

Answer Key

Touchpad MODULAR SQL (Ver. 2.0)

1. Introduction to DBMS

EXERCISE



- A.** 1. c 2. a 3. b 4. d 5. d
- B.** 1. T 2. F 3. T 4. F 5. T
- C.** 1. DBMS 2. field 3. E.F. Codd 4. record 5. Master Table
- D.** 1. A database is defined as an organised collection of data that can be visualised as a container of information. With the help of a database, data can be stored, modified, and retrieved.
2. A column in a table is also called an attribute.
3. SQL is a high-level interactive language that allows users to specify what is required to be done in the form of queries.
4. A table is a collection of logically related records. The multiple records of a database are arranged together in a tabular structure to make a table.
5. A record is a collection of multiple data which is stored in related fields that can be treated as a single unit.
- E.** 1. A Database Management System (DBMS) is a software package that controls the creation, maintenance, and use of a database. It performs the tasks of maintaining databases so that the information is readily available. It helps to manage the data of different fields like banks, offices, railways, airlines, etc.
2. Microsoft Access, Microsoft SQL Server, Oracle, MySQL, IBM DB2, PostgreSQL, etc. are some of the most commonly used relational database management systems.
3. • Primary key: Primary key is a field which is used to uniquely identify records in a database. It is a unique field and it cannot be left blank. There can be only one primary key in a table. If there are more than one primary keys, then there will be two identities against a row, like a student has two roll numbers in a class. There can be only one identity against a row. So, you cannot have more than one primary key. To make retrieval of records faster, Primary keys are indexed in database.



- Foreign Key: The main table of a database is referred to as the 'Master Table' and the tables in which the related data is stored are called 'Transaction Table'. The tables are related and linked through a field which is common. This common key field in the transaction table is called the 'Foreign key' and its value is related to the primary key values of the master table.

4. Three important features of SQL are as follows:

- It has very high performance and efficiency.
- It is easy to use.
- It is available free of cost.

5. • DBMS: A Database Management System (DBMS) is a software package that controls the creation, maintenance, and use of a database. It performs the tasks of maintaining databases so that the information is readily available. It helps to manage the data of different fields like banks, offices, railways, airlines, etc.

- RDBMS: A relational database is a type of database that stores data in several tables. These tables may interlinked to get a common piece of information. The software package that is used to handle relational databases is known as the Relational Database Management System (RDBMS). Microsoft Access, Microsoft SQL Server, Oracle, MySQL, IBM DB2, PostgreSQL, etc. are some of the most commonly used relational database management systems.

F. 1. Primary Key

2. She should use DML commands. Common DML statements are: SELECT, INSERT INTO, UPDATE, DELETE

IN THE LAB

Do yourself.

2. Introduction to MYSQL

EXERCISE



- A.** 1. d 2. b 3. c 4. b 5. b
- B.** 1. DESC 2. Table 3. One 4. Modifying 5. Constraints
- C.** 1. T 2. T 3. F 4. F 5. F
- D.** 1. A table is a collection of organised data in the form of rows and columns.
2. SQL constraints are used to define rules for table data.
3. The syntax of the DROP TABLE command is as follows:

```
DROP TABLE table_name;
```



4. Syntax to add a column is as follows:

```
ALTER TABLE table
```

```
ADD [COLUMN] column_name datatype [FIRST|AFTER existing_column];
```

5. The TIME data type is used to store time values such as hour, minute, and second in a database table.

E. 1. MySQL is an open-source RDBMS software that is available free of cost. It can run on major operating systems, including Linux, Unix and Windows. It is popular for web-based applications and online publishing. It uses a query language called SQL (Structured Query Language).

2. The constraints used in MySQL are:

- NULL: The NULL constraint allows a column to store the NULL value, which means that if the column does not receive any value during insert or update operations, it will accept the values of the other columns in insert operations without showing any error.
- NOT NULL: The NOT NULL constraint specifies that a column cannot accept NULL values. You cannot insert a new record, or update a record without specifying a value to this column.
- UNIQUE: The UNIQUE constraint ensures that each value in a column is unique. The UNIQUE and PRIMARY KEY constraints both ensure that a column or set of columns is unique.
- PRIMARY KEY: Each record in a table is uniquely identified by the PRIMARY KEY constraint. Primary key must have unique values and cannot contain NULL values. A table can only have one primary key, which can be composed of single or multiple columns (fields).
- FOREIGN KEY: The FOREIGN KEY constraint is used to establish a relationship between two tables. It ensures that the value in the referencing table column matches a value in the referenced table's primary key column.
- Composite Primary Key: When a table requires unique identification for each row but one column is unable to fulfill the requirement of uniqueness, in that case two or more columns can be used for making a primary key.
- CHECK: The CHECK constraint is used to specify a default value for a column. If no other value is specified, the default value will be added to all new records.
- DEFAULT: The CHECK constraint restricts the value range that can be placed in a column.

3. The DROP TABLE command is used to delete a table from a database. The syntax of the DROP TABLE command is as follows:

```
DROP TABLE table_name;
```

4. The MODIFY COLUMN clause is used with the ALTER TABLE command to change the data type of a column. The syntax to change the data type of a column is as follows:

```
ALTER TABLE table_name MODIFY COLUMN column_name datatype;
```

5. Five data types in SQL are listed below:

DATA TYPE	DESCRIPTION
INTEGER or INT	Stores integer values (no decimal).
CHAR (n)	Character string with fixed length of n.
VARCHAR (n)	Character string with variable length of n.
DECIMAL (p,s)	Stores decimals values where 'p' is precision value (total no. of digits) and 's' is scale value (digits after decimal).
DATE	Stores date in the format of YYYY-MM-DD values.

F. 1. Syntax to remove a field (column) from a table:

```
ALTER TABLE table_name DROP COLUMN column_name;
```

2. Constraint used to restrict value range: CHECK constraint

IN THE LAB

Do yourself.

3. Handling Records in MySQL

EXERCISE



- A.** 1. c 2. d 3. a 4. c 5. d
- B.** 1. Insert 2. Select 3. % 4. DISTINCT 5. TRUE
- C.** 1. T 2. F 3. F 4. F 5. F
- D.** 1. The SELECT command retrieves zero or more rows from the table. It helps us to join information from different tables and filter specific information as per the required criteria.
2. The syntax of the DISTINCT clause is as follows:

```
SELECT DISTINCT <column_name> FROM <table_name>;
```

3. SQL statements consist of reserved words or symbols which are used to perform arithmetical operations, comparisons, etc. These reserved words or symbols are called 'Operators'.
4. The LIKE operator in SQL searches for a character string that matches the specified pattern using wildcards in a column.
5. The syntax of SELECT query in SQL:

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



- E. 1. Records are inserted using the INSERT INTO command, which adds new data into a table.

Example:

```
INSERT INTO Students (StudentID, Name, Marks)
VALUES (1, 'Amit', 85);
```

2. The UPDATE command is used to modify existing records in a table.

Syntax:

```
UPDATE table_name
SET column1 = value1, column2 = value2
WHERE condition;
```

3. Five relational operators with example are as follows:.

Operator	Description	Example
=	Checks if value of 'a' is equal to value of 'b'. If yes, condition becomes true, else false.	(a = b) is false
!=	Checks if value of 'a' is not equal to value of 'b'. If yes, condition becomes true, else false.	(a != b) is true
<>	Checks if value of 'a' is equal to value of 'b' or not. If yes, condition becomes true, else false.	(a <> b) is true
<	Checks if value of 'a' is less than value of 'b'. If yes, condition becomes true, else false.	(a < b) is false
>	Checks if value of 'a' is greater than value of 'b'. If yes, condition becomes true, else false.	(a > b) is true

4. BETWEEN operator is used to indicate a range which otherwise is done using relational operator.

5. SELECT: The SELECT command retrieves zero or more rows from the table. It helps us to join information from different tables and filter specific information as per the required criteria.

The syntax to use the SELECT command:

```
SELECT * FROM table_name;
```

Where * is used to select all the columns of the specified table.

UPDATE: Sometimes, you may need to change the inserted records into a table. In that case, the UPDATE command will work. The syntax of the UPDATE command is:

```
UPDATE table_name SET field_change=value WHERE field_criteria=value;
```

- F. 1. She can add the marks of her friend using the INSERT INTO command.

Example:

```
INSERT INTO MarksTable VALUES (5, 'FriendName', 78);
```



2. She should use the LIKE operator to retrieve the capital of a state (when searching or matching values in SQL queries).

Example:

```
SELECT Capital FROM States  
WHERE StateName LIKE 'Delhi';
```

IN THE LAB

Do yourself.

Test Sheet 1

(Based on chapters 1 to 3)

Section A

- | | | | | | | |
|-----------|---------------|-----------------|----------|----------------|----------------|------|
| A. | 1. a | 2. d | 3. d | 4. b | 5. d | 6. d |
| B. | 1. F | 2. F | 3. T | 4. T | 5. F | 6. T |
| C. | 1. E. F. Codd | 2. Master Table | 3. Table | 4. Constraints | 5. Primary key | 6. % |

Section B

- A.**
1. In a relational database, when two tables are linked through a common field, the common field in the second table, which references the primary key of the first table, is called a foreign key.
 2. A record is a collection of multiple data which is stored in related fields that can be treated as a single unit.
 3. A table is a collection of logically related records.
 4. The syntax of the DROP TABLE command is as follows:

```
DROP TABLE table_name;
```
 5. SQL statements consist of reserved words or characters used to perform arithmetical operations, comparisons, etc. These reserved words or characters are called 'Operators'.
 6. The SELECT command retrieves zero or more rows from a table. It helps us to join information from different tables and filter specific information as per the required criteria.
- B.**
1. Any four features of a database are:
 - It stores a large amount of data in an organised manner.
 - It allows quick and easy searching of data.
 - It helps reduce data duplication (redundancy).
 - It provides easy updating, adding and deleting of records.



2. The DELETE command is used to remove records from a table. The INSERT command is used to insert records in a table.
3. Perform the following steps to download and install MySQL:
 1. Visit the following link: <https://dev.mysql.com/downloads>
 2. Click on MySQL Installer for Windows link.
 3. Click on the Download button.
 4. Once the setup file is downloaded, double-click on it to start the installation. A window will prompt asking for allowance to make changes to the system. Allow the changes. Then the MySQL Installer wizard appears.
 5. Click on the Next button with the default settings.
 6. Click on the Next button.
 7. Click on the Execute button. The installation will start.
 8. When the installation completes, click on the Next button twice. Product Configuration page appears.
 9. Click on Next button twice. The Accounts and Roles page appears.
 10. Type the password for the root user.
 11. Click on the Next button twice.
 12. Click on the Execute button.
 13. Click on the Finish button.
4. The UPDATE command is used to change the existing records in a table. The syntax of the UPDATE command is:

`UPDATE table_name SET field_change=value WHERE field_criteria=value;`

5. SQL provides the following Boolean operators:

Operator	Description	Example
AND	It returns true if both Boolean expressions are true.	<code>SELECT * FROM employee WHERE department = 'IT' AND age = 28;</code>
OR	It returns true if either of the Boolean expressions is true.	<code>SELECT * FROM employee WHERE age = 25 OR age = 26;</code>
NOT	It reverses the value of any other Boolean operator, it means it returns a false if the expression gives a true and vice versa.	<code>SELECT * FROM employee WHERE department = 'IT' AND NOT age = 28;</code>
>	Checks if value of 'a' is greater than value of 'b'. If yes, condition becomes true, else false.	(a > b) is true

6. The SELECT command retrieves zero or more rows from a table. For example,

```
SELECT * from Students;
```

The UPDATE command, on the other hand, is used to change the inserted records into a table.

```
UPDATE Students SET Marks = 485 where StudentID = 10002;
```

4. More on MySQL

EXERCISE



- A.** 1. b 3. a 4. c 5. b 6. d
- B.** 1. T 2. F 3. T 4. F 5. T
- C.** 1. order by 2. having 3. where 4. sum() 5. date
- D.** 1. GROUP BY command is used to group rows that have the same values in a column and helps in applying aggregate functions on each group.

2.

```
SELECT name, price
```

```
FROM products
```

```
WHERE code = 'P06';
```

3. Aggregate functions in MySQL are functions that perform calculations on multiple values and return a single value such as SUM, AVG, MAX, MIN and COUNT.).

4. A function is a predefined command that performs a specific operation and returns a value.

5. NOW() function returns the exact time at which the function executes as per system clock.

- E.** 1. i. Display all Deposit transactions:

```
SELECT *
```

```
FROM TRANSACTION
```

```
WHERE TYPE = 'Deposit';
```

- ii. Deposit and Withdrawals in January 2022:

```
SELECT A no, type, amount
```

```
FROM TRANSACTION
```

```
WHERE MONTH(DOT) = 1;
```

Output:

A_no	Type	Amount
102	Deposit	120000



iii. First date of transaction for A_no = 105:

```
SELECT MIN(DOT)
FROM TRANSACTION
WHERE A_no = 105;
```

Output:

NULL (No transaction exists for A_no = 105)

iv. Account + Transaction details for amount < 25000:

```
SELECT A.A_no, A.A_name, T.Amount, T.DOT
FROM ACCOUNT A, TRANSACTION T
WHERE A.A_no = T.A_no
AND T.Amount < 25000;
```

Output:

A_no	A_name	Amount	DOT
102	Rohan Gupta	20000	2022-05-12
103	Ali Reza	10000	2022-10-22

2. i.

A_name
Rohan Gupta
Rishabh Jain
Simran Kaur

ii.

A_no
101
103
102
103
102

iii. This condition is logically incorrect so:

```
COUNT(*) = 0
SUM(Amount) = NULL
```

iv.

COUNT(*)	SUM(Amount)
3	160000

3. **GROUP BY Clause:** The GROUP BY clause is used to group rows that have the same values in one or more columns. It is mainly used along with aggregate functions such as COUNT(), SUM(), AVG(), MAX(), and MIN(). The main purpose of GROUP BY is to create summary reports from large data sets.

For example, if we want to find the number of students in each subject group, GROUP BY helps to combine rows having the same subject and then perform calculations on them.

Example:

```
SELECT Subject_group, COUNT(*)  
FROM STUDENTS  
GROUP BY Subject_group;
```

ORDER BY Clause: The ORDER BY clause is used to sort the result set in either ascending (ASC) or descending (DESC) order. It does not group data; instead, it arranges the final output in a specific sequence based on one or more columns.

It is commonly used when we want to display data in a meaningful order, such as alphabetical order, increasing marks, or newest dates first.

Example:

```
SELECT Stu_name, Marks  
FROM STUDENTS  
ORDER BY Marks DESC;
```

4. **Aliases** are used to give a table or column, a temporary name. Aliases are often used to make column names more readable and presentable. An alias only exists for the duration of that query. It's not permanent. Its impact remains during query execution. Alias is valid during the scope of the statement.

Example (Column Alias)

```
SELECT A_name AS Name, Address AS City  
FROM ACCOUNT;
```

Here:

- A_name is displayed as Name
- Address is displayed as City

Example (Table Alias)

```
SELECT A.A_name, T.Amount  
FROM ACCOUNT A, TRANSACTION T  
WHERE A.A_no = T.A_no;
```

Here:

- ACCOUNT is given alias A
- TRANSACTION is given alias T

So instead of writing full table names again and again, we use shorter names like A and T.

5.

S.No.	Function	Purpose	Functionality
1	DATE ()	Extracts the date part of a date or complete date from the expression	SELECT DATE ('2022-10-17 11:25:30');
2	MONTH ()	The month from the date passed	SELECT MONTH ('2022-02-02');
3	YEAR ()	Returns the year out of the date	SELECT YEAR ('2022-05-01');

- F. Ananya can add a new column Date of Birth to her existing table using the ALTER TABLE command.

SQL Query:

```
ALTER TABLE Friends
```

```
ADD Date_of_Birth DATE;
```

IN THE LAB

Do yourself.

5. Advanced Features of MySQL

EXERCISE



- A.** 1. b 2. d 3. b 4. c 5. a
- B.** 1. WHERE 2. Cartesian product 3. INNER JOIN
4. CREATE VIEW 5. Database
- C.** 1. T 2. T 3. F 4. T 5. F
- D.** 1. Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2
FROM table_name
WHERE condition;
```

2. An index is a structure that is used to improve the performance of database activity. Index helps in quick data retrieval and it acts like an index page of a book that helps in quick access to the required page of the book.
3. Unique Index: This index is associated with the application of unique constraint to a column. A table can have multiple unique indexes.



Primary Index: This index is associated with the creation of primary key. It provides a unique identity to each row and does not allow NULL values.

4. **OUTER JOIN:** An OUTER JOIN returns all the rows from one table and the matching rows from the other table. If there is no matching row in the other table, the result will still include the row but with NULL (no value) for columns from the table that doesn't have a match.

INNER JOIN: An INNER JOIN is a way to combine data from two or more tables based on a common column that they share. It returns only the rows where the values in the common columns are exactly the same in both tables. If there is no match between the tables, those rows are not shown in the result.

- E.** 1. A subquery can be defined as a query within another query. It can be embedded within various clauses of the main query, such as the WHERE, HAVING, or FROM clause. A subquery helps retrieve data that will be used by the main query.

Syntax:

```
SELECT column1, column2
FROM table1
WHERE column1 = (SELECT column1 FROM table2 WHERE condition);
```

Example 1: Using MIN function

```
SELECT *
FROM STUDENTS
WHERE Age = (SELECT MIN(Age) FROM STUDENTS);
```

This query finds the student(s) having the minimum age.

The inner query finds the smallest age, and the outer query displays full details of that student.

Example 2: Using MAX function with condition

```
SELECT O_number, O_amt
FROM FRESH
WHERE O_amt = (SELECT MAX(O_amt) FROM FRESH WHERE O_date > '2025-01-10');
```

Here, the subquery finds the maximum order amount after a specific date, and the main query retrieves matching records.

2. A view is a virtual table created based on one or more tables. A view has rows and columns like a normal table in a database, but it does not store data itself. Instead, it displays data stored in underlying tables. A view is created using the CREATE VIEW command.

- **Creating a View**

A view is created using the CREATE VIEW command.

```
CREATE VIEW BUS_ROUTERS AS
SELECT Name, City
```

```
FROM CUSTOMERS
WHERE Transport = 'Bus';
```

This creates a view that shows only customers using bus transport.

- Updating a View

A view can be updated if:

- It is based on a single table
- It does not contain GROUP BY, JOIN, or aggregate functions

Example:

```
UPDATE PRODUCT_SALES
SET Discount = Discount + 100
WHERE Discount = 200;
```

Or using replacement view:

```
CREATE OR REPLACE VIEW PRODUCT_SALES AS
SELECT Product_name, Price, Quantity
FROM PRODUCTS;
```

- Inserting into a View

```
INSERT INTO PRODUCT_SALES (Product_name, Price, Quantity, Discount)
VALUES ('Monitor', 12000, 10, 400);
```

- Deleting a View

To delete only data:

```
DELETE FROM PRODUCT_SALES WHERE Discount = 400;
```

To remove the entire view:

```
DROP VIEW PRODUCT_SALES;
```

3. Different types of MySQL Indexes are:

1. Primary Index: This index is associated with the creation of primary key. It provides a unique identity to each row and does not allow NULL values.

Example:

```
CREATE TABLE STUDENTS (
Stu_id INT PRIMARY KEY,
Stu_name VARCHAR(50),
Age INT
);
```



2. Unique Index: This index is associated with the application of unique constraint to a column.

A table can have multiple unique indexes.

Example:

```
CREATE UNIQUE INDEX idx_email  
ON STUDENTS(email);
```

3. Full-Text Index: This type of index works with columns that have large quantities of text and it helps in quick data-based searches.

Example:

```
CREATE TABLE ARTICLES (  
id INT,  
content TEXT,  
FULLTEXT(content)  
);
```

4. Composite Index: This type of index is created with the creation of a composite primary key is defined, where more than one column is used to uniquely identify a row. It helps in resolving queries that work on multiple columns.

Example:

```
CREATE INDEX idx_name_class  
ON STUDENTS(Stu_name, Class);
```

5. Spatial Index: It works effectively on spatial data that is formed over geographical data types.

Example:

```
CREATE TABLE PLACES (  
id INT,  
location POINT,  
SPATIAL INDEX(location)  
);
```

F. Rules to be kept in mind before Updating Views are as follows:

- A view must be based on only one table to be updated.
- You cannot update a view that uses JOIN, GROUP BY, DISTINCT, or UNION.
- Only columns shown in the view can be updated.
- You must include all NOT NULL columns from the base table.
- You need the correct permissions to update a view.
- Views containing calculations or aggregate functions cannot be updated.
- Updates must follow all rules of the original table structure.

Do yourself.

Test Sheet 2

(Based on chapters 4 & 5)

Section A

- | | | | | | | |
|-----------|----------------------|----------|---------|---------|------|------|
| A. | 1. b | 2. c | 3. d | 4. d | 5. c | 6. d |
| B. | 1. T | 2. T | 3. T | 4. T | 5. T | 6. T |
| C. | 1. ORDER BY | 2. WHERE | 3. DATE | 4. Same | | |
| | 5. Cartesian product | 6. = | | | | |

Section B

- A.**
1. GROUP BY command is used to group rows that have the same values in a column and helps in applying aggregate functions on each group.
 2. Aggregate functions in MySQL are functions that perform calculations on multiple values and return a single value such as SUM, AVG, MAX, MIN and COUNT.).
 3. NOW() function returns the exact time at which the function executes as per system clock.
 4. An index is a structure that is used to improve the performance of database activity. Index helps in quick data retrieval and it acts like an index page of a book that helps in quick access to the required page of the book.
 5. OUTER JOIN: An OUTER JOIN returns all the rows from one table and the matching rows from the other table. If there is no matching row in the other table, the result will still include the row but with NULL (no value) for columns from the table that doesn't have a match.

INNER JOIN: An INNER JOIN is a way to combine data from two or more tables based on a common column that they share. It returns only the rows where the values in the common columns are exactly the same in both tables. If there is no match between the tables, those rows are not shown in the result.

6. Syntax:

```
CREATE VIEW view_name AS
SELECT column1, column2
FROM table_name
WHERE condition;
```

- B.**
1. GROUP BY Clause: The GROUP BY clause is used to group rows that have the same values in one or more columns. It is mainly used along with aggregate functions such as COUNT(), SUM(), AVG(), MAX(), and MIN(). The main purpose of GROUP BY is to create summary reports from large data sets.

For example, if we want to find the number of students in each subject group, GROUP BY helps to combine rows having the same subject and then perform calculations on them.

Example:

```
SELECT Subject_group, COUNT(*)
FROM STUDENTS
GROUP BY Subject_group;
```

ORDER BY Clause: The ORDER BY clause is used to sort the result set in either ascending (ASC) or descending (DESC) order. It does not group data; instead, it arranges the final output in a specific sequence based on one or more columns.

It is commonly used when we want to display data in a meaningful order, such as alphabetical order, increasing marks, or newest dates first.

Example:

```
SELECT Stu_name, Marks
FROM STUDENTS
ORDER BY Marks DESC;
```

2.	S.No.	Function	Purpose	Functionality
	1	DATE ()	Extracts the date part of a date or complete date from the expression	SELECT DATE ('2022-10-17 11:25:30');
	2	MONTH ()	The month from the date passed	SELECT MONTH (' 2022-02-02 ');
	3	YEAR ()	Returns the year out of the date	SELECT YEAR (' 2022-05-011');

3. Different types of MySQL Indexes are:

1. **Primary Index:** This index is associated with the creation of primary key. It provides a unique identity to each row and does not allow NULL values.

Example:

```
CREATE TABLE STUDENTS (
    Stu_id INT PRIMARY KEY,
    Stu_name VARCHAR(50),
    Age INT
);
```

2. **Unique Index:** This index is associated with the application of unique constraint to a column. A table can have multiple unique indexes.

Example:

```
CREATE UNIQUE INDEX idx_email  
ON STUDENTS(email);
```

3. Full-Text Index: This type of index works with columns that have large quantities of text and it helps in quick data-based searches.

Example:

```
CREATE TABLE ARTICLES (  
id INT,  
content TEXT,  
FULLTEXT(content)  
);
```

4. Composite Index: This type of index is created with the creation of a composite primary key is defined, where more than one column is used to uniquely identify a row. It helps in resolving queries that work on multiple columns.

Example:

```
CREATE INDEX idx_name_class  
ON STUDENTS(Stu_name, Class);
```

5. Spatial Index: It works effectively on spatial data that is formed over geographical data types.

Example:

```
CREATE TABLE PLACES (  
id INT,  
location POINT,  
SPATIAL INDEX(location)  
);
```

4. • Bank Table

```
CREATE TABLE Bank  
(  
BankID INT,  
BankName VARCHAR(30),  
Branch VARCHAR(30),  
IFSCCode VARCHAR(15)  
);
```

This table stores bank details.

BankID is used to identify each bank.

- Customer Table

```
CREATE TABLE Customer
(
    CustomerID INT,
    CustomerName VARCHAR(30),
    AccountNo INT,
    BankID INT
);
```

This table stores customer details.

BankID is the common field between both tables.

- Equijoin

```
SELECT Bank.BankID, Bank.BankName, Bank.Branch,
       Customer.CustomerName, Customer.AccountNo
FROM Bank, Customer
WHERE Bank.BankID = Customer.BankID;
```

An Equijoin joins two tables by using the equality operator =.

Here, records from both tables are matched where Bank.BankID = Customer.BankID.

- Natural Join

```
SELECT *
FROM Bank
NATURAL JOIN Customer;
```

A Natural Join automatically joins two tables using the column with the same name.

Here, BankID is present in both tables, so MySQL joins both tables using BankID.