

UNIT 2

Database Query and SQL Math Functions

- **POW() or POWER()**

POWER(A, B) or POW(A, B) returns the number A raised to the power of another number B. Here the number A is the base and the number B is the exponent.

Needs two numbers as parameters.

SYNTAX: SELECT POW(A, B);

Examples:

1) mysql> select power(2,3);

```
+-----+
| power(2,3) |
+-----+
|      8      |
+-----+
```

1 row in set (0.05 sec)

2) mysql> select pow(2,3);

```
+-----+
| pow(2,3) |
+-----+
|      8      |
+-----+
```

1 row in set (0.00 sec)

3) mysql> select pow(2.0 , 3.0);

```
+-----+
| pow(2.0,3.0) |
+-----+
|      8      |
+-----+
```

1 row in set (0.00 sec)

- **ROUND()**

This function is used to round the number to the specified number of decimal places. Parameters required: the number to be rounded and the number of decimal places required.

If the number of decimal places required is not mentioned, then the result will not have decimal places

SYNTAX: SELECT ROUND(NUMBER, NUMBER OF DECIMAL PLACES)

Examples :

1) mysql> select round(2.25);

```
+-----+
| round(2.25) |
+-----+
|      2      |
+-----+
```

1 row in set (0.01 sec)

2) mysql>select round(2.25, 1);

```
+-----+
| round(2.25, 1) |
```

```
+-----+
|      2.3      |
```

```
+-----+
1 row in set (0.00 sec)
```

3) mysql>select round(2.25, 2);

```
+-----+
| round(2.25, 2) |
```

```
+-----+
|      2.25      |
```

```
+-----+
1 row in set (0.00 sec)
```

4) mysql>select round (2.26, 0);

```
+-----+
| round(2.26, 0) |
```

```
+-----+
|          2      |
```

```
+-----+
1 row in set (0.00 sec)
```

5) mysql>select round(135.43, 0);

```
+-----+
| round(135.43, 0) |
```

```
+-----+
|         135      |
```

```
+-----+
1 row in set (0.00 sec)
```

6) mysql>select round(135.53, 0);

```
+-----+
| round(135.53, 0) |
```

```
+-----+
|         136      |
```

```
+-----+
1 row in set (0.00 sec)
```

7) mysql>select round(135.55, 1);

```
+-----+
| round(135.55, 1) |
```

```
+-----+
|        135.6     |
```

```
+-----+
1 row in set (0.00 sec)
```

8) mysql>select round(135.55, -1);

```
+-----+
| round(135.55, -1) |
```

```
+-----+
|          140      |
```

```

+-----+
1 row in set (0.00 sec)
9) mysql>select round(134.45, -1);
+-----+
| round(134.45, -1) |
+-----+
|          130      |
+-----+
1 row in set (0.00 sec)
10)mysql>select round(134.45, -2);
+-----+
| round(134.45, -2) |
+-----+
|          100      |
+-----+
1 row in set (0.00 sec)
11)mysql>select round(154.45, -2);
+-----+
| round(154.45, -2) |
+-----+
|          200      |
+-----+
1 row in set (0.00 sec)

12)mysql>select round(1454.45, -2);
+-----+
| round(1454.45, -2) |
+-----+
|         1500      |
+-----+
1 row in set (0.00
sec)
13)mysql>select round(1444.45, -2);
+-----+
| round(1444.45, -2) |
+-----+
|         1400      |
+-----+
+
+-----+
+
1 row in set (0.00
sec)
14)mysql>select round(1444.45, -3);
+-----+
| round(1444.45, -3) |
+-----+
|          1000      |
+-----+
+ 1 row in set (0.00 sec)

```

```
15)mysql>select round(1544.45, -3);
```

```
+-----+
| round(1544.45, -3) |
+-----+
|          2000      |
+-----+
```

1 row in set (0.00 sec)

- **MOD()**

This function can be used to find modulus (remainder) when one number is divided by another.

Examples:

```
mysql>select mod(5,3)
```

```
+-----+
| mod(5,3) |
+-----+
|         2 |
+-----+
```

1 row in set (0.00 sec)

```
1) mysql> select mod(5,4);
```

```
+-----+
| mod(5,4) |
+-----+
|         1 |
+-----+
```

1 row in set (0.00 sec)

```
2) mysql>select mod(4,2);
```

```
+-----+
| mod(4,2) |
+-----+
|         0 |
+-----+
```

1 row in set (0.00 sec)

Text/String/Character Functions:

- **UCASE() / UPPER()**

Used to convert a character or text to uppercase.

Examples:

```
1) mysql>SELECT UCASE('hello');
```

```
+-----+
| UCASE('hello') |
+-----+
| HELLO          |
+-----+
```

1 row in set (0.00 sec)

2) mysql>SELECT Upper('hello');

```
+-----+
| Upper('hello') |
```

```
+-----+
| HELLO          |
```

```
+-----+
1 row in set (0.00 sec)
```

- **LCASE() / LOWER() :**

Used to convert a character or text to lowercase.

Examples:

1) mysql>select lcase('HELLO');

```
+-----+
| lcase('HELLO') |
```

```
+-----+
| hello          |
```

```
+-----+
1 row in set (0.00 sec)
```

2) mysql>select LOWER('HELLO');

```
+-----+
| LOWER('HELLO') |
```

```
+-----+
| hello          |
```

```
+-----+
1 row in set (0.00 sec)
```

- **MID() :** To extract a specified number of characters from the string.
First parameter is the text/string. Second parameter is the starting index and the third parameter is the number of characters required.
(Note: index starts with 1 and not 0.)

Examples:

1) mysql>SELECT MID('ABCDEFGHJKLMNOP', 1,4);

```
+-----+
| MID('ABCDEFGHJKLMNOP', 1,4) |
```

```
+-----+
| ABCD                        |
```

```
+-----+
1 row in set (0.00 sec)
```

2) mysql>SELECT MID('ABCDEFGHJKLMNOP', 1);

```
+-----+
| MID('ABCDEFGHJKLMNOP', 1) |
```

```
+-----+
| ABCDEFGHJKLMNOP          |
```

```
+-----+
+1 row in set (0.00 sec)
```

3) mysql> SELECT MID('ABCDEFGHJKLMNOP', -2,-1);

```
+-----+
| MID('ABCDEFGHJKLMNOP', -2,-1) |
```

```
+-----+
|                                |
```

+1 row in set (0.00 sec)

4) mysql>SELECT MID('ABCDEFGHJKLMNOP', 0,4);

```
+-----+
| MID('ABCDEFGHJKLMNOP', 0,4) |
```

```
+-----+
|                                |
```

1 row in set (0.00 sec)

(Please note the output of example 3, 4)

5) mysql>SELECT MID('ABCDEFGHJKLMNOP', 3,4);

```
+-----+
| MID('ABCDEFGHJKLMNOP', 3,4) |
```

```
+-----+
| CDEF                                |
```

+1 row in set (0.00 sec)

6) mysql>SELECT MID('ABCDEFGHJKLMNOP', -4,2);

```
+-----+
| MID('ABCDEFGHJKLMNOP', -4,2) |
```

```
+-----+
| MN                                |
```

+1 row in set (0.00 sec)

- **SUBSTRING()** : Same as MID() function

To extract a specified number of characters from thestring.

Examples:

1) mysql>SELECT SUBSTRING('ABCDEFGHJKLMNOP', 3,4);

```
+-----+
| SUBSTRING('ABCDEFGHJKLMNOP', 3,4) |
```

```
+-----+
| CDEF                                |
```

+1 row in set (0.00 sec)

2) mysql>SELECT SUBSTRING('ABCDEFGHJKLMNOP', 0,4);

```
+-----+
| SUBSTRING('ABCDEFGHJKLMNOP', 0,4) |
```

```
+-----+
|                                |
```

+1 row in set (0.00 sec)

3) mysql>SELECT SUBSTRING('ABCDEFGHJKLMNOP', 1,4);

```
+-----+
| SUBSTRING('ABCDEFGHJKLMNOP', 1,4) |
+-----+
| ABCD                               |
+-----+
```

+1 row in set (0.00 sec)

4) mysql>SELECT SUBSTRING('ABCDEFGHJKLMNOP', 4,2);

```
+-----+
| SUBSTRING('ABCDEFGHJKLMNOP', 4,2) |
+-----+
| DE                                 |
+-----+
```

+1 row in set (0.00 sec)

5) mysql>SELECT SUBSTRING('ABCDEFGHJKLMNOP', -4,2);

```
+-----+
| SUBSTRING('ABCDEFGHJKLMNOP', -4,2) |
+-----+
| MN                                 |
+-----+
```

+1 row in set (0.00 sec)

- **SUBSTR()** : Same as that of MID() and SUBSTRING()

Examples:

1) mysql>SELECT SUBSTR('ABCDEFGHJKLMNOP', -4,3);

```
+-----+
| SUBSTR('ABCDEFGHJKLMNOP', -4,3) |
+-----+
| MNO                             |
+-----+
```

+1 row in set (0.00 sec)

2) mysql>SELECT SUBSTR('ABCDEFGHJKLMNOP', 1,3);

```
+-----+
| SUBSTR('ABCDEFGHJKLMNOP', 1,3) |
+-----+
| ABC                             |
+-----+
```

+1 row in set (0.00 sec)

3) mysql>SELECT SUBSTR('ABCDEFGHJKLMNOP', 4,3);

```
+-----+
| SUBSTR('ABCDEFGHJKLMNOP', 4,3) |
+-----+
| DEF                             |
+-----+
```

+1 row in set (0.00 sec)

- **LENGTH()** :

This function returns the number of characters in the given text.

Examples:

1) mysql> SELECT LENGTH('HELLO WORLD');

```
+-----+
| LENGTH('HELLO WORLD') |
+-----+
| 11                     |
+-----+
```

+1 row in set (0.00 sec)
2) mysql> SELECT LENGTH(' ');

```
+-----+
| LENGTH(' ') |
```

```
+-----+
|      3      |
```

+1 row in set (0.00 sec)

3) mysql> SELECT LENGTH(' ');

```
+-----+
| LENGTH(' ') |
```

```
+-----+
|      1      |
```

```
+-----+
```

1 row in set (0.00 sec)

- **LEFT()** :Returns the specified number of characters including space starting from the leftmost characters.

Parameters required : text, number of characters to be extracted.

Examples:

1) mysql>SELECT LEFT('ABCDEFGHJKLMNOP',1);

```
+-----+
| LEFT('ABCDEFGHJKLMNOP',1) |
```

```
+-----+
| A                          |
```

+1 row in set (0.00 sec)

2) mysql> SELECT LEFT('ABCDEFGHJKLMNOP',2);

```
+-----+
| LEFT('ABCDEFGHJKLMNOP',2) |
```

```
+-----+
| AB                         |
```

+1 row in set (0.00 sec)

3) mysql> SELECT LEFT('ABCDEFGHJKLMNOP',3);

```
+-----+
| LEFT('ABCDEFGHJKLMNOP',3) |
```

```
+-----+
| ABC                       |
```

+1 row in set (0.00 sec)

4) mysql> SELECT LEFT('ABCDEFGHJKLMNOP',-1);

```
+-----+
| LEFT('ABCDEFGHJKLMNOP',-1) |
```

```
+-----+
```

+1 row in set (0.00 sec)

(**Note** : In the above example , the number of characters to be selected is -1 and hence characters are not extracted)

- **RIGHT()** :Returns the specified number of characters including space starting from the right of the text.

Parameters required : text, number of characters to be extracted.

Examples:

- 1) mysql> SELECT RIGHT('ABCDEFGHJKLMNOP',1);

```
+-----+
| RIGHT('ABCDEFGHJKLMNOP',1) |
+-----+
| P                           |
+-----+
```

+1 row in set (0.00 sec)

(Extracting 1 character)

- 2) mysql> SELECT RIGHT('ABCDEFGHJKLMNOP',2);

```
+-----+
| RIGHT('ABCDEFGHJKLMNOP',2) |
+-----+
| OP                           |
+-----+
```

+1 row in set (0.00 sec)

(Extracting 2 characters)

- 3) mysql> SELECT RIGHT('ABCDEFGHJKLMNOP',3);

```
+-----+
| RIGHT('ABCDEFGHJKLMNOP',3) |
+-----+
| NOP                           |
+-----+
```

+1 row in set (0.00 sec)

- 4) mysql> SELECT RIGHT('ABCDEFGHJKLMNOP',4);

```
+-----+
| RIGHT('ABCDEFGHJKLMNOP',4) |
+-----+
| MNOP                           |
+-----+
```

+1 row in set (0.00 sec)

- 5) mysql> SELECT RIGHT('ABCDEFGHJKLMNOP',-1);

```
+-----+
| RIGHT('ABCDEFGHJKLMNOP',-1) |
+-----+
```

- **INSTR()** : Checks whether the second string/text is present in the first string. If present it returns the starting index. Otherwise returns 0.

Examples:

- 1) mysql> SELECT INSTR('ABCDEFGHJKLMNOP','ABC');

```
+-----+
| INSTR('ABCDEFGHJKLMNOP','ABC') |
+-----+
| 1                               |
+-----+
```

+1 row in set (0.00 sec)

2) mysql> SELECT INSTR('ABCDEFGHJKLMNOP','BC');

```
+-----+
| INSTR('ABCDEFGHJKLMNOP','BC') |
+-----+
|                2                |
+-----+
```

+1 row in set (0.00 sec)

3) mysql> SELECT INSTR('ABCDEFGHJKLMNOP','EFG');

```
+-----+
| INSTR('ABCDEFGHJKLMNOP','EFG') |
+-----+
|                5                |
+-----+
```

+1 row in set (0.00 sec)

4) mysql> SELECT INSTR('ABCDEFGHJKLMNOP','QRST');

```
+-----+
| INSTR('ABCDEFGHJKLMNOP','QRST') |
+-----+
|                0                |
+-----+
```

+1 row in set (0.00 sec)

- **LTRIM()** : To trim the spaces, if any, from the beginning of the text.

Examples:

1) mysql> SELECT LTRIM(' HELLO');

```
+-----+
| LTRIM(' HELLO') |
+-----+
| HELLO           |
+-----+
```

- **RTRIM()** : To trim the spaces, if any, from the end of the text.

Examples:

1) mysql> SELECT RTRIM('HELLO ');

```
+-----+
| RTRIM('HELLO ') |
+-----+
| HELLO           |
+-----+
```

+1 row in set (0.00 sec)

2) mysql> SELECT CONCAT(RTRIM('HELLO'), 'WORLD');

```
+-----+
| CONCAT(RTRIM('HELLO '), 'WORLD') |
+-----+
| HELLOWORLD                        |
+-----+
```

1 row in set (0.00 sec)

- **TRIM()** : To trim the spaces, if any, from the beginning and end of the text.

Examples:

1) mysql> SELECT CONCAT(TRIM('HELLO'), 'WORLD');

```
+-----+
| CONCAT(TRIM('HELLO '), 'WORLD') |
+-----+
| HELLOWORLD                       |
+-----+1 row in set (0.00 sec)
```

Note: CONCAT() combines two strings/texts

2) mysql> SELECT TRIM(' HELLO ');

```
+-----+
| TRIM(' HELLO ') |
+-----+
| HELLO            |
+-----+
1 row in set (0.00 sec)
```

Try yourself:

Give the output of the following:

1. SELECT POWER(3,3);
2. SELECT POW(3,2);
3. SELECT ROUND(123.45,1);
4. SELECT ROUND(123.45,-1);
5. SELECT ROUND(123.45,0);
6. SELECT ROUND(153.45,2);
7. SELECT ROUND(155.45,0);
8. SELECT ROUND(245,-2);
9. SELECT ROUND(255,- 2);
10. SELECT ROUND(897,3);
11. SELECT ROUND(457,-3);
12. SELECT
 ROUND(1567,-3);
13. SELECT RIGHT('MORNING', 2);
14. SELECT MID(TRIM('GOOD '), 1,
4);
15. SELECT INSTR('GOOD MORNING' ,
'GOOD');

Answers

1) 27, 2)9, 3)123.5 4)120, 5)123 6)200, 7)155, 8)200, 9)300, 10)1000, 11)0, 12)2000,
13)NG , 14)GOOD, 15)1

	Date Functions
1	NOW():- Function that returns the current date and time. <pre>mysql> select now(); +-----+ now() +-----+ 2022-09-04 12:21:25 +-----+ 1 row in set (0.08 sec)</pre>
2	DATE():- This function returns the date part from Date and time. <pre>mysql> select date(now()); +-----+ date(now()) +-----+ 2022-09-04 +-----+ 1 row in set (0.35 sec)</pre>
3	MONTH() returns the month part of a date. <pre>mysql> select month(now()); +-----+ month(now()) +-----+ 9 +-----+ 1 row in set (0.08 sec)</pre>
4	MONTHNAME() :-gives month name from a date. <pre>mysql> select monthname(now()); +-----+ monthname(now()) +-----+ September +-----+</pre>
5	YEAR():- Date function returns year part of a date. <pre>mysql> select year(now()); +-----+ year(now()) +-----+</pre>
6	DAY():- Day function provide the day part of Date. <pre>mysql> select day('2022-04-23'); +-----+ day('2022-04-23') +-----+</pre>

7	DAYNAME():- Returns the weekday name of a date. <pre>mysql> select dayname('2022/08/23'); +-----+ dayname('2022/08/23') +-----+</pre>
	Aggregate functions:- It perform calculation on multiple values and return a single value. It is also know as Multi-row functions
	List of aggregate functions
	MAX(),MIN(),AVG(),SUM(),COUNT() Aggregate function will not consider NULL values for calculation. Consider the following table named student. <pre>+-----+-----+-----+ Sl_no Student_Name Mark +-----+-----+-----+ 1 Aravind 69 2 Geetha 75 3 Sooraj 61 4 Kevin 81 5 Jude NULL +-----+-----+-----+</pre>
1	MAX():- returns maximum value from a dataset. <pre>mysql> select max(mark) from student; +-----+ max(mark) +-----+ 81 +-----+ 1 row in set (0.15 sec)</pre>
2	MIN():- Returns the minimum value from a dataset. <pre>mysql> select min(mark) from student; +-----+ min(mark) +-----+ 61 +-----+ 1 row in set (0.00 sec)</pre>
3	AVG():- Returns the average value from a dataset. <pre>mysql> select avg(mark) from student; +-----+ avg(mark) +-----+ 71.5000 +-----+ 1 row in set (0.04 sec)</pre>

4 SUM():- It returns the sum of values in a dataset.

```
mysql> select sum(mark) from student;
+-----+
| sum(mark) |
+-----+
|      286 |
+-----+
1 row in set (0.00 sec)
```

5 COUNT(*):-Will return the total number of rows present in the Table.

```
mysql> select count(*) from student;
+-----+
| count(*) |
+-----+
|         5 |
+-----+
1 row in set (0.26 sec)
```

6 COUNT(Field Name) will count the number of values(excluding NULL) present in the dataset.

```
mysql> select count(mark) from student;
+-----+
| count(mark) |
+-----+
|           4 |
+-----+
1 row in set (0.00 sec)
```

Here answer 4 is displayed even though 5 rows are present in the student table because one NULL value is present in the column named mark.

Group By clause:- The GROUP BY statement **groups rows that have the same values into summary rows**. The GROUP BY statement is often used with aggregate functions (COUNT() , MAX() , MIN() , SUM() , AVG()) to group the result-set by one or more columns.

Consider the following table named **teacher**.

tid	tname	subject	experience
1	Ganesh	Physics	2
2	Neelesh	History	4
3	Johny	Chemistry	2
4	Devakumar	Chemistry	4
5	Tony	Physics	2

1 Write SQL statement to display number of Teacher teaching **each subject**.

Select count(*) from teacher **group by subject**;

```
+-----+
| count(*) |
+-----+
|         2 |
|         1 |
|         2 |
+-----+
```

	The WHERE clause in MySQL is used with SELECT, INSERT, UPDATE, and DELETE queries to filter rows from the table or relation. It describes a specific condition when retrieving records from tables.
2	Display number of employees in each subject whose experience is more than 2. select count(*) from teacher where experience>2 group by subject; WHERE clause is coming before group by. <pre> +-----+ count(*) +-----+ 1 1 +-----+ 2 rows in set (0.00 sec) </pre>
	Having Clause The HAVING clause is often used with the GROUP BY clause to filter rows based on a specified condition. If we omit the GROUP BY clause, the HAVING clause behaves like the WHERE clause.
1	Display the subject and number of teachers in each subject where group count is more than 1. select subject, count(*) from teacher group by subject having count(*)>1; HAVING clause is coming after group by
2	Display details of teacher whose tid more than 3 select * from teacher having tid>3; OR select * from teacher where tid>3; HAVING clause is behaving like WHERE clause in this example.
	ORDER BY Order by clause is used to arrange the rows in Ascending or Descending order of values in a Column.
1	Display the details of teachers ascending of experience. select * from teacher order by experience; OR select * from teacher order by experience ASC; <pre> +----+-----+-----+-----+ tid tname subject experience +----+-----+-----+-----+ 1 Ganesh Physics 2 3 Johny Chemistry 2 5 Tony Physics 2 2 Neelesh History 4 4 Devakumar Chemistry 4 +----+-----+-----+-----+ </pre>
2	Display the teachers details in the Descending order of experience. select * from teacher order by experience DESC; <pre> +----+-----+-----+-----+ tid tname subject experience +----+-----+-----+-----+ 2 Neelesh History 4 4 Devakumar Chemistry 4 1 Ganesh Physics 2 3 Johny Chemistry 2 5 Tony Physics 2 +----+-----+-----+-----+ </pre>

Questions

- Which of the following would arrange the rows in ascending order in SQL.
a. SORT BY b. ALIGN BY c. GROUP BY d. ORDER BY
- Prachi has given the following command to obtain the highest marks
`Select max(marks) from student where group by class;`
but she is not getting the desired result. Help her by writing the correct command.
a. `Select max(marks) from student where group by class;`
b. `Select class, max(marks) from student group by marks;`
c. `Select class, max(marks) group by class from student;`
d. `Select class, max(marks) from student group by class;`
- Help Ritesh to write the command to display the name of the youngest student?
a. `select name,min(DOB) from student ;`
b. `select name,max(DOB) from student ;`
c. `select name,min(DOB) from student group by name ;`
d. `select name,maximum(DOB) from student;`
- A relation 'Vehicles' is given below :

V_no	Type	Company	Price	Qty
AW125	Wagon	Maruti	250000	25
J0083	Jeep	Mahindra	4000000	15
S9090	SUV	Mitsubishi	2500000	18
M0892	Mini van	Datsun	1500000	26
W9760	SUV	Maruti	2500000	18
R2409	Mini van	Mahindra	350000	15

Write SQL commands to:

- Display the average price of each type of vehicle having quantity more than 20.
- Count the type of vehicles manufactured by each company.
- Display the total price of all the types of vehicles.

5. Consider the table 'FANS' and answer the following.

FAN_ID	FAN_NAME	FAN_CITY	FAN_DOB	FAN_MODE
F001	SUSHANT	MUMBAI	1998-10-02	MAIL
F002	RIYA	MUMBAI	1997-12-12	LETTER
F003	ANIKA	DELHI	2001-06-30	BLOG
F004	RUDRA	AJMER	2005-08-22	MAIL
F006	MIARA	KOLKATTA	1998-11-01	BLOG

Write MySQL queries for the following:

- To display the details of fans in descending order of their DOB
- To count the total number of fans of each fan mode
- iv. To display the dob of the youngest fan.

6. Write commands in SQL for (i) and (ii) and output for (iii)

Table : Store

StoreId	Name	Location	City	NoOfEmp	DateOpen	SalesAmt
S101	Planet Fashion	Bandra	Mumbai	7	2015-10-16	40000
S102	Vogue	Karol Bagh	Delhi	8	2015-07-14	120000
S103	Trends	Powai	Mumbai	10	2015-06-24	30000
S104	Super Fashion	Thane	Mumbai	11	2015-02-06	45000
S105	Annabelle	South Extn.	Delhi	8	2015-04-09	60000
S106	Rage	Defence Colony	Delhi	5	2015-03-01	20000

- (i) To display the details of the store in alphabetical order of name.
- (ii) To display the City and the number of stores located in that City, only if the number of stores is more than 2.
- (iii) SELECT COUNT(STOREID), NOOFEMP FROM STORE GROUP BY NOOFEMP HAVING MAX(SALESAMT)<60000;

Answers

1. d
2. d
3. b
- 4

- a. select Type, avg(Price) from Vehicle group by Type having Qty>20;
- b. select Company, count(distinct Type) from Vehicle group by Company;
- c. Select Type, sum(Price* Qty) from Vehicle group by Type;

5.

- i. SELECT * FROM FANS ORDER BY FAN_DOB DESC;
- ii. SELECT FAN_MODE, COUNT(*) FROM FANS GROUP BY FAN_MODE;
- iii. SELECT MAX(FAN_DOB) FROM FANS;

6.

- (i) SELECT * FROM STORE ORDER BY NAME;
- (ii) SELECT CITY, COUNT(*) FROM STORE GROUP BY STORE HAVING COUNT(*)>2;
- (iii) Count(StoreId) | NoOfEmp |

+	+	+
1	10	
1	11	
1	5	
1	7	

Worksheet 1

A. Give output of the following:

1. Select Round(546.345, -2);
2. Select Round(546.345, -3);
3. Select Truncate(124.56, 1);
4. Select Truncate(124.56, -2);
5. Select SIGN(-341.45);
6. Select SIGN(0);
7. Select MOD(34,5);
8. Select MOD(6,8);
9. Select MOD(12.6, 8);
10. Select Pow(2,4);
11. Select Char(78,65,77,65,78);
12. Select Char(76.5,65.3,'77.6',65,78);
13. Select Substr('JS09876/XII-H/12',-8,7);
14. Select Trim(Leading 'Pp' from 'PppProgram Features');
15. Select Instr('COORDINATION COMMITTEE ORDER','OR');
16. Select left('COORDINATION COMMITTEE ORDER', length('committee'));
17. Select right('COORDINATION COMMITTEE ORDER', length('committee order'));

B. Write MySQL statements for the following:

- I. Display first letter of the provided string in CAPITAL letter and rest all in small letters.
- II. Display the phone number 07762227042 in the format (07762)-22-7042.

Worksheet 2

1. select Char(70,65,67,69);
2. select Char(65, 67.3, '69.3');
3. SELECT CONCAT(NAME,dept) AS 'NAME DEPT' FROM Employee;
4. SELECT LOWER(NAME) FROM Employee
5. select substr('ABCDEFGH', 3,4);
6. select substr('ABCDEFGH', -3,4);
7. SELECT UCASE('pqrxyz');
8. SELECT RTRIM('abcdefgh ');
9. select trim(' pqr Periodic Test 1 is over pqr ');
10. select instr('O P JINDAL SCHOOL, KHARSIA ROAD RAIGARH, JINDAL STEEL AND POWER LIMITED','JINDAL');
11. SELECT LENGTH('O P JINDAL SCHOOL, RAIGARH');
12. SELECT LENGTH(12345);
13. select left('JS12345/OPJS/XII/A',7);
14. select substr('JS12345/OPJS/XII/A',9,4);
15. select right('JS12345/OPJS/XII/A',5);

16. select MOD(23,5);
17. select power(3,4);
18. select ROUND(20.392,1);
19. select sign(-945);
20. select sqrt(25);
21. select truncate(129.345,1);

Worksheet 3

1. select pow(2,3), power(-2,3), pow(3,4);
2. select round(12345.789,2), round(1434.56,-1);
3. select round(62.789),round(6.89,0);
4. select truncate(466.789,1), truncate(645.56,-1);
5. select sqrt(81)+20 from dual;
6. select mod(23,2), mod(78,4);
7. select sign(15,-15), sign(-25), sign(70);
8. select char(65,67,69) from dual;
9. select concat("info","rmatics");
10. select concat("ISM -",concat("xii","I"));
11. select lower("INFORM"),lcase("Class XII");
12. select upper("Class xii"),ucase("informatics");
13. select substring("India is the Best",3,2),substr("Indian",-2,1);
14. select length(trim(" abcde defe "));
15. select instr("Informatics","r");
16. select length("ab cde fge");
17. select left("Informatics",4) from dual;
18. select right("Informatics",6);
19. select mid("Indian School Muscat",8,6);
20. Select curdate(), current_date();
21. Select date(now());
22. Select month(now());
23. Select year("2012-02-21");
24. Select dayname(now());
25. Select dayofmonth("2011-03-23");
26. Select dayofweek(now());
27. Select dayofyear("2016-02-04");
28. Select dayofyear("2012-02-02");
29. Select NOW(),SLEEP(3),SYSDATE();