SQL 1

Q) Create table student with fields roll_no, name, gender and mark with the roll_no as primary key and assign suitable constraints (like check and not null) for each attributes.

```
CREATE TABLE student
(
    roll_no int PRIMARY KEY,
    name char(20) NOT NULLl,
    gender char(10) NOT NULL,
    mark int check(mark<=100)
);
```

a) Insert 5 records.

```
INSERT INTO student VALUES(101,'Safwan','male',95);
INSERT INTO student VALUES(102,'Faris','male',98);
INSERT INTO student VALUES(103,'Aysha','female',90);
INSERT INTO student VALUES(104,'Nandana','female',65);
INSERT INTO student VALUES(105,'Nashir','male',80);
SELECT * FROM student;
```

```
 roll_no | name | gender | mark
---------+---------+--------+------
 101 | Safwan | male | 95
 102 | Faris | male | 98
 103 | Aysha | female | 90
 104 | Nandana | female | 65
 105 | Nashir | male | 80
```

b) Display all boy students with their name.

```
SELECT name FROM student WHERE gender='male';
 name
--------
 Faris
 Safwan
 Nashir
```

c) Find the average mark.

```
SELECT AVG(mark) FROM student;
avg
---------------------
85.6000000000000000
```

d) Display the roll no., name and mark of student who got highest mark.

```
SELECT roll_no,name,mark FROM student WHERE mark=(select max(mark) FROM student);
```

```
roll_no | name | mark
---------+-------+------
102 | Faris | 98
```

e) Alter the table by adding one more field place.

```
ALTER TABLE student ADD place char(20);
SELECT * FROM student;
```

```
roll_no | name | gender | mark | place
---------+---------+--------+------+-------
101 | Safwan | male | 95 |
102 | Faris | male | 98 |
103 | Aysha | female | 90 |
104 | Nandana | female | 65 |
105 | Nashir | male | 80 |
```

f) Update the field place.

```
UPDATE student SET place='taliparamba' WHERE roll_no=101;
UPDATE student SET place='kannur' WHERE roll_no=102;
UPDATE student SET place='kasargod' WHERE roll_no=103;
UPDATE student SET place='kozhikode' WHERE roll_no=104;
UPDATE student SET place='wayanad' WHERE roll_no=105;
SELECT * FROM student ;
```

```
roll_no | name | gender | mark | place
---------+---------+--------+------+----------------------
101 | Safwan | male | 95 | taliparamba
102 | Faris | male | 98 | kannur
103 | Aysha | female | 90 | kasargod
104 | Nandana | female | 65 | kozhikode
105 | Nashir | male | 80 | wayanad
```

g) Display the name and place of all girl students who have marks greater than 35 and less than 70.

```
SELECT name,place FROM student WHERE gender='female' AND mark BETWEEN 80 AND 89 ;
```

```
name | place
---------+----------------------
Nandana | kozhikode
```

SQL 2

Q) Create table department with fields dpt_id as primary key and d_name as not null. Create another table employee with fields emp_id , e_name, salary, dpt_id and dob. Assign constraints for emp_id as as primary key and auto increment, dpt_id as foreign key, e_name, salary and dob as not null.

```
CREATE TABLE department
(
   dpt_id int PRIMARY KEY ,
   d_name char(20) NOT NULL
);

CREATE TABLE employee
(
   emp_id int PRIMARY KEY AUTO_INCREMENT ,
   e_name char(20),
   salary int NOT NULL,
   dpt_id int,
   dob  date NOT NULL,
   foreign key (dpt_id) references department(dpt_id)
);
```

a) Insert 5 records into both tables.

```
INSERT INTO department VALUES (1001,'administration');
INSERT INTO department VALUES (1002,'marketing');
INSERT INTO department VALUES (1003,'purchasing');
INSERT INTO department VALUES (1004,'it');
INSERT INTO department VALUES (1005,'sales');
SELECT * from department;
```

```
+--------+----------------+
| dpt_id | d_name         |
+--------+----------------+
|   1001 | administration |
|   1002 | marketing      |
|   1003 | purchasing     |
|   1004 | it             |
|   1005 | sales          |
+--------+----------------+
(5 rows)
```

```
INSERT INTO employee(e_name,salary,dpt_id,dob)VALUES('shahid ',100000,1003,'2000-02-04');
INSERT INTO employee(e_name,salary,dpt_id,dob)VALUES('sinan',80000,1002,'1998-07-07');
INSERT INTO employee(e_name,salary,dpt_id,dob) VALUES('shinadh',70000,1001,'1999-06-11');
INSERT INTO employee(e_name,salary,dpt_id,dob) VALUES('fathima',60000,1004,'1997-12-23');
INSERT INTO employee(e_name,salary,dpt_id,dob)VALUES('Salman',50000,1005,'1997-12-21');
SELECT * from employee;
```

```
+--------+---------+--------+--------+------------+
| emp_id | e_name  | salary | dpt_id | dob        |
+--------+---------+--------+--------+------------+
|      1 | shahid  | 100000 |   1003 | 2000-02-04 |
|      2 | sinan   |  80000 |   1002 | 1998-07-07 |
|      3 | shinadh |  70000 |   1001 | 1999-06-11 |
|      4 | fathima |  60000 |   1004 | 1997-12-23 |
|      5 | Salman  |  50000 |   1005 | 1997-12-21 |
+--------+---------+--------+--------+------------+
```

(5 rows)

b) Display the employees who got salary more than 60000 and less than 1 lakh.

```
SELECT * FROM employee
WHERE salary
BETWEEN 60000 AND 100000;
```

```
+--------+---------+--------+--------+------------+
| emp_id | e_name  | salary | dpt_id | dob        |
+--------+---------+--------+--------+------------+
|      1 | shahid  | 100000 |   1003 | 2000-02-04 |
|      2 | sinan   |  80000 |   1002 | 1998-07-07 |
|      3 | shinadh |  70000 |   1001 | 1999-06-11 |
|      4 | fathima |  60000 |   1004 | 1997-12-23 |
+--------+---------+--------+--------+------------+
(2 rows)
```

c) Display the d_name, e_name and salary of employees who get salary more than 50000.

```
SELECT DISTINCT
    department.d_name ,
    employee.e_name,
    employee.salary
FROM department,employee
WHERE salary>50000 and department.dept_id=employee.dpt_id;
```

```
+----------------+---------+--------+
| d_name         | e_name  | salary |
+----------------+---------+--------+
| purchasing     | shahid  | 100000 |
| marketing      | sinan   |  80000 |
| administration | shinadh |  70000 |
| it             | fathima |  60000 |
+----------------+---------+--------+
```

d) Rename the field e_name with emp_name.

```
ALTER table employee RENAME column e_name to emp_name;
SELECT * from employee;
```

```
+--------+----------+--------+--------+------------+
| emp_id | emp_name | salary | dpt_id | dob        |
+--------+----------+--------+--------+------------+
|      1 | shahid   | 100000 |   1003 | 2000-02-04 |
|      2 | sinan    |  80000 |   1002 | 1998-07-07 |
|      3 | shinadh  |  70000 |   1001 | 1999-06-11 |
|      4 | fathima  |  60000 |   1004 | 1997-12-23 |
|      5 | Salman   |  50000 |   1005 | 1997-12-21 |
```

e) Create a view name emp_view with fields emp_id, emp_name and dob, display the view.

```
CREATE VIEW emp_view as
SELECT emp_id,emp_name,dob FROM employee;

SELECT * from emp_view;
```

```
+--------+----------+------------+
| emp_id | emp_name | dob        |
+--------+----------+------------+
|      1 | shahid   | 2000-02-04 |
|      2 | sinan    | 1998-07-07 |
|      3 | shinadh  | 1999-06-11 |
|      4 | fathima  | 1997-12-23 |
|      5 | Salman   | 1997-12-21 |
```

f) Display emp_id and salary of all employees in descending order of their salary.

```
SELECT emp_id,salary
FROM employee
ORDER BY salary DESC;
```

```
| emp_id | salary |
+--------+--------+
|      1 | 100000 |
|      2 |  80000 |
|      3 |  70000 |
|      4 |  60000 |
|      5 |  50000 |
```

g) Display the name of department with number of employees.

```
SELECT d_name,
COUNT(emp_id) as
    "no.of employees"FROM department,employee
WHERE
    department.dpt_id=employee.dpt_id group by d_name;
```

```
| d_name         | no.of employees |
+----------------+-----------------+
| administration |               1 |
| marketing      |               1 |
| purchasing     |               1 |
| it             |               1 |
| sales          |               1 |
```

SQL 3

Q) Create a table depositor with fields acc_no as primary key, depositor_name, branch and balance. Assign suitable constraints for each attributes. Create another table borrower with fields loan_no as primary key, acc_no as foreign key and amount as not null.

```
CREATE TABLE depositor
(
   acc_no int PRIMARY KEY,
   depositor_name char(20),
   branch char(15) NOT NULL,
   balance int NOT NULL
);

CREATE TABLE borrower
(
   loan_no int PRIMARY KEY,
   acc_no int,
   FOREIGN KEY(acc_no) REFERENCES depositor(acc_no),amount int NOT NULL
);
```

a) Insert five records into both tables.

```
INSERT INTO depositor VALUES(101,'abusafwan','taliparamb',12000);
INSERT INTO depositor VALUES(102,'nadir','kattampally',15000);
INSERT INTO depositor VALUES(103,'amras','chalad',20000);
INSERT INTO depositor VALUES(104,'khalid','pappinisseri',10000);
INSERT INTO depositor VALUES(105,'mizhab','valapattanam',18000);
SELECT * FROM depositor;
```

```
| acc_no | depositor_name | branch       | balance |
+--------+----------------+--------------+---------+
|    101 | abusafwan      | taliparamb   |   12000 |
|    102 | nadir          | kattampally  |   15000 |
|    103 | amras          | chalad       |   20000 |
|    104 | khalid         | pappinisseri |   10000 |
|    105 | mizhab         | valapattanam |   18000 |
+--------+----------------+--------------+---------+
```

```
INSERT INTO borrower VALUES(1001,102,50000);
INSERT INTO borrower VALUES(1002,102,25000);
INSERT INTO borrower VALUES(1003,103,35000);
INSERT INTO borrower VALUES(1004,104,26000);
INSERT INTO borrower VALUES(1005,105,80000);

 SELECT * FROM borrower;
```

```
| loan_no | acc_no | amount |
+---------+--------+--------+
|    1001 |    102 |  50000 |
|    1002 |    102 |  25000 |
|    1003 |    103 |  35000 |
|    1004 |    104 |  26000 |
|    1005 |    105 |  80000 |
```

b) Write the queries using various group functions on amount field.

```
SELECT COUNT(amount) FROM borrower;
```

| COUNT(amount) |
+---------------+
| 5 |
+---------------+

SELECT MAX(amount) FROM borrower;

| MAX(amount) |
+-------------+
| 80000 |

SELECT MIN(amount) FROM borrower;

| MIN(amount) |
+-------------+
| 25000 |

SELECT SUM(amount) FROM borrower;

| SUM(amount) |
+-------------+
| 216000 |

SELECT AVG(amount) FROM borrower;

| AVG(amount) |
+-------------+
| 43200.0000 |
+-------------+

c) Display the count of depositors according to their branch.

SELECT branch,
COUNT(depositor_name)
FROM depositor
GROUP BY branch;

| branch | COUNT(depositor_name) |
+--------------+----------------------+
taliparamb	1
kattampally	1
chalad	1
pappinisseri	1
valapattanam	1
+--------------+----------------------+

d) Display the name of customers who have an account but not loan.

SELECT depositor_name
FROM depositor
WHERE acc_no NOT IN(select acc_no FROM borrower);

| depositor_name |
+----------------+
| abusafwan |

e) Drop the column amount from borrower table.

ALTER TABLE borrower DROP COLUMN amount;
SELECT * FROM borrower;

| loan_no | acc_no |
+---------+--------+
1001	102
1002	102
1003	103
1004	104
1005	105
+---------+--------+

SQL 4

Q) Create a table Teacher with fields staff_id,name,d_no,salary and designation with staff_id as primary key, name as not null, dno as foreign key, salary and designation are not null. Create another table Dept with field d_no as primary key, d_name as not null.

```
CREATE TABLE Dept
(
   d_no int PRIMARY KEY,
   d_name char(20) NOT NULL
);


CREATE TABLE Teacher
(
   staff_id int PRIMARY KEY,
   name char(20) NOT NULL,
   d_no int,
   salary int NOT NULL,
   designation char(20) NOT NULL,
   FOREIGN KEY(d_no) REFERENCES Dept(d_no)
);
```

a)Insert 5 records into the table

```
INSERT INTO Dept VALUES(11,'Computer science');
INSERT INTO Dept VALUES(12,'B.com Corporation');
INSERT INTO Dept VALUES(13,'BBA');
INSERT INTO Dept VALUES(15,'BA English');
INSERT INTO Dept VALUES(14,'Physics');
SELECT * FROM Dept;
```

d_no	d_name
11	Computer science
12	B.com Corporation
13	BBA
14	Physics
15	BA English

```
INSERT INTO Teacher VALUES(101,'Safwan',11,7500,'Head of Department');
INSERT INTO Teacher VALUES(102,'Nashir',13,5000,'Assistant professor');
INSERT INTO Teacher VALUES(103,'Mizhab',14,6000,'Professor');
INSERT INTO Teacher VALUES(104,'Ayisha',12,4000,'Assistant professor');
INSERT INTO Teacher VALUES(105,'Shahid',15,5500,'Professor');
SELECT * FROM Teacher;
```

staff_id	name	d_no	salary	designation
101	Safwan	11	7500	Head of Department
102	Nashir	13	5000	Assistant professor
103	Mizhab	14	6000	Professor
104	Ayisha	12	4000	Assistant professor
105	Shahid	15	5500	Professor

b)Write the queries using various character functions on name field.

```
SELECT UPPER(name) FROM Teacher;
```

UPPER(name)
SAFWAN
NASHIR
MIZHAB
AYISHA
SHAHID

```
SELECT LOWER(name) FFROM Teacher;
```

LOWER(name)
safwan
nashir
mizhab
ayisha
shahid

```
SELECT  LENGTH(name) FROM Teacher;
```

LENGTH(name)
6
6

```
|           6 |
|           6 |
|           6 |
```

c)Display the number of staffs in each department.

```
SELECT d_name,
COUNT(staff_id) as
      "no.of staffs" from Dept,Teacher
 WHERE Dept.d_no=Teacher.d_no
GROUP BY d_name;
```

```
| d_name            | no.of staffs |
+-------------------+--------------+
| Computer science  |            1 |
| B.com Corporation |            1 |
| BBA               |            1 |
| Physics           |            1 |
| BA English        |            1 |
```

d)Add 20% extra salary to all employees who works in Physics department.

```
UPDATE Teacher
SET salary=(salary+0.2*salary)
WHERE d_no=
(
          SELECT d_no
          FROM Dept
          WHERE d_name='physics'
);
SELECT * FROM Teacher;
```

```
| staff_id | name   | d_no | salary | designation         |
+----------+--------+------+--------+---------------------+
|      101 | Safwan |   11 |   7500 | Head of Department  |
|      102 | Nashir |   13 |   5000 | Assistant professor |
|      103 | Mizhab |   14 |   7200 | Professor           |
|      104 | Ayisha |   12 |   4000 | Assistant professor |
|      105 | Shahid |   15 |  5500 | Professor           |
```

e)Display the name of teachers who works in CS department.

```
 SELECT  name
 FROM Teacher
 WHERE d_no IN
 (
   SELECT d_no
   FROM Dept
   WHERE d_name='computer science'
);
```

```
| name   |
+--------+
| Safwan |
```

f) Display the name of teacher who got minimum salary?

```
SELECT name
FROM Teacher
WHERE salary=
(
  SELECT MIN(salary)
  FROM Teacher
);
```

```
 | name   |
+--------+
 | Ayisha |
```

g) Display the name of teacher who got minimum salary?

```
    SELECT name
    FROM Teacher
    WHERE salary<
    (
        SELECT AVG(salary)
        FROM Teacher
    );
```

```
    | name   |
   +--------+
   | Nashir |
   | Ayisha |
   | Shahid |
```

h)Create a view named V1 with fields staff_id,name and d_name. Display the view.

```
  CREATE VIEW V1 as
  SELECT Teacher.staff_id,
          Teacher.name,Dept.d_name
  FROM Teacher inner join
          Dept on Teacher.d_no=Dept.d_no
  GROUP BY Teacher.staff_id,Dept.d_name;
  SELECT * FROM V1;
```

```
  | staff_id | name   | d_name           |
 +----------+--------+------------------+
  |      101 | Safwan | Computer science |
  |      103 | Mizhab | Physics          |
```

SQL 5

Q) Create a table customer with fileds cust_id, cust_name, city, gender with cust_id as primary key and assign suitable constraints for each attributes. Create another table order with fields order_id as primary key, cust_id as foreign key, ordered_item and order_date.

```
CREATE TABLE customer
(
   cust_id int PRIMARY KEY,
   cust_name char(20) NOT NULL,
   city char(20) NOT NULL,
   gender char(20) NOT NULL
);

CREATE TABLE orders
(
   order_id int PRIMARY KEY,
   cust_id int,
   ordered_item char(20) NOT NULL,
   order_date date,
   FOREIGN KEY(cust_id) REFERENCES customer(cust_id)
);
```

a) Insert 5 records into the table.

```
INSERT INTO customer VALUES(101,'pinkman','poothappara','male');
INSERT INTO customer VALUES(102,'walter','manna','male');
INSERT INTO customer VALUES(103,'tommy','kannur','male');
INSERT INTO customer VALUES(104,'arthur','padannapalam','male');
INSERT INTO customer VALUES(105,'helene','marakkarkandy','female');
SELECT * FROM customer;
```

```
| cust_id | cust_name | city          | gender |
+---------+-----------+---------------+--------+
|     101 | pinkman   | poothappara   | male   |
|     102 | walter    | manna         | male   |
|     103 | tommy     | kannur        | male   |
|     104 | arthur    | padannapalam  | male   |
|     105 | helene    | marakkarkandy | female |
+---------+-----------+---------------+--------+
(5 rows)
```

```
INSERT INTO orders VALUES(511,101,'scale','2022-10-11');
INSERT INTO orders VALUES(512,102,'pencil','2022-10-19');
INSERT INTO orders VALUES(513,103,'eraser','2022-10-20');
INSERT INTO orders VALUES(514,101,'pen','2022-10-23');
INSERT INTO orders VALUES(515,105,'protractor','2022-10-25');
SELECT * FROM orders;
```

```
| order_id | cust_id | ordered_item | order_date |
+----------+---------+--------------+------------+
|      511 |     101 | scale        | 2022-10-11 |
|      512 |     102 | pencil       | 2022-10-19 |
```

513	103	eraser	2022-10-20
514	101	pen	2022-10-23
515	105	protractor	2022-10-25

b) Display the name of all customers whose city letter starting as 'k' .

```
SELECT cust_name
FROM customer
WHERE city LIKE 'k%';cust_name;
```

| cust_name |
+-----------+
| tommy |

Display the name of all customers whose city letter ending as ‘a’ .

```
SELECT cust_name
FROM customer
WHERE city like '%a';
```

| cust_name |
+-----------+
| pinkman |
| walter |

(2 row)

Display the name of all customers whose cities second letter as ‘a’ .

```
SELECT cust_name FROM customer WHERE city like '_a%';
```

| cust_name |
+-----------+
| walter |
| tommy |
| arthur |
| helene |

c) Display the customer name and order id of a customer with order_id 514

```
SELECT orders.order_id,customer.cust_name
FROM orders inner join customer on
        customer.cust_id=orders.cust_id
GROUP BY orders.order_id,customer.cust_name
HAVING order_id=514;
```

| order_id | cust_name |
| 514 | pinkman |

d) Display the details of customers whose name contains a letter 'e'.

```
SELECT cust_name
FROM customer
WHERE cust_name like '%e%';
```

```
| cust_name |
+-----------+
| walter    |
| helene    |
```

e) Display the name and city of customers with the order date 23/10/2022.

```
SELECT cust_name,city
 FROM customer
 WHERE cust_id=
 (
   SELECT cust_id
   FROM orders
   WHERE order_date='2022-10-23
');
```

```
| cust_name | city        |
+-----------+-------------+
| pinkman   | poothappara |
```

f) Add one more field order_status into order table.

```
ALTER TABLE  orders
ADD column order_status text;
SELECT * FROM orders;
```

```
| order_id | cust_id | ordered_item | order_date | order_status |
+----------+---------+--------------+------------+--------------+
|      511 |     101 | scale        | 2022-10-11 | NULL         |
|      512 |     102 | pencil       | 2022-10-19 | NULL         |
|      513 |     103 | eraser       | 2022-10-20 | NULL         |
|      514 |     101 | pen          | 2022-10-23 | NULL         |
|      515 |     105 | protractor   | 2022-10-25 | NULL         |
```

g) Create view named 'cust' with the details of customers who did not order. Display the view

```
 CREATE VIEW cust as
 SELECT * FROM customer
 WHERE cust_id not IN(select cust_id FROM orders);
```

```
SELECT * FROM cust;
```

```
+---------+-----------+--------------+--------+
| cust_id | cust_name | city         | gender |
+---------+-----------+--------------+--------+
|     104 | arthur    | padannapalam | male   |
```

SQL 6

Q) Create table employee with fields e_no, e_name, gender and salary with the e_no as primary key and assign suitable constraints (like check and not null) for each attributes.

```
CREATE TABLE employee
(
   e_no int PRIMARY KEY,
   e_name char(20) NOT NULL,
   gender char(10) NOT NULL,
   salary int CHECK(salary<=9000)
);
```

a) Insert 5 records.

```
INSERT INTO employee VALUES(101,'Safwan','male',9000);
INSERT INTO employee VALUES(102,'Faris','male',8500);
INSERT INTO employee VALUES(103,'Aysha','female',7500);
INSERT INTO employee VALUES(104,'Nandana','female',6500);
INSERT INTO employee VALUES(105,'Nashir','male',8000);
SELECT * FROM employee;
```

```
| e_no | e_name  | gender | salary |
+------+---------+--------+--------+
|  101 | Safwan  | male   |   9000 |
|  102 | Faris   | male   |   8500 |
|  103 | Aysha   | female |   7500 |
|  104 | Nandana | female |   6500 |
|  105 | Nashir  | male   |   8000 |
```

b) Display all male employees with their name.

```
SELECT e_name FROM student WHERE gender='male';
```

```
| e_name |
+--------+
| Safwan |
| Faris  |
| Nashir |
```

c) Change the data type of e_name char to varchar data type.

```
ALTER TABLE employee MODIFY e_name varchar(20);
```

d) Display the e_no,e_ name and salary of employee who got highest salary.

```
SELECT e_no,e_name,salary
FROM employee
WHERE salary=
```

```
(
   SELECT MAX(salary)
   FROM employee
);
```

```
| e_no | e_name | salary |
+------+--------+--------+
|  101 | Safwan |   9000 |
```

e) Alter the table by delete one column.

```
ALTER TABLE employee DROP COLUMN gender;
SELECT * FROM employee;
```

```
| e_no | e_name  | salary |
+------+---------+--------+
|  101 | Safwan  |   9000 |
|  102 | Faris   |   8500 |
|  103 | Aysha   |   7500 |
|  104 | Nandana |   6500 |
|  105 | Nashir  |   8000 |
```

f) Rename column e_name as emp_name.

```
ALTER TABLE employee
RENAME  COLUMN e_name TO  emp_name;
```

```
| e_no | emp_name | salary |
+------+----------+--------+
|  101 | Safwan   |   9000 |
|  102 | Faris    |   8500 |
|  103 | Aysha    |   7500 |
|  104 | Nandana  |   6500 |
|  105 | Nashir   |   8000 |
```

SQL 7

Q)Create 2 users, and grant (INSERT,SELECT,DELETE) permission to user 1, grant (SELECT) to user2.Revoke the permission DELETE to user 1.

```
sudo su
//password for nasc
nasclab2
mysql -u root

//creation of database
CREATE DATABASE university;
//enter into database
use university

//creation of table student
CREATE TABLE student(name varchar(20), roll int);
```

//creation of first user
CREATE USER 'ksd' IDENTIFIED BY 'ksd';

// creation of second user
CREATE USER 'knr' IDENTIFIED BY 'knr';

//Giving privilegs (insert, delete, select) to user ksd
GRANT INSERT, DELETE, SELECT ON student TO 'ksd';

//Giving previlage select to user ksd
GRANT SELECT ON student TO 'knr';

// exit from root
cntrl+d

// login to ksd user.
mysql -u ksd -p
//entering password for user ksd
ksd

//enter to db university
use university
INSERT INTO student VALUES('Arya',01);
INSERT INTO student VALUES('Jeena',02);
INSERT INTO student VALUES('Amit',03);
SELECT * FROM student;

| name | roll_no |
+-------+---------+
Arya	1
Jeena	2
Amit	3

DELETE FROM student WHERE roll_no = 02;
SELECT * FROM student;

| name | roll_no |
+------+---------+
| Arya | 1 |
| Amit | 3 |

// exit from user ksd
ctrl+D
// enter to user knr
mysql -u knr -p
//entering password for user knr
knr
//using db
use university
INSERT INTO student VALUES('Vipin',04);

ERROR 1142 (42000):INSERT command denied to user 'knr'@'localhost' for table 'student'

SELECT * FROM student;

| name | roll_no |
+------+---------+
| Arya | 1 |
| Amit | 3 |

// exit from knr user
ctrl+D

//entering to root
sudo su
mysql -u root
//using db
use university

//restrict previlage delete to user ksd
REVOKE DELETE ON student FROM 'ksd';

// enter to ksd user
mysql -u ksd -p
//entering password
ksd

//using db
DELETE FROM student WHERE roll=01;

ERROR 1142 (42000): DELETE command denied to user 'kst'@'localhost' for table 'student

SQL 8

Q) Create table library with fields book_id, book_name, author and price with the book_id as primary key ,and book_name as not null.

```
CREATE TABLE library
(
   book_id int PRIMARY KEY,
   book_name char(50) NOT NULL,
   author char(20),
   price int
);
```

a) Insert 5 records.

```
DELIMITER //
CREATE PROCEDURE ins
(
   i int,
   n char(50),
   a char(20),
   p int
) BEGIN INSERT INTO library VALUES(i,n,a,p); END//
```

```
DELIMITER ;
CALL ins(101,'Balabhoomi','Balaguruswami',500);
CALL ins(102,'Labour india','Abraham',1500);
CALL ins(103,'Bagvat gita','Balaguruswami',2000);
CALL ins(104,'Alchemist','Paulo coelho',1000);
CALL ins(105,'Titanic','Di Caprio',2500);
SELECT * FROM library;
```

```
| book_id | book_name    | author        | price |
+---------+--------------+---------------+-------+
|     101 | Balabhoomi   | Balaguruswami |   500 |
|     102 | Labour india | Abraham       |  1500 |
|     103 | Bagvat gita  | Balaguruswami |  2000 |
|     104 | Alchemist    | Paulo coelho  |  1000 |
|     105 | Titanic      | Di Caprio     |  2500 |
```

b)Select the table library using procedure.

```
DELIMITER //
CREATE PROCEDURE sele() BEGIN SELECT * FROM library; END//

DELIMITER ;
CALL sele();
```

```
| book_id | book_name    | author        | price |
+---------+--------------+---------------+-------+
|     101 | Balabhoomi   | Balaguruswami |   500 |
|     102 | Labour india | Abraham       |  1500 |
|     103 | Bagvat gita  | Balaguruswami |  2000 |
|     104 | Alchemist    | Paulo coelho  |  1000 |
|     105 | Titanic      | Di Caprio     |  2500 |
```

c) Select book_name and author from library.

```
DELIMITER //
CRAETE PROCEDURE sel() BEGIN SELECT book_name,author FROM library; END//

DELIMITER ;
CALL sel();
```

```
| book_name    | author        |
+--------------+---------------+
| Balabhoomi   | Balaguruswami |
| Labour india | Abraham       |
| Bagvat gita  | Balaguruswami |
| Alchemist    | Paulo coelho  |
| Titanic      | Di Caprio     |
```

d)Update the price of book_id 101.

```
DELIMITER //
```

```
CREATE PROCEDURE upd(in i int)
BEGIN UPDATE library SET price = price + 100
WHERE book_id = i;
  END//

DELIMITER ;
CALL upd(101);
CALL sele();
```

```
| book_id | book_name    | author        | price |
+---------+--------------+---------------+-------+
|     101 | Balabhoomi   | Balaguruswami |   600 |
|     102 | Labour india | Abraham       |  1500 |
|     103 | Bagvat gita  | Balaguruswami |  2000 |
|     104 | Alchemist    | Paulo coelho  |  1000 |
|     105 | Titanic      | Di Caprio     |  2500 |
```

e) Delete a the row from table library by passing the argument as any book_id.

```
  DELIMITER //
  CREATE PROCEDURE del(in i int)
  BEGIN DELETE FROM library
  WHERE book_id = i;
    END//

  DELIMITER ;
  CALL del(103);
  CALL sele();
```

```
   | book_id | book_name    | author        | price |
 +---------+--------------+---------------+-------+
|     101 | Balabhoomi   | Balaguruswami |   600 |
|     102 | Labour india | Abraham       |  1500 |
|     104 | Alchemist    | Paulo coelho  |  1000 |
|     105 | Titanic      | Di Caprio     |  2500 |
```

SQL 9

Q) Create table student with fields roll_no,stud_name, marks with the roll_no as primary key ,and stud_name as not null.

```
  CREATE TABLE student
  (
     roll_no int PRIMARY KEY,
     stud_name char(50)  NOT NULL,
     marks int
  );
```

a) Insert 5 records using procedure.

```
DELIMITER //
CREATE PROCEDURE ins
(
   r int,
   n char(50),
   m int
) BEGIN INSERT INTO student VALUES(r,n,m); END//

DELIMITER ;
CALL ins(1,'Safwan',92);
CALL ins(2,'Faris',88);
CALL ins(3,'Aysha',90);
CALL ins(4,'Jinu',78);
CALL ins(5,'Najiya',82);

SELECT * FROM student;
```

```
| roll_no | stud_name | marks |
+---------+-----------+-------+
|       1 | Safwan    |    92 |
|       2 | Faris     |    88 |
|       3 | Aysha     |    90 |
|       4 | Jinu      |    78 |
|       5 | Najiya    |    82 |
```

b)Select the table student using procedure.

```
DELIMITER //
CREATE PROCEDURE sele()
BEGIN SELECT * FROM student;
  END//

DELIMITER ;
CALL sele();
```

```
| roll_no | stud_name | marks |
+---------+-----------+-------+
|       1 | Safwan    |    92 |
|       2 | Faris     |    88 |
|       3 | Aysha     |    90 |
|       4 | Jinu      |    78 |
|       5 | Najiya    |    82 |
```

c) Select stud_name and marks from student.

```
DELIMITER //
CREATE PROCEDURE sel()
BEGIN SELECT stud_name,marks FROM student;
  END//
```

```
DELIMITER ;
CALL sel();

 | stud_name | marks |
+-----------+-------+
| Safwan    |    92 |
| Faris     |    88 |
| Aysha     |    90 |
| Jinu      |     78 |
| Najiya    |    82 |
```

d)Update the mark of student having roll_no 4.

```
DELIMITER //
CREATE PROCEDURE up(in m int)
BEGIN UPDATE student SET marks = 80
WHERE roll_no = m;
  END//

DELIMITER ;
CALL up(4);
CALL sele();

| roll_no | stud_name | marks |
+---------+-----------+-------+
|       1 | Safwan    |    92 |
|       2 | Faris     |    88 |
|       3 | Aysha     |    90 |
|       4 | Jinu      |    80 |
|       5 | Najiya    |     82 |
```

e) Delete a the row from table student by passing the argument as any roll_no.

```
DELIMITER //
CREATE PROCEDURE
      del(in i int) begin
DELETE FROM student
WHERE roll_no = i;
 END//

DELIMITER ;
CALL del(5);
CALL sele();

| roll_no | stud_name | marks |
+--------+-----------+-------+
|      1 | Safwan    |    92 |
|      2 | Faris     |    88 |
|      3 | Aysha     |    90 |
|      4 | Jinu      |    80 |
```

SQL 10

Q) Create a table loan with fields loan_no,branch_name, and amount with the loan_no as primary key .Create another table borrower with fields customer_name and loan_no where customer _name as not null.

```
CREATE TABLE loan
(
  loan_no int PRIMARY KEY,
  branch_name char(50)  NOT NULL,
  amount int
);


CREATE TABLE borrower
(
  customer_name char(50)  NOT NULL,
  loan_no int
);
```

a) Insert 5 records on both tables.

```
 INSERT INTO loan VALUES(170,'Downtown',3000);
 INSERT INTO loan VALUES(230,'Redwood',4000);
 INSERT INTO loan VALUES(260,'Perryridge',1700);
 INSERT INTO loan VALUES(234,'London',5600);
 INSERT INTO loan VALUES(238,'Newyork',4300);

  SELECT * FROM loan;

 | loan_no | branch_name | amount |
+---------+-------------+--------+
|     170 | Downtown    |   3000 |
|     230 | Redwood     |   4000 |
|     234 | London      |   5600 |
|     238 | Newyork     |   4300 |
|     260 | Perryridge  |   1700 |

 INSERT INTO borrower VALUES('John',170);
 INSERT INTO borrower VALUES('Angel',230);
 INSERT INTO borrower VALUES('Haris',260);
 INSERT INTO borrower VALUES('Meena',211);
 INSERT INTO borrower VALUES('jenni',238);

 SELECT * FROM borrower;

 | customer_name | loan_no |
+---------------+---------+
 | John          |     170 |
```

Angel	230
Haris	260
Meena	211
jenni	238

b) Display the loan details of customer.

```
SELECT * FROM loan
INNER JOIN borrower ON
    loan.loan_no=borrower.loan_no;
```

loan_no	branch_name	amount	customer_name	loan_no
170	Downtown	3000	John	170
230	Redwood	4000	Angel	230
260	Perryridge	1700	Haris	260
238	Newyork	4300	jenni	238

c) Display the all borrowers details with loan details.(borrower left outer join loan)

```
SELECT * FROM borrower
LEFT OUTER JOIN loan ON
    borrower.loan_no=loan.loan_no;
```

customer_name	loan_no	loan_no	branch_name	amount
John	170	170	Downtown	3000
Angel	230	230	Redwood	4000
Haris	260	260	Perryridge	1700
Meena	211	NULL	NULL	NULL
jenni	238	238	Newyork	4300

d)Display all loan details with customer name.(borrower right outer join loan)

```
SELECT * FROM borrower
RIGHT  OUTER JOIN loan ON
    borrower.loan_no=loan.loan_no;
```

customer_name	loan_no	loan_no	branch_name	amount
John	170	170	Downtown	3000
Angel	230	230	Redwood	4000
NULL	NULL	234	London	5600
jenni	238	238	Newyork	4300
Haris	260	260	Perryridge	1700

e) Display the details of all borrowers and customers

```
(
    SELECT * FROM borrower
    LEFT OUTER JOIN loan ON
        borrower.loan_no=loan.loan_no
```

```
)   UNION
(
   SELECT * FROM borrower
   RIGHT OUTER JOIN loan ON
            borrower.loan_no=loan.loan_no
);
```

```
| customer_name | loan_no | loan_no | branch_name | amount |
+---------------+---------+---------+-------------+--------+
| John          |     170 |     170 | Downtown    |   3000 |
| Angel         |     230 |     230 | Redwood     |   4000 |
| Haris         |     260 |     260 | Perryridge  |   1700 |
| Meena         |     211 |    NULL | NULL        |   NULL |
| jenni         |     238 |     238 | Newyork     |   4300 |
| NULL          |    NULL |     234 | London      |   5600 |
```

f) Display all details of customer who have loan also.

```
  SELECT * FROM borrower
  NATURAL JOIN loan;
```

```
 | loan_no | customer_name | branch_name | amount |
+---------+---------------+-------------+--------+
|     170 | John          | Downtown    |   3000 |
|     230 | Angel         | Redwood     |   4000 |
|     260 | Haris         | Perryridge  |   1700 |
|     238 |jenni          | Newyork     |   4300 |
```

SQL 11

Q) Create a table tutorials with fields id,title,author,create and date with id as primary key.Back up the details of tutorials table using cursor.

```
 CREATE TABLE tutorials
 (
   id int PRIMARY KEY,
   title varchar(50),
   author char(20),
   date varchar(20)
);
```

```
INSERT INTO tutorials VALUES(1,'Java','Krishna','2019-09-01');
INSERT INTO tutorials VALUES(2,'jfreecharts','Satish','2019-05-01');
INSERT INTO tutorials VALUES(3,'Javaspring','Amit','2019-05-01');
INSERT INTO tutorials VALUES(4,'Android','Ram','2019-03-01');
INSERT INTO tutorials VALUES(5,'Cassandra','Pruthvi','2019-04-01');
```

```
SELECT * FROM tutorials;
```

```
| id | title      | author  | date       |
```

```
+----+-------------+---------+------------+
|  1 | Java        | Krishna | 2019-09-01 |
|  2 | jfreecharts | Satish  | 2019-05-01 |
|  3 | Javaspring  | Amit    | 2019-05-01 |
|  4 | Android     | Ram     | 2019-03-01 |
|  5 | Cassandra   | Pruthvi | 2019-04-01 |
```

```
CREATE TABLE backup
(
   Id int PRIMARY KEY,
   Title varchar(50),
   Author char(20),
   Date varchar(20)
);

DELIMITER //
CREATE PROCEDURE exampleproc()
BEGIN DECLARE done
INT DEFAULT 0;
DECLARE tutorialid INTEGER;
DECLARE tutorialtitle,tutorialauthor,tutorialdate varchar(20);
DECLARE cur cursor FOR SELECT * FROM tutorials;
DECLARE continue handler FOR NOT found SET done = 1;
        open cur;
        label:LOOP fetch cur INTO
        tutorialid,tutorialtitle,tutorialauthor,tutorialdate;
INSERT INTO backup VALUES
(
    tutorialid,
    tutorialtitle,
    tutorialauthor,
    tutorialdate
);
IF done = 1 THEN leave label;
END IF;
END LOOP;
close cur;
 END//

DELIMITER ;
CALL exampleproc();
SELECT * FROM backup;
```

```
 | id | title       | author  | date       |
+----+-------------+---------+------------+
|  1 | Java        | Krishna | 2019-09-01 |
|  2 | jfreecharts | Satish  | 2019-05-01 |
|  3 | Javaspring  | Amit    | 2019-05-01 |
|  4 | Android     | Ram     | 2019-03-01 |
|  5 | Cassandra   | Pruthvi | 2019-04-01 |
```

SQL 12

Q) Create a table customer with fields c_no as primary key, c_name, purchased_item, cost and date.Assign suitable constraints for each attributes.

```
CREATE TABLE customer
(
   c_no int PRIMARY KEY,
   c_name varchar(50),
   purchased_item char(50),
   cost int,
   Date varchar(20)
);
```

a) Insert 5 records.

```
INSERT INTO customer VALUES(101,'Krishna','Kurthi',1500,'2019-09-01');
INSERT INTO customer VALUES(102,'Satish','Cleaning supplies',500,'2019-05-11');
INSERT INTO customer VALUES(103,'Amit','Watch',2000,'2019-05-01');
INSERT INTO customer VALUES(104,'Ram','Perfumes',1500,'2019-03-21');
INSERT INTO customer VALUES(105,'Pruthvi','Fridge',15000,'2019-04-01');
SELECT * FROM customer;
```

```
| c_no | c_name  | purchased_item    | cost  | Date       |
+------+---------+-------------------+-------+------------+
|  101 | Krishna | Kurthi            |  1500 | 2019-09-01 |
|  102 | Satish  | Cleaning supplies |   500 | 2019-05-11 |
|  103 | Amit    | Watch             |  2000 | 2019-05-01 |
|  104 | Ram     | Perfumes          |  1500 | 2019-03-21 |
|  105 | Pruthvi | Fridge            | 15000 | 2019-04-01 |
```

b) Add one more column purchased_no.

```
ALTER TABLE customer
add column purchased_no int;
SELECT * FROM customer;
```

```
| c_no | c_name  | purchased_item    | cost  | Date       | purchased_no |
+------+---------+-------------------+-------+------------+--------------+
|  101 | Krishna | Kurthi            |  1500 | 2019-09-01 |         NULL |
|  102 | Satish  | Cleaning supplies |   500 | 2019-05-11 |         NULL |
|  103 | Amit    | Watch             |  2000 | 2019-05-01 |         NULL |
|  104 | Ram     | Perfumes          |  1500 | 2019-03-21 |         NULL |
|  105 | Pruthvi | Fridge            | 15000 | 2019-04-01 |         NULL |
```

c) Update the records of purchased_no.

```
UPDATE customer SET
   purchased_no = 1001
WHERE c_no = 101;

UPDATE customer SET
```

```
    purchased_no = 1002
 WHERE c_no = 102;

 UPDATE customer SET
   purchased_no = 1003
 WHERE c_no = 103;

UPDATE customer SET
   purchased_no = 1004
WHERE c_no = 104;


 UPDATE customer SET
    purchased_no = 1005
 WHERE c_no = 105;

 SELECT * FROM customer;

 | c_no | c_name  | purchased_item | cost   | Date           | purchased_no |
+------+---------+------------------+-------+------------+--------------+
|  101 | Krishna  | Kurthi              |  1500 | 2019-09-01 |         1001 |
|  102 | Satish     | Cleaning supplies |   500 | 2019-05-11 |         1002 |
|  103 | Amit       | Watch               |  2000 | 2019-05-01 |         1003 |
|  104 | Ram       | Perfumes            |  1500 | 2019-03-21 |         1004 |
|  105 | Pruthv i  | Fridge              | 15000 | 2019-04-01 |         1005 |
```

d) Display the avarage cost from customer.

```
 SELECT AVG(cost) FROM customer;

 | AVG(cost) |
 +-----------+
 | 4100.0000 |
```

e) Delete customer details whose c_no=103.

```
 DELETE FROM customer WHERE c_no = 103;
 SELECT * FROM customer;

 | c_no | c_name  | purchased_item | cost    | Date           | purchased_no |
+------+---------+------------------+-------+------------+--------------+
|  101 | Krishna | Kurthi              |  1500  | 2019-09-01 |         1001 |
|  102 | Satish  | Cleaning supplies  |   500   | 2019-05-11 |         1002 |
|  104 | Ram     | Perfumes            |  1500   | 2019-03-21 |         1004 |
|  105 | Pruthvi | Fridge              | 15000  | 2019-04-01 |         1005 |
```

f) Sort the customer’s details on ascending order of customer’s name.

```
 SELECT * FROM customer
 ORDER BY c_name ASC;
```

c_no	c_name	purchased_item	cost	Date	purchased_no
101	Krishna	Kurthi	1500	2019-09-01	1001
105	Pruthvi	Fridge	15000	2019-04-01	1005
104	Ram	Perfumes	1500	2019-03-21	1004
102	Satish	Cleaning supplies	500	2019-05-11	1002

g) Display customer name ,purchased item with date.

```
SELECT c_name,purchased_item,Date FROM customer;
```

c_name	purchased_item	Date
Krishna	Kurthi	2019-09-01
Satish	Cleaning supplies	2019-05-11
Ram	Perfumes	2019-03-21
Pruthvi	Fridge	2019-04-01

h) Delete column cost from customer.

```
ALTER TABLE customer
DROP column cost;

SELECT * FROM customer;
```

c_no	c_name	purchased_item	Date	purchased_no
101	Krishna	Kurthi	2019-09-01	1001
102	Satish	Cleaning supplies	2019-05-11	1002
104	Ram	Perfumes	2019-03-21	1004
105	Pruthvi	Fridge	2019-04-01	1005

i) Delete table customer.

```
DROP TABLE customer;
```