CHAPTER 03

Queries in SQL

In this Chapter...

- SQL SELECT Statement
- Eliminating Redundant Data (DISTINCT Keyword)
- SQL WHERE Clause
- Operators in MySQL
- Working with NULL Values
- Changing Data with DML Commands
- Manipulating Data of a Table/Relation
- Restructuring a Table (ALTER TABLE Command)
- DROP TABLE Command

A database query is a piece of code that is sent to a database in order to get information back from the database. It is used as the way of retrieving the information from database.

You can say that **query** is basically a **question** that you ask the database. The result of the query is the information that is returned by the database management system.

Queries are constructed using SQL (Structured Query Language) which resembles a high-level programming language.

SQL SELECT Statement

The most commonly used SQL command is SELECT statement. The SQL SELECT statement is used to query or retrieve data from a table in the database. A query may retrieve information from specified columns or from all of the columns in the table.

The retrieved information is stored in a result table, commonly known as the **result set**. To create a simple SQL SELECT statement, you must specify the column(s) name and the table name. The whole query is called SQL SELECT statement.

Syntax

```
SELECT column_list
FROM table_name;
```

Here, **table_name** is the name of the table from which the information is retrieved and **column_list** includes one or more columns from which data is retrieved.

In SQL, **SELECT** clause is used to list the attributes desired in the result of a query and **FROM** clause is used to list the relations from which such columns are to be extracted.

Some terminologies used in SQL commands are given below

Keyword A keyword refers to a special word that has a special meaning to SQL. Reserved keywords are part of the grammar of the SQL language that is used by SQL server to parse and understand the SQL statements.

For example, SELECT, FROM, INSERT, etc., are keywords.

Commands or Statements These are instructions given by you to a SQL database.

For example, `SELECT student_name FROM student;` is a statement.

Clause Commands consist of one or more logically distinct parts, called clauses. Clauses begin with a keyword for which they are generally named, and consist of keywords and arguments.

For example, consider the statement

```
SELECT student_name FROM student;
```

Here, 'SELECT student_name' and 'FROM student' are clauses.

Arguments complete or modify the meaning of a clause.

For example, in the above example, 'student' is the argument and FROM is the keyword of FROM clause.

Selecting Specific Columns

To select any specific column or information from the table, we use the following command

Syntax

```
SELECT<column_name1>,[<column_name2>,….,<column_nameN>]
FROM <table_name>;
```

column_name1,column_name2,…..,column_nameN specifies the name of the columns.

table_name specifies the name of the table from which the data is fetched out.

For example,

Let us consider the following table COMPANY

EMP_ CODE	EMP_ NAME	EMP_ADDRESS	EMP_SALARY	EMP_DEPT_NO	DOJ
100	Rahul Sharma	C-21, Arya Nagar	25600	D05	2012-02-06
101	Vikas Mittal	A/44, Mayur Vihar	26000	D03	2002-12-04
102	Puneet Jain	50-MIG, Rohini	21000	D05	2004-08-06
103	Sachin Vats	A-21, Ankur Vihar	23500	D02	2007-09-06
104	Uday Singh	D-34, Indraprasth	26000	D03	2001-07-06
105	Ravi Shukla	45/A, Vivek Nagar	23500	D01	2012-05-06
106	Vinay Rana	120, DDA Colony	24000	D03	2001-02-06

Suppose you want to retrieve only EMP_SALARY and EMP_DEPT_NO from the above table, write the SELECT command as follows.

```
mysql>  SELECT EMP_SALARY, EMP_DEPT_NO
        FROM COMPANY;
```

Above query produces the following output

```
+------------+---------------+
| EMP_SALARY |  EMP_DEPT_NO  |
+------------+---------------+
|   25600    |  D05          |
|   26000    |  D03          |
|   21000    |  D05          |
|   23500    |  D02          |
|   26000    |  D03          |
|   23500    |  D01          |
|   24000    |  D03          |
+------------+---------------+
7 rows in set (0.00 sec)
```

Selecting All Columns

To select all the columns of a table or entire table we can use an asterisk(*) symbol in place of column_name list.

Syntax

```
SELECT <*> FROM <table_name>;
```

For example, to retrieve all columns from the previous table **COMPANY** use the command given below

```
mysql>SELECT * FROM COMPANY;
```

Above query produces the following output

```
+----------+-------------+-------------------+------------+-------------+------------+
| EMP_CODE | EMP_NAME    | EMP_ADDRESS       | EMP_SALARY | EMP_DEPT_NO |    DOJ     |
+----------+-------------+-------------------+------------+-------------+------------+
|    100   | Rahul Sharma| C-21, Arya Nagar  |   25600    |     D05     | 2012-02-06 |
|    101   | Vikas Mittal| A/44, Mayur Vihar |   26000    |     D03     | 2002-12-04 |
|    102   | Puneet Jain | 50-MIG, Rohini    |   21000    |     D05     | 2004-08-06 |
|    103   | Sachin Vats | A-21, Ankur Vihar |   23500    |     D02     | 2007-09-06 |
|    104   | Uday Singh  | D-34, Indraprasth |   26000    |     D03     | 2001-07-06 |
|    105   | Ravi Shukla | 45/A, Vivek Nagar |   23500    |     D01     | 2012-05-06 |
|    106   | Vinay Rana  | 120, DDA Colony   |   24000    |     D03     | 2001-02-06 |
+----------+-------------+-------------------+------------+-------------+------------+
7 rows in set (0.00 sec)
```

Eliminating Redundant Data (DISTINCT Keyword)

When you are working with the SQL SELECT statement, you will more than likely come across duplicate rows when viewing your query results. This duplication can cause various problems depending upon the applications that use the information. This duplication can happen where no primary or unique key constraints exist, or where these constraints have been disabled.

To remove this redundancy, the SQL DISTINCT keyword is used. By using this keyword, you will be able to remove all duplicate rows from your query result. This keyword is used in conjunction with SELECT statement for fetching only unique records.

Syntax

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

DISTINCT is an optional keyword that needs to precede the columns that are specified in the SELECT clause.

Using DISTINCT, the system will evaluate that data contained in all of the columns as a single unit, on a row per row basis, and will eliminate any duplicates that it find. It will then return the results of the unique rows that will be remaining.

For example, suppose we want to select only the distinct values from the column named **EMP_DEPT_NO** from the **COMPANY** table, we use the following SELECT command for this purpose,

```
mysql> SELECT DISTINCT EMP_DEPT_NO
       FROM COMPANY;
```

Above query produces the following output

```
+-------------+
| EMP_DEPT_NO |
+-------------+
|     D05     |
|     D03     |
|     D02     |
|     D01     |
+-------------+
4 rows in set (0.00 sec)
```

SQL WHERE Clause

The WHERE clause is used when you want to retrieve specific information from a table exluding other irrelevent data.

For example, when you want to see the information about students in class 10th only then you do not need the information about the students in other classes. Retrieving information about all the students would increase the processing time for the query.

So, SQL offers a feature called WHERE clause, which we can use to restrict the data that is retrieved. The condition you provide in the WHERE clause filters the rows retrieved from the table and gives you only those rows which you wanted to see.

WHERE clause can be used along with SELECT, DELETE, UPDATE statements.

When a WHERE clause is present, the database program goes through the entire table one row at a time and examines each row to determine if the given condition is satisfied. If it is satisfied for a row, the row will be displayed in the output.

The **WHERE** command (clause) with **SELECT** command (clause) can retrieve records from a table with some given conditions. As the select query executes, SQL processes one row at a time.

Each time the conditional statement is met (returns true), a row is returned as a result.

Syntax

```
SELECT <column_name>
FROM <table_name> WHERE<condition>;
```

The non-numeric values (string and dates) in the **WHERE** command must be enclosed in single quotes.

For example, suppose we want to retrieve the **EMP_NAME** and **EMP_SALARY** for those who belongs to department number **D05** from the **COMPANY** table.

The following command is used to perform this task

```
mysql> SELECT EMP_NAME, EMP_SALARY
       FROM COMPANY
       WHERE EMP_DEPT_NO = 'D05';
```

Above query produces the following output

```
+-------------+------------+
| EMP_NAME    | EMP_SALARY |
+-------------+------------+
| Rahul Sharma|    25600   |
| Puneet Jain |    21000   |
+-------------+------------+
2 rows in set (0.00 sec)
```

Operators in MySQL

MySQL supports different types of operators, some of them are described below

Arithmetic Operators

These operators are used to perform mathematical calculations, such as addition, subtraction, multiplication, division and remainder.

Some most important arithmetic operators used in MySQL are

OPERATOR	DESCRIPTION
+ (Addition)	Add the two arguments together
− (Subtraction)	Subtract the second argument from the first argument
* (Multiplication)	Multiplies the two arguments
/ (Division)	Divide the first argument by the second argument
% (Remainder)	Divide the first argument from the second argument and provides the remainder of that operation

Syntax

```
SELECT <Expression1> <arithmetic operator>
       <Expression2>
       FROM <table_name>
       WHERE <condition>;
```

For example, query to display **EMP_NAME, EMP_DEPT_NO** and 20% of **EMP_SALARY** for each employee for social fund.

```
mysql> SELECT EMP_NAME, EMP_DEPT_NO,
       EMP_SALARY*0.20
       FROM COMPANY;
```

Above query produces the following output

```
+-------------+------------+-----------------+
| EMP_NAME    | EMP_DEPT_NO | EMP_SALARY*0.20 |
+-------------+------------+-----------------+
| Rahul Sharma|     D05    |       5120      |
| Vikas Mittal|     D03    |       5200      |
| Puneet Jain |     D05    |       4200      |
| Sachin Vats |     D02    |       4700      |
| Uday Singh  |     D03    |       5200      |
| Ravi Shukla |     D01    |       4700      |
| Vinay Rana  |     D03    |       4800      |
+-------------+------------+-----------------+
7 rows in set (0.00 sec)
```

Arithmetic operators can be implemented through simple SELECT statement without any table. This acts like a function.

For example, `SELECT 35*2+5;`

Above query produces the following output

```
+----------------+
|     35*2+5     |
+----------------+
|       75       |
+----------------+
1 row in set (0.03 sec)
```

Comparison Operators

These operators are used to test or compare the value of two operands, *i.e.*, between two variables or between a variable and a constant. If the condition is false, then the result is zero (0) and if the condition is true, then the result is non-zero. These operators are also called **relational operators.**

Some of the comparison/relational operators used in MySQL are as follows

OPERATOR	DESCRIPTION
=	Equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
<> or !=	Not equal to (not ISO standard)
!<	Not less than (not ISO standard)
!>	Not greater than (not ISO standard)

Syntax

```
SELECT <column name>|*|<expression>
FROM <table name> WHERE<expression>
<comparison operator> <expression>;
```

For example, query to display **EMP_NAME** and **EMP_SALARY** for those employees whose salary is greater than or equal to 25000.

```
mysql>  SELECT EMP_NAME,EMP_SALARY
        FROM COMPANY
        WHERE EMP_SALARY> = 25000;
```

Above query produces the following output

```
+-------------+------------+
| EMP_NAME    | EMP_SALARY |
+-------------+------------+
| Rahul Sharma|    25600   |
| Vikas Mittal|    26000   |
| Uday Singh  |    26000   |
+-------------+------------+
3 rows in set (0.00 sec)
```

When we use relational operators with character data type, < means earlier in the alphabet and > means later in the alphabet 'Bangalore '<' 'Brajil' as 'a' comes before 'r' in alphabet.

Logical Operators

The logical operators compare two conditions at a time to determine whether a row can be selected for the output. Logical operators are also called **boolean operators**, because these operators return a boolean data type value as TRUE, or FALSE.

When retrieving data using a SELECT statement, you can use these operators in the WHERE clause, which allows you to combine more than one condition.

Some of the Boolean/Logical operators used in MySQL are as follows

OPERATOR	DESCRIPTION
AND	Logical AND compares two expressions and return true, when both expressions are true
OR	Logical OR compares two expressions and return true, when atleast one of the expressions is true
NOT	NOT takes a single expression as an argument and changes its value from false to true or from true to false. You can use an exclamation point (!) in place of this operator

Syntax

```
SELECT <column name>|*|<expression>
FROM <table name>
WHERE <expressions> <boolean operator>
<expressions>;
```

For example, query to display **EMP_CODE** and **EMP_NAME** for those employees whose **EMP_DEPT_NO** is D05 and **EMP_SALARY** is greater than 22000.

```
mysql>  SELECT EMP_CODE,EMP_NAME
        FROM COMPANY
        WHERE (EMP_DEPT_NO='D05'AND
        EMP_SALARY >22000);
```

Above query produces the following output

```
+----------+----------------+
| EMP_CODE |   EMP_NAME     |
+----------+----------------+
| 100      | Rahul Sharma   |
+----------+----------------+
1 row in set (0.00 sec)
```

For example, query to display **EMP_CODE** and **EMP_NAME** for those employees whose **EMP_DEPT_NO** is D05 or **'EMP_SALARY'** is greater than 22000.

```
mysql>  SELECT EMP_CODE, EMP_NAME
        FROM COMPANY
        WHERE(EMP_DEPT_NO = 'D05' OR
        EMP_SALARY>22000);
```

Above query produces the following output

```
+----------+----------------+
| EMP_CODE | EMP_NAME       |
+----------+----------------+
|    100   | Rahul Sharma   |
|    101   | Vikas Mittal   |
|    102   | Puneet Jain    |
|    103   | Sachin Vats    |
|    104   | Uday Singh     |
|    105   | Ravi Shukla    |
|    106   | Vinay Rana     |
+----------+----------------+
7 rows in set (0.00 sec)
```

For example, query to display **EMP_CODE** and **EMP_NAME** for the employees whose **EMP_SALARY** is not greater than 22000.

```
mysql>  SELECT EMP_CODE, EMP_NAME
        FROM COMPANY
        WHERE(NOT EMP_SALARY>22000);
                    or
mysql>  SELECT EMP_CODE, EMP_NAME
        FROM COMPANY
        WHERE !(EMP_SALARY>22000);
```

Above query produces the following output

```
+----------+----------------+
| EMP_CODE | EMP_NAME       |
+----------+----------------+
|    102   | Puneet Jain    |
+----------+----------------+
  1 row in set (0.00 sec)
```

BETWEEN Operator

The BETWEEN operator is used in a WHERE clause to check that the field values are within a specified inclusive range.

The values can be numbers, text or dates. The range consists of a beginning expression, followed by an **AND** keyword and an end expression.

The operator returns 1 if the search value is present within the range otherwise returns 0. BETWEEN clause is sensitive in order, and the first value in the clause must be first in alphabetic or numeric order.

Syntax

```
SELECT <column name>|*| <expression>
FROM <table name>
WHERE <column name> BETWEEN <value1> AND <value2>;
```

For example, query to display **EMP_CODE, EMP_NAME** and **EMP_SALARY** for those employees whose salary lies between 23000 to 27000.

```
mysql> SELECT EMP_CODE,EMP_NAME, EMP_SALARY
       FROM COMPANY
       WHERE (EMP_SALARY BETWEEN 23000
       AND 27000);
```

Above query produces the following output

```
+----------+--------------+------------+
| EMP_CODE | EMP_NAME     | EMP_SALARY |
+----------+--------------+------------+
|    100   | Rahul Sharma |    25600   |
|    101   | Vikas Mittal |    26000   |
|    103   | Sachin Vats  |    23500   |
|    104   | Uday Singh   |    26000   |
|    105   | Ravi Shukla  |    23500   |
|    106   | Vinay Rana   |    24000   |
+----------+--------------+------------+
6 rows in set (0.00 sec)
```

IN Operator

This operator checks a value within a set of values separated by commas and retrieve the rows from the table which are matching.

The IN operator returns 1 when the search value is present within the set otherwise returns 0. The IN operator allows you to specify multiple values in a WHERE clause.

Syntax

```
SELECT <column name>|*|<expression>
FROM <table name>
WHERE <column name>IN <(value1, value2,…)>;
```

For example, query to display **EMP_CODE**, **EMP_NAME** and **DOJ** of the employees whose **EMP_CODE** is 101,104,105.

```
mysql> SELECT EMP_CODE, EMP_NAME, DOJ
       FROM COMPANY
       WHERE EMP_CODE IN (101,104,105);
```

Above query produces the following output

```
+----------+-------------+------------+
| EMP_CODE | EMP_NAME    |     DOJ    |
+----------+-------------+------------+
|    101   | Vikas Mittal| 2002-12-04 |
|    104   | Uday Singh  | 2001-07-06 |
|    105   | Ravi Shukla | 2012-05-06 |
+----------+-------------+------------+
3 rows in set (0.00 sec)
```

LIKE Operator

This operator is used to search a specified pattern in a column. It is useful when you want to search rows to match a specific pattern, or when you do not know the entire value.

The SQL LIKE clause is used to compare a value to similar values using wildcard operators.

We describe patterns by using two special wildcard operators, given below

(i) **The Percent Sign** (%) It is used to match any substring.

(ii) **The Underscore** (_) It is used to match any character.

The symbols can also be used in combinations.

Syntax

```
SELECT <column name> |*| <expression> <column name>
FROM <table name>
WHERE <column name> LIKE <pattern>;
```

Here are number of examples showing WHERE clause having different LIKE clause with '%' and '_' operators.

COMMAND	DESCRIPTION
`WHERE SALARY LIKE '100%';`	Find any values that start with 100.
`WHERE SALARY LIKE '%100%';`	Find any values that have 100 in any position.
`WHERE SALARY LIKE '_00%';`	Find any values that have 00 in the second and third positions.
`WHERE SALARY LIKE '1_%_%';`	Find any values that start with 1 and are atleast 2 characters in length.
`WHERE SALARY LIKE '%2';`	Find any values that end with 2.
`WHERE SALARY LIKE '_2%3';`	Find any values that have a 2 in the second position and end with a 3.
`WHERE SALARY LIKE '2_ _3';`	Find any values in a four-digit number that start with 2 and end with 3.

For patterns to include the special pattern characters (that is, % and _), SQL allows the specification of an escape character. The escape character is used immediately before a special pattern character to indicate that the special pattern character is to be treated like a normal character.

We define the escape character for a like comparison using the **escape keyword**.

To illustrate, consider the following patterns, which use a backslash (\ ) as the escape character

Like 'ab\%cd%' escape '\' matches all strings beginning with "ab%cd".

Like 'ab\\cd%' escape '\' matches all strings beginning with "ab\cd".

For example, query to display **EMP_CODE, EMP_NAME** and **EMP_ADDRESS** of the employees whose name started with an alphabet 'V'.

```
mysql> SELECT EMP_CODE,EMP_NAME, EMP_ADDRESS
       FROM COMPANY
       WHERE EMP_NAME LIKE 'V%';
```

Above query produces the following output

```
+----------+-------------+-------------------+
| EMP_CODE | EMP_NAME    |     EMP_ADDRESS   |
+----------+-------------+-------------------+
|   101    | Vikas Mittal| A/44, Mayur Vihar |
|   106    | Vinay Rana  |  120, DDA Colony  |
+----------+-------------+-------------------+
2 rows in set (0.00 sec)
```

Patterns are case-sensitive, that is, upper-case characters do not match with lower-case characters or *vice-versa*.

Operator Precedence

MySQL processes expressions according to a very specific operator precedence. When an expression in an SQL statement is processed, it is evaluated according to the order in which elements are included in the statement and the precedence in which operators are assigned. The operators with higher precedence are evaluated first.

The operator precedences are shown below in the table

PRECEDENCE	OPERATOR	DESCRIPTION
1.	!	Logical Negation
2.	-,~	Unary minus, Unary bit conversion
3.	^	Bitwise exclusive or operation
4.	*, /, %	Multiplication, Division, Modulus
5.	–, +	Subtraction, Addition
6.	=, < =, >, > =, ! =, LIKE, IN	All comparison operator except BETWEEN
7.	BETWEEN	Between operator
8.	AND, &&	Logical AND operator
9.	OR, II	Logical OR operator
10.	: =	Assignment operator

The operators listed here are shown from the highest precedence to the lowest, *for example,* the arithmetic operators has precedence over the comparison operators.

However, operators that appear on the same line of the list have the same level of precedence, so, they are evaluated in the order in which they appear in the expression.

For example, the multiplication (*) and division (/) operators have the same level of precedence so they are evaluated in the order in which they appear.

Working with NULL Values

In MySQL, the empty values are represented as NULL in a table. If a table having null values, then you can display columns with null values or without null values and you can replace NULL values with another value.

To handle NULL values in MySQL, you should use the following commands

IS NULL Clause

If we want to search the column whose value is NULL in a table, then we use IS NULL clause.

Syntax

```
SELECT<column name>|*|<expression>
FROM <table_name>
WHERE <column_name> IS NULL;
```

For example,

Let us consider the following table Teacher

T_No	T_Name	T_Salary	DOJ
T01	Aradhna	17000	2013-01-08
T02	Ritika	NULL	2013-12-14
T03	Ravindra	NULL	2013-11-23
T04	Dushyant	16000	2014-01-10
T05	Swati	19000	2014-02-10

Query to display column whose value is NULL.

```
mysql> SELECT *
       FROM Teacher
       WHERE T_Salary IS NULL;
```

Above query produces the following output

```
+------+---------+---------+------------+
| T_NO | T_NAME  | T_SALARY|    DOJ     |
+------+---------+---------+------------+
| T02  | Ritika  |  NULL   | 2013-12-14 |
| T03  | Ravindra|  NULL   | 2013-11-23 |
+------+---------+---------+------------+
2 rows in set (0.00 sec)
```

IS NOT NULL Clause

If we want to search the column whose value is not NULL in a table then we use IS NOT NULL clause.

Syntax

```
SELECT<column name |*| <expression>
FROM <table_name>
WHERE <column_name> IS NOT NULL;
```

For example, mysql>SELECT *
FROM Teacher
WHERE T_Salary IS NOT NULL;

Above query produces the following output

```
+------+---------+---------+------------+
| T_NO | T_NAME  | T_SALARY |    DOJ     |
+------+---------+---------+------------+
| T01  | Aradhna |  17000  | 2013-01-08 |
| T04  | Dushyant|  16000  | 2014-01-10 |
| T05  | Swati   |  19000  | 2014-02-10 |
+------+---------+---------+------------+
3 rows in set (0.00 sec)
```

IFNULL Function

This function takes two arguments. If the first argument is not equal to NULL, then the function returns the first argument otherwise, the second argument is returned. This function generally used to replace NULL value with another value.

Syntax

```
SELECT <column_name1>, IFNULL (column_name,"Another
value") FROM <table_name>
WHERE <condition>;
```

For example, mysql> SELECT T_No, T_Name, IFNULL <T_Salary, 20000> FROM Teacher;

Above query produces the following output

```
+------+---------+-------------------------+
| T_NO | T_NAME  | IFNULL<T_Salary,20000>  |
+------+---------+-------------------------+
| T01  | Aradhna |          17000          |
| T02  | Ritika  |          20000          |
| T03  | Ravindra|          20000          |
| T04  | Dushyant|          16000          |
| T05  | Swati   |          19000          |
+------+---------+---------+---------------+
5 rows in set (0.00 sec)
```

The IFNULL () function is available in MySQL, and not in SQL server or Oracle.

Changing Data with DML Commands

DML (Data Manipulation Language) is a category of SQL used to access and manipulate data in existing database. In DML commands, INSERT command is used to insert the data into a table.

Inserting Data

Inserting is a process of adding information into a table. When a table is created, it does not contain any data. Data is inserted in one row at a time.

Inserting a New Row-INSERT INTO Command

The INSERT command is used to add a single record or multiple records into a table.

Syntax

```
INSERT INTO <table_name>(col_1, col_2, col_3,..,
col_n) VALUES(value_1, value_2,... value_n);
```

Here, table_name defines the name of a table, where data will be inserted. col_1, col_2, col_3,..., col_n are the columns of the current table. value_1, value_2,..., value_n are the data values against each column.

For example, The following information exists in the table STUDENT

```
+----------+------------+-----------+
| Roll_No  |   Name     |  Subject  |
+----------+------------+-----------+
|   101    |   Rahul    |  Art      |
|   102    |   Vikas    |  Science  |
|   103    |   Puneet   |  Science  |
|   104    |   Sachin   |  Art      |
|   105    |   Uday     |  Commerce |
+----------+------------+-----------+
```

To add a new row into the STUDENT table use the INSERT INTO command as follow

```
mysql>INSERT INTO STUDENT(Roll_No, Name, Subject)
VALUES(106, 'Ajay', 'Science');
```

Now, the table STUDENT will look like as follows

To add a new row into the STUDENT table use the INSERT INTO command as follow

```
mysql>INSERT INTO STUDENT(Roll_No, Name, Subject)
VALUES(106, 'Ajay', 'Science');
```

Now, the table STUDENT will look like as follows

```
+----------+------------+-----------+
| Roll_No  |   Name     |  Subject  |
+----------+------------+-----------+
|   101    |   Rahul    |  Art      |
|   102    |   Vikas    |  Science  |
|   103    |   Puneet   |  Science  |
|   104    |   Sachin   |  Art      |
|   105    |   Uday     |  Commerce |
|   106    |   Ajay     |  Science  |
+----------+------------+-----------+
```

Inserting Null Values

When you execute the INSERT statement, the database server inserts a NULL value into any column for which you provide no value, as well as for all columns that have no default values.

For example,

```
mysql>INSERT INTO STUDENT(Roll_No, Name) VALUES(107, 'Viku');
```

The above command inserts only **Roll_No** and **Name** of the table **STUDENT** but the remaining column **Subject** (as same in previous table) becomes NULL.

You can also specify the NULL Keyword in the VALUES clause to indicate that a column should be assigned a NULL value.

For example,

```
mysql>INSERT INTO STUDENT(Roll_No, Name, Subject) VALUES(107, 'Viku', NULL);
```

Now, the table STUDENT will look like as follows

Roll_No	Name	Subject
101	Rahul	Art
102	Vikas	Science
103	Puneet	Science
104	Sachin	Art
105	Uday	Commerce
106	Ajay	Science
107	Viku	NULL

Inserting Numeric and Character Values

We insert both numeric and character type values in a table. In MySQL, the numeric values are accepted in the data types called INT, SMALLINT, MEDIUMINT, FLOAT, DOUBLE etc., and character data types are accepted in CHAR, VARCHAR etc.

We give the char values in quotes (' ') and numerical values without quotes.

For example,

```
mysql> INSERT INTO STUDENT1(Roll_No, Name, Total, Percentage) VALUES (129,'Vinay', 1560, 70.5);
```

The above command inserts both character and numeric values

Here, Roll_No Integer value
Name Character value
Total Double int value
Percentage Float value

Output

```
Query OK, 1 row affected <0.03 sec>
```

The column in which a value is not inserted in the above command contains NULL value.

Inserting Date Values

A simple date value in MySQL exists in 'YYYY-MM-DD' format.

For example, first four digits YYYY depicting Year, next two digits MM depicting Month and last two digits DD depicting Date.

For example,

```
mysql> INSERT INTO STUDENT1(Roll_ No, Name, Total, DOB) VALUES(129, 'Vinay', 1560, '1984-12-29');
```

Output

```
Query OK, 1 row affected <0.03 sec>
```

Manipulating Data of a Table/Relation

UPDATE Command

The UPDATE command is used to update a single record or multiple records in a table. The UPDATE command is used to modify the existing rows in a table. To update multiple fields, each field assignment is separated with a comma(,) symbol.

The WHERE clause in the UPDATE syntax specifies which record or records should be updated. If you omit the WHERE clause, all records will be updated.

Syntax

```
UPDATE<table name > SET <column1> = <value1>, [<column2> = <value2> ,.....]
WHERE <condition>;
```

In above query, update the EMP_SALARY from the given table COMPANY.

```
mysql>UPDATE COMPANY SET EMP_SALARY=28000
      WHERE EMP_CODE=100;
```

To view the output of the above command use SELECT command.

```
mysql>SELECT * FROM COMPANY;
```

Above query produces the following output

EMP_CODE	EMP_NAME	EMP_ADDRESS	EMP_SALARY	EMP_DEPT_NO	DOJ
100	Rahul Sharma	C-21, Arya Nagar	28000	D05	2012-02-06
101	Vikas Mittal	A/44, Mayur Vihar	26000	D03	2002-12-04
102	Puneet Jain	50-MIG, Rohini	21000	D05	2004-08-06
103	Sachin Vats	A-21, Ankur Vihar	23500	D02	2007-09-06
104	Uday Singh	D-34, Indraprasth	26000	D03	2001-07-06
105	Ravi Shukla	45/A, Vivek Nagar	23500	D01	2012-05-06
106	Vinay Rana	120, DDA Colony	24000	D03	2001-02-06

7 rows in set (0.00 sec)

Using Expressions in Update

Expression are also used in the SET clause of the UPDATE command to manipulate the values.

For example, if you want to increase the salary of each employee by 1000.

Then you could use the following query

```
mysql>UPDATE COMPANY SET EMP_SALARY=EMP_SALARY+1000;
```

To view the output of the above command use SELECT command.

```
mysql>SELECT * FROM COMPANY;
```

Above query produces the following output

EMP_CODE	EMP_NAME	EMP_ADDRESS	EMP_SALARY	EMP_DEPT_NO	DOJ
100	Rahul Sharma	C-21, Arya Nagar	29000	D05	2012-02-06
101	Vikas Mittal	A/44, Mayur Vihar	27000	D03	2002-12-04
102	Puneet Jain	50-MIG, Rohini	22000	D05	2004-08-06
103	Sachin Vats	A-21, Ankur Vihar	24500	D02	2007-09-06
104	Uday Singh	D-34, Indraprasth	27000	D03	2001-07-06
105	Ravi Shukla	45/A, Vivek Nagar	24500	D01	2012-05-06
106	Vinay Rana	120, DDA Colony	25000	D03	2001-02-06

7 rows in set (0.00 sec)

You can use the WHERE command to update the salary of some employee's, as follows

```
mysql>UPDATE COMPANY
      SET EMP_SALARY = EMP_SALARY + 1000
      WHERE EMP_DEPT_NO = 'D05';
```

To view the output of the above command use SELECT command.

```
mysql>SELECT * FROM COMPANY;
```

Above query produces the following output

EMP_CODE	EMP_NAME	EMP_ADDRESS	EMP_SALARY	EMP_DEPT_NO	DOJ
100	Rahul Sharma	C-21, Arya Nagar	30000	D05	2012-02-06
101	Vikas Mittal	A/44, Mayur Vihar	27000	D03	2002-12-04
102	Puneet Jain	50-MIG, Rohini	23000	D05	2004-08-06
103	Sachin Vats	A-21, Ankur Vihar	24500	D02	2007-09-06
104	Uday Singh	D-34, Indraprasth	27000	D03	2001-07-06
105	Ravi Shukla	45/A, Vivek Nagar	24500	D01	2012-05-06
106	Vinay Rana	120, DDA Colony	25000	D03	2001-02-06

7 rows in set (0.00 sec)

Updating to NULL Values

You can use UPDATE command to enter the NULL values just as other values. *For example*, the address of the employee Ravi Shukla is changed. But for the time being, the new address is not known, thus NULL value is to be inserted for the address of Ravi Shukla. This can be done as follows

```
mysql>UPDATE COMPANY SET EMP_ADDRESS = NULL
      WHERE EMP_NAME = 'Ravi Shukla';
```

To view the ouput of the above command, use SELECT command

```
mysql>SELECT * FROM COMPANY;
```

Above query produces the following output

EMP_CODE	EMP_NAME	EMP_ADDRESS	EMP_SALARY	EMP_DEPT_NO	DOJ
100	Rahul Sharma	C-21, Arya Nagar	30000	D05	2012-02-06
101	Vikas Mittal	A/44, Mayur Vihar	27000	D03	2002-12-04
102	Puneet Jain	50-MIG, Rohini	23000	D05	2004-08-06
103	Sachin Vats	A-21, Ankur Vihar	24500	D02	2007-09-06
104	Uday Singh	D-34, Indraprasth	27000	D03	2001-07-06
105	Ravi Shukla	NULL	24500	D01	2012-05-06
106	Vinay Rana	120, DDA Colony	25000	D03	2001-02-06

7 rows in set (0.00 sec)

DELETE Command

To discard unwanted data from a database, the DELETE command is used. The DELETE command uses a WHERE clause. If you don't use a WHERE clause, all rows in the table will be deleted.

Syntax `DELETE FROM <table name> WHERE <condition>;`

For example, query to delete the record of employee Puneet Jain from the table COMPANY.

```
mysql> DELETE FROM COMPANY WHERE EMP_NAME ='Puneet Jain';
```

To view the output of the above command use SELECT command.

```
mysql>SELECT * FROM COMPANY;
```

Now, above query produces the following output

EMP_CODE	EMP_NAME	EMP_ADDRESS	EMP_SALARY	EMP_DEPT_NO	DOJ
100	Rahul Sharma	C-21, Arya Nagar	30000	D05	2012-02-06
101	Vikas Mittal	A/44, Mayur Vihar	27000	D03	2002-12-04
103	Sachin Vats	A-21, Ankur Vihar	24500	D02	2007-09-06
104	Uday Singh	D-34, Indraprasth	27000	D03	2001-07-06
105	Ravi Shukla	NULL	24500	D01	2012-05-06
106	Vinay Rana	120, DDA Colony	25000	D03	2001-02-06

6 rows in set (0.00 sec)

Delete All Rows

To delete all rows in a table without deleting the table structure, the following command is used

Syntax `DELETE FROM <table name>;`
or `DELETE * FROM <table name>;`

It is very easy to drop an existing MySQL table, but you need to be very careful while deleting any existing table because data lost will not be recovered after deleting a table.

Restructuring a Table (ALTER TABLE Command)

One can change the structure of a table after creating it, by using ALTER command. To restructure a table either by adding new columns or deleting existing columns, the ALTER command is used. It is also used to modify the structure of a table by modifying the definition of its columns.

The ALTER command is used to perform the following functions

Adding a Column in a Table

Syntax `ALTER TABLE <table_name> ADD <column_name> datatype <value>;`

For example, query to add a column named "EXPERIENCE" of data type INT of size 3 to the table COMPANY used in above section. `mysql>ALTER TABLE COMPANY ADD EXPERIENCE INT(3);`

To view the output of the above command use SELECT command.

```
mysql>SELECT * FROM COMPANY;
```

Now, above query produces the following output

```
+----------+-------------+-----------------+------------+-------------+------------+------------+
| EMP_CODE | EMP_NAME    | EMP_ADDRESS     | EMP_SALARY | EMP_DEPT_NO | DOJ        | EXPERIENCE |
+----------+-------------+-----------------+------------+-------------+------------+------------+
|      100 | Rahul Sharma| C-21, Arya Nagar|      30000 |         D05 | 2012-02-06 |       NULL |
|      101 | Vikas Mittal| A/44, Mayur Vihar|     27000 |         D03 | 2002-12-04 |       NULL |
|      103 | Sachin Vats | A-21, Ankur Vihar|     24500 |         D02 | 2007-09-06 |       NULL |
|      104 | Uday Singh  | D-34, Indraprasth|     27000 |         D03 | 2001-07-06 |       NULL |
|      105 | Ravi Shukla | NULL            |      24500 |         D01 | 2012-05-06 |       NULL |
|      106 | Vinay Rana  | 120, DDA Colony |      25000 |         D03 | 2001-02-06 |       NULL |
+----------+-------------+-----------------+------------+-------------+------------+------------+
6 rows in set (0.00 sec)
```

Deleting a Column from a Table

Syntax `ALTER TABLE<table_name> DROP COLUMN<column_name>;`
or `ALTER TABLE<table_name> DROP <column_name>;`

For example, query to drop a column name "EXPERIENCE" from the table COMPANY.

```
mysql>ALTER TABLE COMPANY DROP COLUMN EXPERIENCE;
```

To view the output of the above command use SELECT command.

```
mysql>SELECT * FROM COMPANY;
```

Now, above query produces the following output

```
+----------+-------------+-----------------+------------+-------------+------------+
| EMP_CODE | EMP_NAME    | EMP_ADDRESS     | EMP_SALARY | EMP_DEPT_NO | DOJ        |
+----------+-------------+-----------------+------------+-------------+------------+
|      100 | Rahul Sharma| C-21, Arya Nagar|      30000 |         D05 | 2012-02-06 |
|      101 | Vikas Mittal| A/44, Mayur Vihar|     27000 |         D03 | 2002-12-04 |
|      103 | Sachin Vats | A-21, Ankur Vihar|     24500 |         D02 | 2007-09-06 |
|      104 | Uday Singh  | D-34, Indraprasth|     27000 |         D03 | 2001-07-06 |
|      105 | Ravi Shukla | 45/A, Vivek Nagar|     24500 |         D01 | 2012-05-06 |
|      106 | Vinay Rana  | 120, DDA Colony |      25000 |         D03 | 2001-02-06 |
+----------+-------------+-----------------+------------+-------------+------------+
6 rows in set (0.00 sec)
```

Changing the Data Type of a Column in a Table

Syntax `ALTER TABLE <table_name> MODIFY COLUMN <column_name> datatype<value>;`

or `ALTER TABLE <table_name> MODIFY <column_name> datatype<value>;`

For example, query to change the data type of a column DOJ to datatype VARCHAR (15) from the table COMPANY used in above section.

```
mysql>ALTER TABLE COMPANY MODIFY COLUMN DOJ VARCHAR
(15);
```

Above query produce a following output

```
+----------------------------------------+
| Query OK, 6 rows  affected (0.17 sec)  |
| Records : 6 Duplicates : 0 Warnings : 0 |
+----------------------------------------+
```

Add and Drop Constraints

Syntax

To ADD PRIMARY KEY or UNIQUE KEY constraint

```
ALTER TABLE table_name
ADD CONSTRAINT <constraint_name>
PRIMARY KEY/UNIQUE KEY <column_name>;
```

To ADD FOREIGN KEY constraint

```
ALTER TABLE table_name
ADD CONSTRAINT <constraint_name>
FOREIGN KEY(col1, col2,..., coln)
REFERENCES <parent_table>(col1, col2,...,coln);
```

To ADD CHECK constraint

```
ALTER TABLE table_name
ADD CONSTRAINT <constraint_name> CHECK (column_name
condition);
```

To DROP constraints

```
ALTER TABLE table_name
DROP CONSTRAINT constraint_name;
```

*For example,*To add primary key constraint employee_pk in EMPLOYEE table.

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT employee_pk
PRIMARY KEY(emp_id);
```

*For example,*To drop a constraint chk_person from table PERSONS.

```
ALTER TABLE PERSONS
DROP CONSTRAINT chk_Persons;
```

Disable and Enable Constraints

Syntax To DISABLE constraints

```
ALTER TABLE table_name
DISABLE CONSTRAINT constraint_name;
```

To Enable constraints

```
ALTER TABLE table_name
ENABLE CONSTRAINT constraint_name;
```

*For example,*To disable a foreign key constraint employee_fk from EMPLOYEE table.

```
ALTER TABLE EMPLOYEE
DISABLE CONSTRAINT employee_fk;
```

*For example,*To enable a primary key constraint employee_pk from EMPLOYEE table.

```
ALTER TABLE EMPLOYEE
ENABLE CONSTRAINT employee_pk;
```

RENAME Command The RENAME command is used to change the name of the table or the database object.

Syntax `RENAME table oldtable_name TO newtable_name;`

For example, To rename the STUDENT table as my_students.

```
RENAME table STUDENT TOmy_students;
```

DROP TABLE Command

SQL DROP TABLE statement is used to remove table in a database. When you use the SQL DROP TABLE statement to remove a table, the database deletes all objects.

Syntax

```
DROP TABLE table_name;
```

Here, `table_name` is the name of table to be deleted.

ORDER BY Clause

The ORDER BY keyword is used to sort the result set along a specified column with the SELECT command. The ORDER BY keyword sorts the records in ascending order by default. If you want to sort the records in a descending order, you can use the DESC keyword.

Syntax

```
SELECT column_name(s)
FROM table_name
ORDER BY column_name(s)ASC/DESC;
```

e.g. If we have the following PERSONS table

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

and we want to sort the persons by their last name in ascending order.

We use the following SELECT statement

```
SELECT * FROM PERSONS
ORDER BY LastName ASC;
```

The result set will look like this

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
2	Svendson	Tove	Borgvn 23	Sandnes

Chapter Practice

PART 1
Objective Questions

• Multiple Choice Questions

1. The clause used to check NULL values is
(a) IS NULL (b) IS NOT NULL
(c) Both (a) and (b) (d) None of these

Ans. (*a*) The IS NULL clause is used to check NULL values in a field.

2. The ..... operator is used for pattern matching.
(a) BETWEEN (b) LIKE
(c) IN (d) LOOKSLIKE

Ans. (*b*) The LIKE operator is used to match patterns in a field.

3. The two characters used for pattern matching using LIKE operator are
(a) _ , * (b) * , / (c) // , / (d) %, _

Ans. (*d*) The %, _ are the two characters used for pattern matching of values in a column.

4. The clause used to specify a condition in a query is
(a) MATCH (b) WHOSE
(c) WHERE (d) None of these

Ans. (*c*) The WHERE clause is used to specify a condition in a query.

5. To delete all the records from a table "Product" the command will be
(a) `DEL FROM Product;`
(b) `DELETE FROM Product;`
(c) `REMOVE ALL FROM Product;`
(d) `DELETE All;`

Ans. (*b*) To delete all the records from a table "Product" the command will be

```
DELETE FROM product;
```

6. The ..... character displays all the columns of a table in a SELECT query.
(a) # (b) @
(c) * (d) /

Ans. (*c*) The * character displays all the columns in a SELECT query.

7. The ..... command removes a table completely.
(a) DELETE (b) REMOVE
(c) DROP (d) UPDATE

Ans. (*c*) The DROP command removes a table completely along with its data.

8. The "SET" clause is used along with ..... command.
(a) DELETE (b) DESCRIBE
(c) CREATE (d) UPDATE

Ans. (*d*) The UPDATE command updates data of a table . It uses the "SET" clause to specify the field to be updated.

9. What is true about the following SQL statement? **(NCERT)**

```
mysql> SELECT*FROM Student;
```

(a) Displays contents of table 'Student'.
(b) Displays column names and contents of table 'Student'.
(c) Results in error as improper case has been used.
(d) Displays only the column names of table 'Student'.

Ans. (*b*) The command displays entire contents of the table along with column names.

10. What will be the output of following query? **(NCERT)**

```
INSERT INTO Student
VALUES ("Suhana",109,'F'),
VALUES ("Rivaan",102,'M'),
VALUES ("Atharv",103,'M'),
VALUES ("Rishika",105,'F'),
VALUES ("Garvit",104,'M'),
VALUES ("Shaurya",109,'M');
```

(a) Error
(b) No Error
(c) Depends on compiler
(d) Successful completion of the query

Ans. (*a*) Multiple values cannot be inserted in a single INSERT command.

11. Which function is used to replace NULL value with another value?
(a) IFNULL (b) IS NULL
(c) IS NOT NULL (d) None of these

Ans. (*a*) IFNULL function is used to replace NULL value with another value.

12. Which operator is used to compare a value to a specified list of values?

(a) ANY (b) BETWEEN
(c) ALL (d) IN

Ans. (*d*) The IN operator easily tests the expression, if it matches any value in a specified list of value.

13. If we have not specified ASC or DESC after a SQL ORDER By clause, the following is used by default

(a) DESC (b) ASC
(c) There is no default value (d) None of these

Ans. (*b*) If we have not specified any sorting with the ORDER By clause. SQL always uses the ASC as a default sorting order.

14. Which of the following is the correct order of a SQL statement?

(a) SELECT, GROUP By, WHERE, HAVING
(b) SELECT, WHERE, GROUP BY, HAVING
(c) SELECT, HAVING, WHERE, GROUP BY
(d) SELECT, WHERE, HAVING, GROUP BY

Ans. (*b*) In SQL statement, the WHERE clause always comes before GROUP BY and HAVING clause always comes after GROUP BY. Hence, option (b) is correct.

• Case Based MCQs

Direction *Read the case and answer the following questions.*

15. Ronita wants to store the data of some products in a table as follows

PNo	PName	Qty	Date_Of_Mfg
P01	Pencil	20	2020-09-01
P02	Eraser	5	1990-09-11
P03	Book	16	2000-04-03
P04	Notebook	15	2016-12-11
P05	Color	10	2015-02-04

She also wants to perform some operations and manipulations on the table . Help her to find the solutions of following questions

(i) A command that displays the details of all the products will be

(a) `SELECT * FROM Product;`
(b) `SHOW * FROM Product;`
(c) `DISPLAY * FROM Product;`
(d) `SELECT ALL details FROM Product;`

(ii) The default date format in which date has to be stored in MySQL is

(a) DD-MM-YYYY (b) DD-YY-MM
(c) MM-YY-DD (d) YYYY-MM-DD

(iii) Which command she can use to add a new column to the table?

(a) INSERT (b) UPDATE
(c) ADD COLUMN (d) ALTER

(iv) Suggest her a proper data type for the "PName" column.

(a) Varchar
(b) Double
(c) Float
(d) Integer

(v) She is confused whether she has to use the "COLUMN" clause with the ALTER TABLE command to add a column to the table. What should she do ?

(a) COLUMN clause is must.
(b) COLUMN clause is optional.
(c) COLUMN clause is must for adding integer columns only.
(d) None of the above

Ans. (i) (*a*) `SELECT * FROM Product;`
(ii) (*d*) By default, MySQL stores date in YYYY-MM-DD format.
(iii) (*d*) The ALTER command can be used to make any changes to the structure of a table.
(iv) (*a*) The varchar is a variable length data type that can used for columns storing string/character type of data.
(v) (*b*) With the ALTER TABLE command the COLUMN clause is optional, in adding columns to a table.

16. Sonali wants to perform certain operations on a table "Exam" storing exam details . She is not sure about some of the commands and is getting errors.

Help her in proper execution of her operations.

Exam_ID	ExamName	MaxMarks	PassMarks
E01	HalfYearly	45	14.0
E02	Term-I	35	10.5
E03	PreBoard	50	15.0
E04	UnitTest	20	7.0
E05	Term-II	35	10.5

(i) The command she should write to increase the PassMarks by 5 will be

(a) `UPDATE Exam SET PassMarks=5;`
(b) `UPDATE Exam SET PassMarks+5;`
(c) `UPDATE Exam SET PassMarks= PassMarks+5;`
(d) `UPDATE Exam SET PassMarks= PassMarks+5`

(ii) She wants to add a new column "NegativeMarks" of type integer. She wrote the following command. What is the error?

```
ALTER Exam ADD NegativeMarks integer;
```

(a) Command cannot have ; at the end.
(b) Command is missing "TABLE" clause after "ALTER".
(c) Command has to be written in uppercase.
(d) The "COLUMN" clause has to be added after "ADD"

(iii) What data type she has to use for the "PassMarks" column?

(a) Integer (b) Float
(c) String (d) Char

(iv) She is confused whether she has to use DELETE or DROP command for deleting all data of the table keeping the structure. Which command she has to use?

(a) DELETE (b) DROP
(c) Any of (a) or (b) (d) None of these

(v) Can she add another ExamID as "E05"? Assuming Exam ID is primary key of table

(a) Yes (b) No
(c) Yes , if other values are NULL.
(d) None of these

Ans. (i) (*d*) The UPDATE command updates data of table and the modifying field can be set to a value or expression.

(ii) (*b*) The actual command will be
```
ALTER TABLE Exam ADD NegativeMarks integer;
```

(iii) (*b*) The "PassMarks" column stores fractional values , hence the data type should be float.

(iv) (*a*) The DELETE command can be used to delete all the records from table Exam.

(v) (*b*) ExamID is the primary key of the table, so it cannot have duplicate values.

PART 2
Subjective Questions

• Short Answer Type Questions

1. Differentiate between ALTER and UPDATE commands in SQL. **(NCERT)**

Ans.

ALTER command	UPDATE command
It belongs to DDL category.	It belongs to DML category.
It changes the structure of the table.	It modifies data of the table.
Columns can be added, modified , deleted etc.	Data can be changed, updated with values and expressions.

2. How is char data type different from varchar data type? **(NCERT)**

Ans.

Char	Varchar
It is fixed length.	It is variable length.
Wastage of memory.	Memory usage only as per data size.
Less useful.	Better data type.

3. Explain the use of ORDER BY clause.

Ans. The ORDER BY clause is used to arrange the records in ascending or descending order. Data present in a table can be arranged as per requirement on a specific field in ascending or descending order. The default is ascending order. To arrange in descending order the DESC clause is to be used. To arrange in ascending order ASC may be used.

e.g.
```
SELECT * FROM Employee ORDER BY EMP_SALARY DESC;
```
The above command arranges the records in descending order of salary.

4. What is the use of AS keyword with SELECT statement?

Ans. In MySQL, the AS keyword is used to temporarily rename the column's name, i.e. the AS keyword is used to define the column alias. This renaming is a temporary change and the actual column name does not change in the database.

5. Name two wildcard characters used in conjunction with the LIKE operator.

Ans. Two wildcard characters used in conjunction with the LIKE operator are given below :

(i) Per cent sign (%) for matching any substring.
(ii) Underscore sign (_) for matching a single character.

6. Write the queries for the following questions using the table Product with the following fields.

(P_ Code, P_Name, Qty, Price)

(i) Display the price of product having code as P06.

(ii) Display the name of all products with quantity greater than 50 and price less than 500.

Ans. (i)
```
SELECT Price FROM Product WHERE P_Code="P06";
```
The criteria of the records that are to be displayed can be specified with WHERE clause of SQL.

(ii)
```
SELECT P_Name FROM Product WHERE Qty>50 AND Price<500;
```
The criteria of the records that are to be displayed can be specified with WHERE clause of SQL. Here, the condition is quantity > 50 and price<500 .

7. Is it compulsory to provide values for all columns of a table while adding records? Give an example.

Ans. No it is not compulsory to provide values for all columns of a table while adding records. We can use NULL values wherever values are missing.

e.g.
```
INSERT INTO Employee VALUES (1,NULL,"Sales",89000);
```

8. Amit wrote the command to create a table "Student" as :
```
CREATE TABLE Student(RollNo integer, Name varchar(20), Marks float(8,2));
```
What does (8,2) mean here?

Ans. While specifying float columns in a table the width and the number of decimals have to be specified. Here 8 is the total width and 2 is the number of decimal places for the Marks column.

9. Rakesh wants to increase the price of some of the products by 20% , of his store whose price is less than 200. Assuming the following structure , what will be the query?

PNo	PName	Quality	Price

Ans. `UPDATE ITEM SET Price=Price + Price * 0.2 WHERE Price<200 ;`

The UPDATE command updates data of a table . While updating, the expression for update value can be assigned to the updating field. The records to be updated can be specified as WHERE condition.

10. Write the use of LIKE clause and a short explanation on the two characters used with it.

Ans. This operator is used to search a specified pattern in a column. It is useful when you want to search rows to match a specific pattern or when you do not know the entire value.The SQL LIKE clause is used to compare a value to similar values using wildcard characters.

We describe patterns by using two special wildcard characters, given below:

(i) The per cent sign (%) is used to match any substring.
(ii) The underscore (_) is used to match any single character.

The symbols can also be used in combinations.

11. Given the command below.

DELETE FROM Toys WHERE ToyName LIKE "S_t%";

Which records will be deleted by the above command?

Ans. The command has a LIKE clause with "S_t%" which means all the toy names that start with the letter 'S' and has 3rd letter as 't' will deleted.

12. In the following query how many rows will be deleted? **(NCERT)**

```
DELETE Student
WHERE Student_ID=109;
```

(Assuming a Student table with primary key Student_ID)

Ans. `DELETE FROM Student WHERE Student_ID=109;`

Here, the "FROM" clause is missing , so the command will produce an error.

13. If the value in the column is repeatable, how do you find out the unique values? **(NCERT)**

Ans. The DISTINCT clause in SQL is used to display only distinct values in a column of a table. Hence, if the column allows duplicate values the unique values can be extracted using the DISTINCT clause.

`SELECT DISTINCT CLASS FROM Student ;`

This displays only the unique classes.

14. What do you mean by an operator? Name any four operators used in queries.

Ans. An operator is a component of an expression that represents the action that should be taken over a set of values.

Four operators used in queries are

(i) Arithmetic operators
(ii) Comparison operators
(iii) Boolean/Logical operators
(iv) Between operator

15. How NOT operator is used with WHERE clause? Give an example.

Ans. The WHERE clause is used to retrieve some given data according to the condition and NOT operator reverses the result of it.

For example,

```
mysql>SELECT Name, Class, Games FROM
Student_table WHERE NOT Games = 'FootBALL';
```

16. Consider the following table with their fields.

```
EMPLOYEE (E_CODE, E_NAME, DESIG, SALARY,
                                DOJ)
```

List the names, salary, PF, HRA, DA of all the employees in the EMPLOYEE Table. HRA is 25% of salary, DA is 10% of salary and PF is 5% of salary. The result should be in descending order of salary.

Ans.

```
mysql>SELECT E_NAME, SALARY, SALARY *0.25 AS
HRA, SALARY * 0.10 AS DA,
        SALARY *0.05 AS PF
        FROM EMPLOYEE
        ORDER BY SALARY DESC;
```

17. What are the functions of ALTER TABLE command?

Ans. The main functions of ALTER TABLE command are

(i) Add or drop columns.
(ii) Change the column definition of a column.
(iii) Add or drop constraint.
(iv) Rename a column.

18. Write syntax of the conditions given below.

(i) Add a column in a table.
(ii) Delete a column from a table.

Ans.

(i) ALTER TABLE<table_name>ADD <column_name>datatype<value>;
(ii) ALTER TABLE<table_name>DROP COLUMN<column_name>;

19. Consider the following table PREPAID. Write MySQL commands for the statements given below.

S_No	C_Name	Model	Connection
1.	Sita	Nokia	Airtel
2.	Geeta	Samsung	Idea
3.	Ritesh	LG	BSNL
4.	Jayant	Micromax	Reliance

(i) DELETE a column name Model.

(ii) DELETE a customer record where connection type is BSNL.

Ans.

(i) mysql> ALTER TABLE PREPAID DROP Model;
(ii) mysql> DELETE FROM PREPAID WHERE Connection = 'BSNL';

20. Is it possible to disable a constraint? Give reasons in support of your answer.

Ans. Yes, we can disable a constraint using keyword DISABLE.

```
ALTER TABLE table_name
DISABLE CONSTRAINT constraint_name;
```

21. Sarthak, a student of class XII, created a table "CLASS". Grade is one of the columns of this table. To find the details of students whose Grades have not been entered. He wrote the following MySQL query, which did not give the desired result?

```
SELECT * FROM CLASS WHERE Grade = "Null";
```

Help Sarthak to run the query by removing the errors from the query and write the correct query.

Ans. Query to find the details of students whose Grade have not been entered:

```
SELECT * FROM CLASS WHERE Grade IS NULL;
```

22. Define UPDATE command of MySQL with its basic syntax and also give one of its example.

Ans. An UPDATE command is used to directly change or modify the values stored in one or more fields in a specified record.

Syntax,

```
UPDATE<table_name>SET[<column1>=<value 1>, <column 2>= <value 2>.....]
```

Where <condition>;

For example,

Consider the given table PREPAID in above question.

mysql>UPDATE PREPAID SET Model = 'Sony' WHERE S_No=2;

23. What will be the output of the following queries on the basis of EMPLOYEE table?

Table: EMPLOYEE

Emp_Id	Name	Salary
E01	Siya	54000
E02	Joy	NULL
E03	Allen	32000
E04	Neev	42000

(i) SELECT Salary + 100 FROM EMPLOYEE WHERE Emp_Id = 'E02';

(ii) SELECT Name FROM EMPLOYEE WHERE Emp_Id = 'E04';

Ans. The output of the following queries

(i)

Salary+100
NULL

(ii)

Name
Neev

• Long Answer Type Questions

24. Consider the table Hospital storing details of patients as follows

Table : Hospital

PatId	PName	Dept	Charges	DtofAdm
P1	Varun	PAED	700	2021-09-02
P2	Sunita	PAED	900	NULL
P3	Samarpan	ENT	1000	2020-08-09
P4	Rishabh	ORTHO	500	1995-06-05
P5	Bineeta	ORTHO	450	2012-03-04

Write SQL commands for (i) to (iv).

(i) To create the table with appropriate data types.

(ii) To display only the names and departments of patients of "ENT" .

(iii) To increase the charges of all departments by 20%.

(iv) To add a new column Phone of type integer.

Ans. (i)
```
CREATE TABLE Hospital(PatId char(5), Pname varchar(30), Dept varchar(20), Charges int, DtofAdm Date);
```
(ii)
```
SELECT PName, Dept FROM Hospital WHERE Dept = "ENT";
```
(iii)
```
UPDATE Hospital SET Charges = Charges + Charges * 0.2;
```
(iv)
```
ALTER TABLE Hospital ADD Phone integer;
```

25. With respect to the table "Toys" write SQL commands to perform the following

ToyID	ToyName	Price	Type
T01	Doll	520	Girls
T02	Video Game	200	Boys
T03	Gun	1600	Boys
T04	Anabelle	1000	Girls
T05	Hot wheels	100	Boys

(i) To add a record "T06,Toy Train,900,Boys" into the table.

(ii) To display ToyName and Price for all toys.

(iii) To display only toynames whose price is greater than 1000.

(iv) To add a column "Quantity" of type int to store quantities of toys.

Ans. (i)
```
INSERT INTO Toys VALUES("T06", "Toy Train", 900, "Boys");
```
(ii)
```
SELECT Toyname, Price FROM Toys;
```
(iii)
```
SELECT Toyname FROM Toys WHERE price>1000;
```
(iv)
```
ALTER TABLE Toys ADD Quantity integer;
```

26. Explain different DML commands with one example of each.

Ans. DML stands for Data Manipulation Language DML commands are those that work with the data and records of a table.

Assuming a table structure as follows

Table : Movie

MovieID	Mname	Type	Cost

Commands :

(i) **SELECT** The SELECT command displays the records of a table with or without conditions.

e.g. To display details of all movies.

```
SELECT * FROM Movie;
```

```
SELECT Mname, Cost FROM Movie WHERE Cost > 100000;
```

(ii) **UPDATE** The UPDATE command updates or modifies the data of a table by a value or expression.

e.g. To increase cost of all movies by 1000.

```
UPDATE Movie SET Cost=cost+1000;
```

(iii) **DELETE** The DELETE command is used to delete records from a table.

e.g. To delete all the records of the table.

```
DELETE FROM Movie;
```

or Delete the records of movie Type"Fiction".

```
DELETE FROM Movie WHERE Type="Fiction";
```

(iv) **INSERT INTO** The INSERT INTO command is used to insert / add records to a table.

e.g. To add a new record to the table.

```
INSERT INTO Movie VALUES("M01",
"Enter the Dragon","Action",2500000);
```

27. Write SQL queries with respect to the Movie table given below **(NCERT)**

Table : Movie

Movie ID	Movie Name	Category	Release Date	Production Cost	Business Cost
001	Hindi_ Movie	Musical	2018-04-23	124500	130000
002	Tamil_ Movie	Action	2016-05-17	112000	118000
003	English_ Movie	Horror	2017-08-06	245000	360000
004	Bengali_ Movie	Adventure	2017-08-04	72000	100000
005	Telugu_ Movie	Action	-	100000	-
006	Punjabi_ Movie	Comedy	-	30500	-

(i) Retrieve movie information without mentioning their column names.

(ii) List business done by the movies showing only MovieID, MovieName and Business Cost.

(iii) List the different categories of movies.

(iv) Find **net profit** of each movie showing its ID, Name and NetProfit.

Ans. (i)
```
SELECT * FROM Movie;
```
The * operator consider all the columns of a table while displaying records with SELECT command.

(ii)
```
SELECT MovieID,MovieName,BusinessCost FROM Movie;
```
The SELECT command when used with the column list , it displays only the specified columns.

(iii)
```
SELECT DISTINCT Category FROM Movie;
```
The DISTINCT clause extracts unique values from a column of a table.

(iv)
```
SELECT MovieID, MovieName,
BusinessCost-ProductionCost AS " NetProfit"
FROM Movie;
```
Columns along with the calculated columns can be specified with the SELECT command to display them, where the expression of calculation can be specified in the field list.

28. Explain the use of the following clauses

(i) BETWEEN

(ii) ORDER BY

(iii) DISTINCT

(iv) LIKE

Ans. (i) **BETWEEN** The BETWEEN clause is used to specify ranges in a query. The values can be numbers, text or dates. The range consists of a beginning expression, followed by an AND keyword and an end expression.

For example, The command to displays the details of employees whose salary lies between 10000 and 20000.

```
SELECT*FROM Employee WHERE Sal BETWEEN 10000 AND 20000;
```

(ii) **ORDER BY** The ORDER BY clause is used to arrange the records in ascending or descending order. The clauses ASC and DESC are used to specify ascending or descending.

For example, The command to arrange the records of Employee table in descending order of names.

```
SELECT * FROM Employee ORDER BY Name DESC;
```

(iii) **DISTINCT** The DISTINCT clause lists only the unique values in a field.

For example, The command to displays only the different departments in the Employee table.

```
SELECT DISTINCT Dept FROM Employee;
```

(iv) **LIKE** The LIKE clause is used for pattern matching in SQL.

For example, The command to displays the details of employees whose name starts with "S".

```
SELECT*FROM Employee WHERE Ename LIKE "S%",
```

29. Write SQL commands for the questions from (i) to (iv) on the basis of table SHOP.

S_No	P_Name	S_Name	Qty	Cost	City
S1	Biscuit	Priyagold	120	12.00	Delhi
S2	Bread	Britannia	200	25.00	Mumbai
S3	Chocolate	Cadbury	350	40.00	Mumbai
S4	Sauce	Kissan	400	45.00	Chennai

(i) Display all products whose quantity is between 100 and 400.

(ii) Display data for all products sorted by their quantity.

(iii) To list S_Name, P_Name, Cost for all the products whose quantity is less than 300.

(iv) To display S_No, P_Name, S_Name, Qty in descending order of quantity from the SHOP table.

Ans.

(i)
```
mysql> SELECT * FROM SHOP WHERE Qty BETWEEN 100
AND 400;
```

(ii)
```
mysql> SELECT * FROM SHOP ORDER BY Qty;
```

(iii)
```
mysql> SELECT S_Name, P_Name, Cost FROM SHOP
WHERE Qty<300;
```

(iv)
```
mysql> SELECT S_No, P_Name, S_Name, Qty FROM
SHOP ORDER BY Qty DESC;
```

30. Consider the table STUDENT given below and write SQL commands for (i) to (iv).

Student_No	Class	Name	Game	G_Grade	Section	Marks
01	7	Rahul	Swimming	B	A	99
02	8	Sameer	Tenis	A	B	20
03	10	Dushyant	FootBall	C	C	87
04	12	Kapil	Tennis	D	C	90
05	6	Ravinder	Cricket	A	B	97

(i) Display Student_No and G_Grade of all students from table STUDENT.

(ii) List the Name from table STUDENT whose Student _No is 04 or 05 or 02.

(iii) Display Game and Marks for those students whose name starts with alphabet 'D'.

(iv) Write a query to display Name and Section for those students whose Marks lies between 85 to 100.

Ans.

(i)
```
mysql>SELECT Student_No, G_Grade FROM
STUDENT;
```

(ii)
```
mysql>SELECT Name FROM STUDENT WHERE Student
_No IN (04, 05, 02);
```

(iii)
```
mysql>SELECT Game, Marks FROM STUDENT WHERE
Name LIKE 'D%';
```

(iv)
```
mysql>SELECT Name, Section FROM STUDENT WHERE
Marks BETWEEN 85 AND 100;
```

31. In a database there are two tables:

Table: ITEM

Item_Code	Item_Name	Price
111	Refrigerator	90000
222	Television	75000
333	Computer	42000
444	Washing Machine	27000

Table: BRAND

Item_Code	Brand_Name
111	LG
222	Sony
333	HCL
444	IFB

Write MySQL queries for the following:

(i) To display Item_Code, Item_Name and corresponding Brand_Name of those items, whose Price is between 20000 and 40000 (both values included).

(ii) To display Item_Code, Price and Brand_Name of the item which has Item_Name as Computer.

(iii) To increase the prices of all the items by 10%.

(iv) Update the Brand_Name with Toshiba whose Item _Code is 333.

Ans.

(i)
```
SELECT I.Item_Code, I.Item_Name,B.Brand_Name
FROM ITEM I, BRAND B
WHERE I.Item_Code=B.Item_Code AND
I.Price BETWEEN 20000 AND 40000;
```

(ii)
```
SELECT I.Item_Code, I.PRICE,
B.Brand_Name
FROM ITEM I, BRAND B
Where I.Item_Code = B.Item_Code AND
I.Item_Name="Computer";
```

(iii)
```
UPDATE ITEM SET
Price=Price+(Price*0.1);
```

(iv)
```
UPDATE BRAND SET Brand_Name = 'Toshiba' WHERE
Item_Code = 333;
```

32. In a database Multiplexes, there are two tables with the following data. Write MySQL queries for (i) to (iv), which are based on TICKETDETAILS and AGENTDETAILS.

Table: TICKETDETAILS

Tcode	Name	Tickets	A_code
S001	Meena	7	A01
S002	Vani	5	A02
S003	Meena	9	A01
S004	Karish	2	A03
S005	Suraj	1	A02

Table: AGENTDETAILS

A_code	AName
A01	Mr. Robin
A02	Mr. Ayush
A03	Mr. Trilok
A05	Mr. Johon

(i) To display Tcode, Name and AName of all the records where the number of tickets sold is more than 5.

(ii) To display total number of tickets booked by agent "Mr. Ayush".

(iii) To display Acode, AName and corresponding Tcode where AName ends with "k".

(iv) To display maximum tickets and AName according to Acode.

Ans. (i)
```
SELECT Tcode, Name, AName
FROM TICKETDETAILS, AGENTDETAILS
WHERE TICKETDETAILS.A_code =
AGENTDETAILS.A_code AND Tickets > 5;
```
(ii)
```
SELECT AName, SUM(DISTINCT Tickets)
FROM TICKETDETAILS, AGENTDETAILS
WHERE AName = "Mr.Ayush"
AND A_Code = "A02";
```
(iii)
```
SELECT Acode, AName, Tcode
FROM TICKETDETAILS, AGENTDETAILS
WHERE TICKETDETAILS.A_code =
AGENTDETAILS.A_code AND AName LIKE
"%k";
```
(iv)
```
SELECT AName, MAX(Tickets) FROM
TICKETDETAILS,AGENTDETAILS
WHERE TICKETDETAILS.A_code
= AGENTDETAILS.A_code;
```

33. Write SQL commands for the question from (i) to (viii) on the basis of table MASTER (contains S.No. of employees)

S.No	Name	Age	Department	Salary
1	Shyam	21	Computer	12000
2	Shiv	25	Maths	15000
3	Rakesh	31	Hindi	14000
4	Sharmila	32	History	20000
5	Dushyant	25	Software	30000

(i) Write a command to update the salary of the employee to 40000, whose S. No is 3.

(ii) Write a query to add a column Date_of_ Joining to the table MASTER.

(iii) Show Age, Department of those employees whose salary is greater than 12000.

(iv) List all data of table MASTER.

(v) Write a query to change the data type of a column Name to varchar with size 35.

(vi) Write a command to delete from the table MASTER those employees whose name is Rakesh.

(vii) Write a command to update the department of the employee to english, whose name is Dushyant.

(viii) Write a command to delete the table with the structure.

Ans. (i)
```
mysql>UPDATE MASTER SET Salary= 40000 WHERE
S.No =3;
```
(ii)
```
mysql>ALTER TABLE MASTER ADD Date_of_Joining
DATE;
```
(iii)
```
mysql>SELECT Age, Department FROM MASTER WHERE
Salary>12000;
```
(iv)
```
mysql>SELECT * FROM MASTER;
```
(v)
```
mysql>ALTER TABLE MASTER MODIFY Name VARCHAR
(35);
```
(vi)
```
mysql>DELETE FROM  MASTER WHERE Name =
"Rakesh";
```
(vii)
```
mysql>UPDATE MASTER SET Department = "English"
WHERE Name = "Dushyant";
```
(viii)
```
mysql>DROP TABLE MASTER;
```

34. Consider the table 'PERSONS' given below. Write commands in SQL for (i) to (iv).

Table: PERSONS

Pld	SurName	FirstName	Gender	City	PinCode	BasicSalary
1	Sharma	Geet	F	Udhamwara	182141	50000
2	Singh	Surinder	M	Kupwara Nagar	193222	75000
3	Jcob	Peter	M	Bhawani	185155	45000
4	Alvis	Thomas	M	Ahmed Nagar	380025	50000
5	Mohan	Garima	M	Nagar Coolangatta	390026	33000
6	Azmi	Simi	F	New Delhi	110021	40000
7	Kaur	Manpreet	F	Udhamwara	182141	42000

(i) Display the SurName, FirstName and City of people residing in Udhamwara city.

(ii) Display the Person Id(PId), City and PinCode of PERSONS in descending order of PinCode.

(iii) Display the FirstName and City of all the females getting BasicSalary above 40000.

(iv) Display FirstName and BasicSalary of all the persons whose FirstName start with "G".

Ans (i)
```
SELECT SurName, FirstName, City FROM PERSONS
WHERE City = 'Udhamwara';
```
(ii)
```
SELECT PId, City, PinCode FROM PERSONS
ORDER BY PinCode DESC;
```
(iii)
```
SELECT FirstName, City FROM PERSONS
WHERE Gender = 'F' AND BasicSalary > 40000;
```
(iv)
```
SELECT FirstName, BasicSalary FROM PERSONS
WHERE FirstName LIKE 'G%';
```

35. Answer the questions (i) to (iv) on the basis of the following tables SHOPPE and ACCESSORIES.

Table: SHOPPE

Id	SName	Area
S001	ABC Computeronics	CP
S002	All Infotech Media	GK II
S003	Tech Shoppe	CP
S004	Geeks Tecno Soft	Nehru Place
S005	Hitech Tech Store	Nehru Place

Table: ACCESSORIES

No	Name	Price	Id
A01	Mother Board	12000	S01
A02	Hard Disk	5000	S01
A03	Keyboard	500	S02
A04	Mouse	300	S01
A05	Mother Board	13000	S02
A06	Keyboard	400	S03
A07	LCD	6000	S04
T08	LCD	5500	S05
T09	Mouse	350	S05
T10	Hard Disk	4500	S03

(i) To display Name and Price of all the Accessories in ascending order of their Price.

(ii) To display Id and SName of all Shoppe located in Nehru Place.

(iii) To display Name, Price of all Accessories and their respective SName, where they are available.

(iv) To display name of accessories whose price is greater than 1000.

Ans. (i)
```
SELECT Name, Price
FROM ACCESSORIES
ORDER BY Price;
```
(ii)
```
SELECT Id, SName
FROM SHOPPE
WHERE Area = 'Nehru Place';
```
(iii)
```
SELECT Name, Price, SName
FROM ACCESSORIES A, SHOPPE S
WHERE A.Id = S.Id;
```
but this query enable to show the result because A.Id and S.Id are not identical.

(iv)
```
SELECT Name From
ACCESSORIES
WHERE Price>1000;
```

36. Consider the following tables STORE and answer the questions

Table : STORE

ItemNo	Item	Scode	Qty	Rate	LastBuy
2005	Sharpener Classic	23	60	8	31-JUN-09
2003	Balls	22	50	25	01-FEB-10
2002	Gel Pen Premium	21	150	12	24-FEB-10
2006	Gel Pen Classic	21	250	20	11-MAR-09
2001	Eraser Small	22	220	6	19-JAN-09
2004	Eraser Big	22	110	8	02-DEC-09
2009	Ball Pen 0.5	21	180	18	03-NOV-09

Write SQL commands for the following statements.

(i) To display details of all the items in the STORE table in ascending order of LastBuy.

(ii) To display ItemNo and Item name of those items from STORE table, whose Rate is more than ₹ 15.

(iii) To display the details of those items whose Supplier code (Scode) is 22 or Quantity in Store (Qty) is more than 110 from the table STORE.

(iv) To display the item with its quantity which include pen in their name.

Ans. (i)
```
SELECT * FROM STORE ORDER BY LastBuy;
```
(ii)
```
SELECT ItemNo, Item FROM STORE WHERE Rate>15;
```
(iii)
```
SELECT * FROM STORE WHERE Scode = 22 OR
Qty>110;
```
(iv)
```
SELECT Item, Qty FROM STORE WHERE Item LIKE
'%Pen%';
```

37. Consider the following tables STUDENT and STREAM. Write SQL commands for the statements (i) to (iv).

Table: STUDENT

SCODE	NAME	AGE	STRCDE	POINTS	GRADE
101	Amit	16	1	6	NULL
102	Arjun	13	3	4	NULL
103	Zaheer	14	2	1	NULL
105	Gagan	15	5	2	NULL
108	Kumar	13	6	8	NULL
109	Rajesh	17	5	8	NULL
110	Naveen	13	3	9	NULL
113	Ajay	16	2	3	NULL
115	Kapil	14	3	2	NULL
120	Gurdeep	15	2	6	NULL

Table: STREAM

STRCDE	STRNAME
1	SCIENCE+COMP
2	SCIENCE+BIO
3	SCIENCE+ECO
4	COMMERCE+MATHS
5	COMMERCE+SOCIO
6	ARTS+MATHS
7	ARTS+SOCIO

(i) To display the name of streams in alphabetical order from table STREAM.

(ii) To update GRADE to 'A' for all those students, who are getting more than 8 as POINTS.

(iii) ARTS+MATHS stream is no more available. Make necessary change in table STREAM.

(iv) To display student's name whose stream name is science and computer.

Ans. (i)
```
SELECT STRNAME FROM STREAM ORDER BY STRNAME;
```
(ii)
```
UPDATE STUDENT SET GRADE = 'A' WHERE POINTS > 8;
```
(iii)
```
DELETE FROM STREAM WHERE STRNAME = 'ARTS + MATHS';
```
(iv)
```
SELECT NAME FROM STUDENT WHERE STUDENT.STRCDE = STREAM.STRCDE AND STRNAME = "SCIENCE + COMP";
```

38. Consider the following tables GARMENT and FABRIC. Write SQL commands for the statements (i) to (iii).

Table: GARMENT

GCODE	DESCRIPTION	PRICE	FCODE	READYDATE
10023	PENCIL SKIRT	1150	F03	19-DEC-08
10001	FORMAL SHIRT	1250	F01	12-JAN-08
10012	INFORMAL SHIRT	1550	F02	06-JUN-08
10024	BABY TOP	750	F03	07-APR-07
10090	TULIP SKIRT	850	F02	31-MAR-07
10019	EVENING GOWN	850	F03	06-JUN-08
10009	INFORMAL PANT	1500	F02	20-OCT-08
10007	FORMAL PANT	1350	F01	09-MAR-08
10020	FROCK	850	F04	09-SEP-07
10089	SLACKS	750	F03	20-OCT-08

Table : FABRIC

FCODE	TYPE
F04	POLYSTER
F02	COTTON
F03	SILK
F01	TERELENE

(i) To display GCODE and DESCRIPTION of each GARMENT in descending order of GCODE.

(ii) To display the details of all the GARMENTs, which have READYDATE in between 08-DEC-07 and 16-JUN-08 (inclusive of both the dates).

(iii) To display garment's description with their price whose fabric is silk.

Ans. (i)
```
SELECT GCODE, DESCRIPTION FROM GARMENT ORDER BY GCODE DESC;
```
(ii)
```
SELECT*FROM GARMENT WHERE READYDATE BETWEEN '08-DEC-07' AND '16-JUN-08';
```
(iii)
```
SELECT DESCRIPTION, PRICE FROM GARMENT WHERE GARMENT.FCODE = FABRIC.FCODE AND TYPE = "SILK";
```

39. Consider the following tables. Write SQL commands for the statements (i) to (iv).

Table : SENDER

SenderID	SenderName	SenderAddress	SenderCity
ND01	R Jain	2, ABC Appts	New Delhi
MU02	H Sinha	12, Newtown	Mumbai
MU15	S Jha	27/A, Park Street	Mumbai
ND50	T Prasad	122-K, SDA	New Delhi

Table: RECIPIENT

RecID	SenderID	RecName	RecAddress	RecCity
KO05	ND01	R Bajpayee	5, Central Avenue	Kolkata
ND08	MU02	S Mahajan	116, A Vihar	New Delhi
MU19	ND01	H Singh	2A, Andheri East	Mumbai
MU32	MU15	P K Swamy	B5, C S Terminus	Mumbai
ND48	ND50	S Tripathi	13, B1 D, Mayur Vihar	New Delhi

(i) To display the names of all Senders from Mumbai.

(ii) To display the RecID, SenderName, SenderAddress, RecName, RecAddress for every Recipient.

(iii) To display Recipient details in ascending order of RecName.

(iv) To display the detail of recipients who are in Mumbai.

Ans. (i)
```
SELECT SenderName FROM SENDER WHERE SenderCity = 'Mumbai';
```

(ii) SELECT RecID, SenderName, SenderAddress, RecName, RecAddress
FROM RECIPIENT, SENDER WHERE RECIPIENT.SenderID = SENDER.SenderID;

(iii) SELECT * FROM RECIPIENT ORDER BY RecName;

(iv) SELECT * FROM RECIPIENT WHERE RecCity = "Mumbai";

40. Write the SQL commands for (i) to (v) on the basis of the table HOSPITAL

Table: HOSPITAL

No.	Name	Age	Department	Dateofadm	Charges	Sex
1	Sandeep	65	Surgery	23/02/98	300	M
2	Ravina	24	Orthopaedic	20/01/98	200	F
3	Karan	45	Orthopaedic	19/02/98	200	M
4	Tarun	12	Surgery	01/01/98	300	M
5	Zubin	36	ENT	12/01/98	250	M
6	Ketaki	16	ENT	24/02/98	300	F
7	Ankita	29	Cardiology	20/02/98	800	F
8	Zareen	45	Gynaecology	22/02/98	300	F
9	Kush	19	Cardiology	13/01/98	800	M
10	Shailya	31	Nuclear Medicine	19/02/98	400	M

(i) To show all information about the patients of Cardiology Department.

(ii) To list the name of female patients, who are in Orthopaedic Department.

(iii) To list names of all patients with their date of admission in ascending order.

(iv) To display Patient's Name, Charges, Age for male patients only.

(v) To display name of doctor are older than 30 years and charges for consultation fee is more than 500.

Ans. (i) SELECT * FROM HOSPITAL WHERE Department = 'Cardiology';

(ii) SELECT Name FROM HOSPITAL WHERE Department = 'Orthopaedic' AND Sex = 'F';

(iii) SELECT Name FROM HOSPITAL ORDER BY Dateofadm;

(iv) SELECT Name, Charges, Age FROM HOSPITAL WHERE Sex = 'M';

(v) SELECT NAME FROM HOSPITAL WHERE Age>30 AND Charges>500;

41. Write SQL commands for (i) to (v) on the basis of table INTERIORS.

Table: INTERIORS

No.	ITEMNAME	TYPE	DATEOFSTOCK	PRICE	DISCOUNT
1	Red rose	Double Bed	23/02/02	32000	15
2	Soft touch	Baby cot	20/01/02	9000	10
3	Jerry's home	Baby cot	19/02/02	8500	10
4	Rough wood	Office Table	01/01/02	20000	20
5	Comfort zone	Double Bed	12/01/02	15000	20
6	Jerry look	Baby cot	24/02/02	7000	19
7	Lion king	Office Table	20/02/02	16000	20
8	Royal tiger	Sofa	22/02/02	30000	25
9	Park sitting	Sofa	13/12/01	9000	15
10	Dine Paradise	Dining Table	19/02/02	11000	15
11	White Wood	Double Bed	23/03/03	20000	20
12	James 007	Sofa	20/02/03	15000	15
13	Tom look	Baby cot	21/02/03	7000	10

(i) To show all information about the Sofa from the INTERIORS table.

(ii) To list the ITEMNAME, which are priced at more than 10000 from the INTERIORS table.

(iii) To list ITEMNAME and TYPE of those items, in which DATEOFSTOCK is before 22/01/02 from the INTERIORS table in descending order of ITEMNAME.

(iv) To insert a new row in the INTERIORS table with the following data

{14, 'TrueIndian', 'Office Table', '25/03/03', 15000, 20}

(v) To display the name of item with their price which have discount more than 20.

Ans. (i) SELECT * FROM INTERIORS WHERE TYPE = 'Sofa';

(ii) SELECT ITEMNAME FROM INTERIORS WHERE PRICE > 10000;

(iii) SELECT ITEMNAME, TYPE FROM INTERIORS WHERE DATEOFSTOCK < '22/01/02'
ORDER BY ITEMNAME DESC;

(iv) INSERT INTO INTERIORS VALUES (14,'TrueIndian', 'Office Table', '25/03/03',15000,20);

(v) SELECT ITEMNAM, PRICE FROM INTERIORS WHRE DISCOUNT>20;

42. Write SQL commands for (i) to (iv) on the basis of table STUDENT.

TABLE: STUDENT

SNO	NAME	STREAM	FEES	AGE	SEX
1	ARUN KUMAR	COMPUTER	750.00	17	M
2	DIVYA JENEJA	COMPUTER	750.00	18	F
3	KESHAR MEHRA	BIOLOGY	500.00	16	M
4	HARISH SINGH	ENG. DR	350.00	18	M
5	PRACHI	ECONOMICS	300.00	19	F
6	NISHA ARORA	COMPUTER	750.00	15	F
7	DEEPAK KUMAR	ECONOMICS	300.00	16	M
8	SARIKA VASWANI	BIOLOGY	500.00	15	F

(i) List the name of all the students, who have taken stream as COMPUTER.

(ii) To display the number of students stream wise.

(iii) To display all the records in sorted order of name.

(iv) To display the stream of student whose name is Harish.

Ans. (i) SELECT NAME FROM STUDENT WHERE STREAM ='COMPUTER';

(ii) SELECT STREAM, COUNT(*) FROM STUDENT GROUP BY STREAM;

(iii) SELECT * FROM STUDENT ORDER BY NAME;

(iv) SELECT STREAM FROM STUDENT WHERE NAME LIKE "%HARISH%";

Chapter Test

Multiple Choice Questions

1. A table "Bus" exists with no rows and 6 columns . What is its cardinality?
(a) 0 (b) Such as table cannot exist
(c) 1 (d) 2

2. A table should have a
(a) foreign key (b) alternate key
(c) primary key (d) composite key

3. Riya wants to remove a column "Name" from her table , which command she has to use?
(a) ALTER TABLE (b) CLEAR
(c) UPDATE (d) None of these

4. The clause with ALTER TABLE command that renames a column is
(a) RENAME (b) CHANGE
(c) DROP (d) CHANGENAME

5. A table can have ..... alternate keys.
(a) 1 (b) 2
(c) 3 (d) multiple

Short Answer Type Questions

6. Explain the use of alias in a query statement.

7. Explain usage of IS NULL and IS NOT NULL clauses. (**NCERT**)

8. With respect to the following table structure write queries for the following

GameID	GName	Type	Players

(i) To display the details of games of "OUTDOOR" type.
(ii) To display GName and Players for games where players is more than 2.

9. Explain working of AND and OR operators in queries.

10. An organization ABC maintains a database EMP_DEPENDENT to record the following details about its employees and their dependents. (**NCERT**)

EMPLOYEE(AadhaarNo, Name, Address, Department, EmpID)

DEPENDENT(EmpID, DependentName, Relationship)

Use the EMP_DEPENDENT database to answer the following SQL queries:
(i) Find employee details working in a department, say 'PRODUCTION'.
(ii) Find employee names having no dependent.

Long Answer Type Questions

11. With respect to the following table "BOOK" write SQL queries

Table : Book

BookID	Bname	Publisher	Price	DtofPub
B1	Science Fiction	TMH	1200	2020-09-08
B2	Stories	PHI	900	NULL
B3	Ramayana	PHI	1700	NULL
B4	Beginners Cooking	Oswal	1400	1990-12-03

(i) Display details of books published before year 2000.
(ii) Display names and publishers of books whose price is less than 1000.
(iii) Display names of books who do not have a date of publication.
(iv) Increase price of all books by 200.

12. Write SQL queries with respect to the Employee table given below.

Table : Employee

Eno	Ename	Dept	Desig	DtofJoin	Salary
1	Jack	Sales	MGR	2012-09-12	89000
2	Priya	Accts	MGR	2005-04-22	56000
3	Ria	Pers	Clerk	2000-01-09	25000
4	Anil	Pers	Officer	1994-04-03	67000
5	Sumit	Sales	Officer	NULL	19000
6	Akash	Sales	Officer	NULL	20000

(i) Display name and department of employees whose name begins with" S".
(ii) Display details of employees whose designation ends with "r".
(iii) Display details of employees whose name has 1st letter "P" 3rd letter "i".
(iv) Display name,deptartment and salary of employees whose department name ends with "s".

13. Given the two tables.

Table : Student

Roll	Name	Marks	HostelId
1	Fiza	89	H1
2	Swati	78	H2
3	Anil	55	H3
4	Ria	68	H4
5	Prakash	12	H5

Table : Hostel

HostelId	Hname	Location
H1	Ganga	Kol
H2	Yamuna	Che
H3	Satluj	Mum
H4	Godavari	Bang

(i) Identify the primary keys of the two tables.
(ii) Identify the foreign key of Student table.
(iii) Can a student have HostelID "H5"?

Answers

Multiple Choice Questions

1. *(b)* ***2.*** *(c)* ***3.*** *(a)* ***4.*** *(b)* ***5.*** *(d)*