DBMS PRACTICAL-7

PRE-LAB

1) What is the output of following query
SELECT A.id FROM A WHERE A.age > ALL (SELECT B.age FROM B WHERE B. name = "ABC")

DBMS lab-7

190031187
Radhakrishna

prelab

1. SELECT A.id FROM A WHERE A.age > All (SELECT B.age FROM B WHERE B.name = "ABC")

The output of the above query will be

Id's of person whose age is greater than "ABC"

2) What is the purpose of WHERE Clause in mysql?

2. WHERE

The WHERE clause is used to filter records. The WHERE clause is used to extract only those records that fullfill a specified condition.

3) Differentiate between AND, OR and NOT operators in mysql

3. AND :- AND operator is used to set multiple conditions with the WHERE clause, alongside SELECT, UPDATE or DELETE

Ex:- SELECT * from emp WHERE sal<100000 AND age > 25

OR :- OR operator is also used to ~~display~~ filter records based on more than one condition. The OR operator displays record if any of the conditions seperated by OR is true

NOT :-

NOT operator displays record if the condition is NOT TRUE

5) Database table by name Loan_Records is given below.

Borrower	Bank_Manager	Loan_Amount
Ramesh	Sunderajan	10000.00
Suresh	Ramgopal	5000.00
Mahesh	Sunderajan	7000.00

SELECT Count(*) FROM ((SELECT Borrower, Bank_Manager FROM Loan_Records) AS S NATURAL JOIN (SELECT Bank_Manager, Loan_Amount FROM Loan_Records) AS T);

What is the output of the following SQL query?

5. SELECT count(*) FROM ((SELECT Borrower, Bank_Manager FROM Loan_Records) AS S NATURAL JOIN (SELECT Bank_Manager, Loan_Amount FROM Loan_Records) AS T);

output for the above query is

5

i State the difference between (Null Value Function) nvl() & Coalesce().

NVL and COALESCE are used to acheive the same functionality of providing a default value in case the column returns a null

NVL accepts only 2 Arguments whereas COALESCE can take multiple arguments

4) Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X=1, Y=1) is inserted in the table. Let MX and My denote the respective maximum values of X and Y among all records in the table at any point in time. Using MX and MY, new records are inserted in the table 128 times with X and Y values being MX+1, 2*MY+1 respectively. It may be noted that each time after the insertion, values of MX and MY change. What will be the output of the following SQL query after the steps mentioned above are carried out? [) Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X=1, Y=1) is inserted in the table. Let MX and My denote the respective maximum values of X and Y among all records in the table at any point in time. Using MX and MY, new records are inserted in the table 128 times with X and Y values being MX+1, 2*MY+1 respectively. It may be noted that each time after the insertion, values of MX and MY change. What will be the output of the following SQL query after the steps mentioned above are carried out?
SELECT Y FROM T WHERE X=7;

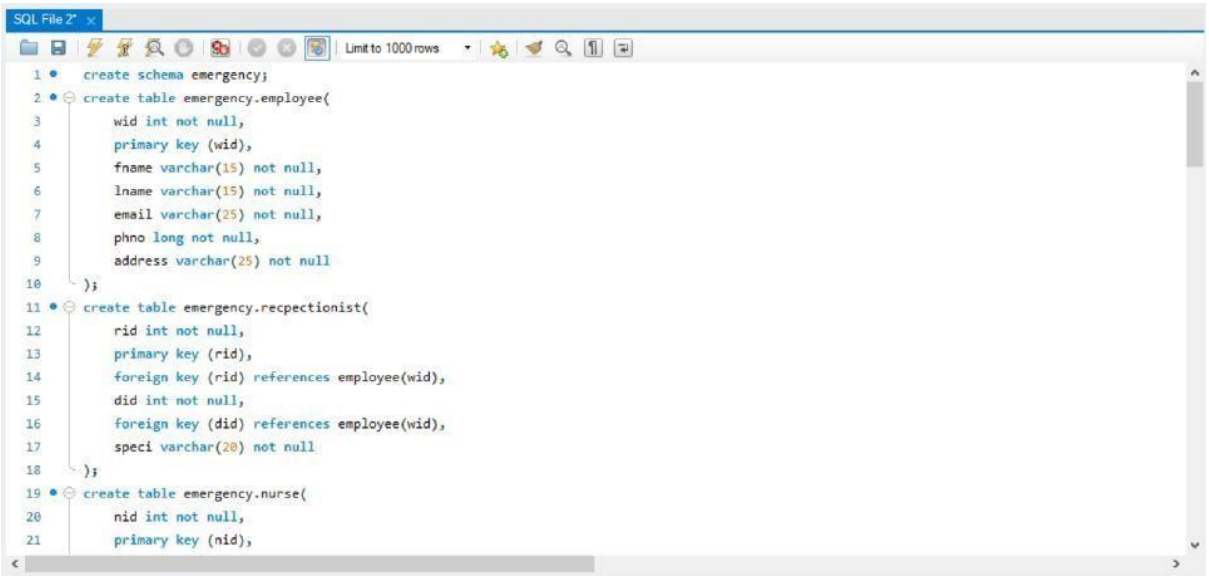
4.

X	Y
1	1
2	3
3	7
4	15
5	31
6	63
7	127
⋮	⋮

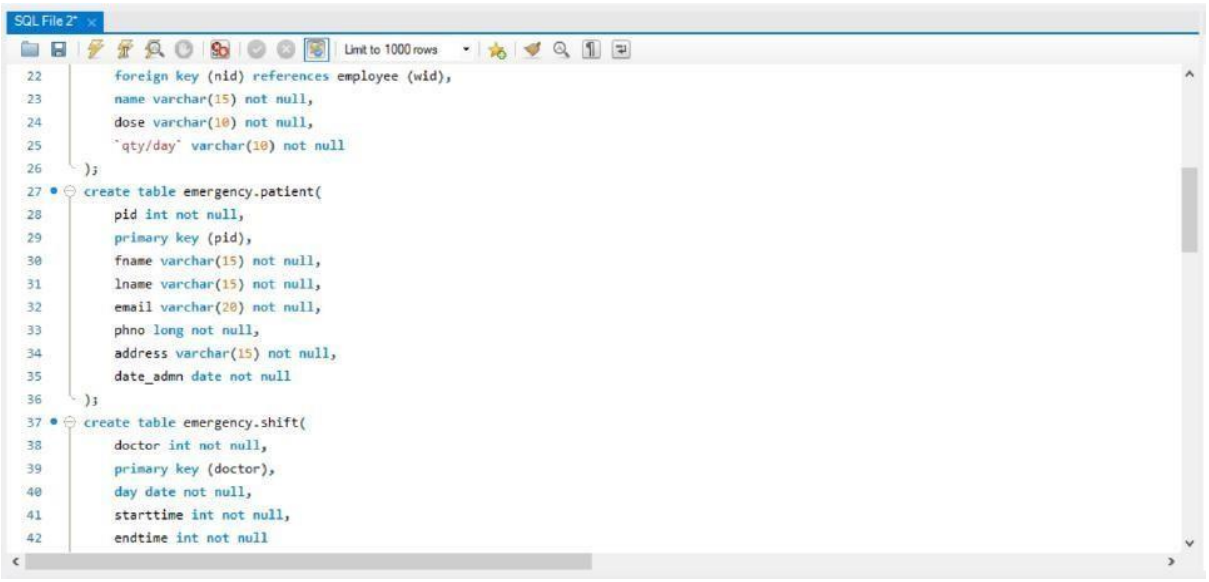
Answ is 127

Case Study 2 :EMERGENCY ROOM INFORMATION SYSTEM

1) Create the database in MySQL and create the necessary tables for the given case study using appropriate keys and relationships between the tables



```
create schema emergency;
create table emergency.employee(
    wid int not null,
    primary key (wid),
    fname varchar(15) not null,
    lname varchar(15) not null,
    email varchar(25) not null,
    phno long not null,
    address varchar(25) not null
);
create table emergency.recpectionist(
    rid int not null,
    primary key (rid),
    foreign key (rid) references employee(wid),
    did int not null,
    foreign key (did) references employee(wid),
    speci varchar(20) not null
);
create table emergency.nurse(
    nid int not null,
    primary key (nid),
```



```
    foreign key (nid) references employee (wid),
    name varchar(15) not null,
    dose varchar(10) not null,
    `qty/day` varchar(10) not null
);
create table emergency.patient(
    pid int not null,
    primary key (pid),
    fname varchar(15) not null,
    lname varchar(15) not null,
    email varchar(20) not null,
    phno long not null,
    address varchar(15) not null,
    date_admn date not null
);
create table emergency.shift(
    doctor int not null,
    primary key (doctor),
    day date not null,
    starttime int not null,
    endtime int not null
```

```
SQL File 2*
Limit to 1000 rows
43  );
44  create TABLE emergency.pcase(
45      bednum int not null,
46      primary key (bednum),
47      pid int not null,
48      foreign key (pid) references patient(pid)
49  );
50  create table emergency.appointment(
51      pid int not null,
52      foreign key (pid) references patient(pid),
53      amt int not null
54  );
55  insert into emergency.employee
56  values(100,'Arun','Kumar','abcd@gmail.com',9988776655,'Hyderabad'),
57        (101,'David','Raju','ghij@gmail.com',9987634567,'Secunderabad'),
58        (102,'Harish','Reddy','klmn@gmail.com',9986492479,'Bowenpally'),
59        (103,'John','Samuel','pqrt@gmail.com',9985350391,'Hyderabad'),
60        (104,'Kira','Kumar','ghjk@gmail.com',9984208303,'Amaravathi'),
61        (105,'Aravind','Babu','ghhjk@gmail.com',9983066215,'Guntur'),
62        (106,'Praveen','Kumar','qwert@gmail.com',9981924127,'Tullur'),
63        (107,'Ramesh','Kumar','poiu@gmail.com',9980782039,'Mangalagiri'),
```

2) Insert atleast 10 records into every table that is implemented in the case study

```
SQL File 2*
Limit to 1000 rows
64        (108,'Jayanth','Kumar','poutu@gmail.com',9979639951,'Vijayawada'),
65        (109,'Eswar','Raju','vghj@gmail.com',9978497863,'Hyderabad');
66  insert into emergency.recpectionist
67  values(104,100,'Anesthecian'),
68        (106,101,'Denist'),
69        (108,102,'Cardiologist'),
70        (109,105,'Neurosurgeon');
71  insert into emergency.nurse
72  values(103,'bacd','once','2ml'),
73        (107,'xyz','twice','4ml'),
74        (109,'pqr','thrice','100mg');
75  insert into emergency.patient
76  values(200,'David','Samson','aaa@gmail.com',7654328976,'Mumbai','2019-10-13'),
77        (201,'Bharath','Kumar','bbb@gmail.com',8765438907,'Hyderabad','2020-01-22'),
78        (202,'Eswar','Prasad','ccc@gmail.com',9876548038,'Vijayawada','2020-03-15'),
79        (203,'Jaya','Laxmi','ddd@gmail.com',8765489765,'Delhi','2020-07-06'),
80        (204,'Laxmi','Devi','eee@gmail.com',7654430692,'Mumbai','2020-03-12'),
81        (205,'Pramod','Reddy','fff@gmail.com',6543371619,'Hyderabad','2020-02-05'),
82        (206,'Charan','Kumar','ggg@gmail.com',5432312546,'Vijayawada','2020-04-22'),
83        (207,'Kalyan','Reddy','hhh@gmail.com',4321253473,'Delhi','2020-07-10'),
84        (208,'Rajesh','Yadav','iklmn@gmail.com',3210194400,'Chennai','2020-03-03'),
```

```
SQL File 2*
Limit to 1000 rows
85        (209,'Naveen','Kumar','ighfr@gmail.com',2099135327,'Hyderabad','2020-04-22');
86  insert into emergency.pcase
87  values(20,200),
88        (21,201),
89        (22,202),
90        (23,203),
91        (24,204),
92        (25,205),
93        (26,206),
94        (27,207),
95        (28,208),
96        (29,209);
97  insert into emergency.appointment
98  values(200,200),
99        (201,200),
100       (202,300),
101       (203,600),
102       (204,200),
103       (205,300),
104       (206,200),
105       (207,200),
```

```
SQL File 2*
Limit to 1000 rows
        (25,205),
        (26,206),
        (27,207),
        (28,208),
        (29,209);
insert into emergency.appointment
values(200,200),
       (201,200),
       (202,300),
       (203,600),
       (204,200),
       (205,300),
       (206,200),
       (207,200),
       (208,200),
       (209,600);
insert into emergency.shift
values(100,'2020-04-12',9,5),
       (101,'2020-06-20',4,8),
       (102,'2020-08-01',6,12),
       (105,'2020-08-05',13,18);
```

3) Write an MYSQL query to add columns actiontaken and personell to the table case?

```
SQL File 6*   SQL File 7*   pcase
Limit to 1000 rows
alter table emergency.pcase
add column actiontaken varchar(15) not null,
add column personell varchar(25) not null;
SELECT * FROM emergency.pcase;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

bednum	pid	actiontaken	personell
20	200		
21	201		
22	202		
23	203		
24	204		
25	205		
26	206		
27	207		
28	208		
29	209		

pcase 1 | Apply | Revert

4) Write an MYSQL query to display who are patients required to stay in the hospital?

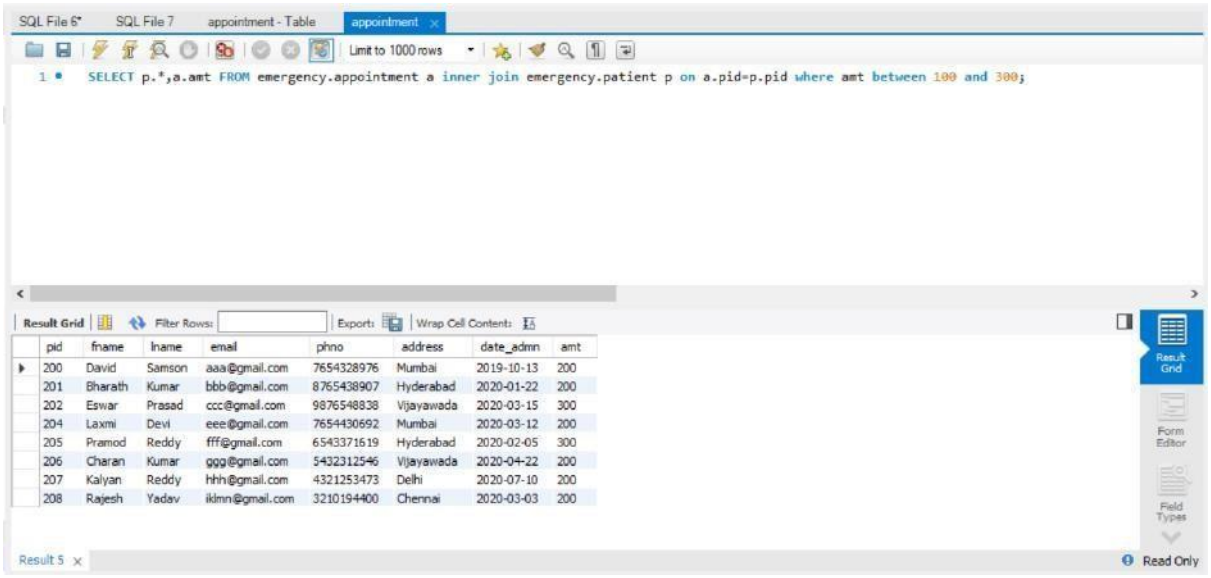
```
Query 1   pcase
Limit to 1000 rows
use emergency;
select distinct p.*,bednum from patient p,pcase c where p.pid=c.pid;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

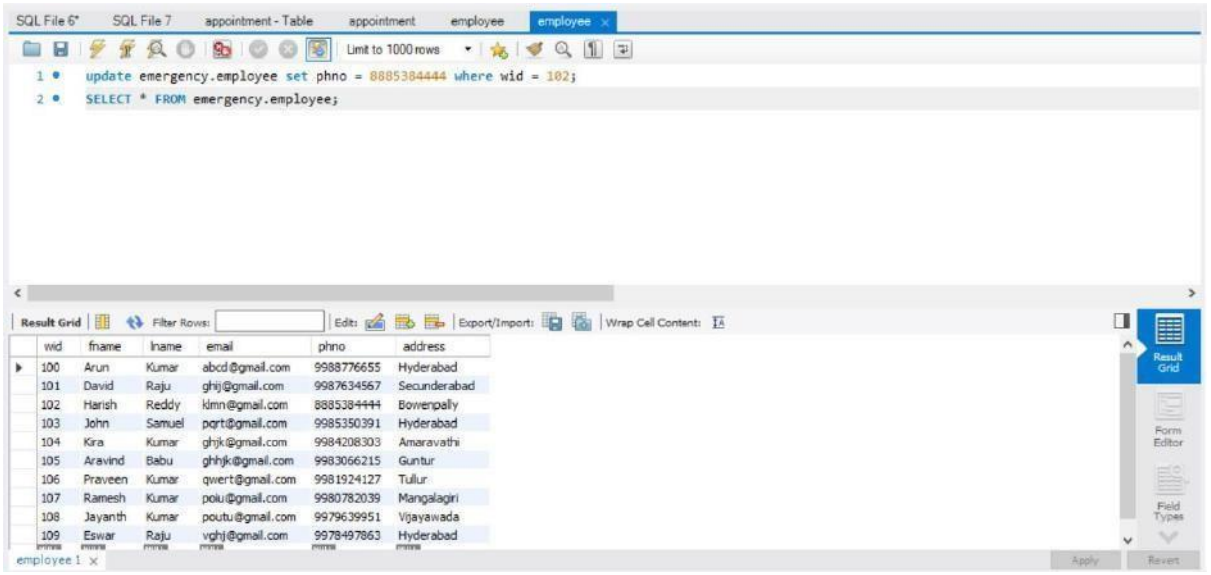
pid	fname	lname	email	phno	address	date_admn	bednum
200	David	Samson	aaa@gmail.com	7654328976	Mumbai	2019-10-13	20
201	Bharath	Kumar	bbb@gmail.com	8765438907	Hyderabad	2020-01-22	21
202	Eswar	Prasad	ccc@gmail.com	9876548838	Vijayawada	2020-03-15	22
203	Jaya	Laxmi	ddd@gmail.com	8765489765	Delhi	2020-07-06	23
204	Laxmi	Devi	eee@gmail.com	7654430692	Mumbai	2020-03-12	24
205	Pramod	Reddy	fff@gmail.com	6543371619	Hyderabad	2020-02-05	25
206	Charan	Kumar	ggg@gmail.com	5432312546	Vijayawada	2020-04-22	26
207	Kalyan	Reddy	hhh@gmail.com	4321253473	Delhi	2020-07-10	27
208	Rajesh	Yadav	iklmn@gmail.com	3210194400	Chennai	2020-03-03	28
209	Naveen	Kumar	ighfr@gmail.com	2099135327	Hyderabad	2020-04-22	29

Result 2 | Read Only

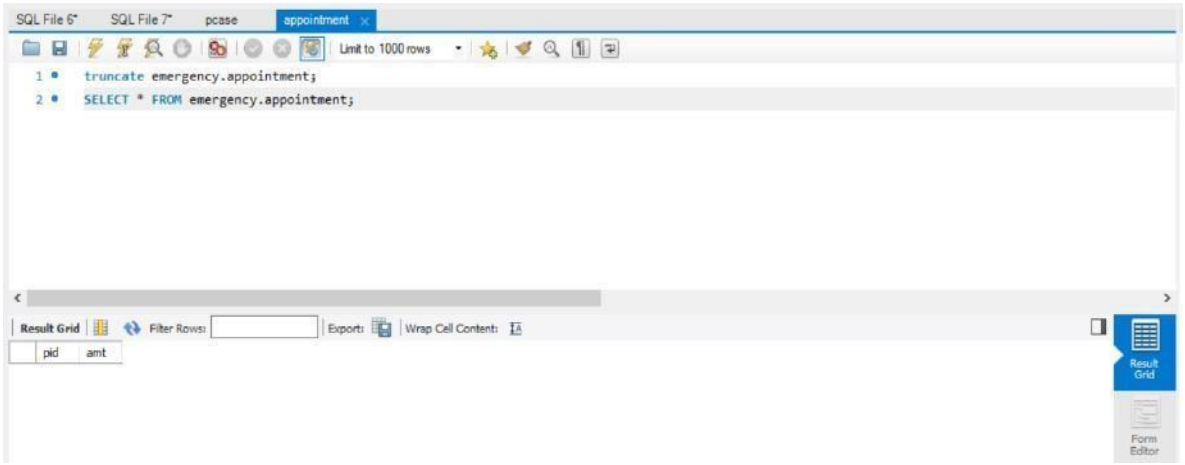
5) Write an MYSQL query to display who paid amount between 100 and 300 for the appointment



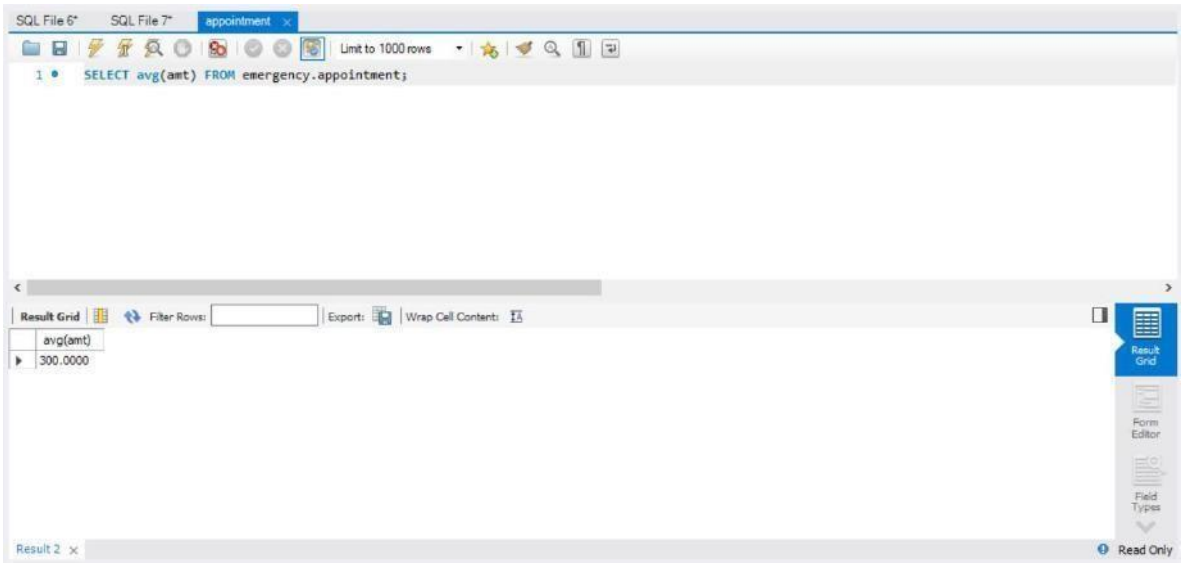
6) Write an MYSQLquery to update the value of phno of WID=102 of worker table



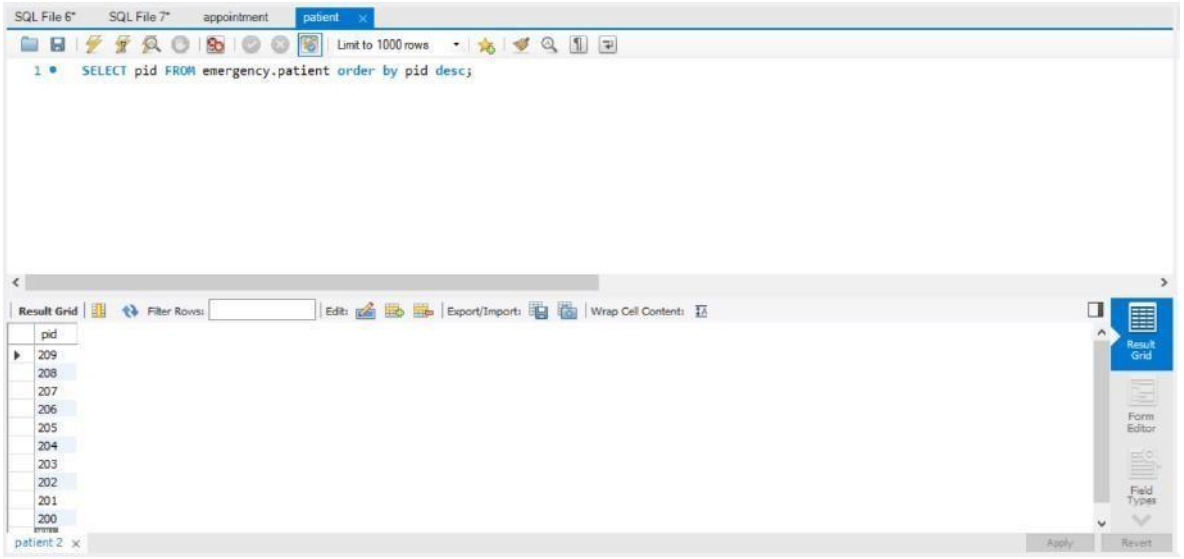
7) Write an MYSQL query to truncate the values of appointment table



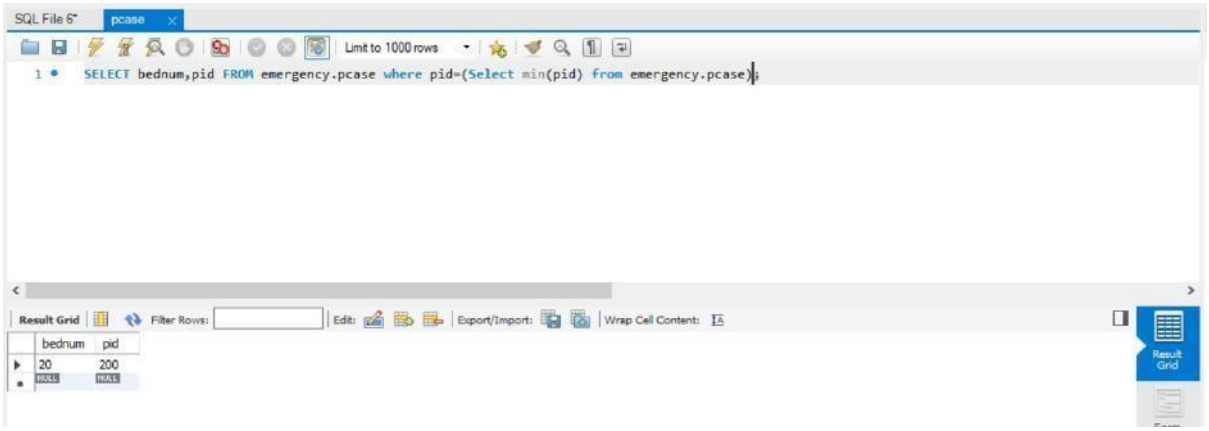
8) Write an MYSQL query to calculate average amount of appointment.



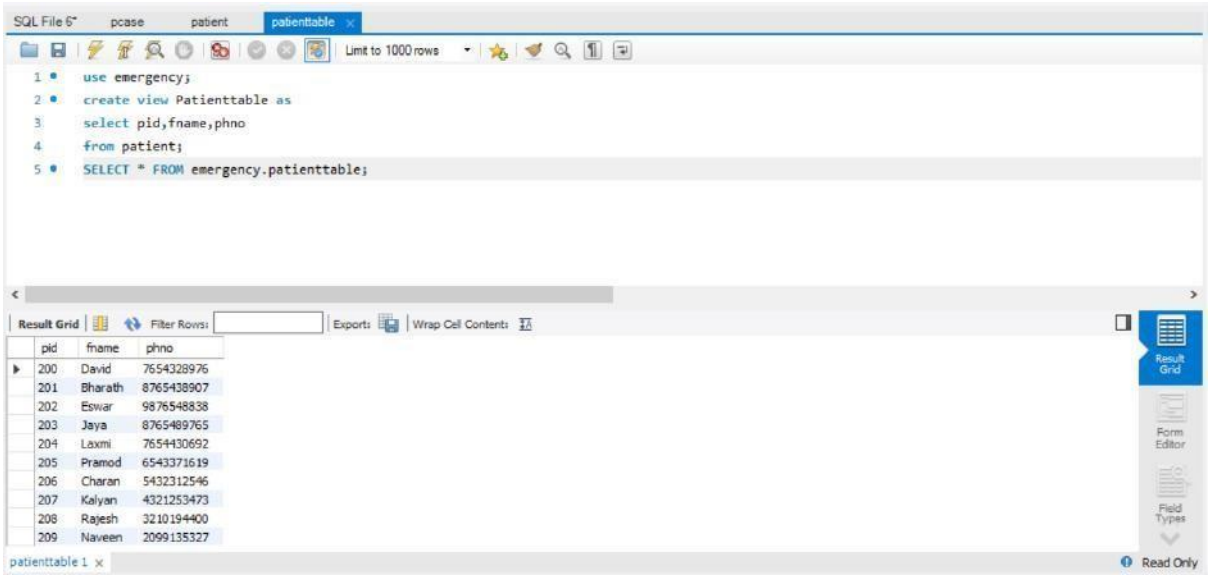
9) Write an MYSQL query to display patient number in descending order.



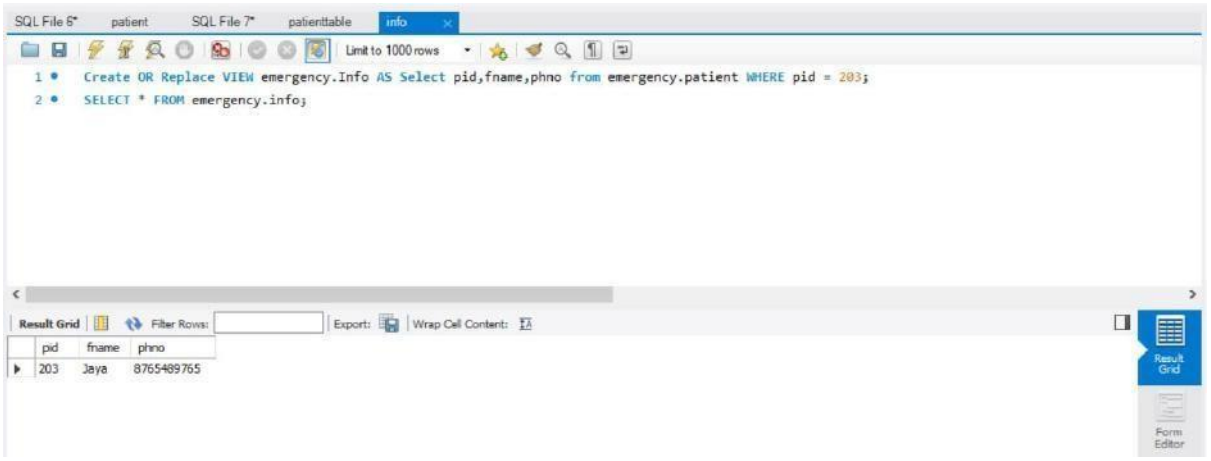
10) Write an MYSQL query to display bed number of patient with least patient id.



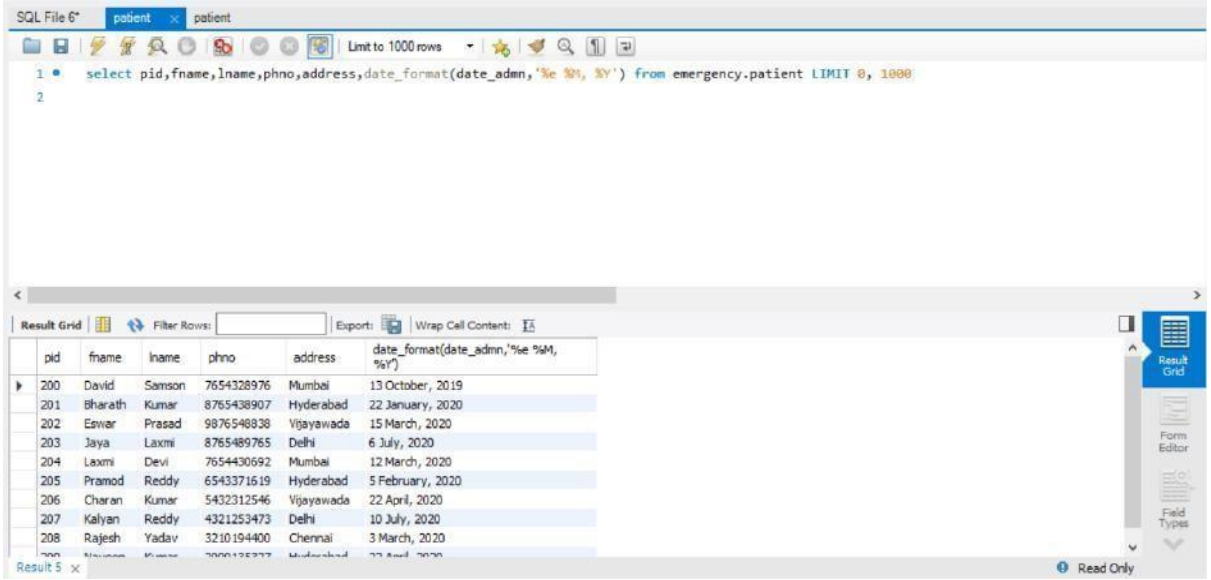
11) Write an MYSQL query to display bed number of patient with least patient id.



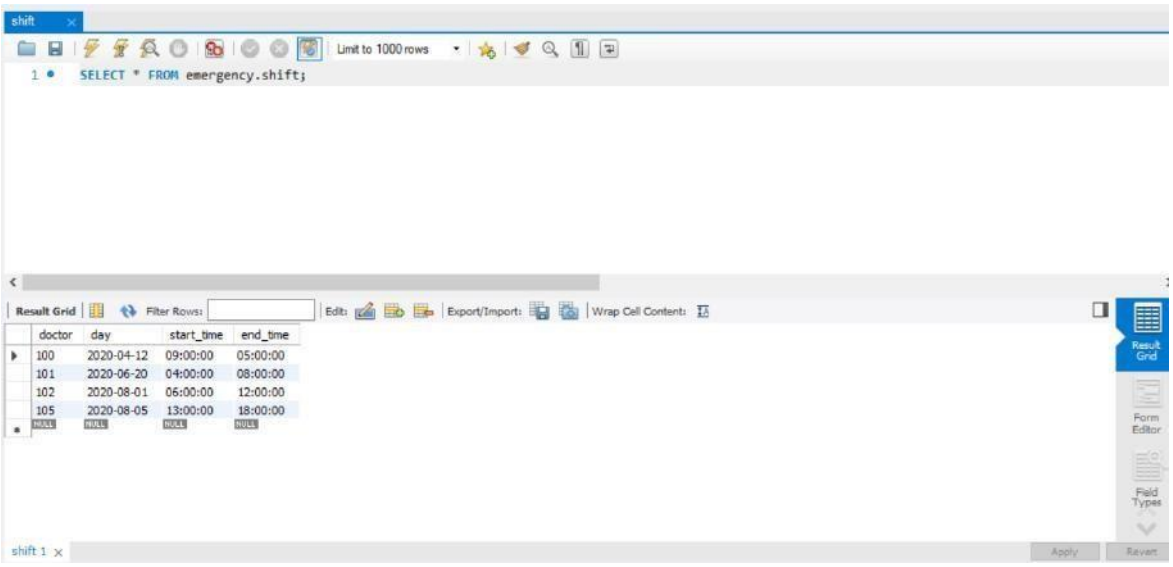
12) Write a MYSQL query to update view the Patientview where patientid>3 by using replace view statement



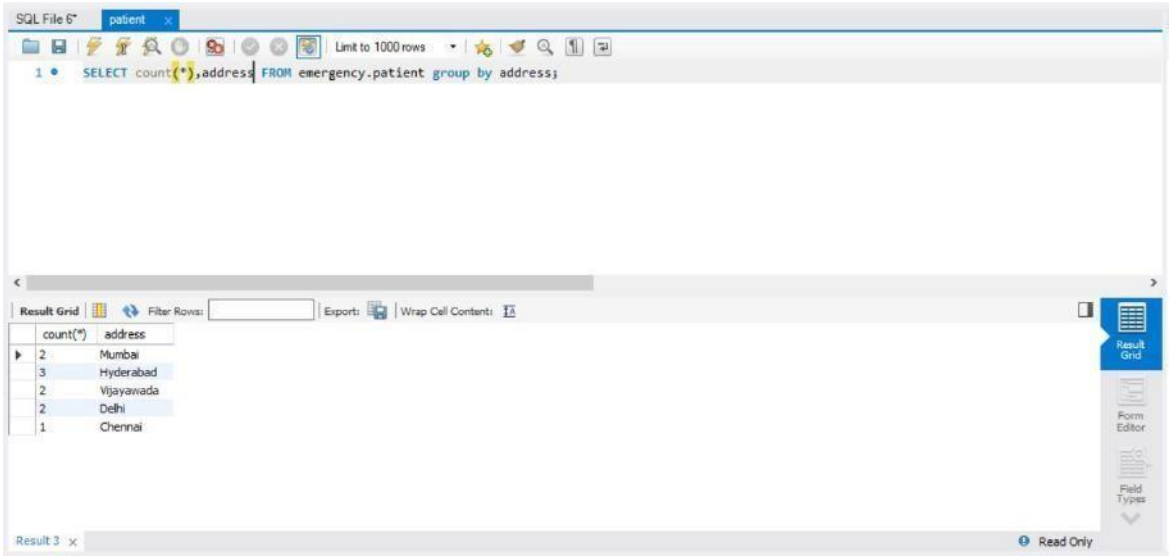
13) Display the patient date of joining into format “dd month yyyy’ for example as “24 March 2020”



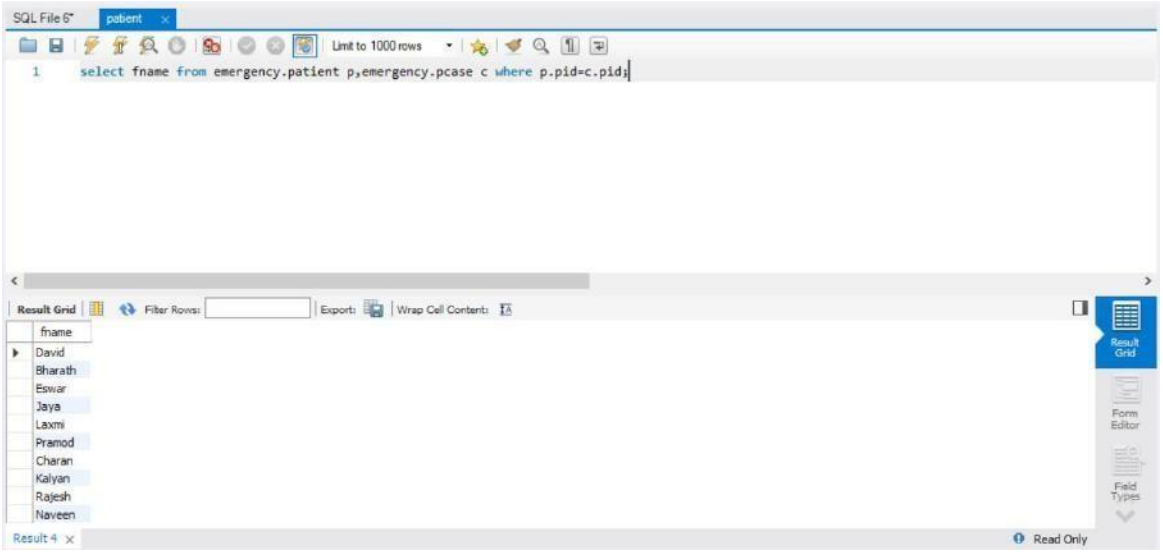
14) Display the patient joining time in 24 hour format. For eg: 3:30pm should be displayed as 15:30



15) Write a MYSQL query to display count of patients who come from same city.



16) Write an MYSQLquery to first name of patient who admitted and allotted the bed



CASE STUDY 2: TOUR OPERATING SYSTEM

1) Create all the tables required with constraints and relationships between them

```
create Schema Tour_Operating_System;
CREATE TABLE Tour_Operating_System.login (
    log_id INT NOT NULL,
    PRIMARY KEY (log_id),
    log_name VARCHAR(20) NOT NULL,
    log_pwd VARCHAR(15) NOT NULL
);
CREATE TABLE Tour_Operating_System.Person (
    id INT NOT NULL,
    PRIMARY KEY (id),
    name VARCHAR(20) NOT NULL,
    city VARCHAR(20) NOT NULL,
    state VARCHAR(20) NOT NULL,
    phone LONG NOT NULL,
    dob DATE NOT NULL,
    age INT NOT NULL
);
CREATE TABLE Tour_Operating_System.Customer (
    cust_id INT NOT NULL,
    PRIMARY KEY (cust_id),
    c_name VARCHAR(20) NOT NULL,
    c_addr VARCHAR(20) NOT NULL,
    c_mobile LONG NOT NULL
);
CREATE TABLE Tour_Operating_System.Tour (
    tr_id INT NOT NULL,
    PRIMARY KEY (tr_id),
    tr_start VARCHAR(20) NOT NULL,
    tr_dest VARCHAR(20),
    tr_date DATE NOT NULL,
    tr_status VARCHAR(20) NOT NULL,
    no_guides INT NOT NULL,
    no_tourist INT NOT NULL
);
CREATE TABLE Tour_Operating_System.Hotel (
    htl_id INT NOT NULL,
    PRIMARY KEY (htl_id),
    htl_name VARCHAR(20) NOT NULL,
    htl_place VARCHAR(20) NOT NULL,
    htl_rent INT NOT NULL,
    htl_type INT NOT NULL
);
```

2) Insert atleast 10 records into the tables

```
insert into tour_operating_system.login (log_id,log_name,log_pwd)
values(100,'Hello','abcd1234'),
    (101,'welcome','pqrs3456'),
    (102,'Good','xyzw1234'),
    (103,'Honesty','qwerty5678');
insert into tour_operating_system.person(id,name,city,state,phone,dob,age)
values(200,'Kiran','Hyderabad','Telangana',9876543218,'2000-10-12',19),
(201,'hari','Vijayawada','Andhra Pradesh',7865489765,'1980-06-25',39),
(202,'Raju','Hyderabad','Telangana',8976543352,'1970-01-01',49),
(203,'Eswar','Guntur','Andhra Pradesh',8765467800,'1985-04-15',35),
(204,'David','Vijayawada','Andhra Pradesh',8554392248,'2000-07-28',20),
(205,'Devi','Mangalagiri','Andhra Pradesh',8343316696,'1975-11-10',45),
(206,'Rani','Hyderabad','Telangana',8132241144,'1998-02-22',22),
(207,'Jaya','Vijayawada','Andhra Pradesh',7921165592,'2001-06-06',19),
(208,'Kalyan','Guntur','Andhra Pradesh',7710090040,'1990-09-18',30),
(209,'Gopal','Hyderabad','Telangana',7499014488,'1992-12-31',28);
insert into tour_operating_system.customer(cust_id,c_name,c_addr,c_mobile)
values(300,'Raju','Hyd',8765467800),
(301,'Hari','Delhi',8554392248),
(302,'Kiran','Mumbai',8343316696),
(303,'Giri','Kolkata',8139241144),
```

```
(304,'Devi','Goa',7935165592),
(305,'jaya','Chennai',7731090040),
(306,'Kalyan','Mysore',7527014488),
(307,'Kiran','Guntur',7322938936);
insert into tour_operating_system.tour(tr_id, tr_start, tr_dest, tr_date, tr_status, no_guides, no_tourist)
values(400,'vij','hyd','2020-09-30','Sucessful',5,25),
(401,'hyd','vij','2020-10-01','Cancelled',3,18),
(402,'gnt','tnl','2020-10-02','Sucessful',4,24),
(403,'tnl','gnt','2020-10-03','Cancelled',5,26),
(404,'vij','Mumbai','2020-10-04','Sucessful',6,28);
insert into tour_operating_system.hotel(htl_id, htl_name, htl_place, htl_rent, htl_type)
values(501,'Taj','Vijayawada',5000,4),
(502,'DV Manor','Vijayawada',4000,3),
(503,'Minerva','Guntur',3000,3),
(504,'Falaknuma palace','Hyderabad',8000,6),
(505,'ITC','Mumbai',9500,7);
insert into tour_operating_system.booking(bk_id, bk_date, bk_title, bk_type)
values(601,'2020-09-20','Vij-Hyd','Online'),
(602,'2020-09-21','Hyd-Vij','Offline'),
(603,'2020-09-22','Vij-Mumbai','Online'),
(604,'2020-09-23','Gnt-tnl','Offline');
```

3) Write an SQL Query to Update the value of bk_id=1001 in Booking table.

```
UPDATE tour_operating_system.booking
SET
    bk_id = 1001
WHERE
    bk_date = '2020-09-21';

Select * from tour_operating_system.booking;
```

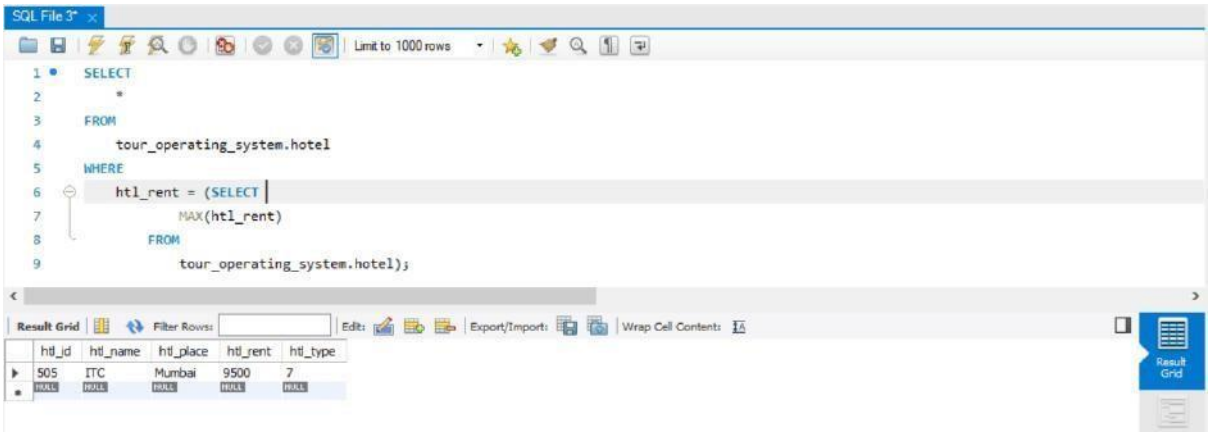
bk_id	bk_date	bk_title	bk_type
601	2020-09-20	Vij-Hyd	Online
603	2020-09-22	Vij-Mumbai	Online
604	2020-09-23	Gnt-tnl	Offline
1001	2020-09-21	Hyd-Vij	Offline

4) Write an SQL Query to Truncate the values of Login table.

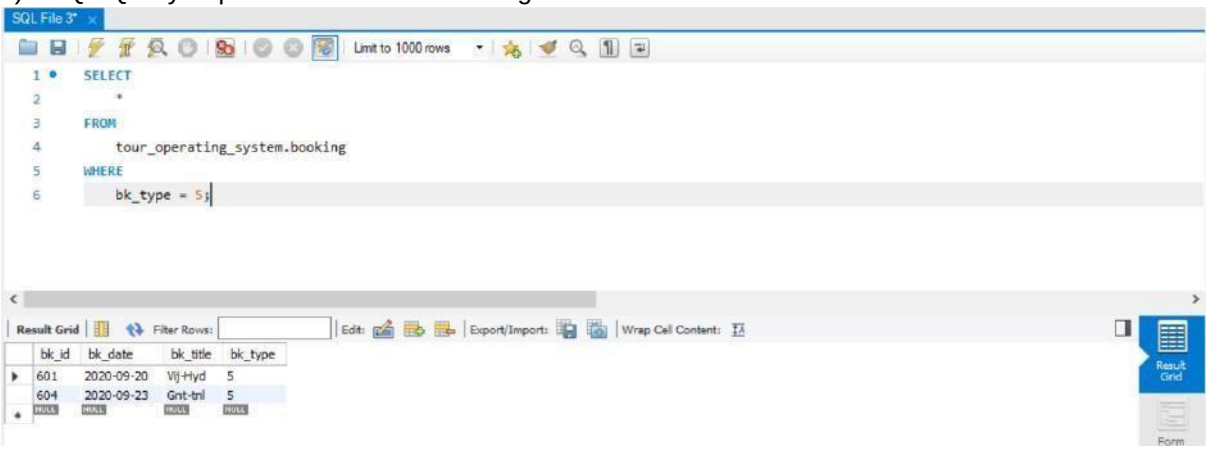
```
truncate tour_operating_system.login;
SELECT * FROM tour_operating_system.login;
```

log_id	log_name	log_pwd

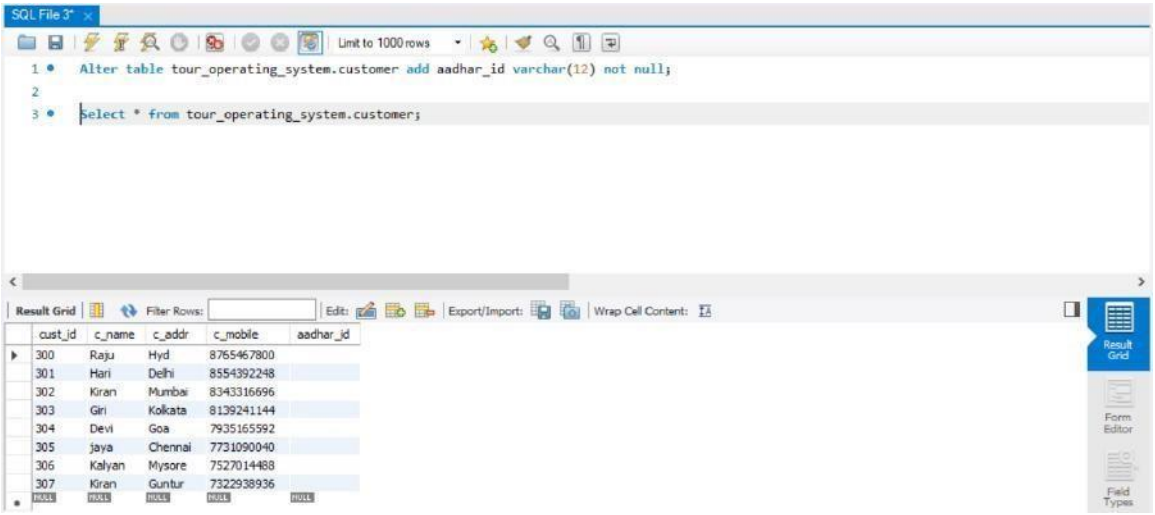
5) Which hotel rent is the highest amount for accommodations?



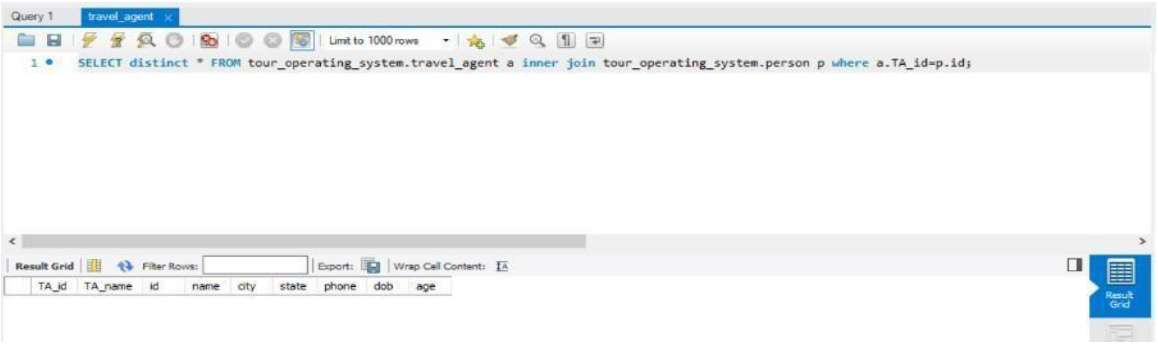
6) SQL Query to print the details of Booking information who have booked 5 star rated rooms.



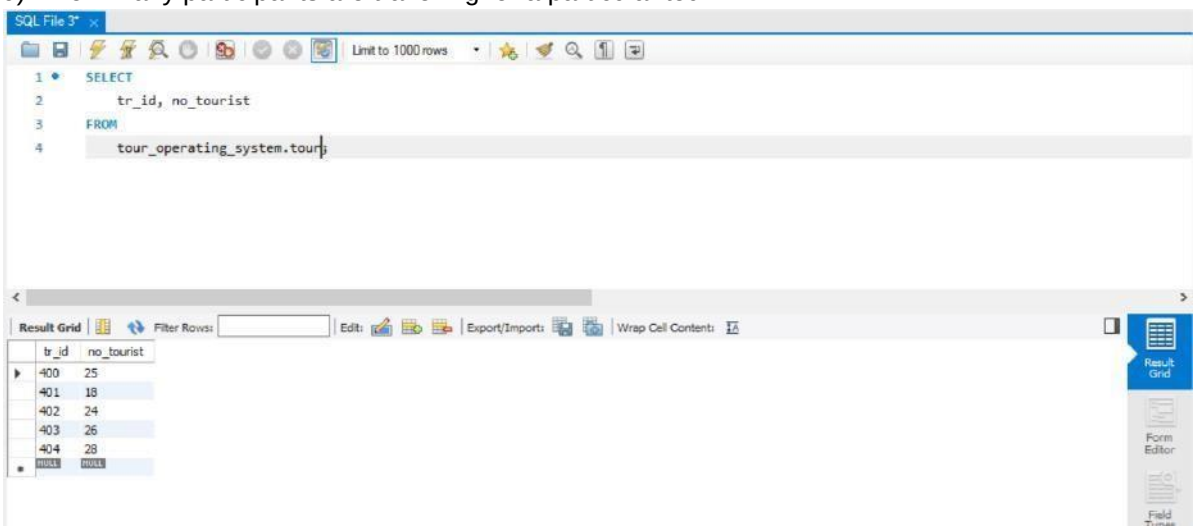
7) Insert a column named aadhar_id of the customer.



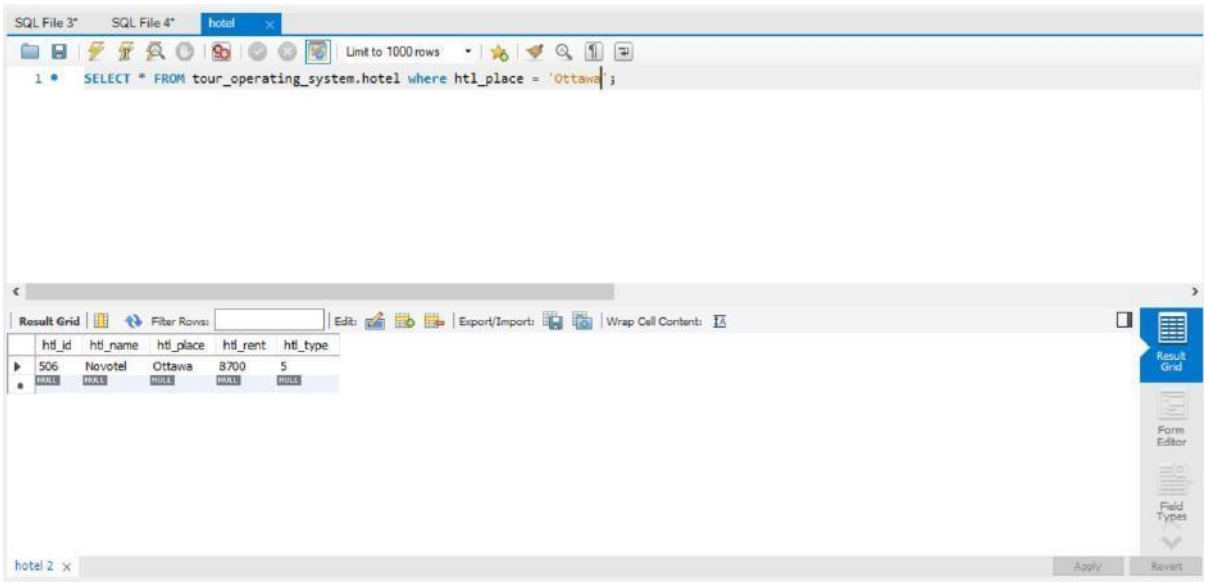
8) Display the persons whose TA_id and role_id is matched.



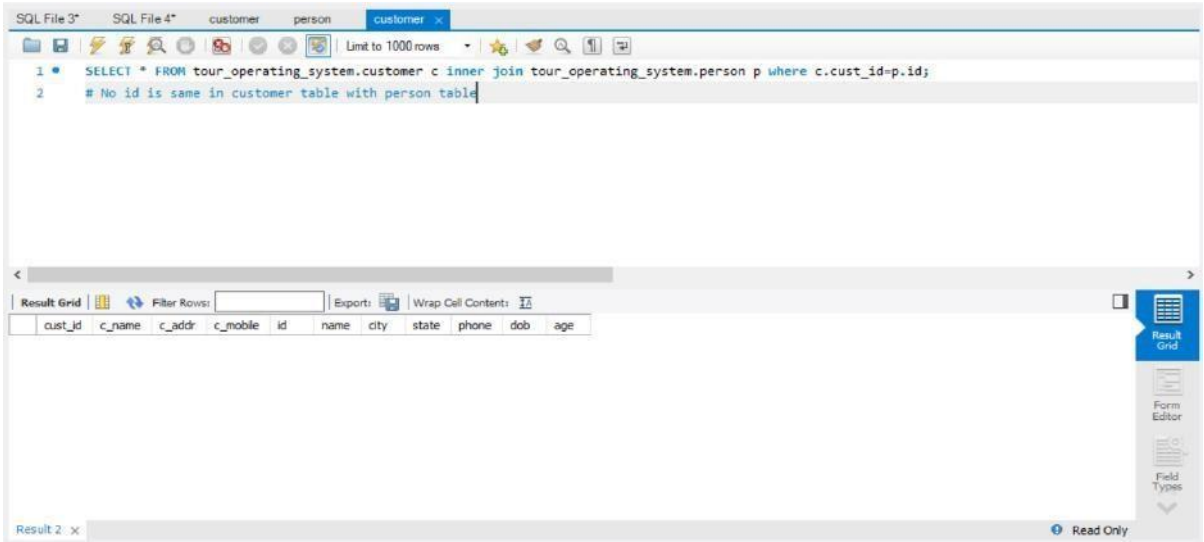
9) How many participants are travelling for a particular tour?



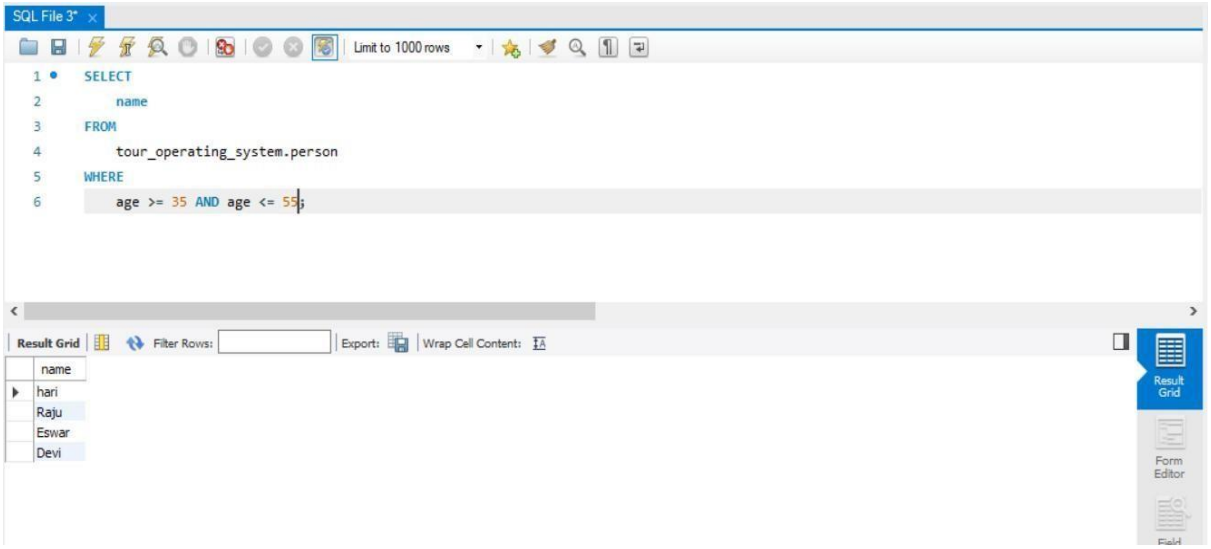
10) Search the names of the hotel in ottawa.



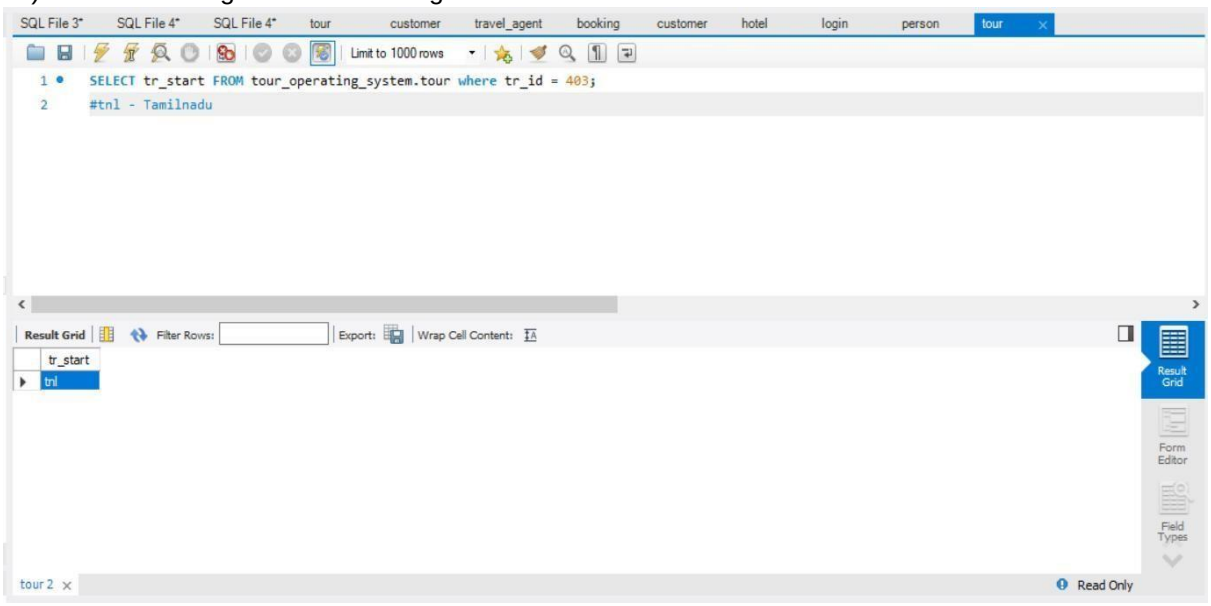
11) Extract the name the customer whose person id and customer id is equal using join.



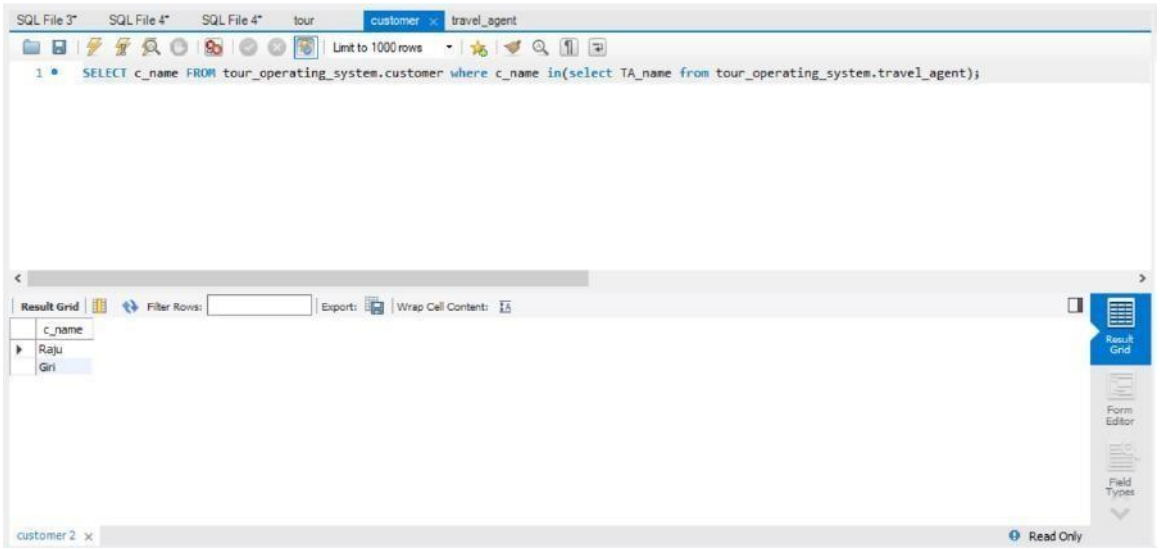
12) Display the name of the participants whose age is between 35 and 55;



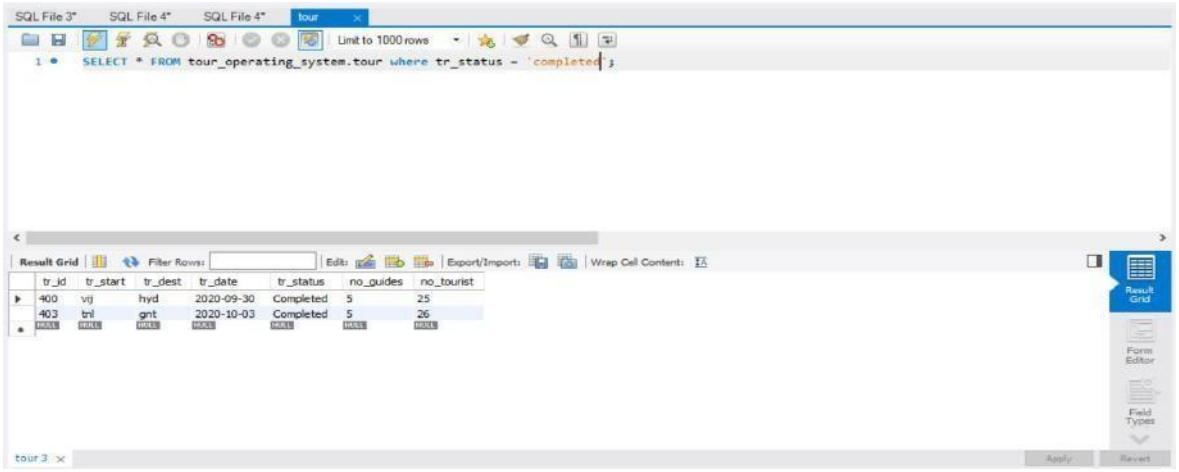
13) What is the origin of the travelling for tour id = 403



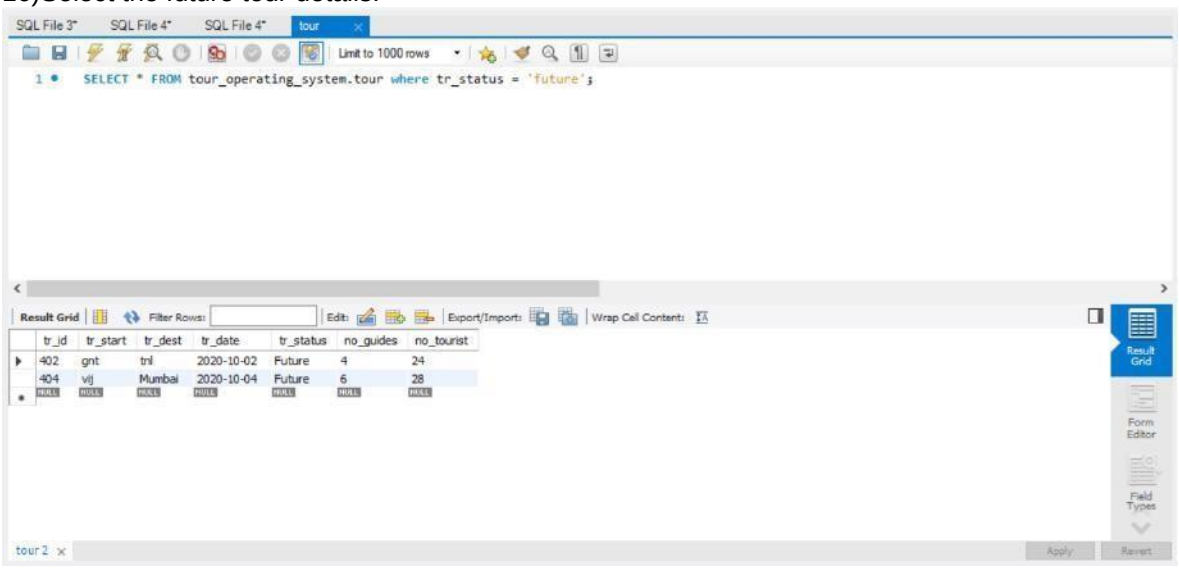
14) Select a person whose name is available in Customer and TravelAgent using nested query.



15) Select the past tour details.



16)Select the future tour details.



POST-LAB

1) Julia asked her students to create some coding challenges. Write a query to print the *hacker_id*, *name*, and the total number of challenges created by each student. Sort your results by the total number of challenges in descending order. If more than one student created the same number of challenges, then sort the result by *hacker_id*. If more than one student created the same number of challenges and the count is less than the maximum number of challenges created, then exclude those students from the result.

Input Format

The following tables contain challenge data:

- *Hackers:* The *hacker_id* is the id of the hacker, and *name* is the name of the hacker.
- *Challenges:* The *challenge_id* is the id of the challenge, and *hacker_id* is the id of the student who created thechallenge

Column	Type
hacker_id	Integer
name	String

Column	Type
challenge_id	Integer
hacker_id	Integer

Explanation

For *Sample Case 1*, we can get the following details:

hacker_id	name	challenges_created
12299	Rose	6
34856	Angela	6
79345	Frank	4
80491	Patrick	3
81041	Lisa	1

Students 12299 and 34856 both created 6 challenges. Because 6 is the maximum number of challenges created, these students are included in the result.

ANS.

```
SELECT c.hacker_id, h.name, COUNT(c.challenge_id) AS cnt FROM Hackers AS h JOIN
Challenges AS c ON h.hacker_id = c.hacker_id GROUP BY c.hacker_id, h.name HAVING
cnt = (SELECT COUNT(c1.challenge_id) FROM Challenges AS c1 GROUP BY c1.hacker_id
ORDER BY COUNT(*) DESC LIMIT 1) OR cnt NOT IN (SELECT COUNT(c2.challenge_id) FROM
Challenges AS c2 GROUP BY c2.hacker_id HAVING c2.hacker_id <> c.hacker_id) ORDER BY
cnt DESC, c.hacker_id;
```

2) Query all columns for all American cities in the **CITY** table with populations larger than 100000. The **CountryCode** for America is USA.The **CITY** table is described as follows:

CITY

Field	Type
ID	NUMBER
NAME	VARCHAR2(17)
COUNTRYCODE	VARCHAR2(3)
DISTRICT	VARCHAR2(20)
POPULATION	NUMBER

ANS.

```
SELECT * FROM CITY WHERE COUNTRYCODE = 'USA' AND POPULATION > 100000;
```