Ques: What is the objective to study DBMS?

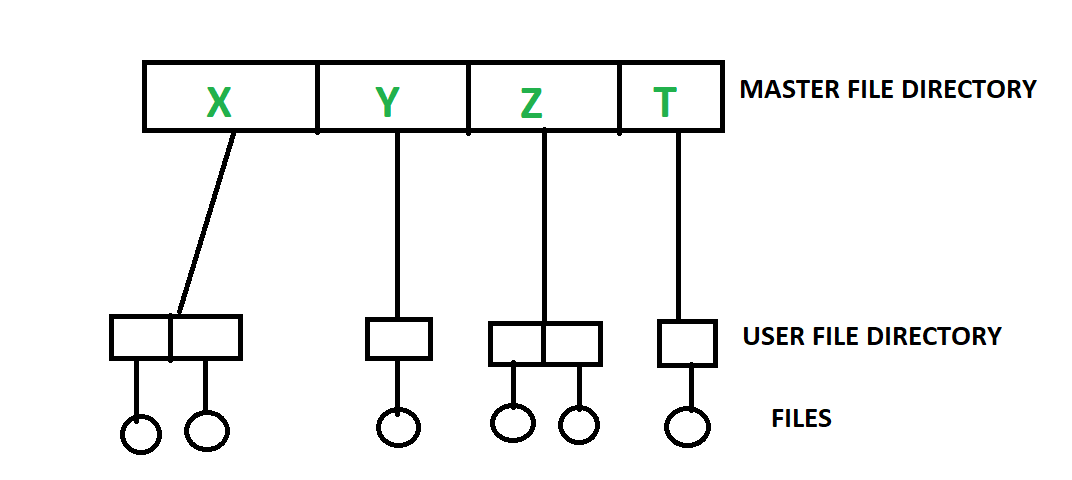
Ans: The primary goal of a DBMS is **to provide an environment that is both convenient and efficient to use in retrieving and storing data base information**. The DBMS is the interface between the user of application programs on one hand and the data base on the other.

Ques: The difference between file system and database system.

Ans:

**File System :**   
The file system is basically a way of arranging the files in a storage medium like a hard disk. The file system organizes the files and helps in the retrieval of files when they are required. File systems consist of different files which are grouped into directories. The directories further contain other folders and files. The file system performs basic operations like management, file naming, giving access rules, etc.

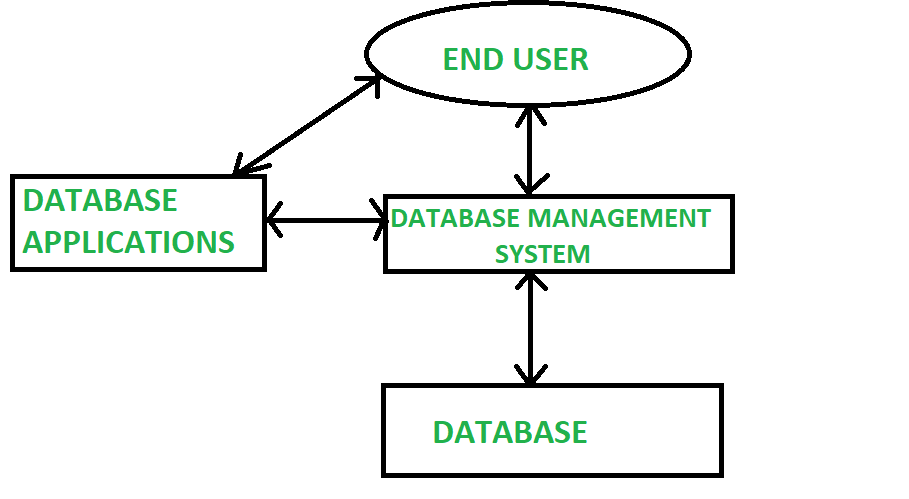
**Example:** NTFS(New Technology File System), EXT(Extended File System).



**DBMS(Database Management System) :**   
Database Management System is basically software that manages the collection of related data. It is used for storing data and retrieving the data effectively when it is needed. It also provides proper security measures for protecting the data from unauthorized access. In Database Management System the data can be fetched by SQL queries and relational algebra. It also provides mechanisms for data recovery and data backup.

**Example:**

Oracle, MySQL, MS SQL server.



Alternative solution :

**Difference between File System and DBMS:**

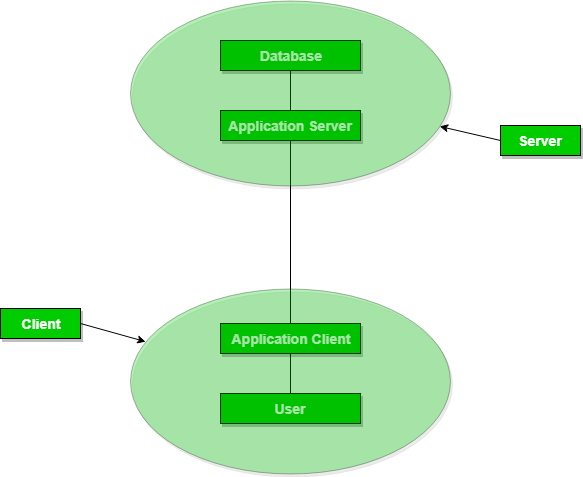
| Basis | File System | DBMS |
| --- | --- | --- |
| Structure | The file system is software that manages and organizes the files in a storage medium within a computer. | DBMS is software for managing the database. |
| Data Redundancy | Redundant data can be present in a file system. | In DBMS there is no redundant data. |
| Backup and Recovery | It doesn’t provide backup and recovery of data if it is lost. | It provides backup and recovery of data even if it is lost. |
| Query processing | There is no efficient query processing in the file system. | Efficient query processing is there in DBMS. |
| Consistency | There is less data consistency in the file system. | There is more data consistency because of the process of normalization. |
| Complexity | It is less complex as compared to DBMS. | It has more complexity in handling as compared to the file system. |
| Security Constraints | File systems provide less security in comparison to DBMS. | DBMS has more security mechanisms as compared to file systems. |
| Cost | It is less expensive than DBMS. | It has a comparatively higher cost than a file system. |
| Data Independence | There is no data independence. | In DBMS data independence exists. |
| User Access | Only one user can access data at a time. | Multiple users can access data at a time. |
| Meaning | The user has to write procedures for managing databases | The user not required to write procedures. |
| Sharing | Data is distributed in many files. So, not easy to share data | Due to centralized nature sharing is easy |
| Data Abstraction | It give details of storage and representation of data | It hides the internal details of Database |
| Integrity Constraints | Integrity Constraints are difficult to implement | Integrity constraints are easy to implement |
| Example | Cobol, C++ | Oracle, SQL Server |

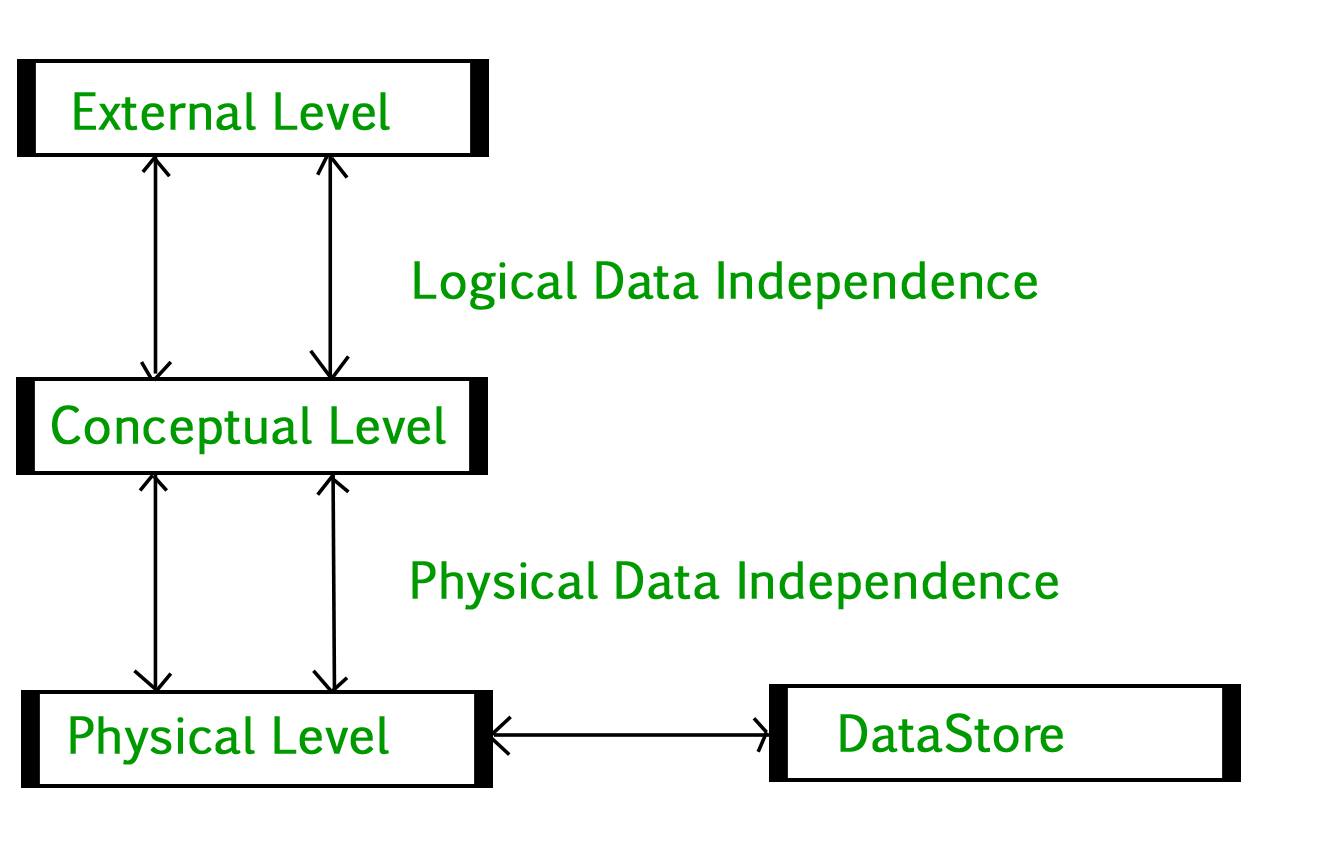
Give examples DBMS?

Ans: Some examples of popular database software or DBMSs include **MySQL, Microsoft Access, Oracle Database and Microsoft SQL**

Ques: Three-tier architecture of DBMS.

Ans: In this type, there is another layer between the client and the server. The client does not directly communicate with the server. Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management 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. as shown below:



Note: DBMS 3-tier architecture divides the complete system into three inter-related but independent modules. As shown below: 

1. **Physical Level:** At the physical level, the information about the location of database objects in the data store is kept. Various users of DBMS are unaware of the locations of these objects .In simple terms ,physical level of a database describes how the data is being stored in secondary storage devices like disks and tapes and also gives insights on additional storage details.
2. **Conceptual Level:**At conceptual level, data is represented in the form of various database tables. For Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware of their storage .Also referred as logical schema ,it describes what kind of data is to be stored in the database.
3. **External Level:** An external level specifies a view of the data in terms of conceptual level tables.  Each external level view is used to cater to the needs of a particular category of users. For Example, FACULTY of a university is interested in looking course details of students, STUDENTS are interested in looking at all details related to academics, accounts, courses and hostel details as well. So, different views can be generated for different users. The main focus of external level is data abstraction.

Ques: Two-tier architecture of DBMS

Ans: The overall design of the Database Management System (DBMS) depends on its architecture. A large amount of data on web servers, Personal Computers (PC) and other elements are linked with networks with the help of basic client or server architecture.

PCs and workstations are part of Client architecture that are connected over the network. The architecture of DBMS depends on how the users are linked to the database.

There are three kinds of DBMS Architecture, which are as follows −

* Tier-1 Architecture.
* Tier-2 Architecture.
* Tier-3 Architecture.

### Tier-2 Architecture

The 2-tier Architecture is based on a client-server machine.

and this type of architecture, the applications on client-side interact directly with the database present at the server-side.

This interaction between client and server uses Application Program Interface like ODBC and JDBC.

* ODBC − Open Database Connectivity
* JDBC − Java Database Connectivity

When there are a large number of users at client side to access the database, this architecture gives a poor performance.

The server side is responsible for delivering the functionalities like query processing and management of transactions.

For example − Oracle, Sybase, Microsoft SQL Server etc.

The Tier-2 architecture of DBMS is diagrammatically represented as follows −



Ques: One-tier architecture of DBMS

Ans: In One-Tier Architecture the database is directly available to the user, the user can directly sit on the DBMS and use it i.e.; the client, server, and the Database are all present on the same machine. For Example- To learn SQL we set up an SQL server and the database on the local system. This enables us to directly interact with the relational database and execute operations.

Note:

The industry won’t use this architecture they logically go for 2-Tier and 3-Tier Architecture.

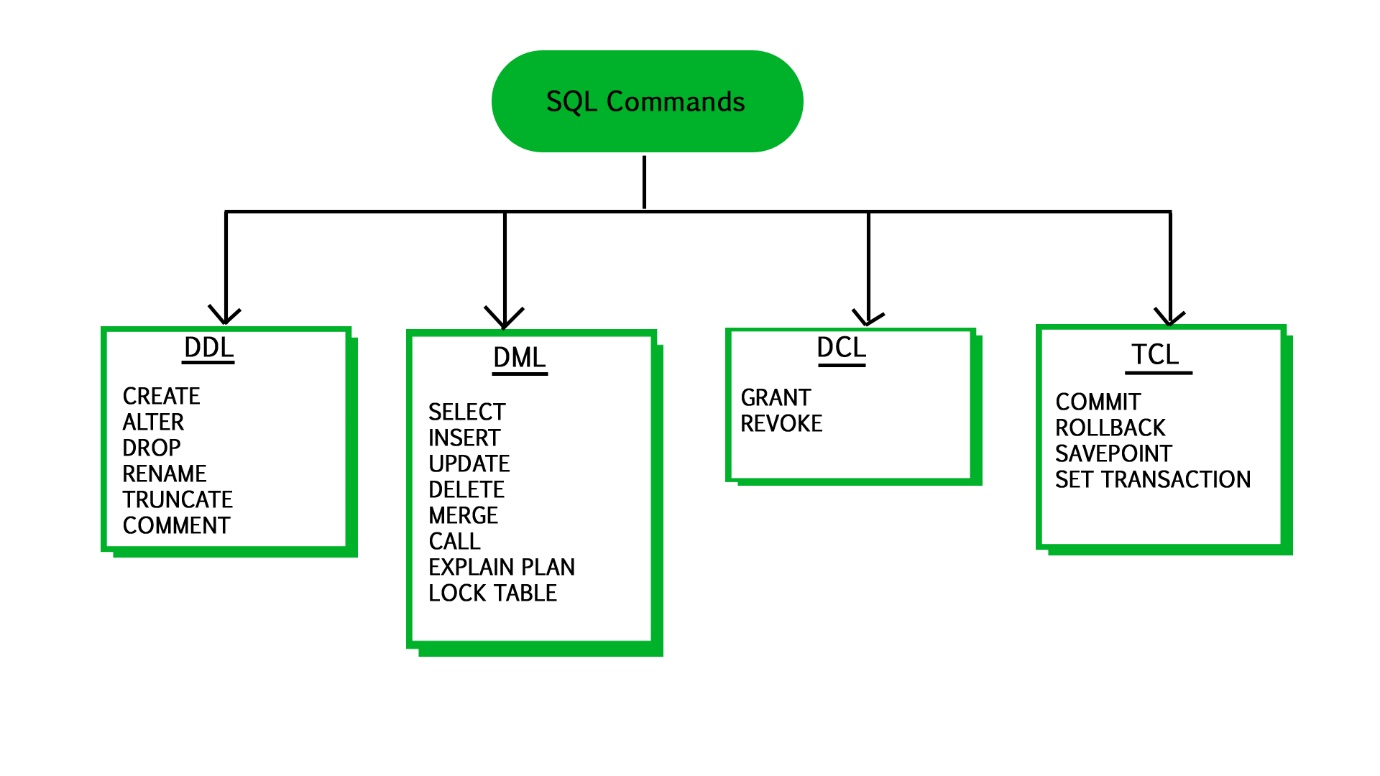
Ques: Write are the Full form of DDL, DML and DCL?

Ans:

DDL : Data Definition Language

DML: Data Manipulation Language

TCL: Transaction Control Language

DCL: Data Control Language. 

Ques: Explain DDL, DML, DCL and TCL.

## Ans: What is SQL?

**SQL** is a database language designed for the retrieval and management of data in a relational database . SQL is the standard language for database management. All the RDBMS systems like MySQL, MS Access, Oracle, Sybase, Postgres, and SQL Server use SQL as their standard database language. SQL programming language uses various commands for different operations. We will learn about the like DCL, TCL, DQL, DDL and DML commands in SQL with examples.

## Why Use SQL?

Here, are important reasons for using SQL

* It helps users to access data in the RDBMS system.
* It helps you to describe the data.
* It allows you to define the data in a database and manipulate that specific data.
* With the help of SQL commands in DBMS, you can create and drop databases and tables.
* SQL offers you to use the function in a database, create a view, and stored procedure.
* You can set permissions on tables, procedures, and views.

## Brief History of SQL

Here, are important landmarks from the history of SQL:

* 1970 – Dr. Edgar F. “Ted” Codd described a relational model for databases.
* 1974 – Structured Query Language appeared.
* 1978 – IBM released a product called System/R.
* 1986 – IBM developed the prototype of a relational database, which is standardized by ANSI.
* 1989- First ever version launched of SQL
* 1999 – SQL 3 launched with features like triggers, object-orientation, etc.
* SQL2003- window functions, XML-related features, etc.
* SQL2006- Support for XML Query Language
* SQL2011-improved support for temporal databases

## Types of SQL

Here are five types of widely used SQL queries.

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

## What is DDL?

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:

### CREATE

CREATE statements is used to define the database structure schema:

**Syntax:**

CREATE TABLE TABLE\_NAME (COLUMN\_NAME DATATYPES[,....]);

**For example**:

Create database university;

Create table students;

Create view for\_students;

### DROP

Drops commands remove tables and databases from RDBMS.

Syntax

DROP TABLE ;

**For example:**

Drop object\_type object\_name;

Drop database university;

Drop table student;

### ALTER

Alters command allows you to alter the structure of the database.

**Syntax:**

To add a new column in the table

ALTER TABLE table\_name ADD column\_name COLUMN-definition;

To modify an existing column in the table:

ALTER TABLE MODIFY(COLUMN DEFINITION....);

**For example:**

Alter table guru99 add subject varchar;

### TRUNCATE:

This command used to delete all the rows from the table and free the space containing the table.

**Syntax:**

TRUNCATE TABLE table\_name;

**Example:**

TRUNCATE table students;

## What is 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.

There are three basic constructs which allow database program and user to enter data and information are:

Here are some important DML commands in SQL:

* INSERT
* UPDATE
* DELETE

### INSERT:

This is a statement is a SQL query. This command is used to insert data into the row of a table.

**Syntax:**

INSERT INTO TABLE\_NAME (col1, col2, col3,.... col N)

VALUES (value1, value2, value3, .... valueN);

Or

INSERT INTO TABLE\_NAME

VALUES (value1, value2, value3, .... valueN);

**For example:**

INSERT INTO students (RollNo, FIrstName, LastName) VALUES ('60', 'Tom', Erichsen');

### UPDATE:

This command is used to update or modify the value of a column in the table.

**Syntax:**

UPDATE table\_name SET [column\_name1= value1,...column\_nameN = valueN] [WHERE CONDITION]

**For example:**

UPDATE students

SET FirstName = 'Jhon', LastName= 'Wick'

WHERE StudID = 3;

### DELETE:

This command is used to remove one or more rows from a table.

**Syntax:**

DELETE FROM table\_name [WHERE condition];

**For example:**

DELETE FROM students

WHERE FirstName = 'Jhon';

## What is DCL?

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.

### Examples of DCL commands:

Commands that come under DCL:

* Grant
* Revoke

### Grant:

This command is use to give user access privileges to a database.

**Syntax:**

GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;

**For example:**

GRANT SELECT ON Users TO'Tom'@'localhost;

### Revoke:

It is useful to back permissions from the user.

**Syntax:**

REVOKE privilege\_nameON object\_nameFROM {user\_name |PUBLIC |role\_name}

**For example:**

REVOKE SELECT, UPDATE ON student FROM BCA, MCA;

## What is TCL?

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

### Commit

This command is used to save all the transactions to the database.

**Syntax:**

Commit;

**For example:**

DELETE FROM Students

WHERE RollNo =25;

COMMIT;

### Rollback

Rollback command allows you to undo transactions that have not already been saved to the database.

**Syntax:**

ROLLBACK;

**Example:**

DELETE FROM Students

WHERE RollNo =25;

### SAVEPOINT

This command helps you to sets a savepoint within a transaction.

**Syntax:**

SAVEPOINT SAVEPOINT\_NAME;

**Example:**

SAVEPOINT RollNo;

## What is DQL?

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

### SELECT:

This command helps you to select the attribute based on the condition described by the WHERE clause.

**Syntax:**

SELECT expressions

FROM TABLES

WHERE conditions;

**For example:**

SELECT FirstName

FROM Student

WHERE RollNo > 15;

* .

Unit- 2

Ques: Write SQL query to get the Student name whose salary is between 1000 and 2000.

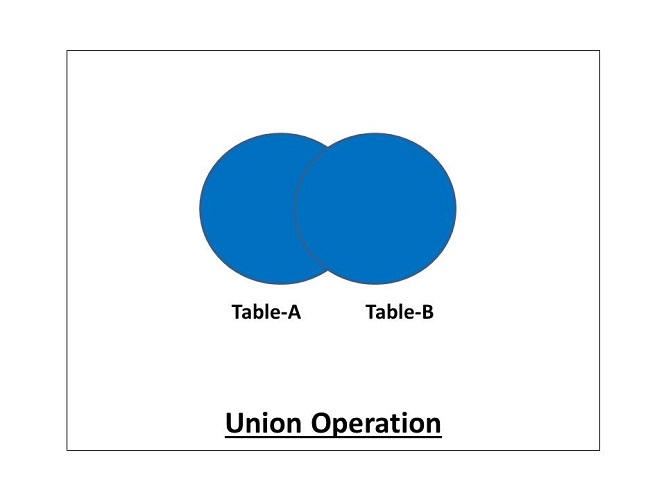
Ques. Explain Union and Intersection in DBMS.

#### Ans: UNION

**The Union is a binary set operator in DBMS. It is used to combine the result set of two select queries.**Thus, It combines two result sets into one. In other words, the result set obtained after union operation is the collection of the result set of both the tables.

But two necessary conditions need to be fulfilled when we use the union command. These are:

1. Both SELECT statements should have an equal number of fields in the same order.
2. The data types of these fields should either be the same or compatible with each other.

The Union operation can be demonstrated as follows: 

The syntax for the union operation is as follows:

SELECT (coloumn\_names) from table1 [WHERE condition] UNION **SELECT** (coloumn\_names) from table2 [WHERE condition];

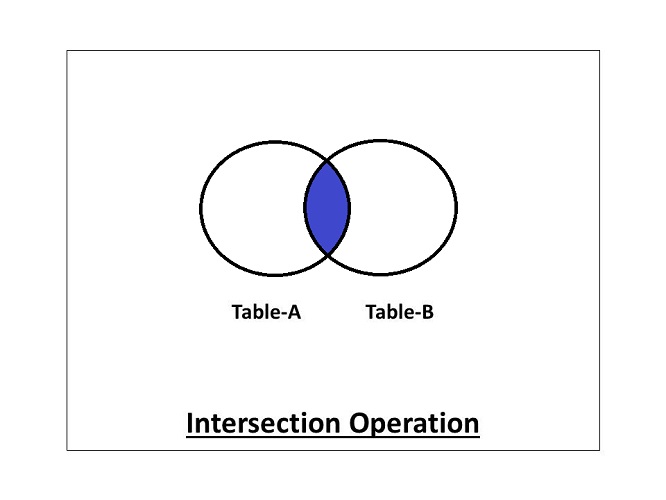
The MySQL query for the union operation can be as follows:

SELECT color\_name FROM colors\_a UNION SELECT color\_name FROM colors\_b;

#### INTERSECTION:

**Intersection is a binary set operator in DBMS. The intersection operation between two selections returns only the common data sets or rows between them.**It should be noted that the intersection operation always returns the distinct rows. The duplicate rows will not be returned by the intersect operator.

The intersection operation can be demonstrated as follows:



The syntax for the intersection operation is as follows:

SELECT (coloumn\_names) from table1[WHERE condition] INTERSECT **SELECT** (coloumn\_names) from table2 [WHERE condition];

Ques. Explain join in DBMS.

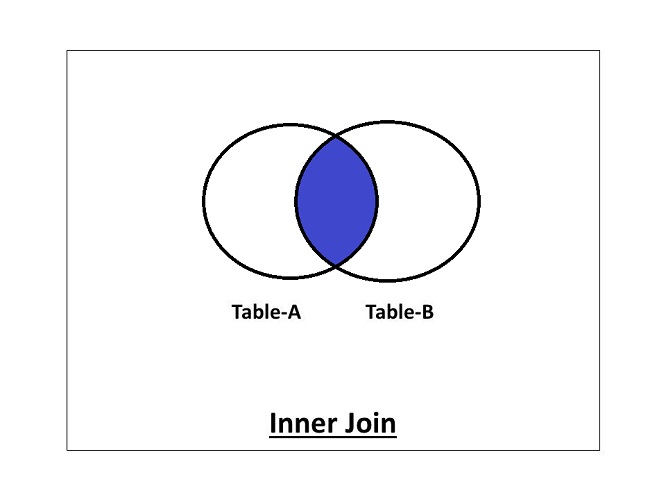
Ans : **In DBMS, a join statement is mainly used to combine two tables based on a specified common field between them.**If we talk in terms of Relational algebra, it is the cartesian product of two tables followed by the selection operation. Thus, we can execute the product and selection process on two tables using a single join statement. We can use either 'on' or 'using' clause in MySQL to apply predicates to the join queries.

**A Join can be broadly divided into two types:**

1. **Inner Join**
2. **Outer Join**

#### Inner Join:

**Inner Join is a join that can be used to return all the values that have matching values in both the tables.**Inner Join can be depicted using the below diagram.



**The inner join can be further divided into the following types:**

1. **Equi Join**
2. **Natural Join**

Now let us learn about these inner joins one-by-one.

**1. Equi Join**

**Equi Join is an inner join that uses the equivalence condition for fetching the values of two tables.**

The MySQL query for equi join can be as follows:

Select employee.empId, employee.empName, department.deptName from employee Inner Join department on employee.deptId = department.deptId;

**2. Natural Join**

**Natural Join is an inner join that returns the values of the two tables on the basis of a common attribute that has the same name and domain.**It does not use any comparison operator. It also removes the duplicate attribute from the results.

The MySQL query for natural join can be as follows:

Select \* from employee Natural Join department;

The above query will return the values of tables removing the duplicates. If we want to specify the attribute names, the query will be as follows:

Select employee.empId, employee.empName, department.deptId, department.deptName from employee Natural Join department;

#### Outer Join

**Outer Join is a join that can be used to return the records in both the tables whether it has matching records in both the tables or not.**

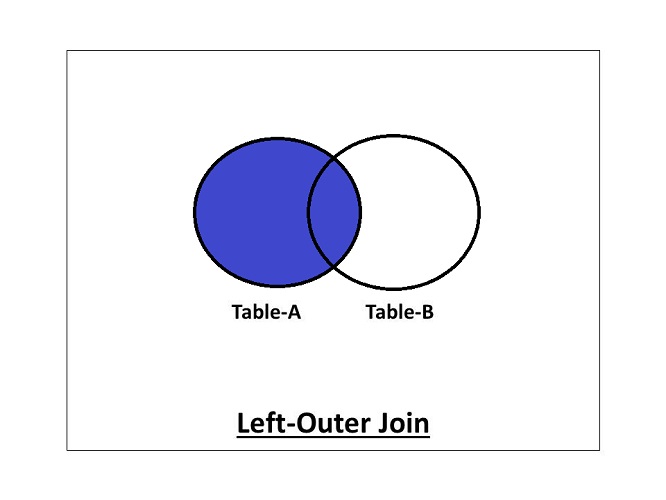
***The outer join can be further divided into three types:***

1. **Left-Outer Join**
2. **Right-Outer Join**
3. **Full-Outer Join**

we'll learn about these outer joins one-by-one.

**1. Left-Outer Join:**

**The Left-Outer Join is an outer join that returns all the values of the left table, and the values of the right table that has matching values in the left table.**If there is no matching result in the right table, it will return null values in that field. The Left-Outer Join can be depicted using the below diagram.

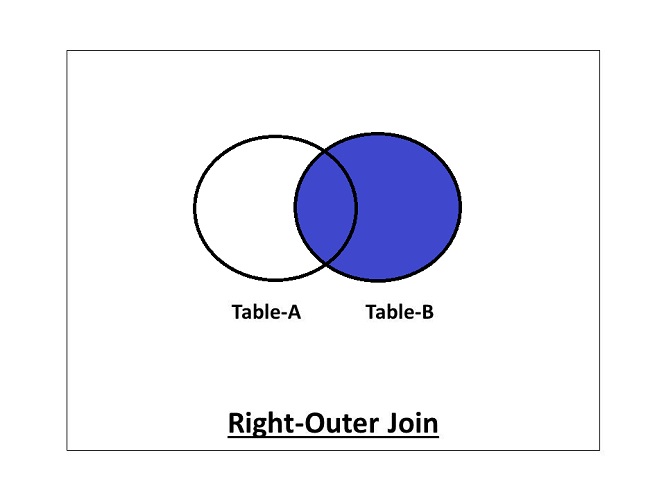


The MySQL query for left-outer join can be as follows:

Select employee.empId, employee.empName, department.deptName from employee Left Outer Join department on employee.deptId = department.deptId;

**2. Right-Outer Join:**

**The Right-Outer Join is an outer join that returns all the values of the right table, and the values of the left table that has matching values in the right table.**The Right-Outer Join can be depicted using the below diagram.

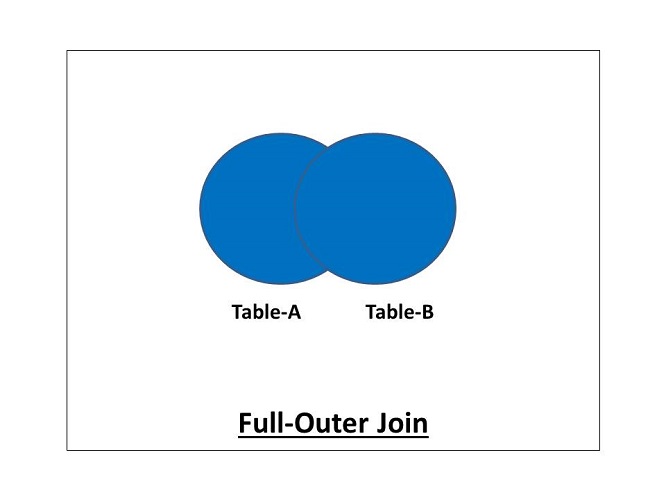


The MySQL query for right-outer join can be as follows:

Select employee.empId, employee.empName, department.deptName from employee Right Outer Join department on employee.deptId = department.deptId;

**3. Full-Outer Join:**

**The Full-Outer join contains all the values of both the tables whether they have matching values in them or not.**The Full-Outer Join can be depicted using the below diagram.



The MySQL query for full-outer join can be as follows:

Select \* from employee Full Join department;

Ques: Explain Cartesian Product in DBMS with example.

Ans:  Cartesian product is the result of joining every row in one table with every row in another table. This occurs when there is no WHERE clause to restrict rows. While this is legitimate in some cases, most occurrences of a Cartesian product are mistakes. The Cartesian product is somewhat analogous to a correlated subquery—that is, it performs a full table scan on the second table for every row in the first table. An example of a Cartesian product is:

It combines R1 and R2 without any condition. It is denoted by X.

Degree of R1 XR2 = degree of R1 + degree of R2

{degree = total no of columns}

### Example

Consider R1 table −

| **egNo** | **Branch** | **Section** |
| --- | --- | --- |
| 1 | CSE | A |
| 2 | ECE | B |
| 3 | CIVIL | A |
| 4 | IT | B |

Table R2

| **Name** | **RegNo** |
| --- | --- |
| Bhanu | 2 |
| Priya | 4 |

R1 X R2

| **RegNo** | **Branch** | **Section** | **Name** | **RegNo** |
| --- | --- | --- | --- | --- |
| 1 | CSE | A | Bhanu | 2 |
| 1 | CSE | A | Priya | 4 |
| 2 | ECE | B | Bhanu | 2 |
| 2 | ECE | B | Priya | 4 |
| 3 | CIVIL | A | Bhanu | 2 |
| 3 | CIVIL | A | Priya | 4 |
| 4 | IT | B | Bhanu | 2 |
| 4 | IT | B | Priya | 4 |

Ques: Explain Select and Project.

Ans: **Select Operation :** This operation is used to select rows from a table (relation) that specifies a given logic, which is called as a predicate. The predicate is a user defined condition to select rows of user's choice.

**Project Operation :** If the user is interested in selecting the values of a few attributes, rather than selection all attributes of the Table (Relation), then one should go for PROJECT Operation.

Select retrieves the tuples (rows) in a relation (table) for which the condition in 'predicate' section (WHERE clause) stands true.

Project retrieves the attributes (columns) specified.