

## **DBMS & SQL NOTES**

**Database:** A database is a collection of related data which represents some aspect of the real world. A database system is designed to be built and populated with data for a certain task.

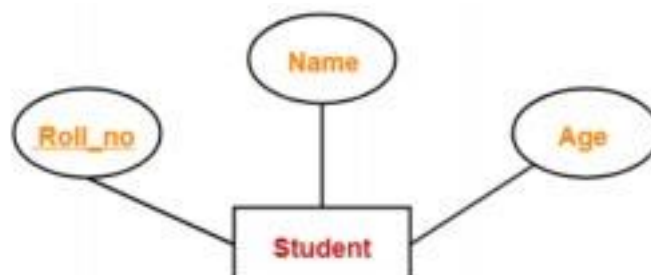
**Database Management System (DBMS)** is a software for storing and retrieving users' data while considering appropriate security measures. It consists of a group of programs which manipulate the database. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

Database management systems were developed to handle the following difficulties of typical File-processing systems supported by conventional operating systems.

1. Data redundancy and inconsistency
2. Difficulty in accessing data
3. Data isolation – multiple files and formats
4. Integrity problems
5. Atomicity of updates
6. Concurrent access by multiple users
7. Security problems

### **ER diagram:**

- ER diagram or **Entity Relationship diagram** is a conceptual model that gives the graphical representation of the logical structure of the database.
- It shows all the constraints and relationships that exist among the different components.
- An ER diagram is mainly composed of following three components- Entity Sets, Attributes and Relationship Set.



- Roll\_no is a primary key that can identify each entity uniquely.
- Thus, by using a student's roll number, a student can be identified uniquely.

### **Entity Set:**

An entity set is a set of the same type of entities.

- **Strong Entity Set:**

- A strong entity set is an entity set that contains sufficient attributes to uniquely identify all its entities.
- In other words, a primary key exists for a strong entity set.
- Primary key of a strong entity set is represented by underlining it.

- **Weak Entity Set:**

- A weak entity set is an entity set that does not contain sufficient attributes to uniquely identify its entities.
- In other words, a primary key does not exist for a weak entity set.
- However, it contains a partial key called a discriminator. ◦ Discriminator can identify a group of entities from the entity set.
- Discriminator is represented by underlining with a dashed line.

### **Relationship:**

A relationship is defined as an association among several entities.

- **Unary Relationship Set** - Unary relationship set is a relationship set where only one entity set participates in a relationship set.
- **Binary Relationship Set** - Binary relationship set is a relationship set where two entity sets participate in a relationship set.
- **Ternary Relationship Set** - Ternary relationship set is a relationship set where three entity sets participate in a relationship set.
- **N-ary Relationship Set** - N-ary relationship set is a relationship set where 'n' entity sets participate in a relationship set

- **Cardinality Constraint:**

Cardinality constraint defines the maximum number of relationship instances in which an entity can participate.

- **One-to-One Cardinality** - An entity in set A can be associated with at most one entity in set B. An entity in set B can be associated with at most one entity in set A.
- **One-to-Many Cardinality** - An entity in set A can be associated with any number (zero or more) of entities in set B. An entity in set B can be associated with at most one entity in set A.

- **Many-to-One Cardinality** - An entity in set A can be associated with at most one entity in set B. An entity in set B can be associated with any number of entities in set A.
- **Many-to-Many Cardinality** - An entity in set A can be associated with any number (zero or more) of entities in set B. An entity in set B can be associated with any number (zero or more) of entities in set A.

### Attributes:

Attributes are the descriptive properties which are owned by each entity of an Entity Set.

#### **Types of Attributes:**

- **Simple Attributes** - Simple attributes are those attributes which cannot be divided further. Ex. Age
- **Composite Attributes** - Composite attributes are those attributes which are composed of many other simple attributes. Ex. Name, Address
- **Multi Valued Attributes** - Multi valued attributes are those attributes which can take more than one value for a given entity from an entity set. Ex. Mobile No, Email ID
- **Derived Attributes** - Derived attributes are those attributes which can be derived from other attribute(s). Ex. Age can be derived from DOB.
- **Key Attributes** - Key attributes are those attributes which can identify an entity uniquely in an entity set. Ex. Roll No.

### Constraints:

Relational constraints are the restrictions imposed on the database contents and operations. They ensure the correctness of data in the database.

- **Domain Constraint** - Domain constraint defines the domain or set of values for an attribute. It specifies that the value taken by the attribute must be the atomic value from its domain.
- **Tuple Uniqueness Constraint** - Tuple Uniqueness constraint specifies that all the tuples must be necessarily unique in any relation.
- **Key Constraint** - All the values of the primary key must be unique. The value of the primary key must not be null.
- **Entity Integrity Constraint** - Entity integrity constraint specifies that no attribute of primary key must contain a null value in any relation.
- **Referential Integrity Constraint** - It specifies that all the values taken by the foreign key must either be available in the relation of the primary key or be null.

### Closure of an Attribute Set:

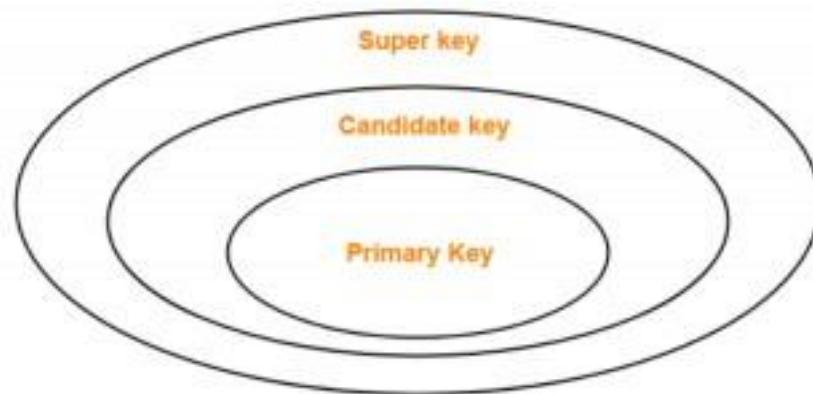
The set of all those attributes which can be functionally determined from an attribute set is called a closure of that attribute set.

### Keys:

A key is a set of attributes that can identify each tuple uniquely in the given relation.

### Types of Keys:

- **Super Key** - A superkey is a set of attributes that can identify each tuple uniquely in the given relation. A super key may consist of any number of attributes.
- **Candidate Key** - A set of minimal attribute(s) that can identify each tuple uniquely in the given relation is called a candidate key.
- **Primary Key** - A primary key is a candidate key that the database designer selects while designing the database. Primary Keys are unique and NOT NULL.



- **Alternate Key** - Candidate keys that are left unimplemented or unused after implementing the primary key are called as alternate keys.
- **Foreign Key** - An attribute 'X' is called as a foreign key to some other attribute 'Y' when its values are dependent on the values of attribute 'Y'. The relation in which attribute 'Y' is present is called as the referenced relation. The relation in which attribute 'X' is present is called as the referencing relation.
- **Composite Key** - A primary key composed of multiple attributes and not just a single attribute is called a composite key.
- **Unique Key** - It is unique for all the records of the table. Once assigned, its value cannot be changed i.e. it is non-updatable. It may have a NULL value.

### Functional Dependency:

In any relation, a functional dependency  $\alpha \rightarrow \beta$  holds if- Two tuples having same value of attribute  $\alpha$  also have same value for attribute  $\beta$ .

## Types of Functional Dependency:

- **Trivial Functional Dependencies –**

Thus, if RHS of a functional dependency is a subset of LHS, then it is called a  $\subseteq$ .  
○ A functional dependency  $X \rightarrow Y$  is said to be trivial if and only if  $Y \subseteq X$ .  
○ trivial functional dependency.

- **Non-Trivial Functional Dependencies –**

- A functional dependency  $X \rightarrow Y$  is said to be non-trivial if and only if  $Y \not\subseteq X$ .
- that is not a part of LHS, then it is called a non-trivial functional dependency.

Thus, if there exists at least one attribute in the RHS of a functional dependency  $\nsubseteq$