CTH EDUCATION

4TH SEMESTER - CSE

DBMS MODEL QUESTIONS FOR SBTE

1. Define data. What are the characteristics of data ?
2. What is Database ? Why we need database ?
3. What is DBMS ? What are the advantage of DBMS ?
4. Define DBMS Architecture ? Also explain its type.
5. Explain in details different views of DBMS architecture.
6. Differentiate between logical & physical data independence.
7. Define RDBMS. What are the advantage of RDBMS ?
8. Differentiate between DBMS and RDBMS.
9. What is relational model ? Explain in details.
10. Write the difference between schema and instances.
11. Define ACID properties in relational database management system.
12. What is key ? Explain different types of key in brief.
13. Explain the term super key and candidate key with examples.
14. Explain entity integrity constraint and referential integrity constraint.
15. What are the codd's rules for relational database system explain in brief ?
16. Explain the following terms :
    a. Entity
    b. Tupple
    c. Attributes
    d. Degree
    e. Cardinality
17. What is normalization in DBMS ? Why we need normalization ?
18. Write the difference between 3NF and BCNF.
19. Explain fourth normal form (4NF) and fifth normal form(5NF).
20. What is functional dependency in DBMS ?
21. Explain trivial and non-trivial functional dependency with example.
22. Explain in brief Inference rule/Armstrong's Axioms.
23. Describe the E-R model with an example ? Also write its advantage.
24. Explain different components of E-R model in brief.
25. Discuss E-R modelling symbols in details.
26. Draw an ER diagram for a company.

CTH EDUCATION SAHARSA ALOK SIR

27. Construct an ER diagram of your college and establish relationship between entities.
28. What is MariaDB ? Write the features of MariaDB.
29. Explain in details about data types of MariaDB.
30. Define the term SQL. Write the advantage of SQL.
31. Write the structure of SQL SELECT statements.
32. List and explain aggregate function used in SQL with examples.
33. Explain in brief about various SQL statements(DDL, DML, DCL & TCL).
34. Explain string function and date function in details.
35. Explain Grant and Revoke commands in SQL.
36. Write the structure of SQL SELECT statements.
37. Explain in brief about various SQL statements.
38. Write syntax of SQL Order by and Group by clauses.
39. Write short notes on nested query.
40. What is a join ? Discuss different types of joins.
41. Differentiate between natural join and inner join.
42. What is NoSQL Database ? Explain its type.
43. Differentiate between relational database and NoSQL database.
44. Write the difference between SQL and NoSQL Database.
45. What is MongoDB ? What are the features of MongoDB ?
46. Difference between RDBMS and MongoDB.
47. How to choose the right database for your service.

CTH EDUCATION JAMARSA ALOK SIR

Unit -01 : An Overview of the Database Management System

- What is database ?
- Why database ?
- Database system,
- Database management system (DBMS),
- Advantage of DBMS.

Questions to be discussed :

1. Define data. What are the characteristics of data ?
2. What is Database ? Why we need database ?
3. What is DBMS ? What are the advantage of DBMS ?

Data : Known fact about any entity.

Example :

R_No. , Name,

Record : Collection of interrelated data.

Exp :

R-No. Name S.No. Marks

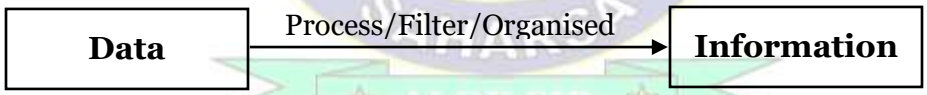
Databade(DB): Collection of record.

DBMS: Software to create, manipulate & delete database.

id	ISSN-L	ISSNs	PublisherId	Journal_Title
0	2056-9890	2056-9890	1	Acta Crystallographica Section E Crystallographic Communications
1	2077-0472	2077-0472	2	Agriculture
2	2073-4395	2073-4395	2	Agronomy
3	2076-2615	2076-2615	2	Animals
4	2076-3417	2076-3417	2	Applied Sciences
5	2306-5354	2306-5354	2	Bioengineering
6	2079-7737	2079-7737	2	
7	2079-6374	2079-6374	2	

What is Data ?

- Data is a set of characters that is collects and translated for some purpose.
- In simple words, data can be facts related to any object.
- Example, your name, age, height, weight, etc. are some data related to you.
- A picture, image, file, pdf, etc. can also be considered data.
- The main reason behind collection of data is analysis.
- There are multiple types of data :
  - Text data(A to Z)
  - Number data(0 to 9)
  - Alphanumeric data(Symbols)
  - Image data(JPEG, JPG, PNG)
  - Audio(MP3)
  - Video(MP4)
- Computer data is processed by CPU and stored in hard disk or other storage device.
- In computer data is stored in the form of binary digits(0 or 1).
- Data is a latin word that's meaning is "something given".
- All software is divided into two major categories : programs and data.
- We already know about data, and program is a collection of instructions for manipulating data.



Characteristics of Data in a Database ?

- The data in the database possess several characteristics :
  - Consistent,
  - Accurate,
  - Complete
  - Secured,
  - Centrally managed and shared among multiple applications.

What is Information ?

- It is a organized or filtered data that has some meaningful value.
- Information is also the processed data used to make decisions and take action.
- Processed data must meet the following criteria :
  - Accuracy : The information must be accurate.
  - Completeness : The information must be complete.
  - Timeliness : The information must be available when it's needed.

DATA	INFORMATION
Data is collection of facts, which it self have no meaning.	Meaningful data known as information.
Data are text and numerical values.	Information is refined form of actual data.
Data doesn't depend on Information.	While Information depend on Data.
Data does not directly helps in decision.	Information directly helps in decision.

What is Database ?

- A database is a systematic collection of data.
- They support electronic storage and manipulation of data.
- Databases make data management easy.
- The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data.
- There are many databases available like MySQL, Sybase, Oracle, MongoDB, PostgreSQL, SQL Server, etc.
- Databases are managed by the database management system (DBMS).

Example :

- An online telephone directory uses a database to store data of people, phone numbers, and other contact details.
- A model that checks the availability of rooms in a hotel.

Types of Databases

- Distributed databases:
- Relational databases:
- Object-oriented databases:
- Centralized database:
- Open-source databases:
- Cloud databases:
- Personal database:
- Hierarchical:

Why use a database ?

- Database is used to store very large numbers of records efficiently.
- It is very quick and easy to find information.
- It is easy to add new data and to edit or delete old data.
- Data can be searched & sorted easily.
- More than one person can access the same database at the same time - multi-access.
- Security may be better than in paper files.

What is a Database Management System (DBMS)?

- DBMS is a software which is used to create, manipulate & delete database.
- It also helps to control access to the database.
- Database Management System had first implemented in the 1960s.
- Charles Bachman's Integrated Data Store (IDS) is said to be the first DBMS in history.
- Database management system is a software which is used to manage the database.
- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- It provides protection and security to the database.
- In the case of multiple users, it also maintains data consistency.

DBMS	Type	Written in
MySQL	RDBMS	C and C++
Oracle	Multi-model DBMS	Assembly language, C and C++
Microsoft SQL Server	Linux and Windows	C and C++
IBM DB2	RDBMS	Assembly, C, C++ and Java
MariaDB	RDBMS	Bash, C, C++, and Perl
MongoDB	Document-oriented database	C++, JavaScript and Python

Characteristics of DBMS

- DBMS contains automatic backup and recovery procedures.
- It contains ACID properties which maintain data in a healthy state in case of failure.
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data.
- It is used to provide security of data.
- It can view the database from different viewpoints according to the requirements of the user.

Advantages of DBMS

- **Controls database redundancy :** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
- **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
- **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
- **Reduce time:** It reduces development time and maintenance need.
- **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
- **multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

Unit -02 : An Architecture of the Database system

- Three level of architecture,
  - logical view,
  - physical view,
  - conceptual view,
- Logical data independence,
- Physical data independence.

Questions to be discussed :

1) Define DBMS Architecture ? Also explain its type.
2) Expalain in details different views of DBMS architecture.
3) Differentiate between logical & physical data independence.

CTH EDUCATION SAHARSA ALOK SIR

What is DBMS Architecture ?

- A Database Architecture is a representation of DBMS design.
- It helps to design, develop, implement, and maintain the database management system.
- A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, and altered.
- It also helps to understand the components of a database.
- A Database stores critical information and helps access data quickly and securely.
- Therefore, selecting the correct Architecture of DBMS helps in easy and efficient data management.
- DBMS architecture depends upon how users are connected to the database.

Types of DBMS Architecture :

There are three types of DBMS architecture :

1) 1-Tier Architecture
2) 2-Tier Architecture
3) 3-Tier Architecture

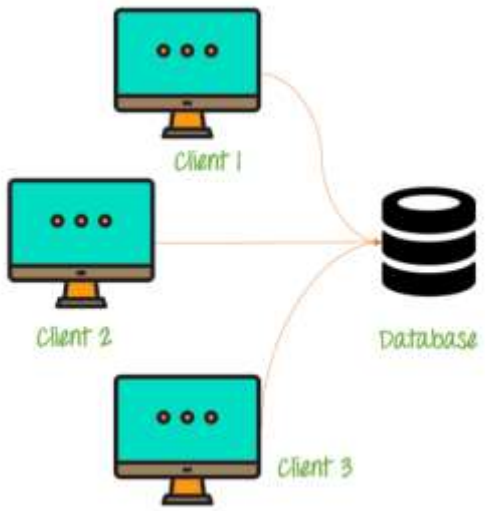
One-Tier Architecture :

- 1 - Tier Architecture in DBMS is the simplest architecture of Database.
- In One- Tier Architecture the client, server, and Database all present on the same machine.
- In One- Tier Architecture the database is directly available to the user.
- Any changes done here will directly be done on the database itself.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

Example : Anytime you install a Database in your system and access it to practice.

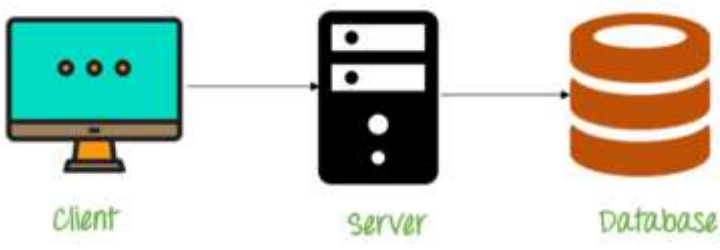
Two-tier architecture :

- In 2-Tier Architecture the presentation layer runs on a client and data is stored on a server.
- Two tier architecture provides security to the DBMS, data is not exposed to the user directly.
- It also provides direct and faster communication.
- The two-tier architecture is similar to a basic client-server model.
- The server side is responsible for providing data and transaction management functionalities.
- On the client-side, the user interfaces and application programs are run.
- An advantage of this type is that maintenance and understanding are easier.
- This model gives poor performance when there are a large number of users.



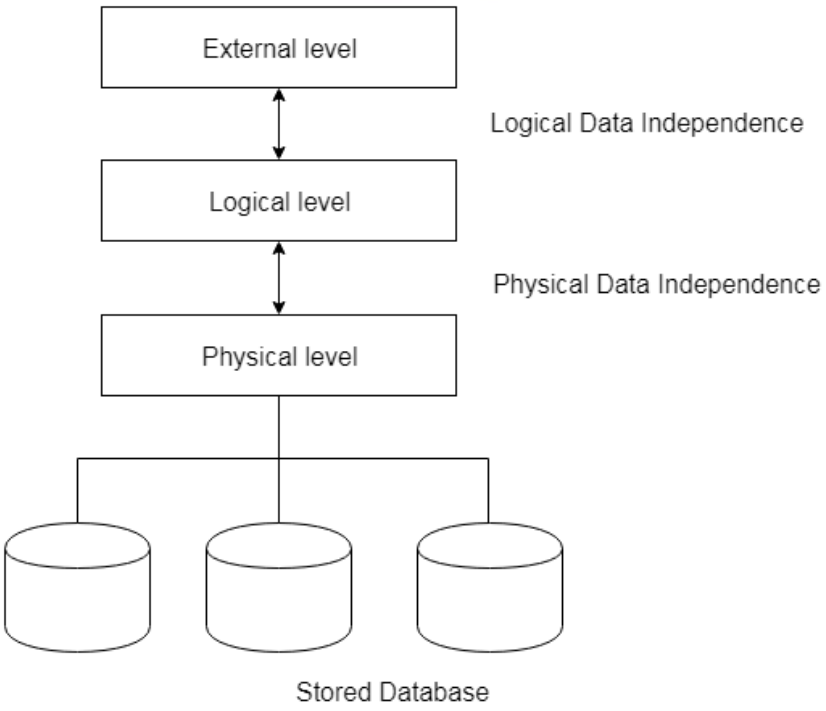
Three Tier architecture :

- 3-Tier architecture design is an extension of the 2-tier client-server architecture.
- In this type, there is another layer between the client and the server.
- The client does not directly communicate with the server.
- In this client interacts with an application server which further communicates with the database system and then the query processing takes place.
- This intermediate layer acts as a medium for the exchange of partially processed data between server and client.
- This type of architecture is used in the case of large web applications.



Level of Database architecture :

- Database systems comprise complex data-structures.
- To make it efficient in terms of retrieval of data, and reduce complexity, developers use abstraction.
- Abstracion means hide irrelevant details from the users.
- The main purpose of data abstraction is to achieve data independence & save time and cost when the database is modified.
- There are mainly **3** levels of data abstraction :
  1. Physical/Internal
  2. Conceptual
  3. External



Physical Level :

- This is the lowest level of data abstraction.
- It tells us how the data is actually stored in memory.
- The access methods like sequential or random access is used for the data access.
- Suppose we need to store the details of an employee.
- Blocks of storage and the amount of memory used for these purposes are kept hidden from the user.

Logical/Conceptual Level :

- This level comprises the information that is actually stored in the database in the form of tables.
- It also stores the relationship among the data entities in relatively simple structures.
- At this level, the information available to the user at the view level is unknown.
- We can store the various attributes of an employee and relationships.

Example : The manager can also be stored the details of employee.

External Level :

- This is the highest level of abstraction.
- Only a part of the actual database is viewed by the users.
- This level exists to ease the accessibility of the database by an individual user.
- Users view data in the form of rows and columns.
- Tables and relations are used to store data.
- Multiple views of the same database may exist.
- Users can just view the data and interact with the database, storage and implementation details are hidden from them.

What is Data Independence of DBMS?

- It is defined as a property of DBMS that helps you to change the database schema at one level without affecting another level.
- Data independence provides data transparency to the DBMS.
- There are two levels of data independence :
  1) Physical Data Independence
  2) Logical Data Independence

Physical Data Independence :

- It is the characteristic to modify physical schema without any changes to the logical schema.
- It helps you to separate conceptual levels from the physical level.
- Compared to Logical Independence, it is easy to achieve.
- With Physical independence, you can easily change the physical storage structures.

- Due to Physical independence, any of the below change will not affect the conceptual layer.
  - Using a new storage device like Hard Drive or Magnetic Tapes
  - Switching to different data structures.
  - Changing the access method.
  - Modifying indexes.
  - Change of Location of Database from say C drive to D Drive

Logical Data Independence :

- It is the characteristic to modify the logical schema without affecting the external schema.
- It helps you to separate conceptual levels from the external level.
- Compared to Physical Data independence, it is challenging to achieve.
- Due to Logical independence, any of the below change will not affect the external layer.
  - Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
  - Merging two records into one
  - Breaking an existing record into two or more records

Logical Data Independence	Physical Data Independence
It is mainly concerned with the structure or changing the data definition.	Mainly concerned with the storage of the data.
It is difficult as the retrieving of data is mainly dependent on the logical structure of data.	It is easy to retrieve.
Compared to Physical independence it is difficult to achieve logical data independence.	Compared to Logical Independence it is easy to achieve physical data independence.
Concerned with conceptual schema	Concerned with internal schema
Example: Add/Modify/Delete a new attribute	Example: change in compression techniques, hashing algorithms, storage devices, etc

Unit-03 : Relational Database Management System (RDBMS)

- Introduction to RDBMS,
- RDBMS terminology,
- Relational model,
- Base tables,
- Keys,
  - primary key,
  - foreign key,
- Constraints,
- Code rules.

Unit -03 : Relational DBMS :

1. Define RDBMS. What are the advantage of RDBMS ?
2. Differentiate between DBMS and RDBMS.
3. What is relational model ?
4. Write the difference between schema and instances.
5. Define ACID properties in relational database management system.
6. What is key ? Explain different types of key in brief.
7. Explain the term super key and candidate key with examples.
8. Explain entity integrity constraint and referential integrity constraint.
9. What are the codd's rules for relational database system explain in brief ?
10. Explain the following terms :
    a. Entity
    b. Tupple
    c. Attributes
    d. Degree
    e. Cardinality

What is RDBMS ?

- RDBMS stands for Relational Database Management Systems.
- It is called RDBMS because it is based on relational model introduced by E.F. Codd.
- It is a software used to store and retrieve data from the table.
- In other words RDBMS stores information in the form of rows & columns.
- It is a advanced version of a DBMS that allows access to data in a more efficient way.
- All modern DBMS like SQL, ORACLE, My-SQL and MS Access are based on RDBMS.

Advantage of RDBMS :

- Easy structure : Due to tabular format it is easy to understand by user.
- Data integrity : It maintan the data consistency, accuracy & stability by integrity constarints.
- Easy to access : There is no pattern to access data.
- High security : Table can be tagged as confidential.
- High performance : Due to the use of Indexes, searching make it faster.
- It provides better backup and recovery procedures.
- It provides multiple interfaces.
- Multiple users can access the database which is not possible in DBMS.
- It follow ACID properties

Difference between DBMS and RDBMS :

DBMS	RDBMS
It store data in the form of file.	It store data in a tabular form.
No connection between data.	Data in the form of a table are linked together.
DBMS supports a single user.	RDBMS supports multiple users.
Data redundancy is common.	Here, redundancy of data is reduced.
It does not support distributed database.	RDBMS supaports distributed database.
It is designed to handle small amount of data	It is designed to handle large amount of data.
Example : XML, Microsoft Access.	**Example** : Oracle, SQL Server.

RDBMS Terminology :

Relational terms	Equivalent terms
Relation	Table
Tupple	Row, Record
Cardinality	Number of rows
Attribute	Column, field
Degree	Number of columns
Domain	Set of legal values
Primary key	Unique identifier

What is table ?

- The RDBMS database uses tables to store data.
- A table is a collection of related data entries and contains rows and columns to store data.
- A table is also called relation.
- A table is the simplest example of data storage in RDBMS.

Example :

ID	Name	AGE	COURSE
1	Ajeet	24	CSE
2	aryan	20	ECE
3	Mahesh	21	MECH
4	Rajan	22	ELE
5	Vimal	26	CIVIL

What is record/tupple ?

- A row of a table is also called record.
- It contains the specific information of each individual entry in the table.
- Each rowof a table is known as tupple.
- It is a horizontal entity in the table.

  Example : The above table contains 5 records.

1	Ajeet	24	CSE

What is column/attribute ?

- A column is a vertical entity in the table.
- Each column of a table is called attribute.
- A column is also known as field of a table.
- It contains all information associated with a specific field in a table.

Example: "name" is a column in above table which contains information about student's name.

Ajeet	
Aryan	
Mahesh	
Ratan	
Vimal	

Cardinality :

- The number of row in a table is called as cardinality.
- A Cardinality is defined as the number of tupple in a relation.
- A tupple in a relation can changes like added or deleted.
- The cardinality is the extension of the relation at any given moment.
- In the above table the cardinality is 5.

Degree :

- The number of columns associated with a relation is called degree.
- In the above table the degre is 4.

Student_id	Student_Name	Branch
101	ABC	CSE
102	XYZ	CSE
103	MNP	CSE

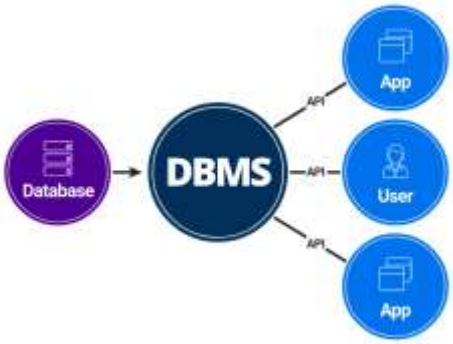
Row OR Tupple

Column OR attributes

Domain : The set of allowable values in a field is called domain.

DATABASE : A database is a collection of organised or arranged data that can be easily accessed, updated/modified or controlled.

DBMS : DBMS is a software used to identify, manage, and create a database.



Types of Database model :

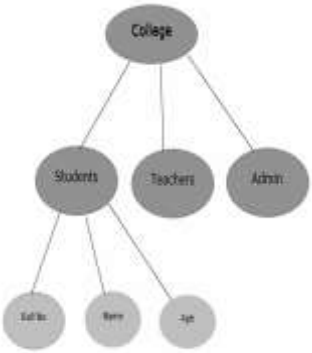
There are four types of DBMS based on data model :

1. Relational database.
2. Object oriented database.
3. Hierarchical database.
4. Network database.

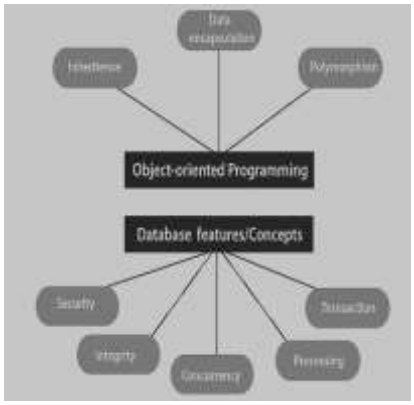
Relational Model :

Std ID	Name	City
201	Bablu	Patna
204	Lucky	Chennai
205	Pinky	Bangalore

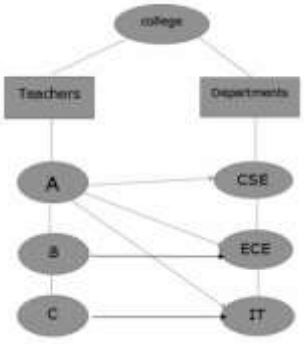
Hierarchical DBMS



Object-Oriented Model



Network Model



What is relational model ?

- The relational model is the theoretical basis of relational databases.
- The relational model of data is based on the concept of relations.
- It represents how data is stored in relational databases.
- It stores data in the form of relations (tables).
- In relational model data is stored in multiple tables where tables are related to each other using primary keys and foreign keys and indexes.
- It helps to fetch data faster using SQL query.
- Relational Model was proposed by E.F. Codd in 1970.

Consider a relation STUDENT with attributes ROLL_NO, NAME, BRANCH and AGE.

ROLL_NO	NAME	BRANCH	AGE
101	RAM	CSE	18
102	RAMESH	CSE	18
103	SUJIT	CSE	20
104	SURESH	CSE	18

- The conceptual model of Database design using ER diagram, After designing we need to convert the conceptual model in the relational model which can be implemented using any RDBMS languages.
- RDBMS languages : Oracle SQL, MySQL etc.

Difference between schema and instances :

Schema	Instance
It defines the basic structure of the database.	It is the set of Information stored at a particular time.
It is the overall description of the database.	It is the collection of information stored in a database.
Schema is same for whole database.	Instances can be changed using addition, deletion & updation.
It does not change frequently.	It can be changes frequently.

Relational Database :

- A relational database is a collection of information that stores data in one or more tables.
- It make easy to view and understand how different data structures relate to each other.
- It is used to store or manage the data that are in the form of tables only.
- Simply it is a way of structuring information in the form of tables.
- Relational database is most commonly used database.
- It contains number of tables and each table has its own primary key.

Properties of Relational Database :

There are four properties of a relational model known as ACID properties :

Atomicity :

- The entire transactions takes place at once.
- This ensures the data operation will complete either with success or failure.

Consistency :

- If we perform any operation over data, its value before and after the operation should be same.

Isolation :

- Multiple transactions occur independently without interference.

Durability :

- The changes of a successful transaction occurs even if the system fails.

What are the keys in DBMS ?

- A key is an attribute or set of attributes.
- It is used to uniquely identify a record in a table.
- It is also used to establish a relationships between tables.
- Keys play an important role in the relational database.
- There are several types of Keys :
  1. Primary Key
  2. Candidate Key
  3. Alternate Key
  4. Super Key
  5. Foreign Key

Primary key :

- Primary key is used to uniquely identify a tupple in a relation.
- It must have unique column value.
- Primary key can't be null.
- Only one primary key per table allowed.
- It can never be changed or update after assigned.
- Primary key is used to implement entity integrity.

R_No.	Name	Branch	PAN_No.	Mobile No.
101	abc	CSE	CMCRK2435H	9883561854
102	xyz	ECE	PMCRK2875B	7845328453
103	wxy	CSE	MKCRK2875C	7979085321
104	mnp	CIVIL	DLPRK5436R	7979085321

Here, R_No is considered as primary key.

Candidate Key :

- A candidate key is an attribute or set of attribute which can uniquely identify a record.
- Those attributes or combination of attributes are called candidate keys.
- Candidate keys are defined as distinct set of attributes from which primary key can be selected.
- A candidate key is a minimal super key(A super key with no redundant attributes).
- A Candidate key is a subset of Super keys and is devoid of any unnecessary attributes that are not important for uniquely identifying tuples.
- The value for the Candidate key is unique and not-null for all tuples.

Example :

{R_No}
{Pan_No}
{Mobile No.}

Alternate Keys :

- Out of all candidate keys, only one selected as primary key & remaining keys are known as alternate keys.
- In the above student table :
  - R_No. is best suited for primary key.
  - Rest of all attributes like PAN_No and Mobile No. are considered as alternate keys.

Super key :

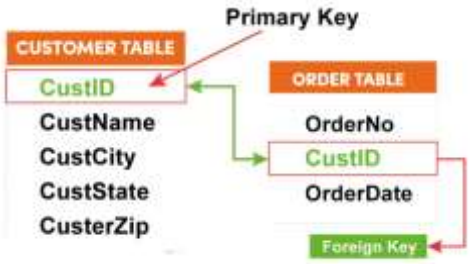
- It is a combination of all possible attributes that can uniquely identify the record from table.
- A super key is a superset of candidate keys.
- More than one super keys are possible in one relation.
- A super key may have additional attribute that are not needed for uniquely identify the record.

Example : In the above table.

{R_No}	{R_No, Name}
{PAN_No.}	{R_No, Name, Branch}
{Mobile No.}	{R_No, Name, Branch, Pan No.}
{Name, PAN_No.}	{Branch, Pan_no}
{Name, Mobile No}	{R_No, Name, Branch, Pan No., Mobile No.} etc.

Foreign key :

- A key used to link two table together.
- It is an attribute or set of attributes in one table refers to primary key in another table.
- It may be a null value column.
- More than one foreign key in a table.
- It is used to implement referential integrity.



Course_Id	Stu_City	Stu_Id
E001	Delhi	1001
E002	Siliguri	1002
E003	Kolkata	1003
E004	Mumbai	1004

Stu_Id	Stu_Name	Stu_Age
1001	Badal	23
1002	Canchan	20
1003	Sujit	21

What is constraints ?

- A constraints are the condition or restriction on data in a table.
- Constraints are always apply on column or attributes.
- That is, we can specify the limit on the type of data that can be stored in a particular column in a table using constraints.
- There are several constaraints which are given below :

Unique :

- If we make any column is unique, that means duplicates are not allowed in this particular column.
- This constraint when specified with a column, tells that all the values in the column must be unique.

Not Null :

- This constraint tells that we cannot store a null value in a column.
- That is, if a column is specified as NOT NULL then we will not be able to store null in this particular column.

Primary Key :

- A primary key is a field which can uniquely identify each row in a table.
- Primary key must be unique and not null.

Check :

- It helps to validate the values of a column must be satisfy a particular condition.
- That is, it helps to ensure that the value stored in a column meets a specific condition.

Foreign Key :

- A Foreign key is a field which can uniquely identify each row in a another table.

Default :

- This constraint specifies a default value for the column when no value is specified by the user.

Integrity Constraints :

- Integrity constraints are a set of rules.
- It is used to ensure accuracy and consistency of data in relational database.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Data integrity is the overall accuracy, completeness, and consistency of data
- Thus, integrity constraint is used to guard against accidental damage to the database.
- There are 4 types of integrity constraints :
  1. Domain constraints
  2. Entity Integrity constraints
  3. Referential Integrity constraints
  4. Key constraints

Domain constraints :

- All the value that appear in a column of a relation must be taken from the same domain.
- A domain is a set of allowable value that may be assigned to an attribute.

R_No.	Name	Age
101	Mohan	18
102	Raja	20
103	Rakesh	19
104	Umesh	D

Not allowed because Age is an integer value.

Entity Integrity constraints :

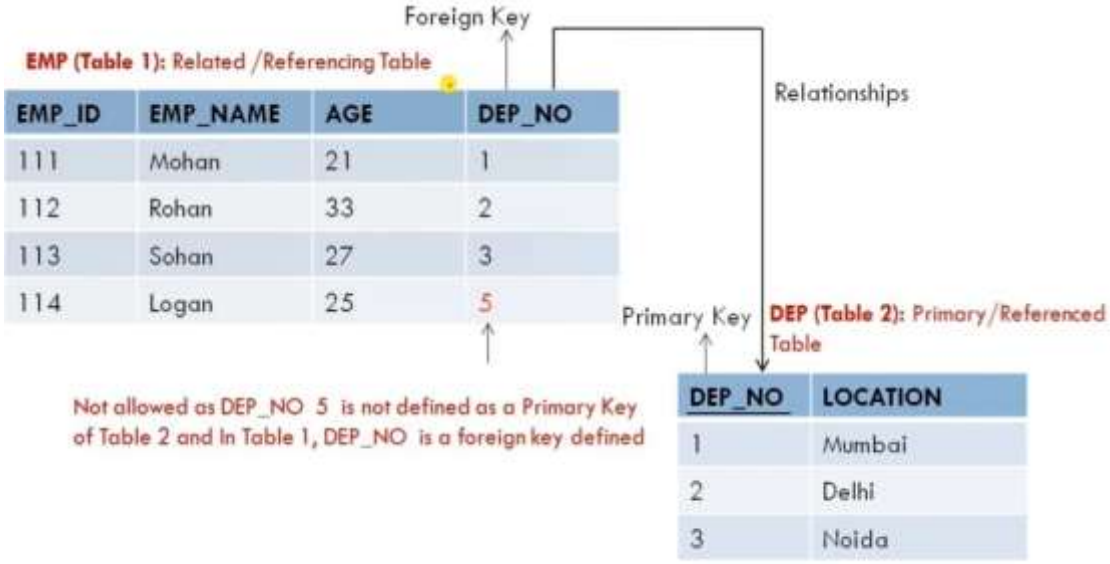
- It is related to primary key.
- If attribute of a table is prime attribute then it can't accept null value.
- Simply, It tells that the primary key can't be null.
- Because primary key is used to identify the record in a table.

R_No.	Name	Age
101	Mohan	18
102	Raja	20
103	Rakesh	19
	Umesh	21

Not allowed because, primary key can't accept null value.

Referential Integrity constraints :

- It is related to foreign key.
- It is enforced when a foreign key reference the primary key of a table.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.



Rules :

- You can not delete a recod from a primary table if matching recod exist in related table.
- You can not change a primary key value in the primary table.
- You can enetr a null value in the foreign key, that the recors are unrelated.

Key constraints

- There can be a number of keys in an entity set but only one will be the primary key out of all keys.
- In a relational table a primary key can have a unique as well as a null value.

R_No.	Name	Age
101	Mohan	18
102	Raja	20
103	Rakesh	19
102	Umesh	21

Not allowed because, all rows must be unique.

Difference between Entity Integrity constraints & Referential Integrity constraints :

Entity Integrity constraints	Referential Integrity constraints
It is related to primary key.	It is related to foreign key.
Entity integrity constraint refers to single table.	It refers to relationship between the tables.
It states that value in primary key fields cannot be null.	It states that a foreign key must have a matching primary key in another table or must be null.
Primary key assures the entity integrity constraint is applied to a table.	The relationship between tables is established by using foreign keys.

Data integrity :

- Data integrity is the overall accuracy, completeness, and consistency of data

Data redundancy :

- Data redundancy occurs when the same piece of data is stored in two or more separate places and **i**s a common occurrence in many businesses.

CTH EDUCATION SAHARSA ALOK SIR

Unit-04 : Database Design

- Normalization
- Normal forms – 1NF , 2NF , 3NF , BCNF , 4NF , and 5NF,
- E-R Diagram ,
- Mapping E-R diagram to database tables.

Unit -04 : Database Design :

1. What is normalization in DBMS ? Why we need normalization ?
2. Write the difference between 3NF and BCNF.
3. Explain fourth normal form(4NF) and fifth normal form(5NF).
4. What is functional dependency in DBMS ?
5. Explain trivial and non-trivial functional dependency with example.
6. Explain in brief Inference rule/Armstrong's Axioms.
7. Describe the E-R model with an example ? Also write its advantage.
8. Explain different components of E-R model in brief.
9. Discuss E-R modelling symbols in details.
10. Draw an ER diagram for a company.
11. Construct an ER diagram of your college and establish relationship between entities.

CTH EDUCATION SAHARSA ALOK SIR

What is Normalization ?

- Normalization is the technique, which is used to organize a data in database.
- It is the process of reducing redundancy by eliminating insertion, deletion and modification problem & increase data integrity in the database.
- Normalization divides the larger table into the smaller table and links them using relationship.
- Normalization works through a series of stages called Normal forms.
- The normal forms apply to individual relations.
- The normal form is used to reduce redundancy from the database table.

Why we need the normalization ?

- If the database is in denormalize form there are some anomalies occurs in the database.
  1. Update anomaly
  2. Insert anomaly
  3. Delete anamoly
- After normalization we can solve above anomalies problems so, we need normalization technique to normalize the database.
- Example : Employee table

Emp_id	Emp_Name	Emp_Address	Emp_Dept.
101	Mohan	A1	D001
102	Mohan	A1	D002
103	Raja	A3	
104	Aditya	A4	D004

Advantages of Normalization :

- Reduces redundant data
- Provides data consistency within the database
- More flexible database design
- Higher database security
- Better and quicker execution

Denormalization :

- It is the opposite of normalization.
- It is the process of increasing redundancy in the database.

Functional Dependency :

- Functional dependency (FD) is an important concept associated with normalization.
- It is a method which describe the relationship between the attributes.
- A functional dependency is denoted by an arrow "→".
- The functional dependency of **A** on **B** is represented by A → B.
- Here, A is the determinant of B and B is the dependent on A.

Consider a relation with four attributes A, B, C and D,

R (ABCD)

- A → BCD
- B → CD

- For the first functional dependency **A → BCD**, attributes **B, C** and **D** are functionally dependent on attribute **A**.
- Function dependency **B → CD** has two attributes **C** and **D** functionally depending upon attribute **B**.

Advantages of Functional Dependency :

1. It is used to maintain the quality of data in the database**.**
2. It expresses the facts about the database design**.**
3. It helps in clearly defining the meanings and constraints of databases.
4. It helps to identify bad designs.
5. Functional Dependency removes data redundancy where the same values should not be repeated at multiple locations in the same database table.
6. The process of Normalization starts with identifying the candidate keys in the relation.
7. Without functional dependency, it's impossible to find candidate keys and normalize the database.

Types of Functional dependency :

- There are several types of functional dependency :
    1. Trivial FD
    2. Non-trivial FD
    3. Partial FD
    4. Transitive FD
    5. Full functional dependency

Trivial functional dependency :

- A → B has trivial functional dependency if B is a subset of A.
- The following dependencies are also trivial like: A → A, B → B

Example : ABC → BC

Non-trivial functional dependency :

- A → B has a non-trivial functional dependency if B is not a subset of A.
- When A intersection B is NULL, then A → B is called as complete non-trivial.

Example : AB → BC(Non-trivial)

AB → CD(Completely non-trivial)

Partial FD :

- All non key attributes must depend on primary key attributes.
- Consider a relation with four attributes A, B, C and D,

  R (ABCD)

  AB → C

  B → D

  Key : AB

  B → D(Partially depend on AB)

Transitive FD :

- If there is a relation between non key attribute, then it is transitive dependency.

  AB → C

  C → D

  Key : AB

  Here, AB should have identified D but also C → D not AB→ D directly.

Full functional dependency :

- If there is a dependency of the form X → Y then the removal of attributes from X makes X → Y invalid.
- Consider a relation with four attributes A, B, C and D,

  R (ABCD)

  AB → C

Properties of FD/Inference Rule (IR)/ Armstrong's axioms :

- William Armstrong in 1974 suggested a few rules related to functional dependency.
- It can apply to a set of FD to derive other FD.
- The Functional dependency has 5 types of inference rule :
  1. Reflexive Rule (IR_1)
  2. Augmentation Rule (IR_2)
  3. Transitive Rule (IR_3)
  4. Union Rule (IR_4) :
  5. Decomposition Rule (IR_5) :

Reflexive Rule (IR_1)

- In the reflexive rule, if Y is a subset of X, then X determines Y.
- If $X \supseteq Y$ then $X \rightarrow Y$

Augmentation Rule (IR_2)

- The augmentation is also called as a partial dependency.
- In augmentation, if X determines Y, then XZ determines YZ for any Z.
- If $X \rightarrow Y$ then $XZ \rightarrow YZ$

Example :

For R(ABCD), **if** $A \rightarrow B$ then $AC \rightarrow BC$

Transitive Rule (IR_3)

- In the transitive rule, if X determines Y and Y determine Z, then X must also determine Z.
- If $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

Union Rule (IR_4) :

- Union rule says, if X determines Y and X determines Z, then X must also determine Y and Z.
- If $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$

Decomposition Rule (IR_5) :

- Decomposition rule is also known as project rule.
- It is the reverse of union rule.
- This Rule says, if X determines Y and Z, then X determines Y and X determines Z separately.
- If $X \rightarrow YZ$ then $X \rightarrow Y$ and $X \rightarrow Z$

Types of Normal Forms :

- Normalization works through a series of stages called Normal forms.
- The normal forms apply to individual relations.
- The relation is said to be in particular normal form if it satisfies constraints.
- Normal forms : 1NF , 2NF , 3NF , BCNF , 4NF , and 5NF.

First Normal Form (1NF)

- A relation will be 1NF if it contains an atomic value.
- It states that an attribute of a table cannot hold multiple values.
- A relation is said to be in 1NF if :
  - It only consists of atomic values
  - No repeating groups are present

Roll No	Name	Course
101	Mohan	DBMS, CG
102	Rakesh	DBMS
103	Ram	OS, Python

1NF →

Roll No	Name	Course
101	Mohan	DBMS
101	Mohan	CG
102	Rakesh	DBMS
103	Ram	OS
103	Ram	Python

Second Normal Form (2NF)

- A relation is said to be in second normal form, if
  - It is already in 1NF
  - It should must be free from partial FD.
- In the 2NF, all non-key attributes are fully functional dependent on the candidate key.
- In other words non-key attributes not depend on a part of a candidate key.
- Example :

TEACHER_ID	SUBJECT	TEACHER_AGE
25	DBMS	30
25	CG	30
47	Python	35
83	DS	38
83	OS	38

Candidate key {TEACHER_ID, SUBJECT}

- In the given table, non-prime attribute TEACHER_AGE is dependent on TEACHER_ID which is a proper subset of a candidate key.
- So, it violates the rule for 2NF.
- To convert the given table into 2NF, we decompose it into two tables :

TEACHER_DETAIL table :

TEACHER_ID	TEACHER_AGE
25	30
47	35
83	38

TEACHER_SUBJECT table :

TEACHER_ID	SUBJECT
25	DBMS
25	CG
47	Python
83	DS
83	OS

Third Normal Form (3NF) :

- A relation is said to be in 3NF if :
  - It is already in 2NF and
  - Free from transitive dependency.
- A relation is in 3NF if it holds atleast one of the following conditions for every non-trivial function dependency X → Y.
  1. X is a super key.
  2. Y is a prime attribute, i.e., each element of Y is part of some candidate key.

EMPLOYEE_DETAIL table :

EMP_ID	EMP_NAME	EMP_ZIP	EMP_CITY
222	Ram	852201	Saharsa
333	Mohan	800001	Patna
444	Rakesh	800001	Patna
555	Aman	110001	Delhi

Candidate key : {EMP_ID}

Non-prime attributes : EMP_NAME, EMP_ZIP & EMP_CITY.

- Here, EMP_CITY dependent on EMP_ZIP and EMP_ZIP dependent on EMP_ID.
- The non-prime attributes EMP_CITY transitively dependent on super key(EMP_ID).
- It violates the rule of third normal form.
- Then, we need to move the EMP_CITY to the new <EMPLOYEE_ZIP> table, with EMP_ZIP as a Primary key.

EMPLOYEE table :

EMP_ID	EMP_NAME	EMP_ZIP
222	Ram	852201
333	Mohan	800001
444	Rakesh	800001
555	Aman	110001

EMPLOYEE_ZIP table :

EMP_ZIP	EMP_CITY
852201	Saharsa
800001	Patna
110001	Delhi

Boyce Codd normal form (BCNF) :

- A relation is said to be in BCNF if :
  - It is already in 3NF and
  - For every FD, LHS is super key.
- BCNF is the special case of 3NF.
- A table is in BCNF if every functional dependency X → Y, X is the super key of the table.

Roll No	Name	Voter Id	Age
101	Ram	S0152	24
102	Mohan	K0021	30
103	Rakesh	M0231	32
104	Aman	P0125	20

C.K {Roll No, Voter Id}

FD :

Roll No → Name

Roll No → Voter Id

Voter Id → Age

Voter Id → Roll No

Here, LHS of every FD should be candidate/super key therefore given relation in BCNF.

Difference between 3NF & BCNF :

3NF	BCNF
In 3NF the functional dependencies are already in 1NF and 2NF.	In BCNF the functional dependencies are already in 1NF, 2NF and 3NF.
3NF is less strict than BCNF.	BCNF is more strict than 3NF.
In 3NF there should be no transitive dependency.	In BCNF for any relation A→B, A should be a super key of relation.
It is easier to achieve.	It is difficult to achieve.
Lossless decomposition can be achieved by 3NF.	Lossless decomposition is hard to achieve in BCNF.

Fourth Normal Form(4NF) :

- A relation is said to be in 4NF if :
  - It is already in BCNF.
  - Free from Multi-valued Dependency.

What is Multi-valued Dependency ?

1. For a dependency A → B, if for a single value of A, multiple value of B exists, then the table may have multi-valued dependency.

Example

s_id	course	hobby
1	Science	Cricket
1	Maths	Hockey
2	C#	Cricket
2	Php	Hockey

How to satisfy 4th Normal Form ?

To make the above relation satify the 4th normal form, we can decompose the table into 2 tables.

CourseOpted Table

s_id	course
1	Science
1	Maths
2	C#
2	Php

Hobbies Table,

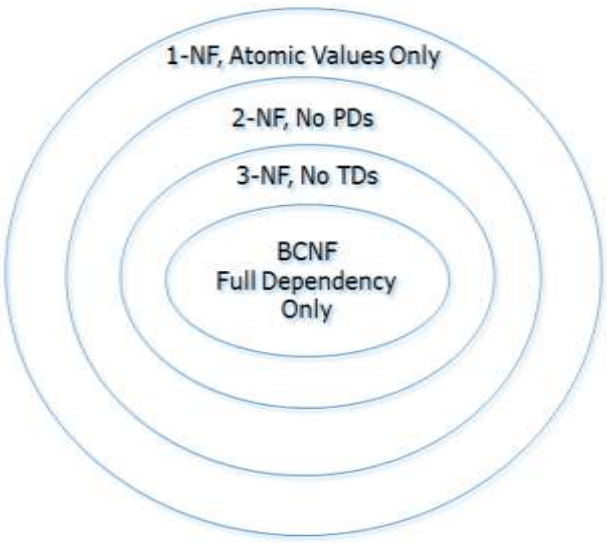
s_id	hobby
1	Cricket
1	Hockey
2	Cricket
2	Hockey

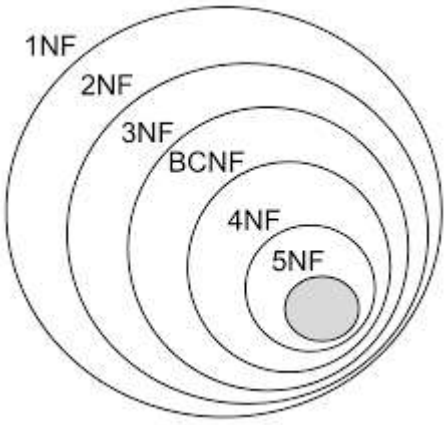
Fifth normal form (5NF) :

- A relation is said to be in 5NF if
  - it is already in 4NF and
  - Free from join dependency and joining should be lossless.
- 5NF is also known as Project-join normal form (PJ/NF).

Join dependency :

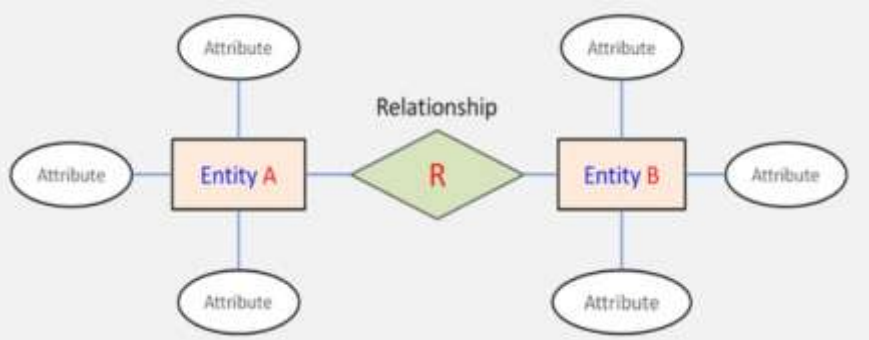
- A relation R is in 4NF and decomposed in R1, R2, R3,.....,Rn then, join all the relation from R1 to Rn gives R.

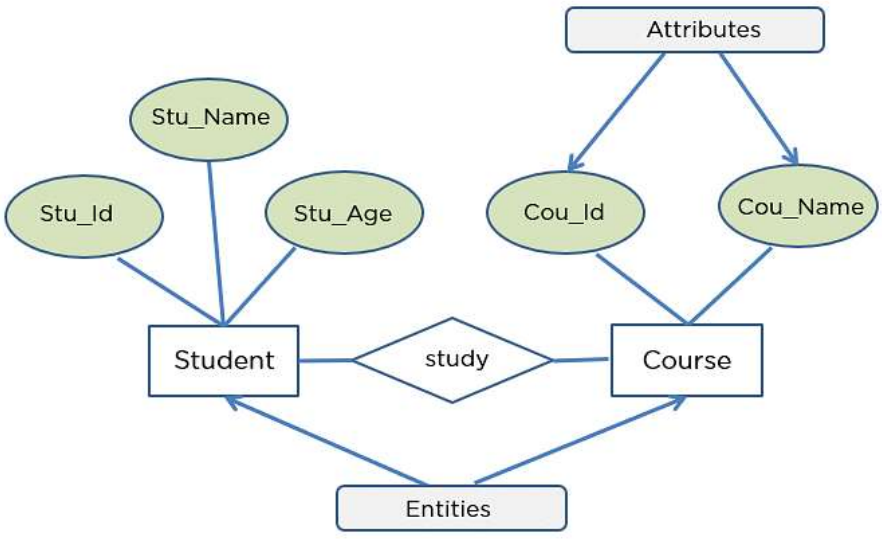




E-R Diagram(ERD) :

- E-R diagram stands for an Entity-Relationship model.
- ERD is a conceptual data model that describe the structure of database.
- It gives graphical repersentaion of requirments to design database in terms of entity, relationship and attributes.
- ERD is a diagrammatic representation of logical structure of database design.
- ER model describes relationship among entities and attributes.
- E-R diagram is developed by Peter Chen in 1976.
- There are three components in ERD :
  - Entities,
  - Relationships and
  - Attributes.





Advantage of E-R Model.

- It is an effective communication tool among database designer.
- It is easy to undersatand and better visual representation.
- It is lightly integrated with relational database model so conversion is easy.
- Conceptually it is very simple
- ER model is very simple because if we know relationship between entities and attributes, then we can easily draw an ER diagram.

Component of E-R Diagram :

There are three main components of the ER Model are-

1. **Entities**,
2. **Relationships**
3. **Attributes**

Entity :

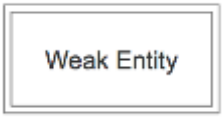
- A real world object or thing with independent existence is known as entity.
- Each entity has particular properties called attributes that describe it.
- An entity may be any object, class, person or place.
- Example :
    - Concider an Employee is an entity.
    - It has four attributes : Name, Address, Age, Contact No.
- **There are two types of entity :**
    1. Strong entity
    2. Weak entity

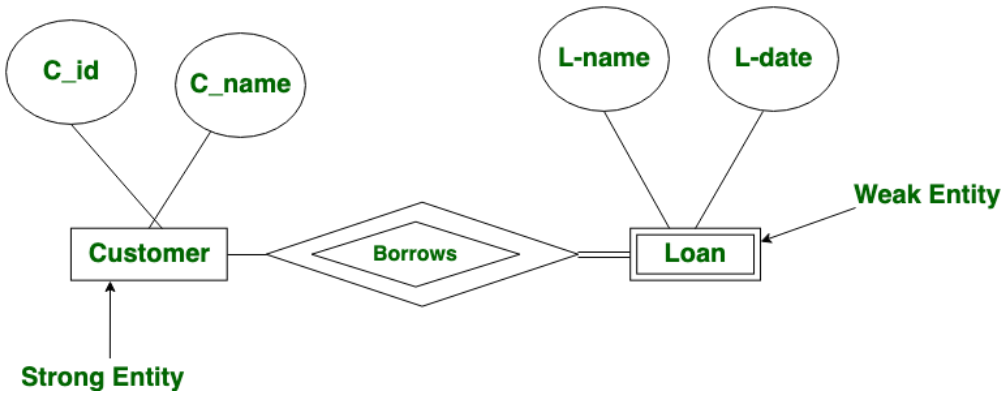
Strong entity :

- An entity should have a key attribute which uniquely identifies each record.
- In the ER diagram, an entity can be represented as rectangles.

Weak entity :

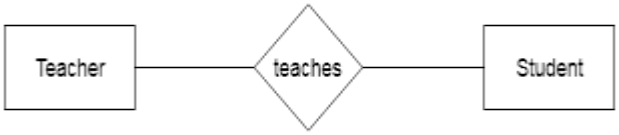
- An entity that depends on another entity called a weak entity.
- The weak entity doesn't contain any key attribute of its own.
- The weak entity is represented by a double rectangle.





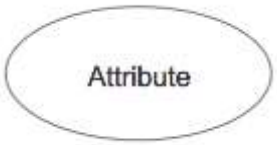
Relationship :

- It gives association among entity.
- Between entity one or more relations are possible.
- There are three types of relations :
  1. One to one(1 : 1)
  2. One to many(1 : M)
  3. Many-to-one(M : 1)
  4. Many to many(M : M)



Attributes :

- It is used to describe the property of an entity.
- Eclipse is used to represent an attribute.
- Each entity has attributes which are the properties of each entity.
- There are several types of attributes :
  - Simple
  - Composite
  - Single valued
  - Multivalued
  - Derived



Simple attribute :

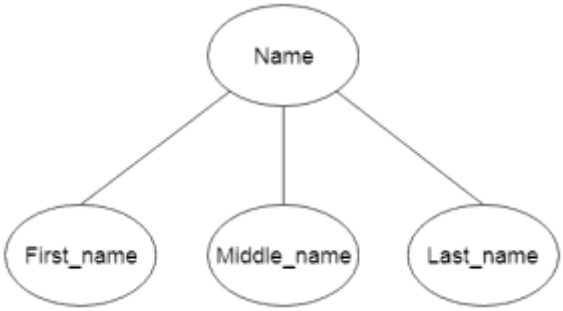
- It has atomic value can't be divided further.

  Example : Roll No, age etc.

Composite Attribute :

- An attribute that composed of many other attributes is known as a composite attribute.
- It can be further divided into simple attribute.

  Example : Name, Address etc



Single valued attribute :

- The attributes that holds single valued.

  Example : PAN No, Voter ID etc.

Multivalued Attribute :

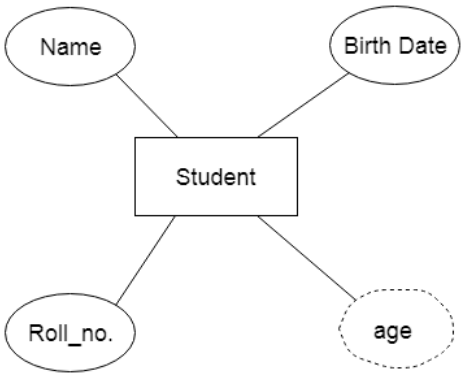
- An attribute can have more than one value are known as a multivalued attribute.
- The double oval is used to represent multivalued attribute.

  Example: Phone number, Email id etc.

Derived Attribute :

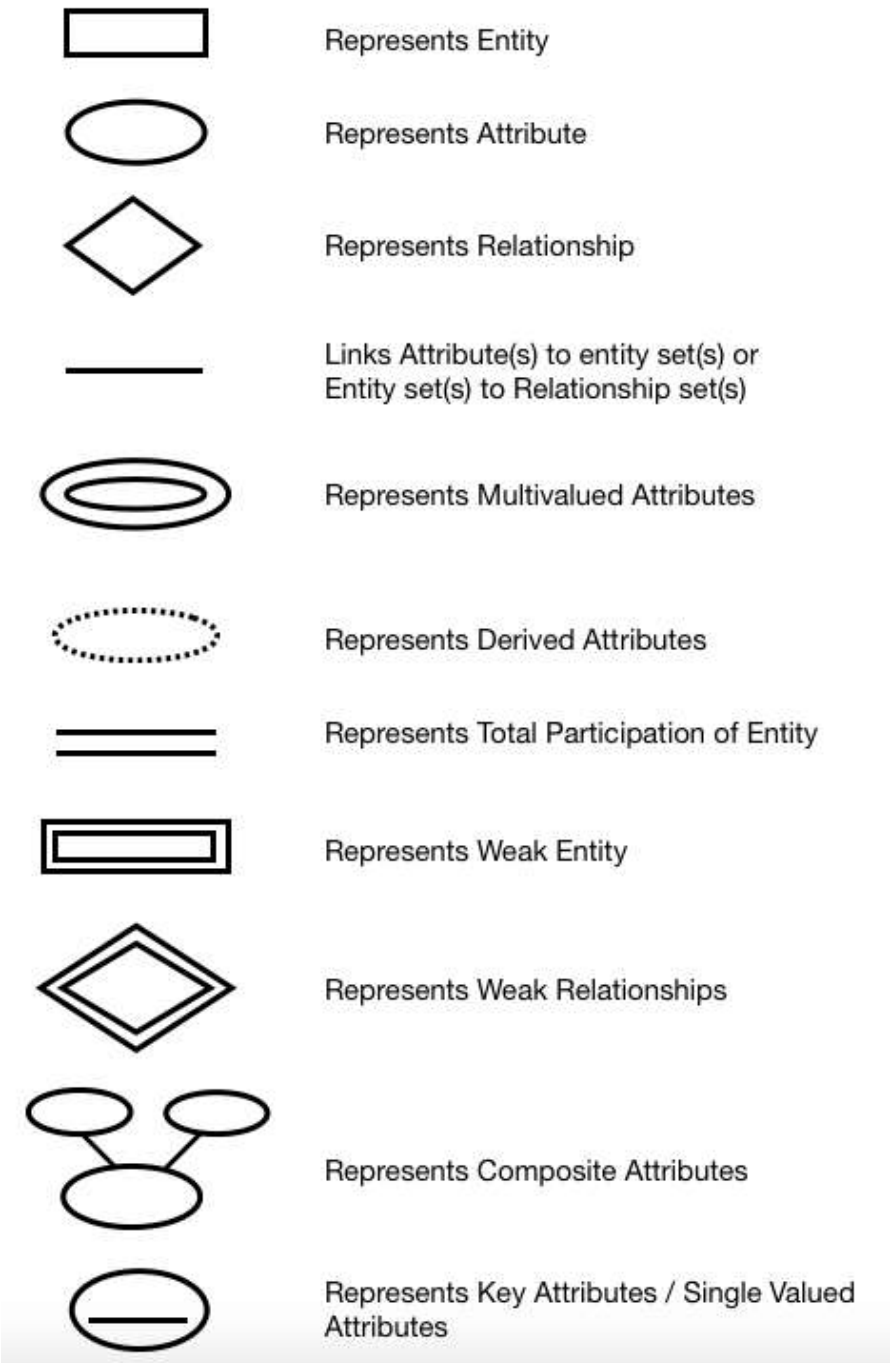
- An attribute that can be derived from other attribute is known as a derived attribute.
- It can be represented by a dashed ellipse.

Example, A person's age changes over time and can be derived from another attribute D.O.B.



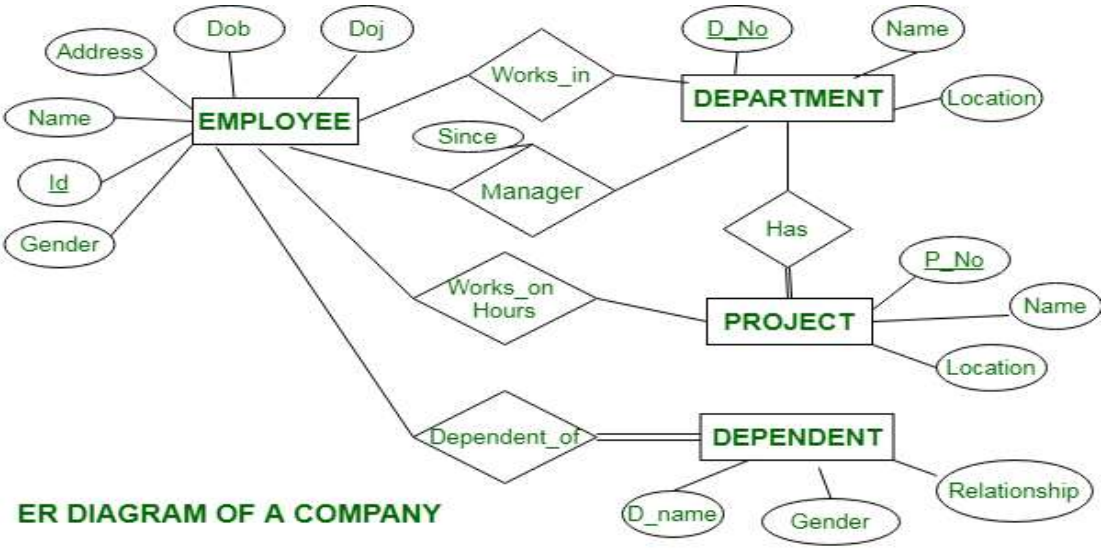
Discuss E-R modelling symbols.

- There are many symbols which are used in ER model



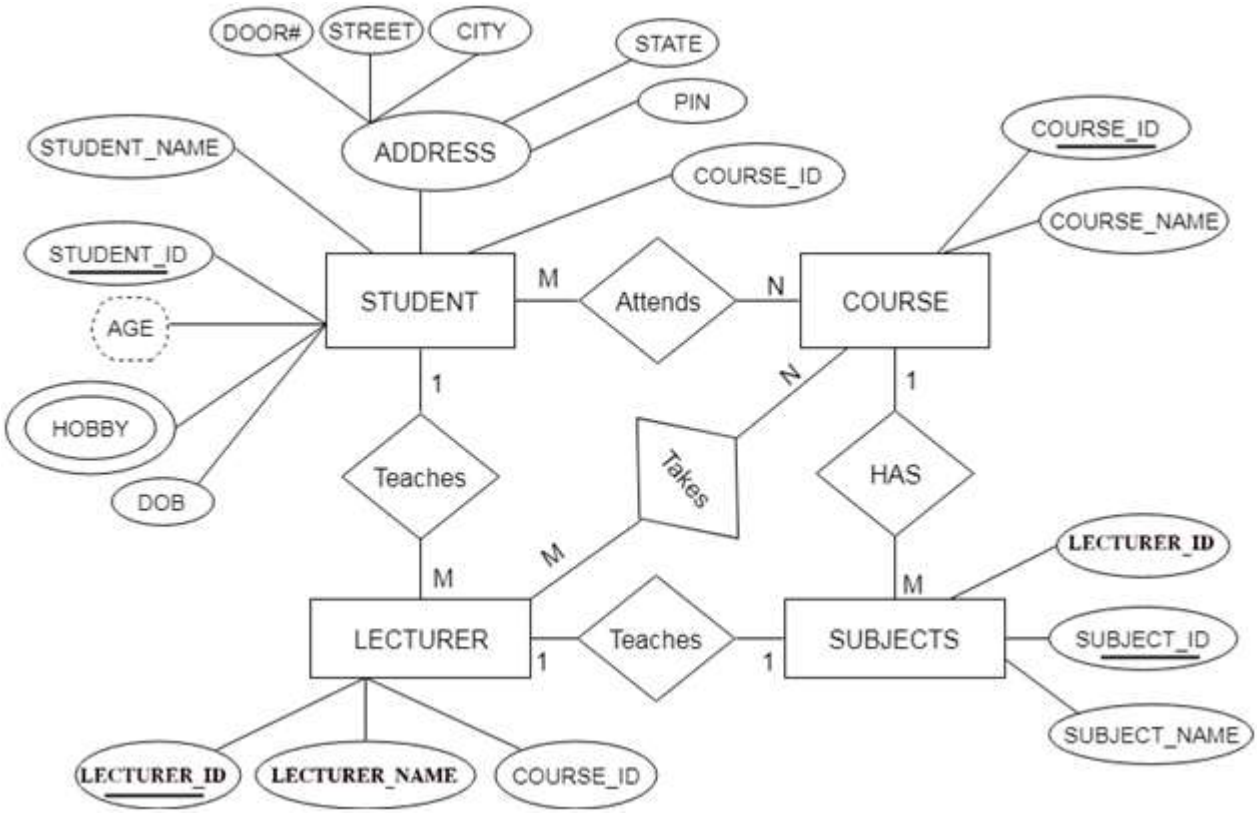
Q. Construct an ER diagram of your college and establish a proper relationship between entities.

Draw an ER diagram for a company.



ER DIAGRAM OF A COMPANY

Draw an ER diagram for a company.



Unit-05 : MariaDB

- Introduction to MariaDB,
- Data types,
- SQL commands, Create, insert, update, delete, drop, alter,
- SQL function (string function, date function), indexing , Key , primary key , foreign key.

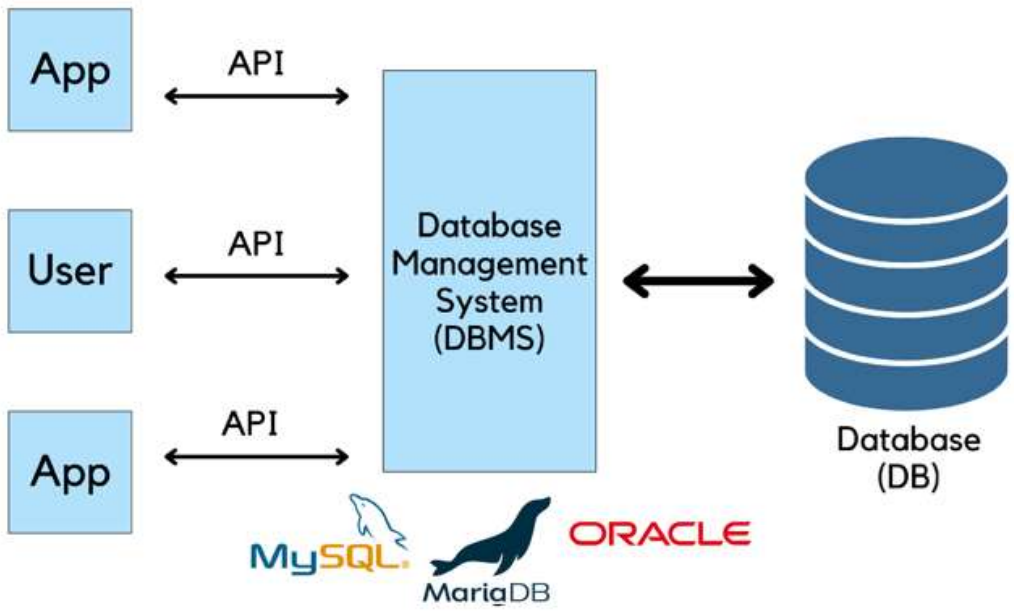
Questions to be discussed :

1. What is MariaDB ? Write the features of MariaDB.
2. Explain in details about data types of MariaDB.
3. Define the term SQL. Write the advantage of SQL.
4. Write the structure of SQL SELECT statements.
5. List and explain aggregate function used in SQL with examples.
6. Explain in brief about various SQL statements(DDL, DML & DCL).
7. Explain string function and date function in details.
8. Explain Grant and Revoke commands in SQL.

CTH EDUCATION SAHARSA ALOK SIR

Introduction to MariaDB :

- MariaDB is the most popular open-source RDBMS.
- It could be used as a great replacement of MySQL.
- It is developed by original developers of MySQL in 2009.
- MariaDB can also stores data in various tables because it is a RDBMS.
- To establish the relationship between these tables, primary keys and foreign keys are used.
- The MariaDB project is the brainchild of Michael "Monty" Widenius, the founder of MySQL.



Features of MariaDB :

- It is faster, scalable and robust.
- It is licensed under GPL, LGPL, or BSD.
- It uses a standard and popular querying language.
- It runs on different operating systems and supports variety of programming languages.
- It also offers support for PHP.

Different types of open source licenses :

1. General Public License(GPL)
2. Lesser General Public License(LGPL)
3. Berkley Source Distribution(BSD)

MariaDB Data Types :

- Before designing a database in MariaDB, you should consider the available data types so that you can select the most optimal ones for storing data.
- MariaDB provides you with many data types including :
  1. Numeric data types
  2. String data types
  3. Temporal data types
  4. Spatial data types

MariaDB numeric data types :

Numeric Types	Description
TINYINT	A very small integer
SMALLINT	A small integer
MEDIUMINT	A medium-sized integer
INT	A standard integer
BIGINT	A large integer
DECIMAL	A fixed-point number
FLOAT	A single-precision floating-point number
DOUBLE	A double-precision floating-point number
BIT	A bit

MariaDB Boolean data type :

- MariaDB uses the TINYINT(1) to represent Boolean values.
- In MariaDB, zero (0) means false and non-zero means true.
- The BOOLEAN and BOOL are the synonym of TINYINT(1).

String Types	Description
CHAR	A fixed-length nonbinary (character) string
VARCHAR	A variable-length non-binary string
BINARY	A fixed-length binary string
VARBINARY	A variable-length binary string

String Types	Description
TINYTEXT	A very small non-binary string
TEXT	A small non-binary string
MEDIUMTEXT	A medium-sized non-binary string
LONGTEXT	A large non-binary string
ENUM	An enumeration
SET	A set

MariaDB temporal types :

- MySQL temporal types including types that represent a date without time, a time without date, a datetime, a timestamp, and year.

Temporal Data Types	Description
DATE	A date value in CCYY-MM-DD format
TIME	A time value in hh:mm:ss format
DATETIME	A date and time value in CCYY-MM-DD hh:mm:ss format
TIMESTAMP	A timestamp value in CCYY-MM-DD hh:mm:ss format
YEAR	A year value in CCYY or YY format

MariaDB spatial data types :

- MariaDB supports many spatial data types that contain various kinds of geographical values :

Spatial Data Types	Description
GEOMETRY	A spatial value of any type
POINT	A point (a pair of X-Y coordinates)
LINESTRING	A curve (one or more POINT values)
POLYGON	A polygon
GEOMETRYCOLLECTION	A collection of GEOMETRY values
MULTILINESTRING	A collection of LINESTRING values
MULTIPOINT	A collection of POINT values
MULTIPOLYGON	A collection of POLYGON values

What is SQL ?

- SQL stands for Structured Query Language.
- SQL is pronounced as SEQUEL or Structured English QUEry Language.
- It is initially created by IBM and modified by various software designers.
- It is a computer language for storing, manipulating and retrieving data stored in a relational database.
- SQL is the standard language for Relational Database System.
- This fourth generation language is a very simple and powerful because it uses English like statements.
- SQL is a high level language that provide a greater degree of abstraction than procedural language.
- All the RDBMS like MySQL, MS Access, Oracle, Sybase, Informix and SQL Server use SQL as their standard database language.

Advantage of SQL ?

- Simple and easy to learn can handle complex situation.
- Creating, deleting and modifying table structure.
- Defining the relationship between two or more tables.
- Inserting data into two tables.
- Update data and controlling a database.
- Extracting data in meaningful ways.

Difference between SQL and MySQL :

SQL	MySQL
SQL stands for Structured Query Language.	MySQL is a Relational DBMS.
SQL is a language.	MySQL is a Database.
SQL is used to query database.	MySQL is used to stores data.
SQL commands are used in Oracle, MariaDB, MySQL etc.	MySQL uses SQL.

SQL Commands :

- The standard SQL commands to interact with relational databases are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP.
  - **CREATE** – Creates a new table in database.
  - **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 entire tables and databases.

These commands can be classified into the following groups based on their nature :

DDL - Data Definition Language :

Sr.No.	Command & Description
1	**CREATE :** Creates a new table, a view of a table, or other object in the database.
2	**ALTER :** Modifies an existing database object, such as a table.
3	**DROP :** Deletes an entire table, a view of a table or other objects in the database.

DML - Data Manipulation Language :

Sr.No.	Command & Description
1	**SELECT :** Retrieves certain records from one or more tables.
2	**INSERT :** Creates a record.
3	**UPDATE :** Modifies records.
4	**DELETE :** Deletes records.

DCL - Data Control Language :

Sr.No.	Command & Description
1	**GRANT :** Gives a privilege to user.
2	**REVOKE :** Takes back privileges granted from user.

SQL SELECT statement :

- The SQL **SELECT** statement is used to fetch the data from a database table which returns the data in the form of a result table.
- These result tables are called result-sets.

Syntax :

```
SELECT column1, column2, columnN
FROM table_name;
```

Functions in SQL :

- For doing operations on data, SQL has many built-in functions.

Aggregate functions :

- These functions are used to do operations from the values of the column and returned as a single value.
  1. AVG()
  2. COUNT()
  3. FIRST()
  4. LAST()
  5. MAX()
  6. MIN()
  7. SUM()

Students-Table

ID	NAME	MARKS
1	Harsh	90
2	Suresh	50
3	Pratik	80
4	Dhanraj	95
5	Ram	85

AVG() : It returns the average value after calculating from values in a numeric column.

Syntax:

```
SELECT AVG(column_name) FROM table_name;
```

Example :

```
SELECT AVG(MARKS) AS AvgMarks FROM Students;
```

Output : AvgMarks : 80

COUNT() : It is used to count the number of rows returned in a SELECT statement.

Syntax:

```
SELECT COUNT(column_name) FROM table_name;
```

Example :

SELECT COUNT(*) AS NumStudents FROM Students;

Output : NumStudents : 5

FIRST() : The FIRST() function returns the first value of the selected column.

Syntax :

SELECT FIRST(column_name) FROM table_name;

Example :

SELECT FIRST(MARKS) AS MarksFirst FROM Students;

Output : MarksFirst : 90

LAST() : The LAST() function returns the last value of the selected column.

Syntax :

SELECT LAST(column_name) FROM table_name;

Example :

SELECT LAST(MARKS) AS MarksLast FROM Students;

Output : MarksLast : 85

MAX() : The MAX() function returns the maximum value of the selected column.

Syntax:

SELECT MAX(column_name) FROM table_name;

Example :

SELECT MAX(MARKS) AS MaxMarks FROM Students;

Output : MaxMarks : 95

MIN() : The MIN() function returns the minimum value of the selected column.

Syntax :

SELECT MIN(column_name) FROM table_name;

Queries : Fetching minimum marks among students from the Students table.

Example :

SELECT MIN(MARKS) AS MinMarks FROM Students;

Output : MinMarks : 50

SUM() : The SUM() function returns the sum of all the values of the selected column.

Syntax :

SELECT SUM(column_name) FROM table_name;

Queries : Fetching summation of total marks among students from the Students table.

Example :

SELECT SUM(MARKS) AS TotalMarks FROM Students;

Output : TotalMarks : 400

SQL String functions :

- String functions are used to perform an operation on input string and return an output string.
- Following are the string functions defined in SQL:

1. **ASCII() :** This function is used to find the ASCII value of a character.
   Syntax : SELECT ascii('t');
   Output : 116

2. **CHAR_LENGTH() :** This function is used to find the length of a word.
   Syntax : SELECT char_length('Hello');
   Output : 5

3. **CHARACTER_LENGTH() :** This function is used to find the length of a line.
   Syntax : SELECT CHARACTER_LENGTH('cth education');
   Output : 13

4. **CONCAT() :** This function is used to add two words or strings.
   Syntax: SELECT 'cth' || ' ' || 'education' FROM dual;
   Output: 'ctheducation'

5. **REVERSE():** This function is used to reverse a string.
   Syntax: SELECT REVERSE('ctheducation.in');
   Output: 'ni.noitacudehtc'

SQL Date functions :

- In MySql the default date functions are :

NOW() : Returns the current date and time.

Example :

```
SELECT NOW();
```

Output : 2022-10-03 09: 06: 55

CURDATE() : Returns the current date.

Example :

```
SELECT CURDATE();
```

Output : 2022-10-03

CURTIME() : Returns the current time.

Example:

```
SELECT CURTIME();
```

Output : 05:07:15

SQL indexes :

- An index is a schema object.
- It is used by the server to speed up the retrieval of rows by using a pointer.
- It can reduce disk I/O by using a rapid path access method to locate data quickly.
- Indexes can be created or dropped with no effect on the data.

Creating an Index :

Syntax:

```
CREATE INDEX index
ON TABLE column;
```

where the **index** is the name given to that index and **TABLE** is the name of the table on which that index is created and **column** is the name of that column for which it is applied.

For multiple columns :

Syntax:

```
CREATE INDEX index
ON TABLE (column1, column2,.....);
```

Unit-06 : Manipulating Data with MariaDB

- SQL statements,
  - select,
  - like clause ,
  - group-by ,
  - order-by ,
- joins :
  - left join,
  - right join ,
  - natural join ,
  - union , correlated and nested queries ,
- Backup & restore.

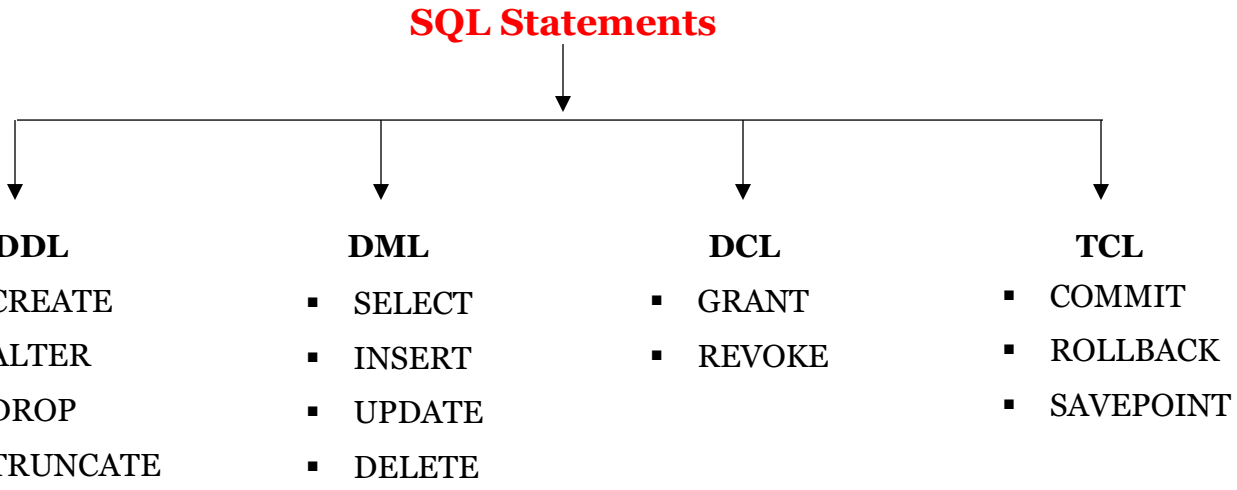
Questions to be discussed :

1. Write the structure of SQL SELECT statements.
2. Explain in brief about various SQL statements.
3. Write syntax of SQL Orderby and Groupby clauses.
4. Write short notes on nested query.
5. What is a join ? Discuss different types of joins.
6. Differentiate between natural join and inner join.

By : Alok Sir (Mob. No.: +91-80 84 370 470)

What are SQL statements ?

- A SQL statement is a set of instruction.
- It consists of identifiers, parameters, variables, data types, and SQL reserved words that compiles successfully.
- There are 4 types of SQL Statements :
  1. DDL (Data Definition Language)
  2. DML (Data Manipulation Language)
  3. DCL (Data Control Language)
  4. TCL (Transaction Control Language)



DDL(Data Definition Language) :

- In DDL, we have four different SQL statements.

Command	Description
CREATE	CREATE statement is used to create a new table in an existing database.
ALTER	Alter statement can add a column, modify a column, drop a column, rename a column or rename a table.
DROP	DROP statement is used to remove a table definition and all the data, indexes, triggers, constraints and permission specifications for the table.
TRUNCATE	TRUNCATE removes all rows from a table, without logging the individual row deletions.

DML(Data Manipulation Language) :

- In DML, we have four different SQL statements, Select, Insert, Update, and Delete.

Command	Description
SELECT	The SELECT statement is used to select records from the table, with or without a condition.
INSERT	INSERT statement is used to insert a set of values into a database table. Insert statement it used with Values.
UPDATE	The UPDATE statement is used to update existing values in a table, which is based on some condition.
DELETE	Delete statement is used to delete the existing record in the table, which is based on some condition.

DCL(Data Control Language) :

- It defines the control over the data in the database.
- We have two different commands, which are

Command	Description
GRANT	Grant is allowed to do the specified user to the specified tasks.
REVOKE	It is used to cancel previously granted or denied permissions.

TCL(Transaction Control Language) :

- In (TCL), the commands are used to manage the transactions in the database.
- These are used to manage the changes made by DML statements.

Command	Description
COMMIT	Commit command is used to permanently save any transaction into the DB.
ROLLBACK	Rollback command is used to restore the database for the last committed state. It's also used with a save point to jump to the save point.
SAVEPOINT	SAVEPOINT command is used to temporarily save a transaction so that you can roll back to that point whenever necessary.

How to Create a SQL Statement :

1. Start your query with the SELECT statement.
2. Add field names you want to display.
3. Write the name of table along with FROM.
4. Add your statement clause or selection criteria if required.

Syntax :

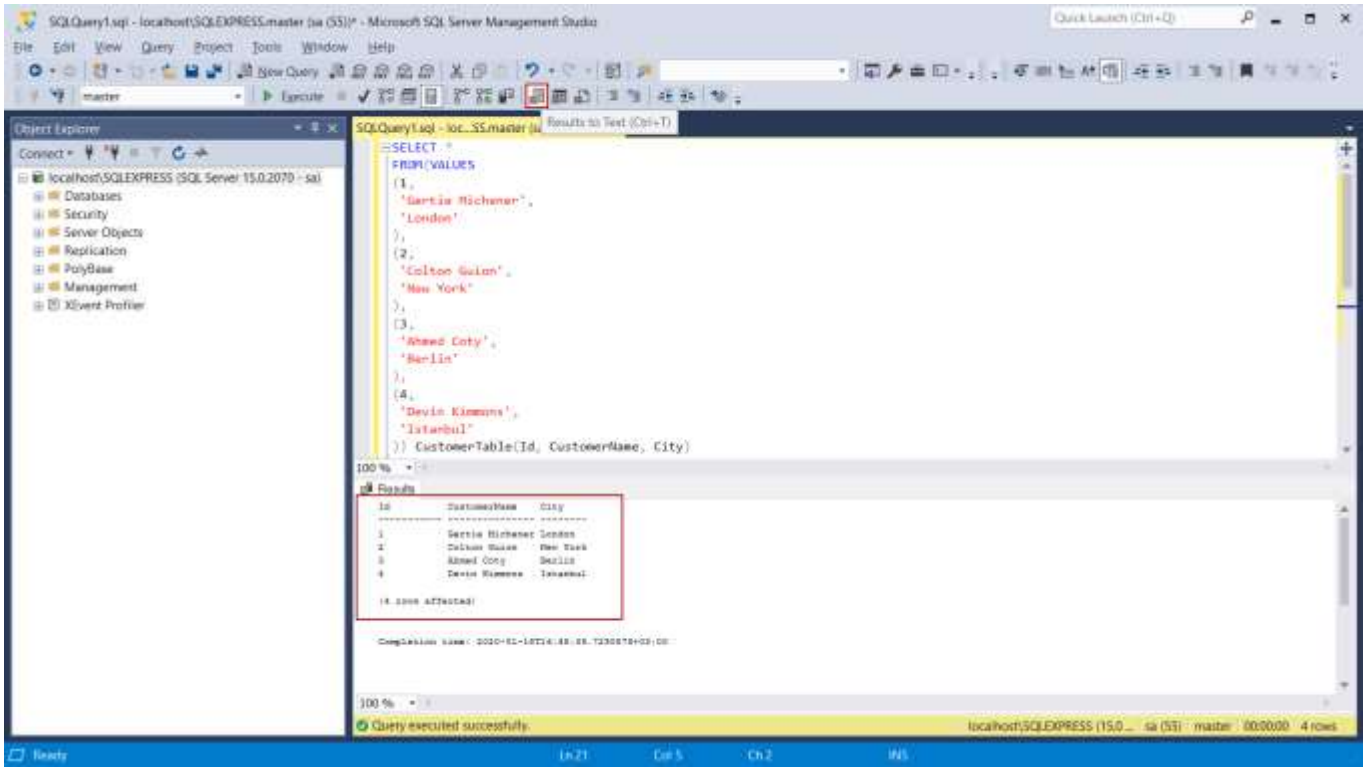
```
SELECT column1, column2, ...
FROM table_name;
```

Example :

```
SELECT CustomerName, City
FROM Customers-table;
```

Where do I write SQL code ?

- Microsoft SQL Server Management Studio (SSMS).
- Microsoft SQL Server Management Studio allows users to create and edit SQL queries and manage databases.



What is clause in SQL ?

- Clauses are in-built functions available to us in SQL.
- With the help of clauses, we can deal with data easily stored in the table.
- Clauses help us filter and analyze data quickly.
- When we have large amounts of data stored in the database, we use Clauses to query and get data required by the user.
- There are several clause available in SQL :
  - Where
  - And
  - Or
  - Like
  - Limit
  - Order By
  - Group By

Clause Name	Description
Where	When using a Where clause, we need to mention at least one condition.
And	When using an And clause, we need to mention at least two conditions.
Or	When using an Or clause, we need to mention at least two conditions
Like	LIKE clause is beneficial to find specific patterns in the data. We use specific symbols i.e (%) and (_).
Limit	We need to specify a number after the limit clause. Float and exponential values can't be utilized.
Order By	We use order by clause to sort data in ascending or descending order as required by the user. By default, the data is sorted in ascending order.
Group By	We use order by clause to get the summary of data in rows and is mostly taken in usage with the aggregate functions like Count, Sum, etc.

LIKE CLAUSE :

- LIKE clause is used to compare a value to similar values using wildcard operators.
- There are two wildcards used in conjunction with the LIKE operator.
  1. The percent sign (%)
  2. The underscore (_)
- The percent sign represents zero, one or multiple characters.
- The underscore represents a single number or character.
- These symbols can be used in combinations.

Syntax :

```
SELECT FROM table_name
WHERE column LIKE 'XXXX%'
```

Sr.No.	Statement & Description
1	**WHERE SALARY LIKE '200%'** Finds any values that start with 200.
2	**WHERE SALARY LIKE '%200%'** Finds any values that have 200 in any position.
3	**WHERE SALARY LIKE '_00%'** Finds any values that have 00 in the second and third positions.
4	**WHERE SALARY LIKE '2_%_%'** Finds any values that start with 2 and are at least 3 characters in length.
5	**WHERE SALARY LIKE '%2'** Finds any values that end with 2.
6	**WHERE SALARY LIKE '_2%3'** Finds any values that have a 2 in the second position and end with a 3.

GROUP BY :

- The GROUP BY clause is used to arrange identical data into groups.
- The GROUP BY clause is used with the SELECT statement.
- It is used with aggregate functions like COUNT, MAX, MIN, SUM, AVG etc.
- Group by controls the presentation of tuples(rows).

Syntax :

```
SELECT column_Name
FROM table_Name
WHERE condition
GROUP BY column_Name;
```

ORDER BY :

- The ORDER BY clause is used to sort the result-set in ascending or descending order.
- The ORDER BY clause sorts the records in ascending order by default.
- To sort the records in descending order, use the DESC keyword.
- **ASC** denotes ascending order, while **DESC** denotes descending order.

Syntax :

```
SELECT expressions
FROM tables
[WHERE condition]
ORDER BY expression[asc/desc];
```

Difference between Groupby and Orderby :

GROUP BY	ORDER BY
Group by statement is used to group the rows that have the same value.	Order by statement sort the result-set either in ascending or in descending order.
It may be allowed in CREATE VIEW statement.	It does not use in CREATE VIEW statement.
In select statement, it is always used before the order by keyword.	In select statement, it is always used after the group by keyword.
It is used with aggregate functions.	It is used with aggregate functions.
Group by controls the presentation of rows.	Order by controls the presentation of columns.

Nested query :

- If a query is written inside a query is called nested queries.
- The result of inner query is used in execution of outer query.
- There are two types of nested queries:
  1. **Independent Nested Queries**
  2. **Co-related Nested Queries**

Independent Nested Queries :

- In this, query execution starts from innermost query to outermost queries.
- The execution of inner query is independent of outer query, but the result of inner query is used in execution of outer query.
- Various operators like IN, NOT IN, ANY, ALL etc are used in writing independent nested queries.

Co-related Nested Queries :

- In co-related nested queries, the output of inner query depends on the row which is being currently executed in outer query.

SQL Join :

- Joins is used to combine records from two or more tables in a database.
- It combines two or more tables based on a common field between them.
- The join is performed in the WHERE clause.
- Various operators can be used to join tables, such as =, <, >, <=, >=, !=, BETWEEN and LIKE.
- However, the most common operator is the equal(=) to symbol.
- There are different types of joins available in SQL :
  1. INNER JOIN – returns rows when there is a match in both tables.
  2. LEFT JOIN – returns all rows from the left table, even if there are no matches in the right table.
  3. RIGHT JOIN – returns all rows from the right table, even if there are no matches in the left table.
  4. FULL JOIN – returns rows when there is a match in one of the tables.
  5. SELF JOIN – is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.

Natural Join :

- Natural Join joins two tables based on same attribute name and datatypes.
- To perform natural join there must be one common attribute(Column) between two tables.
- Natural join will retrieve from multiple relations.
- Don't use ON clause in a natural join.

Syntax :

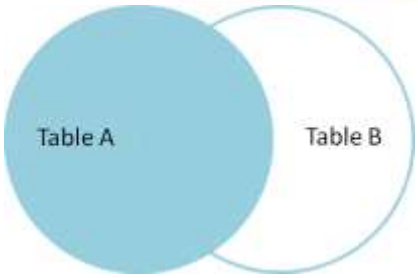
```
SELECT *
FROM TABLE1
NATURAL JOIN TABLE2;
```

Features of Natural Join :

- The columns must be the same data type.
- It will perform the Cartesian product.
- It finds consistent tuples and deletes inconsistent tuples.
- Then it deletes the duplicate attributes.

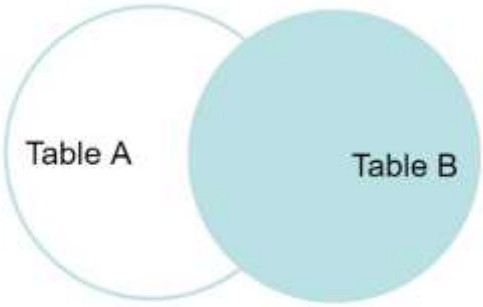
LEFT JOIN :

- Left join returns all the rows of the table on the left side and matches rows for the table on the right side.
- For the rows for which there is no matching row on the right side, the result-set will contain null.
- LEFT JOIN is also known as LEFT OUTER JOIN.



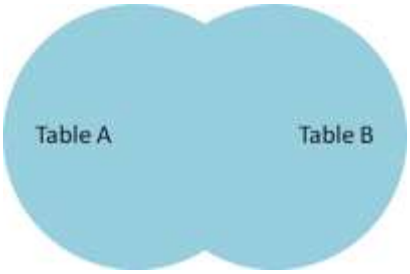
RIGHT JOIN :

- RIGHT JOIN is similar to LEFT JOIN.
- This join returns all the rows of the table on the right side and matching rows for the table on the left side.
- For the rows for which there is no matching row on the left side, the result-set will contain null.



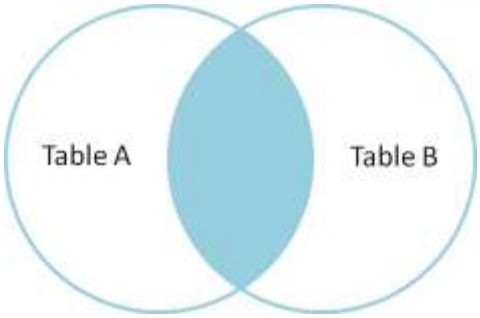
FULL JOIN :

- FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN.
- The result-set will contain all the rows from both tables.
- For the rows for which there is no matching, the result-set
- will contain NULL values.



INNER JOIN :

- The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied.
- This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.



SELF JOIN :

- A SELF JOIN is a join that is used to join a table with itself.
- If there is a need to combine data with other data in the same table itself.
- In that case, we use Self Join.

Difference between natural join and inner join :

NATURAL JOIN	INNER JOIN
Natural Join joins two tables based on same attribute name and datatypes.	Joins two table on the basis of the column which is explicitly specified in the ON clause.
In this, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column.	In Inner Join, The resulting table will contain all the attribute of both the tables including duplicate columns also.
Here, If there is no condition specifies then it returns rows based on the common column.	In Inner Join, only those records will return which exists in both the tables.
Don't use ON clause in a natural join.	Can be use ON clause in a inner join.
Syntax : SELECT * FROM table1 NATURAL JOIN table2;	**Syntax :** SELECT * FROM table1 INNER JOIN table2 ON table1.Column_Name = table2.Column_Name;

How to Backup and Restore MariaDB database :

Backup :

- It is very important to back up MariaDB databases, and databases in general.
- The database often contains most of a company's mission-critical data (sales, clients, etc.).
- Performing backups enables a system administrator to recover data after several types of events:
  - Operating system crash
  - Power failure
  - File system crash
  - Hardware problem

Restoring :

- A restore can be done with the command mysql :
- # mysql -u root -p inventory < /backup/mariadb.dump
- Here, **root** - User to connect with to restore the MariaDB backup (generally root or some other superuser) **-p** - Password for this user **inventory** - Selected database for restore backup **/backup/mariadb.dump** - Backup file

Unit-07 : NoSQL Database Technology

- Introduction to NoSQL database,
- Difference between relational & NoSQL database,
- NoSQL features, types, advantage,
- Architecture of MongoDB, documents, collections, dynamic schemas, Mongo shell, Mongo sever & client, data types, embedded documents, creating configuration file for Mongo.

Questions to be discussed :

1. What is NoSQL Database ? Explain its type.
2. Differentiate between relational database and NoSQL database.
3. Write the difference between SQL and NoSQL Database.
4. What is MongoDB ? What are the features of MongoDB ?
5. Difference between RDBMS and MongoDB :

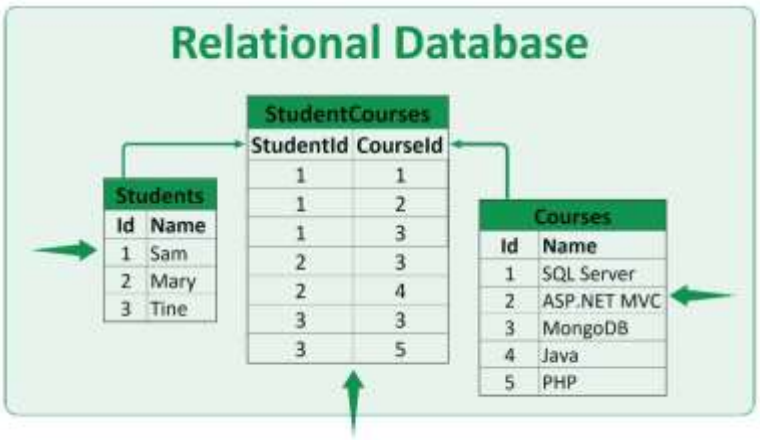
CTH EDUCATION SAHARSA ALOK SIR

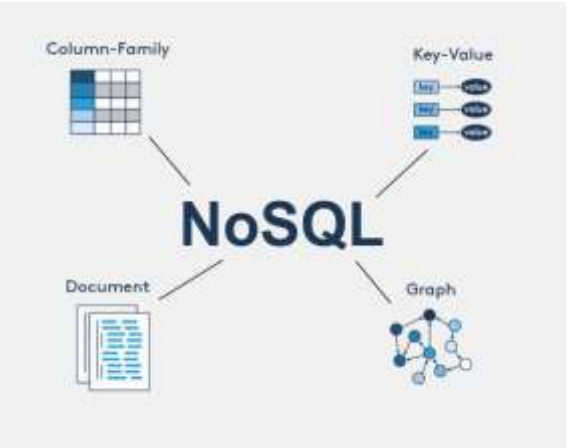
NoSQL Database :

- NoSQL Database stands for Non-SQL Database.
- NoSQL Database is a non-relational database.
- NoSQL database doesn't use tables for storing data.
- It provides a mechanism for storage and retrieval of data other than tabular model.
- In NoSQL database for storing data no any fixed structure is required, it stores in any manner.
- It is generally used to store big data and real-time web applications.
- MongoDB is a type NoSQL Database.

Differeneces between Relational database and NoSQL :

Relational Database	NoSQL Database
It stores data in tabular form.	It stores data in non-tabular form.
Relational database manages only structured data.	NoSQL database can manage structured, unstructured and semi-structured data.
Relational Database has a static schema.	NoSQL Database has dynamic schema.
Relational Database supports a powerful query language.	NoSQL Database supports a very simple query language.
Relational Database follows ACID properties.	NoSQL Database is only eventually consistent.
It supports complex transactions with joins.	It support only simple transactions.





Types of NoSQL databases :

There are four major types of NoSQL databases :

1. Document databases,
2. key-value databases,
3. Wide-column stores, and
4. Graph databases.

Document databases :

- Document databases store data in documents similar to JSON (JavaScript Object Notation) objects.
- Each document contains pairs of fields and values.
- The values can typically be a variety of types including things like strings, numbers, booleans, arrays, or objects.

Key-value databases :

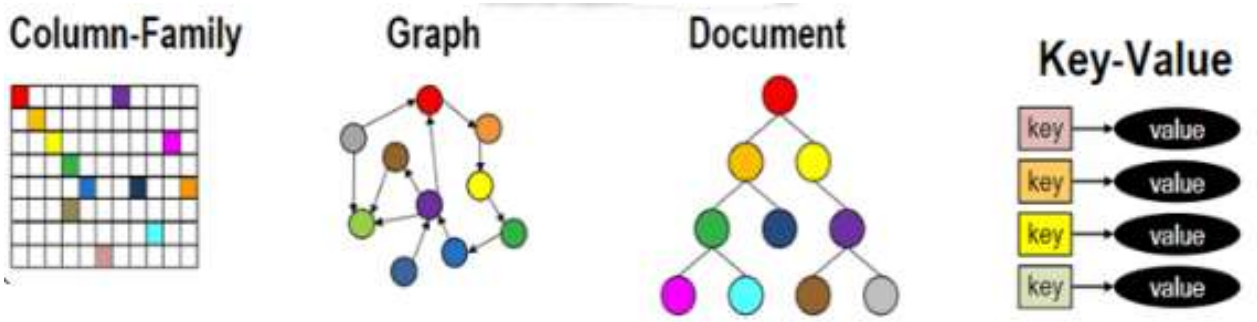
- Key-value databases are a simpler type of database where each item contains keys and values.

Wide-column stores :

- Wide-column stores store data in tables, rows, and dynamic columns.

Graph databases :

- Graph databases store data in nodes and edges.
- Nodes typically store information about people, places, and things, while edges store information about the relationships between the nodes.

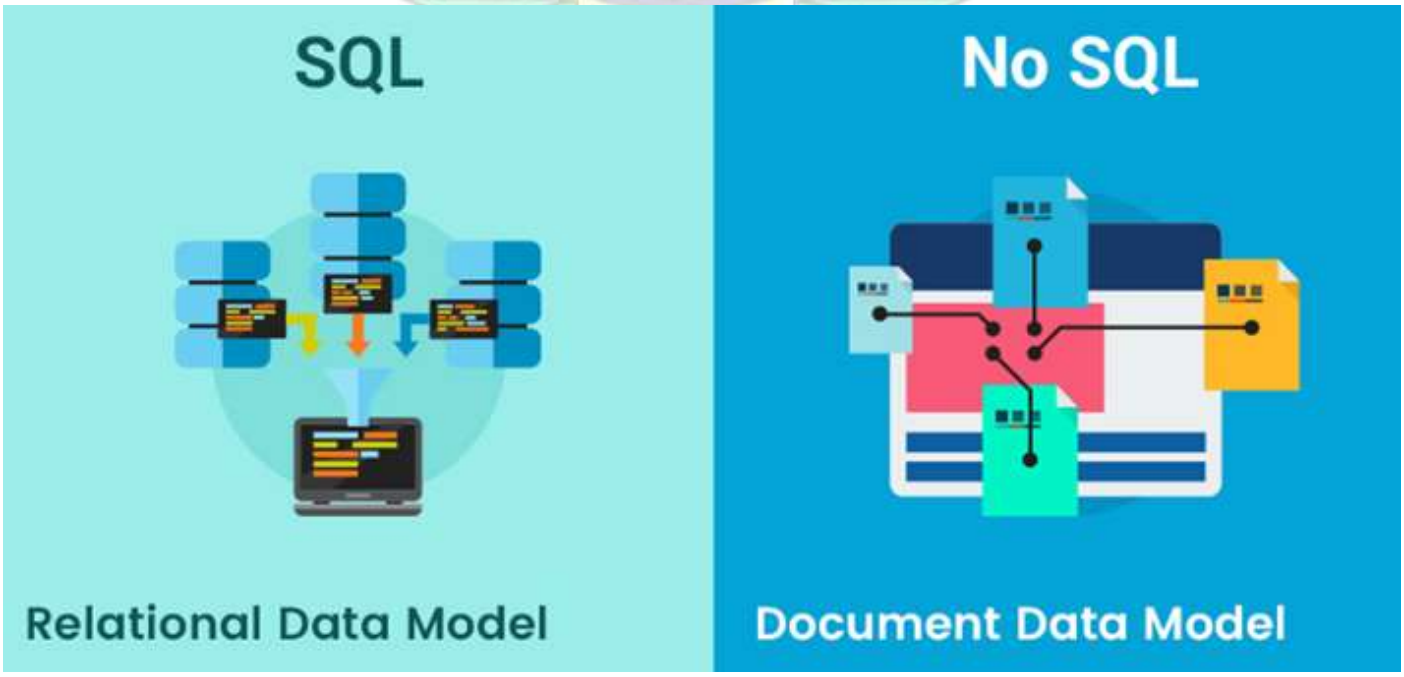


Advantages of NoSQL :

- It supports query language.
- It provides fast performance.
- It provides horizontal scalability.

Difference between SQL and NoSQL Database :

SQL Databases	NoSQL Databases
Tables with fixed rows and columns.	Document-oriented database.
These databases have fixed or static or predefined schema.	They have dynamic schema.
Developed in the 1970s with a focus on reducing data duplication.	Developed in the late 2000s with a focus on scaling and allowing for rapid application.
Oracle, MySQL, Microsoft SQL Server, and PostgreSQL	MongoDB ,GraphQL, HBase, Neo4j etc
Follows ACID property	Follows CAP(consistency, availability, partition tolerance)



What is MongoDB ?

- MongoDB is a document-oriented No SQL database.
- MongoDB is written in C++ programming language.
- MongoDB stores data as documents, so it is known as document-oriented database.
- It is an open-source document that provides high performance, high availability, and automatic scaling.
- It is developed and supported by a company named 10gen.
- MongoDB is available under General Public license(GPL) for free.
- The initial development of MongoDB began in 2007, but it is introduced in the market in 2009.
- MongoDB was developed by a company named 10gen which is now known as MongoDB Inc.
- MongoDB2.4.9 was the latest and stable version which was released on January 10, 2014.

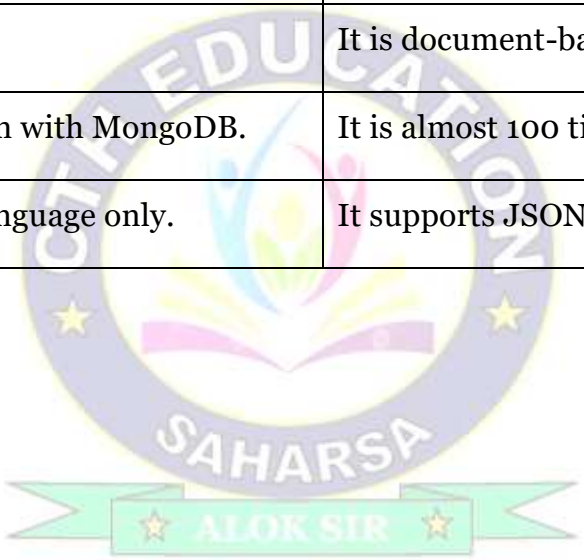
Features of MongoDB :

These are some important features of MongoDB:

- Support ad hoc queries
- Index Support
- Load balancing
- Provides high performance.
- Stores files of any size easily without complicating your stack.
- Easy to administer in the case of failures.
- It also supports JSON data model with dynamic schemas
- Now a day many companies using MongoDB to create new types of applications, improve performance and availability.

Difference between Relational DB and MongoDB :

Relational DB	MongoDB
It is a relational database.	It is a non-relational and document-oriented database.
Not suitable for hierarchical data storage.	Suitable for hierarchical data storage.
It has a predefined schema.	It has a dynamic schema.
It centers around ACID properties. (Atomicity, Consistency, Isolation, and Durability).	It centers around the CAP theorem (Consistency, Availability, and Partition tolerance).
It is row-based.	It is document-based.
It is slower in comparison with MongoDB.	It is almost 100 times faster than RDBMS.
It supports SQL query language only.	It supports JSON query language.



Unit-08 : Selecting the Right Database

- Selecting of right database, RDBMS or NoSQL,
- Selection of database based on :
  - Performance,
  - Data size & type of data,
  - Frequency of accessing data,
  - Business needs,
  - Type of application.

How to choose the right database for your service :

- Choosing which database to use is one of the most important decisions you can make when working on a new micro service.
- If you realize down the line that you've made the wrong choice, migrating to another database is a very costly and risky procedure.
- Each database technology (and type) has advantages and disadvantages.
- Consider the following criteria for choosing the right database technology for your service:
  - Query Patterns
  - Consistency
  - Storage Capacity
  - Performance
  - Maturity and Stability
  - Business need
  - Cost

Q. If you have selecting from RDBMS and NoSQL then which database is right ?

Ans : NoSQL