DBMS ASSIGNMENTS

NAME: Amogh Garg
ROLL NUMBER: 2020UCO1688
SECTION: COE-3

Q 1: Consider the following relational schema SAILORS (sid, sname, rating, date_of_birth) BOATS (bid, bname, color) RESERVES (sid, bid, date, time slot)
Write the following queries in SQL and relational algebra
a) Find sailors who've reserved at least one boat
b) Find names of sailors who've reserved a red or a green boat in the month of March.
c) Find names of sailors who've reserved a red and a green boat
d) Find sid of sailors who have not reserved a boat after Jan 2018.
e) Find sailors whose rating is greater than that of all the sailors named "John"
f) Find sailors who've reserved all boats
g) Find name and age of the oldest sailor(s)
h) Find the age of the youngest sailor for each rating with at least 2 such sailors

CREATION OF THE TABLE:
create table sailors (sid int primary key, sname varchar (20), rating int, date_of_birth date);
create table boats (bid int primary key, bname varchar (20), color varchar (10));
create table reserves (sid int not null, bid int not null, dt date not null, timeslot int, foreign key (sid) references sailors(sid), foreign key (bid) references boats (bid));

```
MariaDB [question_1]> describe sailors;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| sid           | int(11)     | NO   | PRI | NULL    |       |
| sname         | varchar(20) | YES  |     | NULL    |       |
| rating        | int(11)     | YES  |     | NULL    |       |
| date_of_birth | date        | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.012 sec)

MariaDB [question_1]> describe boats;
+-------+-------------+------+-----+---------+-------+
| Field | Type        | Null | Key | Default | Extra |
+-------+-------------+------+-----+---------+-------+
| bid   | int(11)     | NO   | PRI | NULL    |       |
| bname | varchar(20) | YES  |     | NULL    |       |
| color | varchar(10) | YES  |     | NULL    |       |
+-------+-------------+------+-----+---------+-------+
3 rows in set (0.010 sec)

MariaDB [question_1]> describe reserves;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| sid      | int(11) | NO   | MUL | NULL    |       |
| bid      | int(11) | NO   | MUL | NULL    |       |
| dt       | date    | NO   |     | NULL    |       |
| timeslot | int(11) | YES  |     | NULL    |       |
+----------+---------+------+-----+---------+-------+
4 rows in set (0.029 sec)
```

INSERTION OF VALUES:

```
insert into sailors values (1, 'Amogh', 7, "1999-01-03");
insert into sailors values (2, 'Vasco-de-Gamma', 9, "1998-07-12");
insert into sailors values (3, 'MS Dhoni', 9, "1996-05-22");
insert into sailors values (4, 'Singh', 8, "1993-01-23");
insert into sailors values (5, 'Gulliver',8,"2001-09-01");
insert into Boats values (101, 'Interlake', 'blue');
insert into Boats values (102, 'Interlake', 'red');
insert into Boats values (103, 'Clipper', 'green');
insert into Boats values (104, 'Marine', 'red');
insert into Reserves values (1, 101, '2017-10-10',1);
insert into Reserves values (1, 102, '2017-10-10',2);
insert into Reserves values (1, 103, '2017-10-10',2);
insert into Reserves values (1, 104, '2017-10-10',2);
insert into Reserves values (1, 101, '2019-10-10',1);
insert into Reserves values (2, 102, '2011-03-01',3);
insert into Reserves values (2, 102, '2019-11-07',3);
insert into Reserves values (3, 101, '2017-11-07',2);
insert into Reserves values (3, 102, '2017-08-07',2);
insert into Reserves values (4, 103, '2017-03-19',1);
insert into Reserves values (2, 103, '2017-03-19',3);
```

```
MariaDB [question_1]> select * from sailors;
+-----+-----------------+--------+---------------+
| sid | sname           | rating | date_of_birth |
+-----+-----------------+--------+---------------+
|   1 | Amogh           |      7 | 1999-01-03    |
|   2 | Vasco-de-Gamma  |      9 | 1998-07-12    |
|   3 | MS Dhoni        |      9 | 1996-05-22    |
|   4 | Singh           |      8 | 1993-01-23    |
|   5 | Gulliver        |      8 | 2001-09-01    |
+-----+-----------------+--------+---------------+
5 rows in set (0.001 sec)

MariaDB [question_1]> select * from boats;
+-----+-----------+-------+
| bid | bname     | color |
+-----+-----------+-------+
| 101 | Interlake | blue  |
| 102 | Interlake | red   |
| 103 | Clipper   | green |
| 104 | Marine    | red   |
+-----+-----------+-------+
4 rows in set (0.001 sec)

MariaDB [question_1]> select * from reserves;
+-----+-----+------------+----------+
| sid | bid | dt         | timeslot |
+-----+-----+------------+----------+
|   1 | 101 | 2017-10-10 |        1 |
|   1 | 102 | 2017-10-10 |        2 |
|   1 | 103 | 2017-10-10 |        2 |
|   1 | 104 | 2017-10-10 |        2 |
|   1 | 101 | 2019-10-10 |        1 |
|   2 | 102 | 2011-03-01 |        3 |
|   2 | 102 | 2019-11-07 |        3 |
|   3 | 101 | 2017-11-07 |        2 |
|   3 | 102 | 2017-08-07 |        2 |
|   4 | 103 | 2017-03-19 |        1 |
|   2 | 103 | 2017-03-19 |        3 |
+-----+-----+------------+----------+
11 rows in set (0.001 sec)
```

QUERIES: a) select sname from sailors where sid in (select sid from reserves);

```
MariaDB [question_1]> select sname from sailors where sid in (select sid from reserves);
+---------------+
| sname         |
+---------------+
| Amogh         |
| Vasco-de-Gamma |
| MS Dhoni      |
| Singh         |
+---------------+
4 rows in set (0.004 sec)
```

b) select sname from sailors where sid in (select r.sid from boats b, reserves r where r.bid = b.bid AND b.color = "red" and (select extract(month from r.dt)="03") union select r2.sid from boats b2, reserves r2 where r2.bid = b2.bid AND b2.color = "green" and (select extract(month from r2.dt)="03"));

```
+----------------+
| sname          |
+----------------+
| Vasco-de-Gamma |
| Singh          |
+----------------+
2 rows in set (0.013 sec)
```

c) select distinct S1.sname from sailors S1, reserves R1, boats B1, reserves R2, boats B2 where S1.sid=R1.sid and R1.bid=B1.bid and S1.sid=R2.sid and R2.bid=B2.bid and B1.color="red" and B2.color="green";

```
+----------------+
| sname          |
+----------------+
| Amogh          |
| Vasco-de-Gamma |
+----------------+
2 rows in set (0.001 sec)
```

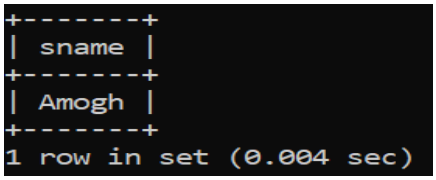
d) select sid from sailors where sid not in (select sid from reserves where dt>="2018-01-01");

```
+-----+
| sid |
+-----+
|   3 |
|   4 |
|   5 |
+-----+
3 rows in set (0.003 sec)
```

e) select sname from sailors where rating > all (select rating from sailors where sname="Amogh");

```
+----------------+
| sname          |
+----------------+
| Vasco-de-Gamma |
| MS Dhoni       |
| Singh          |
| Gulliver       |
+----------------+
4 rows in set (0.003 sec)
```

f) select sname from sailors s where not exists (select * from boats b where not exists (select * from reserves r where r.sid=s.sid AND r.bid=b.bid));



g) select sname,TIMESTAMPDIFF(YEAR,date_of_birth,"2021-11-16") as age from sailors where date_of_birth >=all(select date_of_birth from sailors);

h) select rating,min(TIMESTAMPDIFF(YEAR,date_of_birth,"2021-11-16")) as minage from sailors group by rating having count(*)>1;

Q2. Consider the following relational schema: CUSTOMER (cust_num, cust_lname , cust_fname, cust_balance); PRODUCT (prod_num, prod_name, price) INVOICE (inv_num, prod_num, cust_num, inv_date ,unit_sold, inv_amount);
Write SQL queries and relational algebraic expression for the following
a) Find the names of the customer who have purchased no item. Set default value of Cust_balance as 0 for such customers.
b) Write the trigger to update the CUST_BALANCE in the CUSTOMER table when a new invoice record is entered for the customer.
c) Find the customers who have purchased more than three units of a product on a day.
d) Write a query to illustrate Left Outer, Right Outer and Full Outer Join.
e) Count number of products sold on each date.
f) As soon as customer balance becomes greater than Rs. 100,000, copy the customer_num in new table called "GOLD_CUSTOMER"
g) Add a new attribute CUST_DOB in customer table

CREATION OF TABLES:

create table customer (cust_num int, cust_lname varchar(50), cust_fname varchar(50) not null, cust_balance int default 0, primary key (cust_num));
create table product (prod_num int, prod_name varchar(50) not null, price int not null, primary key(prod_num));
create table invoice (inv_num int, prod_num int not null, cust_num int not null, inv_date date not null, unit_sold int not null, inv_amount int not null, primary key(inv_num), foreign key(prod_num) references product(prod_num), foreign key(cust_num) references customer(cust_num), check(unit_sold>0));

```
MariaDB [question_2]> describe customer;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| cust_num     | int(11)     | NO   | PRI | NULL    |       |
| cust_lname   | varchar(50) | YES  |     | NULL    |       |
| cust_fname   | varchar(50) | NO   |     | NULL    |       |
| cust_balance | int(11)     | YES  |     | 0       |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.012 sec)

MariaDB [question_2]> describe product;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| prod_num  | int(11)     | NO   | PRI | NULL    |       |
| prod_name | varchar(50) | NO   |     | NULL    |       |
| price     | int(11)     | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.011 sec)

MariaDB [question_2]> describe invoice;
+------------+---------+------+-----+---------+-------+
| Field      | Type    | Null | Key | Default | Extra |
+------------+---------+------+-----+---------+-------+
| inv_num    | int(11) | NO   | PRI | NULL    |       |
| prod_num   | int(11) | NO   | MUL | NULL    |       |
| cust_num   | int(11) | NO   | MUL | NULL    |       |
| inv_date   | date    | NO   |     | NULL    |       |
| unit_sold  | int(11) | NO   |     | NULL    |       |
| inv_amount | int(11) | NO   |     | NULL    |       |
+------------+---------+------+-----+---------+-------+
6 rows in set (0.011 sec)
```

INSERTION OF DATA:

Insert into customer (cust_num, cust_lname, cust_fname, cust_balance) values (1, ‘Garg’, ‘Amogh’, 0), (2, ‘Ambani’, ‘Mukesh’, 250), (3, ‘Modi’, ‘Narender’, 1000);
Insert into product (prod_num,prod_name,price) values (2, ‘Gold’, 15000), (1, ‘Earphones’, 1250), (3, ‘Shoes’, 400);
Insert into invoice values (1,2,1,'2019-01-01',4,600000),(2,1,2,'2019-01-01',2,2500), (3,2,2,'2019-01-04',1,150000);

```
MariaDB [question_2]> select * from customer;
+----------+------------+------------+--------------+
| cust_num | cust_lname | cust_fname | cust_balance |
+----------+------------+------------+--------------+
|        1 | Garg       | Amogh      |            0 |
|        2 | Ambani     | Mukesh     |          250 |
|        3 | Modi       | Narender   |         1000 |
+----------+------------+------------+--------------+
3 rows in set (0.001 sec)

MariaDB [question_2]> select * from product;
+----------+-----------+-------+
| prod_num | prod_name | price |
+----------+-----------+-------+
|        1 | Earphones |  1250 |
|        2 | Gold      | 15000 |
|        3 | Shoes     |   400 |
+----------+-----------+-------+
3 rows in set (0.001 sec)

MariaDB [question_2]> select * from invoice;
+---------+----------+----------+------------+-----------+------------+
| inv_num | prod_num | cust_num | inv_date   | unit_sold | inv_amount |
+---------+----------+----------+------------+-----------+------------+
|       1 |        2 |        1 | 2019-01-01 |         4 |     600000 |
|       2 |        1 |        2 | 2019-01-01 |         2 |       2500 |
|       3 |        2 |        2 | 2019-01-04 |         1 |     150000 |
+---------+----------+----------+------------+-----------+------------+
3 rows in set (0.001 sec)
```

QUERIES: a) select cust_fname, cust_lname from customer where cust_balance=0;

```
MariaDB [question_2]> select cust_fname, cust_lname from customer where cust_balance=0;
+------------+------------+
| cust_fname | cust_lname |
+------------+------------+
| Amogh      | Garg       |
+------------+------------+
```

b) create trigger upd_cust before insert on invoice for each row update customer c set c.cust_balance=c.cust_balance+new.inv_amount where c.cust_num=new.cust_num;

c) select cust_num, cust_fname, cust_lname from customer where cust_num in (select cust_num from invoice group by cust_num, inv_date, prod_num having sum(unit_sold)>3);

```
+----------+------------+------------+
| cust_num | cust_fname | cust_lname |
+----------+------------+------------+
|        1 | Amogh      | Garg       |
+----------+------------+------------+
1 row in set (0.006 sec)
```

d) select cust_fname, cust_lname, i.inv_amount from customer c left join invoice i on c.cust_num=i.cust_num;
select cust_fname, cust_lname, i.inv_amount from customer c right join invoice i on c.cust_num=i.cust_num;
select cust_fname, cust_lname, i.inv_amount from customer c left join invoice i on c.cust_num=i.cust_num union select cust_fname, cust_lname, i.inv_amount from customer c right join invoice i on c.cust_num=i.cust_num;

```
MariaDB [question_2]> select cust_fname, cust_lname, i.inv_amount from customer c left join invoice i on c.cust_num=i.cust_num;
+------------+------------+------------+
| cust_fname | cust_lname | inv_amount |
+------------+------------+------------+
| Amogh      | Garg       |     600000 |
| Mukesh     | Ambani     |       2500 |
| Mukesh     | Ambani     |     150000 |
| Narender   | Modi       |       NULL |
+------------+------------+------------+
4 rows in set (0.001 sec)

MariaDB [question_2]> select cust_fname, cust_lname, i.inv_amount from customer c right join invoice i on c.cust_num=i.cust_num;
+------------+------------+------------+
| cust_fname | cust_lname | inv_amount |
+------------+------------+------------+
| Amogh      | Garg       |     600000 |
| Mukesh     | Ambani     |       2500 |
| Mukesh     | Ambani     |     150000 |
+------------+------------+------------+
3 rows in set (0.001 sec)

MariaDB [question_2]> select cust_fname, cust_lname, i.inv_amount from customer c left join invoice i on c.cust_num=i.cust_num union
ount from customer c right join invoice i on c.cust_num=i.cust_num;
+------------+------------+------------+
| cust_fname | cust_lname | inv_amount |
+------------+------------+------------+
| Amogh      | Garg       |     600000 |
| Mukesh     | Ambani     |       2500 |
| Mukesh     | Ambani     |     150000 |
| Narender   | Modi       |       NULL |
+------------+------------+------------+
4 rows in set (0.007 sec)
```

e) select inv_date, sum(unit_sold) as total_sales from invoice group by inv_date;

```
MariaDB [question_2]> select inv_date, sum(unit_sold) as total_sales from invoice group by inv_date;
+------------+-------------+
| inv_date   | total_sales |
+------------+-------------+
| 2019-01-01 |           6 |
| 2019-01-04 |           1 |
+------------+-------------+
2 rows in set (0.001 sec)
```

f) create table gold_customer(cust_num int, cust_lname varchar(20), cust_fname varchar(20), primary key(cust_num));
create trigger in_gold after update on customer for each row insert into gold_customer(select cust_num, cust_lname, cust_fname from customer where cust_num=new.cust_num and cust_balance >100000 and cust_num not in (select cust_num from gold_customer));

```
MariaDB [question_2]> create table gold_customer(cust_num int, cust_lname varchar(20), cust_fname varchar(20), primary key(cust_num));
Query OK, 0 rows affected (0.029 sec)

MariaDB [question_2]> create trigger in_gold after update on customer for each row insert into gold_customer(select cust_num, cust_lname, cust_fname from customer where  cu
st_num=new.cust_num and cust_balance >100000 and cust_num not in (select cust_num from gold_customer));
Query OK, 0 rows affected (0.013 sec)
```

g) alter table customer add column cust_dob date;

```
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| cust_num     | int(11)     | NO   | PRI | NULL    |       |
| cust_lname   | varchar(50) | YES  |     | NULL    |       |
| cust_fname   | varchar(50) | NO   |     | NULL    |       |
| cust_balance | int(11)     | YES  |     | 0       |       |
| cust_dob     | date        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.009 sec)
```

Q.3 Consider the following relational schema: DEPARTMENT (Department_ID, Name, Location_ID) JOB (Job_ID, Function) EMPLOYEE (Employee_ID, name, DOB, Job_ID, Manager_ID, Hire_Date, Salary, department_id)

Answer the following queries using SQL and relational algebra:

1) Write a query to count number of employees who joined in March 2015

2) Display the Nth highest salary drawing employee details.

3) Find the budget (total salary) of each department.

4) Find the department with maximum budget.

5) Create a view to show number of employees working in Delhi and update it automatically when the database is modified.

6) Write a trigger to ensure that no employee of age less than 25 can be inserted in the database.

CREATION OF TABLE:

create table deparment (department_id int, name varchar(20) not null, location_id int, primary key (department_id));
create table job (job_id int primary key, function varchar(20));
create table employee (employee_id int primary key, name varchar(20), job_id int not null,

manager_id int, hire_date date, salary int, department_id int not null, foreign key (job_id) references job (job_id), foreign key (department_id) references deparment (department_id));

```
MariaDB [question_3]> describe deparment;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| department_id | int(11)     | NO   | PRI | NULL    |       |
| name          | varchar(20) | NO   |     | NULL    |       |
| location_id   | int(11)     | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.012 sec)

MariaDB [question_3]> describe job;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| job_id   | int(11)     | NO   | PRI | NULL    |       |
| function | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.027 sec)

MariaDB [question_3]> describe employee;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| employee_id   | int(11)     | NO   | PRI | NULL    |       |
| name          | varchar(20) | YES  |     | NULL    |       |
| job_id        | int(11)     | NO   | MUL | NULL    |       |
| manager_id    | int(11)     | YES  |     | NULL    |       |
| hire_date     | date        | YES  |     | NULL    |       |
| salary        | int(11)     | YES  |     | NULL    |       |
| department_id | int(11)     | NO   | MUL | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
7 rows in set (0.013 sec)
```

INSERTION OF DATA:

insert into deparment values (101, ‘Admin’, 1), (102, ‘Marketing’, 2), (103, ‘Sales’, 3);
insert into job values (300, ‘Head’), (301, ‘Manager’), (302, ‘Salesman’);
insert into employee values (500, ‘Martin’, 300, 1000, ‘2020-01-01’, 5000, 101), (501, ‘Joe’, 301, 1001, ‘2020-01-02’, 10000, 102), (502, ‘Smith’, 302, 1002, ‘2020-02-01’, 7500, 103), (503, ‘Jack’, 300, 1003, ‘2021-01-01’, 5000, 101);

```
MariaDB [question_3]> select * from deparment;
+---------------+-----------+-------------+
| department_id | name      | location_id |
+---------------+-----------+-------------+
|           101 | Admin     |           1 |
|           102 | Marketing |           2 |
|           103 | Sales     |           3 |
+---------------+-----------+-------------+
3 rows in set (0.001 sec)

MariaDB [question_3]> select * from job;
+--------+----------+
| job_id | function |
+--------+----------+
|    300 | Head     |
|    301 | Manager  |
|    302 | Salesman |
+--------+----------+
3 rows in set (0.001 sec)

MariaDB [question_3]> select * from employee;
+-------------+--------+--------+------------+------------+--------+---------------+
| employee_id | name   | job_id | manager_id | hire_date  | salary | department_id |
+-------------+--------+--------+------------+------------+--------+---------------+
|         500 | Martin |    300 |       1000 | 2020-01-01 |   5000 |           101 |
|         501 | Joe    |    301 |       1001 | 2020-01-02 |  10000 |           102 |
|         502 | Smith  |    302 |       1002 | 2020-02-01 |   7500 |           103 |
|         503 | Jack   |    300 |       1003 | 2021-01-01 |   5000 |           101 |
+-------------+--------+--------+------------+------------+--------+---------------+
4 rows in set (0.001 sec)
```

QUERIES:

1. select count(employee_id) from employee where hire_date > ‘2020-01-01’ and hire_date < ‘2020-02-01’;

```
MariaDB [question_3]> select count(employee_id) from employee where hire_date > '2020-01-01' and hire_date < '2020-02-01';
+-------------------+
| count(employee_id) |
+-------------------+
|                 1 |
+-------------------+
1 row in set (0.007 sec)
```

2. select * from employee order by salary desc;

```
MariaDB [question_3]> select * from employee order by salary desc;
+-------------+--------+--------+------------+------------+--------+---------------+
| employee_id | name   | job_id | manager_id | hire_date  | salary | department_id |
+-------------+--------+--------+------------+------------+--------+---------------+
|         501 | Joe    |    301 |       1001 | 2020-01-02 |  10000 |           102 |
|         502 | Smith  |    302 |       1002 | 2020-02-01 |   7500 |           103 |
|         500 | Martin |    300 |       1000 | 2020-01-01 |   5000 |           101 |
|         503 | Jack   |    300 |       1003 | 2021-01-01 |   5000 |           101 |
+-------------+--------+--------+------------+------------+--------+---------------+
4 rows in set (0.001 sec)
```

3.select department_id, sum(salary) from employee group by department_id;

```
MariaDB [question_3]> select department_id, sum(salary) from employee group by department_id;
+---------------+-------------+
| department_id | sum(salary) |
+---------------+-------------+
|           101 |       10000 |
|           102 |       10000 |
|           103 |        7500 |
+---------------+-------------+
3 rows in set (0.001 sec)
```

4. select department_id, sum(salary) from employee group by department_id order by sum(salary) desc limit 1;

```
MariaDB [question_3]> select department_id, sum(salary) from employee group by department_id order by sum(salary) desc limit 1;
+---------------+-------------+
| department_id | sum(salary) |
+---------------+-------------+
|           102 |       10000 |
+---------------+-------------+
1 row in set (0.004 sec)
```

5. create view delhi_pop as select count(employee_id) from employee,deparment where deparment.location_id=1;

```
MariaDB [question_3]> create view delhi_pop as select count(employee_id) from employee,deparment where location_id=1;
Query OK, 0 rows affected (0.013 sec)
```

6. create trigger check_age before insert on employee for each row begin if new.dob > 1993-01-01 then signal sqlstate ‘4500’ set message_text = ‘Age must be atleast 25 years!’;