

XII - IP
SLIP TEST ON MySQL

Time : 50 Mts

Max.Marks : 25

I. Write outputs of the following : (26 X ½=13 M)

(a) SELECT CHAR(65.7,98,69);

```
+-----+
| CHAR(65.7,98,69) |
+-----+
| BbE               |
+-----+
```

(b) SELECT UPPER("Welcome to WBB");

```
mysql> SELECT UPPER("Welcome to WBB");
+-----+
| UPPER("Welcome to WBB") |
+-----+
| WELCOME TO WBB          |
+-----+
```

(c) SELECT SUBSTR('VIJAYAWADA',-8,5);

```
mysql> SELECT SUBSTR('VIJAYAWADA',-8,5);
+-----+
| SUBSTR('VIJAYAWADA',-8,5) |
+-----+
| JAYAW                     |
+-----+
```

(d) SELECT RTRIM(' Nidadavolu ');

```
mysql> SELECT RTRIM(' Nidadavolu ');
+-----+
| RTRIM(' Nidadavolu ') |
+-----+
| Nidadavolu             |
+-----+
```

(e) SELECT TRIM(LEADING 'K' FROM 'KKKHyderaKKKbadKKKK');

```
mysql> SELECT TRIM(LEADING 'K'
-> FROM 'KKKHyderaKKKbadKKKK');
+-----+
| TRIM(LEADING 'K'
FROM 'KKKHyderaKKKbadKKKK') |
+-----+
| HydraKKKbadKKKK          |
+-----+
```

(f) SELECT INSTR('MY STRING','TR');

```
mysql> SELECT INSTR('MY STRING','TR');
+-----+
| INSTR('MY STRING','TR') |
+-----+
| 5                         |
+-----+
```

(g) SELECT LENGTH('LieuTEnant');

```
mysql> SELECT LENGTH('LieuTEnant');
+-----+
| LENGTH('LieuTEnant') |
+-----+
| 10                    |
+-----+
```

(h) SELECT MID("NAGAsaki",3,4);

```
mysql> SELECT MID("NAGAsaki",3,4);
+-----+
| MID("NAGAsaki",3,4) |
+-----+
| GAsa                 |
+-----+
```

(i) SELECT 30 MOD 11;

```
mysql> SELECT 30 MOD 11;
+-----+
| 30 MOD 11 |
+-----+
| 8          |
+-----+
```

(j) SELECT POWER(1,10);

```
mysql> SELECT POWER(1,10);
+-----+
| POWER(1,10) |
+-----+
| 1            |
+-----+
```

(k) SELECT ROUND(836.2345,-2);

```
mysql> SELECT ROUND(836.2345,-2);
+-----+
| ROUND(836.2345,-2) |
+-----+
| 800                 |
+-----+
```

(l) SELECT ROUND(876.2375,2);

```
mysql> SELECT ROUND(876.2375,2);
+-----+
| ROUND(876.2375,2) |
+-----+
| 876.24             |
+-----+
```

(m) SELECT ROUND(876.7345,0);

```
mysql> SELECT ROUND(876.7345,0);
+-----+
| ROUND(876.7345,0) |
+-----+
| 877                |
+-----+
```

(n) SELECT SIGN(754);

```
mysql> SELECT SIGN(754);
+-----+
| SIGN(754) |
+-----+
| 1          |
+-----+
```

(o) SELECT SQRT(15-29%15);

```
mysql> SELECT SQRT(15-29%15);
+-----+
| SQRT(15-29%15) |
+-----+
|                1 |
+-----+
```

(p) SELECT TRUNCATE(7373212.12854,0);

```
mysql> SELECT TRUNCATE(7373212.12854,0);
+-----+
| TRUNCATE(7373212.12854,0) |
+-----+
|                7373212 |
+-----+
```

(q) SELECT TRUNCATE(7373212.12854,2);

```
mysql> SELECT TRUNCATE(7373212.12854,2);
+-----+
| TRUNCATE(7373212.12854,2) |
+-----+
|                7373212.12 |
+-----+
```

(r) SELECT TRUNCATE(7373212.12854,-2);

```
mysql> SELECT TRUNCATE(7373212.12854,-2);
+-----+
| TRUNCATE(7373212.12854,-2) |
+-----+
|                7373200 |
+-----+
```

(s) SELECT CURDATE() + 05; (24.09.2022)

```
mysql> SELECT CURDATE() + 05;
+-----+
| CURDATE() + 05 |
+-----+
|        20220929 |
+-----+
```

(t) SELECT DATE('2022-12-25 Welcome');

```
mysql> SELECT DATE('2022-12-25 Welcome');
+-----+
| DATE('2022-12-25 Welcome') |
+-----+
|        2022-12-25 |
+-----+
```

(u) SELECT YEAR('1975-05-22');

```
mysql> SELECT YEAR('1975-05-22');
+-----+
| YEAR('1975-05-22') |
+-----+
|                1975 |
+-----+
```

(v) SELECT MONTHNAME('1981-11-02');

```
mysql> SELECT MONTHNAME('1981-11-02');
+-----+
| MONTHNAME('1981-11-02') |
+-----+
|        November |
+-----+
```

(w) SELECT DAY('2015-06-26 WBB');

```
mysql> SELECT DAY('2015-06-26 WBB');
+-----+
| DAY('2015-06-26 WBB') |
+-----+
|                26 |
+-----+
```

(x) SELECT DAY('WBB 2015-06-15');

```
mysql> SELECT DAY('WBB 2015-06-15');
+-----+
| DAY('WBB 2015-06-15') |
+-----+
|                NULL |
+-----+
```

(y) SELECT DAYNAME('2022-09-25');

```
mysql> SELECT DAYNAME('2022-09-25');
+-----+
| DAYNAME('2022-09-25') |
+-----+
|        Sunday |
+-----+
```

(z) SELECT DAYOFWEEK('2022-09-23');

```
mysql> SELECT DAYOFWEEK('2022-09-23');
+-----+
| DAYOFWEEK('2022-09-23') |
+-----+
|                6 |
+-----+
```

2. Harshit, is learning SQL and confused with the output of following queries on table , predict the output for him: (2)

TABLE : LOANS

ID	CUS_NAME	LOAN_AMT	INSTALLMENT	INTEREST RATE	STARTDATE	AGE
C1	SAMEER	300000	36	12.00	2019-07-19	36
C2	ARYAN	500000	60	10.00	2018-03-22	65
C4	RAM	800000	48	NULL	2018-03-08	48
C6	PRERNA	300000	24	10.00	2020-12-06	54
C7	SHIKHA	900000	36	12.50	2020-01-03	42
C8	RADHA	1000000	60	NULL	2017-07-29	62

He has written the following queries:

(i) select (year(curdate)-year(StartDate))*12 as Installments_over from loans

A) (year(curdate)-year(StartDate))*12

36
48
48
24
24
60

(ii) select CUS_NAME,monthname(Startdate) from LOANS where 60-age<=0;

A) CUS_NAME monthname(Startdate)
ARYAN MARCH
RADHA JULY

Based on the table given above, help Harshit, write queries for the following:

(iii) To display the earliest loan start date.

A) SELECT MIN(STARTDATE) FROM LOANS;

(iv) To display the names and loan amount of those customers whose loan started in 'March'.

A) SELECT CUS_NAME, LOAN_AMT FROM LOANS WHERE MONTHNAME(STARTDATE)='MARCH';

3. Rishita is working on a MySQL table named 'PRODUCTS' having following structure: (2)

Field	Type
pid	Varchar(5)
Pname	Varchar(20)
descp	Varchar(30)
qty	integer
Price	float

i) To fetch last 3 characters of pid.

A) SELECT SUBSTR(Pid,-3,3) FROM PRODUCTS;

ii) To display the product name in upper case, removing all trailing and preceding spaces.

A) SELECT UCASE(TRIM(Pname));

4. A Coaching Institute STAR COACHING ACADEMY has maintained the record of their students using SQL table "ENGG". (4)

ID	NAME	AGE	CITY	SUB	FEE	MODE
P1	SAMEER	20	DELHI	ELECTR	45000	OFFLINE
P2	ARYAN	21	BHOPAL	COMP	54000	ONLINE
P4	RAM	18	CHENNAI	COMP	45000	OFFLINE
P6	LATA	20	BHOPAL	MECH	60000	OFFLINE
P7	SHIKHA	18	INDORE	ELECTR	34000	ONLINE
P8	RADHA	19	DELHI	AERO	23000	ONLINE
P10	ARMAAN	18	BHOPAL	MECH	34000	OFFLINE

Write SQL queries for the following:

a) To display the city, sum of fees obtained from students city-wise in decreasing order of city.

A) SELECT CITY, SUM(FEE) FROM ENGG GROUP BY CITY DESC;

```
mysql> SELECT CITY, SUM(FEE)
-> FROM ENGG GROUP BY CITY DESC;
```

CITY	SUM(FEE)
INDORE	34000
DELHI	68000
CHENNAI	45000
BHOPAL	148000

b) To display the mode and count of students enrolled in each mode.

A) SELECT MODE, COUNT FROM ENGG GROUP BY MODE;

```
mysql> SELECT MODE, COUNT(*) FROM ENGG
-> GROUP BY MODE;
```

MODE	COUNT(*)
OFFLINE	4
ONLINE	3

c) To display the sub, average fees from students subject-wise (for those subjects having more than one student)

A) SELECT SUB, AVG(FEE) FROM ENGG GROUP BY SUB HAVING COUNT(*)>1;

```
mysql> SELECT SUB, AVG(FEE) FROM ENGG
-> GROUP BY SUB HAVING COUNT(*)>1;
```

SUB	AVG(FEE)
COMP	49500.0000
ELECTR	39500.0000
MECH	47000.0000

d) To display the maximum and minimum fees for each mode of classes.

A) SELECT MODE, MAX(FEE), MIN(FEE) FROM ENGG GROUP BY MODE;

```
mysql> SELECT MODE, MAX(FEE), MIN(FEE)
-> FROM ENGG GROUP BY MODE;
```

MODE	MAX(FEE)	MIN(FEE)
OFFLINE	60000	34000
ONLINE	54000	23000

5. Sahil has recently joined a mobile shop that maintains the database of all mobile stock in the form of relation MobileMaster:

M_ID	M_Company	M_name	M_Price	QTY	Mfg_date
MB001	SAMSUNG	GALAXY	5400	13	2013-02-12
MB003	NOKIA	N1100	1200	NULL	2007-06-24
MB004	REDMI	NOTE 7	13000	10	2019-03-20
MB005	SONY	XPERIAM	45000	6	2017-11-10
MB006	OPPO	SELFIEEX	17000	7	2010-08-01
MB007	REDMI	NOTE11	25000	15	2021-11-21
MB008	ONE PLUS	NORD	30000	12	2020-02-15
MB010	ONE PLUS	ONEPLUS 9	35000	11	2021-12-25

Help him frame the SQL statements for the queries below: (2)

(a) To display the mobile company, mobile name and price in descending order of their price.

A) SELECT M_Company, M_name, M_Price FROM MobileMaster ORDER BY M_Price DESC;

```
mysql> SELECT M_Company,M_name,M_Price
-> FROM MobileMaster ORDER BY M_Price DESC;
```

M_Company	M_name	M_Price
SONY	XPERIAM	45000
ONE PLUS	ONEPLUS9	35000
ONE PLUS	NORD	30000
REDMI	NOTE11	25000
OPPO	SELFIEEX	17000
REDMI	NOTE 7	13000
SAMSUNG	GALAXY	5400
NOKIA	N1100	1200

```
mysql> SELECT M_name FROM MobileMaster
-> WHERE QTY IS NOT NULL;
```

M_name
GALAXY
NOTE 7
XPERIAM
SELFIEEX
NOTE11
NORD
ONEPLUS9

(b) To display the mobile name and quantity of all mobiles except 'MB003'

A) SELECT M_name,QTY FROM MobileMaster WHERE M_ID <>'MB003';

```
mysql> SELECT M_name,QTY FROM MobileMaster
-> WHERE M_ID <>'MB003';
```

M_name	QTY
GALAXY	13
NOTE 7	10
XPERIAM	6
SELFIEEX	7
NOTE11	15
NORD	12
ONEPLUS9	11

SELECT M_name,QTY FROM MobileMaster WHERE M_ID !='MB003';

OR

SELECT M_name,QTY FROM MobileMaster WHERE M_ID NOT IN ('MB003');

(c) To display the mobile details manufactured after the year 2019

A) SELECT * FROM MobileMaster WHERE Year(Mfg_date)>2019;

```
mysql> SELECT * FROM MobileMaster
-> WHERE Year(Mfg_date)>2019;
```

M_ID	M_Company	M_name	M_Price	QTY	Mfg_date
MB007	REDMI	NOTE11	25000	15	2021-11-21
MB008	ONE PLUS	NORD	30000	12	2020-02-15
MB010	ONE PLUS	ONEPLUS9	35000	11	2021-12-25

(d) To display the mobile names whose quantity is not null.

A) SELECT M_name FROM MobileMaster WHERE QTY IS NOT NULL;

Predict the output for the following queries for Sahil:(2)

(a) Select M_Company, avg(qty) From MobileMaster Group by M_Company

M_Company	avg(qty)
NOKIA	NULL
ONE PLUS	11.5000
OPPO	7.0000
REDMI	12.5000
SAMSUNG	13.0000
SONY	6.0000

(b) Select substr(M_Company,1,3),M_Price From MobileMaster Where Qty>10;

Substr(M_Company,1,3)	M_Price
SAM	5400
RED	25000
ONE	30000
ONE	35000

(c) Select M_Name ,M_Price*.01 as discount From MobileMaster Where M_Price between 25000 and 35000;

M_Name	discount
NOTE11	250.00
NORD	300.00
ONEPLUS9	350.00

(d) select count(*), count(qty), max(Mfg_date) From MobileMaster;

count(*)	count(qty)	max(Mfg_date)
8	7	2021-12-25