DATABASE MANAGEMENT SYSTEM LAB FILE

Name: Shantanu Goyal
Roll Number: 2019UCS2026

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

A1:
CREATION OF THE TABLE

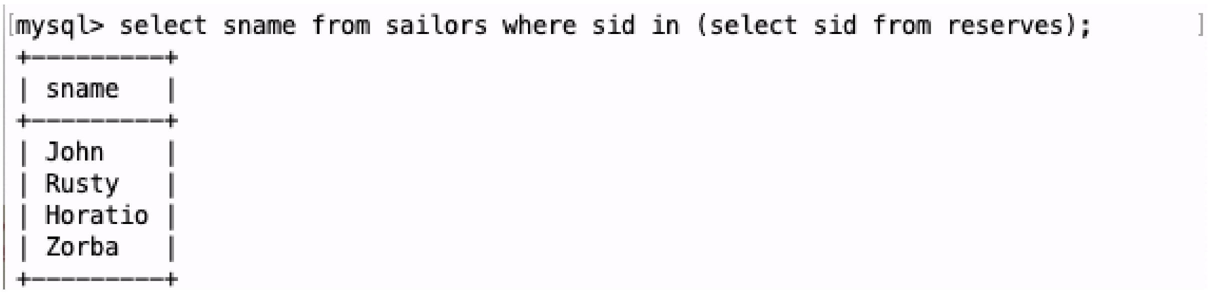
```
create table sailors( sid int primary key,
sname varchar(20),
rating int,
date_of_birth date);

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));
```

INSERTING VALUES

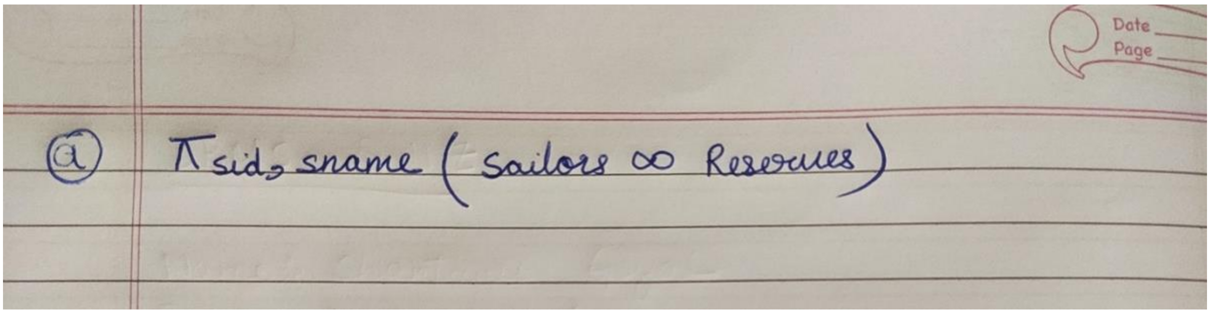
```
insert into sailors values (1, 'John', 7, "1999-01-03");
insert into sailors values (2, 'Rusty', 9, "1998-07-12");
insert into sailors values (3, 'Horatio', 9, "1996-05-22");
insert into sailors values (4, 'Zorba', 8, "1993-01-23");
insert into sailors values (5, 'Julius',8,"2001-09-01");

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);
```

QUERIES

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



```
mysql> select sname from sailors where sid in (select sid from reserves);
+---------+
| sname   |
+---------+
| John    |
| Rusty   |
| Horatio |
| Zorba   |
+---------+
```

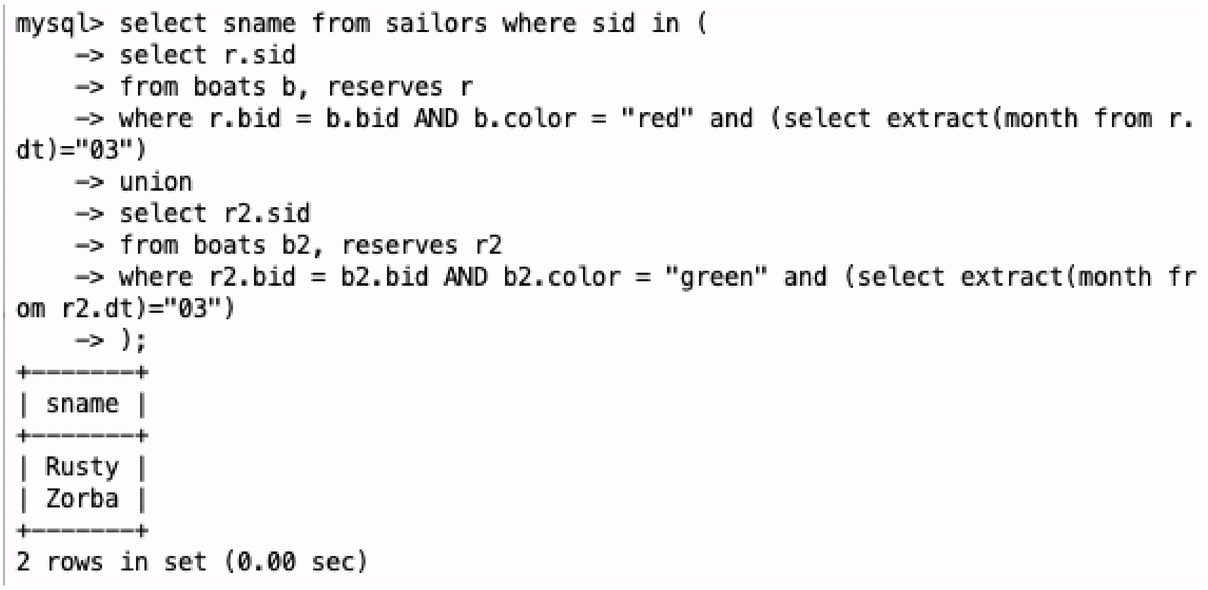


Date

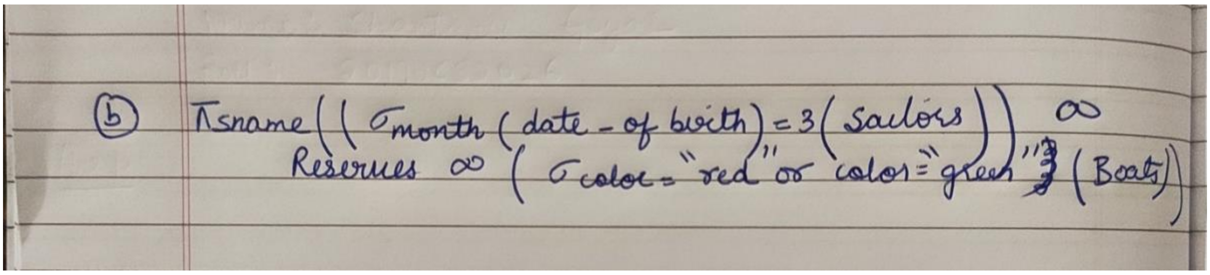
Page

ⓐ $\pi_{sid, sname}$ (Sailors ∞ Reserves)

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")
);



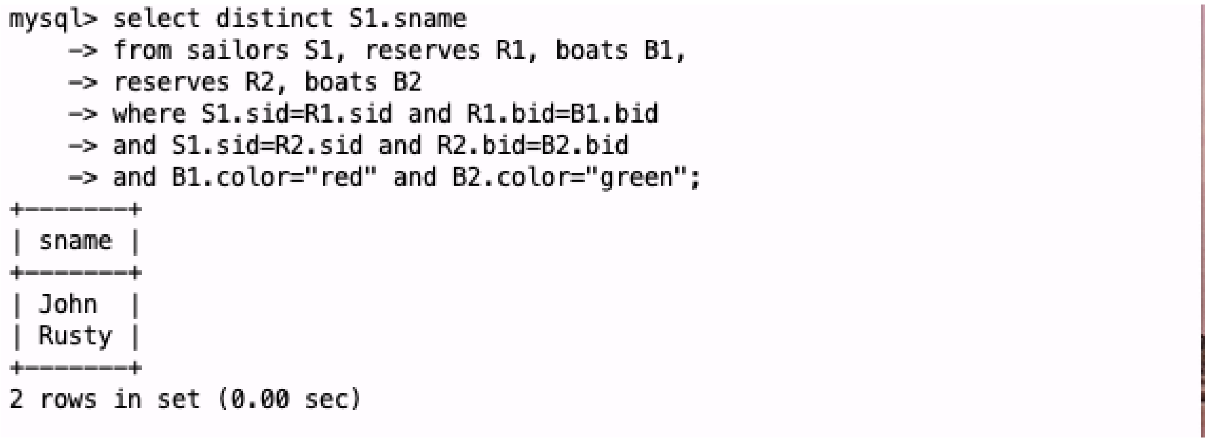
```
mysql> select sname from sailors where sid in (
    -> select r.sid
    -> from boats b, reserves r
    -> where r.bid = b.bid AND b.color = "red" and (select extract(month from r.
dt)="03")
    -> union
    -> select r2.sid
    -> from boats b2, reserves r2
    -> where r2.bid = b2.bid AND b2.color = "green" and (select extract(month fr
om r2.dt)="03")
    -> );
+--------+
| sname  |
+--------+
| Rusty  |
| Zorba  |
+--------+
2 rows in set (0.00 sec)
```



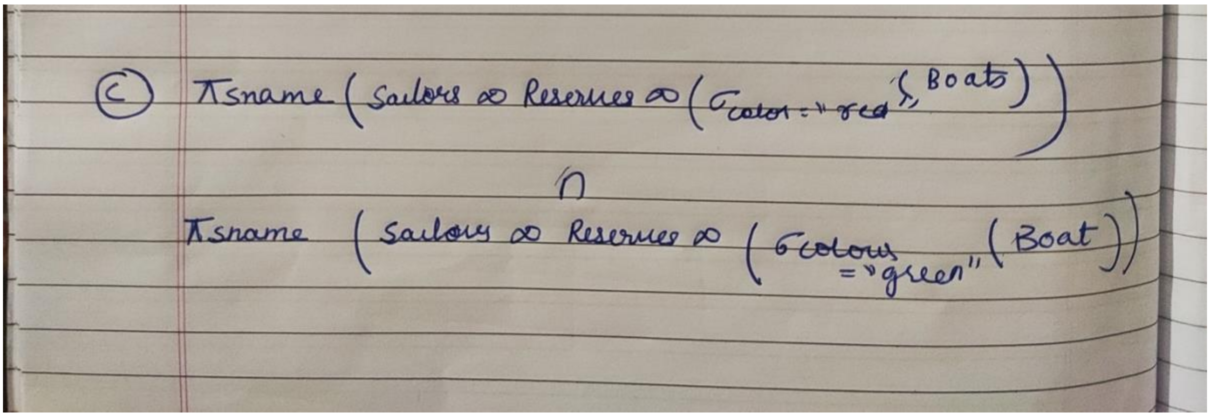
(b) $\pi_{sname}((\sigma_{month(date-of\ birth)=3}(Sailors)) \bowtie Reserves \bowtie (\sigma_{color=\text{"red" or color="green"}}(Boats)))$

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";
```



```
mysql> 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  |
+--------+
| John   |
| Rusty  |
+--------+
2 rows in set (0.00 sec)
```

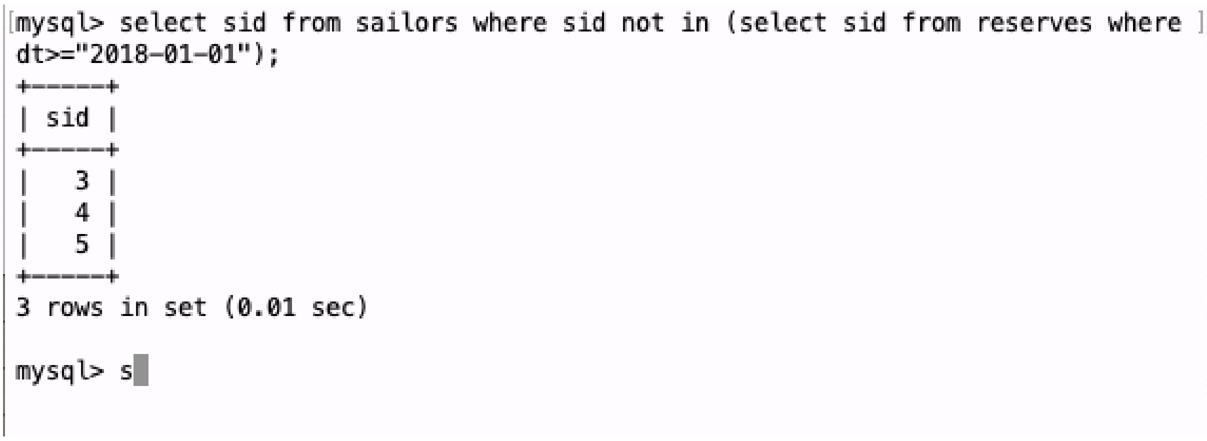


Ⓒ $\pi_{sname}(Sailors \infty Reserves \infty (\sigma_{color="red"}(Boats)))$

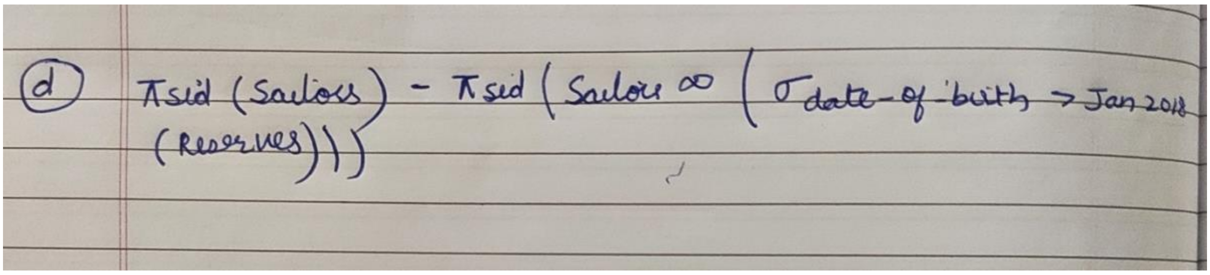
$\cap$

$\pi_{sname}(Sailors \infty Reserves \infty (\sigma_{colour="green"}(Boat)))$

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

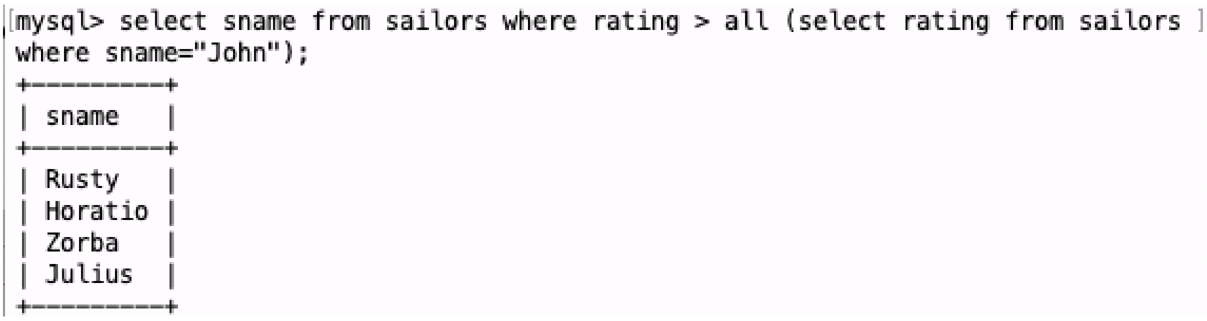


```
mysql> select sid from sailors where sid not in (select sid from reserves where
dt>="2018-01-01");
+------+
| sid |
+------+
|   3 |
|   4 |
|   5 |
+------+
3 rows in set (0.01 sec)

mysql> s
```

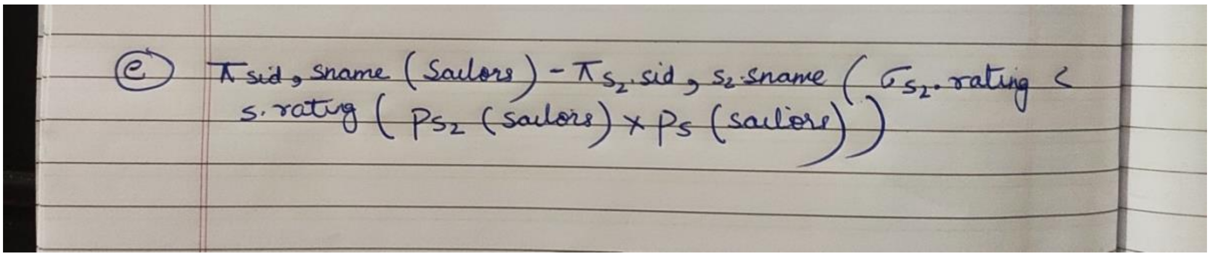


(d) π_{sid} (Sailors) $-$ π_{sid} (Sailors ∞ ($\sigma_{date-of-birth > Jan 2018}$ (Reserves))))

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

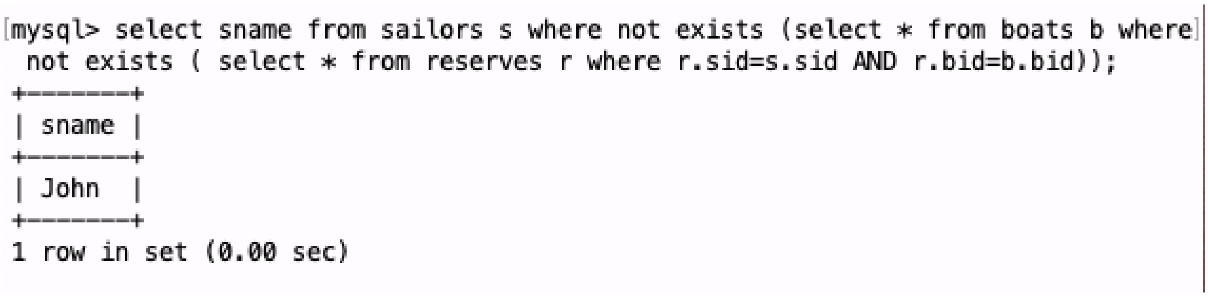


```
mysql> select sname from sailors where rating > all (select rating from sailors
where sname="John");
+----------+
| sname    |
+----------+
| Rusty    |
| Horatio  |
| Zorba    |
| Julius   |
+----------+
```

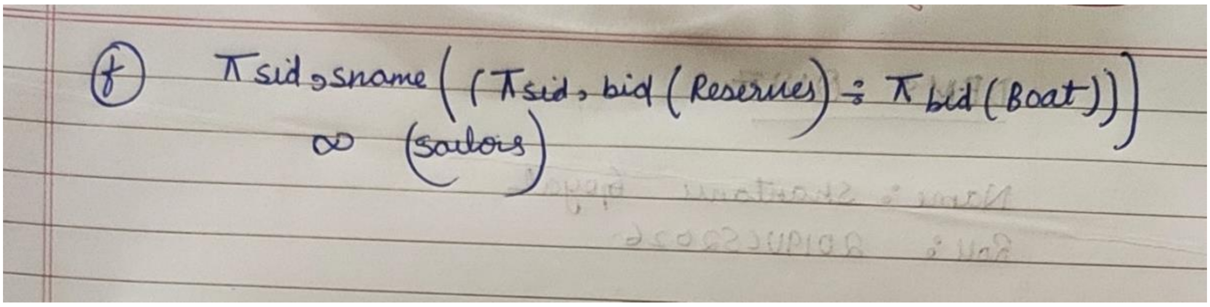


(e) $\pi_{sid, sname}$ (Sailors) $-$ $\pi_{S_2.sid, S_2.sname}$ ($\sigma_{S_2.rating < s.rating}$ (ρ_{S_2} (sailors) $\times$ ρ_{S} (sailors)))

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));

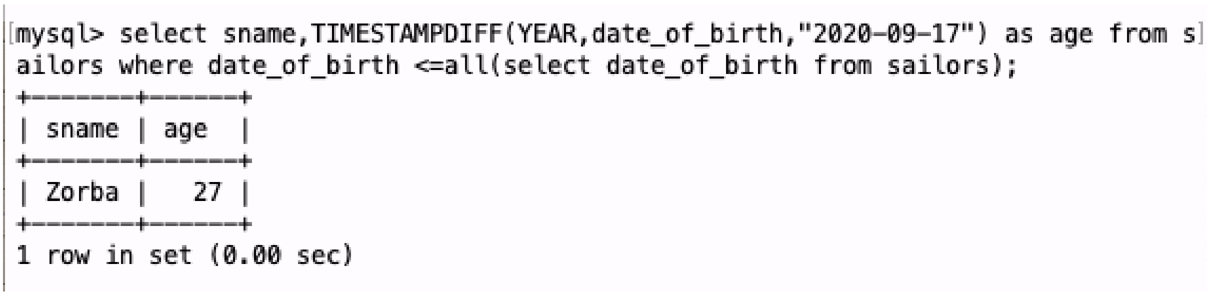


```
mysql> select sname from sailors s where not exists (select * from boats b where
 not exists ( select * from reserves r where r.sid=s.sid AND r.bid=b.bid));
+-------+
| sname |
+-------+
| John  |
+-------+
1 row in set (0.00 sec)
```

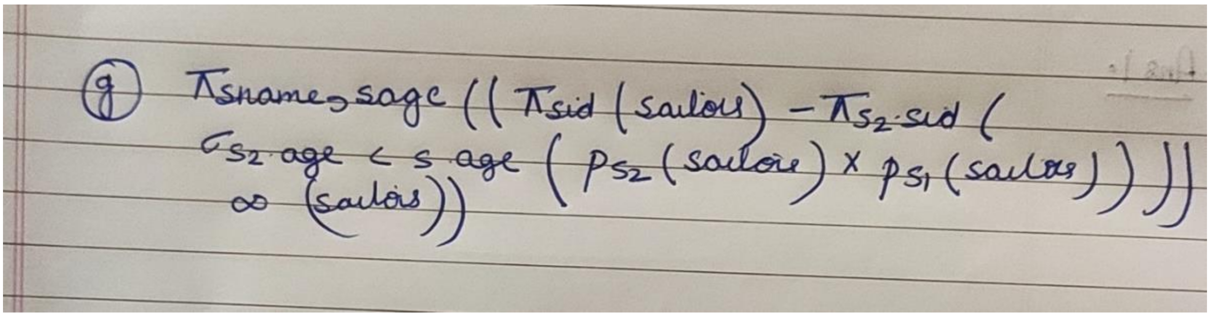


(f) $\pi_{sid, sname}\left(\left(\pi_{sid, bid}(Reserves) \div \pi_{bid}(Boat) \right) \right) \infty \; (sailors))$

g) select sname,TIMESTAMPDIFF(YEAR,date_of_birth,"2020-09-17") as age from sailors where date_of_birth <=all(select date_of_birth from sailors);

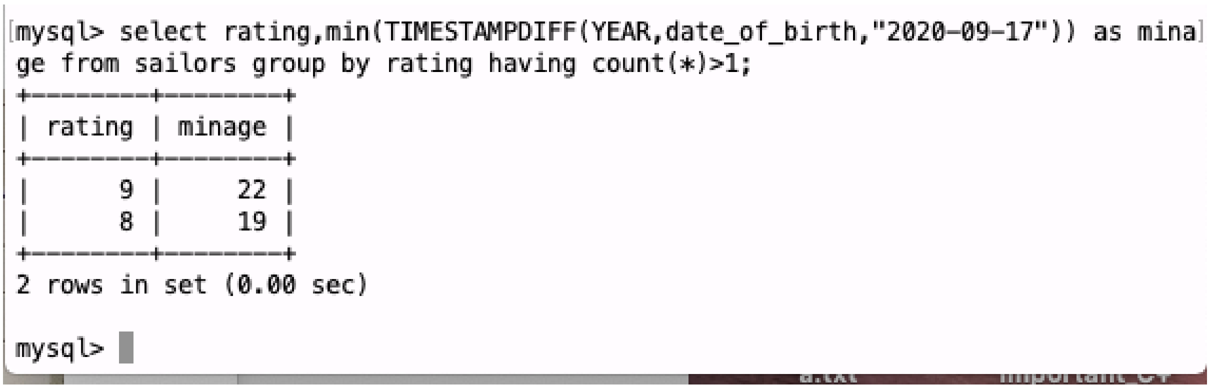


```
mysql> select sname,TIMESTAMPDIFF(YEAR,date_of_birth,"2020-09-17") as age from s
ailors where date_of_birth <=all(select date_of_birth from sailors);
+-------+------+
| sname | age  |
+-------+------+
| Zorba |   27 |
+-------+------+
1 row in set (0.00 sec)
```

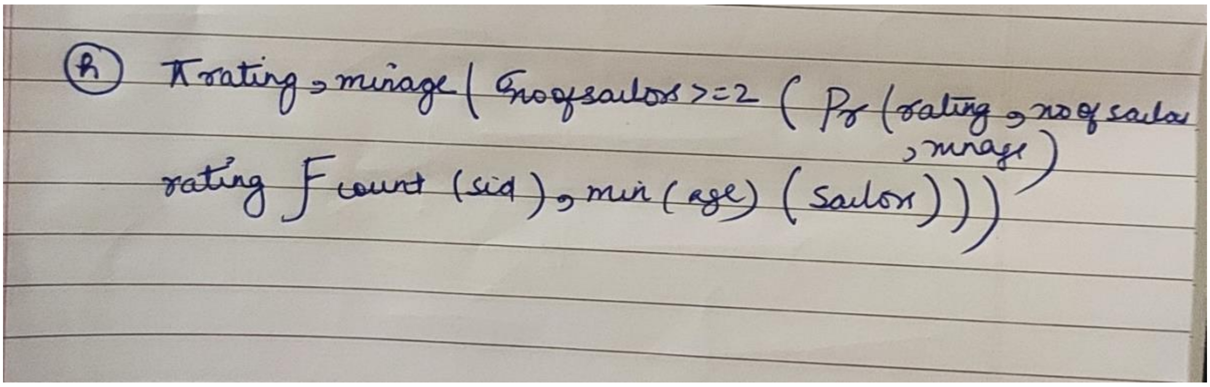


(g) $\pi_{sname, sage}\left(\left(\pi_{sid}(sailors) - \pi_{s2.sid}\left(\sigma_{s2.age < s.age}\left(\rho_{s2}(sailore) \times \rho_{s1}(sailors) \right) \right) \right) \infty \; (sailors) \right)$

h) select rating,min(TIMESTAMPDIFF(YEAR,date_of_birth,"2020-09-17")) as minage from sailors group by rating having count(*)>1;



```
mysql> select rating,min(TIMESTAMPDIFF(YEAR,date_of_birth,"2020-09-17")) as mina
ge from sailors group by rating having count(*)>1;
+--------+--------+
| rating | minage |
+--------+--------+
|      9 |     22 |
|      8 |     19 |
+--------+--------+
2 rows in set (0.00 sec)

mysql> 
```



(h) $\pi_{rating, minage}(\sigma_{noofsailors>=2}(\rho_r(rating, no of sailors, minage)({}_{rating}\mathcal{F}_{count(sid), min(age)}(Sailors))))$

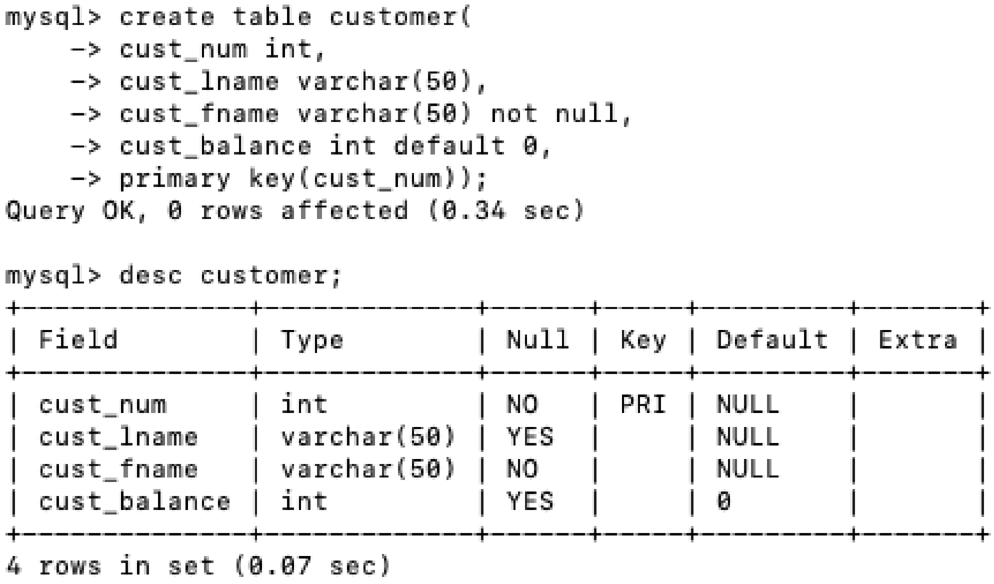
Q2. Consider the following relational schema:

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

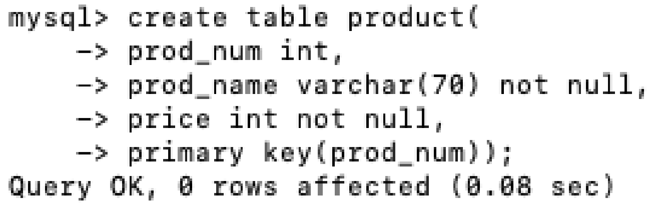
A2:
Creating tables

Customer table

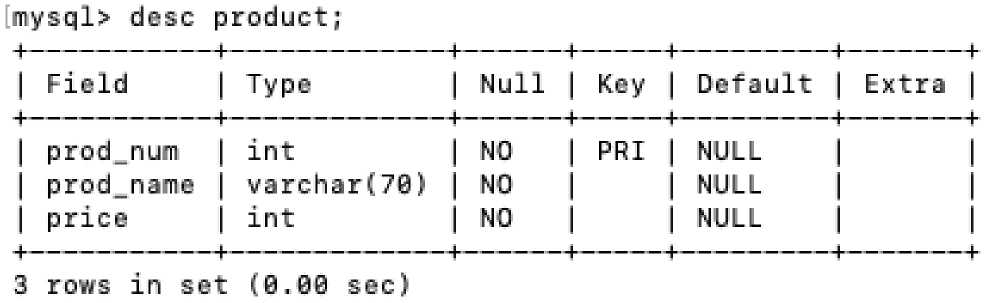


```
mysql> create table customer(
    -> cust_num int,
    -> cust_lname varchar(50),
    -> cust_fname varchar(50) not null,
    -> cust_balance int default 0,
    -> primary key(cust_num));
Query OK, 0 rows affected (0.34 sec)

mysql> desc customer;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| cust_num     | int         | NO   | PRI | NULL    |       |
| cust_lname   | varchar(50) | YES  |     | NULL    |       |
| cust_fname   | varchar(50) | NO   |     | NULL    |       |
| cust_balance | int         | YES  |     | 0       |       |
+--------------+-------------+------+-----+---------+-------+
4 rows in set (0.07 sec)
```

Product table

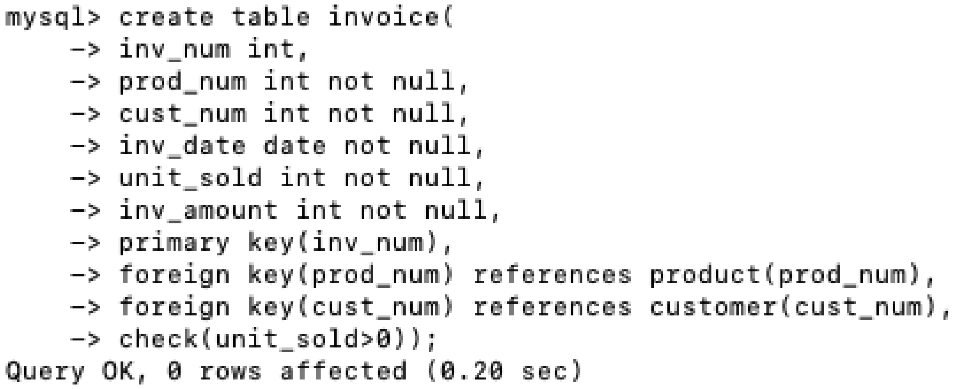


```
mysql> create table product(
    -> prod_num int,
    -> prod_name varchar(70) not null,
    -> price int not null,
    -> primary key(prod_num));
Query OK, 0 rows affected (0.08 sec)
```

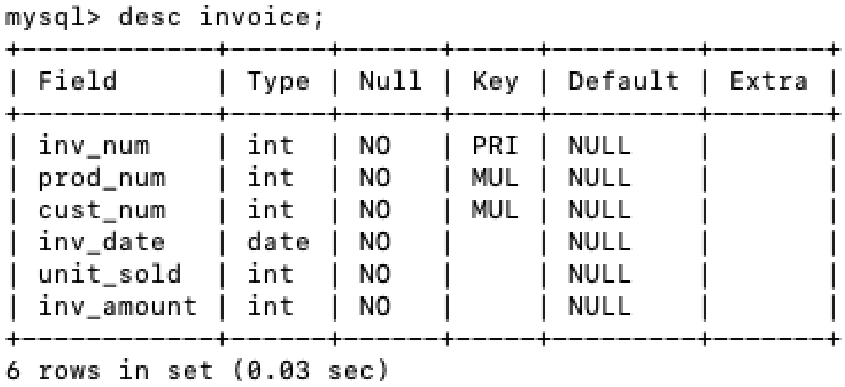


```
mysql> desc product;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| prod_num  | int         | NO   | PRI | NULL    |       |
| prod_name | varchar(70) | NO   |     | NULL    |       |
| price     | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Invoice Table

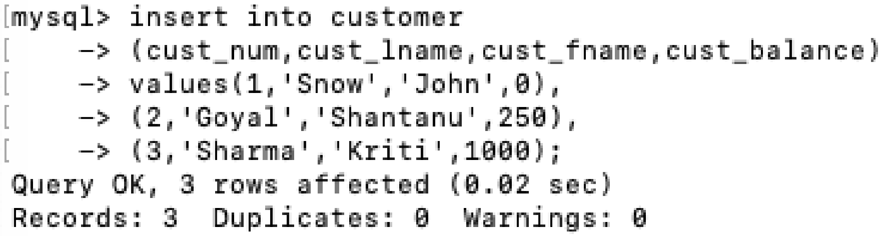


```
mysql> create table invoice(
    -> inv_num int,
    -> prod_num int not null,
    -> cust_num int not null,
    -> inv_date date not null,
    -> unit_sold int not null,
    -> inv_amount int not null,
    -> primary key(inv_num),
    -> foreign key(prod_num) references product(prod_num),
    -> foreign key(cust_num) references customer(cust_num),
    -> check(unit_sold>0));
Query OK, 0 rows affected (0.20 sec)
```

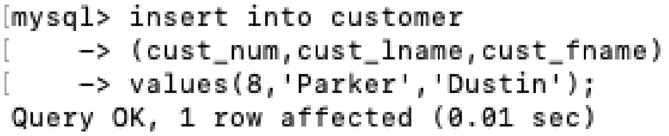


```
mysql> desc invoice;
+------------+------+------+-----+---------+-------+
| Field      | Type | Null | Key | Default | Extra |
+------------+------+------+-----+---------+-------+
| inv_num    | int  | NO   | PRI | NULL    |       |
| prod_num   | int  | NO   | MUL | NULL    |       |
| cust_num   | int  | NO   | MUL | NULL    |       |
| inv_date   | date | NO   |     | NULL    |       |
| unit_sold  | int  | NO   |     | NULL    |       |
| inv_amount | int  | NO   |     | NULL    |       |
+------------+------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

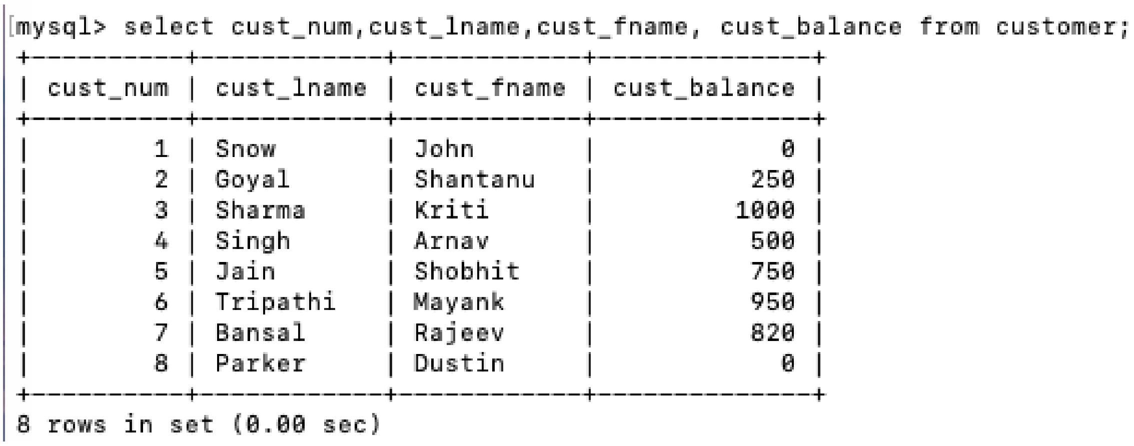
Insert into customer table



```
[mysql> insert into customer
[    -> (cust_num,cust_lname,cust_fname,cust_balance)
[    -> values(1,'Snow','John',0),
[    -> (2,'Goyal','Shantanu',250),
[    -> (3,'Sharma','Kriti',1000);
Query OK, 3 rows affected (0.02 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

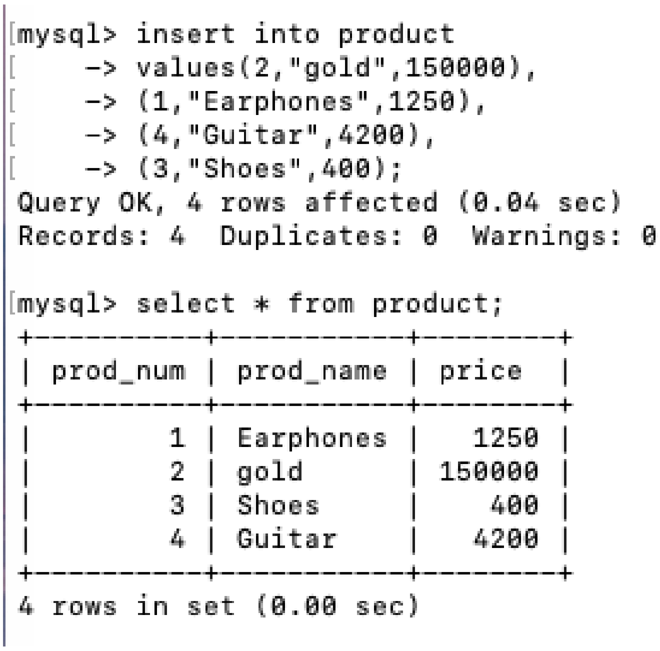


```
[mysql> insert into customer
[    -> (cust_num,cust_lname,cust_fname)
[    -> values(8,'Parker','Dustin');
Query OK, 1 row affected (0.01 sec)
```



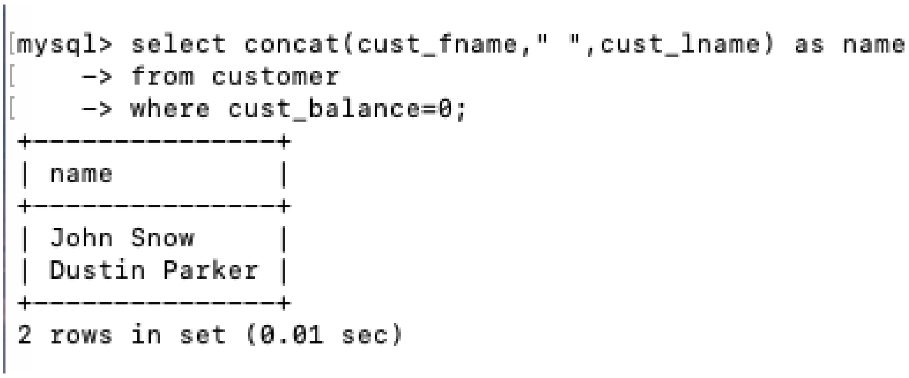
```
[mysql> select cust_num,cust_lname,cust_fname, cust_balance from customer;
+----------+------------+------------+--------------+
| cust_num | cust_lname | cust_fname | cust_balance |
+----------+------------+------------+--------------+
|        1 | Snow       | John       |            0 |
|        2 | Goyal      | Shantanu   |          250 |
|        3 | Sharma     | Kriti      |         1000 |
|        4 | Singh      | Arnav      |          500 |
|        5 | Jain       | Shobhit    |          750 |
|        6 | Tripathi   | Mayank     |          950 |
|        7 | Bansal     | Rajeev     |          820 |
|        8 | Parker     | Dustin     |            0 |
+----------+------------+------------+--------------+
8 rows in set (0.00 sec)
```

Insert into product table

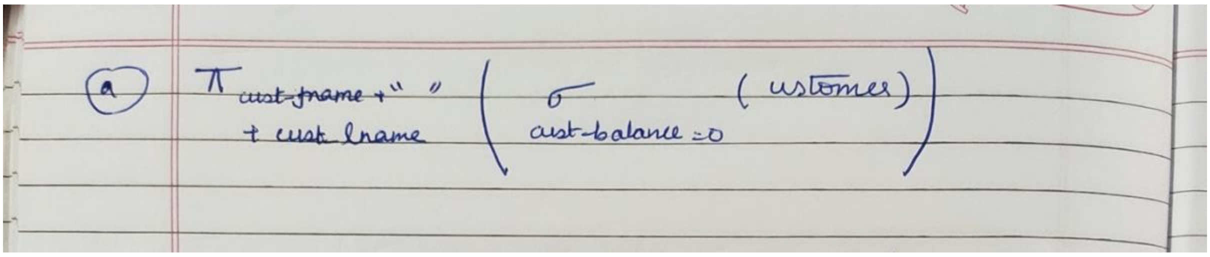


```
[mysql> insert into product
[    -> values(2,"gold",150000),
[    -> (1,"Earphones",1250),
[    -> (4,"Guitar",4200),
[    -> (3,"Shoes",400);
Query OK, 4 rows affected (0.04 sec)
Records: 4  Duplicates: 0  Warnings: 0

[mysql> select * from product;
+----------+-----------+--------+
| prod_num | prod_name | price  |
+----------+-----------+--------+
|        1 | Earphones |   1250 |
|        2 | gold      | 150000 |
|        3 | Shoes     |    400 |
|        4 | Guitar    |   4200 |
+----------+-----------+--------+
4 rows in set (0.00 sec)
```

(a) Find the names of the customer who have purchased no item. Set default value of Cust_balance as 0 for such customers.

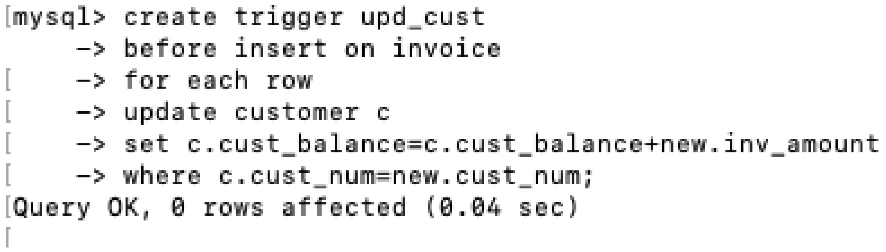


```
mysql> select concat(cust_fname," ",cust_lname) as name
    -> from customer
    -> where cust_balance=0;
+----------------+
| name           |
+----------------+
| John Snow      |
| Dustin Parker  |
+----------------+
2 rows in set (0.01 sec)
```



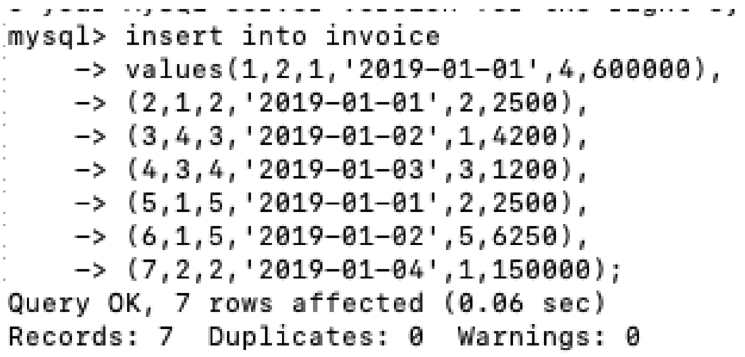
(a) $\pi_{\text{cust-fname} + \text{" "} + \text{cust-lname}} \left(\sigma_{\text{cust-balance} = 0} (\text{ustomer}) \right)$

(b) Write the trigger to update the CUST_BALANCE in the CUSTOMER table when a new invoice record is entered for the customer.

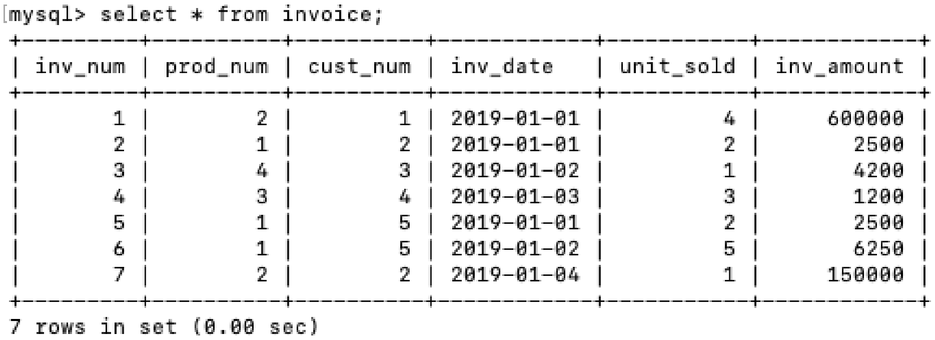


```
[mysql> create trigger upd_cust
     -> before insert on invoice
[    -> for each row
[    -> update customer c
[    -> set c.cust_balance=c.cust_balance+new.inv_amount
[    -> where c.cust_num=new.cust_num;
[Query OK, 0 rows affected (0.04 sec)
[
```

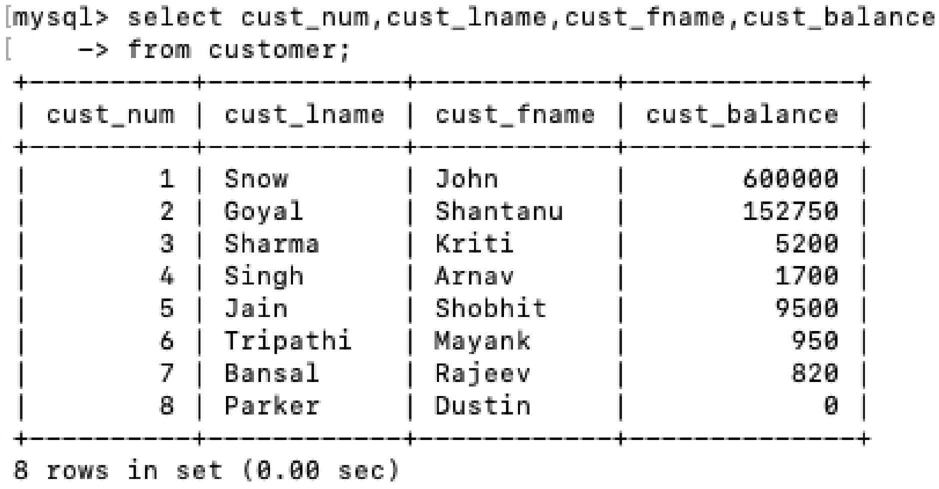
Created the trigger which would update values in customer table



```
mysql> insert into invoice
    -> values(1,2,1,'2019-01-01',4,600000),
    -> (2,1,2,'2019-01-01',2,2500),
    -> (3,4,3,'2019-01-02',1,4200),
    -> (4,3,4,'2019-01-03',3,1200),
    -> (5,1,5,'2019-01-01',2,2500),
    -> (6,1,5,'2019-01-02',5,6250),
    -> (7,2,2,'2019-01-04',1,150000);
Query OK, 7 rows affected (0.06 sec)
Records: 7  Duplicates: 0  Warnings: 0
```



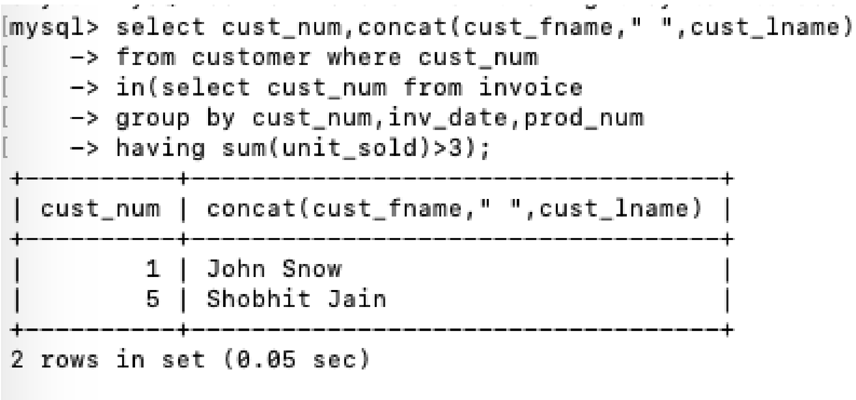
```
[mysql> select * from invoice;
+---------+----------+----------+------------+-----------+------------+
| inv_num | prod_num | cust_num | inv_date   | unit_sold | inv_amount |
+---------+----------+----------+------------+-----------+------------+
|       1 |        2 |        1 | 2019-01-01 |         4 |     600000 |
|       2 |        1 |        2 | 2019-01-01 |         2 |       2500 |
|       3 |        4 |        3 | 2019-01-02 |         1 |       4200 |
|       4 |        3 |        4 | 2019-01-03 |         3 |       1200 |
|       5 |        1 |        5 | 2019-01-01 |         2 |       2500 |
|       6 |        1 |        5 | 2019-01-02 |         5 |       6250 |
|       7 |        2 |        2 | 2019-01-04 |         1 |     150000 |
+---------+----------+----------+------------+-----------+------------+
7 rows in set (0.00 sec)
```



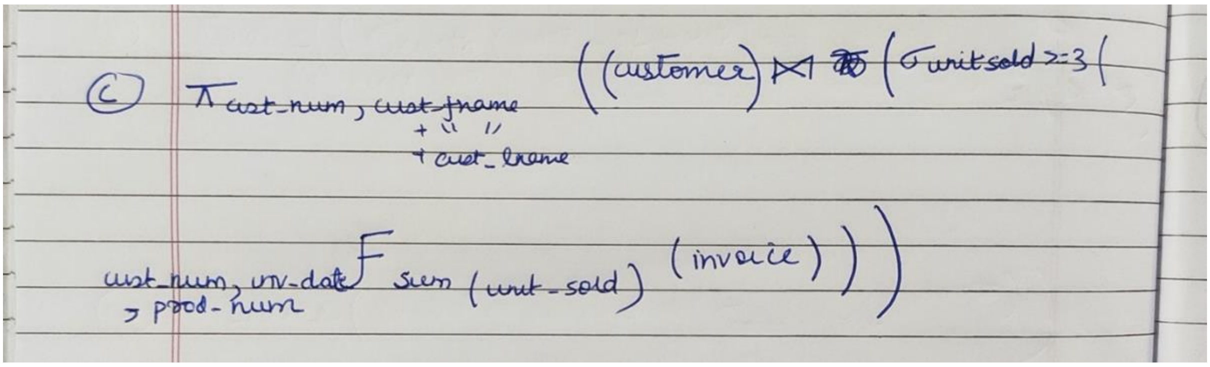
```
[mysql> select cust_num,cust_lname,cust_fname,cust_balance
[    -> from customer;
+----------+------------+------------+--------------+
| cust_num | cust_lname | cust_fname | cust_balance |
+----------+------------+------------+--------------+
|        1 | Snow       | John       |       600000 |
|        2 | Goyal      | Shantanu   |       152750 |
|        3 | Sharma     | Kriti      |         5200 |
|        4 | Singh      | Arnav      |         1700 |
|        5 | Jain       | Shobhit    |         9500 |
|        6 | Tripathi   | Mayank     |          950 |
|        7 | Bansal     | Rajeev     |          820 |
|        8 | Parker     | Dustin     |            0 |
+----------+------------+------------+--------------+
8 rows in set (0.00 sec)
```

We can see that the cust_balance in the customer table is updated

(c) Find the customers who have purchased more than three units of a product on a day

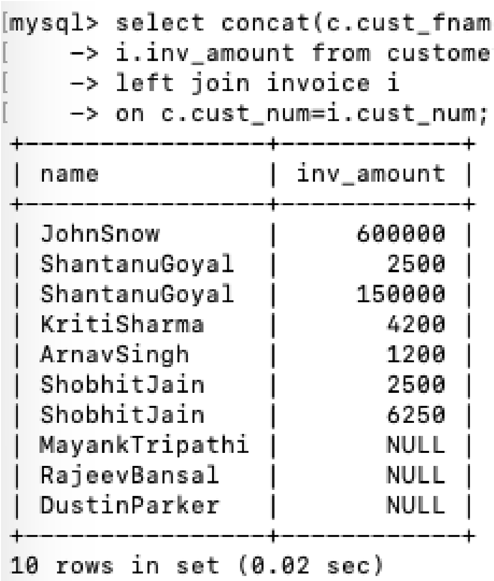


```
mysql> select cust_num,concat(cust_fname," ",cust_lname)
    -> from customer where cust_num
    -> in(select cust_num from invoice
    -> group by cust_num,inv_date,prod_num
    -> having sum(unit_sold)>3);
+----------+-----------------------------------+
| cust_num | concat(cust_fname," ",cust_lname) |
+----------+-----------------------------------+
|        1 | John Snow                         |
|        5 | Shobhit Jain                      |
+----------+-----------------------------------+
2 rows in set (0.05 sec)
```

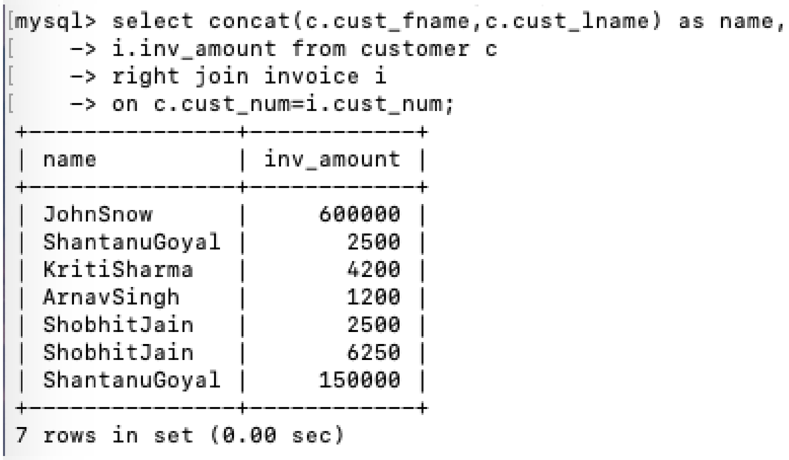


(c) $\pi_{cust\_num,\ cust\_fname + " " + cust\_lname}((customer) \bowtie (\sigma_{unitsold >= 3}({}_{cust\_num,\ inv\_date,\ prod\_num}\mathcal{F}_{sum(unit\_sold)}(invoice))))$

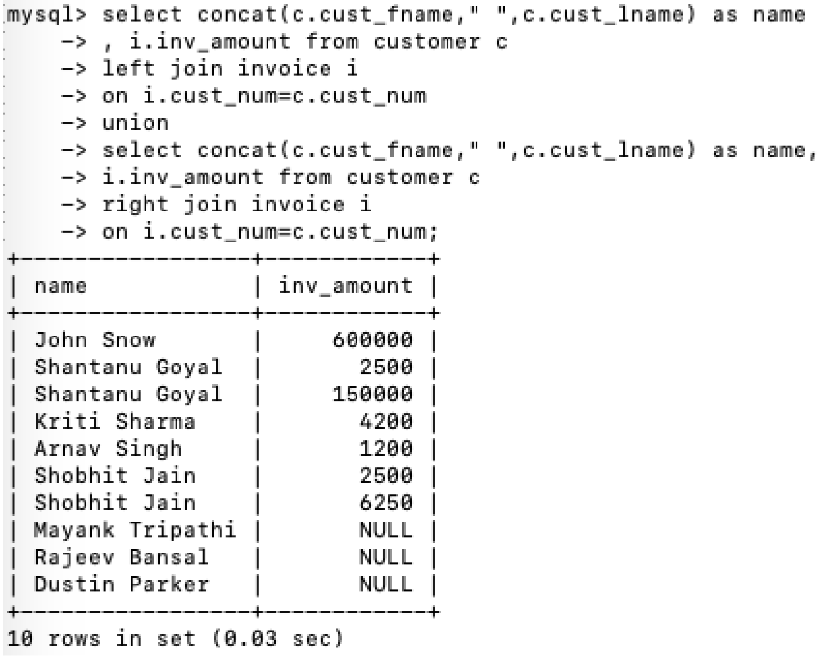
(d) Write a query to illustrate Left Outer, Right Outer and Full Outer Join.



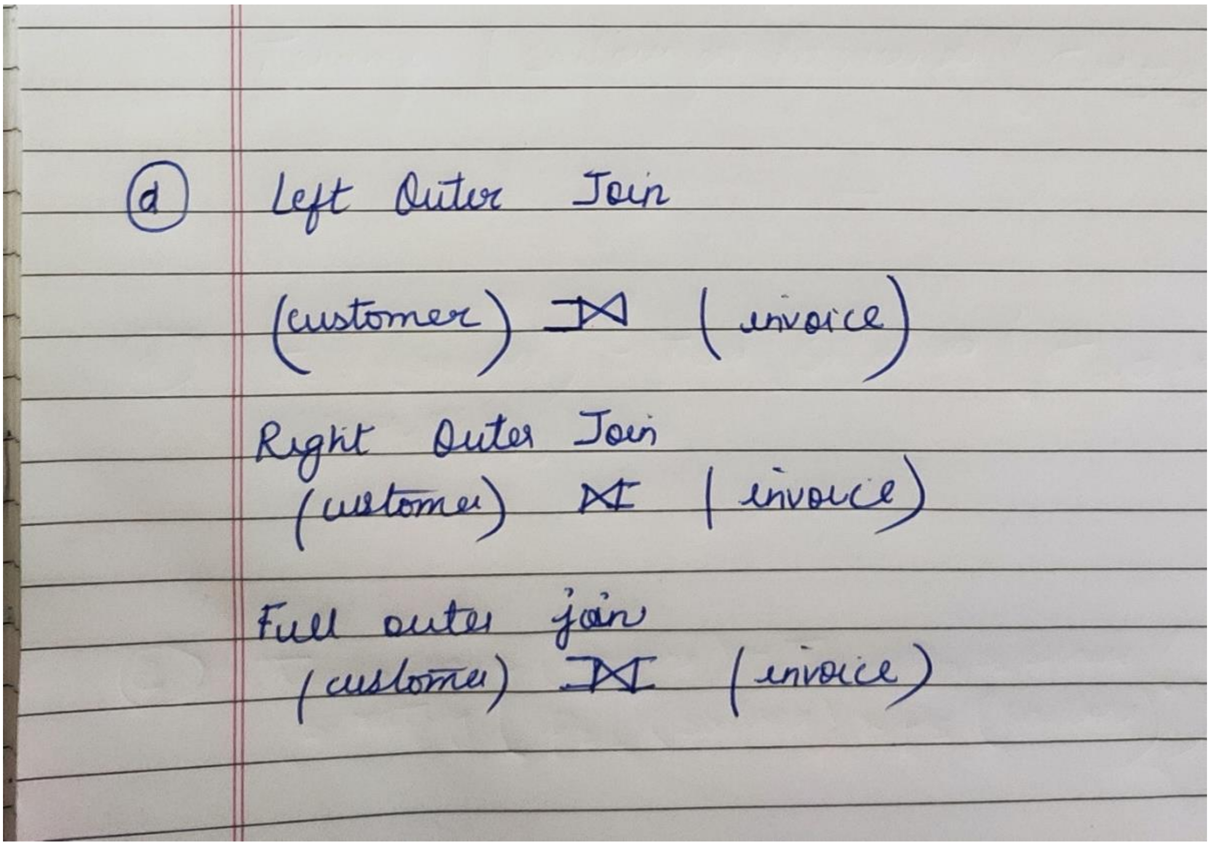
```
[mysql> select concat(c.cust_fname,c.cust_lname) as name,
[    -> i.inv_amount from customer c
[    -> left join invoice i
[    -> on c.cust_num=i.cust_num;
+-----------------+------------+
| name            | inv_amount |
+-----------------+------------+
| JohnSnow        |     600000 |
| ShantanuGoyal   |       2500 |
| ShantanuGoyal   |     150000 |
| KritiSharma     |       4200 |
| ArnavSingh      |       1200 |
| ShobhitJain     |       2500 |
| ShobhitJain     |       6250 |
| MayankTripathi  |       NULL |
| RajeevBansal    |       NULL |
| DustinParker    |       NULL |
+-----------------+------------+
10 rows in set (0.02 sec)
```



```
[mysql> select concat(c.cust_fname,c.cust_lname) as name,
[    -> i.inv_amount from customer c
[    -> right join invoice i
[    -> on c.cust_num=i.cust_num;
+---------------+------------+
| name          | inv_amount |
+---------------+------------+
| JohnSnow      |     600000 |
| ShantanuGoyal |       2500 |
| KritiSharma   |       4200 |
| ArnavSingh    |       1200 |
| ShobhitJain   |       2500 |
| ShobhitJain   |       6250 |
| ShantanuGoyal |     150000 |
+---------------+------------+
7 rows in set (0.00 sec)
```



```
mysql> select concat(c.cust_fname," ",c.cust_lname) as name
    -> , i.inv_amount from customer c
    -> left join invoice i
    -> on i.cust_num=c.cust_num
    -> union
    -> select concat(c.cust_fname," ",c.cust_lname) as name,
    -> i.inv_amount from customer c
    -> right join invoice i
    -> on i.cust_num=c.cust_num;
+-----------------+------------+
| name            | inv_amount |
+-----------------+------------+
| John Snow       |     600000 |
| Shantanu Goyal  |       2500 |
| Shantanu Goyal  |     150000 |
| Kriti Sharma    |       4200 |
| Arnav Singh     |       1200 |
| Shobhit Jain    |       2500 |
| Shobhit Jain    |       6250 |
| Mayank Tripathi |       NULL |
| Rajeev Bansal   |       NULL |
| Dustin Parker   |       NULL |
+-----------------+------------+
10 rows in set (0.03 sec)
```



ⓓ Left Outer Join

(customer) ⟕ (invoice)

Right Outer Join

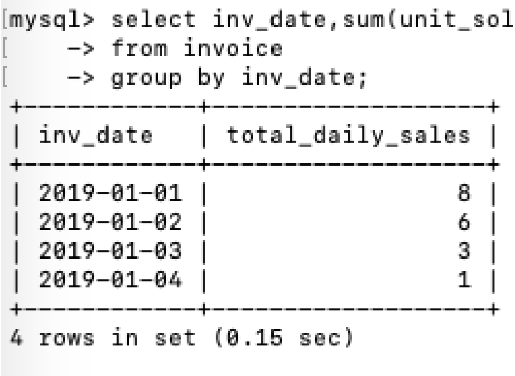
(customer) ⟖ (invoice)

Full outer join

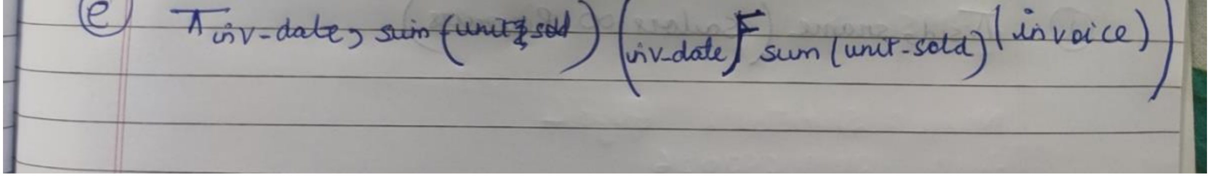
(customer) ⟗ (invoice)

Left outer join, right outer join, full outer join in order between tables customer and invoice.

(e) Count number of products sold on each date.



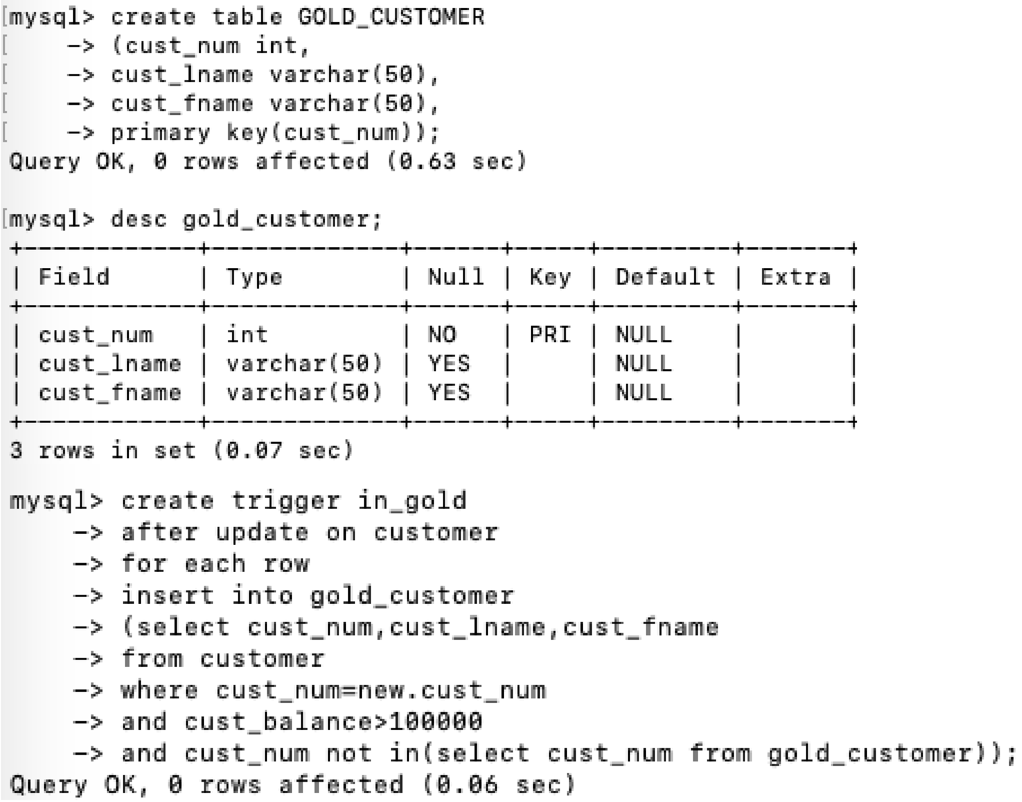
```
mysql> select inv_date,sum(unit_sold) as total_daily_sales
    -> from invoice
    -> group by inv_date;
+------------+-------------------+
| inv_date   | total_daily_sales |
+------------+-------------------+
| 2019-01-01 |                 8 |
| 2019-01-02 |                 6 |
| 2019-01-03 |                 3 |
| 2019-01-04 |                 1 |
+------------+-------------------+
4 rows in set (0.15 sec)
```



Date
Page

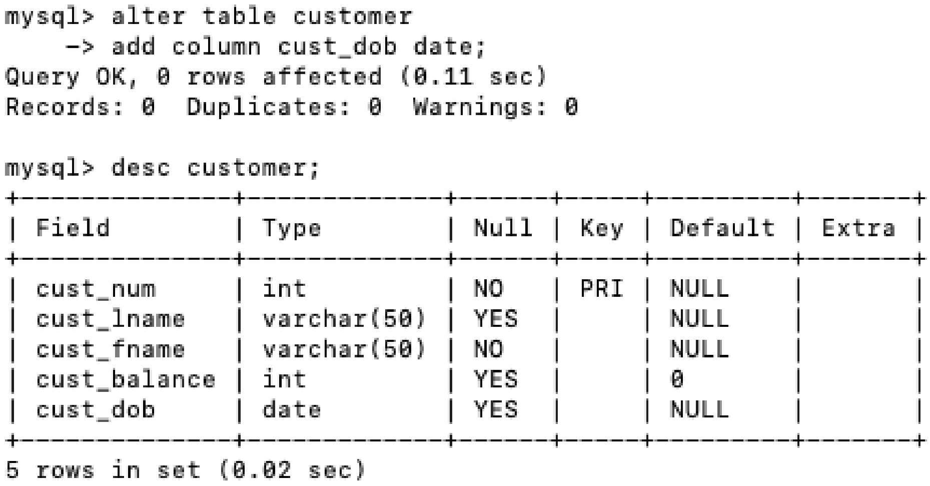
ⓔ $\pi_{inv\text{-}date,\ sum(unit\_sold)} \left({}_{inv\text{-}date}\mathcal{F}_{sum(unit\text{-}sold)} (invoice) \right)$

(f) As soon as customer balance becomes greater than Rs. 100,000, copy the customer_num in new table called ”GOLD_CUSTOMER”



```
[mysql> create table GOLD_CUSTOMER
[    -> (cust_num int,
[    -> cust_lname varchar(50),
[    -> cust_fname varchar(50),
[    -> primary key(cust_num));
Query OK, 0 rows affected (0.63 sec)

[mysql> desc gold_customer;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| cust_num   | int         | NO   | PRI | NULL    |       |
| cust_lname | varchar(50) | YES  |     | NULL    |       |
| cust_fname | varchar(50) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.07 sec)

mysql> 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));
Query OK, 0 rows affected (0.06 sec)
```

(g) Add a new attribute CUST_DOB in customer table



```
mysql> alter table customer
    -> add column cust_dob date;
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc customer;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| cust_num     | int         | NO   | PRI | NULL    |       |
| cust_lname   | varchar(50) | YES  |     | NULL    |       |
| cust_fname   | varchar(50) | NO   |     | NULL    |       |
| cust_balance | int         | YES  |     | 0       |       |
| cust_dob     | date        | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
5 rows in set (0.02 sec)
```