CS619
Final Viva
Preparition

MUNIR HUSSAIN

CS403 FAVOURITE SUBJECT + PPT QUESTIONS

Let’s start with the first set of slides

1

Scope of the project

- Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, features, functions, tasks, deadlines, and ultimately costs. In other words, it is what needs to be achieved and the work that must be done to deliver a project

Functional Requirements

- A functional requirement describes what a software system should do, Functional Requirements: specify the functionality of the system
- Functional requirements are the activities that a system must perform

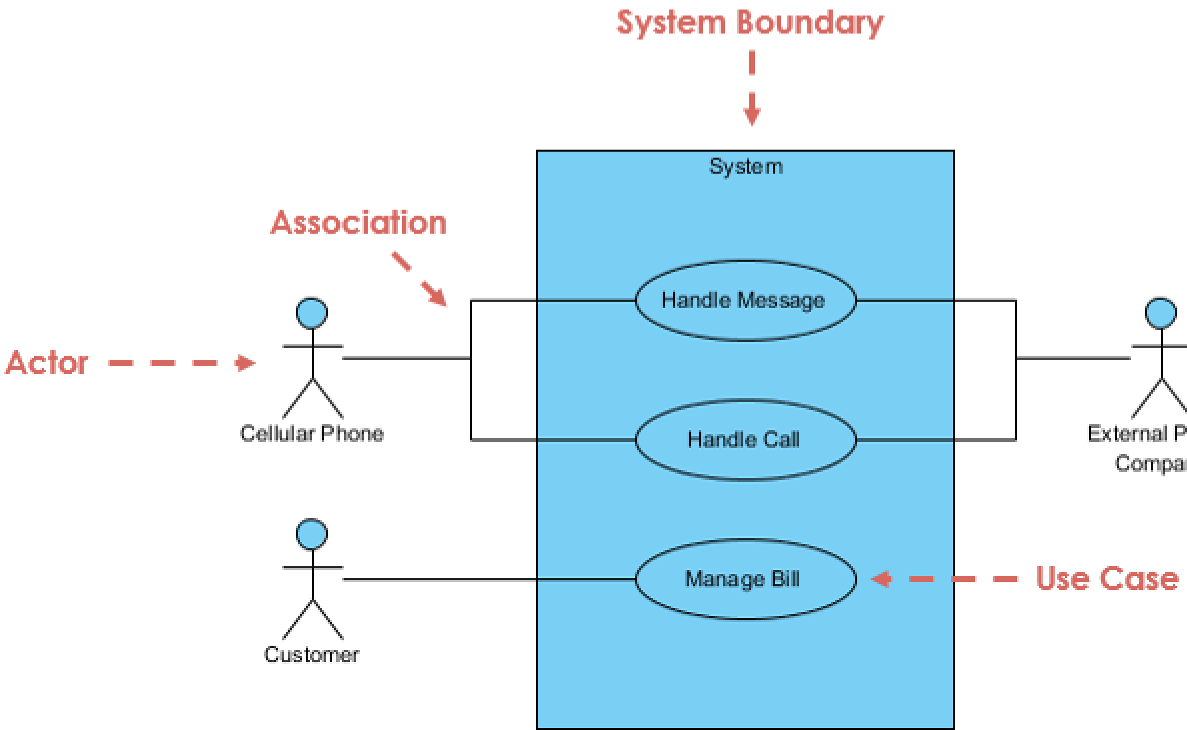
Non-Functional Requirements

- define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs. … They ensure the usability and effectiveness of the entire system

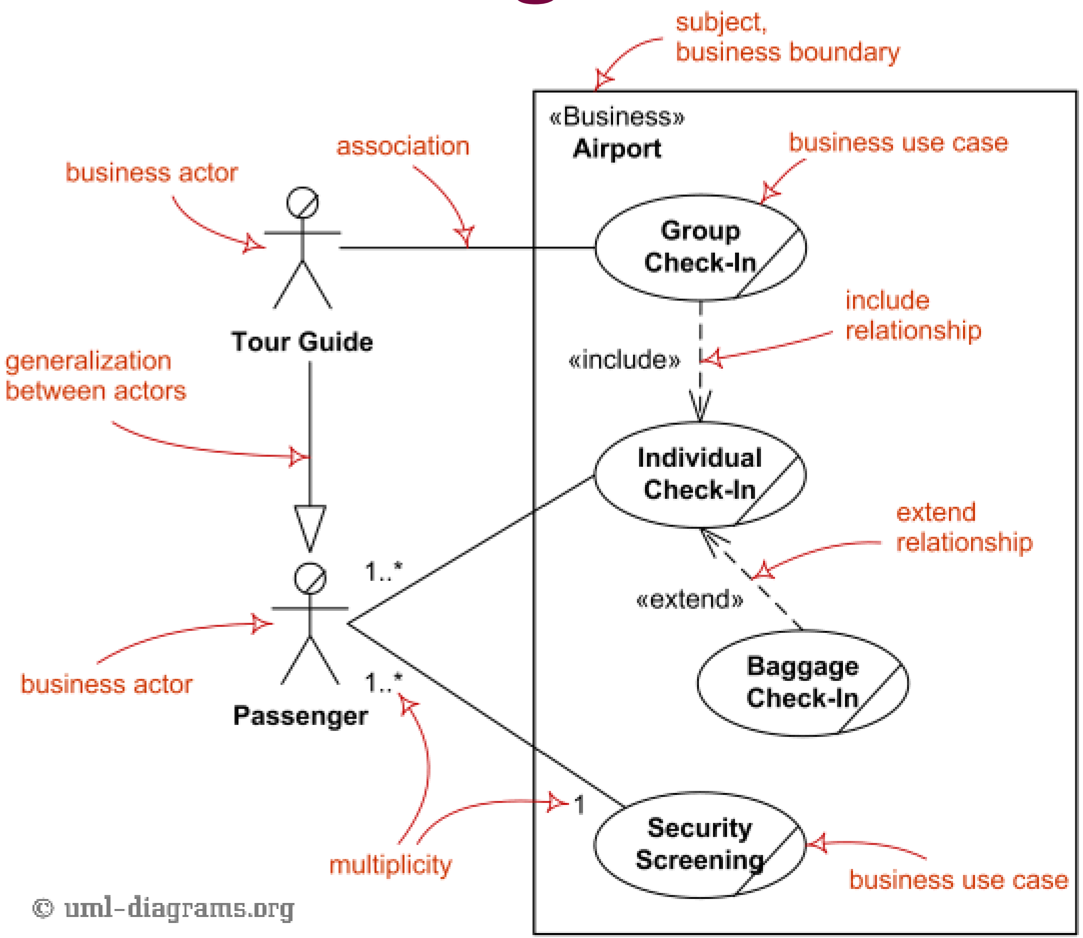
Use Case Diagram

- 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 Diagram Example



Use Case Diagram Relationships



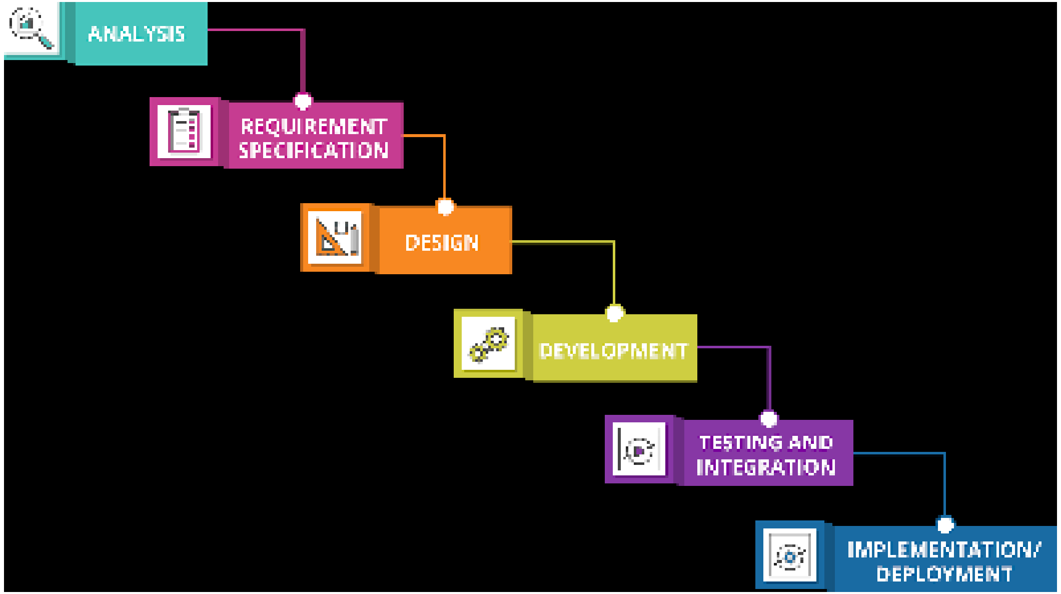
Adopted Methodology

- It is a method to use development of project. We have adopted the VU Process Model which is the combination of Waterfall methodology and Spiral methodology. VU Process Model will combine the benefits of Waterfall and Spiral methodologies.

The Waterfall Model

- The waterfall model is a sequential design process, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation and Maintenance

The Waterfall Model Diagram



ANALYSIS

REQUIREMENT SPECIFICATION

DESIGN

DEVELOPMENT

TESTING AND INTEGRATION

IMPLEMENTATION/ DEPLOYMENT

Reasons for Choosing VU process Model

- The main idea to choose is to get the benefits of both these models.
- It is sequential model with backward repetition.
- We want to achieve the linear nature of waterfall and iterative plus risk reduction nature of spiral model.

Continuous

Reasons for Choosing VU process Model

- In vu process model we will be work in phases to complete the given project.
- 5. All the activities are performed in a sequence in VU Process Model.

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

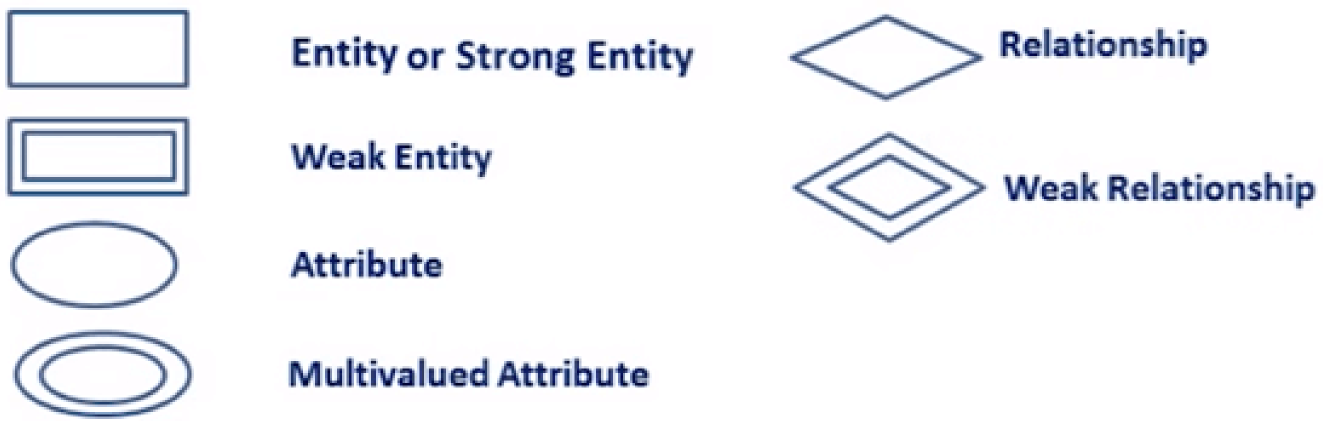
ER Diagrams Symbols & Notations

- Rectangles: This Entity Relationship Diagram symbol represents entity types
- Ellipses : Symbol represent attributes
- Diamonds: This symbol represents relationship types
- Lines: It links attributes to entity types and entity types with other relationship types

ER Diagrams Symbols & Notations

- Primary key: attributes are underlined
- Double Ellipses: Represent multi-valued attributes

ER Diagrams Symbols & Notations

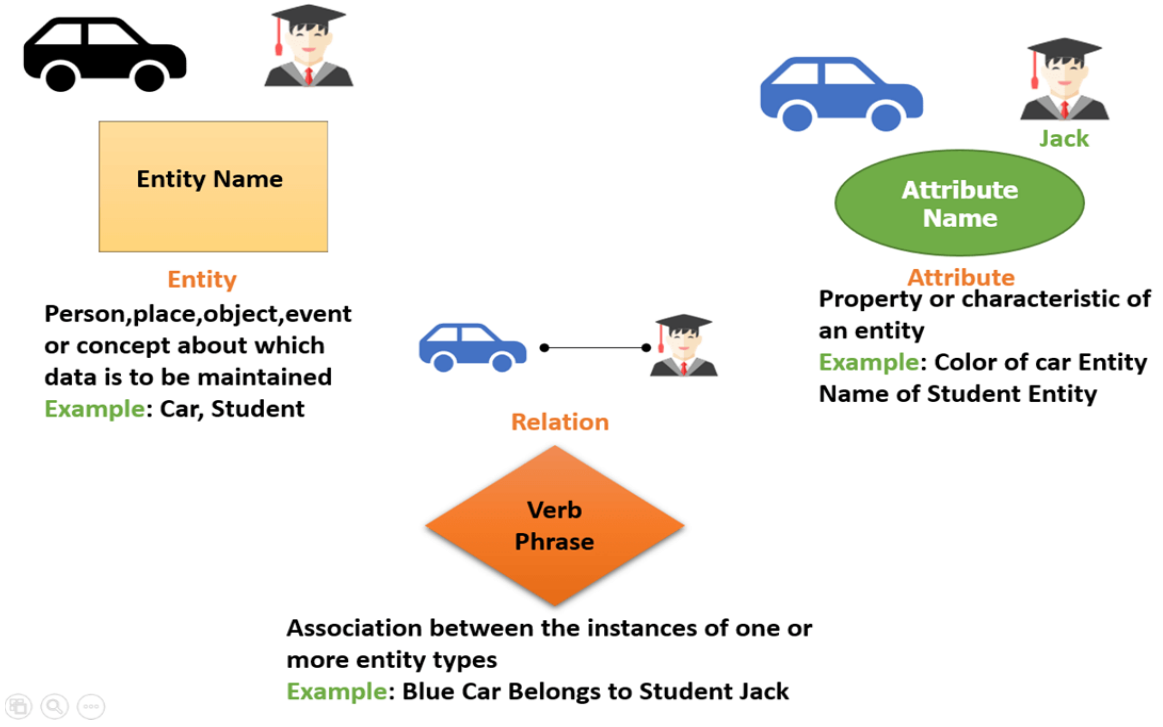


Components of the ER Diagram

This model is based on three basic concepts:

- Entities
- Attributes
- Relationships

ER Diagram Examples



Entity

Person,place,object,event or concept about which data is to be maintained

Example: Car, Student

Relation

Association between the instances of one or more entity types

Example: Blue Car Belongs to Student Jack

Attribute

Property or characteristic of an entity

Example: Color of car Entity

Name of Student Entity

WHAT IS ENTITY?

- An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity

Examples of entities:

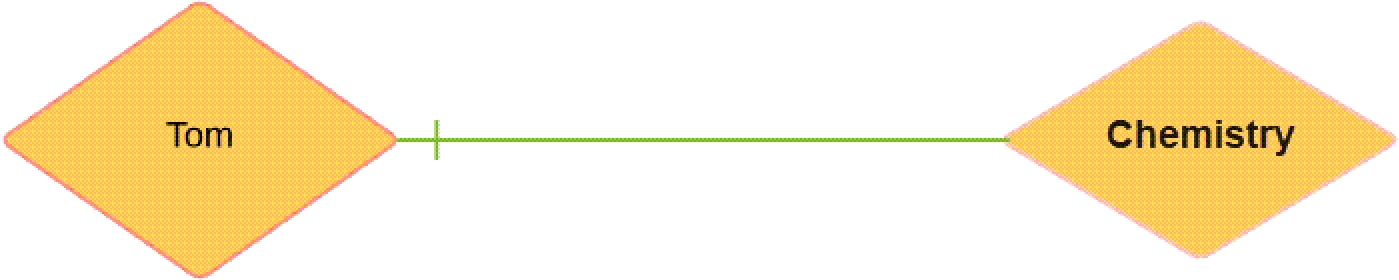
- Person: Employee, Student, Patient
- Place: Store, Building
- Object: Machine, product, and Car
- Event: Sale, Registration, Renewal
- Concept: Account, Course

ENTITY TYPES

- Strong Entity
- Recursive Entity
- Weak Entity

Relationship In ERD

- Relationship is nothing but an association among two or more entities. E.g., Tom works in the Chemistry department.

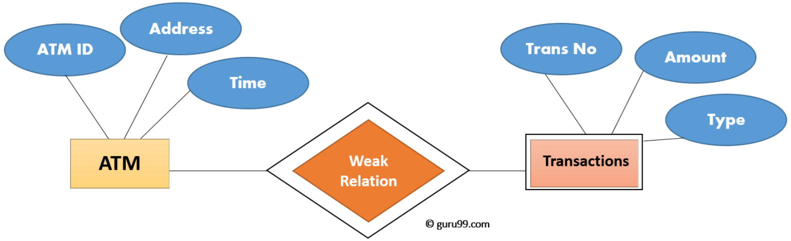


Relationship In ERD example:

- You are attending this lecture
- I am giving the lecture
- Just look entities, we can classify relationships according to relationship-types:
- A student attends a lecture
- A lecturer is giving a lecture.

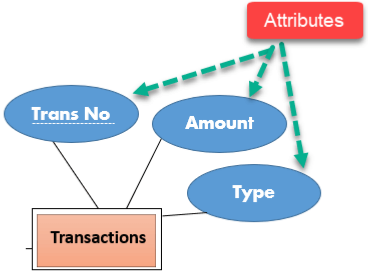
Weak Entities

- A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation.



Attributes In ERD

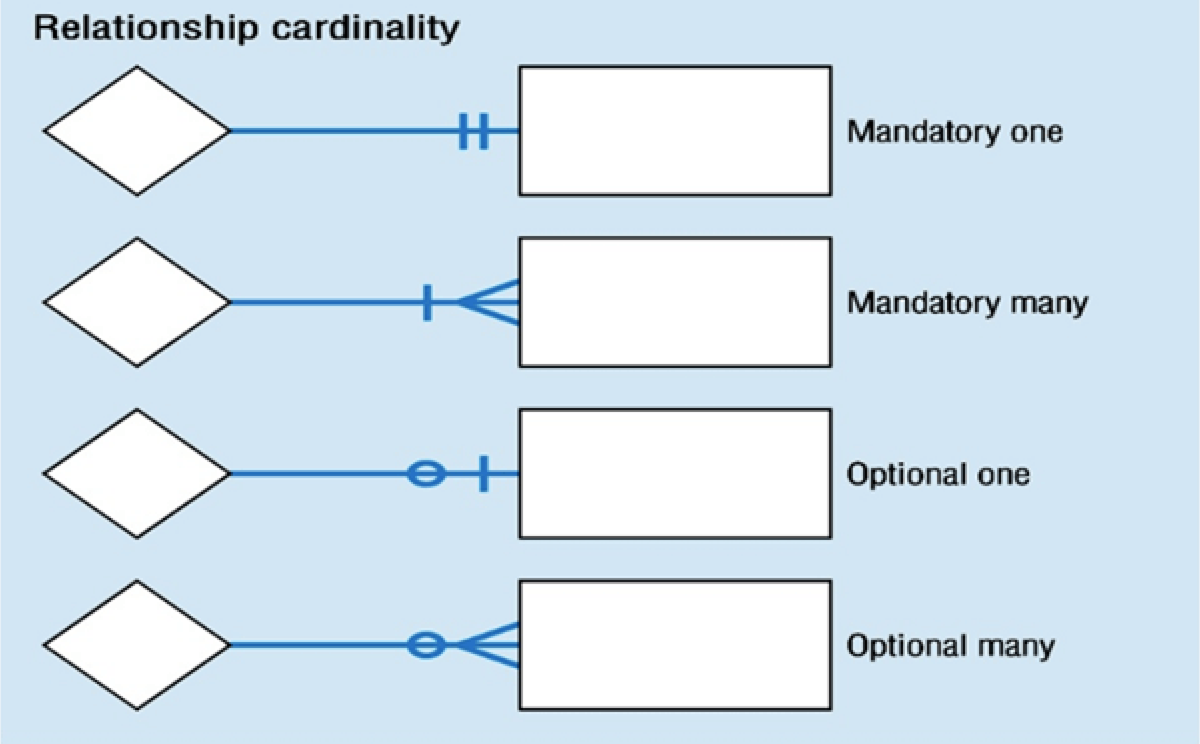
- It is a single-valued property of either an entity-type or a relationship-type.
- For example, a lecture might have attributes: time, date, duration, place, etc.
- An attribute in ER Diagram examples, is represented by an Ellipse



Cardinality

- Defines the numerical attributes of the relationship between two entities or entity sets.
- Different types of cardinal relationships are:
- One-to-One Relationships
- One-to-Many Relationships
- May to One Relationships
- Many-to-Many Relationships

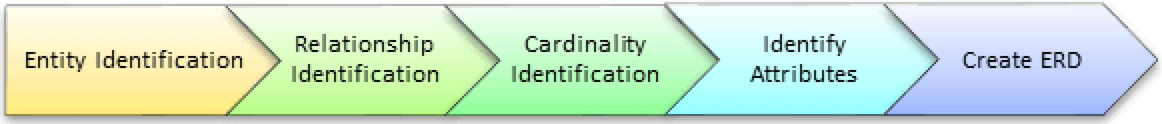
Cardinal Relationships



Modality In ERD

- Components used in the creation of an ERD: Entity – A person, place or thing about which we want to collect and store multiple instances of data. Modality refers to the minimum number of times an instance in one entity can be associated with an instance in the related entity.
-

How to Create an ERD



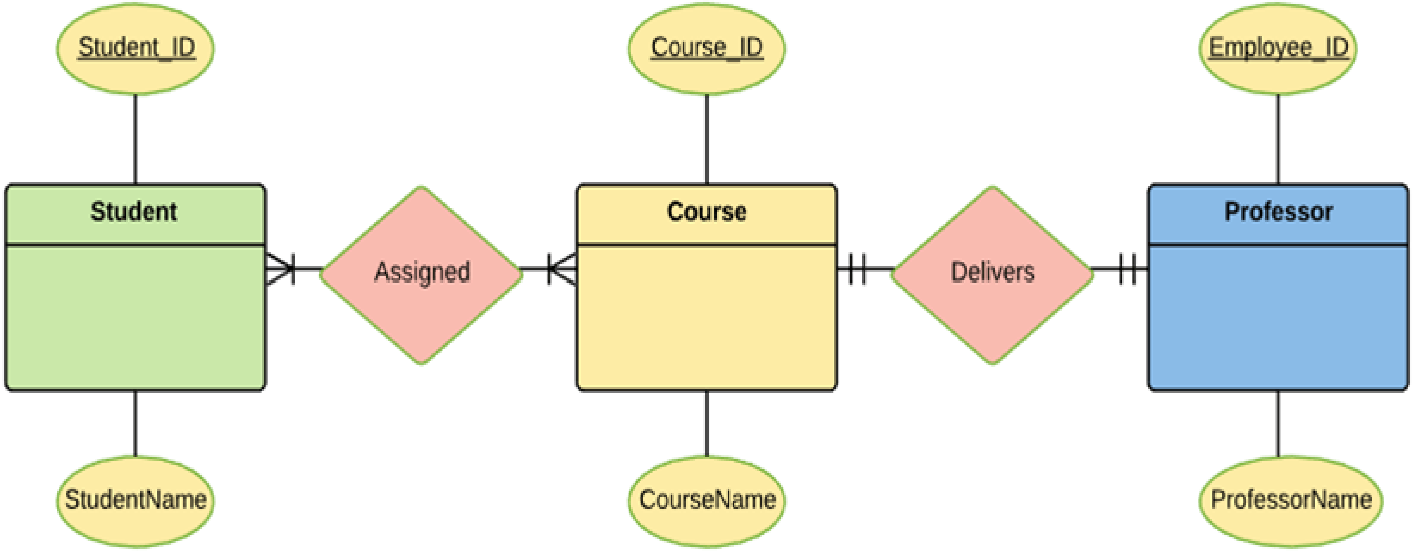
Entity Identification

Relationship Identification

Cardinality Identification

Identify Attributes

Create ERD



What is Class Diagram

- Class Diagram defines the types of objects in the system and the different types of relationships that exist among them. It gives a high-level view of an application. This modeling method can run with almost all Object-Oriented Methods. A class can refer to another class. A class can have its objects or may inherit from other classes

Benefits of Class Diagram

- Class Diagram Illustrates data models for even very complex information systems
- It helps for better understanding of general schematics of an application.
- Allows drawing detailed charts which highlights code required to be programmed
- Helpful for developers and other stakeholders.

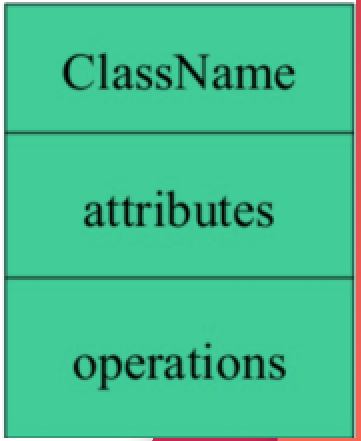
Essential elements of A UML class diagram

Essential elements of UML class diagram are:

- Class Name
- Attributes
- Operations

Class Name In Class Diagram

- The name of the class is only needed in the graphical representation of the class. It appears in the topmost compartment.
- same relationships, attributes, operations, & semantics. The class is rendered as a rectangle, including its name, attributes, and operations in sperate compartments.



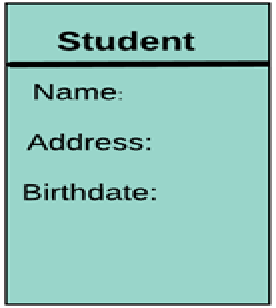
ClassName

attributes

operations

Attributes In Class Diagram

- An attribute is named property of a class which describes the object being modeled. In the class diagram, this component is placed just below the name-compartment.



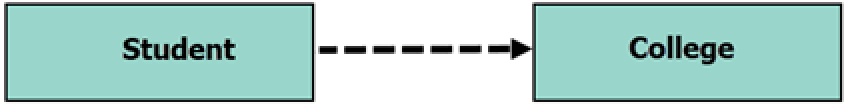
Relationships In Class Diagram

There are mainly three kinds of relationships in UML:

- Dependencies
- Generalizations
- Associations

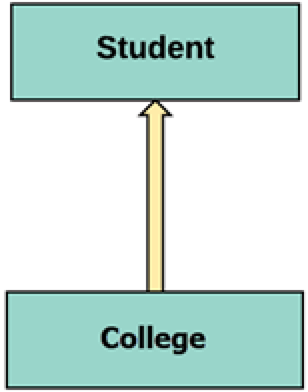
Dependency

- A dependency means the relation between two or more classes in which a change in one may force changes in the other. However, it will always create a weaker relationship. Dependency indicates that one class depends on another.
- E.g.



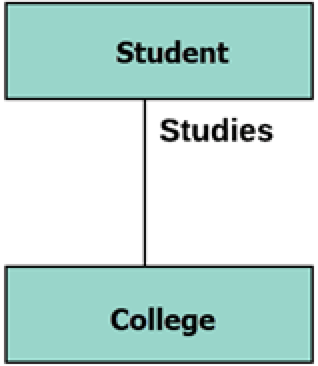
Generalization:

- A generalization helps to connect a subclass to its superclass. A sub-class is inherited from its superclass. Generalization relationship can't be used to model interface implementation. Class diagram allows inheriting from multiple super classes.



Association

- This kind of relationship represents static relationships between classes A and B. For example; an employee works for an organization.

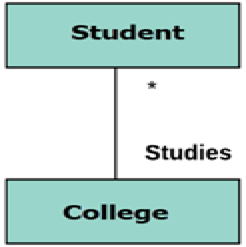


Some rules for Association

- Association is mostly verb or a verb phrase or noun or noun phrase.
- It should be named to indicate the role played by the class attached at the end of the association path.
- Mandatory for reflexive associations

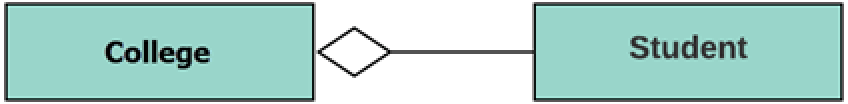
Multiplicity

- A multiplicity is a factor associated with an attribute. It specifies how many instances of attributes are created when a class is initialized. If a multiplicity is not specified, by default one is considered as a default multiplicity.



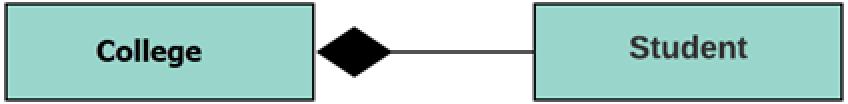
Aggregation

- Aggregation is a special type of association that models a whole- part relationship between aggregate and its parts.



Composition

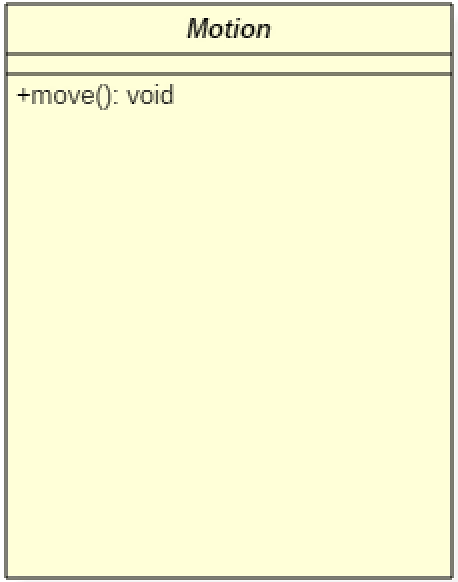
- The composition is a special type of aggregation which denotes strong ownership between two classes when one class is a part of another class.



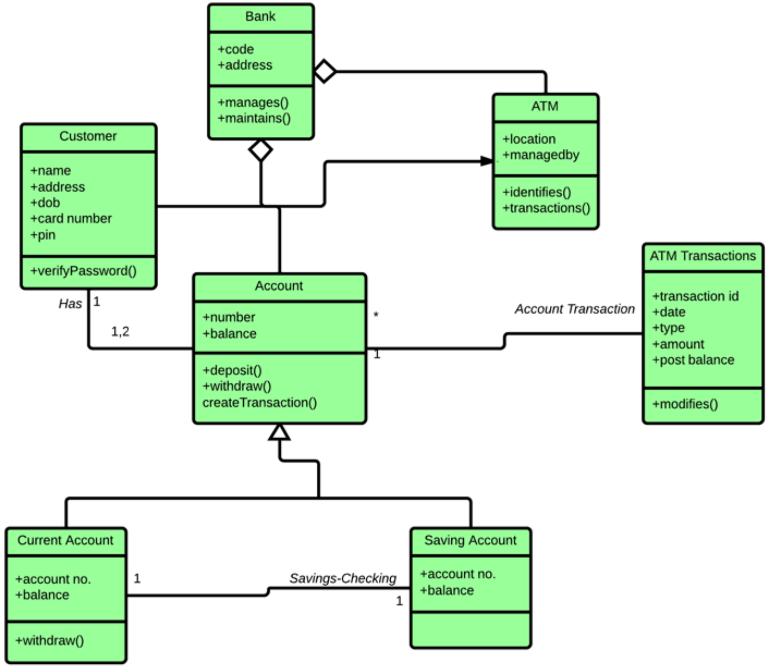
Abstract Classes

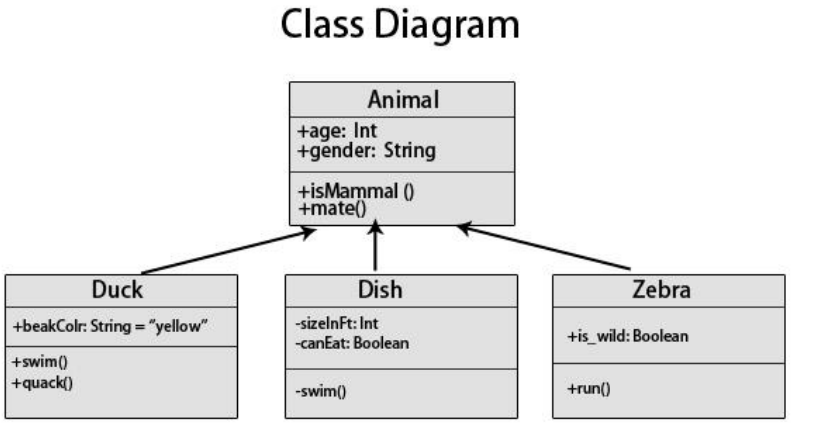
- This abstract class method can be used by any object such as a car, an animal, robot, etc. for changing the current position. It is efficient to use this abstract class method with an object because no implementation is provided for the given function. We can use it in any way for multiple objects

Abstract Class Notation



Example of UML Class Diagram

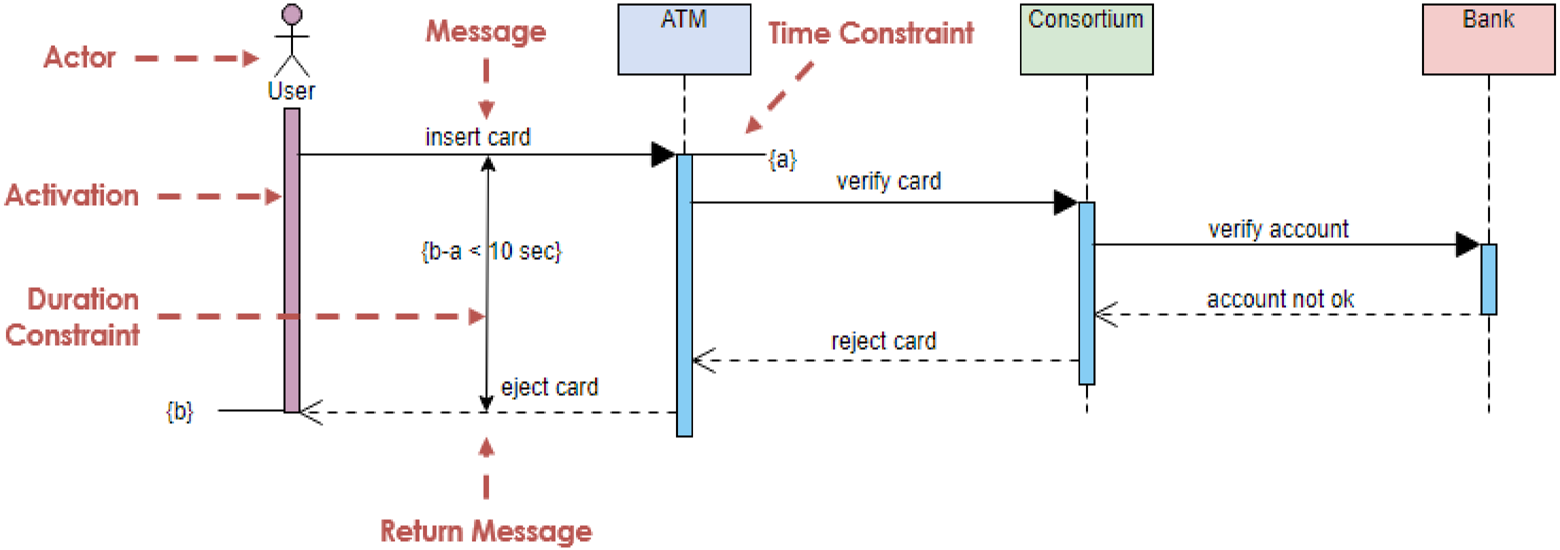




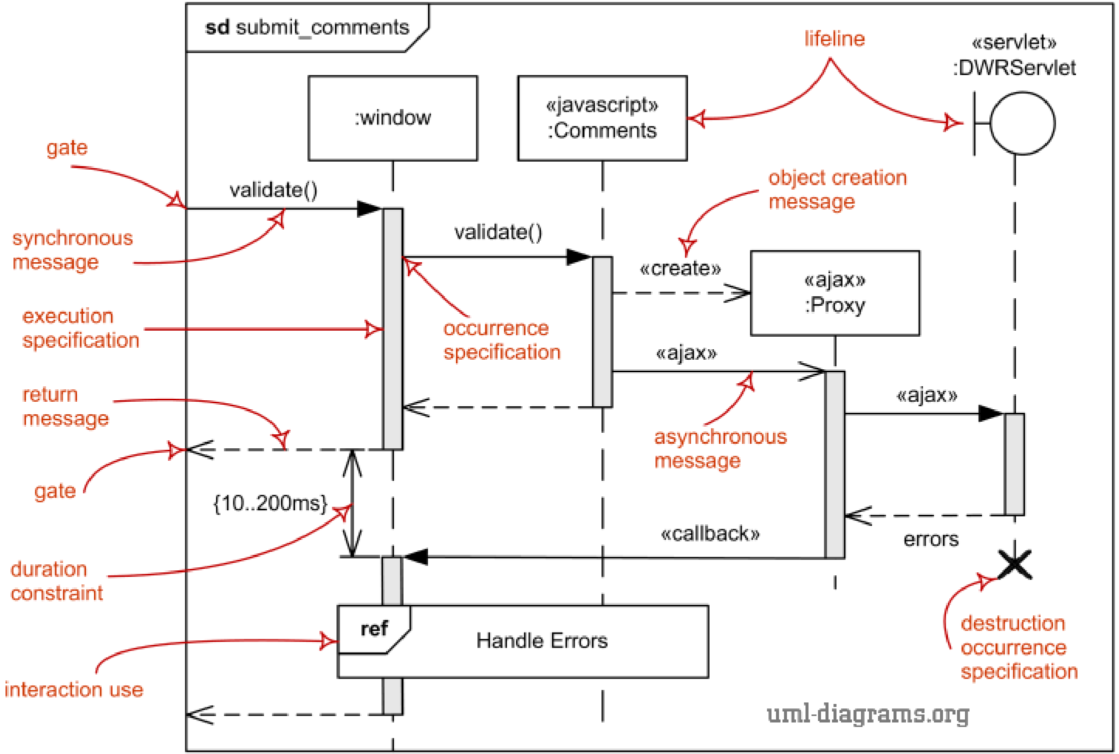
What is a Sequence Diagram

- A SEQUENCE DIAGRAM simply depicts interaction between objects in a sequential order. The purpose of a sequence diagram in UML is to visualize the sequence of a message flow in the system. The sequence diagram shows the interaction between two lifelines as a time-ordered sequence of events

Sequence Diagram Example



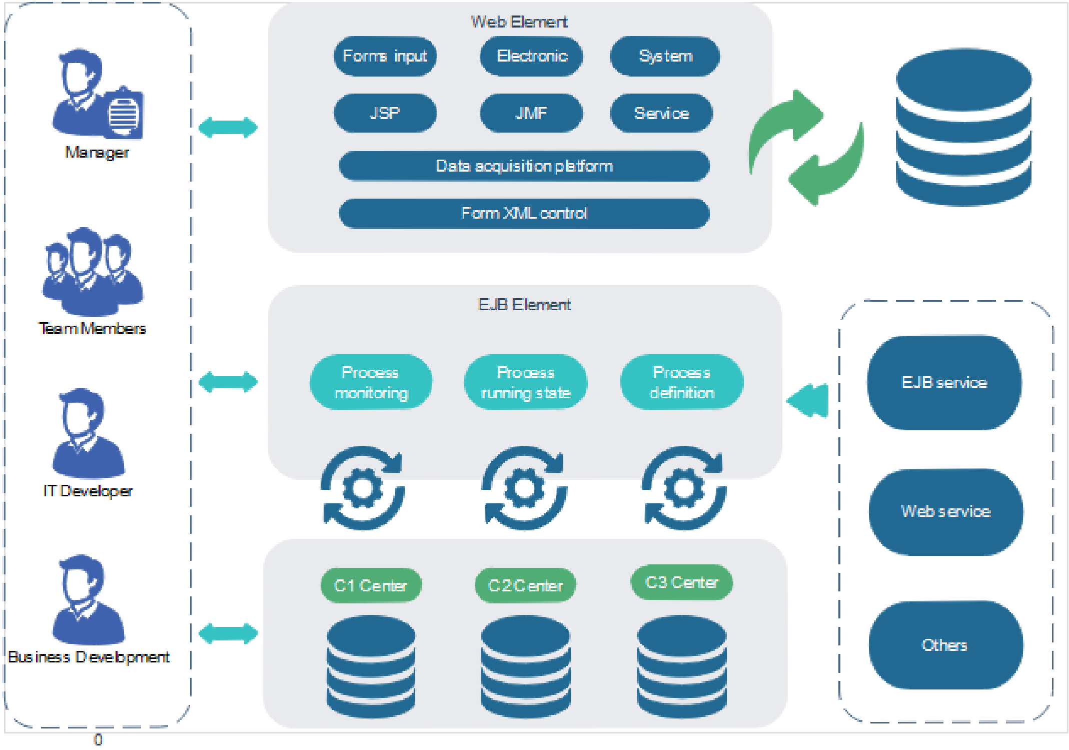
Sequence Diagram Example

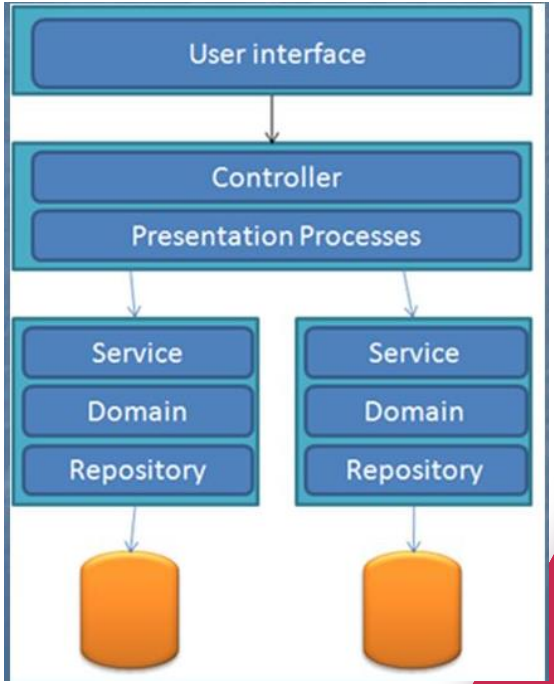


Architecture Design Diagram

- IEEE defines architectural design as 'the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system

Architecture Design Diagram Example



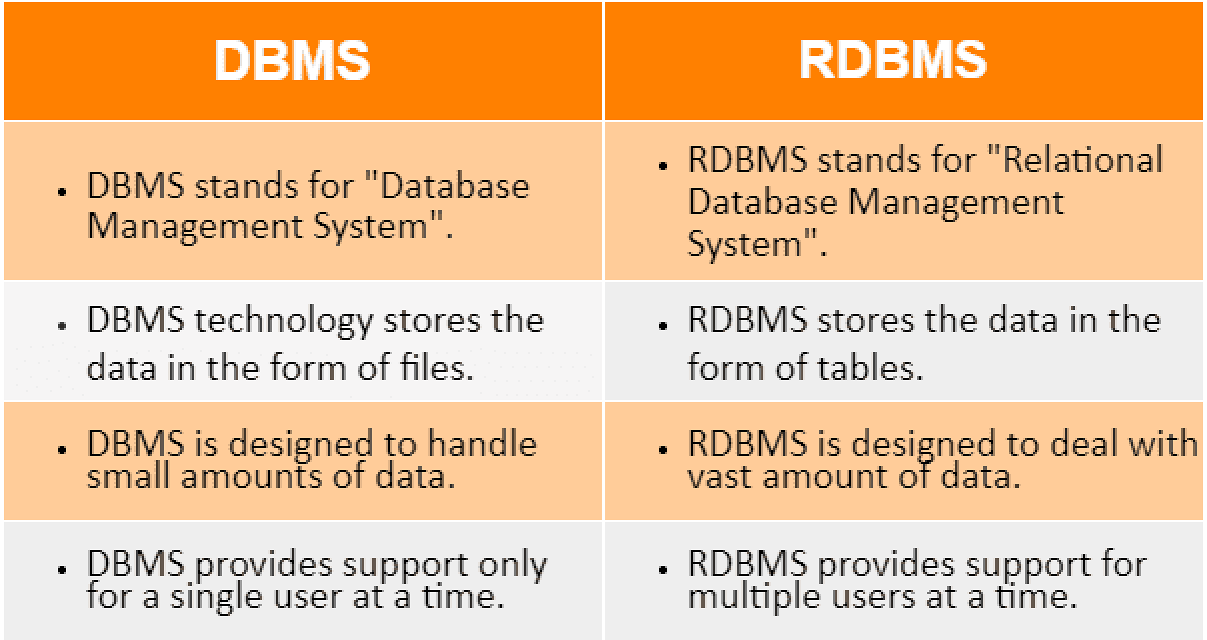


CS403 Database Management System as Favorite Subject Viva Preparation

Database Management System

- Database Management System is the set of programs or system which is used to create and maintain database.
- Data: is the collection of raw, facts and figures like college admission form consists of data.
- Database: is organized collection of related data.

Difference between DBMS and RDBMS



DBMS	RDBMS
• DBMS stands for "Database Management System".	• RDBMS stands for "Relational Database Management System".
• DBMS technology stores the data in the form of files.	• RDBMS stores the data in the form of tables.
• DBMS is designed to handle small amounts of data.	• RDBMS is designed to deal with vast amount of data.
• DBMS provides support only for a single user at a time.	• RDBMS provides support for multiple users at a time.

Truncate, Delete and Drop

- The SQL TRUNCATE TABLE command is used to delete complete data from an existing table.
- The SQL DROP TABLE statement is used to remove a table definition and all the data, indexes, triggers, constraints and permission specifications for that table.

What is trigger

- A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. DML triggers run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or vie

What is Normalization

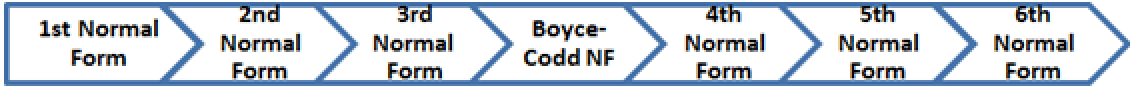
- Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships

Database Normal Forms

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)
- 6NF (Sixth Normal Form)

Database Normal Forms

- 1NF (First Normal Form)
- 2NF (Second Normal Form)
- 3NF (Third Normal Form)
- BCNF (Boyce-Codd Normal Form)
- 4NF (Fourth Normal Form)
- 5NF (Fifth Normal Form)
- 6NF (Sixth Normal Form)



1st Normal Form

2nd Normal Form

3rd Normal Form

Boyce-Codd NF

4th Normal Form

5th Normal Form

6th Normal Form

Database Normalization With Examples

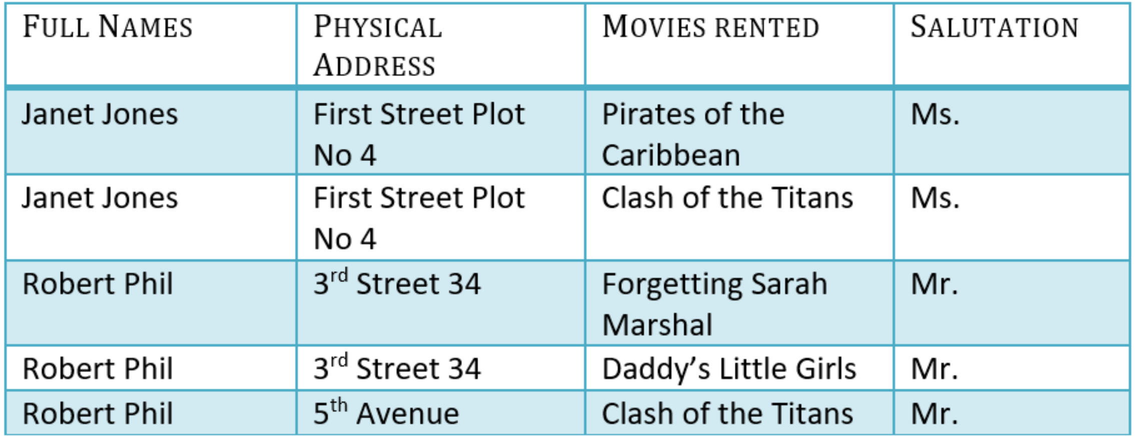
- Database Normalization Example can be easily understood with the help of a case study. Assume, a video library maintains a database of movies rented out. Without any normalization in database, all information is stored in one table as shown below. Let's understand Normalization database with normalization example with solution:

Database Normalization With Examples

Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy’s Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

1NF (First Normal Form) Rules

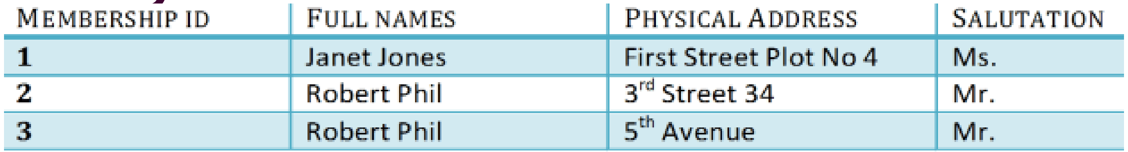
- Each table cell should contain a single value.
- Each record needs to be unique.



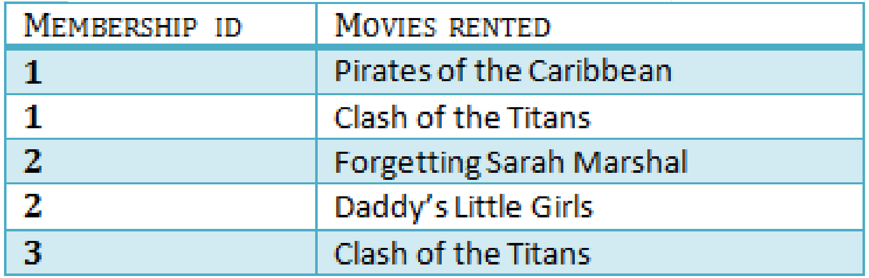
Full Names	Physical Address	Movies rented	Salutation
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5th Avenue	Clash of the Titans	Mr.

2NF (Second Normal Form) Rules

- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key that does not functionally dependant on any subset of candidate key relation



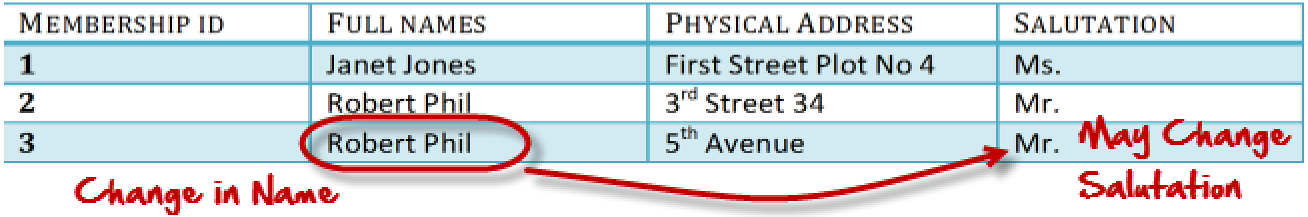
MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.



MEMBERSHIP ID	MOVIES RENTED
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans

What are transitive functional dependencies?

- A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change



MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
1	Janet Jones	First Street Plot No 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

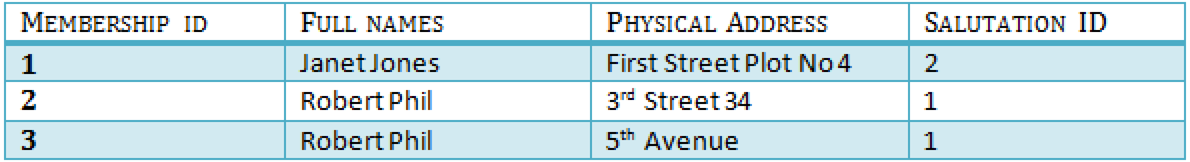
Change in Name

May Change Salutation

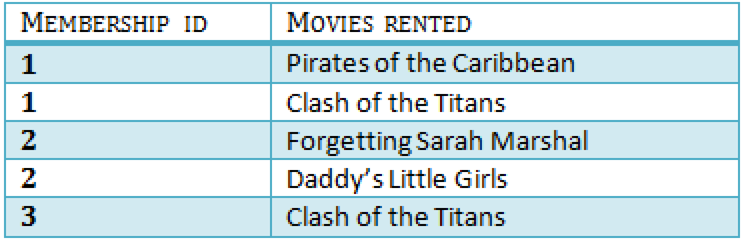
3NF (Third Normal Form) Rules

- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies

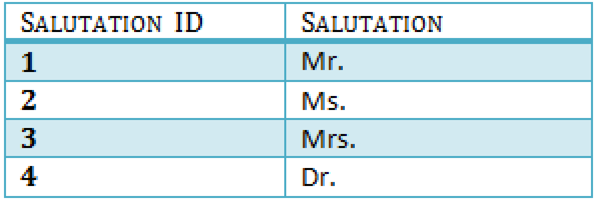
3NF Example



Membership ID	Full names	Physical Address	Salutation ID
1	Janet Jones	First Street Plot No 4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1



Membership ID	Movies rented
1	Pirates of the Caribbean
1	Clash of the Titans
2	Forgetting Sarah Marshal
2	Daddy's Little Girls
3	Clash of the Titans



Salutation ID	Salutation
1	Mr.
2	Ms.
3	Mrs.
4	Dr.

BCNF (Boyce-Codd Normal Form)

- Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one Candidate Key

4NF (Fourth Normal Form) Rules

- If no database table instance contains two or more, independent and multivalued data describing the relevant entity, then it is in 4th Normal Form.

5NF (Fifth Normal Form) Rules

- A table is in 5th Normal Form only if it is in 4NF and it cannot be decomposed into any number of smaller tables without loss of data.

6NF (Sixth Normal Form) Proposed

- 6th Normal Form is not standardized, yet however, it is being discussed by database experts for some time. Hopefully, we would have a clear & standardized definition for 6th Normal Form in the near future

What are Keys in DBMS

- Summary. Eight types of key in DBMS are Super, Primary, Candidate, Alternate, Foreign, Compound, Composite, and Surrogate Key. A super key is a group of single or multiple keys which identifies rows in a table. Primary Key never accept null values while a foreign key may accept multiple null values.

Example

Employee ID	FirstName	LastName
11	Andrew	Johnson
22	Tom	Wood
33	Alex	Hale

Types of Keys in DBMS

- Super Key
- Primary Key
- Candidate Key
- Alternate Key
- Foreign Key
- Compound Key
- Composite Key
- Surrogate Key

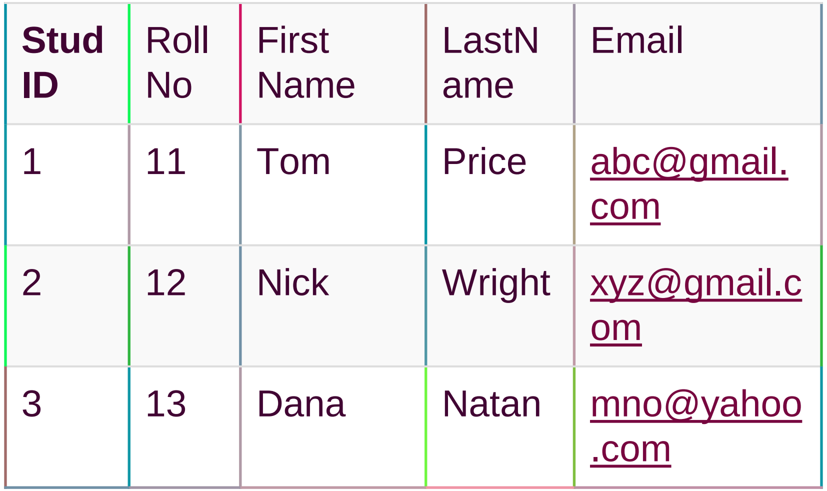
What is the Super key

- A super key is a group of single or multiple keys which identifies rows in a table
- Example EmpSSN and EmpNum name are superkeys.

EmpSSN	EmpNum	Empname
9812345098	AB05	Shown
9876512345	AB06	Roslyn
199937890	AB07	James

What is a Primary Key

- Primary key is a column or group of columns in a table that uniquely identify every row in that table.
- Example
- StudID is P.Key



Stud ID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

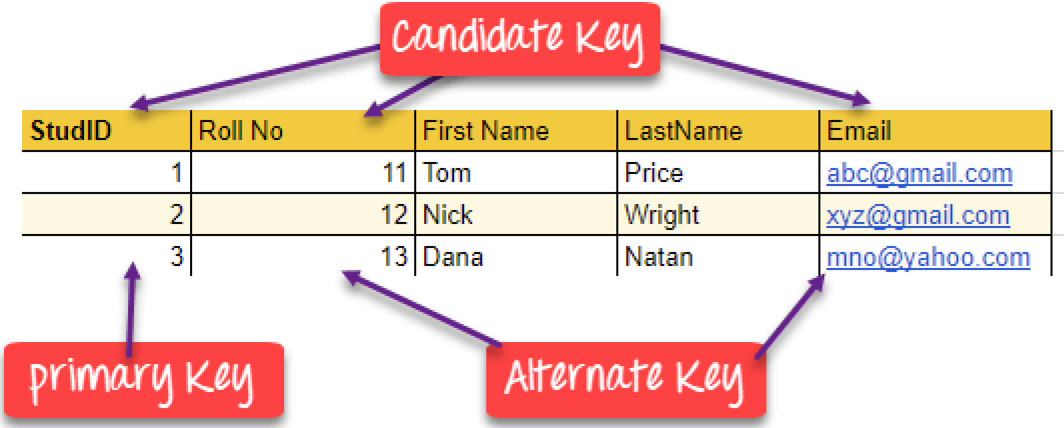
What is Alternate Key

- Alternate Key is a column or group of columns in a table that uniquely identify every row in that table.
- Email A.key

StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

Candidate Key

- Candidate Key is a set of attributes that uniquely identify tuples in a table. Candidate Key is a super key with no repeated attributes



Candidate Key

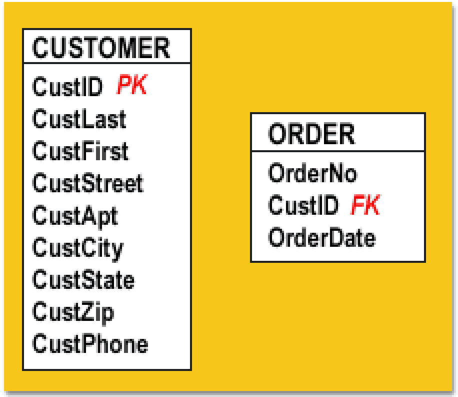
StudID	Roll No	First Name	LastName	Email
1	11	Tom	Price	abc@gmail.com
2	12	Nick	Wright	xyz@gmail.com
3	13	Dana	Natan	mno@yahoo.com

primary Key

Alternate Key

Foreign Key

- Foreign Key is a column that creates a relationship between two tables. The purpose of Foreign keys is to maintain data integrity and allow navigation between two different instances of an entity.



CUSTOMER

CustID *PK*
CustLast
CustFirst
CustStreet
CustApt
CustCity
CustState
CustZip
CustPhone

ORDER

OrderNo
CustID *FK*
OrderDate

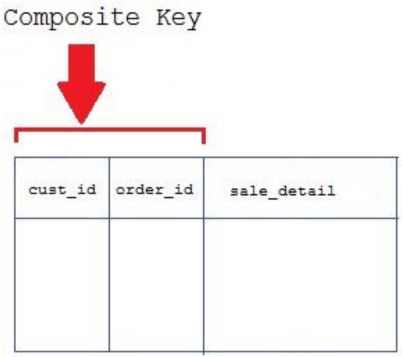
Compound Key

- Compound Key has two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database.
- compound key of Order ID and Product ID

OrderNo	PorductID	Product Name	Quantity
B005	JAP102459	Mouse	5
B005	DKT321573	USB	10

Composite Key

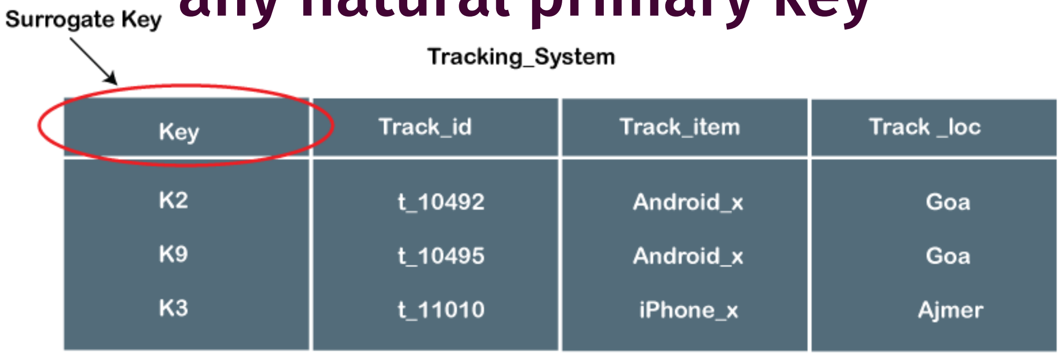
- Composite Key is a combination of two or more columns that uniquely identify rows in a table. The combination of columns guarantees uniqueness, though individual uniqueness is not guaranteed



cust_id	order_id	sale_detail

Surrogate Key

- Surrogate Key An artificial key which aims to uniquely identify each record is called a surrogate key. These kind of key are unique because they are created when you don't have any natural primary key



Surrogate Key

Tracking_System

Key	Track_id	Track_item	Track _loc
K2	t_10492	Android_x	Goa
K9	t_10495	Android_x	Goa
K3	t_11010	iPhone_x	Ajmer

Define table

- Tables are database objects that contain all the data in a database. In tables, data is logically organized in a row-and-column format similar to a spreadsheet. Each row represents a unique record, and each column represents a field in the record

Define tuple

- (1) In a relational database, a tuple is one record (one row). ... Typically separated by commas, the values may be parameters for a function call or a set of data values for a database

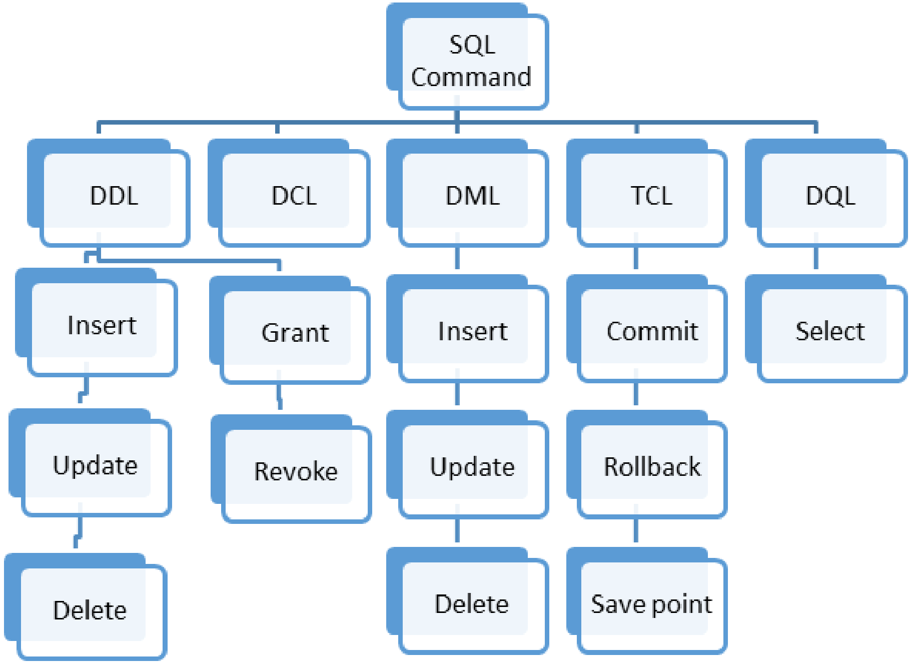
What is SQL

- SQL is the standard language for dealing with Relational Databases. SQL can be used to insert, search, update, and delete database records. SQL can do lots of other operations, including optimizing and maintenance of databases.

Types of SQL Statements

- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- Data Control Language (DCL)
- Transaction Control Language (TCL)
- Data Query Language (DQL

Types of SQL Statements



Data Definition Language

- Data Definition Language helps you to define the database structure or schema. Let's learn about DDL commands with syntax.
- Five types of DDL commands in SQL are

Data Manipulation Language

- Data Manipulation Language (DML) allows you to modify the database instance by inserting, modifying, and deleting its data. It is responsible for performing all types of data modification in a database.
- important DML commands in SQL
- INSERT >UPDATE >DELETE

Data Control Language

- DCL (Data Control Language) includes commands like GRANT and REVOKE, which are useful to give "rights & permissions." Other permission controls parameters of the database system.
- Commands that come under DCL:
- Grant
- Revoke

Transaction control language

- Transaction control language or TCL commands deal with the transaction within the database.

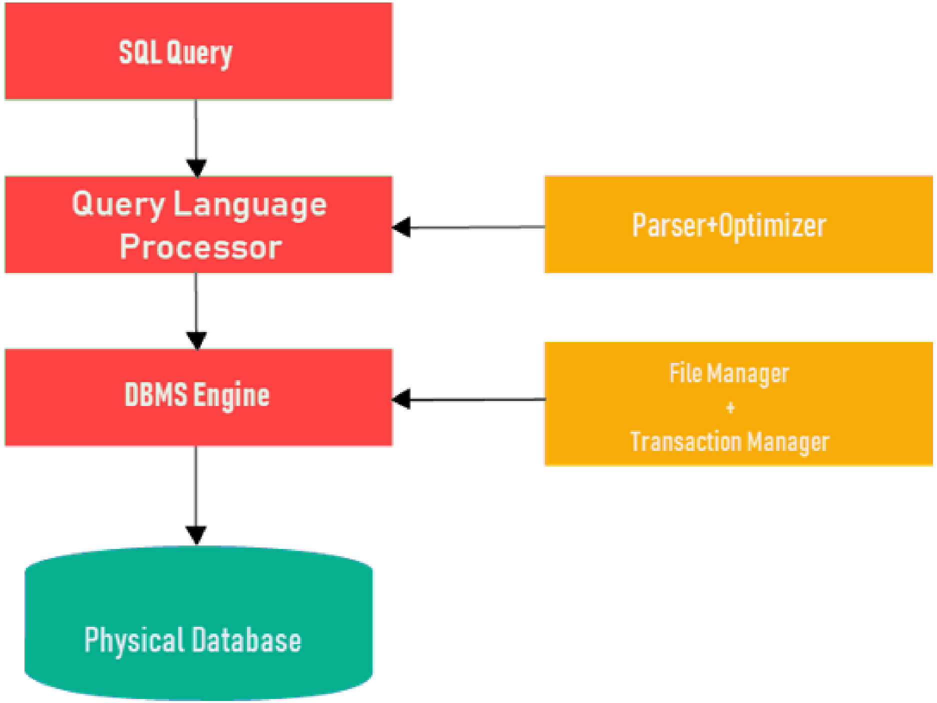
Data Query Language

- Data Query Language (DQL) is used to fetch the data from the database. It uses only one command:

List of SQL Commands

- CREATE - defines the database structure schema
- INSERT - inserts data into the row of a table
- UPDATE - updates data in a database
- DELETE - removes one or more rows from a table
- SELECT - selects the attribute based on the condition described by the WHERE clause
- DROP - removes tables and databases

SQL Process


SQL Query
Query Language Processor
Parser+Optimizer
DBMS Engine
File Manager
+
Transaction Manager
Physical Database

What is NoSQL

- NoSQL is an upcoming category of Database Management Systems. Its main characteristic is its non-adherence to Relational Database Concepts. NoSQL means "Not only SQL". The concept of NoSQL databases grew with internet giants such as Google, Facebook, Amazon etc. who deal with gigantic volumes of data.

What is Functional Dependency

- Functional Dependency (FD) is a constraint that determines the relation of one attribute to another attribute in a Database Management System (DBMS). Functional Dependency helps to maintain the quality of data in the database. It plays a vital role to find the difference between good and bad database design.

Types of Functional Dependencies

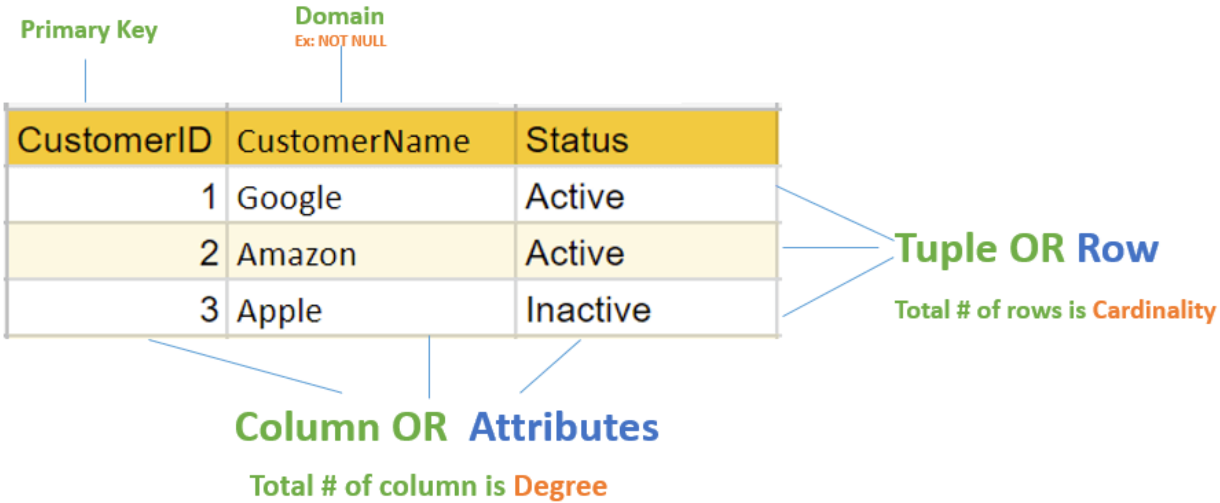
- Multivalued Dependency
- Trivial Functional Dependency
- Non-Trivial Functional Dependency
- Transitive Dependency

What is Relational Model

- Relational Model (RM) represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

What is Relational Model

Table also called Relation



Primary Key

Domain

Ex: NOT NULL

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Tuple OR Row

Total # of rows is Cardinality

Column OR Attributes

Total # of column is Degree

Subquery in DBMS

- Answer: A Subquery is also known as a nested query i.e. a query written inside some query. When a Subquery is executed for each of the rows of the outer query then it is termed as a Correlated Subquery.

Three Level Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture. This framework is used to describe the structure of a specific database system. The three schema architecture is also used to separate the user applications and physical database.

Three Level Architecture

- The three schema architecture is also called ANSI/SPARC architecture or three-level architecture. This framework is used to describe the structure of a specific database system. The three schema architecture is also used to separate the user applications and physical database.

Three Level Architecture

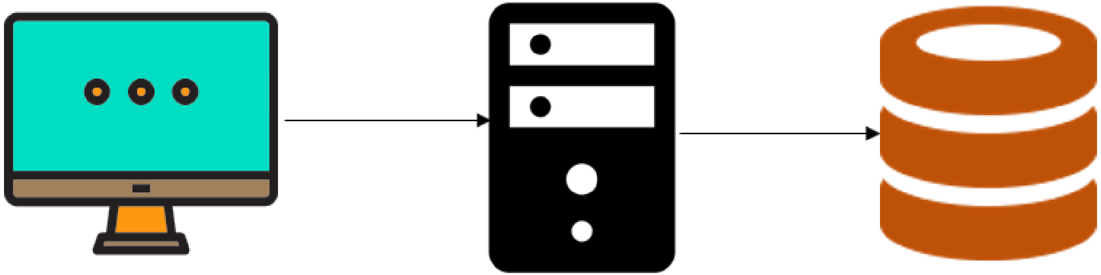
- Internal Level/Schema
- Conceptual Schema/Level
- External Schema/Level

Three Level Architecture Diagram

External Level/ View Level

Conceptual Level/ Logical Level

Internal Level



Client

Server

Database

Three Tier Architecture

Join In SQL

- A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

Joins Types

- Inner Join: This type of join is used to fetch the data among the tables which are common in both the tables.
- Left Join: This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.

Joins Types

- Right Join: This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
- Full Join: This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values

Thanks!

Any questions?

You can find me at:

- @username
- user@mail.me