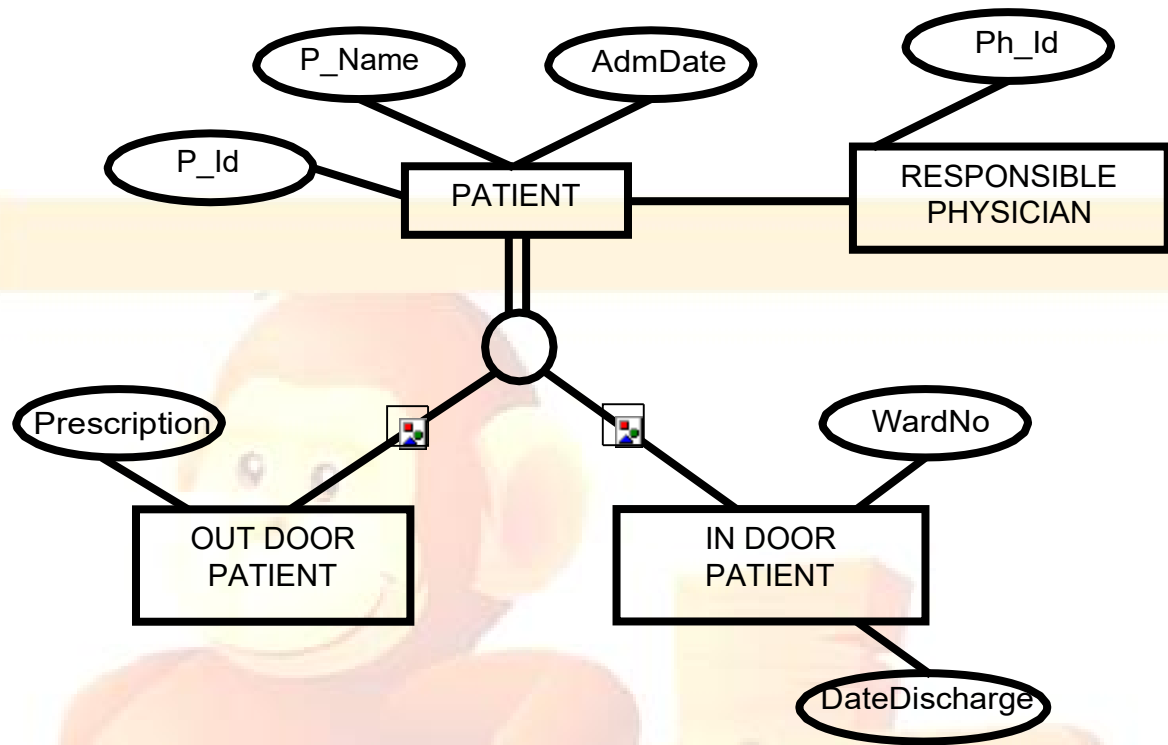


Fig 2-a

In the example the completeness of the relation is shown between the supertype entity and the subtype entity, it shows that for the data of patients we can have only two type of patients and one patient can be either an outdoor patient or indoor patient. In it we can see that we have identified all possible subtypes of the supertype patient. This implies a completeness constraint. One more thing to note here is the linked entity physician to the patient entity. And all the relationships associated with the supertype entity are inherited to subtype entities of the concerned supertype.

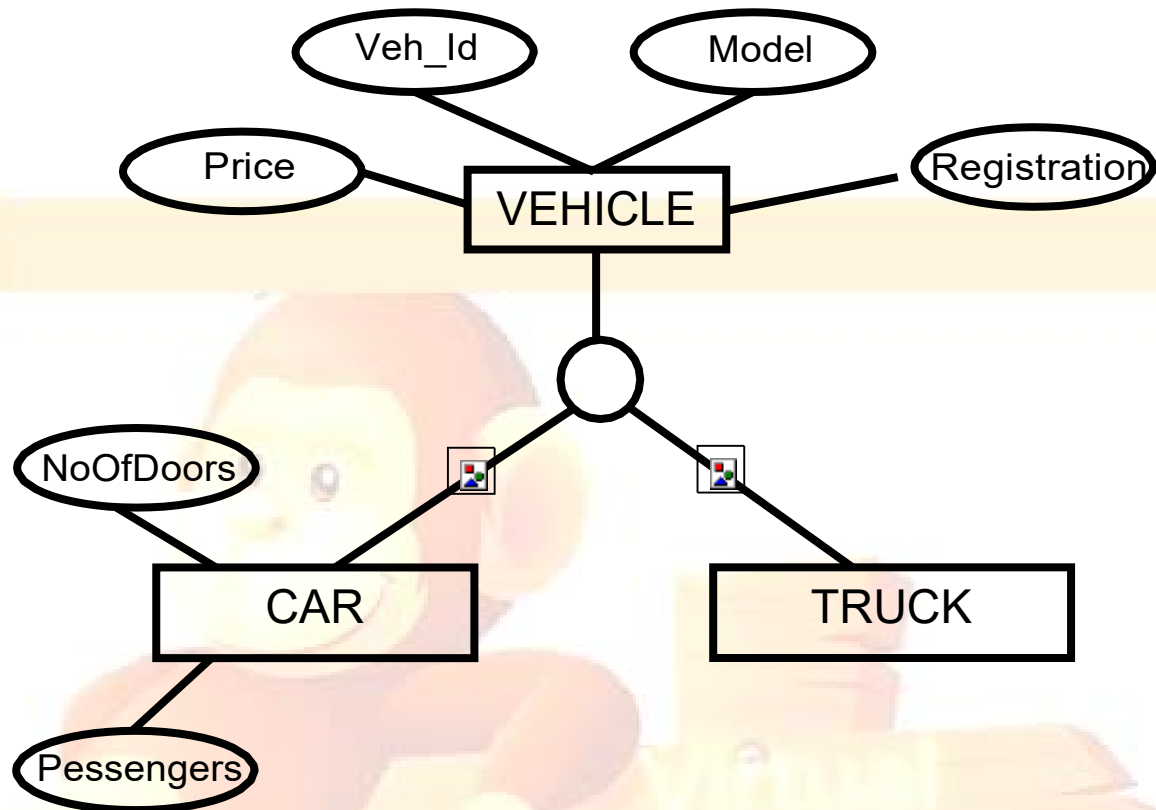


Fig 2-b

The Figure 2b shows the supertype and subtype relationship among different type of vehicles. Here we can see that the Vehicle has only two subtypes, known as Truck and Car, As it is normal to have a number of other vehicles in the company of a certain type but when we have noted just a limited number of vehicles then it means that we are not interested in storing information for all the vehicles as separate entities. They may be stored in the vehicle entity type itself and distinct vehicle may be stored in the subtypes car and truck of the Vehicle.

This is a scenario where we have the freedom to store several entities and neglect others, and it is called as partial completeness constraint rule.

After the discussion of the Total Completeness and Partial completeness let us move to the next constraint that is disjointness and check for its examples.

Again in the Figure 2-a. we have the environment where patient entity type has two subtypes indoor and outdoor patient. To represent disjointness we place the letter "D" in the circle which is splitting the super entity type into two sub entity types. Suppose that the hospital has placed a restriction on the patient to be either a n indoor patient or

outdoor patient, in such a case there exists disjointness which specifies that the patients data can not be place in the database in both the subtype entities. It will be wither indoor or outdoor.

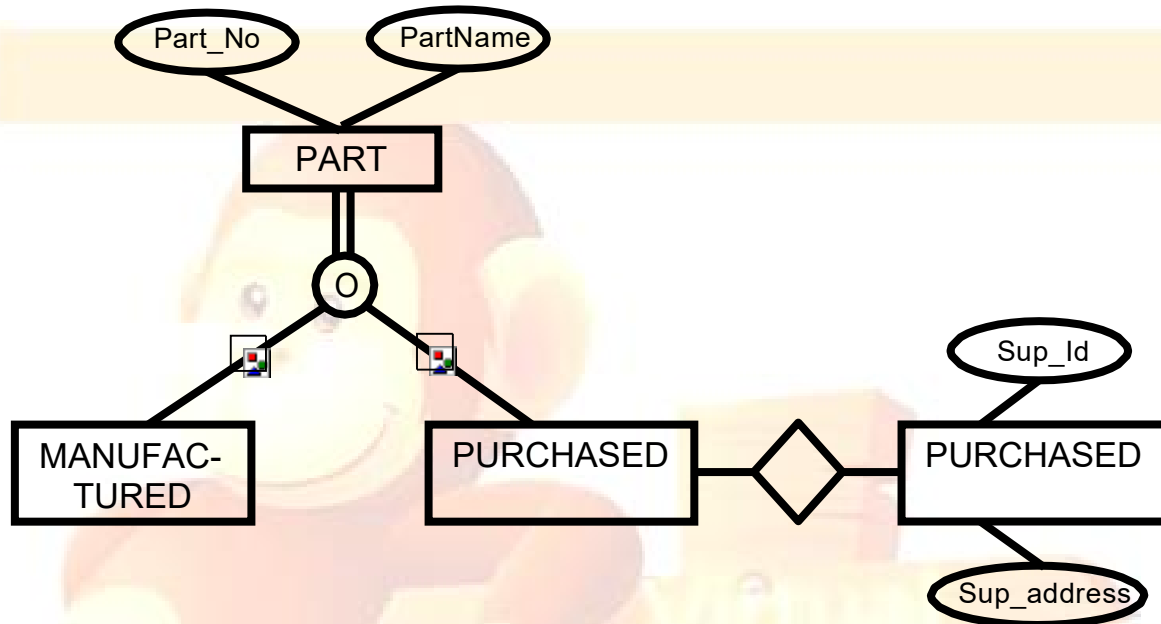


Fig- 3

The figure 3 above shows the second type of disjoint constraint which tells that the entity subtype instance can be repeated for any single entity supertype instance. We can see the relationship of a certain hardware company for the parts provided by the company to its clients. Now there may exist an overlapping situation for a certain part which is to be provided to a certain firm, but the manufactured quantity of that part is not enough to meet the specific order, In this situation the company purchases the remaining the deficient number of parts form the other suppliers. We can easily say that the data for that specific part is to be placed in both the entity subtypes. Because it belongs to both the subtype entities, this is an overlapping situation and expresses disjointness with overlapping. Another important thing which is to be noted here that the purchased part subtype entity has a relationship with another entity where the data for the suppliers is stored from whom the parts are bought. Now this relation does not have nay interaction with the manufactured parts relation as it is not connected with its supertype i.e.—parts supertype entity.

Considering the above discussed we can have four different types of combination existing for the supertype and subtype entities.

- Complete Disjoint
- Complete Overlapping
- Partial Disjoint
- Partial overlapping

### Subtype Discriminator

This is a tool or a technique which provides us a methodology to determine that to which subtype one instance of a supertype belongs.

To determine the relation we place an attribute in the entity supertype which can specify through its value, that to which entity subtype it belongs.

For example we consider the example

There can be two different situations which specify the placement or relationship of a supertype entity instance in a subtype entity instance. First situation is that of disjoint situation where one supertype entity instance can be placed only in one subtype of that supertype. Let us consider the example of vehicles above in Figure-2-b it show that there can be two different vehicles car and truck associated with the supertype vehicle now if we place an attribute named Vehicle\_type in the supertype we can easily determine the type of the associated subtype by placing a C for car and a T for truck instance of the vehicle.

The other situation where the Subtype discriminator is required the overlapping constraint; it is the situation where one supertype attribute can be placed in more than one subtype entities.

Considering again the part example shown in Figure 3, which has an overlapping constraint; In this situation we can have many solution one common solution is to place two attribute in the supertype one for manufactured and other one for purchased. We can combine them as a composite attribute, when we place Y for manufacture and N for

Purchased then it means the part is manufactured by the company, and similarly the following situation will give us further information

Attribute

Manufacture	Purchased	Result
Y	Y	Manufacture Purchased
Y	N	Manufactured
N	Y	Purchased.

Significance of Subtype Discriminator:

Existence of subtype discriminator helps us a lot in finding the corresponding subtype entities, although we can find a subtype entity instance without having a subtype discriminator in the supertype but that involves lots of efforts and might consume a huge time in worst case situations.

This concludes out discussion of The ER Model in the course.

## Lecture No. 12

### Reading Material

“Database Systems Principles, Design and Implementation” written by Catherine Ricardo, Maxwell Macmillan.	
--	--

### Overview of Lecture

In today's lecture we will discuss the ER Data model for an existing system and will go through a practice session for the logical design of the system

The system discussed is an examination section of an educational institute with the implementation of semester system.

### Steps in the Study of system

#### Preliminary study of the system

- Students are enrolled in programs.
- The programs are based on courses
- Different courses are offered at the start of the semester
- Students enroll themselves for these courses at the start of semesters
- Enrolled courses by students and offered courses must not be same.
- The difference is due to the individual situation of every student, because if one student has not pass a certain course 'A' in the previous semester he will not be able to register for a course 'B' offered in this semester as the course 'A' is the prerequisite for course 'B'.
- After valid registration classes start.
- A Course which is offered is assigned to a teacher also
- There can be any mid term exams and in this system we have only one mid term
- All the students are given assignments and quizzes and are awarded marks against their performance.
- Result of the student is prepared on the basis of assignment marks, sessional and mid term marks and the final exam.
- GP (Grade point) for students is calculated in each subject.
- Average grade point is calculated on the basis of GPs in individual subjects

- And the Cumulative GPA is calculated for all the passed semesters.

### Outputs Required

- Teachers and controller need class list or attendance sheet, class result; subject and overall
- Students need transcripts, semester result card, subject result

### Entities associated with the system

- Students
- Teachers
- Controllers

Once the analysis of the system is done it is important to make a draft of the system using a standard tool which specifies the component and design of the system. This design is useful because anyone using the design can work on the existing system and clearly understand the working without working on the system from the scratch.

Tool used for such graphical design is Data Flow Diagram (DFD)

In the Figure -1 of the system we have a context diagram of the system which shows integration of different entities with the examination system, these include Registration system, controller, student and teacher entities.

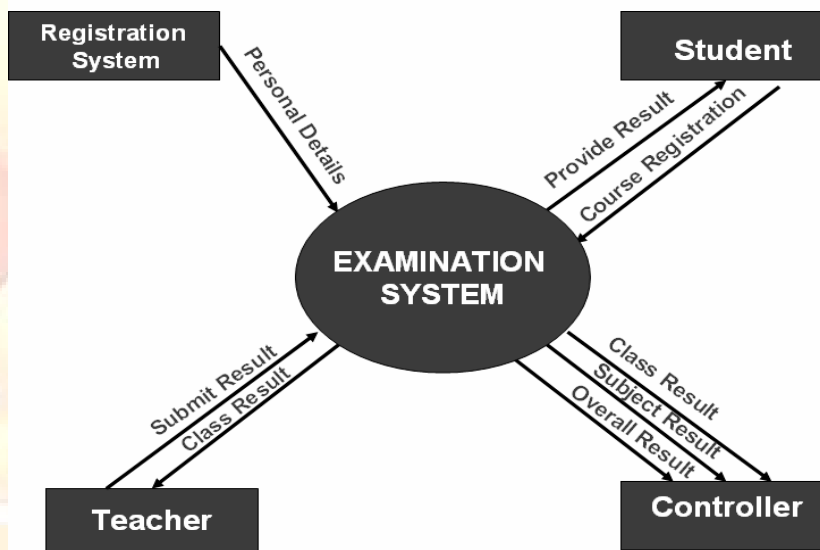


Fig-1

- From the diagram we can understand basic functionality of the system and can find how the data is flowing in the system and how different external entities are communicating or interacting with the system.

- First of all we have registration system, which provides the data of students to the systems once the registration process has been completed, this data is now free of errors in terms of validity of a certain student for a certain course or a semester.
- Second external entity interacting with the system is the teacher, a Teacher is given a list of students who are enrolled in a class and the registration system has declared them as valid students for that very course. Then the teacher allows those students in the class and continues the process of teaching the class, during this process the teacher takes test of the students and prepares papers for the students and also prepares quizzes to be submitted by students. All the data of students' attendance quizzes and assignments along-with different sessional results is then submitted by the teacher to the examination system which is responsible for preparation of results of the students
- Third interacting entity with the system is the controller's office it is provided with the semester overall result, subject results and also the result of each class fir performance evaluation and many other aspects.
- Fourth entity is student which externally interacts with the system for getting its result, the result is submitted to the student and may be in one of different forms such as, transcript and result card etc.

#### Level 0 Diagram

The three major modules which have been identified are given below our level 0 diagram will be based on these three modules and will elaborate and describe each of the modules in details.

- Subject registration
- Result submission
- Result calculation

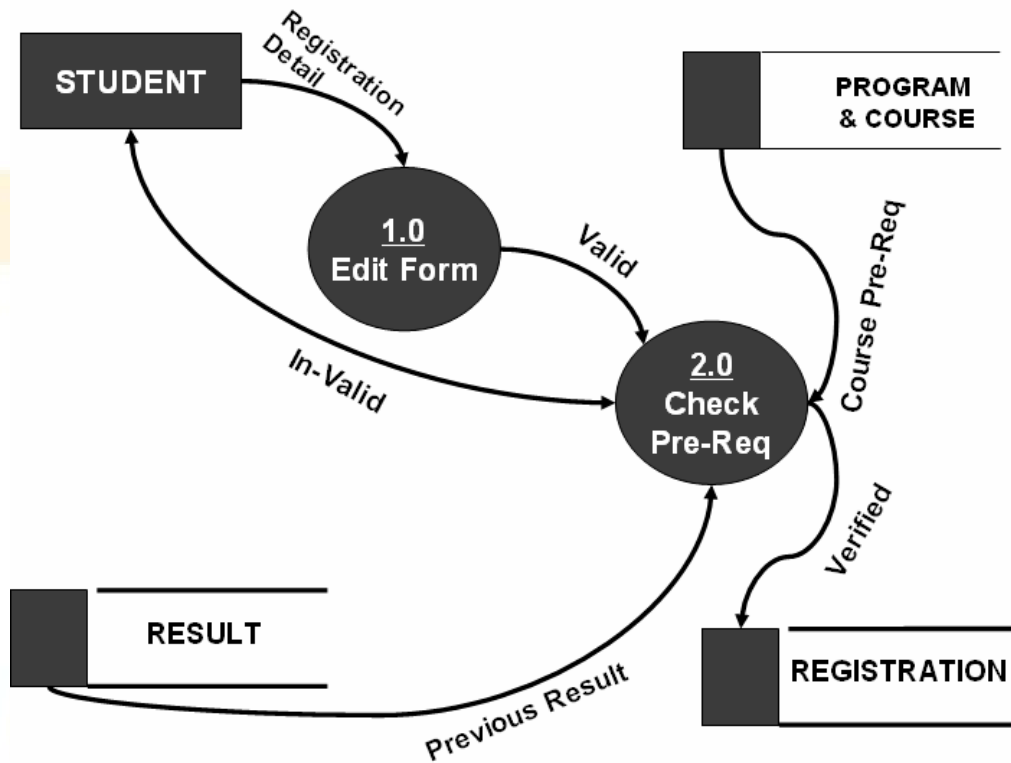


Fig 2

The first module identified in the system is the Registration of the students for the system. As the DFD shows, a student applies for registration along with certain registration information which is required by the system. Process 1.0 of the system checks the validity of information in the form. If the registration form is found to be valid, the information in the form is passed onto the second process where the validity of registration is determined by checking certain prerequisites for the courses to which the student wishes to be enrolled. After the prerequisite checking, the data of the student is stored in a registration database for use by other processes in the system.

During this process, the result of the students is also checked for the previous semester or previously studied subject to confirm whether the student has passed a certain prerequisite subject before he can attempt to enroll for a second course which is based on that prerequisite.

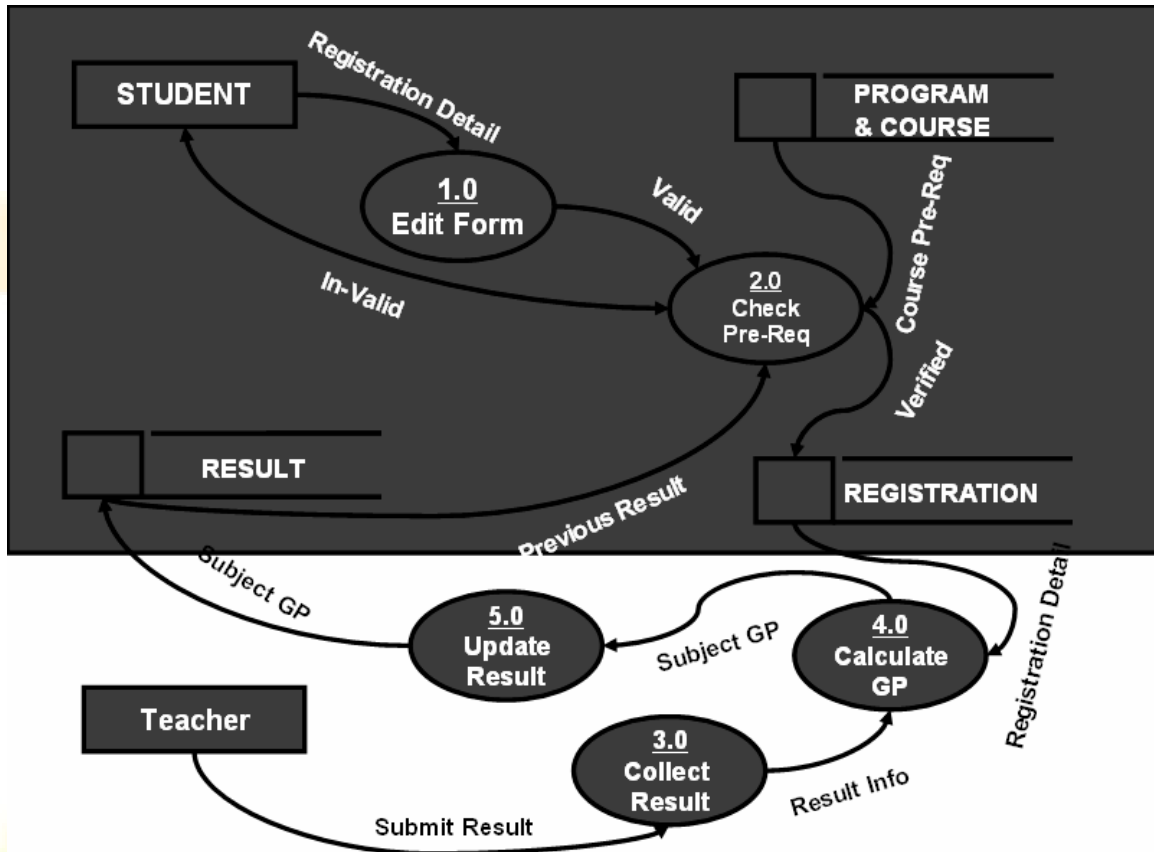


Fig-3

The Second DFD is in fact combination of the last diagram and some new details to the DFD this portion adds the result submission to the whole process of the system The teacher is the external entity here which is submitting the result, the result collection process is numbered 3.0, result is submitted by the teacher in parts, i.e. –separately for assignments, quizzes, tests, sessional and final result. The Collection process then forward the collected result to the Calculate GP Process, this process calculates the Grade point for the subject, the result with GP calculated is then moved forward to the update result process which then makes a change in the result data store by updating the result data for that specific student.

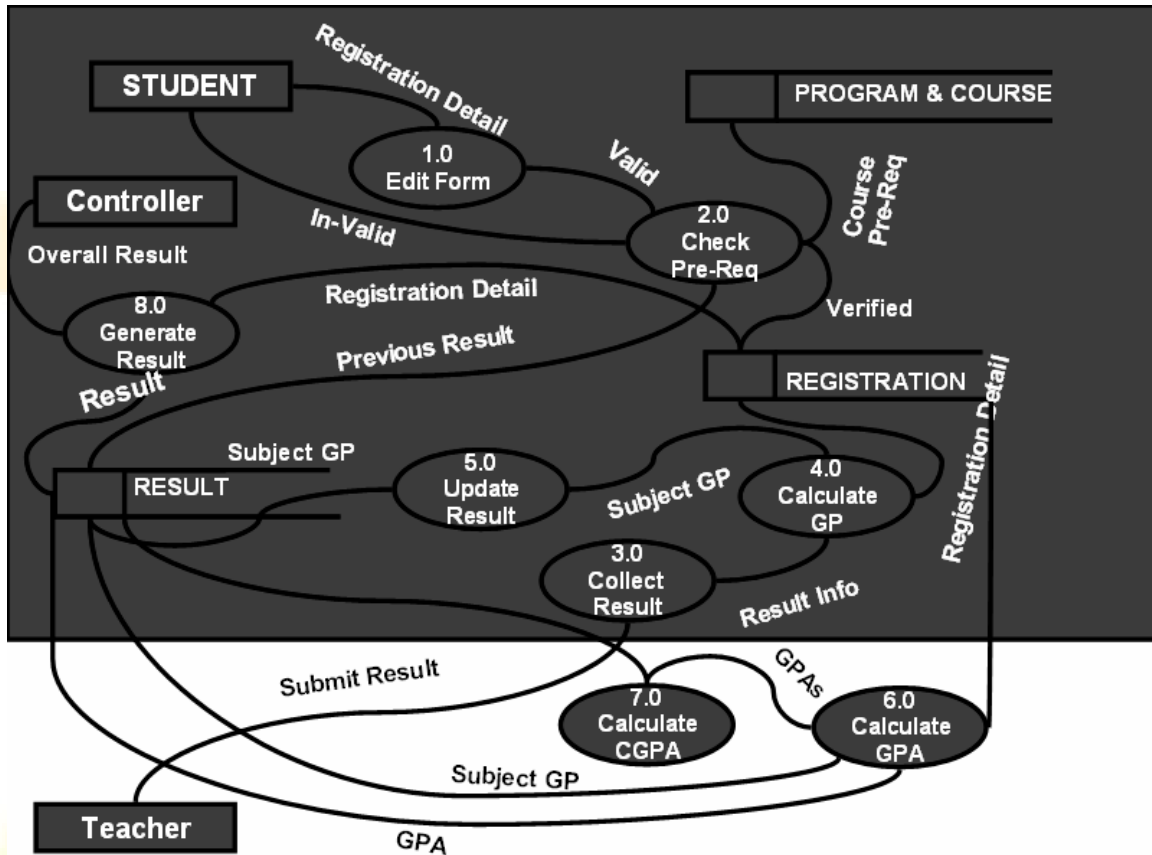


Fig-4

After the process of result submission the result for all the subjects is taken and the GPA is calculated, once the GPA is calculated the it is used for further calculation of CGPA and is forwarded to another process which is numbered 7.0 this process will calculate the CGPA by taking all the results of the current and previous semesters.

Further detailed diagram i.e.—Detailed DFD can be created using the given level 0 DFD and by expanding all the Processes further.

Cross Reference Matrix: doth:

This matrix is used to find out that what values or attributes will appear in which reports, for this purpose we write the major item names on a matrix in the row wise order and the reports which will be generated will be written on top or in column wise order.

## Cross Reference Matrix

	Transcript	Semester Result Card	Attendance Sheet	Class Result (Subject Wise)	Class Result
Course_Name	✓	✓		✓	
CGPA	✓	✓			✓
Date	✓	✓		✓	✓
F_Name	✓	✓			
NameOfStudent		✓			
NameOfProgram		✓			
Reg_No	✓	✓	✓	✓	✓

This process infact is just cross link So the first Item transcript which may be or it will be needed by a specific student, second is Result card, next is attendance sheet then we have Class result (Subject wise) and finally the Class result as a whole, here by subject wise class result means that all the results of a specific class for a specific student considering each component, such as assignments, quizzes, sessional and terminal results.

Similarly all the mentioned items are marked with a tick which may needed by a certain output.

Let us see how the DFD and CRM are used in creating the ER-Diagram

The process of Creating ER-Diagram in fact lies in the Analysis phase and is started with identifying different entities which are present in the system. For this purpose we can use the DFD first of all.

Lets check our DFD, from there we can find the following entities.

Student	Controller
Courses	Teachers
Courses Offered	Programs
Registration	Results
Semester	

Here the point to be noted is that, we have picked the controller as the entity, although the controller is acting as an external entity for providing or getting information from the system, but in case of ER-Diagram the controller can not be represented as an entity because there is only one controller in any examination system and for such an entity instances a complete entity is not used.

So in this way we can exclude the controller entity, we will also take care of other entities before including them in our ED-Diagram. Another such example is results, which may not be as it is, added to the ER-Diagram, because there can be a number of result types at different stages of the Process, so there will be a number of different results.

We use our CRM in creating the ER-Diagram, because when we see the CRM, it has a number of item/attributes appearing on it, now from there we can see that whether these items belong to the same entity or more than one entity. And even if they belong to multiple entities we can find the relationship existing between those entities.

Considering our CRM we have transcript, it has a number of items appearing on it, as we know that there is to appear result for each semester on the transcript. So the attributes which belong to the personal information of the student shall be placed in the student entity and the data which belongs to the students' academic data will be placed in the courses or results entity for that student.

In the next phase we have to draw different entity type and the relationship which exist between those entities.

***These we will discuss in the next lecture that how we draw relationships between different entities.***

## Lecture No. 13

### Reading Material

Case Study	
------------	--

### Overview of Lecture

- E – R Diagram of Examination System
- Conceptual Data Base Design
- Relationships and Cardinalities between the entities
- In the previous lecture we discussed the Preliminary phase of the Examination system. We discussed the outputs required from the system and then we drew the data flow diagrams DFDs. From this lecture we will start the conceptual model of the system through E-R Diagram.

### Identification of Entity Types of the Examination System

We had carried out a detailed preliminary study of the system, also drawn the data flow diagrams and then identified major entity types. Now we will identify the major attributes of the identities, then we will draw the relationships and cardinalities in between them and finally draw a complete E-R Diagram of the system.. So first of all we will see different attributes of the entities.

#### Program:

This entity means that what different courses are being offered by an institute, like MCS, BCS etc. Following are the major attributes of this entity:-

- pr\_Code. It can be used as a primary key of the entity as it would always be unique for example MBA, MCS, etc.
- max\_Dur It means that what is the maximum duration of any particular course , like 1 year , 2 years and so on.
- no\_of\_Semesters How many semesters this program has like four ,six and so on.
- Pr\_Lvl This course is of undergraduate, graduate or post graduate level.

Student:

Following are the major attributes of this entity:-

- Reg\_No This can be used as a primary key for this entity as it will be unique for every student.
- st\_Name This would be the first name of all the students of an institute.
- St\_Father\_name This would represent the father's name of a student.
- St\_date\_of\_Birth. The date of birth of all students including year , month and day.
- st\_Phone\_no
- st\_GPA This is a very important attribute. Now to know the GPA of any student, we need to know the student reg no and the particular semester. So this is a multi valued attribute as to know the GPA, different attributes values are required. So this represented by a relation, which will be discussed in the relationships in between entities.
- st\_Subj\_Detail This is also a multi valued attribute ,as to know the marks in mid terms and final papers , student reg no and the particular subject are required

Teacher:

Following are the major attributes of this entity: -

- teacher\_Reg\_No This can be used as a primary key for this entity as it will be unique for every teacher.
- teacher\_Name This would be the first name of all the teachers of an institute.
- teacher\_Father\_name This would represent the father's name of a teacher.
- Qual. The qualification of a teacher like Masters or Doctorate.
- Experience This can also be a multi-valued attribute or a single valued attribute. If only total experience of any teacher is required then it can be single valued, but if details are required as per the different appointments, then in that case it would be multi valued.
- teacher\_Sal The total salary of the teacher.

There is one thing common in between teacher and student an entity that is the personal details of both, like name, father's name and addresses.

Course:

Following are the major attributes of this entity: -

- course\_Code This can be used as a primary key for this entity as it will be unique for every course like CS-3207.
- course\_Name
- course\_Prereq This would also be a multi valued attribute as there can be multiple requisites of any course . For example, Networking can have pre-requisites of Operating System and Data Structures. In this case this is a

recursive relation as pre-requisite of a course is a course. We will treat it as a recursive relation here.

- Courses\_Offered\_in This is also a multi valued attribute as to know the courses offered , program and semester both are required so this can also be represented by a relation

Semester:

Following are the major attributes of this entity: -

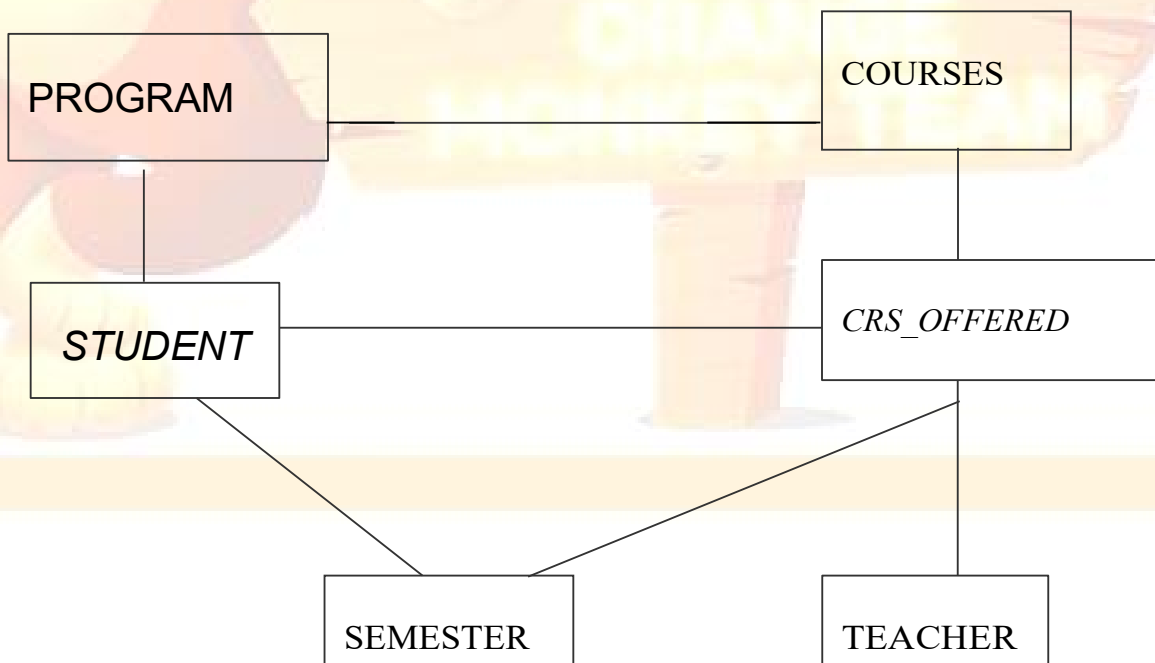
- semester\_Name This can be used as a primary key for this entity as it will be unique for every semester like fall 2003 or spring 2004.
- semester\_Start\_Date The starting date of the semester
- semester\_End\_Date The ending date of the semester

Derived Attributes

There are certain attributes in the examination system which is derived like CGPA of a student can only be achieved from the semesters GPA. Similarly FPA of any particular semester can be achieved from subjects GPA of the semester. So this has to be kept in mind while drawing the E-R Diagram of the system.

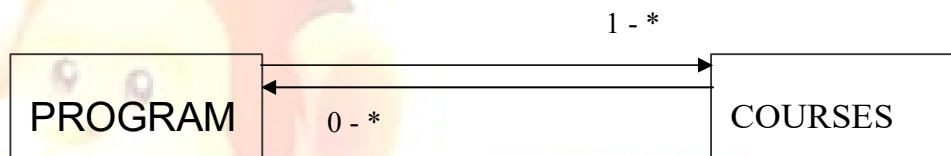
Relationships and Cardinalities in between Entities

Relationships and cardinalities in between entities is very important. We will now see the relationship of different entities one by one. The block diagrams of different entities are as under: -



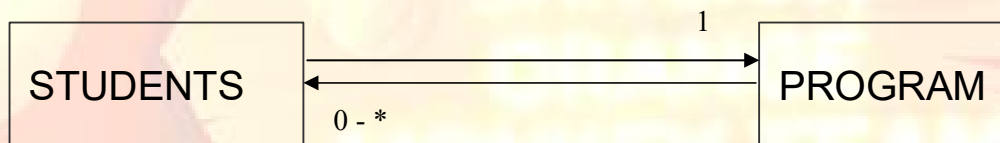
### Program and Courses

The relationship between program and courses is that, if we want to know the courses in any particular program then the course codes and program codes are required. The cardinality between program and courses is of one to many (1 - \*), which means that any program will have minimum one course, and as many as required for that particular program. The cardinality of courses and program can be zero to many (0 - \*). It means that if an institution wants, it can have a course without even any program as well. This course can be included in any program later on.



### Students and Programs

The cardinality in between student and program is one, which means that every student can have minimum and maximum one program at any time. The cardinality in between programs and students can be zero to many (0 - \*), which means that depending upon the requirements of any organization it can have a program which is presently not being offered to any students.



### Semester and Course

The relationship in between semester and course is many to many. But it is essential to know the course offered during any particular semester so there is a requirement of an attribute, which is of relationship and when it is many to many it, can also serve as entity which is represented by a diamond in a rectangle. So here this can be a courses offered attribute, which would also be an entity. The primary key of semester that is semester code and primary key of course that is course code, after combining it becomes composite key which would be used to identify any particular course.

### Course Offered and Teacher

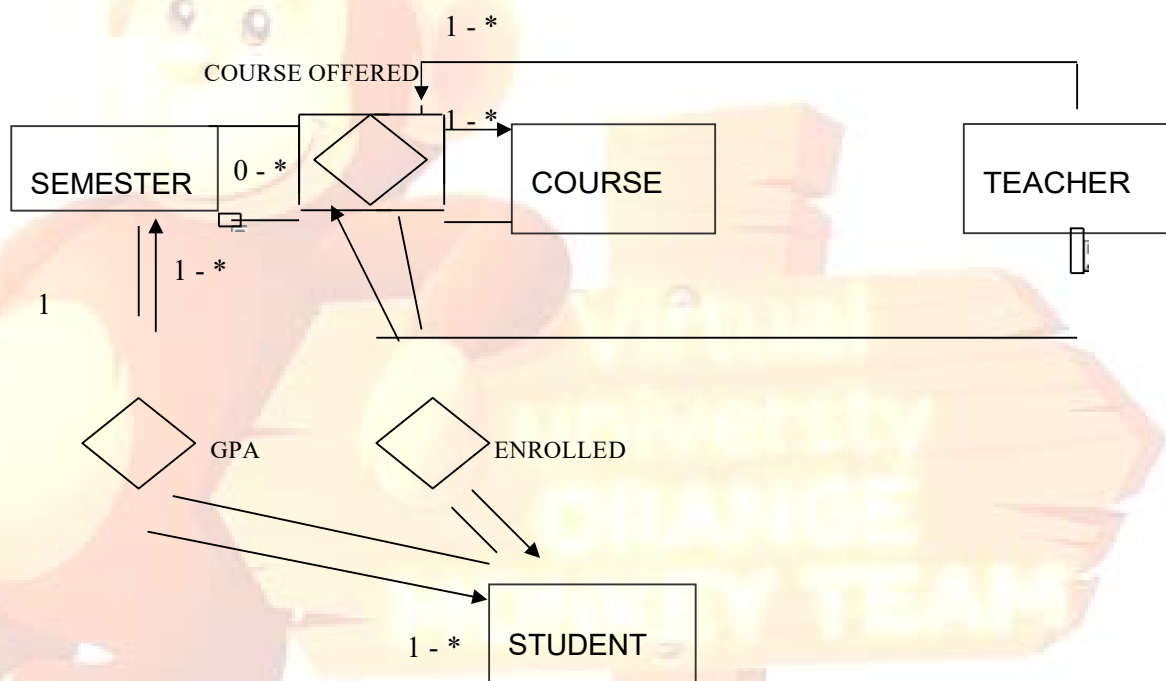
There is a relationship in between course offered and teacher. The cardinality of course offered and teacher is one that is a teacher can only have one course at a time. Similarly the cardinality in between teacher and course offered is one to many, which means that a teacher can teach many courses.

### Student and Course Offered

The relationship in between student and course offered is many to many so this relationship is also through enrolled attribute, which can also serve as entity type. The primary keys of semester, course and student are used as composite key to identify, that which student has got which course.

### Semester and Student

To find out GPA of any student the semester is also required to be known. So the relationship in between these two can be through result whose attribute GPA can be used. There is a many to many relationship in these two entities.



### Conceptual Database Design

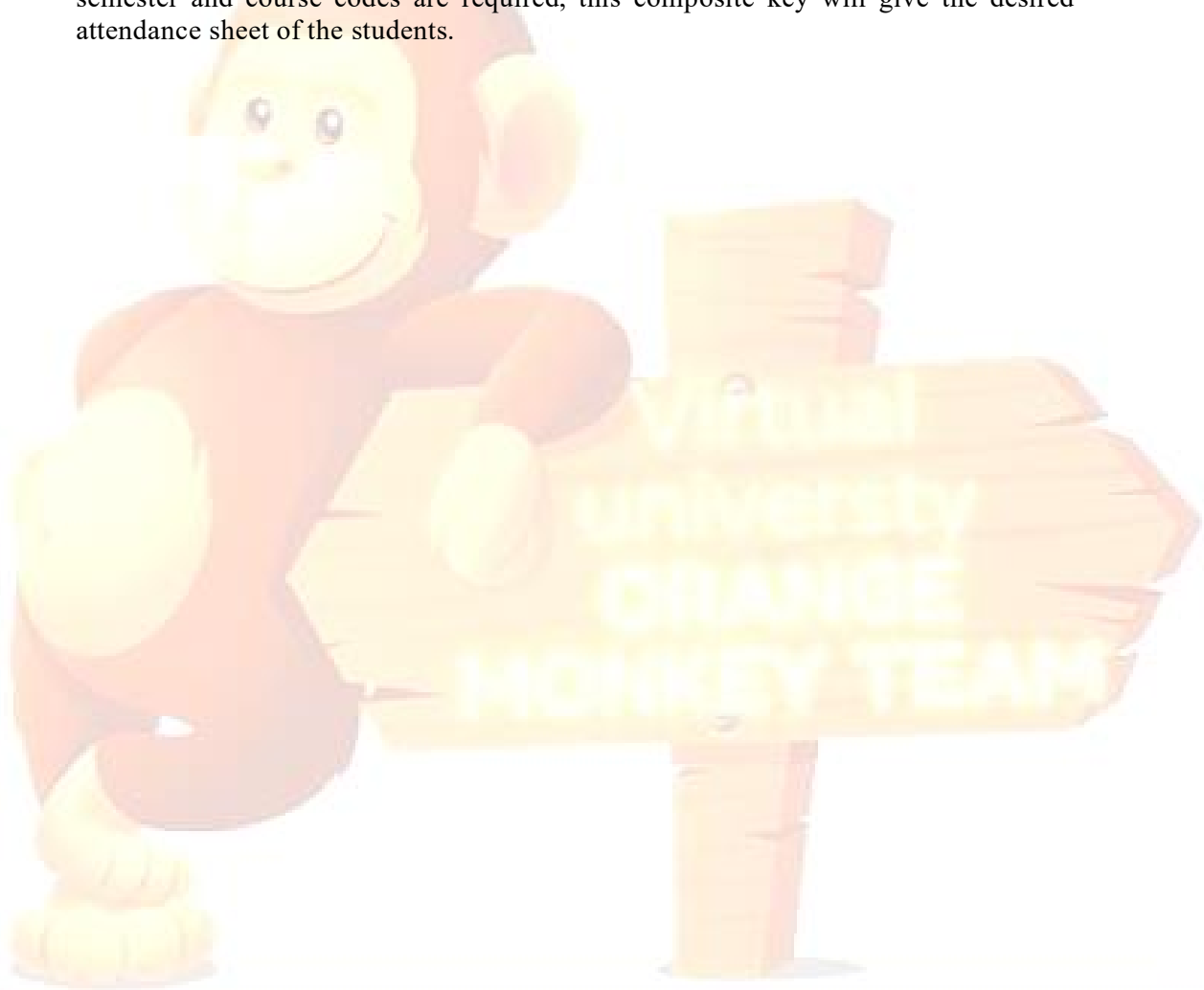
The outcome of analysis phase is the conceptual database design, which is drawn through E-R model. This design is independent of any tool or data model. This design can be implemented in multiple data models like network, relational or hierarchal models.

### Logical Database Design

This is the next phase of database design, in which appropriate data model is chosen, and from here onwards it becomes tool dependent. We will select relational data model and our further lectures will be concerning relational data models

## Conclusion

The E – R Model of Examination system of an educational institute discussed above is just a guideline. There can certainly be changes in this model depending upon the requirements of the organization and the outputs required. After drawing an E-R model, all the outputs, which are required, must be matched with the system. If it does not fulfill all the requirements then whole process must be rehased once again. All necessary modifications and changes must be made before going ahead. For Example, if in this system attendance sheet of the students is required then program code, semester and course codes are required, this composite key will give the desired attendance sheet of the students.



## Lecture No. 14

### Reading Material

“Database Systems Principles, Design and Implementation” written by Catherine Ricardo, Maxwell Macmillan.	Section 6.1 – 6.3.3
“Database Management Systems”, 2 <sup>nd</sup> edition, Raghu Ramakrishnan, Johannes Gehrke, McGraw-Hill	

### Overview of Lecture

- Logical Database Design
- Introduction to Relational Data Model
- Basic properties of a table
- Mathematical and database relations

From this lecture we are going to discuss the logical database design phase of database development process. Logical database design, like conceptual database design is our database design; it represents the structure of data that we need to store to fulfill the requirements of the users or organization for which we are developing the system. However there are certain differences between the two that are presented in the table below:

	Conceptual Database Design	Logical Database Design
1	Developed in a semantic data model (generally E-R data model)	In legacy data models (relational generally in current age)
2	Free of data model in which going to be implemented; many/any possible	Free of particular DBMS in which going to be implemented; many/any possible
3	Results from Analysis Phase	Obtained by translating the conceptual database design into another data model

4	Represented graphically	Descriptive
5	More expressive	Relatively less expressive
6	Going to be transformed and then implemented	Going to be implemented
You can think more, give a try		

Table 1: Differences between Conceptual and Logical Database Designs

As we have already discussed in previous lectures and as is given in row 2 of the above table, the conceptual database design can be transformed into any data model, like, hierarchical, network, relational or object-oriented. So the study of the logical database design requires first involves the study of the data model/(s) that we can possibly use for the purpose. However, in the current age, since early eighties, the most popular choice for the logical database design is the relational data model; so much popular that today it can be considered to be the only choice. Why? Because of its features we are going to discuss in today's lecture. That is why rather than studying different data models we will be studying only the relational data model. Once we study this, the development of logical database design is transformation of conceptual database design to relational one and the process is very simple and straightforward. So from today's lecture our discussion starts on the relational data model. Just for the sake of revision we repeat the definition of data model "a set of constructs/tools used to develop a database design; generally consists of three components which are constructs, manipulation language and integrity constraints". We have discussed it earlier that the later part of the definition (three components) fits precisely with the relational data model (RDM), that is, it has these components defined clearly.

### Relational Data Model

The RDM is popular due to its two major strengths and they are:

- Simplicity
- Strong Mathematical Foundation

The RDM is simple, why, there is just one structure and that is a relation or a table. Even this single structure is very easy to understand, so a user of even of a moderate

genius can understand it easily. Secondly, it has a strong mathematical foundation that gives many advantages, like:

- Anything included/defined in RDM has got a precise meaning since it is based on mathematics, so there is no confusion.
- If we want to test something regarding RDM we can test it mathematically, if it works mathematically it will work with RDM (apart from some exceptions).
- The mathematics not only provided the RDM the structure (relation) but also well defined manipulation languages (relational algebra and relational calculus).
- It provided RDM certain boundaries, so any modification or addition we want to make in RDM, we have to see if it complies with the relational mathematics or not. We cannot afford to cross these boundaries since we will be losing the huge advantages provided by the mathematical backup.

“An IBM scientist E.F. Codd proposed the relational data model in 1970. At that time most database systems were based on one of two older data models (the hierarchical model and the network model); the relational model revolutionized the database field and largely replaced these earlier models. Prototype relational database management systems were developed in pioneering research projects at IBM and UC-Berkeley by the mid-70s, and several vendors were offering relational database products shortly thereafter. Today, the relational model is by far the dominant data model and is the foundation for the leading DBMS products, including IBM's DB2 family, Informix, Oracle, Sybase, Microsoft's Access and SQLServer, FoxBase, and Paradox. Relational database systems are ubiquitous in the marketplace and represent a multibillion dollar industry” [1]

The RDM is mainly used for designing/defining external and conceptual schemas; however to some extent physical schema is also specified in it. Separation of conceptual and physical levels makes data and schema manipulation much easier, contrary to previous data models. So the relational data model also truly supports “Three Level Schema Architecture”.

### Introduction to the Relational Data model

The RDM is based on a single structure and that is a relation. Speaking in terms of the E-R data model, both the entity types and relationships are represented using relations

in RDM. The relation in RDM is similar to the mathematical relation however database relation is also represented in a two dimensional structure called table. A table consists of rows and columns. Rows of a table are also called tuples. A row or tuple of a table represents a record or an entity instance, where as the columns of the table represent the properties or attributes.

stID	stName	clName	doB	sex
S001	M. Suhail	MCS	12/6/84	M
S002	M. Shahid	BCS	3/9/86	M
S003	Naila S.	MCS	7/8/85	F
S004	Rubab A.	MBA	23/4/86	F
S005	Ehsan M.	BBA	22/7/88	M

Table 2: A database relation represented in the form of a table

In the above diagram, a table is shown that consists of five rows and five columns. The top most rows contain the names of the columns or attributes whereas the rows represent the records or entity instances. There are six basic properties of the database relations which are:

- Each cell of a table contains atomic/single value

A cell is the intersection of a row and a column, so it represents a value of an attribute in a particular row. The property means that the value stored in a single cell is considered as a single value. In real life we see many situations when a property/attribute of any entity contains multiple values, like, degrees that a person has, hobbies of a student, the cars owned by a person, the jobs of an employee. All these attributes have multiple values; these values cannot be placed as the value of a single attribute or in a cell of the table. It does not mean that the RDM cannot handle such situations, however, there are some special means that we have to adopt in these situations, and they can not be placed as the value of an attribute because an attribute can contain only a single value. The values of attributes shown in table 1 are all atomic or single.

- Each column has a distinct name; the name of the attribute it represents

Each column has a heading that is basically the name of the attribute that the column represents. It has to be unique, that is, a table cannot have duplicated column/attribute names. In the table 2 above, the bold items in the first row represent the column/attribute names.

- The values of the attributes come from the same domain

Each attribute is assigned a domain along with the name when it is defined. The domain represents the set of possible values that an attribute can have. Once the domain has been assigned to an attribute, then all the rows that are added into the table will have the values from the same domain for that particular column. For example, in the table 2 shown above the attribute doB (date of birth) is assigned the domain "Date", now all the rows have the date value against the attribute doB. This attribute cannot have a text or numeric value.

- The order of the columns is immaterial

If the order of the columns in a table is changed, the table still remains the same. Order of the columns does not matter.

- The order of the rows is immaterial

As with the columns, if rows' order is changed the table remains the same.

- Each row/tuple/record is distinct, no two rows can be same

Two rows of a table cannot be same. The value of even a single attribute has to be different that makes the entire row distinct.

There are three components of the RDM, which are, construct (relation), manipulation language (SQL) and integrity constraints (two). We have discussed the relation so far; the last two components will be discussed later. In the next section we are going to

discuss the mathematical relations briefly that will help to link the mathematical relations with the database relations and will help in a better understanding of the later.

## Mathematical Relations

Consider two sets

$$A = \{x, y\} \quad B = \{2, 4, 6\}$$

Cartesian product of these sets ( $A \times B$ ) is a set that consists of ordered pairs where first element of the ordered pair belongs to set A where as second element belongs to set B, as shown below:

$$A \times B = \{(x,2), (x,4), (x,6), (y,2), (y,4), (y,6)\}$$

A relation is some subset of this Cartesian product, For example,

- $R1 = \{(x,2), (y,2), (x,6), (x,4)\}$
- $R2 = \{(x,4), (y,6), (y,4)\}$

The same notion of Cartesian product and relations can be applied to more than two sets, e.g. in case of three sets, we will have a relation of ordered triplets

Applying the same concept in a real world scenario, consider two sets Name and Age having the elements:

- Name = {Ali, Sana, Ahmed, Sara}
- Age = {15,16,17,18,.....,25}

Cartesian product of Name & Age

$$\text{Name} \times \text{Age} = \{(Ali,15), (Sana,15), (Ahmed,15), (Sara,15), \dots, (Ahmed,25), (Sara,25)\}$$

Now consider a subset CLASS of this Cartesian product

$$\text{CLASS} = \{(Ali, 18), (Sana, 17), (Ali, 20), (Ahmed, 19)\}$$

This subset CLASS is a relation mathematically, however, it may represent a class in the real world where each ordered pair represents a particular student mentioning the name and age of a student. In the database context each ordered pair represents a tuple and elements in the ordered pairs represent values of the attributes. Think in this way, if Name and Age represent all possible values for names and ages of students, then any class you consider that will definitely be a subset of the Cartesian product of the Name and Age. That is, the name and age combination of all the students of any class

will be included in the Cartesian product and if we take out particulars ordered pairs that are related to a class then that will be a subset of the Cartesian product, a relation.

## Database Relations

Let  $A_1, A_2, A_3, \dots, A_n$  be some attributes and  $D_1, D_2, D_3, \dots, D_n$  be their domains. A relation scheme relates certain attributes with their domain in context of a relation. A relation scheme can be represented as:

$R = (A_1:D_1, A_2:D_2, \dots, A_n:D_n)$ , for example,

STD Scheme = (stId:Text, stName: Text, stAdres:Text, doB:Date) OR

STD(stId, stName, stAdres, doB)

Whereas the stId, stName, stAdres and doB are the attribute names and Text, Text, Text and Date are their respective domains. A database relation as per this relation scheme can be:

STD = {(stId:S001, stName:Ali, stAdres: Lahore, doB:12/12/76), (stId:S003, stName:A. Rehman, stAdres: RWP, doB:2/12/77)} OR

STD = {(S001, Ali, Lahore, 12/12/76), (S003, A. Rehman, RWP, 2/12/77)}

The above relation if represented in a two dimensional structure will be called a table as is shown below:

stId	stName	stAdres	doB
S001	Ali	Lahore	12/12/76
S002	A. Rehman	RWP	2/12/77

With this, today's lecture is finished; the discussion on RDM will be continued in the next lecture.

## Summary

In this lecture we have started the discussion on the logical database design that we develop from the conceptual database design. The later is generally developed using E-R data model, whereas for the former RDM is used. RDM is based on the theory of mathematical relations; a mathematical relation is subset of the Cartesian product of two or more sets. Relations are physically represented in the form of two-dimensional

structure called table, where rows/tuples represent records and columns represent the attributes.

**Exercise:**

Define different attributes (assigning name and domain to each) for an entity STUDENT, then apply the concept of Cartesian product on the domains of these attributes, then consider the records of your class fellows and see if it is the subset of the Cartesian product.



## Lecture No. 15

### Reading Material

--	--

### Overview of Lecture

- Database and Math Relations
- Degree and Cardinality of Relation
- Integrity Constraints
- Transforming conceptual database design into logical database design
- Composite and multi-valued Attributes
- Identifier Dependency

In the previous lecture we discussed relational data model, its components and properties of a table. We also discussed mathematical and database relations. Now we will discuss the difference in between database and mathematical relations.

### Database and Math Relations

We studied six basic properties of tables or database relations. If we compare these properties with those of mathematical relations then we find out that properties of both are the same except the one related to order of the columns. The order of columns in mathematical relations does matter, whereas in database relations it does not matter. There will not be any change in either math or database relations if we change the rows or tuples of any relation. It means that the only difference in between these two is of order of columns or attributes. A math relation is a Cartesian product of two sets. So if we change the order of these two sets then the outcome of both will not be same. Therefore, the math relation changes by changing the order of columns. For Example , if there is a set A and a set B if we take Cartesian product of A and B then we take Cartesian product of B and A they will not be equal , so

$$A \times B \neq B \times A$$

Rests of the properties between them are same.

## Degree of a Relation

We will now discuss the degree of a relation not to be confused with the degree of a relationship. You would be definitely remembering that the relationship is a link or association between one or more entity types and we discussed it in E-R data model.

However the degree of a relation is the number of columns in that relation. For

Example consider the table given below:

STUDENT

StID	stName	clName	Sex
S001	Suhail	MCS	M
S002	Shahid	BCS	M
S003	Naila	MCS	F
S004	Rubab	MBA	F
S005	Ehsan	BBA	M

Table 1: The STUDENT table

Now in this example the relation STUDENT has four columns, so this relation has degree four.

### Cardinality of a Relation

The number of rows present in a relation is called as cardinality of that relation. For example, in STUDENT table above, the number of rows is five, so the cardinality of the relation is five.

### Relation Keys

The concept of key and all different types of keys is applicable to relations as well. We will now discuss the concept of foreign key in detail, which will be used quite frequently in the RDM.

### Foreign Key

An attribute of a table B that is primary key in another table A is called as foreign key. For Example, consider the following two tables EMP and DEPT:

EMP (empId, empName, qual, depId)

DEPT (depId, depName, numEmp)

In this example there are two relations; EMP is having record of employees, whereas DEPT is having record of different departments of an organization. Now in EMP the primary key is empId, whereas in DEPT the primary key is depId. The depId which is primary key of DEPT is also present in EMP so this is a foreign key.

### Requirements/Constraints of Foreign Key

Following are some requirements / constraints of foreign key:

There can be more than zero, one or multiple foreign keys in a table, depending on how many tables a particular table is related with. For example in the above example the EMP table is related with the DEPT table, so there is one foreign key depId,

whereas DEPT table does not contain any foreign key. Similarly, the EMP table may also be linked with DESIG table storing designations, in that case EMP will have another foreign key and alike.

The foreign key attribute, which is present as a primary key in another relation is called as home relation of foreign key attribute, so in EMP table the deptId is foreign key and its home relation is DEPT.

The foreign key attribute and the one present in another relation as primary key can have different names, but both must have same domains. In DEPT, EMP example, both the PK and FK have the same name; they could have been different, it would not have made any difference however they must have the same domain.

The primary key is represented by underlining with a solid line, whereas foreign key is underlined by dashed or dotted line.

Primary Key : \_\_\_\_\_

Foreign Key : - - - - -

#### Integrity Constraints

Integrity constraints are very important and they play a vital role in relational data model. They are one of the three components of relational data model. These constraints are basic form of constraints, so basic that they are a part of the data model, due to this fact every DBMS that is based on the RDM must support them.

#### Entity Integrity Constraint:

It states that in a relation no attribute of a primary key (PK) can have null value. If a PK consists of single attribute, this constraint obviously applies on this attribute, so it cannot have the Null value. However, if a PK consists of multiple attributes, then none of the attributes of this PK can have the Null value in any of the instances.

#### Referential Integrity Constraint:

This constraint is applied to foreign keys. Foreign key is an attribute or attribute combination of a relation that is the primary key of another relation. This constraint states that if a foreign key exists in a relation, either the foreign key value must match the primary key value of some tuple in its home relation or the foreign key value must be completely null.

#### Significance of Constraints:

By definition a PK is a minimal identifier that is used to identify tuples uniquely. This means that no subset of the primary key is sufficient to provide unique identification of tuples. If we were to allow a null value for any part of the primary key, we would be demonstrating that not all of the attributes are needed to distinguish between tuples, which would contradict the definition.

Referential integrity constraint plays a vital role in maintaining the correctness, validity or integrity of the database. This means that when we have to ensure the proper enforcement of the referential integrity constraint to ensure the consistency and correctness of database. How? In the DEPT, EMP example above deptId in EMP is foreign key; this is being used as a link between the two tables. Now in every instance of EMP table the attribute deptId will have a value, this value will be used to get the name and other details of the department in which a particular employee works. If the value of deptId in EMP is Null in a row or tuple, it means this particular row is not related with any instance of the DEPT. From real-world scenario it means that this particular employee (whose is being represented by this row/tuple) has not been

assigned any department or his/her department has not been specified. These were two possible conditions that are being reflected by a legal value or Null value of the foreign key attribute. Now consider the situation when referential integrity constraint is being violated, that is, EMP.deptId contains a value that does not match with any of the value of DEPT.deptId. In this situation, if we want to know the department of an employee, then oops, there is no department with this Id, that means, an employee has been assigned a department that does not exist in the organization or an illegal department. A wrong situation, not wanted. This is the significance of the integrity constraints.

#### Null Constraints:

A Null value of an attribute means that the value of attribute is not yet given, not defined yet. It can be assigned or defined later however. Through Null constraint we can monitor whether an attribute can have Null value or not. This is important and we have to make careful use of this constraint. This constraint is included in the definition of a table (or an attribute more precisely). By default a non-key attribute can have Null value, however, if we declare an attribute as Not Null, then this attribute must be assigned value while entering a record/tuple into the table containing that attribute. The question is, how do we apply or when do we apply this constraint, or why and when, on what basis we declare an attribute Null or Not Null. The answer is, from the system for which we are developing the database; it is generally an organizational constraint. For example, in a bank, a potential customer has to fill in a form that may comprise of many entries, but some of them would be necessary to fill in, like, the residential address, or the national Id card number. There may be some entries that may be optional, like fax number. When defining a database system for such a bank, if we create a CLIENT table then we will declare the must attributes as Not Null, so that a record cannot be successfully entered into the table until at least those attributes are not specified.

#### Default Value:

This constraint means that if we do not give any value to any particular attribute, it will be given a certain (default) value. This constraint is generally used for the efficiency purpose in the data entry process. Sometimes an attribute has a certain value that is assigned to it in most of the cases. For example, while entering data for the students, one attribute holds the current semester of the student. The value of this attribute is changed as a student passes through different exams or semesters during its degree. However, when a student is registered for the first time, it is generally registered in the first semesters. So in the new records the value of current semester attribute is generally 1. Rather than expecting the person entering the data to enter 1 in every record, we can place a default value of 1 for this attribute. So the person can simply skip the attribute and the attribute will automatically assume its default value.

#### Domain Constraint:

This is an essential constraint that is applied on every attribute, that is, every attribute has got a domain. Domain means the possible set of values that an attribute can have. For example, some attributes may have numeric values, like salary, age, marks etc. Some attributes may possess text or character values, like, name and address. Yet some others may have the date type value, like date of birth, joining date. Domain specification limits an attribute the nature of values that it can have. Domain is specified by associating a data type to an attribute while defining it. Exact data type name or specification depends on the particular tool that is being used. Domain helps

to maintain the integrity of the data by allowing only legal type of values to an attribute. For example, if the age attribute has been assigned a numeric data type then it will not be possible to assign a text or date value to it. As a database designer, this is your job to assign an appropriate data type to an attribute. Another perspective that needs to be considered is that the value assigned to attributes should be stored efficiently. That is, domain should not allocate unnecessary large space for the attribute. For example, age has to be numeric, but then there are different types of numeric data types supported by different tools that permit different range of values and hence require different storage space. Some of more frequently supported numeric data types include Byte, Integer, and Long Integer. Each of these types supports different range of numeric values and takes 1, 4 or 8 bytes to store. Now, if we declare the age attribute as Long Integer, it will definitely serve the purpose, but we will be allocating unnecessarily large space for each attribute. A Byte type would have been sufficient for this purpose since you won't find students or employees of age more than 255, the upper limit supported by Byte data type. Rather we can further restrict the domain of an attribute by applying a check constraint on the attribute. For example, the age attribute although assigned type Byte, still if a person by mistake enters the age of a student as 200, if this is year then it is not a legal age from today's age, yet it is legal from the domain constraint perspective. So we can limit the range supported by a domain by applying the check constraint by limiting it up to say 30 or 40, whatever is the rule of the organization. At the same time, don't be too sensitive about storage efficiency, since attribute domains should be large enough to cater the future enhancement in the possible set of values. So domain should be a bit larger than that is required today. In short, domain is also a very useful constraint and we should use it carefully as per the situation and requirements in the organization.

### RDM Components

We have up till now studied following two components of the RDM, which are the Structure and Entity Integrity Constraints. The third part, that is, the Manipulation Language will be discussed in length in the coming lectures.

### Designing Logical Database

Logical data base design is obtained from conceptual database design. We have seen that initially we studied the whole system through different means. Then we identified different entities, their attributes and relationship in between them. Then with the help of E-R data model we achieved an E-R diagram through different tools available in this model. This model is semantically rich. This is our conceptual database design. Then as we had to use relational data model so then we came to implementation phase for designing logical database through relational data model.

The process of converting conceptual database into logical database involves transformation of E-R data model into relational data model. We have studied both the data models, now we will see how to perform this transformation.

### Transforming Rules

Following are the transforming rules for converting conceptual database into logical database design:

The rules are straightforward, which means that we just have to follow the rules mentioned and the required logical database design would be achieved

There are two ways of transforming first one is manually that is we analyze and evaluate and then transform. Second is that we have CASE tools available with us which can automatically convert conceptual database into required logical database design

If we are using CASE tools for transforming then we must evaluate it as there are multiple options available and we must make necessary changes if required.

### Mapping Entity Types

Following are the rules for mapping entity types:

Each regular entity type (ET) is transformed straightaway into a relation. It means that whatever entities we had identified they would simply be converted into a relation and will have the same name of relation as kept earlier.

Primary key of the entity is declared as Primary key of relation and underlined.

Simple attributes of ET are included into the relation

For Example, figure 1 below shows the conversion of a strong entity type into equivalent relation:

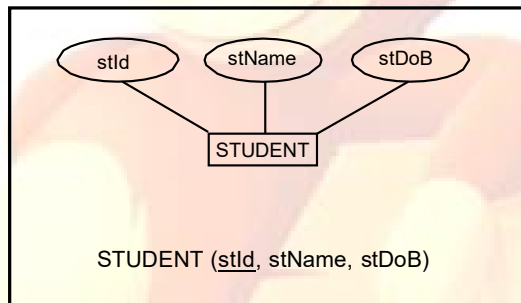


Fig. 1: An example strong entity type

### Composite Attributes

These are those attributes which are a combination of two or more than two attributes. For address can be a composite attribute as it can have house no, street no, city code and country, similarly name can be a combination of first and last names. Now in relational data model composite attributes are treated differently. Since tables can contain only atomic values composite attributes need to be represented as a separate relation

For Example in student entity type there is a composite attribute Address, now in E-R model it can be represented with simple attributes but here in relational data model, there is a requirement of another relation like following:

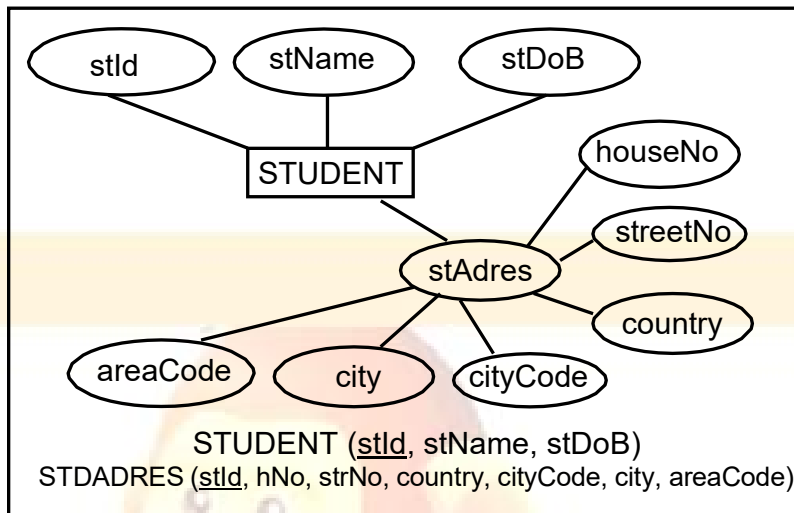


Fig. 2: Transformation of composite attribute

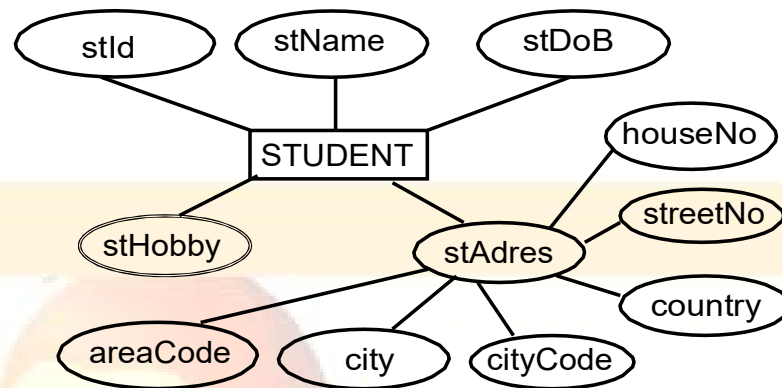
Figure 2 above presents an example of transforming a composite attribute into RDM, where it is transformed into a table that is linked with the STUDENT table with the primary key

#### Multi-valued Attributes

These are those attributes which can have more than one value against an attribute. For Example a student can have more than one hobby like riding, reading listening to music etc. So these attributes are treated differently in relational data model. Following are the rules for multi-valued attributes:-

An Entity type with a multi-valued attribute is transformed into two relations. One contains the entity type and other simple attributes whereas the second one has the multi-valued attribute. In this way only single atomic value is stored against every attribute. The Primary key of the second relation is the primary key of first relation and the attribute value itself. So in the second relation the primary key is the combination of two attributes.

All values are accessed through reference of the primary key that also serves as



STUDENT (stId, stName, stDoB)  
 STDADRES (stId, hNo, strNo, country, cityCode, city, areaCode)  
 STHOBBY(stId, stHobby)

Fig. 3: Transformation of multi-valued attribute

foreign key.

Conclusion

In this lecture we have studied the difference between mathematical and database relations. The concepts of foreign key and especially the integrity constraints are very important and are basic for every database. Then how a conceptual database is transformed into logical database and in our case it is relational data model as it is the most widely used. We have also studied certain transforming rules for converting E-R data model into relational data model. Some other rule for this transformation will be studied in the coming lectures

You will receive exercise at the end of this topic.

## Lecture No. 16

### Reading Material

“Database Systems Principles, Design and Implementation”  
written by Catherine Ricardo, Maxwell Macmillan.

Page 209

### Overview of Lecture:

- Mapping Relationships
- Binary Relationships
- Unary Relationships
- Data Manipulation Languages

In the previous lecture we discussed the integrity constraints. How conceptual database is converted into logical database design, composite and multi-valued attributes. In this lecture we will discuss different mapping relationships.

### Mapping Relationships

We have up till now converted an entity type and its attributes into RDM. Before establishing any relationship in between different relations, it is must to study the cardinality and degree of the relationship. There is a difference in between relation and relationship. Relation is a structure, which is obtained by converting an entity type in E-R model into a relation, whereas a relationship is in between two relations of relational data model. Relationships in relational data model are mapped according to their degree and cardinalities. It means before establishing a relationship there cardinality and degree is important.

### Binary Relationships

Binary relationships are those, which are established between two entity type. Following are the three types of cardinalities for binary relationships:

- One to One
- One to Many
- Many to Many

In the following treatment in each of these situations is discussed.

#### One to Many:

In this type of cardinality one instance of a relation or entity type is mapped with many instances of second entity type, and inversely one instance of second entity type is mapped with one instance of first entity type. The participating entity types will be transformed into relations as has been already discussed. The relationship in this particular case will be implemented by placing the PK of the entity type (or corresponding relation) against one side of relationship will be included in the entity type (or corresponding relation) on the many side of the relationship as foreign key (FK). By declaring the PK-FK link between the two relations the referential integrity constraint is implemented automatically, which means that value of foreign key is either null or matches with its value in the home relation.

For Example, consider the binary relationship given in the figure 1 involving two entity types PROJÉT and EMPLOYÉE. Now there is a one to many relationships between these two. On any one project many employees can work and one employee can work on only one project.

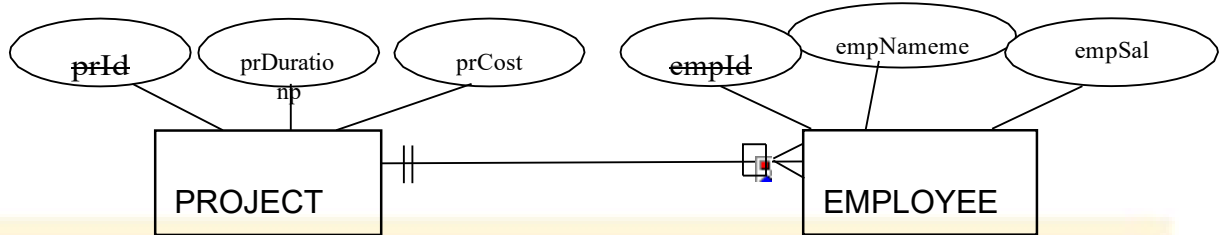


Fig. 1: A one to many relationship

The two participating entity types are transformed into relations and the relationship is implemented by including the PK of PROJECT (prId) into the EMPLOYEE as FK.

So the transformation will be:

PROJECT (prId, prDura, prCost)

EMPLOYEE (empId, empName, empSal, prId)

The PK of the PROJECT has been included in EMPLOYEE as FK; both keys do not need to have same name, but they must have the same domain.

#### Minimum Cardinality:

This is a very important point, as minimum cardinality on one side needs special attention. Like in previous example an employee cannot exist if project is not assigned.

So in that case the minimum cardinality has to be one. On the other hand if an instance of EMPLOYEE can exist without being linked with an instance of the PROJECT then the minimum cardinality has to be zero. If the minimum cardinality is zero, then the FK is defined as normal and it can have the Null value, on the other hand if it is one then we have to declare the FK attribute(s) as Not Null. The Not Null constraint makes it a must to enter the value in the attribute(s) whereas the FK constraint will enforce the value to be a legal one. So you have to see the minimum cardinality while implementing a one to many relationship.

#### Many to Many Relationship:

In this type of relationship one instance of first entity can be mapped with many instances of second entity. Similarly one instance of second entity can be mapped with many instances of first entity type. In many to many relationship a third table is created for the relationship, which is also called as associative entity type. Generally,

the primary keys of the participating entity types are used as primary key of the third table.

For Example, there are two entity types BOOK and STD (student). Now many students can borrow a book and similarly many books can be issued to a student, so in this manner there is a many to many relationship. Now there would be a third relation as well which will have its primary key after combining primary keys of BOOK and STD. We have named that as transaction TRANS. Following are the attributes of these relations: -

- STD (stId, sName, sFname)
- BOOK (bkId, bkTitle, bkAuth)
- TRANS (stId,bkId, isDate,rtDate)

Now here the third relation TRANS has four attributes first two are the primary keys of two entities whereas the last two are issue date and return date.

#### One to One Relationship:

This is a special form of one to many relationship, in which one instance of first entity type is mapped with one instance of second entity type and also the other way round. In this relationship primary key of one entity type has to be included on other as foreign key. Normally primary key of compulsory side is included in the optional side. For example, there are two entities STD and STAPPLE (student application for scholarship). Now the relationship from STD to STAPPLE is optional whereas STAPPLE to STD is compulsory. That means every instance of STAPPLE must be related with one instance of STD, whereas it is not a must for an instance of STD to be related to an instance of STAPPLE, however, if it is related then it will be related to one instance of STAPPLE, that is, one student can give just one scholarship application. This relationship is shown in the figure below:

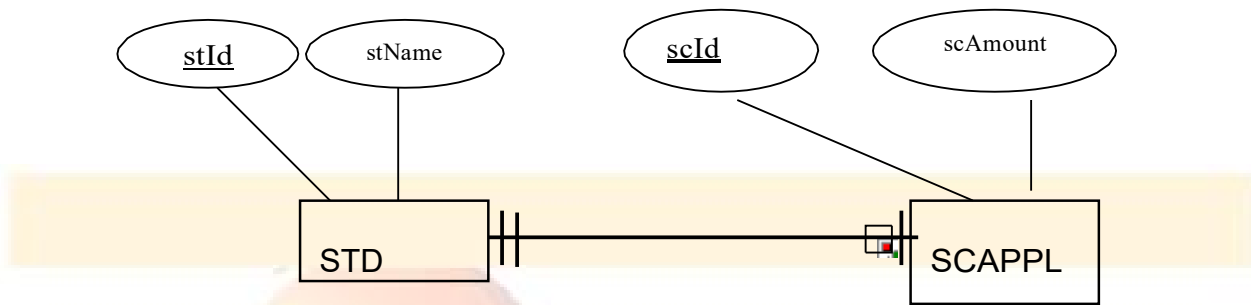


Fig. 2: A one to one relationship

While transforming, two relations will be created, one for STD and HOBBY each. For relationship PK of either one can be included in the other, it will work. But preferably, we should include the PK of STD in HOBBY as FK with Not Null constraint imposed on it.

STD (stId, stName)

STAPPLE (scId, scAmount, stId)

The advantage of including the PK of STD in STAPPLE as FK is that any instance of STAPPLE will definitely have a value in the FK attribute, that is, stId. Whereas if we do other way round; we include the PK of STAPPLE in STD as FK, then since the relationship is optional from STD side, the instances of STD may have Null value in the FK attribute (scId), causing the wastage of storage. More the number records with Null value more wastage.

### Unary Relationship

These are the relationships, which involve a single entity. These are also called recursive relationships. Unary relationships may have one to one, one to many and many to many cardinalities. In unary one to one and one to may relationships, the PK of same entity type is used as foreign key in the same relation and obviously with the different name since same attribute name cannot be used in the same table. The example of one to one relationship is shown in the figure below:

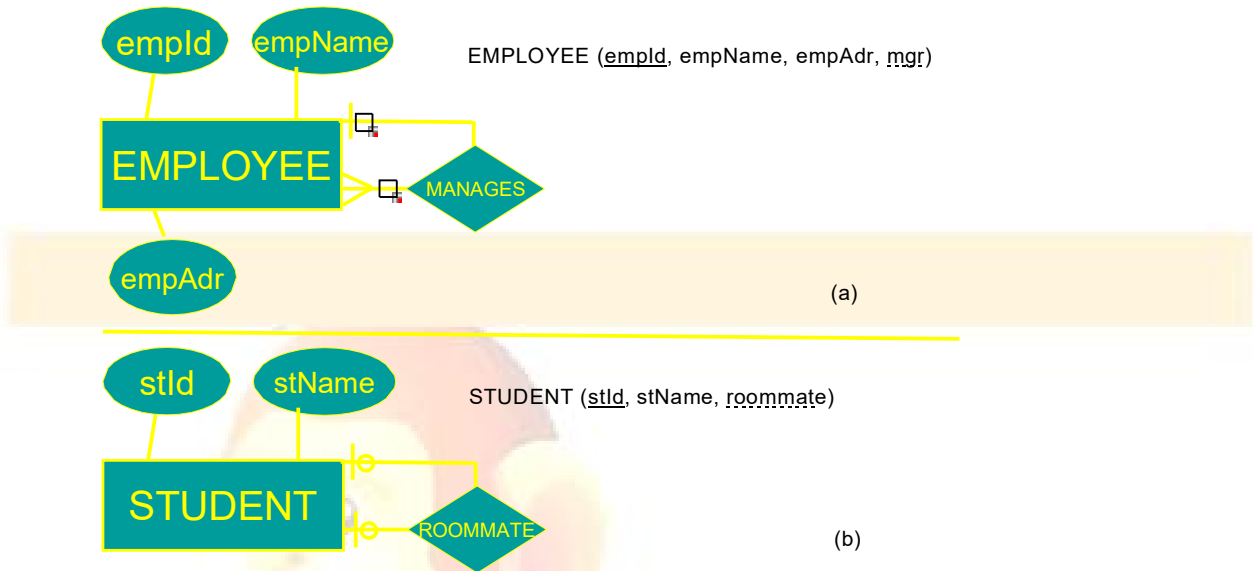


Fig. 3: One to one relationships (a) one to many (b) one to one and their transformation

In many to many relationships another relation is created with composite key. For example there is an entity type PART may have many to many recursive relationships, meaning one part consists of many parts and one part may be used in many parts. So in this case this is a many to many relationship. The treatment of such a relationship is shown in the figure below:

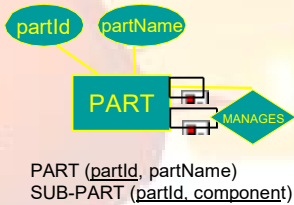


Fig. 4: Recursive many to many relationship and transformation

**Super / Subtype Relationship:**

Separate relations are created for each super type and subtypes. It means if there is one super type and there are three subtypes, so then four relations are to be created. After creating these relations then attributes are assigned. Common attributes are assigned to super type and specialized attributes are assigned to concerned subtypes. Primary key of super type is included in all relations that work for both link and

identity. Now to link the super type with concerned subtype there is a requirement of descriptive attribute, which is called as discriminator. It is used to identify which subtype is to be linked. For Example there is an entity type EMP which is a super type, now there are three subtypes, which are salaried, hourly and consultants. So now there is a requirement of a determinant, which can identify that which subtypes to be consulted, so with empId a special character can be added which can be used to identify the concerned subtype.

#### Summary of Mapping E-R Diagram to Relational DM:

We have up till now studied that how conceptual database design is converted into logical database. E-R data model is semantically rich and it has number of constructs for representing the whole system. Conceptual database is free of any data model, whereas logical database the required data model is chosen; in our case it is relational data model. First we identified the entity types, weak and strong entity types. Then we converted those entities into relations. After converting entities into relations then attributes are identified, different types of attributes are identified. Then relationships were made, in which cardinality and degree was identified. In ternary relationship, where three entities are involved, in this as well another relation is created to establish relationship among them. Then finally we had studied the super and sub types in which primary key of super type was used for both identity and link.

#### Data Manipulation Languages

This is the third component of relational data model. We have studied structure, which is the relation, integrity constraints both referential and entity integrity constraint. Data manipulation languages are used to carry out different operations like insertion, deletion and updation of data in database. Following are the two types of languages:

### Procedural Languages:

These are those languages in which what to do and how to do on the database is required. It means whatever operation is to be done on the database that has to be told that how to perform.

### Non -Procedural Languages:

These are those languages in which only what to do is required, rest how to do is done by the manipulation language itself.

Structured query language (SQL) is the most widely language used for manipulation of data. But we will first study Relational Algebra and Relational Calculus, which are procedural and non – procedural respectively.

## Relational Algebra

Following are few major properties of relational algebra:

- Relational algebra operations work on one or more relations to define another relation leaving the original intact. It means that the input for relational algebra can be one or more relations and the output would be another relation, but the original participating relations will remain unchanged and intact. Both operands and results are relations, so output from one operation can become input to another operation. It means that the input and output both are relations so they can be used iteratively in different requirements.
- Allows expressions to be nested, just as in arithmetic. This property is called closure.
- There are five basic operations in relational algebra: Selection, Projection, Cartesian product, Union, and Set Difference.
- These perform most of the data retrieval operations needed.
- It also has Join, Intersection, and Division operations, which can be expressed in terms of 5 basic operations.

### Exercise:

- Consider the example given in Ricardo book on page 216 and transform it into relational data model. Make any necessary assumptions if required.

## Lecture No. 17

### Reading Material

“Database Systems Principles, Design and Implementation”  
written by Catherine Ricardo, Maxwell Macmillan.

Chapter 6

### Overview of Lecture:

- Five Basic Operators of Relational Algebra
- Join Operation

In the previous lecture we discussed about the transformation of conceptual database design into relational database. In E-R data model we had number of constructs but in relational data model it was only a relation or a table. We started discussion on data manipulation languages (DML) of relational data model (SDM). We will now study in detail the different operators being used in relational algebra.

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation as their result. There are five basic operations of relational algebra. They are broadly divided into two categories:

#### Unary Operations:

These are those operations, which involve only one relation or table. These are Select and Project

#### Binary Operations:

These are those operations, which involve pairs of relations and are, therefore called as binary operations. The input for these operations is two relations and they produce a new relation without changing the original relations. These operations are:

- Union

- Set Difference
- Cartesian Product

#### The Select Operation:

The select operation is performed to select certain rows or tuples of a table, so it performs its action on the table horizontally. The tuples are selected through this operation using a predicate or condition. This command works on a single table and takes rows that meet a specified condition, copying them into a new table. Lower Greek letter sigma ( $\sigma$ ) is used to denote the selection. The predicate appears as subscript to  $\sigma$ . The argument relation is given in parenthesis following the  $\sigma$ . As a result of this operation a new table is formed, without changing the original table. As a result of this operation all the attributes of the resulting table are same, which means that degree of the new and old tables are same. Only selected rows / tuples are picked up by the given condition. While processing a selection all the tuples of a table are looked up and those tuples, which match a particular condition, are picked up for the new table. The degree of the resulting relation will be the same as of the relation itself.

$$|\sigma| = |r(R)|$$

The select operation is commutative, which is as under: -

$$\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R))$$

If a condition 2 (c2) is applied on a relation R and then c1 is applied, the resulting table would be equivalent even if this condition is reversed that is first c1 is applied and then c2 is applied.

For example there is a table STUDENT with five attributes.

#### STUDENT

stId	stName	stAdr	prName	curSem
S1020	Sohail Dar	H#14, F/8-4, Islamabad.	MCS	4
S1038	Shoaib Ali	H#23, G/9-1, Islamabad	BCS	3
S1015	Tahira Ejaz	H#99, Lala Rukh Wah.	MCS	5
S1018	Arif Zia	H#10, E-8, Islamabad.	BIT	5

Fig. 1: An example STUDENT table

The following is an example of select operation on the table STUDENT:

$\sigma_{\text{Curr\_Sem} > 3}(\text{STUDENT})$

The components of the select operations are clear from the above example;  $\sigma$  is the symbol being used (operator), “curr\_sem > 3” written in the subscript is the predicate and STUDENT given in parentheses is the table name. The resulting relation of this command would contain record of those students whose semester is greater than three as under:

$\sigma_{\text{Curr\_Sem} > 3}(\text{STUDENT})$

stId	stName	stAdr	prName	curSem
S1020	Sohail Dar	H#14, F/8-4, Islamabad.	MCS	4
S1015	Tahira Ejaz	H#99, Lala Rukh Wah.	MCS	5
S1018	Arif Zia	H#10, E-8, Islamabad.	BIT	5

Fig. 2: Output relation of a select operation

In selection operation the comparison operators like <, >, =, <=, >=, <> can be used in the predicate. Similarly, we can also combine several simple predicates into a larger predicate using the connectives *and* ( $\wedge$ ) and *or* ( $\vee$ ). Some other examples of select operation on the STUDENT table are given below:

$\sigma_{\text{stId} = \text{'S1015'}}(\text{STUDENT})$

$\sigma_{\text{prName} <> \text{'MCS'}}(\text{STUDENT})$

## The Project Operator

The Select operation works horizontally on the table on the other hand the Project operator operates on a single table vertically, that is, it produces a vertical subset of the table, extracting the values of specified columns, eliminating duplicates, and placing the values in a new table. It is unary operation that returns its argument relation, with certain attributes left out. Since relation is a set any duplicate rows are eliminated. Projection is denoted by a Greek letter ( $\Pi$ ). While using this operator all the rows of selected attributes of a relation are part of new relation. For example consider a relation FACULTY with five attributes and certain number of rows.

## FACULTY

FacId	facName	Dept	Salary	Rank
F2345	Usman	CSE	21000	lecturer
F3456	Tahir	CSE	23000	Asst Prof
F4567	Ayesha	ENG	27000	Asst Prof
F5678	Samad	MATH	32000	Professor

Fig. 3: An example FACULTY table

If we apply the projection operator on the table for the following commands all the rows of selected attributes will be shown, for example:

$\Pi_{\text{FacId, Salary}}(\text{FACULTY})$

FacId	Salary
F2345	21000
F3456	23000
F4567	27000
F5678	32000

Fig. 4: Output relation of a project operation on table of figure 3

Some other examples of project operation on the same table can be:

$\Pi_{\text{Fname, Rank}}(\text{Faculty})$

$\Pi_{\text{Facid, Salary, Rank}}(\text{Faculty})$

### Composition of Relational Operators:

The relational operators like select and project can also be used in nested forms iteratively. As the result of an operation is a relation so this result can be used as an input for other operation. For Example if we want the names of faculty members along with departments, who are assistant professors then we have to perform both the select and project operations on the FACULTY table of figure 3. First selection operator is applied for selecting the associate professors, the operation outputs a relation that is given as input to the projection operation for the required attributes.

$\Pi_{\text{facName, dept}} (\sigma_{\text{rank='Asst Prof'}} (\text{FACULTY}))$

The output of this command will be

facName	Dept
Tahir	CSE
Ayesha	ENG

Fig. 5: Output relation of nested operations' command

We have to be careful about the nested command sequence. For example in the above nested operations example, if we change the sequence of operations and bring the projection first then the relation provided to select operation as input will not have the attribute of rank and so then selection operator can't be applied, so there would be an error. So although the sequence can be changed, but the required attributes should be there either for selection or projection.

### The Union Operation:

We will now study the binary operations, which are also called as set operations. The first requirement for union operator is that the both the relations should be union compatible. It means that relations must meet the following two conditions:

- Both the relations should be of same degree, which means that the number of attributes in both relations should be exactly same
- The domains of corresponding attributes in both the relations should be same.

Corresponding attributes means first attributes of both relations, then second and so on.

It is denoted by U. If R and S are two relations, which are union compatible, if we take union of these two relations then the resulting relation would be the set of tuples either in R or S or both. Since it is set so there are no duplicate tuples. The union operator is commutative which means: -

$R \cup S = S \cup R$

For Example there are two relations COURSE1 and COURSE2 denoting the two tables storing the courses being offered at different campuses of an institute? Now if we want to know exactly what courses are being offered at both the campuses then we will take the union of two tables:

COURSE1

crId	progId	credHrs	courseTitle
C2345	P1245	3	Operating Sytems
C3456	P1245	4	Database Systems
C4567	P9873	4	Financial Management
C5678	P9873	3	Money & Capital Market

COURSE2

crId	progId	credHrs	courseTitle
C4567	P9873	4	Financial Management
C8944	P4567	4	Electronics

COURSE1 U COURSE2

crId	progId	credHrs	courseTitle
C2345	P1245	3	Operating Sytems
C3456	P1245	4	Database Systems
C4567	P9873	4	Financial Management
C5678	P9873	3	Money & Capital Market
C8944	P4567	4	Electronics

Fig. 5: Two tables and output of union operation on those tables

So in the union of above two courses there are no repeated tuples and they are union compatible as well.

**The Intersection Operation:**

The intersection operation also has the requirement that both the relations should be union compatible, which means they are of same degree and same domains. It is represented by  $\cap$ . If R and S are two relations and we take intersection of these two relations then the resulting relation would be the set of tuples, which are in both R and S. Just like union intersection is also commutative.

$$R \cap S = S \cap R$$

For Example, if we take intersection of COURSE1 and COURSE2 of figure 5 then the resulting relation would be set of tuples, which are common in both.

COURSE1  $\cap$  COURSE2

crId	progId	credHrs	courseTitle
C4567	P9873	4	Financial Management

Fig. 6: Output of intersection operation on COURSE1 and COURSE 2 tables of figure 5

The union and intersection operators are used less as compared to selection and projection operators.

**The Set Difference Operator:**

If R and S are two relations which are union compatible then difference of these two relations will be set of tuples that appear in R but do not appear in S. It is denoted by  $(-)$  for example if we apply difference operator on Course1 and Course2 then the resulting relation would be as under:

COURSE1  $-$  COURSE2

CID	ProgID	Cred_Hrs	CourseTitle
C2345	P1245	3	Operating Sytems
C3456	P1245	4	Database Systems
C5678	P9873	3	Money & Capital Market

Fig. 7: Output of difference operation on COURSE1 and COURSE 2 tables of figure 5

Cartesian product:

The Cartesian product needs not to be union compatible. It means they can be of different degree. It is denoted by X. suppose there is a relation R with attributes (A<sub>1</sub>, A<sub>2</sub>,...A<sub>n</sub>) and S with attributes (B<sub>1</sub>, B<sub>2</sub>,.....B<sub>n</sub>). The Cartesian product will be:

R X S

The resulting relation will be containing all the attributes of R and all of S. Moreover, all the rows of R will be merged with all the rows of S. So if there are m attributes and C rows in R and n attributes and D rows in S then the relations R x S will contain m + n columns and C \* D rows. It is also called as cross product. The Cartesian product is also commutative and associative. For Example there are two relations COURSE and STUEDNT

COURSE		STUDENT	
crId	courseTitle	stId	stName
C3456	Database Systems	S101	Ali Tahir
C4567	Financial Management	S103	Farah Hasan
C5678	Money & Capital Market		

COURSE X STUDENT

crId	courseTitle	stId	stName
C3456	Database Systems	S101	Ali Tahir
C4567	Financial Management	S101	AliTahr
C5678	Money & Capital Market	S101	Ali Tahir
C3456	Database Systems	S103	Farah Hasan
C4567	Financial Management	S103	Farah Hasan
C5678	Money & Capital Market	S103	Farah Hasan

Fig. 7: Input tables and output of Cartesian product

**Join Operation:**

Join is a special form of cross product of two tables. In Cartesian product we join a tuple of one table with the tuples of the second table. But in join there is a special requirement of relationship between tuples. For example if there is a relation STUDENT and a relation BOOK then it may be required to know that how many books have been issued to any particular student. Now in this case the primary key of STUDENT that is stId is a foreign key in BOOK table through which the join can be made. We will discuss in detail the different types of joins in our next lecture.

In this lecture we discussed different types of relational algebra operations. We will continue our discussion in the next lecture.



## Lecture No. 18

### Reading Material

“Database Systems Principles, Design and Implementation”  
written by Catherine Ricardo, Maxwell Macmillan.

Section 6.6.1 – 6.6.3

### Overview of Lecture:

- Types of Joins
- Relational Calculus
- Normalization

In the previous lecture we have studied the basic operators of relational algebra along with different examples. From this lecture we will study the different types of joins, which are very important and are used extensively in relational calculus.

### Types of Joins

Join is a special form of cross product of two tables. It is a binary operation that allows combining certain selections and a Cartesian product into one operation. The join operation forms a Cartesian product of its two arguments, performs a selection forcing equality on those attributes that appear in both relation schemas, and finally removes duplicate attributes. Following are the different types of joins: -

1. Theta Join
2. Equi Join
3. Semi Join
4. Natural Join
5. Outer Joins

We will now discuss them one by one

### Theta Join:

In theta join we apply the condition on input relation(s) and then only those selected rows are used in the cross product to be merged and included in the output. It means that in normal cross product all the rows of one relation are mapped/merged with all the rows of second relation, but here only selected rows of a relation are made cross product with second relation. It is denoted as under: -

$$R \bowtie S$$

If R and S are two relations then  $\theta$  is the condition, which is applied for select operation on one relation and then only selected rows are cross product with all the rows of second relation. For Example there are two relations of FACULTY and COURSE, now we will first apply select operation on the FACULTY relation for selection certain specific rows then these rows will have across product with COURSE relation, so this is the difference in between cross product and theta join. We will now see first both the relation their different attributes and then finally the cross product after carrying out select operation on relation.

From this example the difference in between cross product and theta join becomes clear.

#### FACULTY

facId	facName	dept	salary	rank
F234	Usman	CSE	21000	lecturer
F235	Tahir	CSE	23000	Asso Prof
F236	Ayesha	ENG	27000	Asso Prof
F237	Samad	ENG	32000	Professor

#### COURSE

crCode	crTitle	fId
C3456	Database Systems	F234
C3457	Financial Management	
C3458	Money & Capital Market	F236
C3459	Introduction to Accounting	F237

$(\sigma_{\text{rank} = \text{'Asso Prof'}}(\text{FACULTY})) \times \text{COURSE}$

facId	facName	dept	salary	rank	crCode	crTitle	fId
F235	Tahir	CSE	23000	Asso Prof	C3456	Database Systems	F234
F235	Tahir	CSE	23000	Asso Prof	C3457	Financial Management	
F235	Tahir	CSE	23000	Asso Prof	C3458	Money & Capital Market	F236
F235	Tahir	CSE	23000	Asso Prof	C3459	Introduction to Accounting	F237
F236	Ayesha	ENG	27000	Asso Prof	C3456	Database Systems	F234
F236	Ayesha	ENG	27000	Asso Prof	C3457	Financial Management	
F236	Ayesha	ENG	27000	Asso Prof	C3458	Money & Capital Market	F236

F236	Ayesha	ENG	27000	Asso Prof	C3459	Introduction to Accounting	F237
------	--------	-----	-------	--------------	-------	----------------------------------	------

Fig. 1: Two tables with an example of theta join

In this example after fulfilling the select condition of Associate professor on faculty relation then it is cross product with course relation

### Equi-Join:

This is the most used type of join. In equi-join rows are joined on the basis of values of a common attribute between the two relations. It means relations are joined on the basis of common attributes between them; which are meaningful. This means on the basis of primary key, which is a foreign key in another relation. Rows having the same value in the common attributes are joined. Common attributes appear twice in the output. It means that the attributes, which are common in both relations, appear twice, but only those rows, which are selected. Common attribute with the same name is qualified with the relation name in the output. It means that if primary and foreign keys of two relations are having the same names and if we take the equi-join of both then in the output relation the relation name will precede the attribute name. For Example, if we take the equi-join of faculty and course relations then the output would be as under: -

FACULTY				COURSE			
facId	facName	dept	salary	rank	crCode	crTitle	fId
F234	Usman	CSE	21000	lecturer	C3456	Database Systems	F234
F236	Ayesha	ENG	27000	Asso Prof	C3458	Money & Capital Market	F236
F237	Samad	ENG	32000	Professor	C3459	Introduction to Accounting	F237

Fig. 2: Equi-join on tables of figure 1

In the above example the name of common attribute between the two tables is different, that is, it is facId in FACULTY and fId in COURSE, so it is not required to qualify; however there is no harm doing it still. Now in this example after taking equi-join only those tuples are selected in the output whose values are common in both the relations.

### Natural Join:

This is the most common and general form of join. If we simply say join, it means the natural join. It is same as equi-join but the difference is that in natural join, the common attribute appears only once. Now, it does not matter which common attribute should be part of the output relation as the values in both are same. For Example if we take the natural join of faculty and course the output would be as under: -

FACULTY      facId, fld COURSE       $\bowtie$

facId	facName	dept	salary	rank	crCode	crTitle
F234	Usman	CSE	21000	Lecturer	C3456	Database Systems
F236	Ayesha	ENG	27000	Asso Prof	C3458	Money & Capital Market
F237	Samad	ENG	32000	Professor	C3459	Introduction to Accounting

Fig. 4: Natural join of FACULTY and COURSE tables of figure 1

In this example the common attribute appears only once, rest the behavior is same.

Following are the different types of natural join:-

#### Left Outer Join:

In left outer join all the tuples of left relation remain part of the output. The tuples that have a matching tuple in the second relation do have the corresponding tuple from the second relation. However, for the tuples of the left relation, which do not have a matching record in the right tuple have Null values against the attributes of the right relation. The example is given in figure 5 below. It can be described in another way. Left outer join is the equi-join plus the non matching rows of the left side relation having Null against the attributes of right side relation.

#### Right Outer Join:

In right outer join all the tuples of right relation remain part of the output relation, whereas on the left side the tuples, which do not match with the right relation, are left as null. It means that right outer join will always have all the tuples of right relation and those tuples of left relation which are not matched are left as Null.

COURSE			STUDENT	
bkId	bkTitile	stId	stId	stName
B10001	Intro to Database Systems	S104	S101	Ali Tahir
B10002	Programing Fundamentals	S101	S103	Farah Hasan
B10003	Intro Data Structures	S101	S104	Farah Naz
B10004	Modern Operating Systems	S103	S106	Asmat Dar
B10005	Computer Architecture		S107	Liaqat Ali
B10006	Advanced Networks	S104		

COURSE left outer join STUDENT				
bkId	bkTitile	BOOK.stId	STUDENT.stId	stName
B10001	Intro to Database Systems	S104	S104	Farah Naz
B10002	Programing Fundamentals	S101	S101	Ali Tahir
B10003	Intro Data Structures	S101	S101	Ali Tahir
B10004	Modern Operating Systems	S103	S103	Farah Hasan
B10006	Advanced Networks	S104	S104	Farah Naz
B10005	Computer Architecture	Null	Null	Null

COURSE right outer join STUDENT				
bkId	bkTitile	BOOK.stId	STUDENT.stId	stName
B10001	Intro to Database Systems	S104	S104	Farah Naz
B10002	Programing Fundamentals	S101	S101	Ali Tahir
B10003	Intro Data Structures	S101	S101	Ali Tahir
B10004	Modern Operating Systems	S103	S103	Farah Hasan
B10006	Advanced Networks	S104	S104	Farah Naz
Null	Null	Null	S106	Asmat Dar
Null	Null	Null	S107	Liaqat Ali

Fig. 5: Input tables and left outer join and right outer join

#### Outer Join:

In outer join all the tuples of left and right relations are part of the output. It means that all those tuples of left relation which are not matched with right relation are left as Null. Similarly all those tuples of right relation which are not matched with left relation are left as Null.

COURSE outer join STUDENT				
bkId	bkTitile	BOOK.stId	STUDENT.stId	stName
B10001	Intro to Database Systems	S104	S104	Farah Naz
B10002	Programing Fundamentals	S101	S101	Ali Tahir
B10003	Intro Data Structures	S101	S101	Ali Tahir
B10004	Modern Operating Systems	S103	S103	Farah Hasan
B10006	Advanced Networks	S104	S104	Farah Naz
B10005	Computer Architecture	Null	Null	Null
Null	Null	Null	S106	Asmat Dar
Null	Null	Null	S107	Liaqat Ali

Fig. 6: outer join operation on tables of figure 5

#### Semi Join:

In semi join, first we take the natural join of two relations then we project the attributes of first table only. So after join and matching the common attribute of both relations only attributes of first relation are projected. For Example if we take the semi join of two relations faculty and course then the resulting relation would be as under:-

FACULTY



COURSE

facId	facName	Dept	Salary	Rank
F234	Usman	CSE	21000	lecturer
F236	Ayesha	ENG	27000	Asso Prof
F237	Samad	ENG	32000	Professor

Fig. 7: Semi-join operation on tables of figure 1

Now the resulting relation has attributes of first relation only after taking the natural join of both relations.

## Relational Calculus

Relational Calculus is a nonprocedural formal relational data manipulation language in which the user simply specifies what data should be retrieved, but not how to retrieve it. It is an alternative standard for relational data manipulation languages. The relational calculus is not related to the familiar differential and integral calculus in mathematics, but takes its name from a branch of symbolic logic called the predicate calculus. It has two following forms: -

- Tuple Oriented Relational Calculus
- Domain Oriented Relational Calculus

**Tuple Oriented Relational Calculus:**

In tuple oriented relational calculus we are interested primarily in finding relation tuples for which a predicate is true. To do so we need tuple variables. A tuple variable is a variable that takes on only the tuples of some relation or relations as its range of values. It actually corresponds to a mathematical domain. We specify the range of a tuple variable by a statement such as: -

RANGE OF S IS STUDENT

Here, S is the tuple variable and STUDENT is the range, so that S always represents a tuple of STUDENT. It is expressed as

$$\{S \mid P(S)\}$$

We will read it as find the set of all tuples S such that P(S) is true, where P implies the predicate condition now suppose range of R is STUDENT

$$\{R \mid R.Credits > 50\}$$

We will say like find the stuId, stuName, majors etc of all students having more than 50 credits.

**Domain Oriented Relational Calculus:**

## Normalization

There are four types of anomalies, which are of concern, redundancy, insertion, deletion and updation. Normalization is not compulsory, but it is strongly

recommended that normalization must be done. Because normalized design makes the maintenance of database much easier. While carrying out the process of normalization, it should be applied on each table of database. It is performed after the logical database design. This process is also being followed informally during conceptual database design as well.

### *Normalization Process*

There are different forms or levels of normalization. They are called as first, second and so on. Each normalized form has certain requirements or conditions, which must be fulfilled. If a table or relation fulfills any particular form then it is said to be in that normal form. The process is applied on each relation of the database. The minimum form in which all the tables are in is called the normal form of entire database. The main objective of normalization is to place the database in highest form of normalization.

### *Summary*

In this lecture we have studied the different types of joins, with the help of which we can join different tables. We can get different types of outputs from joins. Then we studied relational calculus in which we briefly touched upon tuple and domain oriented relational calculus. Lastly we started the process of normalization which is a very important topic and we will discuss in detail this topic in the coming lectures.

### *Exercise:*

Draw two tables of PROJECT and EMPLOYEE along with different attribute, include a common attribute between the two to implement the PK/FK relationship and populate both the tables. Then apply all types of joins and observe the difference in the output relations

## Lecture No. 19

### Reading Material

“Database Systems Principles, Design and Implementation”  
written by Catherine Ricardo, Maxwell Macmillan.

Section 7.1 – 7.7

“Database Management Systems”, 2<sup>nd</sup> edition, Raghu Ramakrishnan, Johannes Gehrke,  
McGraw-Hill

### Overview of Lecture:

- Functional Dependency
- Inference Rules
- Normal Forms

In the previous lecture we have studied different types of joins, which are used to connect two different tables and give different output relations. We also started the basics of normalization. From this lecture we will study in length different aspects of normalization.

### Functional Dependency

Normalization is based on the concept of functional dependency. A functional dependency is a type of relationship between attributes.

### Definition of Functional Dependency

If A and B are attributes or sets of attributes of relation R, we say that B is functionally dependent on A if each value of A in R has associated with it exactly one value of B in R.

We write this as  $A \twoheadrightarrow B$ , read as “A functionally determines B” or “A determines B”. This does not mean that A causes B or that the value of B can be calculated from the value of A by a formula, although sometimes that is the case. It simply means that if we know the value of A and we examine the table of relation R, we will find only one value of B in all the rows that have the given value of A at any one time. Thus then the two rows have the same A value, they must also have the same B value. However, for a given B value, there may be several different A values. When a functional dependency exists, the attributes or set of attributes on the left side of the arrow is called a determinant. Attribute or set of attributes on left side are called determinant and on right are called dependants. If there is a relation R with attributes (a,b,c,d,e)

$a \twoheadrightarrow b,c,d$

$d \twoheadrightarrow e$

For Example there is a relation of student with following attributes. We will establish the functional dependency of different attributes: -

STD (stId,stName,stAdr,prName,credits)

$stId \twoheadrightarrow stName,stAdr,prName,credits$

$prName \twoheadrightarrow credits$

Now in this example if we know the stID we can tell the complete information about that student. Similarly if we know the prName , we can tell the credit hours for any particular subject.

### Functional Dependencies and Keys:

We can determine the keys of a relation after seeing its functional dependencies. The determinant of functional dependency that determines all attributes of that table is the super key. Super key is an attribute or a set of attributes that identifies an entity uniquely. In a table, a super key is any column or set of columns whose values can be used to distinguish one row from another. A minimal super key is the candidate key , so if a determinant of functional dependency determines all attributes of that relation then it is definitely a super key and if there is no other functional dependency whereas a subset of this determinant is a super key then it is a candidate key. So the functional dependencies help to identify keys. We have an example as under: -

EMP (eId,eName,eAdr,eDept,prId,prSal)

$eId \twoheadrightarrow (eName,eAdr,eDept)$

$eId,prId \twoheadrightarrow prSal$

Now in this example in the employee relation eId is the key from which we can uniquely determine the employee name address and department . Similarly if we know the employee ID and project ID we can find the project salary as well. So FDs help in finding out the keys and their relation as well.

## Inference Rules

Rules of Inference for functional dependencies, called inference axioms or Armstrong axioms, after their developer, can be used to find all the FDs logically implied by a set of FDs. These rules are sound, meaning that they are an immediate consequence of the definition of functional dependency and that any FD that can be derived from a given set of FDs using them is true. They are also complete, meaning they can be used to derive every valid reference about dependencies. Let A, B, C and D be subsets of attributes of a relation R then following are the different inference rules: -

### Reflexivity:

If B is a subset of A, then  $A \rightarrow B$ . This also implies that  $A \rightarrow A$  always holds.

Functional dependencies of this type are called trivial dependencies. For Example

$StName, stAdr \rightarrow stName$

$stName \rightarrow stName$

### Augmentation:

If we have  $A \rightarrow B$  then  $AC \rightarrow BC$ . For Example

If  $stId \rightarrow stName$  then

$StId, stAdr \rightarrow stName, stadr$

### Transitivity:

If  $A \rightarrow B$  and  $B \rightarrow C$ , then  $A \rightarrow C$

If  $stId \rightarrow prName$  and  $prName \rightarrow credits$  then

$stId \rightarrow credits$

### Additivity or Union:

If  $A \rightarrow B$  and  $A \rightarrow C$ , then  $A \rightarrow BC$

If  $empId \rightarrow eName$  and  $empId \rightarrow qual$  Then we can write it as

$empId \rightarrow qual$

### Projectivity or Decomposition

If  $A \rightarrow BC$  then  $A \rightarrow B$  and  $A \rightarrow C$

If  $empId \rightarrow eName, qual$  Then we can write it as

$empId \rightarrow eName$  and  $empID \rightarrow qual$

### Pseudo transitivity:

If  $A \rightarrow B$  and  $CB \rightarrow D$ , then  $AC \rightarrow D$

If  $stID \rightarrow stName$  and  $stName, fName \rightarrow stAdr$  Then we can write it as

$StId, fName \rightarrow stAdr$

## Normal Forms

Normalization is basically; a process of efficiently organizing data in a database. There are two goals of the normalization process: eliminate redundant data (for example, storing the same data in more than one table) and ensure data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored. We will now study the first normal form

**First Normal Form:**

A relation is in first normal form if and only if every attribute is single valued for each tuple. This means that each attribute in each row, or each cell of the table, contains only one value. No repeating fields or groups are allowed. An alternative way of describing first normal form is to say that the domains of attributes of a relation are atomic, that is they consist of single units that cannot be broken down further. There is no multivalued (repeating group) in the relation multiple values create problems in performing operations like select or join. For Example there is a relation of Student

**STD(stId,stName,stAdr,prName,bkId)**

stId	stName	stAdr	prName	bkId
S1020	Sohail Dar	I-8 Islamabad	MCS	B00129
S1038	Shoaib Ali	G-6 Islamabad	BCS	B00327
S1015	Tahira Ejaz	L Rukh Wah	MCS	B08945, B06352
S1018	Arif Zia	E-8, Islamabad.	BIT	B08474

Now in this table there is no unique value for every tuple, like for S1015 there are two values for bookId. So to bring it in the first normal form.

stId	stName	stAdr	prName	bkId
S1020	Sohail Dar	I-8 Islamabad	MCS	B00129
S1038	Shoaib Ali	G-6 Islamabad	BCS	B00327
S1015	Tahira Ejaz	L Rukh Wah	MCS	B08945
S1015	Tahira Ejaz	L Rukh Wah	MCS	B06352
S1018	Arif Zia	E-8, Islamabad.	BIT	B08474

Now this table is in first normal form and for every tuple there is a unique value.

**Second Normal Form:**

A relation is in second normal form (2NF) if and only if it is in first normal form and all the nonkey attributes are fully functionally dependent on the key. Clearly, if a relation is in 1NF and the key consists of a single attribute, the relation is automatically in 2NF. The only time we have to be concerned about 2NF is when the key is composite. Second normal form (2NF) addresses the concept of removing duplicative data. It remove subsets of data that apply to multiple rows of a table and place them in separate tables. It creates relationships between these new tables and their predecessors through the use of foreign keys.

**Summary**

Normalization is the process of structuring relational database schema such that most ambiguity is removed. The stages of normalization are referred to as normal forms and progress from the least restrictive (First Normal Form) through the most restrictive (Fifth Normal Form). Generally, most database designers do not attempt to implement anything higher than Third Normal Form or Boyce-Codd Normal Form.

We have started the process of normalization in this lecture. We will cover this topic in detail in the coming lectures.

**Exercise:**

Draw the tables of an examination system along with attributes and bring those relations in First Normal Form.



## Lecture No. 20

### Reading Material

“Database Systems Principles, Design and Implementation”  
written by Catherine Ricardo, Maxwell Macmillan.

Section 7.7 – 7.10

“Database Management Systems”, 2<sup>nd</sup> edition, Raghu Ramakrishnan, Johannes Gehrke,  
McGraw-Hill

### Overview of Lecture:

- Second and Third Normal Form
- Boyce - Codd Normal Form
- Higher Normal Forms

In the previous lecture we have discussed functional dependency, the inference rules and the different normal forms. From this lecture we will study in length the second and third normal forms.

### Second Normal Form

A relation is in second normal form if and only if it is in first normal form and all nonkey attributes are fully functionally dependent on the key. Clearly if a relation is in 1NF and the key consists of a single attribute, the relation is automatically 2NF.

The only time we have to be concerned 2NF is when the key is composite. A relation that is not in 2NF exhibits the update, insertion and deletion anomalies we will now see it with an example. Consider the following relation.

CLASS (crId, stId, stName, fld, room, grade)

crId, stId  $\longrightarrow$  stName, fld, room, grade

stId  $\longrightarrow$  stName

crId  $\longrightarrow$  fld, room

Now in this relation the key is course ID and student ID. The requirement of 2NF is that all non-key attributes should be fully dependent on the key there should be no

partial dependency of the attributes. But in this relation student ID is dependent on student name and similarly course ID is partially dependent on faculty ID and room, so it is not in second normal form. At this level of normalization, each column in a table that is not a determiner of the contents of another column must itself be a function of the other columns in the table. For example, in a table with three columns containing customer ID, product sold, and price of the product when sold, the price would be a function of the customer ID (entitled to a discount) and the specific product. If a relation is not in 2NF then there are some anomalies, which are as under:

-

- Redundancy
- Insertion Anomaly
- Deletion Anomaly
- Updation Anomaly

The general requirements of 2NF are:-

- Remove subsets of data that apply to multiple rows of a table and place them in separate rows.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

Consider the following table which has the anomalies:-

crId	StId	stName	fld	room	grade
C3456	S1020	Suhail Dar	F2345	104	B
C5678	S1020	Suhail Dar	F4567	106	
C3456	S1038	Shoaib Ali	F2345	104	A
C5678	S1015	Tahira Ejaz	F4567	106	B

Now the first thing is that the table is in 1NF because there are no duplicate values in any tuple and all cells contain atomic value. The first thing is the redundancy. Like in this table of CLASS the course ID C3456 is being repeated for faculty ID F2345 and similarly the room no 104 is being repeated twice. Second is the insertion anomaly. Suppose we want to insert a course in the table, but this course has not been registered to any student. But we cannot enter the student ID, because no student has registered this course yet. So we can also not insert this course. This is called as insertion anomaly which is wrong state of database. Next is the deletion anomaly. Suppose there is a course which has been enrolled by one student only. Now due to some

reason, we want to delete the record of student. But here the information about the course will also be deleted, so in this way this is the incorrect state of database in which infact we want to delete the information about the student record but along with this the course information has also been deleted. So it is not reflecting the actual system. Now the next is updation anomaly. Suppose a course has been registered by 50 students and now we want to change the class rooms of all the students. So in this case we will have to change the records of all the 50 students. So this is again a deletion anomaly. The process for transforming a 1NF table to 2NF is:

- Identify any determinants other than the composite key, and the columns they determine.
- Create and name a new table for each determinant and the unique columns it determines.
- Move the determined columns from the original table to the new table. The determinate becomes the primary key of the new table.
- Delete the columns you just moved from the original table except for the determinant which will serve as a foreign key.
- The original table may be renamed to maintain semantic meaning.

Now to remove all these anomalies from the table we will have to decompose this table, into different tables as under:

CLASS (crId, stId, stName, fld, room, grade)

crId, stId  $\longrightarrow$  stName, fld, room, grade

stId  $\longrightarrow$  stName      crId  $\longrightarrow$  fld, room

Now this table has been decomposed into three tables as under:-

STD (stId, stName)

COURSE (crId, fld, room)

CLASS (crId, stId, grade)

So now these three tables are in second normal form. There are no anomalies available now in this form and we say this as 2NF.

### Third Normal Form

A relational table is in third normal form (3NF) if it is already in 2NF and every non-key column is non-transitively dependent upon its primary key. In

other words, all nonkey attributes are functionally dependent only upon the primary key.

### Transitive Dependency

Transitive dependency is one that carries over another attribute. Transitive dependency occurs when one non-key attribute determines another non-key attribute. For third normal form we concentrate on relations with one candidate key, and we eliminate transitive dependencies. Transitive dependencies cause insertion, deletion, and update anomalies. We will now see it with an example:-

STD(stId, stName, stAdr, prName, prCrds)

stId  $\longrightarrow$  stName, stAdr, prName, prCrds

prName  $\longrightarrow$  prCrds

Now here the table is in second normal form. As there is no partial dependency of any attributes here. The key is student ID . The problem is of transitive dependency in which a non-key attribute can be determined by a non-key attribute. Like here the program credits can be determined by program name, which is not in 3NF. It also causes same four anomalies, which are due to transitive dependencies. For Example:-

### STUDENT

stId	stName	stAdr	prName	prCrds
S1020	Sohail Dar	I-8 Islamabad	MCS	64
S1038	Shoaib Ali	G-6 Islamabad	BCS	132
S1015	Tahira Ejaz	L Rukh Wah	MCS	64
S1015	Tahira Ejaz	L Rukh Wah	MCS	64
S1018	Arif Zia	E-8, Islamabad.	BIT	134

Now in this table all the four anomalies are exists in the table. So we will have to remove these anomalies by decomposing this table after removing the transitive dependency. We will see it as under: -

STD (stId, stName, stAdr, prName, prCrds)

stId  $\longrightarrow$  stName, stAdr, prName, prCrds

prName  $\longrightarrow$  prCrds

The process of transforming a table into 3NF is:

- Identify any determinants, other the primary key, and the columns they determine.
- Create and name a new table for each determinant and the unique columns it determines.
- Move the determined columns from the original table to the new table. The determinate becomes the primary key of the new table.
- Delete the columns you just moved from the original table except for the determinate which will serve as a foreign key.
- The original table may be renamed to maintain semantic meaning.

STD (stId, stName, stAdr, prName)

PROGRAM (prName, prCrds)

We have now decomposed the relation into two relations of student and program. So the relations are in third normal form and are free of all the anomalies

### Boyce - Codd Normal Form

A relation is in Boyce-Codd normal form if and only if every determinant is a candidate key. A relation R is said to be in BCNF if whenever  $X \rightarrow A$  holds in R, and A is not in X, then X is a candidate key for R. It should be noted that most relations that are in 3NF are also in BCNF. Infrequently, a 3NF relation is not in BCNF and this happens only if

- (a) the candidate keys in the relation are composite keys (that is, they are not single attributes),
- (b) there are more than one candidate keys in the relation, and
- (c) the keys are not disjoint, that is, some attributes in the keys are common.

The BCNF differs from the 3NF only when there are more than one candidate keys and the keys are composite and overlapping. Consider for example, the relationship:

enrol (sno, sname, cno, cname, date-enrolled)

Let us assume that the relation has the following candidate keys:

(sno,cno)  
(sno,cname)  
(sname,cno)  
(sname, cname)

(we have assumed sname and cname are unique identifiers). The relation is in 3NF but not in BCNF because there are dependencies

sno -> sname  
cno -> cname

Where attributes are part of a candidate key are dependent on part of another candidate key. Such dependencies indicate that although the relation is about some entity or association that is identified by the candidate keys e.g. (sno, cno), there are attributes that are not about the whole thing that the keys identify. For example, the above relation is about an association (enrolment) between students and subjects and therefore the relation needs to include only one identifier to identify students and one identifier to identify subjects. Provided that two identifiers about the students (sno, sname) and two keys about subjects (cno, cname) means that some information about the students and subjects which is not needed is being provided. This provision of information will result in repetition of information and the anomalies that we discussed at the beginning of this chapter. If we wish to include further information about students and courses in the database, it should not be done by putting the information in the present relation but by creating new relations that represent information about entities student and subject.

These difficulties may be overcome by decomposing the above relation in the following three relations:

(sno, sname)  
(cno, cname)  
(sno, cno, date-of-enrolment)

We now have a relation that only has information about students, another only about subjects and the third only about enrolments. All the anomalies and repetition of information have been removed.

## Higher Normal Forms

After BCNF are the fourth, a fifth and domain key normal form exists. Although till BCNF normal form tables are in required form, but if we want we can move on to fourth and fifth normal forms as well. 4NF deals with multivalued dependency, fifth deals with possible loss less decompositions; DKNF reduces further chances of any possible inconsistency.

### Summary

The goal of normalization is to create a set of relational tables that are free of redundant data and that can be consistently and correctly modified. This means that all tables in a relational database should be in the third normal form (3NF). A relational table is in 3NF if and only if all non-key columns are (a) mutually independent and (b) fully dependent upon the primary key. Mutual independence means that no non-key column is dependent upon any combination of the other columns. The first two normal forms are intermediate steps to achieve the goal of having all tables in 3NF. In order to better understand the 2NF and higher forms, it is necessary to understand the concepts of functional dependencies and loss less decomposition.

### Exercise:

The tables of Examination System which were brought in 1NF in previous lecture bring those tables into 2 and 3NF.

## Lecture No. 21

### Reading Material

“Database Systems Principles, Design and Implementation” written by Catherine Ricardo, Maxwell Macmillan.	Page 238
“Modern Database Management”, Fred McFadden, Jeffrey Hoffer, Benjamin/Cummings	Chapter 6

### Overview of Lecture:

- Summary of normalization
- A normalization example
- Introduction to physical DB design phase

### Normalization Summary

Normalization is a step by step process to make DB design more efficient and accurate. A normalized database helps the DBA to maintain the consistency of the database. However, the normalization process is not a must, rather it is a strongly recommended activity performed after the logical DB design phase. Not a must means, that the consistency of the database can be maintained even with an un-normalized database design, however, it will make it difficult for the designer. Un-normalized relations are more prone to errors or inconsistencies.

The normalization is based on the FDs. The FDs are not created by the designer, rather they exist in the system being developed and the designer identifies them. Normalization forms exist up to 6NF starting from 1NF, however, for most of the situations 3NF is sufficient. Normalization is performed through Analysis or Synthesis process. The input to the process is the logical database design and the FDs that exist in the system. Each individual table is checked for the normalization considering the relevant FDs; if any normalization requirement for a particular normal form is being violated, then it is sorted out generally by splitting the table. The

process is applied on all the tables of the design hence the database is called to be in a particular normal form.

### Normalization Example

In the following an example of normalization process has been discussed. This example is taken from Ricardo book, page 238. The example comprehensively explains the stages of the normalization process. The approach adopted for the normalization is analysis approach, whereby a single large table is assumed involving all the attributes required in the system. Later, the table is decomposed into smaller tables by considering the FDs existing in the system. As has been discussed before, the FDs have to be identified by the designer they are not described as regular from by the users. So the example also explains the transforming of real-world scenarios into FDs.

An example table is given containing all the attributes that are used in different applications in the system under study. The table named WORK consists of the attributes:

WORK (projName, projMgr, empId, hours, empName, budget, startDate, salary, empMgr, empDept, rating)

The purpose of most of the attributes is clear from the name, however, they are explained in the following facts about the system. The facts mentioned in the book are italicized and numbered followed by the explanation.

*1- Each project has a unique name, but names of employees and managers are not unique.*

This fact simply illustrates that values in the projName attribute will be unique so this attribute can be used as identifier if required however the attributes empName, empMgr and projMgr are not unique so they cannot be used as identifiers

*2- Each project has one manager, whose name is stored in projMgr*

The projMgr is not unique as mentioned in 1, however, since there is only one manager for a project and project name is unique, so we can say that if we know the project name we can determine a single project manager, hence the FD

projName  $\rightarrow$  projMgr

3- *Many employees may be assigned to work on each project, and an employee may be assigned to more than one project. The attribute 'hours' tells the number of hours per week that a particular employee is assigned to work on a particular project.*

Since there are many employees working on each project so the projName attribute cannot determine the employee working on a project, same is the case with empId that it cannot determine the particular project an employee is working since one employee is working on many projects. However, if we combine both the empId and projName then we can determine the number of hours that an employee worked on a particular project within a week, so the FD

empId, projName  $\rightarrow$  hours

4- *Budget stores the budget allocated for a project and startDate stores the starting date of a project*

Since the project name is unique, so if we know the project name we can determine the budget allocated for it and also the starting date of the project

projName  $\rightarrow$  budget, startDate

5- *Salary gives the annual salary of the employee*

empId  $\rightarrow$  salary, empName

Although empId has not been mentioned as unique, however, it is generally assumed that attribute describing Id of something are unique, so we can define the above FD.

6- *empMgr gives the name of the employee's manager, who is not the same as project manager.*

Project name is determined by project name, however one employee may work on many projects, so we can not determine the project manager of an employee through the Id of employee. However, empMgr is the manager of employee and can be known from employee Id, so FD in 5 can be extended

$\text{empId} \rightarrow \text{salary, empName, empMgr}$

7- *empDept give the employee's department. Department names are unique. The employee's manager is the manager of the employee's department.*

$\text{empDept} \rightarrow \text{empMgr}$

Because empDept is unique and there is one manager for each department. At the same time, because each employee works in one department, we can also say that

$\text{empId} \rightarrow \text{empDept}$  so the FD in 6 is further extended

$\text{empId} \rightarrow \text{salary, empName, empMgr, empDept}$

8- *Rating give the employee's rating for a particular project. The project manager assigns the rating at the end of employee's work on that project*

Like 'hours' attribute, the attribute 'rating' is also determined by two attributes, the projName and empId, because many employees work on one project and one employee may work on many projects. So to know the rating of an employee on a particular project we need to know the both, so the FD

$\text{projName, empId} \rightarrow \text{rating}$

In all we have the following four FDs:

- 1)  $\text{empId} \rightarrow \text{salary, empName, empMgr, empDept}$
- 2)  $\text{projName, empId} \rightarrow \text{rating, hours}$
- 3)  $\text{projName} \rightarrow \text{projMgr, budget, startDate}$
- 4)  $\text{empDept} \rightarrow \text{empMgr}$

Normalization

So we identified the FDs in our example scenario, now to perform the normalization process. For this we have to apply the conditions of the normal forms on our tables.

Since we have got just one table to begin with so we start our process on this table:

WORK(projName, projMgr, empId, hours, empName, budget, startDate, salary, empMgr, empDept, rating)

#### First Normal Form:

Seeing the data in the example in the book or assuming otherwise that all attributes contain the atomic value, we find out the table is in the 1NF.

#### Second Normal Form:

Seeing the FDs, we find out that the PK for the table is a composite one, comprising of empId, projName. We did not include the determinant of fourth FD, that is, the empDept, in the PK because empDept is dependent on empId and empID is included in our proposed PK. However, with this PK (empID, projName) we have got partial dependencies in the table through FDs 1 and 3 where we see that some attributes are being determined by subset of our PK which is the violation of the requirement for the 2NF. So we split our table based on the FDs 1 and 3 as follows:

PROJECT (~~projName~~, projMgr, startDate)

EMPLOYEE (~~empId~~, empName, salary, empMgr, empDept)

WORK (~~projName~~, ~~empId~~, hours, rating)

All the above three tables are in 2NF since they are in 1NF and there is no partial dependency in them.

#### Third Normal Form

Seeing the four FDs, we find out that the tables are in 2NF and there is no transitive dependency in PROJECT and WORK tables, so these two tables are in 3NF. However, there is a transitive dependency in EMPLOYEE table since FD 1 say empId → empDept and FD 4 say empDept → empMgr. To remove this transitive dependency we further split the EMPLOYEE table into following two:

EMPLOYEE (~~empId~~, empName, salary, empDept)

DEPT (~~empDept~~, empMgr)

Hence finally we got four tables

PROJECT (projName, projMgr, startDate)

EMPLOYEE (empId, empName, salary, empDept)

WORK (projName, empId, hours, rating)

DEPT (empDept, empMgr)

These four tables are in 3NF based on the given FD, hence the database has been normalized up to 3NF.

### Physical Database Design

After completing the logical database design and then normalizing it, we have to establish the physical database design. Throughout the processes of conceptual and logical database designs and the normalization, the primary objective has been the storage efficiency and the consistency of the database. So we have been following good design principles. In the physical database design, however, the focus shifts from storage efficiency to the efficiency in execution. So we deliberately violate some of the rules that we studied earlier, however, this shift in focus should never ever lead to incorrect state of the database. The correctness of the database we have to maintain in any case. When we do not follow the good design principles then it makes it difficult to maintain the consistency or correctness of the database. Since the violation is deliberate, that is, we are aware of the dangers due to violations and we know the reasons for these violations so we have to take care of the possible threats and adopt appropriate measures. Finally, there are different possibilities and we as designers have to adopt particular ones based on certain reasons or objectives. We have to be clear about our objectives.

The physical DB design involves:

- Transforms logical DB design into technical specifications for storing and retrieving data
- Does not include practically implementing the design however tool specific decisions are involved

It requires the following input:

- Normalized relations (the process performed just before)

- Definitions of each attribute (means the purpose or objective of the attributes. Normally stored in some form of data dictionary or a case tool or may be on paper)
- Descriptions of data usage (how and by whom data will be used)
- Requirements for response time, data security, backup etc.
- Tool to be used

Decisions that are made during this process are:

- Choosing data types (precise data types depend on the tool to be used)
- Grouping attributes (although normalized)
- Deciding file organizations
- Selecting structures
- Preparing strategies for efficient access

That is all about today's lecture, the discussion continues in the next lecture.

### Summary

In today's lecture we summarized the normalization process and also saw an example to practically implement the process. We have introduced our next topic that is the physical DB design. We will discuss this topic in the lectures to be followed.

## Lecture No. 22

### Overview of Lecture

- Data Volume and Usage Analysis
- Designing Fields
- Choosing Data Type
- Coding Techniques
- Coding Example
- Controlling Data Integrity

### The Physical Database Design Considerations and Implementation

The physical design of the database is one of the most important phases in the computerization of any organization. There are a number of important steps involved in the physical design of the database. Steps are carried out in sequence and need to be performed precisely so that the result of the first step is properly used as input to the next step.

Before moving onto the Physical database design the design of the database should have undergone the following steps,

Normalization of relations

Volume estimate

Definition of each attribute

Description of where and when data is used (with frequencies)

Expectation or requirements of response time and data security.

Description of the technologies.

For the physical database design we need to check the usage of the data in term of its size and the frequency. This critical decision is to be made to ensure that proper structures are used and the database is optimized for maximum performance and efficiency.

The following steps are necessary once we have the prerequisite complete:

Select the appropriate attribute and a corresponding data type for the attribute.

The process of selecting the attribute to be placed in a specific relation in the physical design. Need considerable care as it is one of the most important and basic aspects for the creation of the database.

Grouping of attributes in the logical order so that the relation is created in such a way that no information is missing from the relation and also no redundant or unnecessary information is placed in the relation.

Looking at the logical design at the time of transformation into physical design there may be stages when the information combined logically in the logical design looks odd when transforming the design into a physical one.

Arrangement of Similar records into the secondary memory (hard disk)

The scheme of storage on hard disk is important as it leads to the efficiency and management of the data on disk. Different types of data access mechanism are available and are useful for rapid access, storage, and modification of data. Different types of database structures can be used for placement of data on disks, management of data in the forms of indexes and different database architecture is vital and leads to better retrieval and recovery of records.

Preparing queries and handling strategies for the proper usage of the database, so that any type of input or output operation performed on the database is executed in an optimized and efficient way.

## DESIGNING FIELDS

Field is the smallest unit of application data recognized by system software, such as a programming language or any database management system.

Designing fields in the databases' physical design as discussed earlier is a major issue and needs to be dealt with great care and accuracy. Data types are the structure defined for placing data in the attributes. Each data type is appropriate for use with certain type of data.

4 major objectives for using data types when specifying attributes in a database are given as under:

- Minimized usage of storage space
- Represent all possible values
- Improve data integrity
- Support all data manipulation

The correct data type selection and decision for proper domain of the attribute is very necessary as it provides a number of benefits.

Most common data types used in the available DBMS of the day have the following set of common attributes.

Data type	Description	Max PL/SQL	Size:
VARCHAR2(size)	Variable length character string having maximum length <i>size</i> . You must specify size	32767 bytes. minimum is 1	bytes
VARCHAR	Now deprecated - VARCHAR is a synonym for VARCHAR2 but this usage may change in future versions.		
CHAR(size)	Fixed length character data of length <i>size</i> bytes. This should be used for fixed length data. Such as codes A100, B102...	32767 Default	bytes and
		minimum size is 1 byte.	
NUMBER(p,s)	Magnitude 1E-130 .. 10E125 maximum precision of 126 binary digits, which is roughly equivalent to 38 decimal digits The scale <i>s</i> can range from -84 to 127. For floating point don't specify <i>p,s</i> REAL has a maximum precision of 63 binary digits, which is roughly equivalent to 18 decimal digits		

LONG	Character data of variable length (A bigger version the VARCHAR2 data type)	32760 bytes Note this is smaller than the maximum width of a LONG column
DATE	Valid date range	from January 1, 4712 BC to December 31, 9999 AD. (in Oracle7 = 4712 AD)
RAW(size)	Raw binary data of length size bytes. You must specify size for a RAW value.	32767 bytes
LONG RAW	Raw binary data of variable length. (not interpreted by PL/SQL)	32760 bytes Note this is smaller than the maximum width of a LONG RAW column
BLOB	Binary Large Object	4Gigabytes

### CODING AND COMPRESSION TECHNIQUES:

There are some attributes which have some sparse set of values, these values when they are represented in any data type are hard to express, for this purpose some codes are used. As the codes defined by the database administrator or the programmer consume less space so they are better for use in situations where we have large number of records and wastage of small amount of space in each record can lead to loss of huge amount of data storage space. Thus causing lowered database efficiency.

<i>STID</i>	<i>STNAME</i>	<i>HOBBY</i>
S1020	Sohail Dar	Reading
S1038	Shoaib Ali	Gardening
S1015	Tahira Ejaz	Reading
S1015	Tahira Ejaz	Movie
S1018	Arif Zia	Reading

Coding techniques are also useful for compression of data values appearing the data, by replacing those data values with the smaller sized codes we can further reduce the space needed by the data for storage in the database.

Following tables give the use of codes and their utilization in the database environment

Coding Example:

Student

<i>STID</i>	<i>STNAME</i>	<i>HOBBY</i>
S1020	Sohail Dar	R
S1038	Shoaib Ali	G
S1015	Tahira Ejaz	R
S1015	Tahira Ejaz	M
S1018	Arif Zia	R

### Hobby Table

<i>CODE</i>	<i>HOBBY</i>
R	Reading
G	Gardening
M	Movies

In the above example we have seen the implementation of the codes as replacement to the data in the actual table, here we actually allocated codes to different hobbies and then replace the codes instead of writing the codes in the table.

We get a number of benefits by the use of data types and the benefit can be in a number of dimensions.

#### *Default value*

Default values are the values which are associated with a specific attribute and can help us to reduce the chances of inserting incorrect values in the attribute space. And also it can help us preventing the attribute value be left empty.

#### *Range Control*

Range control implemented over the data can be very easily achieved by using any data type. As the data type enforces the entry of data in the field according to the limitations of the data type.

#### *Null Value Control*

As we already know that a null value is an empty value and is distinct from zero and spaces, Databases can implement the null value control by using the different data types or their build mechanisms.

#### *Referential Integrity*

Referential Integrity means to keep the input values for a specific attribute in specific limits in comparison to any other attribute of the same or any other relation.

ns by creating indexes that are independent of constraints.

The performance benefits of indexes, however, do come with a cost. Tables with indexes require more storage space in the database. Also, commands that insert, update, or delete data can take longer and require more processing time to maintain the indexes. When you design and create indexes, you should ensure that the performance benefits outweigh the extra cost in storage space and processing resources.

### Index Classification

Indexes are classified as under:

- Clustered vs. Un-clustered Indexes
- Single Key vs. Composite Indexes
- Tree-based, inverted files, pointers

#### Primary Indexes:

Consider a table, with a Primary Key Attribute being used to store it as an ordered array (that is, the records of the table are stored in order of increasing value of the Primary Key attribute.)As we know, each BLOCK of memory will store a few records of this table. Since all search operations require transfers of complete blocks, to search for a particular record, we must first need to know which block it is stored in. If we know the address of the block where the record is stored, searching for the record is very fast. Notice also that we can order the records using the Primary Key attribute values. Thus, if we just know the primary key attribute value of the first record in each block, we can determine quite quickly whether a given record can be found in some block or not. This is the idea used to generate the Primary Index file for the table.

#### Secondary Indexes:

Users often need to access data on the basis of non-key or non-unique attribute; secondary key. Like student name, program name, students enrolled in a particular program .Records are stored on the basis of key attribute; three possibilities

- Sequential search
- Sorted on the basis of name

- Sort for command execution

A part from primary indexes, one can also create an index based on some other attribute of the table. We describe the concept of Secondary Indexes for a Key attribute that is, for an attribute which is not the Primary Key, but still has unique values for each record of the table. The idea is similar to the primary index. However, we have already ordered the records of our table using the Primary key. We cannot order the records again using the secondary key (since that will destroy the utility of the Primary Index). Therefore, the Secondary Index is a two column file, which stores the address of every tuple of the table.

Following are the three-implementation approaches of Indexes:

- Inverted files or inversions
- Linked lists
- B+ Trees

Creating Index

```
CREATE [ UNIQUE ]
      [ CLUSTERED | NONCLUSTERED ]
INDEX index_name
ON { table | view } ( column [ ASC | DESC ] [ ,...n ]
```

We will now see an example of creating index as under:

```
CREATE UNIQUE INDEX pr_prName
ON program(prName)
```

It can also be created on composite attributes. We will see it with an example.

```
CREATE UNIQUE INDEX
St_Name ON
Student(stName ASC, stFName DESC)
```

Properties of Indexes:

Following are the major properties of indexes:

- Indexes can be defined even when there is no data in the table
- Existing values are checked on execution of this command
- It support selections of form as under:  
field <operator> constant

- It support equality selections as under:  
Either “tree” or “hash” indexes help here.
- It support Range selections (operator is one among <, >, <=, >=, BETWEEN)

## Summary

In today's lecture we have discussed the indexes, its classification and creating the indexes as well. The absence or presence of an index does not require a change in the wording of any SQL statement. An index merely offers a fast access path to the data; it affects only the speed of execution. Given a data value that has been indexed, the index points directly to the location of the rows containing that value. Indexes are logically and physically independent of the data in the associated table. You can create or drop an index at anytime without affecting the base tables or other indexes. If you drop an index, all applications continue to work; however, access to previously indexed data might be slower. Indexes, being independent structures, require storage space.



CS403-Database Management System  
Solved MCQS for Mid terms papers  
Solved by JUNAID MALIK and Team



**AL-Junaid Institute**  
Contact no: 03041659294



## LMS Handling

### *Subject Enrollment*

- 1) *Online classes available*  
*HTML, CSS, Javascript*  
*Phyton..etc*
- 2) *Solved Graded Activities*
  - *Assignment's*
  - *Quizzes*
  - *GDB's*
- 3) *Solved Quiz File's*
- 4) *Short Note's*
- 5) *Past paper's &*  
*Current paper's*

**Website Link**  
[vulmshelp.com](http://vulmshelp.com)

### *Final project CS619*

- 1) *SRS*  
*(Software Requirement's*  
*Specification)*
- 2) *DD*  
*(Design Document)*
- 3) *Test phase + viva*
- 4) *Viva preparation*
- 5) *Final Deliverable*

**Gmail**  
[junaidfazal08@gmail.com](mailto:junaidfazal08@gmail.com)

# AL-JUNAID INSTITUTE GROUP

1. Which is not the function of Database Management System?
  - a. A user accessible Catalog
  - b. Concurrency Control Services
  - c. **Failure Response Service**
  - d. Support for Data Communication
2. The conceptual database design can be transformed into any \_\_\_\_\_.
  - a. **Data model**
  - b. Database design
  - c. E-R model
  - d. Entity set
3. Which is the second stage in the database development process?
  - a. Database design
  - b. Preliminary Study
  - c. Physical Design
  - d. **Requirement Analysis**
4. Functional dependencies under interference rule reflexivity are called \_\_\_\_\_ dependencies.
  - a. Augmentation
  - b. Transitivity
  - c. Projectivity
  - d. **Trivial**
5. If many instances of the “Book” entity are associated with the many instances of the “Student” entity, then the relationship between them is:
  - a. One to one
  - b. One to many
  - c. **Many to many**
  - d. Many to one
6. Which of the following constraints enforces entity integrity?
  - a. **PRIMARY KEY**
  - b. FOREIGN KEY
  - c. CHECK
  - d. NOT NULL

# **AL-JUNAID INSTITUTE GROUP**

7. Unary operations involves
- a. **Only one relation**
  - b. Only two relations
  - c. More than two relations
  - d. Not more than five relations
8. A \_\_\_\_\_ entity has a primary key that is partially or totally derived from the parent entity in the relationship.
- a. Strong
  - b. **Weak**
  - c. Business
  - d. Relationship
9. In \_\_\_\_\_ appropriate data model is chosen.
- a. ERD
  - b. Conceptual Database Design
  - c. **Logical Database Design**
  - d. DFA
10. Which of the following is drawback of standardization?
- a. Difficult to understand
  - b. Uniformity
  - c. **Lack of uniqueness**
  - d. Easy to develop applications
11. Recursive relationship is also called \_\_\_\_\_.
- a. **Unary**
  - b. Ternary
  - c. Primary
  - d. Secondary
12. The foreign key attribute, which is present as a primary key in another relation is called as \_\_\_\_\_ of the foreign key attribute.
- a. Binary relation
  - b. Composite relation
  - c. **Home relation**
  - d. Recursive relation

# AL-JUNAID INSTITUTE GROUP

13. **DFD** does not provide us a way to represent \_\_\_\_\_
- Entities
  - Decisions** AL-JUNAID INSTITUTE OF GROUP
  - Date stores
  - Processes
14. A \_\_\_\_\_ is used to maintain a connection between the users of the database system.
- Virtual server
  - File server**
  - Client server
  - FTP server
15. Which of the following is an advantage of using the Traditional File Organization?
- Time consuming
  - Data security
  - Simplicity**
  - Efficiency
16. Which of the following is the input to Analysis Phase?
- DFD**
  - Logical database design
  - Physical database design
  - Data model
17. Name and \_\_\_\_\_ are part of the definitions of an attribute.
- Domain
  - Symbol
  - Property
  - Entity instance
18. Which of the following most certainly implies the need for an entire table to implement?
- A ternary relationship**
  - A recursive relationship
  - An identify relationship
  - A binary relationship
19. Core of the Database Architecture is \_\_\_\_\_

# AL-JUNAID INSTITUTE GROUP

- a. Internal level  
b. External level  
c. Conceptual level  
d. ANSI SPARK
20. \_\_\_\_\_ is same as equi-join with a slight difference.  
a. Natural join  
b. Semi join  
c. Outer join  
d. Theta join
21. Conceptual database design is implemented using a \_\_\_\_\_.  
a. Hierarchical data model  
b. Semantic data model  
c. Relational data model  
d. Network data model
22. Which constraint is generally used for efficiency purposes in the data entry process?  
a. Null value  
b. Integer value  
c. Default value  
d. String value
23. A 3 NF relation is converted to BCNF by  
a. Removing composite keys  
b. Removing multivalued dependencies  
c. Dependent attributes of overlapping composite keys are put in a separate relation  
d. Dependent non-key attributes are put in a separate table
24. Each \_\_\_\_\_ of a table contains atomic/single value.  
a. Cell  
b. Row  
c. Column  
d. Record
25. A relation is said to be in BCNF when  
a. It has overlapping composite keys

# **AL-JUNAID INSTITUTE GROUP**

- b. It has no composite keys
  - c. It has no multivalued dependencies
  - d. **It has no overlapping composite keys which have related attributes**
26. Which one of the following is not a part of three-level architecture?
- a. External level
  - b. Conceptual level
  - c. Internal level
  - d. **Physical level**
27. Which of the following is INCORRECT about naming entities?
- a. Singular nouns are used
  - b. Write in capitals
  - c. Organization specific names can be used
  - d. **Always use abbreviations as they are convenient**
28. An instance is
- a. A particular occurrence of an entity
  - b. A special type of relation
  - c. An attribute of an entity
  - d. **Any particular entity** **AL-JUNAID INSTITUTE OF GROUP**
29. \_\_\_\_\_ is an example of metadata
- a. Ali
  - b. Address
  - c. Person
  - d. **Student**
30. Level of data at which entities or objects exist in reality is called
- a. **Real world data**
  - b. Schema for real world data
  - c. Meta data
  - d. Data about data

# AL-JUNAID INSTITUTE GROUP

31. There are \_\_\_\_\_ basic properties of the database relations.
- Four
  - Three
  - Six
  - Two
32. How many types of dependencies are?
- One
  - Two
  - Three
  - Four
33. In \_\_\_\_\_ join, only selected rows of a relation are made cross product with second relation.
- Normal
  - Theta
  - Outer
  - Semi
34. Which of the following is not a key?
- Super key
  - Alternate key
  - Integrity constraint key
  - Primary key
35. A collection of concepts that can be used to describe the structure of a database
- Database
  - DBMS
  - Data model
  - Data
36. Any combination of attributes with super key is \_\_\_\_\_
- Primary key
  - Foreign key
  - Super key
  - Candidate key
37. Binary relationships are those, which are established between

# **AL-JUNAID INSTITUTE GROUP**

- a. One entity type
  - b. Two entity type**
  - c. Three entity type
  - d. Unlimited entities
38. \_\_\_\_\_ are called participants, when enrolled in a relationship.
- a. Attributes
  - b. Relations
  - c. Entities**
  - d. Keys
39. In case of Context-level Diagram, the system is represented by
- a. One process atleast
  - b. Two processes atleast
  - c. One process only**
  - d. Any number of processes
40. Collection of raw facts is called \_\_\_\_\_.
- a. Information
  - b. Object
  - c. Output
  - d. Data**
41. A software package designed to store and manage database
- a. Database
  - b. DBMS**
  - c. Data model
  - d. Data
42. In \_\_\_\_\_ join common attributes appear twice in the output with selected rows.
- a. Normal join
  - b. Semi join
  - c. Natural join
  - d. Equi join**
43. Within a table, each primary key value \_\_\_\_\_.
- a. Is a minimal super key
  - b. Is always the first field in each table

# AL-JUNAID INSTITUTE GROUP

- c. Must be numeric  
d. Must be unique
44. \_\_\_\_\_ is not an operation of relational algebra.  
a. Addition  
b. Union  
c. Cartesian product  
d. Set difference
45. Candidate keys which are not selected as primary key are called \_\_\_\_\_.  
a. Secondary keys  
b. Foreign keys  
c. Alternate keys  
d. Composite keys
46. Major benefits of data sharing is:  
a. Data of same applications placed at same place.  
b. Data of same applications placed at different places  
c. Data of different application places at same place  
d. Data of different applications stored at placed place
47. Functional dependency between two attributes or sets of attributes i.e. Q and P is represented as \_\_\_\_\_.  
a.  $Q \rightarrow P$   
b.  $Q \ P$   
c.  $Q * P$   
d.  $Q + P$
48. Which of the following is not the component of Relational Data Model?  
a. Integrity Constraint  
b. Manipulation Language  
c. Relation / Table  
d. Domain Constraint
49. The attribute that is calculated from other attribute is called \_\_\_\_\_.  
a. Derived attribute  
b. Multi-valued attribute

# AL-JUNAID INSTITUTE GROUP

- c. Composite attribute
  - d. Simple attribute
50. Consider the two attributes “CNIC\_NO” and “Registration\_NO”. Both the attributes can be used for the unique identification. Now if the user selects the “CNIC\_NO” as primary key, then “Registration\_NO” is a \_\_\_\_ key.
- a. Foreign
  - b. Secondary
  - c. Primary
  - d. Alternative**
51. Which of the following constraint is included in the definition of the table?
- a. Default value
  - b. Foreign key
  - c. Null**
  - d. Domain
52. The database deals with large amount of data. This property is called
- a. Database design
  - b. Scalability**
  - c. Concurrency
  - d. Robustness
53. Which one of the following is NOT a characteristic of meta data?
- a. Data about data
  - b. Describes a data dictionary
  - c. Self-describing
  - d. Includes user data**
54. Theta join is denoted mathematically as \_\_\_\_\_.
- a.  $R \times S$
  - b.  $R \times$
  - c.  $R \times \emptyset S$**
  - d.  $R \times R S$

# AL-JUNAID INSTITUTE GROUP

55. In cross reference matrix transitions are specified on \_\_\_\_\_ and database objects are specified on \_\_\_\_\_.
- Y axis, X axis
  - X axis, Y axis
  - X axis, X axis
  - Y axis, Y axis
56. Each component of DBMS performs \_\_\_\_\_ functions.
- Similar
  - Different
  - Identical
  - Computational
57. Which of the following data model is not record based?
- Hierarchical Data Model
  - Network Data model
  - Semantic Data model
  - Relational Data model
58. The basic purpose of diamond in ER diagram is to show the \_\_\_\_\_.
- Attributes
  - Relationships
  - Composite attribute
  - Entity set
59. Which of the following tool is used to design conceptual database design?
- SQL
  - MS Access
  - Oracle
  - This model is independent of any tool
60. In “one to many” cardinality one instance of a relation is mapped with \_\_\_\_\_.
- Many instances of second entity type
  - One instance of second entity type
  - Primary key of second relation
  - Foreign key of second relation

# AL-JUNAID INSTITUTE GROUP

61. Cross reference matrix is developed at the \_\_\_\_\_ stage of the database.
- Analysis
  - Designing**
  - Implementation
  - Application
62. The population of the data of the organization for which the database is created is called the \_\_\_\_\_ of the database.
- Extension**
  - Intension
  - Model
  - Schema
63. User rights information is stored in \_\_\_\_\_.
- Physical database
  - Catalog**
  - Logical database
  - Buffer
64. Logical data independency provides independency to change \_\_\_\_\_ model.
- External
  - Logical
  - Conceptual**
  - Internal
65. In natural join the common attribute in the output table appears \_\_\_\_\_.
- Twice
  - Three times
  - Four times
  - Once**
66. Which levels are mostly used for detailed DFD?
- Level-0, Level-1**
  - Level-1, Level-2
  - Level-2, Level-3

# AL-JUNAID INSTITUTE GROUP

- d. Level-3, Level-4
67. By default, a non-key attribute in a relation can have \_\_\_\_\_ value.
- a. Binary
  - b. Null**
  - c. Character
  - d. Integer
68. The three levels architecture is useful for \_\_\_\_\_.
- a. Arranging the data in an organized manner
  - b. Hiding the details of internal systems**
  - c. Breaking the data access restriction
  - d. Storing definitions of the structures
69. With the help of \_\_\_\_\_ technique data values with the smaller sized codes can further reduce the space needed by the data for storage in the database.
- a. Compression**
  - b. Expansions
  - c. Negation
  - d. Extension
70. If an entity is linked with itself, than it is called \_\_\_\_\_ relationship.
- a. Binary
  - b. Ternary
  - c. Nary
  - d. Recursive**
71. Duplication of data in controlled redundancy is controlled and \_\_\_\_\_.
- a. Deliberate**
  - b. Necessary
  - c. Precise
  - d. Unintentional
72. Teleprocessing is a sub-type of \_\_\_\_\_.
- a. Single user database
  - b. Multi-user database**

# **AL-JUNAID INSTITUTE GROUP**

- c. File server
  - d. Client server
73. In an ERD, the focus is on the \_\_\_\_\_ and the relationships between them.
- a. Attributes
  - b. Entities**
  - c. Keys
  - d. Indexes
74. The most widely used conceptual model is the \_\_\_\_\_ model.
- a. Implementation
  - b. ER**
  - c. Object oriented
  - d. Internal
75. Student and book are two entities, what is the cardinality between the two?
- a. Many to one
  - b. One to many
  - c. Many to many**
  - d. One to one
76. RDM is a simple there is just one structure and that is \_\_\_\_\_.
- a. A relation or a table**
  - b. E-R model
  - c. Attributes
  - d. Data model
77. A description on a particular collection of data using the given data model.
- a. Database**
  - b. Relation
  - c. Schema**
  - d. None
78. Consider thy key “CNIC, Name” and choose the best option that describes the key most effectively.
- a. The given key is a super key and alternate key.

# AL-JUNAID INSTITUTE GROUP

- b. The given key is not a super key but a candidate key.
  - c. The given key is a super key and candidate key as well.**
  - d. The given key is a super key but not a candidate key.
79. Normalization is a process of restricting a relation to
- a. Minimize duplication of data in a database**
  - b. Maximize duplication of data to ensure reliability
  - c. Make it of uniform size
  - d. Allow addition of data
80. Unary relationship involves a single entity it is also called \_\_\_\_\_.
- a. One to one relationship
  - b. One to many relationship
  - c. Many to many relationship
  - d. Recursive relationship**
81. Which of the following is a part of a DFD but not included in the ERD?
- a. The attribute having a single instance
  - b. The external entity having a single instance**
  - c. The relationship having a single instance
  - d. The role having a single instance
82. The database management system (DBMS) is used to
- a. Store the data
  - b. Access the data
  - c. Manage its users
  - d. All of above**
83. In cross reference matrix attributes are mention on \_\_\_\_\_ and reports are mention on \_\_\_\_\_.
- e. Y axis, X axis
  - f. Y axis, Z axis
  - g. X axis, Y axis**
  - h. X axis, Z axis
84. A table can be logically connected to another table by defining a \_\_\_\_\_.

# **AL-JUNAID INSTITUTE GROUP**

- a. Hyper link
  - b. Common attribute
  - c. Primary key**
  - d. Super key
85. Data Redundancy means
- a. Security of data
  - b. Duplication of data**
  - c. Management of data
  - d. Recovery of data
86. Which of the following is a type of dependency?
- a. DBMS
  - b. ERD
  - c. DFD
  - d. Referential**
87. In your opinion, why relational database is widely acceptable?
- a. Due to its complexity
  - b. Due to its approach
  - c. Due to its dependencies**
  - d. Due to its strength
88. \_\_\_\_\_ is the first comprehensive complete database design.
- a. Logical database design
  - b. Relational database design
  - c. Conceptual database design**
  - d. Hierarchical database design
89. How data security can be assured in the database?
- a. By hiding data from all
  - b. Through concurrent access of users
  - c. By asking users not to share things
  - d. Implementing encryption algorithms**
90. The attributes whose value is not stored in a database is known as \_\_\_\_\_ attribute.

# **AL-JUNAID INSTITUTE GROUP**

- a. Stored
  - b. Single value
  - c. Multi value
  - d. Derived**
91. Logical database design, like conceptual database design is our \_\_\_\_\_
- a. Data model
  - b. Weak entity set
  - c. Strong entity set
  - d. Database design**
92. If a single instance of “student” is associated with the many instances of the “subjects” entity, then the relationship between them is:
- a. One to one
  - b. One to many**
  - c. Many to many
  - d. Many to one
93. The conceptual database design is drawn through \_\_\_\_\_.
- a. Data model
  - b. ER model**
  - c. Database design
  - d. Entity set
94. A relational database is
- a. One that consist of two or more tables
  - b. A database that is able to process tables, queries, forms, reports and macros
  - c. One that consists of two or more tables that are joined in some way
  - d. The same as a flat file database
95. Database is a computerized representation of any organization’s flow of \_\_\_\_\_ and storage of \_\_\_\_\_.
- a. Data, information
  - b. Details, data

# **AL-JUNAID INSTITUTE GROUP**

- c. Data, details
  - d. Information, data**
96. In which stage of the database development process logical design is created?
- a. Requirement analysis
  - b. Database design
  - c. Physical design
  - d. Implementation**
97. Which of the following is NOT an entity?
- a. Employee
  - b. Hobby
  - c. Student
  - d. Playground**
98. After moving the determined columns from the original table to the new table while converting to 2NF, the determinate becomes the \_\_\_\_\_ of the new table.
- a. Primary key**
  - b. Foreign key
  - c. Secondary key
  - d. Composite key
99. The select operation works horizontally on the table, on the other hand the \_\_\_\_\_ operates on a single table vertically.
- a. Binary operator
  - b. Project operator**
  - c. Join operator
  - d. Select operator
100. Free standing data dictionary is created by
- a. DFD
  - b. Database
  - c. CASE tools**
  - d. Cross reference matrix
101. A patient can be an outdoor or indoor. Analyze this statement and tell what type of

# AL-JUNAID INSTITUTE GROUP

constraint will be applied on patient?

- a) Total completeness
  - b) Partial completeness
  - c) Total disjoint
  - d) Partial disjoint
102. If  $A \rightarrow B$  and  $A \rightarrow C$ , then  $A \rightarrow BC$ . The inference Rule applies is:
- a) Union
  - b) Decomposition
  - c) Augmentation
  - d) Reflexivity
103. Which of the following is the correct abbreviation of ERD?
- a) Entity relationship diagram
  - b) Entity relation diagram
  - c) Entity reality diagram
  - d) Entity real distance
104. Database relation is also represented in a two dimensional structure called
- a) Row
  - b) Table
  - c) Entity
  - d) Column
105. Data about data is metadata
- a) True
  - b) False
106. Which of the following describes the job of a database administrator?
- a) Development, implementation operation of the physical database
  - b) Creation of the system catalog
  - c) Monitoring and controlling database security and authorization: setting up controls to ensure the quality and integrity of data
  - d) All of the above

# AL-JUNAID INSTITUTE GROUP

107. Each table must have a \_\_\_\_\_ key.
- a) **Primary**
  - b) Secondary
  - c) Logical
  - d) Foreign
108. A relational database system is based on the concept(s) of:
- a) A network of data records
  - b) Nested tables
  - c) **Tables, row and columns**
  - d) A tree-like structure of data
109.  $A \rightarrow BC$  and  $A \rightarrow B$ , then  $A \rightarrow C$  follow \_\_\_\_\_ inference rule.
- a) Augmentation
  - b) Transitivity
  - c) **Projectivity**
  - d) Additivity
110. Suppose, we have two entities “shape” and “triangle”, then which of the following is a correct statement?
- a) **Shape is a super type entity and triangle is a sub type entity**
  - b) Shape is a sub type entity and triangle is a super type entity
  - c) Shape and triangle both are super type entities
  - d) Shape and triangle both are sub type entities
111. The entity that is NOT dependent on any other entity for its existence is known as \_\_\_\_\_ entity.
- a) Weak
  - b) **Strong**
  - c) Derived
  - d) Dependent
112. In cross reference matrix attributes are specified on \_\_\_\_\_
- a) **X axis**
  - b) Y axis
  - c) Z axis
  - d) None of the above

# **AL-JUNAID INSTITUTE GROUP**

113. Which of the following clarify the semantics of a relationship?
- a) Entity
  - b) Roles**
  - c) Relationship set
  - d) Participants
114. How many types of cardinality is specified about relationship?
- a) Two
  - b) One
  - c) Three
  - d) Four**
115. Which of the following is NOT a component of a DFD?
- a) Dataflow
  - b) Datastore
  - c) External entities
  - d) Relationship between external entities**
116. Which of the following provides us the facility to add properties of one entity to another entity automatically?
- a) Abstraction
  - b) Encapsulation
  - c) Inheritance**
  - d) Dependencies
117. If there is more than one relationship between two entities and we mentioned role on the relationship link, then it is called:
- a) Unary relationship
  - b) Binary relationships
  - c) Ternary relationships
  - d) Multiple relationships**
118. Which one of the following is not an advantage of DBMS
- a) Data consistency
  - b) Recovery procedure is difficult**
  - c) Faster development of new applications
  - d) Better backup and recovery procedures

# AL-JUNAID INSTITUTE GROUP

119. In many to many relationship a third table is created for the relationship, which is also called as \_\_\_\_\_
- a) Binary relation
  - b) Associative entity type**
  - c) Unary relationship
  - d) Composite attributes
120. When one entity instance of another entity for its existence, then it is called \_\_\_\_\_
- a) Existence dependency**
  - b) Identifier dependency
  - c) Referential dependency
  - d) Acceptance dependency
121. Relational data model is widely accepted due to its mathematically proven foundation and its \_\_\_\_\_
- a) Complexity
  - b) Simplicity**
  - c) Hierarchical structure
  - d) Traditional structure
122. If an entity “motor cycle” is associated with at most one “driver”, then the relationship between these two entities is:
- a) One to one**
  - b) One to many
  - c) Many to many
  - d) Many to one
123. Database application is a program which is used for performing certain operations such as \_\_\_\_\_
- a) Insertion of data
  - b) Extraction of data
  - c) Updating the data
  - d) All of above**

# AL-JUNAID INSTITUTE GROUP

124. Creating the relationships between new tables and their predecessors through the use of foreign keys is general requirements of \_\_\_\_\_
- a) First normal form
  - b) Second normal form**
  - c) Third normal form
  - d) Fourth normal form
125. In \_\_\_\_\_ the domains of attributes of a relation are atomic, that is they consist of single units that cannot be broken down further.
- a) First normal form**
  - b) Second normal form
  - c) Third normal form
  - d) Augmentation
126. Atomicity is a feature of \_\_\_\_\_.
- a) 1NF**
  - b) 2NF
  - c) 3NF
  - d) BCNF
127. The \_\_\_\_\_ can see entire information structure of the database.
- a) BA**
  - b) DBTG
  - c) Database user
  - d) External viewer
128. The attribute whose value is not stored in a database is known as \_\_\_\_\_ attribute.
- a) Stored
  - b) Single value
  - c) Multi value
  - d) Derived**

# **AL-JUNAID INSTITUTE GROUP**

129. \_\_\_\_\_ are the structure defined for placing data in the attributes while designing physical design of database.
- a) Objects
  - b) Variables
  - c) Attributes
  - d) Data types**
130. Suppose a relation has five columns and ten rows, select the correct option to identify cardinality of relation.
- a) Five
  - b) Ten**
  - c) Fifteen
  - d) Twenty
131. Which feature of database provides conversion from inconsistent state of DB to a consistent state ensuring minimum data loss?
- a) User accessible catalog
  - b) Data processing
  - c) Recovery service**
  - d) Authorization service
132. The maximum PL/SQL size of data type “VARCHAR2” in DBMS is \_\_\_\_\_ bytes
- a) 42767
  - b) 52767
  - c) 32767**
  - d) 28767
133. \_\_\_\_\_ is state of database where a “course” cannot be inserted in the table, because this course has not been registered to any “student”.
- a) Insertion anomaly**
  - b) Deletion anomaly
  - c) Updation anomaly
  - d) Redundancy

# AL-JUNAID INSTITUTE GROUP

134. The maximum PL/SQL size of data type “BLOB” in DBMS is \_\_\_\_\_
- a) 4 gigabytes
  - b) 8 gigabytes
  - c) 10 gigabytes
  - d) 12 gigabytes
135. \_\_\_\_\_ states that in a relation no attributes of a primary key (PK) can have a null value.
- a) Referential integrity constraint
  - b) Null constraint
  - c) Domain constraint
  - d) Entity integrity constraint
136. A database state where deletion of the information about the student record deletes the course information as well is called \_\_\_\_\_.
- a) Insertion anomaly
  - b) Updation anomaly
  - c) Redundancy
  - d) Deletion anomaly
137. An entity type is
- a) Defined when the database is actually constructed
  - b) A specific type such as an integer, text, date, logical etc.
  - c) A coherent set of similar objects that we want to store data on (e.g. STUDENTS, COURSE, CAR)
  - d) Defined by the database designer
138. “select” and “project” are the examples of \_\_\_\_\_
- a) Unary operation
  - b) Binary operation
  - c) Ternary operation
  - d) Nary operation
139. External scheme evolves as user needs are \_\_\_\_\_ over the time.
- a) Ends

# **AL-JUNAID INSTITUTE GROUP**

- b) Modified
  - c) Switch**
  - d) Stable
140. An attribute that is the collection of some other attribute(s) is known as \_\_\_\_\_  
attributes.
- a) Multi-valued
  - b) Single-valued
  - c) No-valued
  - d) Composite**
141. Which of the following is correct regarding Dataflow diagram?
- a) Single DFD is required to represent a system
  - b) The dataflow must be bidirectional
  - c) Created at increasing levels of detail**
  - d) Used to represent the relationships among the external entities
142. \_\_\_\_\_ are those attributes that are a combination of two or more than two attributes.
- a) Default value
  - b) Null value
  - c) Primary key
  - d) Composite attributes**
143. Controlling redundancy in a database management system DOES NOT help to
- a) avoid duplication
  - b) avoid unnecessary wastage of storage space
  - c) avoid unauthorized access to data**
  - d) avoid inconsistency among data
144. Which of the following concepts is applicable with respect to 3NF?
- a) Full functional dependency
  - b) Any kind of dependency
  - c) Transitive dependency**

# **AL-JUNAID INSTITUTE GROUP**

- d) Partial functional dependency
145. Identify the operation which is NOT one of the parts of the five basic set operations in relational algebra?
- a) **Join**
  - b) Union
  - c) Cartesian Product
  - d) Set Difference
146. Making a change to the conceptual schema of a database but not affecting the existing external schemas is an example of
- a) Physical data independence.
  - b) Concurrency control.
  - c) **Logical data independence.**
  - d) Functional dependency
147. For a third normal form we concentrate on relations with one \_\_\_\_\_ key, and we eliminate transitive dependencies
- a) **Candidate**
  - b) Composite
  - c) Super key
  - d) Primary key
148. The entity relationship diagram is used to graphically represent the \_\_\_\_\_ database model.
- a) Condensed
  - b) Physical
  - c) Logical
  - d) **Conceptual**
149. The \_\_\_\_\_ determines that the link between two entities is optional or mandatory.
- a) Maximum cardinality
  - b) **Minimum cardinality**
  - c) Maximum degree
  - d) Minimum degree

# **AL-JUNAID INSTITUTE GROUP**

150. Select the most appropriate statement. Which of the following data model is used to design logical database design?

- a) Relational
- b) Hierarchal
- c) Object oriented
- d) Any model can be used

vulmsheep.com



**CS403- Database Management Systems**  
**Solved MCQS**  
**From Midterm Papers**

**April 29,2012**

MC100401285

*Moaaz.pk@gmail.com*

*Mc100401285@gmail.com*

PSMD01

**CS403- Database Management Systems**  
**MIDTERM EXAMINATION - Spring 2010**

**Question No: 1 ( Marks: 1 ) - Please choose one**

Which of the following is NOT a feature of Context DFD?

- ▶ one process (which represents the entire system)
- ▶ all sources/sinks (external entities)
- ▶ data flows linking the process to the sources and sinks (external entities)
- ▶ **sub-processes (which explain and decomposed the major process into small processes) (Page 62)**

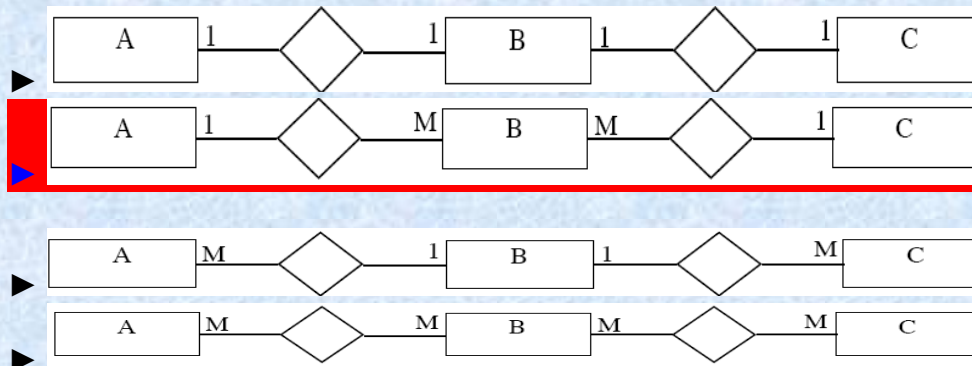
**Question No: 2 ( Marks: 1 ) - Please choose one**

Which of the following is true for the relational model?

- ▶ Degree of a relation is the number of rows in a relation.
- ▶ Null value is a blank or zero value given to an attribute value when its value is inapplicable or its value is unknown.
- ▶ Complex key is a key consisting of more than one attribute.
- ▶ **Constraint is a rule that restricts the values in a database. (Page 18)**

**Question No: 3 ( Marks: 1 ) - Please choose one**

Which one of the following four E-R diagrams is the typical result you obtain when you initially start with an E-R diagram containing just two entities, A and C, in a Many-to-Many relationship, and then introduce an associative entity (B).



[Click here for Detail](#)

**Question No: 4 ( Marks: 1 ) - Please choose one**

Which of the following most certainly implies the need for an entire table to implement?

- ▶ A binary relationship
- ▶ **A ternary relationship** [Click here for Detail](#)
- ▶ A recursive relationship
- ▶ An identifying relationship

**Question No: 5 ( Marks: 1 ) - Please choose one**

Which of the following constraints enforces entity integrity?

- ▶ PRIMARY KEY
- ▶ FOREIGN KEY
- ▶ CHECK
- ▶ **NOT NULL (Page 134)**

**Question No: 6 ( Marks: 1 ) - Please choose one**

Which of the following is not true about relational tables?

- ▶ Column values are of the same kind.
- ▶ Each row is unique.
- ▶ Each column must have a unique name.
- ▶ **The sequence of rows is significant.** [Click here for Detail](#)

**Question No: 7 ( Marks: 1 ) - Please choose one**

In a conceptual model for a university, what type of relationship exists between Grade and Student entities?

- ▶ 1:1
- ▶ **1:M** [Click here for Detail](#)
- ▶ M:M
- ▶ Ternary

**Question No: 8 ( Marks: 1 ) - Please choose one**

Controlling redundancy in a database management system DOES NOT help to

- ▶ avoid duplication (Page 16)
- ▶ avoid unnecessary wastage of storage space
- ▶ **avoid unauthorised access to data** [Click here for detail](#)
- ▶ avoid inconsistency among data

**Question No: 9 ( Marks: 1 ) - Please choose one**

Which of the following is INCORRECT with respect to file systems?

- ▶ At the physical level, pointer or hashed address scheme may be employed to provide a certain degree of data independence at the user level.
- ▶ **A logical record is concerned with efficient storage of information in the secondary storage devices.**
- ▶ Some physical organisations use pointers to record blocks to locate records on disk.
- ▶ The efficiency of a file system depends on how efficiently operations such as retrieve, insert, update, delete may be performed on the information stored in the file.

**Question No: 10 ( Marks: 1 ) - Please choose one**

Which of the following functions are NOT performed by a database administrator?

- ▶ Planning, designing and implementing database systems
- ▶ Establishing standards and procedures for database systems
- ▶ **Communicating with database users (Page 26)**
- ▶ Allocation of storage locations and data structures

**Question No: 11 ( Marks: 1 ) - Please choose one**

Select the correct statement about the ANSI/SPARC architecture.

- ▶ **The conceptual level is a level of indication between the internal level and the external level. (Page 33)**
- ▶ The internal level in a database system will definitely be relational.
- ▶ Any given database has many conceptual schemas and one physical schema, but it has only one external schemas.
- ▶ The external level is not concerned with individual user perceptions, while the conceptual level is concerned with a community user perception.

**Question No: 12 ( Marks: 1 ) - Please choose one**

Which of the following is a correct way to implement one-to-many relationship while designing tables?

- ▶ **by splitting the data into two tables with primary key and foreign key relationships.**  
[Click here for Detail](#)
- ▶ using a junction table with the keys from both the tables forming the composite primary key of the junction table.
- ▶ by splitting each table into three
- ▶ as a single table and rarely as two tables with primary and foreign key relationships.

**Question No: 13 ( Marks: 1 ) - Please choose one**

Which of the following is not a benefit of normalization?

- ▶ Minimize insertion anomalies
- ▶ Minimize deletion anomalies
- ▶ Minimize updation anomalies
- ▶ **Maximize redundancy (Page 162)**

**Question No: 14 ( Marks: 1 ) - Please choose one**

Consider the following relation R and its sample data. (Consider that these are the only tuples for the given relation)

EmpNo	DeptNo	ProjNo
1001	01	12
1001	01	13
1002	01	12
1003	01	14

Which of the following statements is NOT correct?

- ▶ The functional dependency (EmpNo, DeptNo) -> ProjNo holds over R.
- ▶ The functional dependency EmpNo -> DeptNo holds over R.
- ▶ The functional dependency ProjNo -> DeptNo holds over R.
- ▶ **The functional dependency (EmpNo, ProjNo) -> DeptNo holds over R.**

**Question No: 15 ( Marks: 1 ) - Please choose one**

The Entity Relation Model models

- ▶ Entities, Relationships and Processes
- ▶ **Entities and Relationships (Page 71)**
- ▶ Relationships
- ▶ Entities

**Question No: 16 ( Marks: 1 ) - Please choose one**

As part of database naming conventions, attribute names should use suffixes such as ID, NUMBER or CODE for the \_\_\_\_\_.

- ▶ **primary key**
- ▶ foreign key
- ▶ index
- ▶ determinant

**MIDTERM EXAMINATION**

**Spring 2010**

**CS403- Database Management Systems (Session - 4)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

User rights information is stored in

- ▶ Physical database
- ▶ **Catalog (Page 46)**
- ▶ Logical database
- ▶ Buffer

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 2 ( Marks: 1 ) - Please choose one**

Making a change to the conceptual schema of a database but not affecting the existing external schemas is an example of

- ▶ Physical data independence.
- ▶ Concurrency control.
- ▶ **Logical data independence.** [Click here for detail](#)
- ▶ Functional dependency

**Question No: 3 ( Marks: 1 ) - Please choose one**

Which of the following is NOT a feature of Context DFD?

- ▶ one process (which represents the entire system)
- ▶ all sources/sinks (external entities)
- ▶ data flows linking the process to the sources and sinks (external entities)
- ▶ **sub-processes (which explain and decomposed the major process into small processes) (Page 62)**

rep

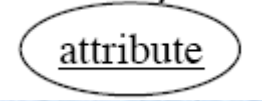
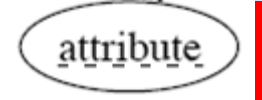
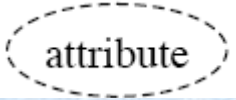

**Question No: 4 ( Marks: 1 ) - Please choose one**

A relation (from the relational database model) consists of a set of tuples, which implies that

- ▶ **all tuples in a relation must be distinct.** (Page 129)
- ▶ relational model supports multi-valued attributes whose values can be represented in sets.
- ▶ for any two tuples, the values associated with all of their attributes may be the same.
- ▶ all tuples in a particular relation may have different attributes.

**Question No: 5 ( Marks: 1 ) - Please choose one**

Choose the symbol that corresponds to a discriminator attributes.

- ▶ 
- ▶  [Click here for detail](#)
- ▶ 
- ▶ 

**Question No: 6 ( Marks: 1 ) - Please choose one**

Identify the constraint that limits the values that can be placed in a column.

- ▶ NOT NULL
- ▶ **CHECK** [Click here for Detail](#)
- ▶ FOREIGN KEY
- ▶ UNIQUE

**Question No: 7 ( Marks: 1 ) - Please choose one**

Given are the relations of student and Instructor

<i>Student</i>		<i>Instructor</i>	
First Name	Last Name	Fname	Lname
Saman	Perera	Ajith	Gamage
Romesh	Dias	Sujith	Hewage
Jeeva	Silva	Saman	Perera
Nadee	Alwis	Kasun	Peiris
Kumari	Costa	Romesh	Dias
Geetha	Zoysa		
Prasad	Fernando		

Consider the following table obtained using Student and Instructor relations.

Fname	Lname
Ajith	Gamage
Sujith	Hewage
Kasun	Peiris

Which relational algebra operation could have been applied on the pair of relations Student and Instructor to obtain the above data?

- ▶ **Instructor – Student**
- ▶ Student  $\cap$  Instructor
- ▶ Instructor  $\div$  Student
- ▶ Student – Instructor

**Question No: 8 ( Marks: 1 ) - Please choose one**

Identify the correct statement with respect to normalization.

- ▶ Normalization is a formal technique that can be used only at the starting phase of the database design.
- ▶ **Normalization can be used as a top-down standalone database design technique.**

[Click here for detail](#)

- ▶ The process of normalization through decomposition must achieve the lossless join property at any cost whereas the dependency reservation property is sometimes sacrificed.
- ▶ The process of normalization through decomposition must achieve the dependency reservation property at any cost whereas the lossless join property is sometimes sacrificed.

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 9 ( Marks: 1 ) - Please choose one**

Consider the relation Interview(CandidateNo, InterviewDate, InterviewTime, StaffNo, RoomNo) and the following functional dependencies.

FD1 : CandidateNo, InterviewDate -> InterviewTime, StaffNo, RoomNo

FD2 : RoomNo, InterviewDate, InterviewTime -> StaffNo, CandidateNo

FD3 : StaffNo, InterviewDate -> RoomNo

Which of the following is correct?

- ▶ **The relation Interview is in 3NF** [Click here for detail](#)
- ▶ The relation Interview is in BCNF.
- ▶ The FD3 violates 3NF.
- ▶ The FD2 violates 2NF.

**Question No: 10 ( Marks: 1 ) - Please choose one**

Identify the INCORRECT statement among the given.

- ▶ An entity may be an object with a physical existence like a car, a house or an Employee.
- ▶ **One cannot consider something which has conceptual existence like a course in a degree program as an entity.** (Page 71)
- ▶ Age can be considered as a single value attribute of a person.
- ▶ An entity type describes the schema or intension for a set of entities which share the same structure.

**Question No: 11 ( Marks: 1 ) - Please choose one**

*Structural constraints* of a relationship type refer to

- ▶ identifying the owner entity type relevant to a given entity type
- ▶ **whether the existence of an entity depends on it being related to another entity via the relationship type.** [Click here for detail](#)
- ▶ the role that a participating entity from the entity type plays in each relationship instance.
- ▶ the constraints applicable in granting access to tables, columns and views in a database schema.

**Question No: 12 ( Marks: 1 ) - Please choose one**

A collection of concepts that can be used to describe the structure of a database

- ▶ Database
- ▶ DBMS
- ▶ **Data model** [Click here for Detail](#) (Page 68)
- ▶ Data

**Question No: 13 ( Marks: 1 ) - Please choose one**

An entity can be logically connected to another by defining a \_\_\_\_\_.

- ▶ hyperlink
- ▶ **common attribute** [Click here for detail](#)
- ▶ primary key
- ▶ superkey

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 14 ( Marks: 1 ) - Please choose one**

The \_\_\_\_ constraint specifies whether each entity supertype occurrence must also be a member of at least one subtype.

- ▶ specialization
- ▶ uniqueness
- ▶ inheritance
- ▶ **completeness**      [Click here for Detail](#)

**Question No: 15 ( Marks: 1 ) - Please choose one**

Database management systems, operating systems, applications and utilities are all examples of \_\_\_\_.

- ▶ hardware
- ▶ **software**      [Click here for detail](#)
- ▶ computer infrastructure
- ▶ input and output

**Question No: 16 ( Marks: 1 ) - Please choose one**

Which of the following concepts is applicable with respect to 2NF?

- ▶ **Full functional dependency**      [Click here for detail](#)
- ▶ Any kind of dependency
- ▶ Transitive dependency
- ▶ Non-transitive dependency

**MIDTERM EXAMINATION Spring 2010**  
**CS403- Database Management Systems (Session - 4)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

**A database system allows the following EXCEPT**

- ▶ management and control of data towards an efficient working of an organisation.
- ▶ more critical functions in organisations to be computerised and the need to keep a large volume of data available in an up to the minute current state increased.
- ▶ **any user to access all its data.**
- ▶ integration of data across multiple applications into a single application.

**Question No: 2 ( Marks: 1 ) - Please choose one**

**User rights information is stored in**

- ▶ Physical database
- ▶ **Catalog (Page 46) rep**
- ▶ Logical database
- ▶ Buffer

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 3 ( Marks: 1 ) - Please choose one**  
**The ER- data model is an example of:**

- ▶ Physical database
- ▶ Logical database
- ▶ Relational database
- ▶ **Conceptual database**      [Click here for Detail](#)

**Question No: 4 ( Marks: 1 ) - Please choose one**  
**Which of the following is true about NOT NULL constraint?**

- ▶ enforce domain integrity
  - ▶ limit the values that can be placed in a column.
  - ▶ **prevents any actions that would destroy links between tables with the corresponding data values**
- [Click here for detail](#)
- ▶ enforces the uniqueness of the values in a set of columns

**Question No: 5 ( Marks: 1 ) - Please choose one**  
**Consider the relation Interview(CandidateNo, InterviewDate, InterviewTime, StaffNo, RoomNo) and the following functional dependencies.**

**FD1 : CandidateNo, InterviewDate -> InterviewTime, StaffNo, RoomNo**

**FD2 : RoomNo, InterviewDate, InterviewTime -> StaffNo, CandidateNo**

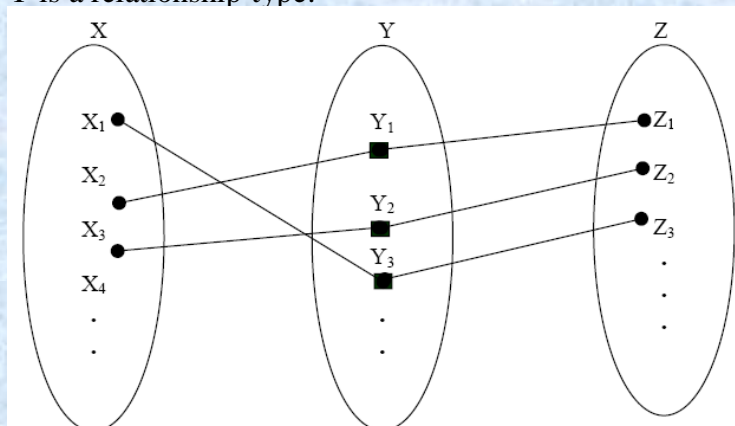
**FD3 : StaffNo, InterviewDate -> RoomNo**

**Which of the following is correct?**

- ▶ The relation Interview is in BCNF.
- ▶ The FD3 violates 3NF.
- ▶ The FD3 violates BCNF.
- ▶ The FD2 violates 2NF.

**Question No: 6 ( Marks: 1 ) - Please choose one**

**Consider the following diagram depicting a kind of a relationship type where X and Z are entities and Y is a relationship type:**



Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@gmail.com](mailto:mc100401285@gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

Select the correct statement among the following on the above diagram.

- ▶ **The relationship type Y is of cardinality ratio 1 : N.** [Click here for Detail](#)
- ▶ The diagram depicts existence dependencies.
- ▶ The participation of X in the Y relationship type is total.
- ▶ The participation of Z in the Y relationship type is partial.

**Question No: 7 ( Marks: 1 ) - Please choose one**

**Select the correct statement among the following.**

- ▶ Role names are not technically necessary in relationship types when all the participating entity types are distinct.
- ▶ **When different entity types participate only once in a single relationship type it is called a recursive relationship. (Page 87)**
- ▶ Cardinality ratios for binary relationship are displayed on Entity Relationship Diagrams by using a diamond shape notation.
- ▶ Partial participation which is also called existence dependency is displayed as a double line connecting the participating entity type to the relationship.

**Question No: 8 ( Marks: 1 ) - Please choose one**

**Which of the following is true about relational schema?**

- ▶ The sequence of columns is significant
- ▶ The sequence of rows is significant.
- ▶ Contains only derived attributes.
- ▶ **Values are atomic. (Page 127)**

**Question No: 9 ( Marks: 1 ) - Please choose one**

**Consider the given relations Student and Instructor as given below. Please note that Fname and Lname also denote the First Name and Last Name respectively.**

<i>Student</i>		<i>Instructor</i>	
First Name	Last Name	Fname	Lname
Saman	Perera	Ajith	Gamage
Romesh	Dias	Sujith	Hewage
Jeeva	Silva	Saman	Perera
Nadee	Alwis	Kasun	Peiris
Kumari	Costa	Romesh	Dias
Geetha	Zoysa		
Prasad	Fernando		

**Which of the following statements is correct with respect to the two relations given above?**

- ▶ The two relations are not union-compatible since their attribute names differ.
- ▶ **The set operations such as CARTESIAN PRODUCT and DIVISION can be applied on these two relations.**
- ▶ To find out those students who work as instructors, it is necessary to perform the operation  $\text{Student} \cap \text{Instructor}$ .
- ▶ To find out the students who are not instructors, it is necessary to perform the operation  $\text{Student} \div \text{Instructor}$ .

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 11 ( Marks: 1 ) - Please choose one** Consider the following relation R and its sample data. (Consider that these are the only tuples for the given relation)

EmpNo	DeptNo	ProjNo
1001	01	12
1001	01	13
1002	01	12
1003	01	14

Which of the following statements is NOT correct?

- ▶ The functional dependency ProjNo  $\rightarrow$  DeptNo holds over R.
- ▶ **The functional dependency (EmpNo, ProjNo)  $\rightarrow$  DeptNo holds over R.**
- ▶ The functional dependency DeptNo  $\rightarrow$  ProjNo holds over R.
- ▶ The functional dependency EmpNo  $\rightarrow$  DeptNo holds over R.

**Question No: 12 ( Marks: 1 ) - Please choose one**

A collection of related data is

- ▶ Logical model
- ▶ **Database (Page 10)**
- ▶ Data
- ▶ Relational model

**Question No: 13 ( Marks: 1 ) - Please choose one**

A weak entity type

- ▶ must have total participation in an identifying relationship
- ▶ does not have a key attribute(s)
- ▶ **both (a) and (b) [Click here for detail](#)**
- ▶ none of the above

**Question No: 14 ( Marks: 1 ) - Please choose one**

A description on a particular collection of data using the given data model

- ▶ Database
- ▶ **Schema (Page 18)**
- ▶ None of the above.
- ▶ Relation

**Question No: 15 ( Marks: 1 ) - Please choose one**

If K is a foreign key in relation R1, then

- ▶ every tuple of R1 has a distinct value for K.
- ▶ K cannot have a null value for tuples in R1.
- ▶ **K is a key for some other relation. [Click here for detail](#)**
- ▶ K is a primary key for R1.

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 16 (Marks: 1) - Please choose one**

Consider the following statements.

- A. An entity integrity constraint states that no primary key value can be null.
- B. A referential integrity constraint is specified between two relations.
- C. A foreign key cannot be used to refer to its own relation.

Identify which of the above statements is/are correct.

▶ **Only A (Page 134)**

- ▶ Only B
- ▶ B and C
- ▶ A and B

**MIDTERM EXAMINATION 2010**  
**CS403- Database Management Systems**

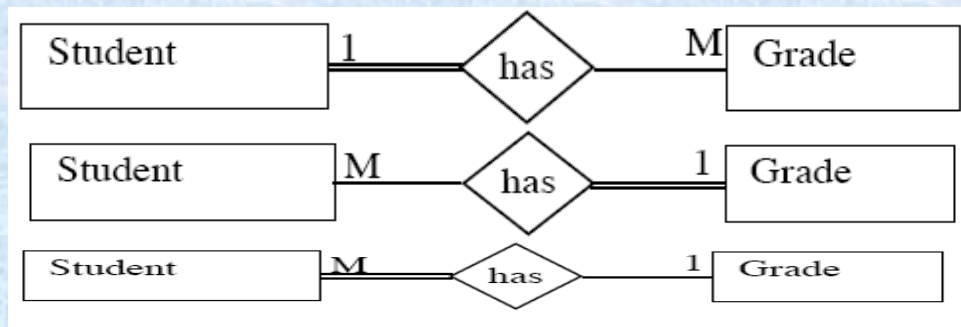
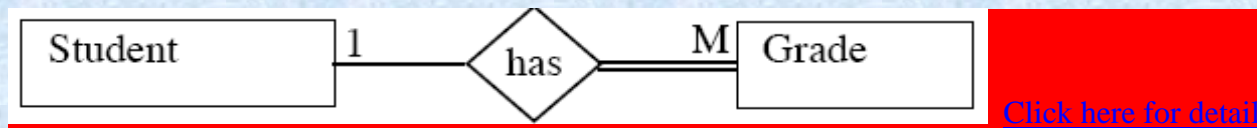
**Question No: 1 (Marks: 1) - Please choose one**

Which of the following constraints enforces entity integrity?

- ▶ PRIMARY KEY
- ▶ FOREIGN KEY
- ▶ CHECK
- ▶ **NOT NULL (Page 134) rep**

**Question No: 2 (Marks: 1) - Please choose one**

Which one of the following E-R diagrams most correctly represents the relationship between Student and Grade entities?



**Question No: 3 ( Marks: 1 ) - Please choose one**

Which of the following constraints enforces entity integrity?

- ▶ PRIMARY KEY
- ▶ FOREIGN KEY
- ▶ CHECK
- ▶ NOT NULL (Page 134) rep

**Question No: 4 ( Marks: 1 ) - Please choose one**

Which of the following enforces a relation into 1st normal form?

- ▶ **The domain of attribute must include only atomic values. (Page 167)**
- ▶ Every non-key attribute is fully functionally dependent on primary key
- ▶ non-key attribute is non-transitively dependent on primary key.
- ▶ Every non-key attribute is partially dependent on super key

**Question No:5 ( Marks: 1 ) - Please choose one**

Consider the following set of functional dependencies (FDs) on the following relational schema.

Emp\_No -> {Ename, Bdate, Address, Dept\_No}

Dept\_No -> {Dname, Mgr\_No}

The additional FD which can be inferred from the above set of FDs is

- ▶ Emp\_No -> {Dname, Mgr\_No}
- ▶ Emp\_Name -> Dept\_No .
- ▶ Emp\_Name -> Dept\_Name .
- ▶ Emp\_Name, Dept\_No -> Mgr\_No .

**Question No: 6 ( Marks: 1 ) - Please choose one**

Which of the following is a feature of PRIMARY KEY constraint?

- ▶ **unique identifier for a row within a database table. [Click here for Detail](#)**
- ▶ allow any actions that would destroy links between tables
- ▶ limit the values that can be placed in a column.
- ▶ enforces that the column will only accept null values.

**Question No: 7 ( Marks: 1 ) - Please choose one**

Structural constraints of a relationship type refer to

- ▶ identifying the owner entity type relevant to a given entity type
- ▶ **whether the existence of an entity depends on it being related to another entity via the relationship type. [Click here for detail](#)**
- ▶ the role that a participating entity from the entity type plays in each relationship instance.
- ▶ the constraints applicable in granting access to tables, columns and views in a database schema.

**Question No: 8 ( Marks: 1 ) - Please choose one**

Which of the following is true about relational schema?

- ▶ The sequence of columns is significant
- ▶ The sequence of rows is significant.
- ▶ Contains only derived attributes.
- ▶ **Values are atomic. (Page 127) rep**

**Question No: 9 ( Marks: 1 ) - Please choose one**

A \_\_\_\_ relationship exists when an association is maintained within a single entity.

- ▶ **unary (Page 144)**
- ▶ ternary
- ▶ binary
- ▶ weak

**Question No: 10 ( Marks: 1 ) - Please choose one**

Which of the following is a correct way to implement one-to-many relationship while designing tables?

- ▶ **by splitting the data into two tables with primary key and foreign key relationships.**  
[Click here for Detail](#)
- ▶ using a junction table with the keys from both the tables forming the composite primary key of the junction table.
- ▶ by splitting each table into three
- ▶ as a single table and rarely as two tables with primary and foreign key relationships.

**Question No: 11 ( Marks: 1 ) - Please choose one**

Identify the correct statement.

- ▶ Entity integrity constraints specify that primary key values can be composite.
- ▶ **Entity integrity constraints are specified on individual relations. [Click here for detail](#)**
- ▶ Entity integrity constraints are specified between weak entities.
- ▶ When entity integrity rules are enforced, a tuple in one relation that refers to another relation must refer to an existing tuple.

**Question No: 12 ( Marks: 1 ) - Please choose one**

A software package designed to store and manage databases

- ▶ Database
- ▶ **DBMS (Page 18)**
- ▶ Data model
- ▶ Data

**Question No: 13 ( Marks: 1 ) - Please choose one**

Who is responsible for authorizing access to the database, for coordinating and monitoring its use?  
Select correct option:

- ▶ Database Designer
- ▶ **Database Administrator (Page 26)**
- ▶ End User
- ▶ Application Programmer

**Question No: 14 ( Marks: 1 ) - Please choose one**

Consider the following relation R and its sample data. (Consider that these are the only tuples for the given relation)

EmpNo	DeptNo	ProjNo
1001	01	12
1001	01	13
1002	01	12
1003	01	14

Which of the following statements is NOT correct?

- ▶ The functional dependency (EmpNo, DeptNo) -> ProjNo holds over R.
- ▶ The functional dependency EmpNo -> DeptNo holds over R.
- ▶ The functional dependency ProjNo -> DeptNo holds over R.
- ▶ **The functional dependency (EmpNo, ProjNo) -> DeptNo holds over R.**

**Question No: 15 ( Marks: 1 ) - Please choose one**

Which of the following statements is NOT correct?

- ▶ The functional dependency (EmpNo, DeptNo) -> ProjNo holds over R.
- ▶ The functional dependency EmpNo -> DeptNo holds over R.
- ▶ The functional dependency ProjNo -> DeptNo holds over R.
- ▶ **The functional dependency (EmpNo, ProjNo) -> DeptNo holds over R.**

**Question No: 16 ( Marks: 1 ) - Please choose one**

Which feature of database provides conversion from inconsistent state of DB to a consistent state ensuring minimum data loss?

Select correct option:

- ▶ User accessible catalog
- ▶ Data processing
- ▶ **Recovery service (Page 47)**
- ▶ Authorization service

**MIDTERM EXAMINATION**  
**Spring 2010**  
**CS403- Database Management Systems (Session - 6)**

**Question No: 1 ( Marks: 1 )**

**Consider the following statements.**

- A. Conceptual schema which is the result of conceptual design is a logical description of all data elements and their relationships.
- B. Internal level of the database architecture consists of the physical view of the database.
- C. External level of the database architecture provides the user view of the database.

With respect to the ANSI/SPARC three level database architecture, which of the above is/are correct?

- ▶ Only A.
- ▶ Only C.
- ▶ Only A and B.
- ▶ **Only B and C. (Page 40)**

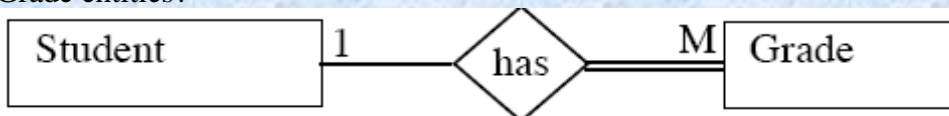
**Question No: 2 ( Marks: 1 )**

The ER- data model is an example of:

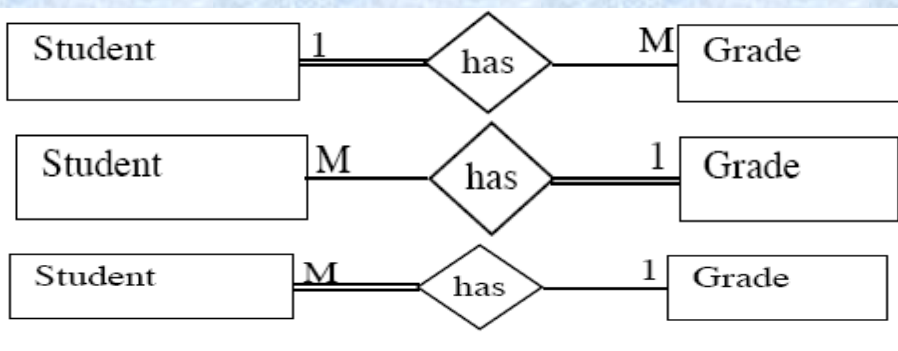
- ▶ Physical database
- ▶ Logical database
- ▶ Relational database
- ▶ **Conceptual database** [Click here for Detail](#) rep

**Question No: 3 ( Marks: 1 )**

Which one of the following E-R diagrams most correctly represents the relationship between Student and Grade entities?



[Click here for detail](#) rep



**Question No: 4 ( Marks: 1 )**

Which of the following constraints enforces referential integrity?

- ▶ **FOREIGN KEY (Page 134)**
- ▶ CHECK
- ▶ PRIMARY KEY
- ▶ UNIQUE

**Question No: 5 ( Marks: 1 )**

Given are the relations of student and Instructor

<i>Student</i>		<i>Instructor</i>	
First Name	Last Name	Fname	Lname
Saman	Perera	Ajith	Gamage
Romesh	Dias	Sujith	Hewage
Jeeva	Silva	Saman	Perera
Nadee	Alwis	Kasun	Peiris
Kumari	Costa	Romesh	Dias
Geetha	Zoysa		
Prasad	Fernando		

Consider the following table obtained using Student and Instructor relations.

Fname	Lname
Ajith	Gamage
Sujith	Hewage
Kasun	Peiris

Which relational algebra operation could have been applied on the pair of relations Student and Instructor to obtain the above data?

- ▶ **Instructor – Student**
- ▶ Student  $\cap$  Instructor
- ▶ Instructor  $\div$  Student
- ▶ Student – Instructor

**Question No: 6 ( Marks: 1 )**

Consider the relation Interview (CandidateNo, InterviewDate, InterviewTime, StaffNo, RoomNo) and the following functional dependencies.

FD1 : CandidateNo, InterviewDate -> InterviewTime, StaffNo, RoomNo

FD2 : RoomNo, InterviewDate, InterviewTime -> StaffNo, CandidateNo

FD3 : StaffNo, InterviewDate -> RoomNo

Which of the following is correct?

- ▶ **The relation Interview is in 3NF** [Click here for detail](#)
- ▶ The relation Interview is in BCNF.
- ▶ The FD3 violates 3NF.
- ▶ The FD2 violates 2NF.

**Question No: 7 ( Marks: 1 )**

Which of the following is INCORRECT statement concerning the database design process?

- ▶ During requirements collection and analysis phase, one can gather the data requirements of database users.
- ▶ By referring to a high level data model, it is possible to understand the data requirements of the users, entity types, relationships and constraints.
- ▶ Transformation of the high level data model into the implementation data model is called logical design or data model mapping.
- ▶ **During the logical design phase of internal storage structures, access paths and file organization for the database files are specified. (Page 53)**

**Question No: 8 ( Marks: 1 )**

Consider the following diagram depicting a kind of a relationship type where X and Z are entities and Y is a relationship type:

Select the correct statement among the following on the above diagram.

- ▶ **The relationship type Y is of cardinality ratio 1 : N. [Click here for Detail](#) rep**
- ▶ The diagram depicts existence dependencies.
- ▶ The participation of X in the Y relationship type is total.
- ▶ The participation of Z in the Y relationship type is partial.

**Question No: 9 ( Marks: 1 )**

Identify the correct statement.

- ▶ Entity integrity constraints specify that primary key values can be composite.
- ▶ **Entity integrity constraints are specified on individual relations. [Click here for detail](#) rep**
- ▶ Entity integrity constraints are specified between weak entities.
- ▶ When entity integrity rules are enforced, a tuple in one relation that refers to another relation must refer to an existing tuple.

**Question No: 10 ( Marks: 1 )**

Identify the correct statement.

- ▶ Referential integrity constraints check whether the primary key values are unique.
- ▶ Referential integrity constraints check whether an attribute value lies in the given range.
- ▶ Referential integrity constraints are specified between entities having recursive relationships.
- ▶ **When Referential integrity rules are enforced, a tuple in one relation that refers to another relation must refer to an existing tuple. (Page 134)**

**Question No: 11 ( Marks: 1 )**

Identify the correct way to implement one-to-one relationship in tables?

- ▶ by splitting the data into two tables with primary key and foreign key relationships.
- ▶ **as a single table and rarely as two tables with primary and foreign key relationships.**

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

[Click here for Detail](#)

- ▶ using a junction table with the keys from both the tables forming the composite primary key of the junction table.
- ▶ by creating two separate tables

**Question No: 12 ( Marks: 1 )**

A collection of related data is

- ▶ Logical model
- ▶ **Database (Page 10) rep**
- ▶ Data
- ▶ Relational model

**Question No: 13 ( Marks: 1 )**

A collection of concepts that can be used to describe the structure of a database

- ▶ Database
- ▶ DBMS
- ▶ **Data model** [Click here for Detail](#) (Page 68)
- ▶ Data

**Question No: 14 ( Marks: 1 )**

A superkey that does not contain a subset of attributes that is itself a superkey is called a \_\_\_\_\_.

- ▶ **candidate key (Page 82)**
- ▶ primary key
- ▶ superkey
- ▶ secondary key

**Question No: 15 ( Marks: 1 )**

As part of database naming conventions, attribute names should use suffixes such as ID, NUMBER or CODE for the \_\_\_\_\_.

- ▶ **primary key**
- ▶ foreign key
- ▶ index
- ▶ determinant

**Question No: 16 ( Marks: 1 )**

Which of the following concepts is applicable with respect to 2NF?

- ▶ **Full functional dependency** [Click here for detail](#)
- ▶ Any kind of dependency
- ▶ Transitive dependency
- ▶ Non-transitive dependency

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**MIDTERM EXAMINATION**  
**Spring 2010**  
**CS403- Database Management Systems (Session - 1)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

**Which of the following functions are NOT performed by a database administrator?**

- ▶ Planning, designing and implementing database systems
- ▶ Establishing standards and procedures for database systems
- ▶ **Communicating with database users (Page 26) rep**
- ▶ Allocation of storage locations and data structures

**Question No: 2 ( Marks: 1 ) - Please choose one**

**Identify the factor which enforces a relation in 3NF?**

- ▶ Every non-key attribute is fully functionally dependent on primary key
- ▶ Every non-key attribute is partially dependent on super key
- ▶ **The domain of attribute must include only atomic values. (Page 167)**
- ▶ Every non-key attribute is non-transitively dependent on primary key.

**Question No: 3 ( Marks: 1 ) - Please choose one**

**Consider two sets A and B. A contains 2 elements and B contains 3. How many elements do their cartesian product contains?**

- ▶ **6 (Page 129)**
- ▶ 9
- ▶ 5
- ▶ 4

**Question No: 4 ( Marks: 1 ) - Please choose one**

**Identify the operation which is NOT one of the parts of the five basic set operations in relational algebra?**

- ▶ **Join [Click here for Detail](#)**
- ▶ Union
- ▶ Cartesian Product
- ▶ Set Difference

**Question No: 5 ( Marks: 1 ) - Please choose one**

**Consider the following statements.**

- A. An entity integrity constraint states that no primary key value can be null.
- B. A referential integrity constraint is specified between two relations.
- C. A foreign key cannot be used to refer to its own relation.

**Muhammad Moaaz Siddiq MCS (3rd)**

**[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)**

**Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari**

Identify which of the above statements is/are correct.

- ▶ Only A (Page 134) rep
- ▶ Only B
- ▶ Only B and C
- ▶ Only A and B

Question No: 6 ( Marks: 1 ) - Please choose one  
User rights information is stored in

- ▶ Physical database
- ▶ **Catalog (Page 46) rep**
- ▶ Logical database
- ▶ Buffer

Question No: 7 ( Marks: 1 ) - Please choose one  
Given are the relations of student and Instructor

<i>Student</i>		<i>Instructor</i>	
First Name	Last Name	Fname	Lname
Saman	Perera	Ajith	Gamage
Romesh	Dias	Sujith	Hewage
Jeeva	Silva	Saman	Perera
Nadee	Alwis	Kasun	Peiris
Kumari	Costa	Romesh	Dias
Geetha	Zoysa		
Prasad	Fernando		

Consider the following table obtained using Student and Instructor relations.

Fname	Lname
Ajith	Gamage
Sujith	Hewage
Kasun	Peiris

Which relational algebra operation could have been applied on the pair of relations Student and Instructor to obtain the above data?

- ▶ **Instructor – Student**
- ▶ Student  $\cap$  Instructor
- ▶ Instructor  $\div$  Student
- ▶ Student – Instructor

**Question No: 8 ( Marks: 1 ) - Please choose one**

**Which one is the corrected way to implement M to M relation which designing data base**

- ▶ using junction table with keys
- ▶ By splitting data into tables with PK and FK
- ▶ As a single table rarely as to tables with PK and FK
- ▶ By creating three tables and linking them through PK and FK

**Question No: 9 ( Marks: 1 ) - Please choose one**

**Which enforce the a relation into 2NF User rights information is stored in**

- ▶ Physical database
- ▶ **Catalog (Page 46) rep**
- ▶ Logical database
- ▶ Buffer

**Question No: 10 ( Marks: 1 ) - Please choose one**

**Which of the following concepts is applicable with respect to 3NF?**

- ▶ Full functional dependency
- ▶ Any kind of dependency
- ▶ **Transitive dependency (Page 172)**
- ▶ Partial functional dependency

**Question No: 11 ( Marks: 1 ) - Please choose one**

**Description on particular collection of data using data model**

- ▶ **Schema [Click here for Detail](#) (Page 19)**
- ▶ Relation
- ▶ Data base
- ▶ None of the above

**Question No: 12 ( Marks: 1 ) - Please choose one**

**Which one is true regarding relation?**

- ▶ Every attribute value Non atomic
- ▶ Attribute in table may not have unique name
- ▶ table order of the column is relevant
- ▶ **the order of the rows is irrelevant [Click here for detail](#)**

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**MIDTERM EXAMINATION**  
**Fall 2010**  
**CS403- Database Management Systems**

**Question No: 1 ( Marks: 1 ) - Please choose one**  
**Which of the following is not a benefit of normalization?**

- ▶ Minimize insertion anomalies
- ▶ Minimize deletion anomalies
- ▶ Minimize updation anomalies
- ▶ **Maximize redundancy (Page 162) rep**

**Question No: 2 ( Marks: 1 ) - Please choose one**  
**Controlling redundancy in a database management system DOES NOT help to**

- ▶ **avoid duplication (Page 16) rep**
- ▶ avoid unnecessary wastage of storage space
- ▶ **avoid unauthorized access to data** [Click here for detail](#)
- ▶ avoid inconsistency among data

**Question No: 3 ( Marks: 1 ) - Please choose one**  
**In a conceptual database model, which of the following most likely represents a valid identifier for a class grades?**

- ▶ StudentID
- ▶ StudentID, CourseID
- ▶ StudentID, CourseID, InstructorID
- ▶ **StudentID, CourseSectionID** [Click here for Detail](#)

**Question No: 4 ( Marks: 1 ) - Please choose one**  
**Identify the operation which is NOT one of the parts of the five basic set operations in relational algebra?**

- ▶ **Join** [Click here for Detail](#) rep
- ▶ Union
- ▶ Cartesian Product
- ▶ Set Difference

**Question No: 5 ( Marks: 1 ) - Please choose one**  
**Making a change to the conceptual schema of a database but not affecting the existing external schemas is an example of**

- ▶ Physical data independence.
- ▶ Concurrency control.
- ▶ **Logical data independence.** [Click here for detail](#)
- ▶ Functional dependency

**Question No: 6 ( Marks: 1 ) - Please choose one**

**Select the correct statement among the following on proper naming of schema constructs:**

- ▶ Entity type name applies to all the entities belonging to that entity type and therefore a plural name is selected for entity type.
- ▶ **In the narrative description of the database requirements, verbs tend to indicate the names of relationship types.** [Click here for Detail](#)
- ▶ The nouns arising from a database requirement description can be considered as names of attributes.
- ▶ Additional nouns which are appearing in the narrative description of the database requirements represent the weak entity type names.

**Question No: 7 ( Marks: 1 ) - Please choose one**

**Identify the constraint that limits the values that can be placed in a column.**

- ▶ Not null
- ▶ **Check** [Click here for Detail](#) **rep**
- ▶ Foreign Key
- ▶ Unique

**Question No: 8 ( Marks: 1 ) - Please choose one**

**Identify the INCORRECT statement among the given.**

- ▶ An entity may be an object with a physical existence like a car, a house or an Employee.
- ▶ **One cannot consider something which has conceptual existence like a course in a degree program as an entity.** (Page 71) **rep**
- ▶ Age can be considered as a single value attribute of a person.
- ▶ An entity type describes the schema or intension for a set of entities which share the same structure.

**Question No: 9 ( Marks: 1 ) - Please choose one**

**Select the correct statement among the following.**

- ▶ Role names are not technically necessary in relationship types when all the participating entity types are distinct.
- ▶ **When different entity types participate only once in a single relationship type it is called a recursive relationship.** (Page 87) **rep**
- ▶ Cardinality ratios for binary relationship are displayed on Entity Relationship Diagrams by using a diamond shape notation .
- ▶ Partial participation which is also called existence dependency is displayed as a double line connecting the participating entity type to the relationship .

**Question No: 10 ( Marks: 1 ) - Please choose one**

**If W, X, Y and Z are attributes of a relation, which of the following inference rules for functional dependencies is correct?**

- ▶ If  $(X, Z) \rightarrow Y$  then  $X \rightarrow Y$  and  $Z \rightarrow Y$ .
- ▶ **If  $X \rightarrow Y$  and  $X \rightarrow Z$  then  $X \rightarrow (Y, Z)$ .**
- ▶ If  $XY$  then  $Y \rightarrow X$
- ▶ If  $X \rightarrow Y$  then  $(X, Z) \rightarrow (Y, W)$ .

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 11 ( Marks: 1 ) - Please choose one**  
**Which of the following is not a benefit of normalization?**

- ▶ Minimize insertion anomalies
- ▶ Minimize deletion anomalies
- ▶ Minimize updation anomalies
- ▶ **Maximize redundancy** [\(Page 162\) rep](#)

**Question No: 12 ( Marks: 1 ) - Please choose one**  
**A candidate key that does not have a null value and is selected to uniquely identify all other attribute values in any given row is called a .**

- ▶ superkey
- ▶ candidate key
- ▶ **primary key** [Click here for Detail](#)
- ▶ secondary key

**Question No: 13 ( Marks: 1 ) - Please choose one**  
**Incase of Context-level Diagram, the system is represented by**  
Select correct option:

- ▶ **One process atleast** [Click here for Detail](#)
- ▶ Two processes atleast
- ▶ One process only
- ▶ Any number of processes

**Question No: 14 ( Marks: 1 ) - Please choose one**  
**A \_\_\_\_\_ is used to maintain a connection between the users of the database system.**

Select correct option:

- ▶ Mail server
- ▶ **File Server** [Click here for detail](#)
- ▶ Client-server
- ▶ None of the given.

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@gmail.com](mailto:mc100401285@gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**MIDTERM EXAMINATION**  
**Fall 2008**  
**CS403- Database Management Systems (Session - 2)**

**Question No: 1 ( Marks: 1 ) - Please choose one**

Which of the following is not a benefit of normalization?

- ▶ Minimize insertion anomalies
- ▶ Minimize deletion anomalies
- ▶ Minimize updation anomalies
- ▶ **Maximize redundancy (Page 162) rep**

**Question No: 2 ( Marks: 1 ) - Please choose one**

Which of the following is NOT a component of a DFD?

- ▶ Dataflow
- ▶ Datastore
- ▶ External entities
- ▶ **Relationship between external entities (Page 57)**

**Question No: 3 ( Marks: 1 ) - Please choose one**

Which of the following is correct regarding Dataflow diagram?

- ▶ Single DFD is required to represent a system
- ▶ The dataflow must be bidirectional
- ▶ **Created at increasing levels of detail [Click here for Detail](#)**
- ▶ Used to represent the relationships among the external entities

**Question No: 4 ( Marks: 1 ) - Please choose one**

Which of the following is CORRECT about database management system's languages?

- ▶ Data definition languages are used to specify the conceptual schema only.
- ▶ Data manipulation languages are used to create the databases.
- ▶ **Data manipulation languages are used for retrieval, insertion, deletion and modification of data.**

[Click here for Detail](#)

- ▶ Data definition languages are only used to update data in the DBMS.

**Question No: 5 ( Marks: 1 ) - Please choose one**

Controlling redundancy in a database management system DOES NOT help to

- ▶ avoid duplication
- ▶ **avoid unnecessary wastage of storage space [Click here for detail](#)**
- ▶ avoid unauthorised access to data
- ▶ avoid inconsistency among data

**Muhammad Moaaz Siddiq MCS (3rd)**

**[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)**

**Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari**

**Question No: 6 ( Marks: 1 ) - Please choose one**

Which of the following concepts is applicable with respect to 3NF?

- ▶ Full functional dependency
- ▶ Any kind of dependency
- ▶ **Transitive dependency (Page 172)**
- ▶ Partial functional dependency

**Question No: 7 ( Marks: 1 ) - Please choose one**

Consider two sets A and B. A contains 3 elements and B contains 4. How many elements do their Cartesian product contains?

- ▶ **12 (Page 129)**
- ▶ 9
- ▶ 16
- ▶ 7

**Question No: 8 ( Marks: 1 ) - Please choose one**

Consider two sets A and B. A contains 2 elements and B contains 3. How many elements do their cartesian product contains?

- ▶ **6 (Page 129)**
- ▶ 9
- ▶ 4
- ▶ 5

**Question No: 9 ( Marks: 1 ) - Please choose one**

In a conceptual database model, which of the following most likely represents a valid identifier for a class grades?

- ▶ StudentID
- ▶ StudentID, CourseID
- ▶ StudentID, CourseID, InstructorID
- ▶ **StudentID, CourseSectionID** [Click here for Detail](#) rep

**Question No: 10 ( Marks: 1 ) - Please choose one**

Identify the correct statement with respect to normalization.

- ▶ Normalization is a formal technique that can be used only at the starting phase of the database design.
- ▶ **Normalization can be used as a top-down standalone database design technique.** [Click here for detail](#) rep
- ▶ The process of normalization through decomposition must achieve the lossless join property at any cost whereas the dependency reservation property is sometimes sacrificed.
- ▶ The process of normalization through decomposition must achieve the dependency reservation property at any cost whereas the lossless join property is sometimes sacrificed.

Muhammad Moaaz Siddiq MCS (3rd)

[mc100401285@Gmail.com](mailto:mc100401285@Gmail.com)

Campus:- Institute of E-Learning & Modern Studies  
(IEMS) Samundari

**Question No: 11 ( Marks: 1 ) - Please choose one**

Consider the following relation and its sample data. (Consider that these are the only tuples for the given relation)

EmpNo	DeptNo	ProjNo
1001	01	12
1001	01	13
1002	01	12
1003	01	14

Which of the following statements is NOT correct?

- ▶ The functional dependency DeptNo  $\rightarrow$  ProjNo holds over R.
- ▶ The functional dependency EmpNo  $\rightarrow$  DeptNo holds over R.
- ▶ The functional dependency ProjNo  $\rightarrow$  DeptNo holds over R.
- ▶ **The functional dependency (EmpNo, ProjNo)  $\rightarrow$  DeptNo holds over R.**

**Question No: 12 ( Marks: 1 ) - Please choose one**

Which of the following is not true about relational tables?

- ▶ Column values are of the same kind.
- ▶ Each row is unique.
- ▶ Each column must have a unique name.
- ▶ **The sequence of rows is significant.** [Click here for Detail](#) **rep**

**Question No: 13 ( Marks: 1 ) - Please choose one**

Identify the operation which is NOT one of the parts of the five basic set operations in relational algebra?

- ▶ **Join** [Click here for Detail](#) **rep**
- ▶ Union
- ▶ Cartesian Product
- ▶ Set Difference

**Question No: 14 ( Marks: 1 ) - Please choose one**

Consider the following statements.

- A. An entity integrity constraint states that no primary key value can be null.
- B. A referential integrity constraint is specified between two relations.
- C. A foreign key cannot be used to refer to its own relation.

Identify which of the above statements is/are correct.

- ▶ **Only A (Page 134)** **rep**
- ▶ Only B
- ▶ Only B and C
- ▶ Only A and B

**Question No: 15 ( Marks: 1 ) - Please choose one**

If K is a foreign key in relation R1, then

- ▶ every tuple of R1 has a distinct value for K.
- ▶ K cannot have a null value for tuples in R1.
- ▶ **K is a key for some other relation.** [Click here for detail](#) **rep**
- ▶ K is a primary key for R1.

**Question No: 16 ( Marks: 1 ) - Please choose one**

Making a change to the conceptual schema of a database but not affecting the existing external schemas is an example of

- ▶ Physical data independence.
- ▶ Concurrency control.
- ▶ **Logical data independence.** [Click here for detail](#) **rep**
- ▶ Functional dependency