CS403 Database Management system (Final Viva)

Cardinality

Definition

Cardinality refers to the uniqueness of data values contained in a column

High cardinality

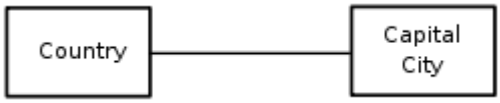
It means that the column contains a large percentage of totally unique values

Low cardinality

It means that the column contains a lot of “repeats” in its data range

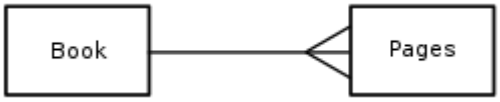
One-to-one

Relationship is a type of cardinality that refers to the relationship between two entities



One-to-many

Relationship is a type of cardinality that refers to the relationship between two entities



Primary key

Primary key is a set of one or more fields/columns of a table that uniquely identify a record in database table. It can not accept null, duplicate values. Only one Candidate Key can be Primary Key.

Foreign Key

Foreign Key is a field in database table that is Primary key in another table. It can accept multiple null, duplicate values.

What is the purpose of foreign key?

It acts as a cross-reference between tables because it **references** the primary key of another table

Candidate key

A Candidate Key is a set of one or more fields/columns that can identify a record uniquely in a table. There can be multipleCandidate Keys in one table. Each Candidate Key can work as Primary Key.

Super Key

Super key is a set of one or more than one keys that can be used to identify a record uniquely in a table. Example: Primary key, Unique key, Alternate key are subset of Super Keys.

Interface

A point where two systems, subjects, organizations, etc. meet and interact.

Inheritance

In object-oriented programming, inheritance enables new objects to take on the properties of existing objects.

Types of Inheritance

- Single Inheritance
- Multi-Level Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multipath inheritance
- Multiple Inheritance

Normalization

Definition

The process of bringing or returning something to a normal condition or state.

Database normalization is the process of organizing data into tables in such a way that the results of using the database are always unambiguous and as intended.

First Normal Form

- It should only have single (atomic) valued attributes/columns.
- Values stored in a column should be of the same domain
- All the columns in a table should have unique names.

Second Normal Form

- It should be in the First Normal form.
- And, it should not have Partial Dependency.

Third Normal Form

- It is in the Second Normal form.
- And, it doesn't have Transitive Dependency.

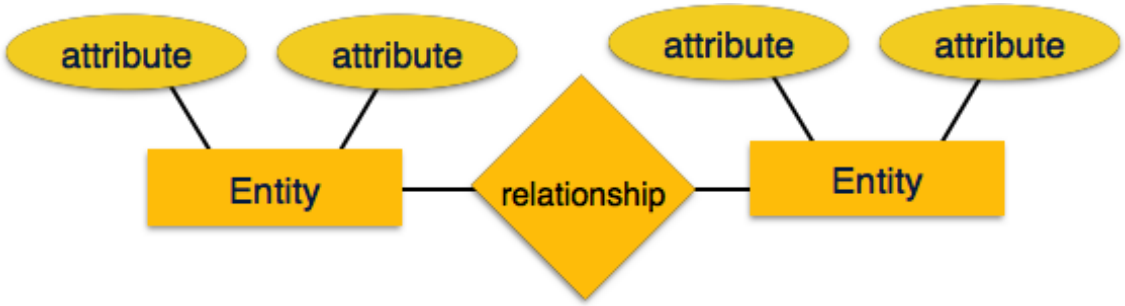
Fourth Normal Form (4NF)

- It is in the Boyce-Codd Normal Form.
- And, it doesn't have Multi-Valued Dependency.

Example of DBMS

- MySQL Database
- Oracle Database
- Microsoft SQL Server

Entities,
Which represent people, places, items, events, or concepts.
Attributes,
Which represent properties or descriptive qualities of an entity. These are also known as data elements.
Relationships, which represent the link between different entities.



Following are the different types of joins: -
1. Theta Join
2. Equi Join
3. Semi Join
4. Natural Join
5. Outer Joins

A join is an SQL operation performed to establish a connection between two or more database tables based on matching columns, thereby creating a relationship between the tables

Database

Definition

An organized collection of information in computerized format.

The GUY how created the DBMS

1970s, relational DBMS. Edgar Codd

DDL

Data definition language (DDL) skills to translate the physical design into actual database objects.

Complex Views
Complex views are by definition views of type which may comprise of many of elements, such as tables, views sequences and other similar objects of the database.
Dynamic Views
Dynamic views are those types of views for which data is not stored and the expressions used to build the view are used to collect the data dynamically.
Schema

Schema is a repository or structure to express the format and other different information about data and database,
Database Management Systems:
Database management system is software of collection of small programs to perform certain operation on data and manage the data.
Two basic operations performed by the DBMS are:

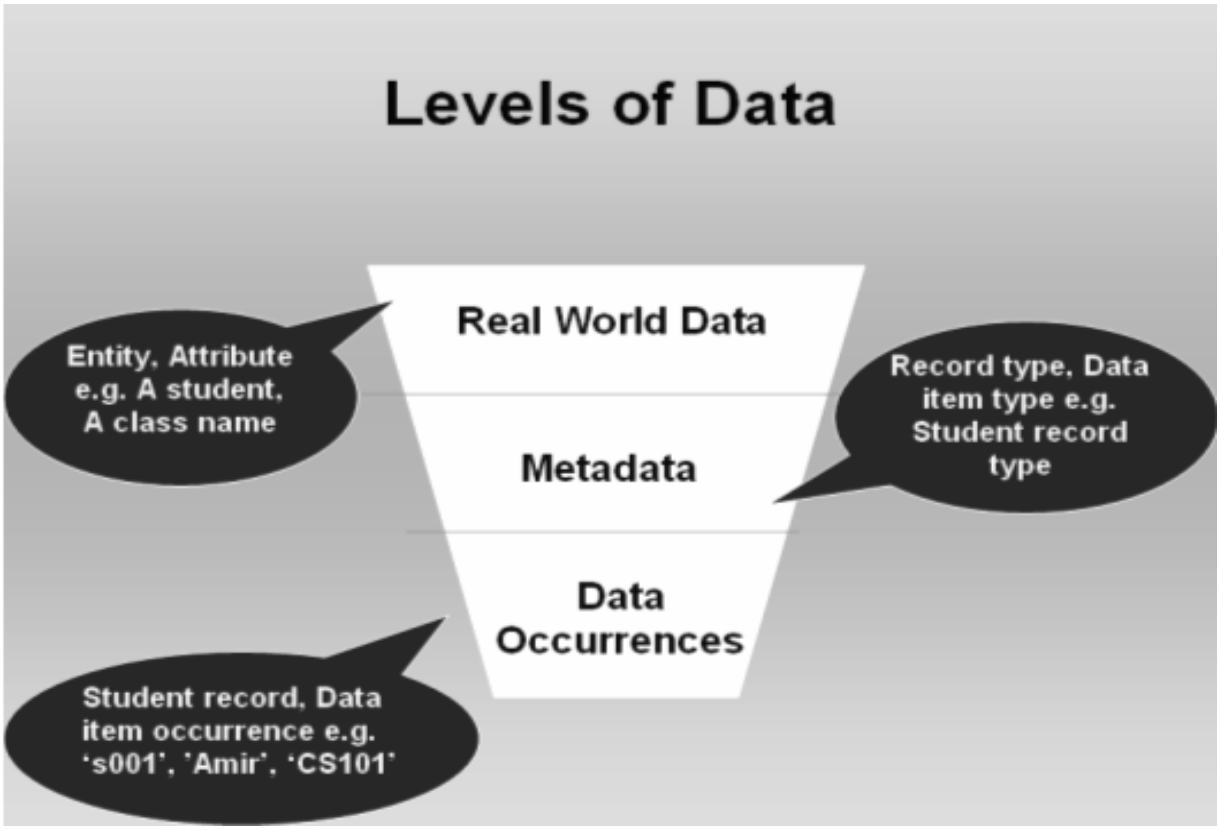
- Management of Data in the Database
- Management of Users associated with the database.

Further Advantages of Database Systems:

Database systems are very much beneficent to enterprises and businesses, some of the advantages are listed below:

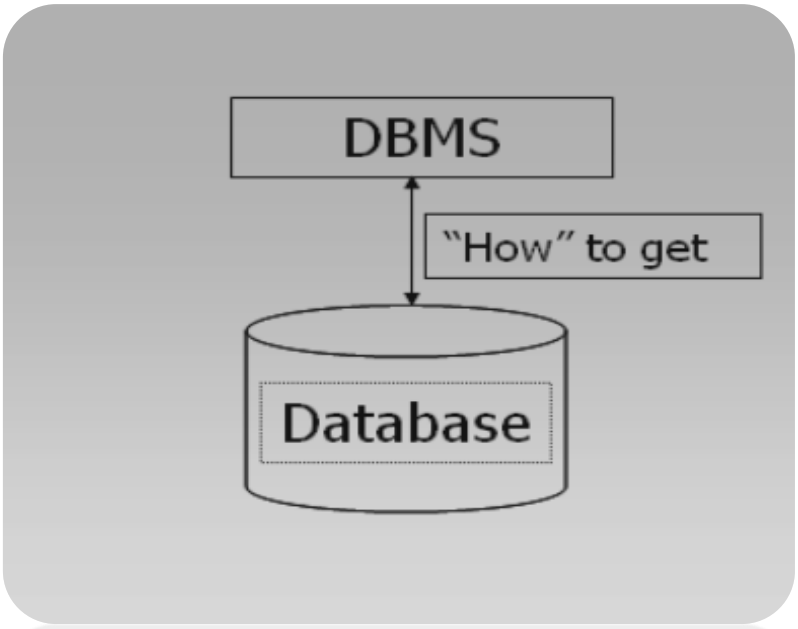
o Data consistency
o Better data security
o Faster development of new applications
o Economy of scale
o Better concurrency control
o Better backup and recovery procedures

Naïve Users
This category of users is that category who simply use the application database programs created by the programmers.
Sophisticated Users:
This type of users has some additional rights over the Naïve users, which means that they can access the data stored in the database any of their desired way.



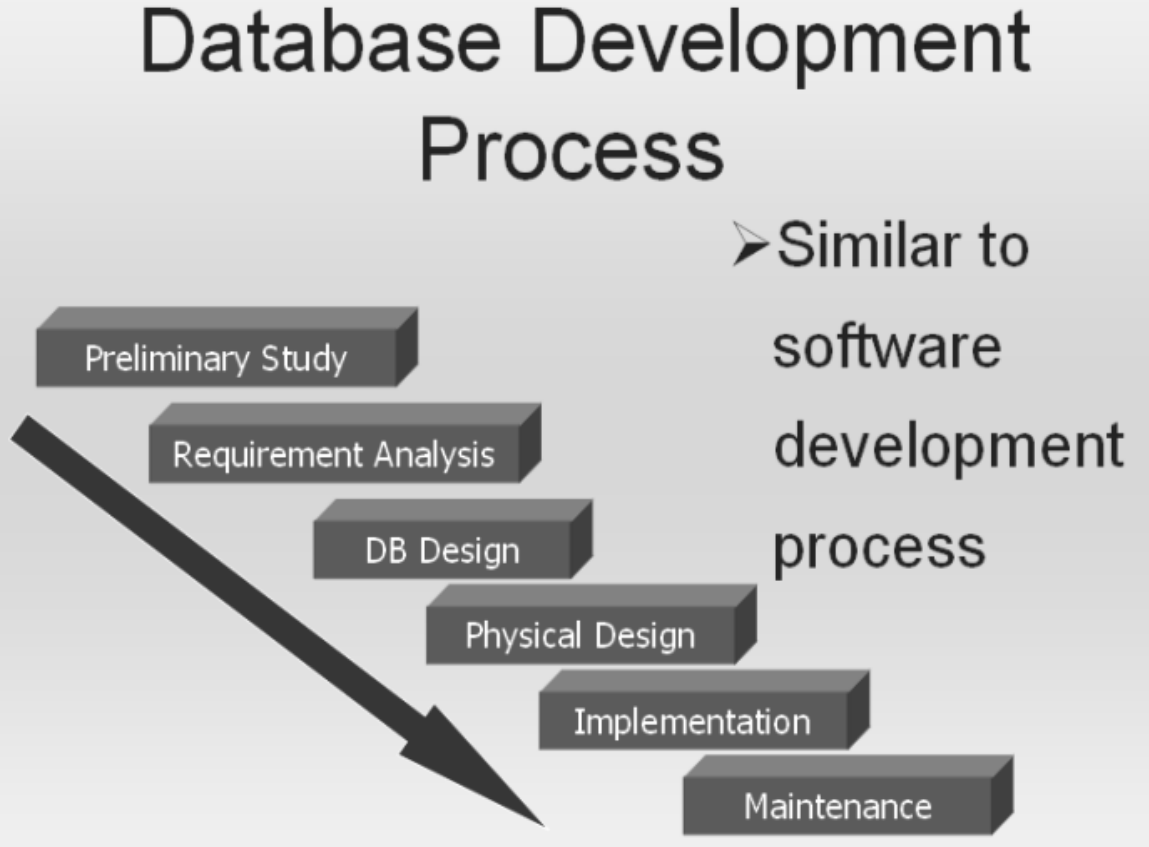
Different B/W Database and DBMS (Database Management System)

Database is used to store data and DBMS uses mechanisms to get data from the database



Internal or Physical View / Schema:

This is the level of the database which is responsible for the storage of data on the storage media and places the data in such a format that it is only readable by the DBMS.



Conceptual or Logical View:

This is the level of database architecture which contains the definition of all the data to be stored in the database and also contains rules and information about that structure and type of that data.

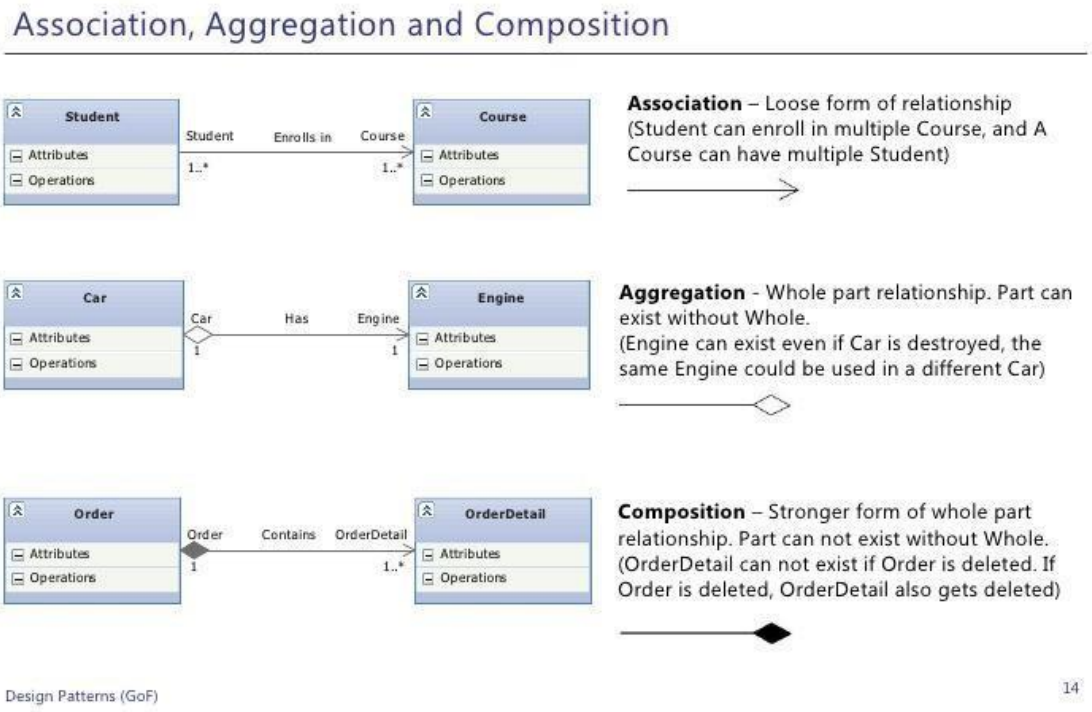
Implementation:

This phase is specific to writing the application programs needed to carry out different activities according to use requirements.

Association
is a relationship where all objects have their own lifecycle and there is no owner.

Aggregation
is a specialized form of Association where all objects have their own lifecycle.

Composition
is again specialized form of Aggregation and we can call this as a "death" relationship.



Generalization

Generalization is a bottom-up approach in which multiple lower-level entities are combined to form a single higher-level entity. Generalization is usually used to find common attributes among entities to form a generalized entity. It can also be thought of as the opposite of specialization.

Use Case Diagram
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

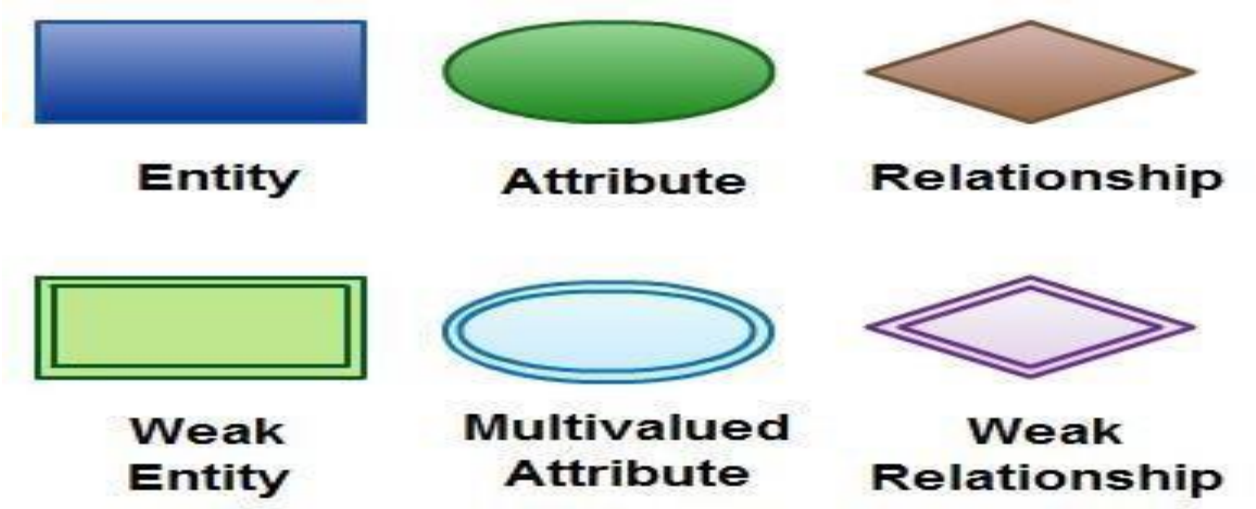
Use Case
Draw use cases using ovals. Label with ovals with verbs that represent the system's functions.

A **"uses"** relationship indicates that one use case is needed by another in order to perform a task.

An **"extends"** relationship indicates alternative options under a certain use case.

An **include relationship** is a relationship between two use cases It indicates that the use case to which the arrow points is included in the use case on the other side of the arrow.

Entity Relationship Diagram (ERD)
An entity-relationship diagram—otherwise known as an ERD—is a data modeling technique that creates an illustration of an information system's entities and the relationships between those entities.



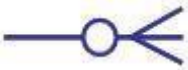
Cardinality and Modality
Cardinality and Modality work together to define the relationship.

Cardinality indicates the maximum number of times an instance in one entity can be associated with instances in the related entity.

Modality indicates the minimum number of times an instance in one entity can be associated with an instance in the related entity.

Reading Modality and Cardinality

from Zero to Many



from One to Many

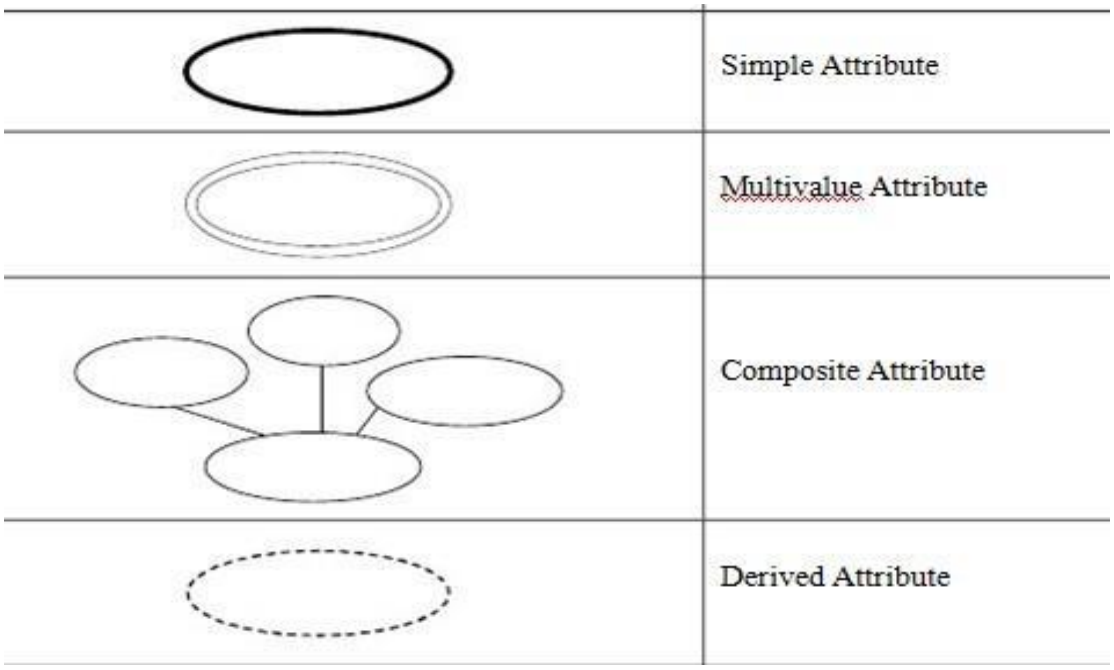
from One to One

i.e., one and only one

from Zero to One

Types of Attributes

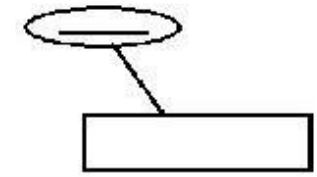
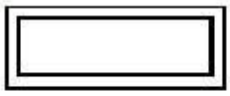
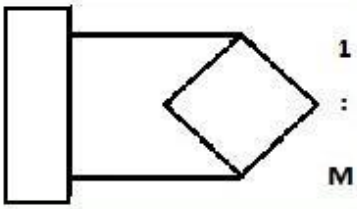
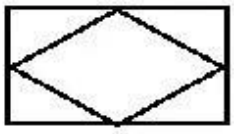
- **Simple attribute** - Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute** - Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute** - Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data_of_birth.
- **Single-value attribute** - Single-value attributes contain single value. For example - Social_Security_Number.
- **Multi-value attribute** - Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc.



Types of Entity –

Strong Entity Types
Recursive Entity Types
Weak Entity Types
Composite Entity Types or Associative Entity Types

Notations Of different Entity Type in ER Diagram

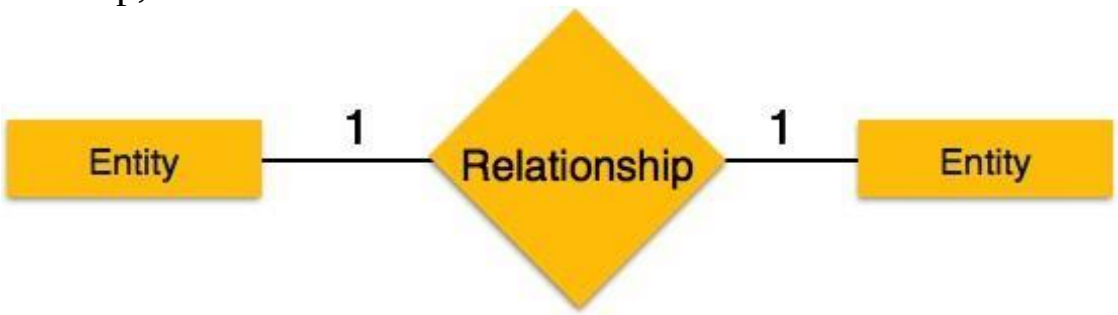
	Entity
	Strong Entity Type
	Weak Entity Type
	Recursive Entity Type
	Composite Entity Type (or) Associative Entity

Relationship
Relationships are represented by diamond-shaped box. Name of the relationship is written inside the diamond-box.

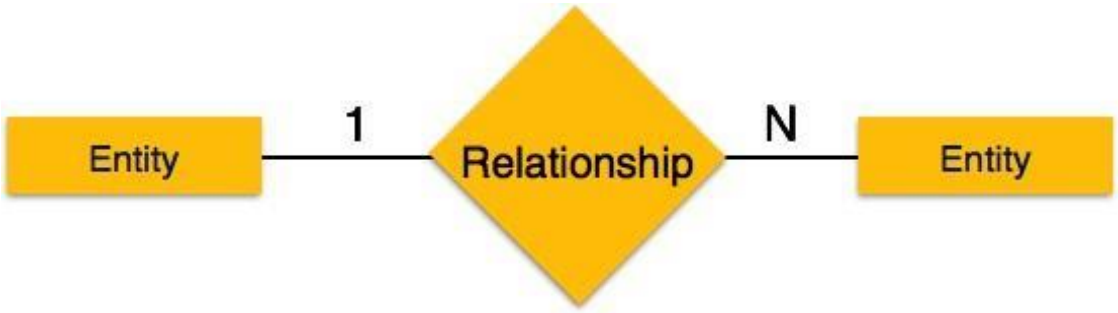
Binary Relationship
A relationship where two entities are participating is called a binary relationship.

Cardinality Relationship
Cardinality is the number of instance of an entity from a relation that can be associated with the relation.

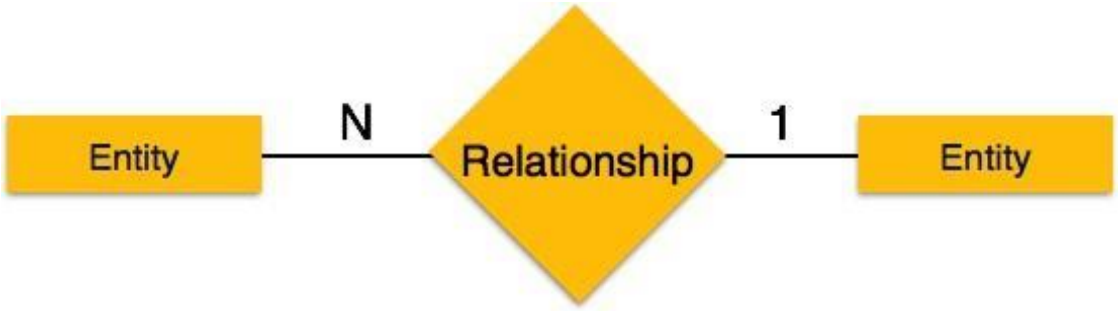
One-to-one - When only one instance of an entity is associated with the relationship, it is marked as '1:1'.



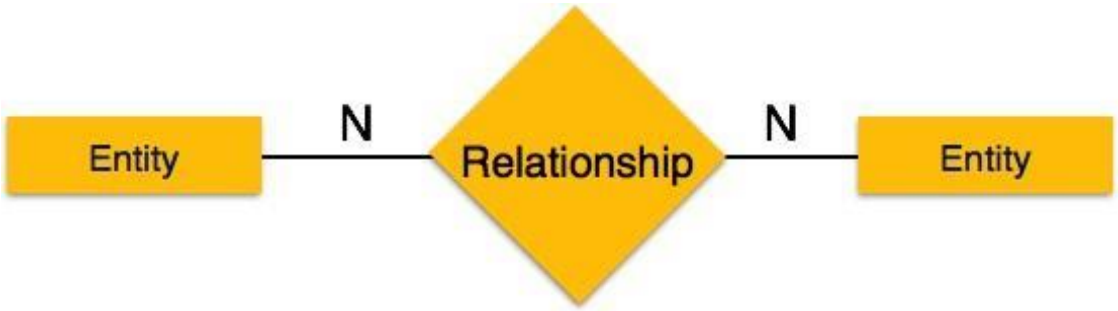
One-to-many - When more than one instance of an entity is associated with a relationship, it is marked as '1:N'.



Many-to-one - When more than one instance of entity is associated with the relationship, it is marked as 'N:1'.



Many-to-many - The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship.

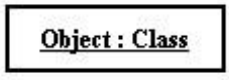


Sequence Diagrams

The Sequence Diagram shows how the objects interact with others in a particular scenario of a use case.

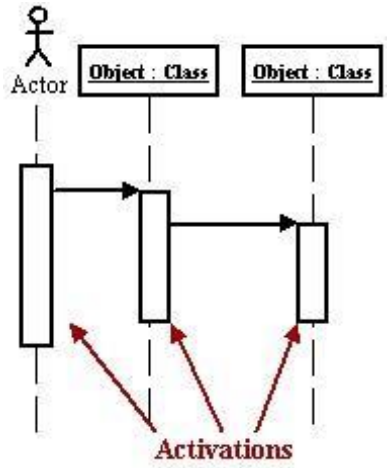
Class roles

Class roles describe the way an object will behave in context. Use the UML object symbol to illustrate class roles, but don't list object attributes.



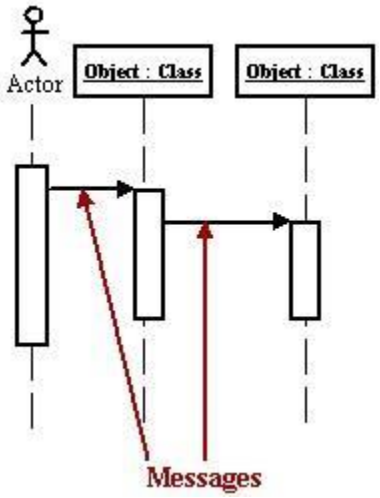
Activation

Activation boxes represent the time an object needs to complete a task.



Messages

Messages are arrows that represent communication between objects. Use half-arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.



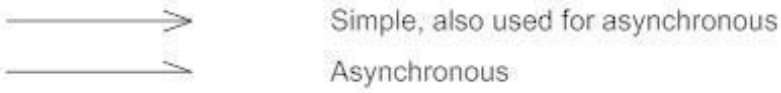
Types of Messages in Sequence Diagrams

Synchronous Message

A synchronous message requires a response before the interaction can continue. It's usually drawn using a line with a solid arrowhead pointing from one object to another.

Asynchronous Message

Asynchronous messages don't need a reply for interaction to continue. Like synchronous messages, they are drawn with an arrow connecting two lifelines; however, the arrowhead is usually open and there's no return message depicted.



Reply or Return Message

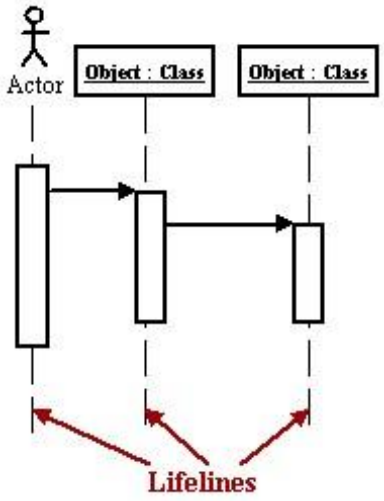
A reply message is drawn with a dotted line and an open arrowhead pointing back to the original lifeline.

Self Message

A message an object sends to itself, usually shown as a U shaped arrow pointing back to itself.

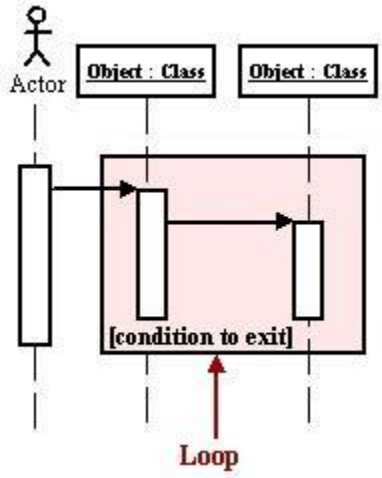
Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.



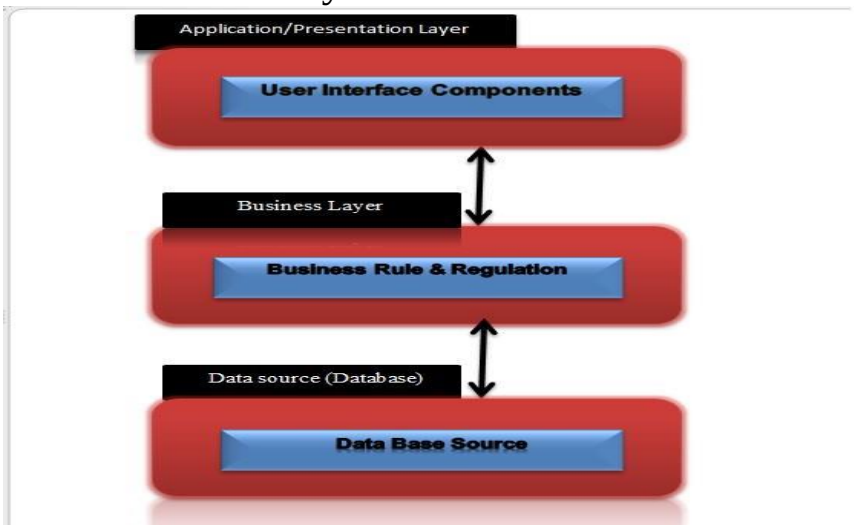
Loops

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [ ].



Architecture Design Diagram

The architecture adopted is Three-tier web architecture. The architecture consists of three layers which are:



1. Presentation tier

Front End is designed in programming language like PHP, C# or Java etc...

2. Application tier

The middle dynamic content processing and generation level server layer is created in PHP, C# or Java etc...

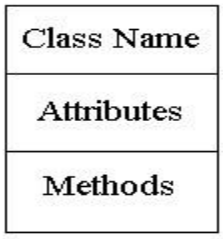
3. Data tier

SQL or MySQL is used as back-end database server.

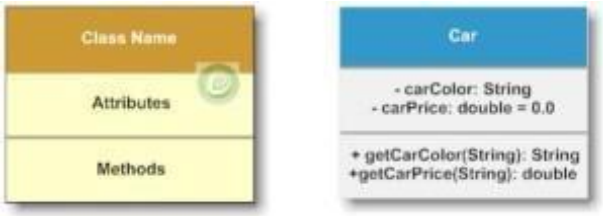
Class Diagram

Class diagrams are used to represents real life objects in our design by capturing its data methods and relationships of these methods with each other.

Syntax



Class Diagram Example



Database Design

In our daily life every object has some data associated with it. This data can be managed through data base design.

Alternate key

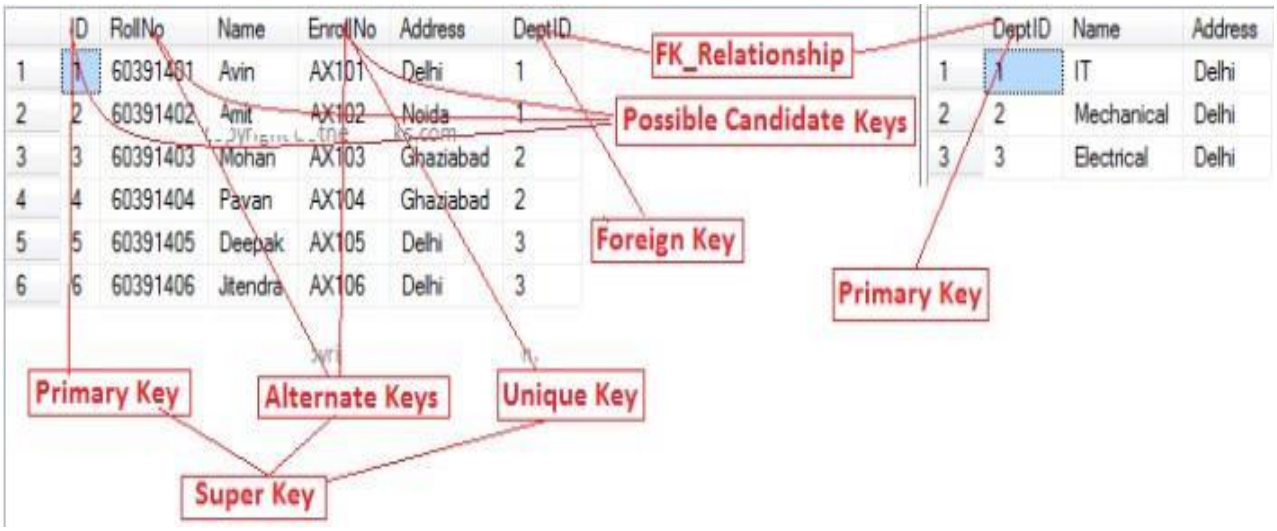
A Alternate key is a key that can be work as a primary key. Basically it is a candidate key that currently is not primary key.

Composite/Compound Key

Composite Key is a combination of more than one fields/columns of a table. It can be a Candidate key, Primary key.

Unique Key

Uniquekey is a set of one or more fields/columns of a table that uniquely identify a record in database table.



Joins

Join is a combination of a Cartesian product followed by a selection process. A Join operation pairs two tuples from different relations, if and only if a given join condition is satisfied.

Theta (θ) Join

Theta join combines tuples from different relations provided they satisfy the theta condition.

Equijoin

When Theta join uses only equality comparison operator, it is said to be equijoin.

Natural Join ( )

Natural join does not use any comparison operator. It does not concatenate the way a Cartesian product does. We can perform a Natural Join only if there is at least one common attribute that exists between two relations.

Outer Joins

Theta Join, Equijoin, and Natural Join are called inner joins. An inner join includes only those tuples with matching attributes and the rest are discarded in the resulting relation. Therefore, we need to use outer joins to include all the tuples from the participating relations in the resulting relation. There are three kinds of outer joins - left outer join, right outer join, and full outer join.

Normalizations mai tables ke tadat km hote hai zaiyda?

Ans: zaiyda hote hn.

ER diagram mai entity or attributes ko mila k kia bnta hai?

Ans: table bnta hai

use case mai boundary line say bhair Actor q rakha jata hai?

Ans: Q k actor nay wo task perform karny hoty hn jo boundary line k andr hoty hn