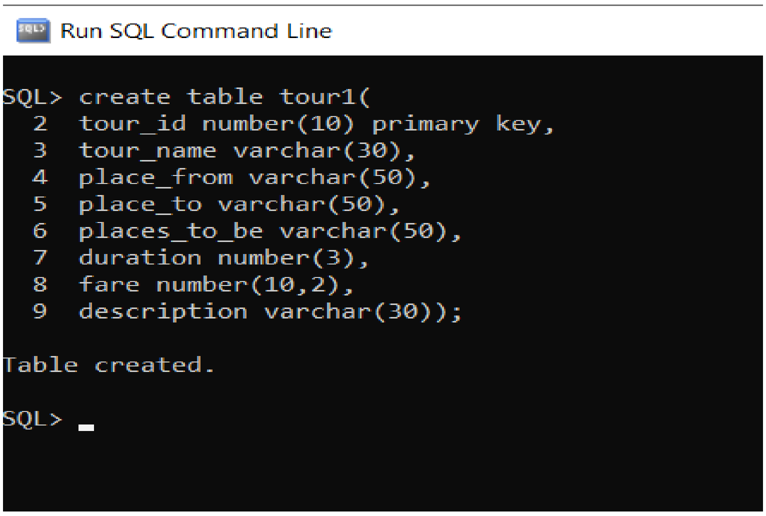
DATABASE MANAGEMENT SYSTEMS CSE2004

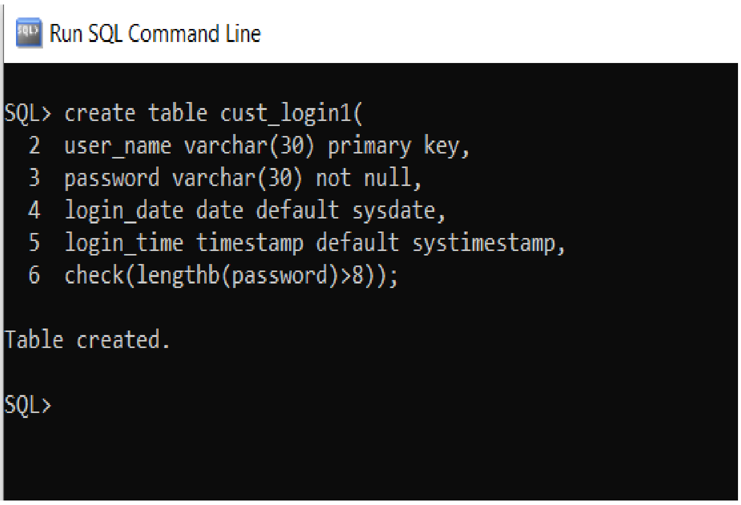
NAME – AVANI ANANYA PANDEY

REGN NUMBER – 20BCE2369

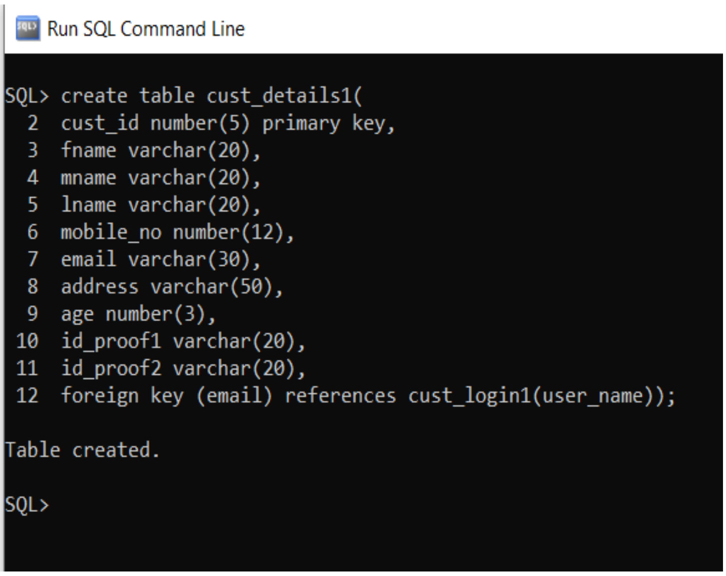
CREATING TABLES:



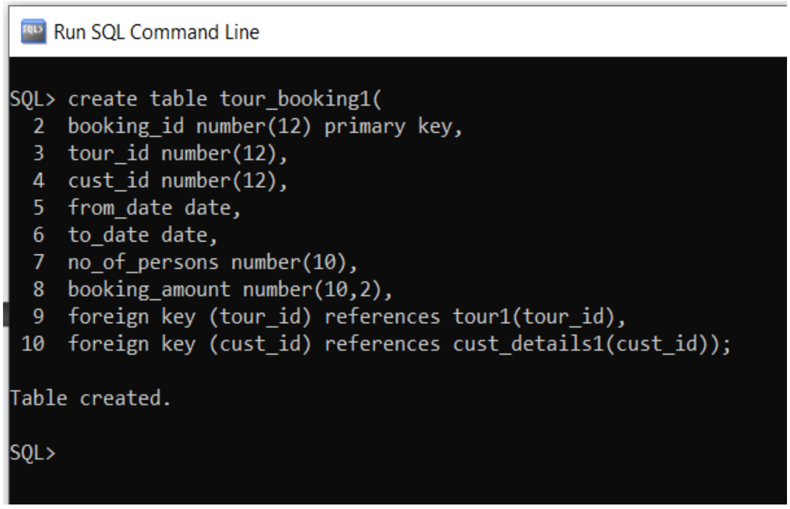
```
SQL> create table tour1(
  2  tour_id number(10) primary key,
  3  tour_name varchar(30),
  4  place_from varchar(50),
  5  place_to varchar(50),
  6  places_to_be varchar(50),
  7  duration number(3),
  8  fare number(10,2),
  9  description varchar(30));

Table created.

SQL> _
```



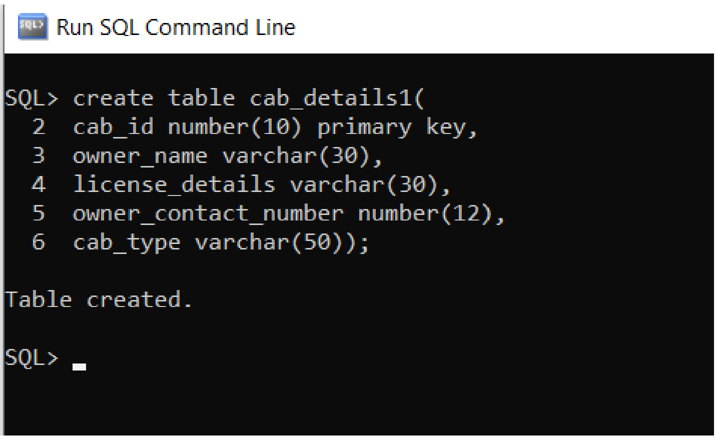
```
SQL> create table cust_login1(
  2  user_name varchar(30) primary key,
  3  password varchar(30) not null,
  4  login_date date default sysdate,
  5  login_time timestamp default systimestamp,
  6  check(lengthb(password)>8));

Table created.

SQL>
```



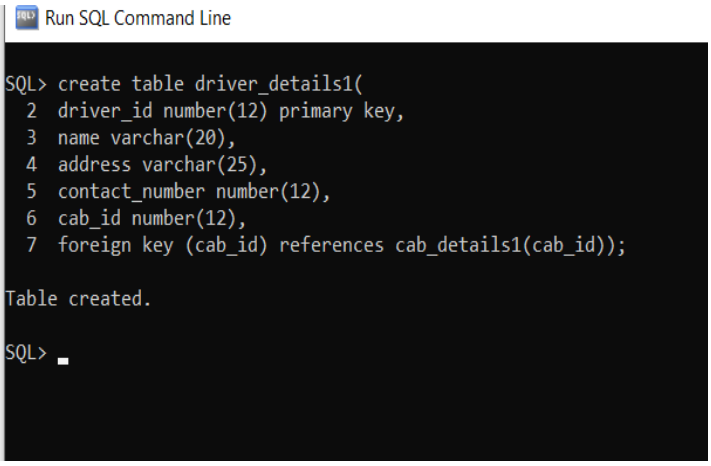
```
SQL> create table cust_details1(
  2  cust_id number(5) primary key,
  3  fname varchar(20),
  4  mname varchar(20),
  5  lname varchar(20),
  6  mobile_no number(12),
  7  email varchar(30),
  8  address varchar(50),
  9  age number(3),
 10  id_proof1 varchar(20),
 11  id_proof2 varchar(20),
 12  foreign key (email) references cust_login1(user_name));

Table created.

SQL>
```



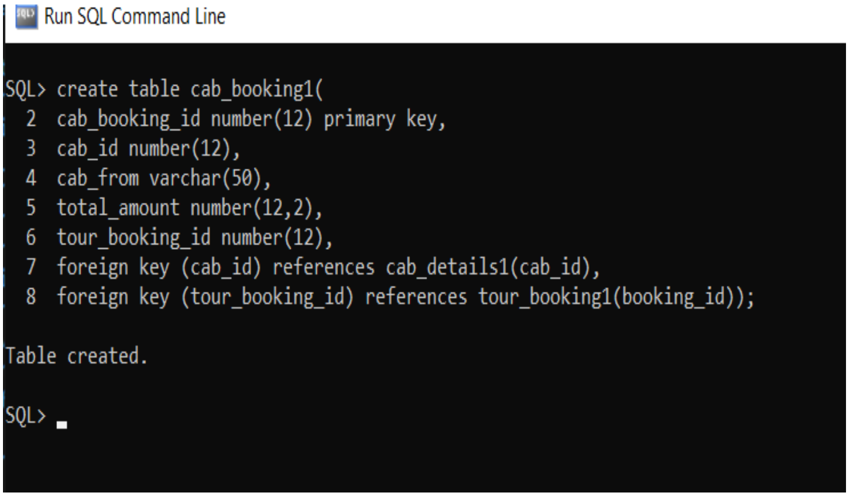
```
SQL> create table tour_booking1(
  2  booking_id number(12) primary key,
  3  tour_id number(12),
  4  cust_id number(12),
  5  from_date date,
  6  to_date date,
  7  no_of_persons number(10),
  8  booking_amount number(10,2),
  9  foreign key (tour_id) references tour1(tour_id),
 10  foreign key (cust_id) references cust_details1(cust_id));

Table created.

SQL>
```



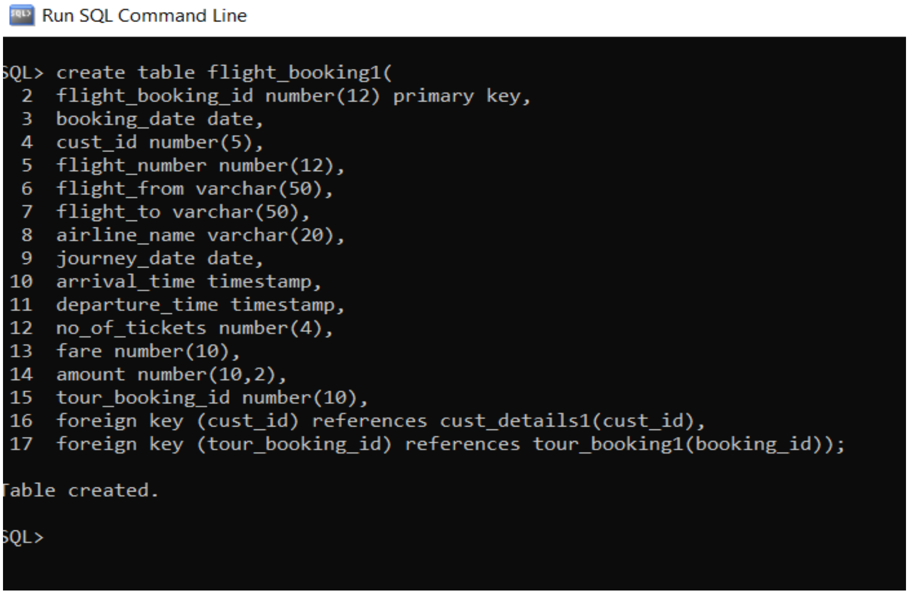
```
SQL> create table cab_details1(
  2  cab_id number(10) primary key,
  3  owner_name varchar(30),
  4  license_details varchar(30),
  5  owner_contact_number number(12),
  6  cab_type varchar(50));

Table created.

SQL> _
```



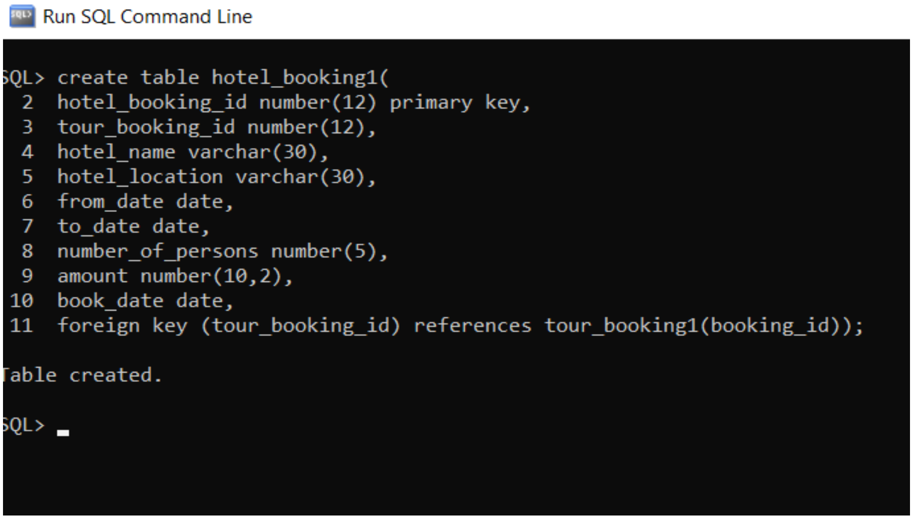
```
SQL> create table driver_details1(
  2  driver_id number(12) primary key,
  3  name varchar(20),
  4  address varchar(25),
  5  contact_number number(12),
  6  cab_id number(12),
  7  foreign key (cab_id) references cab_details1(cab_id));

Table created.

SQL> _
```



```
SQL> create table cab_booking1(
cab_booking_id number(12) primary key,
cab_id number(12),
cab_from varchar(50),
total_amount number(12,2),
tour_booking_id number(12),
foreign key (cab_id) references cab_details1(cab_id),
foreign key (tour_booking_id) references tour_booking1(booking_id));

Table created.

SQL>
```



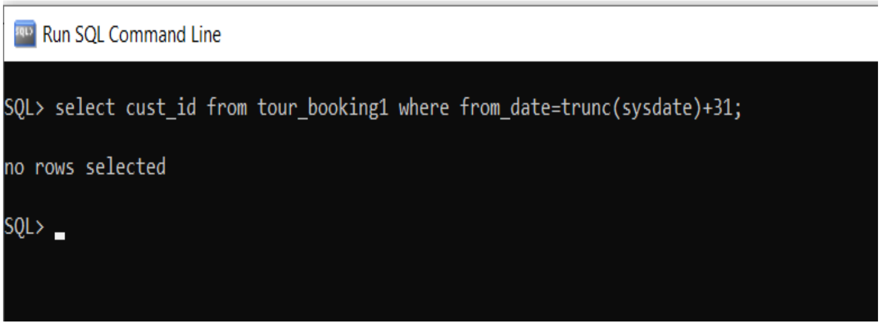
```
SQL> create table flight_booking1(
flight_booking_id number(12) primary key,
booking_date date,
cust_id number(5),
flight_number number(12),
flight_from varchar(50),
flight_to varchar(50),
airline_name varchar(20),
journey_date date,
arrival_time timestamp,
departure_time timestamp,
no_of_tickets number(4),
fare number(10),
amount number(10,2),
tour_booking_id number(10),
foreign key (cust_id) references cust_details1(cust_id),
foreign key (tour_booking_id) references tour_booking1(booking_id));

Table created.

SQL>
```



```
SQL> create table hotel_booking1(
hotel_booking_id number(12) primary key,
tour_booking_id number(12),
hotel_name varchar(30),
hotel_location varchar(30),
from_date date,
to_date date,
number_of_persons number(5),
amount number(10,2),
book_date date,
foreign key (tour_booking_id) references tour_booking1(booking_id));

Table created.

SQL>
```

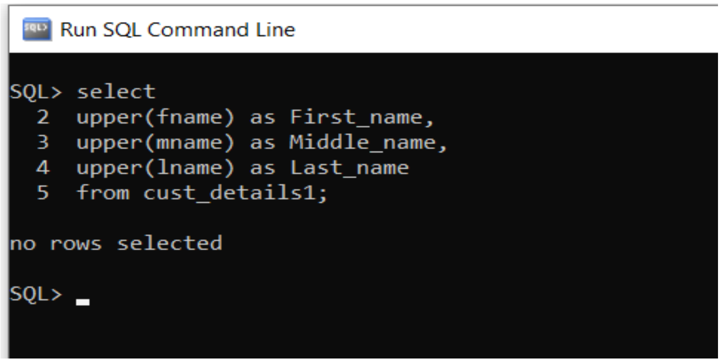
QUESTION 1:

1. Find the customers whose date of journey is one month from today.



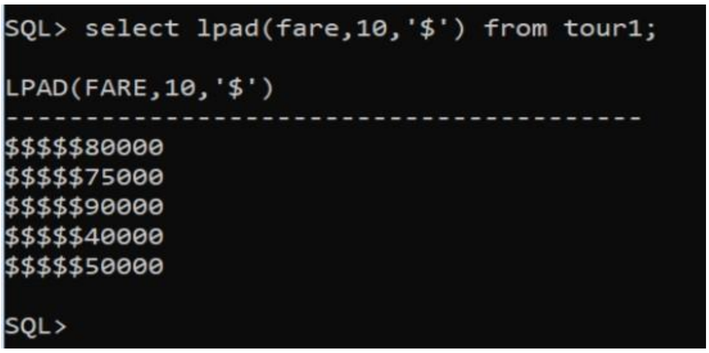
```
SQL> select cust_id from tour_booking1 where from_date=trunc(sysdate)+31;

no rows selected

SQL> _
```

2. Print the Customers‟ name in upper case. (Hint: customer name has three sections, first, middle and last name.)



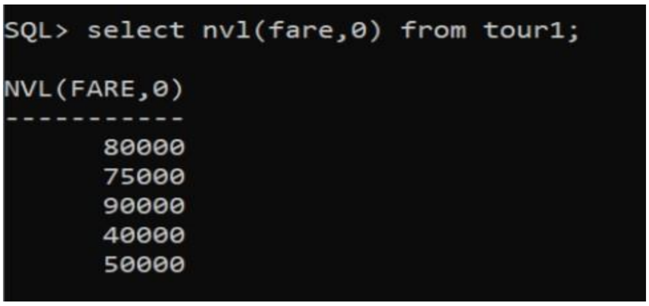
```
SQL> select
     upper(fname) as First_name,
     upper(mname) as Middle_name,
     upper(lname) as Last_name
     from cust_details1;

no rows selected

SQL> _
```

3. Print the fare of tour with left padding character.



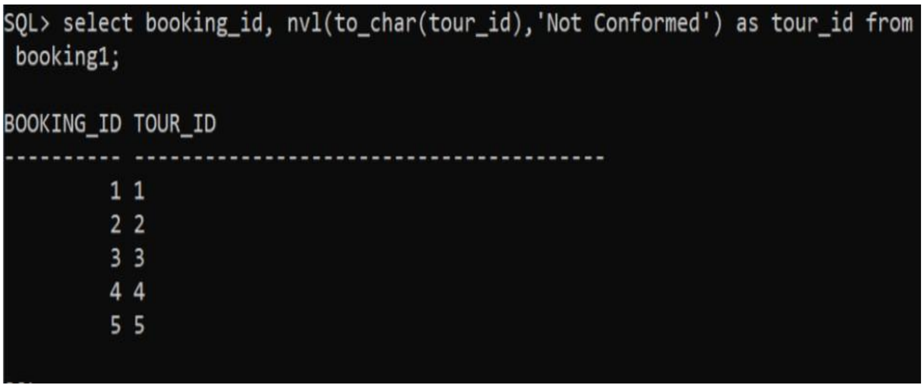
```
SQL> select lpad(fare,10,'$') from tour1;

LPAD(FARE,10,'$')
----------------------------------------
$$$$$80000
$$$$$75000
$$$$$90000
$$$$$40000
$$$$$50000

SQL>
```

4. Display the fare details of all tour, if any value is ZERO, print as NULL value.



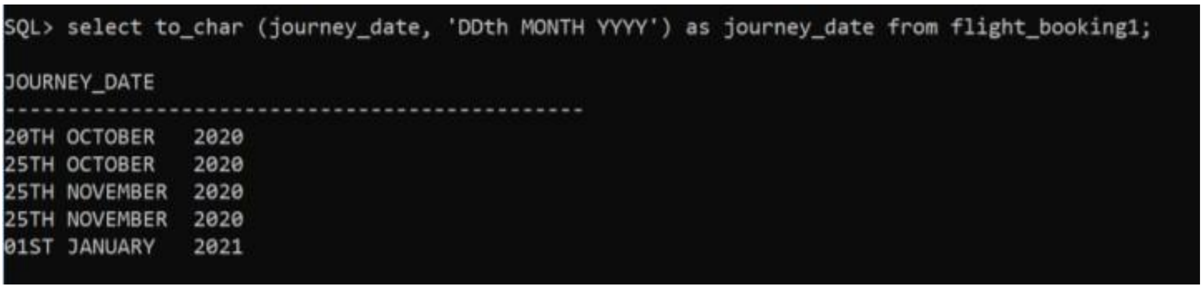
```
SQL> select nvl(fare,0) from tour1;

NVL(FARE,0)
-----------
      80000
      75000
      90000
      40000
      50000
```

5.Display the booking id and tour id, if tour id is null, print 'notconformed'.



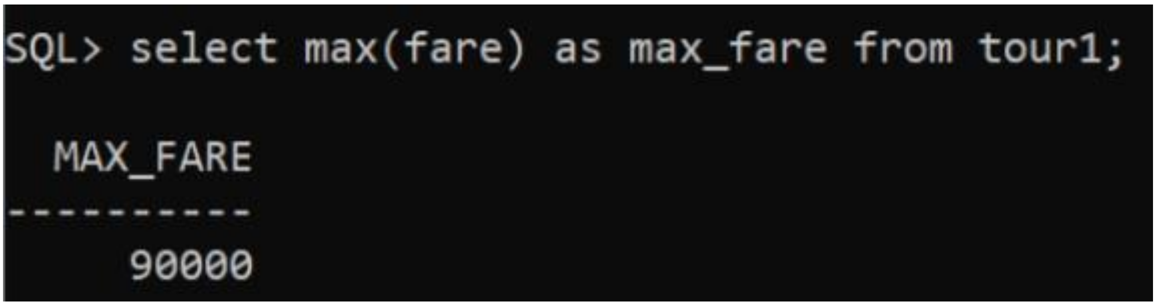
```
SQL> select booking_id, nvl(to_char(tour_id),'Not Conformed') as tour_id from
 booking1;

BOOKING_ID TOUR_ID
---------- ----------------------------------------
         1 1
         2 2
         3 3
         4 4
         5 5
```

6. Print the journey date in the format of '27th August 2020'.



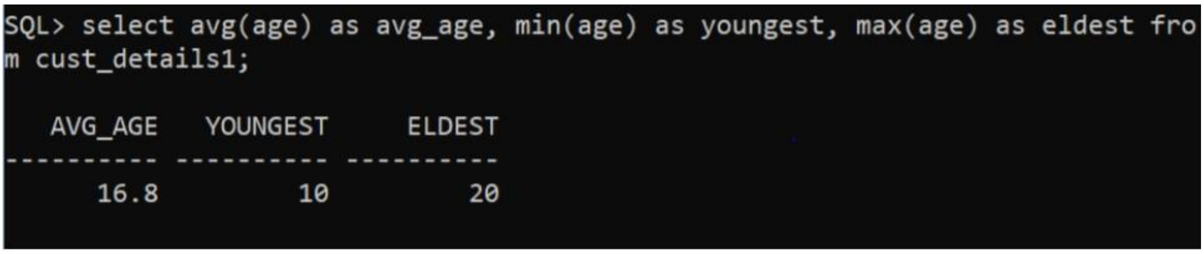
```
SQL> select to_char (journey_date, 'DDth MONTH YYYY') as journey_date from flight_booking1;

JOURNEY_DATE
---------------------------------------------
20TH OCTOBER   2020
25TH OCTOBER   2020
25TH NOVEMBER  2020
25TH NOVEMBER  2020
01ST JANUARY   2021
```

7. Find the maximum fare from the tour.



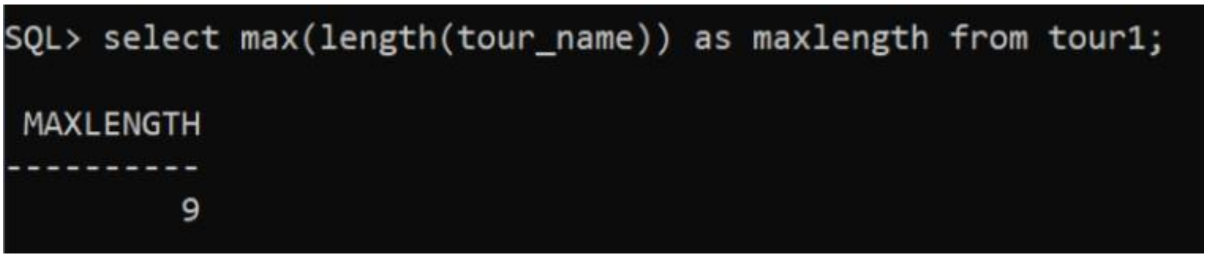
```
SQL> select max(fare) as max_fare from tour1;

  MAX_FARE
----------
     90000
```

8. Find the average age of customer, younger and elder customer details.



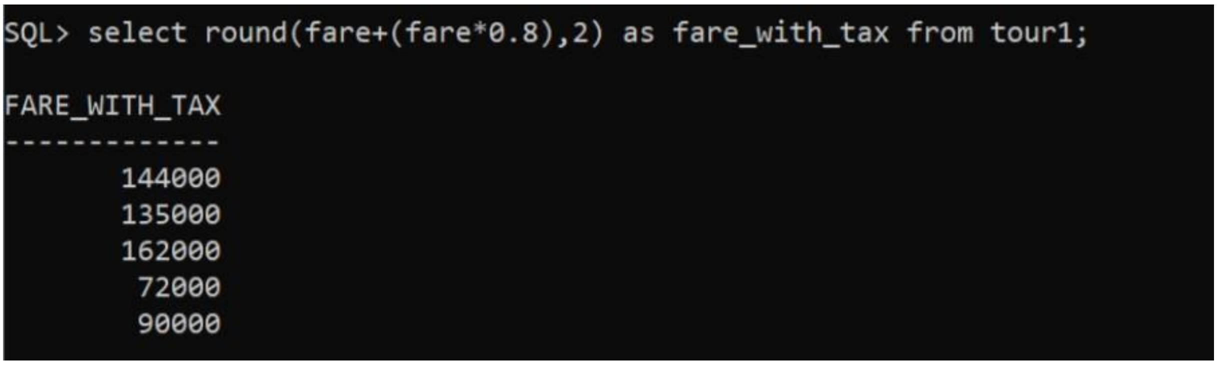
```
SQL> select avg(age) as avg_age, min(age) as youngest, max(age) as eldest fro
m cust_details1;

   AVG_AGE   YOUNGEST     ELDEST
---------- ---------- ----------
      16.8         10         20
```

9. Find the maximum length of tour name available in the database.



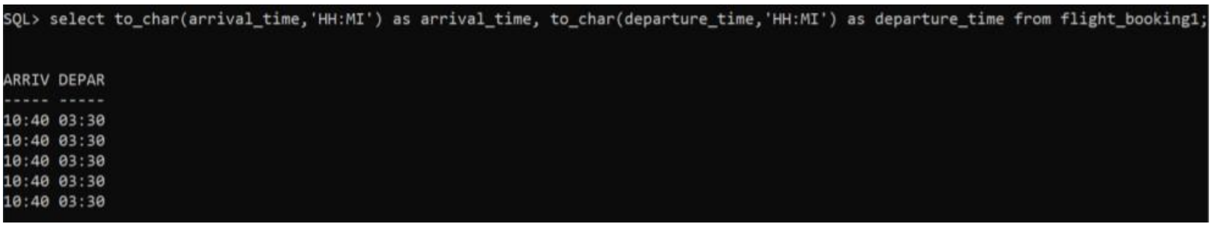
```
SQL> select max(length(tour_name)) as maxlength from tour1;

 MAXLENGTH
----------
         9
```

10. Print the fare amount of the tour as rounded value after inclusion of tax.



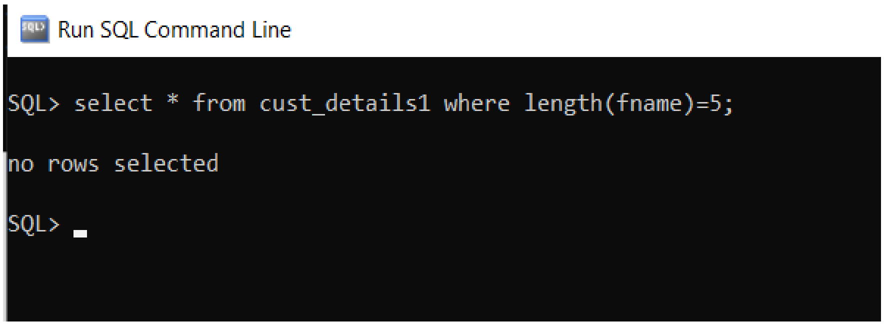
```
SQL> select round(fare+(fare*0.8),2) as fare_with_tax from tour1;

FARE_WITH_TAX
-------------
       144000
       135000
       162000
        72000
        90000
```

11. 13. Display the arrival time, depart time in the format HH:MI (24 hours and minutes).



```
SQL> select to_char(arrival_time,'HH:MI') as arrival_time, to_char(departure_time,'HH:MI') as departure_time from flight_booking1;

ARRIV DEPAR
----- -----
10:40 03:30
10:40 03:30
10:40 03:30
10:40 03:30
10:40 03:30
```

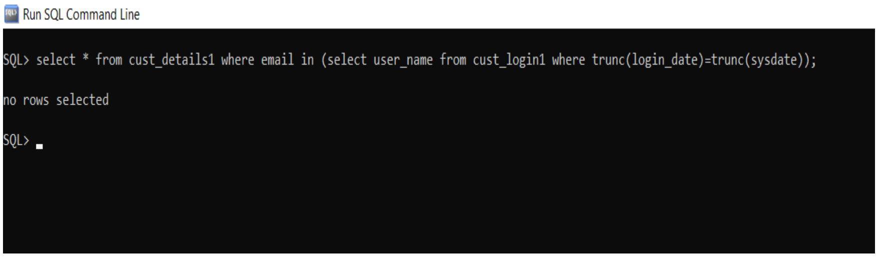
12. 14. Find the all the customer details whose name is exactly 5 characters long.



```
SQL> select * from cust_details1 where length(fname)=5;

no rows selected

SQL>
```

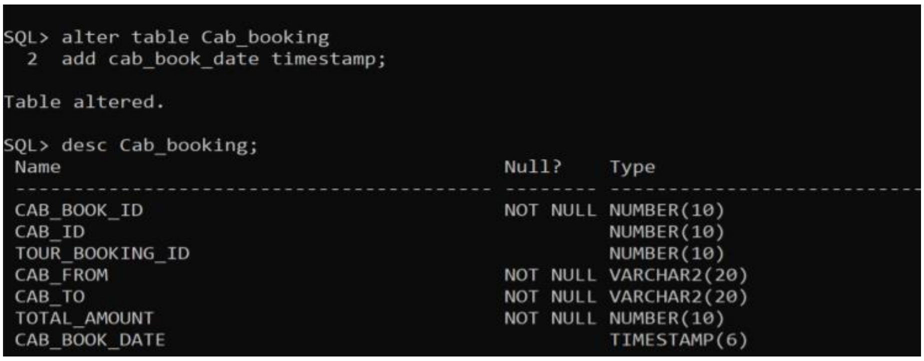
QUESTION 2: Use Nested Query(in Operators)

1. Display all the customer details who has logged in yesterday or any given date.



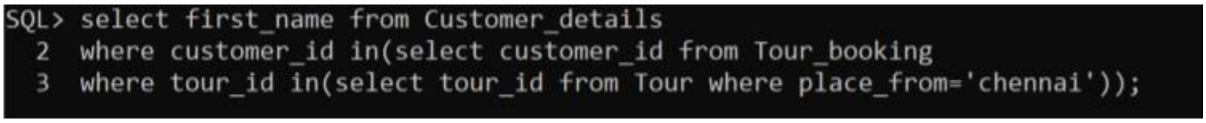
```
SQL> select * from cust_details1 where email in (select user_name from cust_login1 where trunc(login_date)=trunc(sysdate));

no rows selected

SQL>
```

2. Add cab booking date in cab booking table and find the driver(s) contact details that has booking on given date or today.



```
SQL> alter table Cab_booking
  2  add cab_book_date timestamp;

Table altered.

SQL> desc Cab_booking;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CAB_BOOK_ID                               NOT NULL NUMBER(10)
 CAB_ID                                             NUMBER(10)
 TOUR_BOOKING_ID                                    NUMBER(10)
 CAB_FROM                                  NOT NULL VARCHAR2(20)
 CAB_TO                                    NOT NULL VARCHAR2(20)
 TOTAL_AMOUNT                              NOT NULL NUMBER(10)
 CAB_BOOK_DATE                                      TIMESTAMP(6)
```

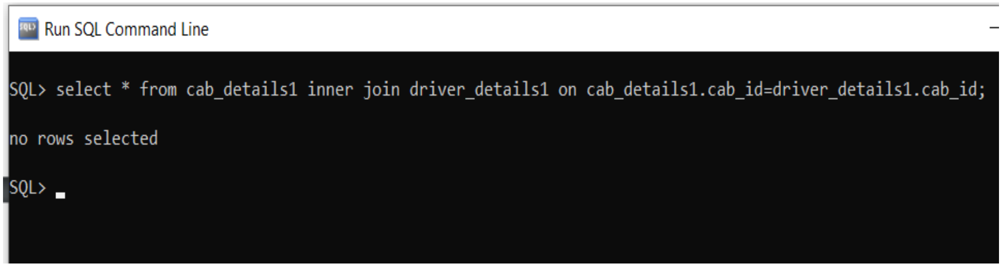
3. Find the Passenger name who has booked the ticket and their journey starts from 'Chennai'.



```
SQL> select first_name from Customer_details
  2  where customer_id in(select customer_id from Tour_booking
  3  where tour_id in(select tour_id from Tour where place_from='chennai'));
```

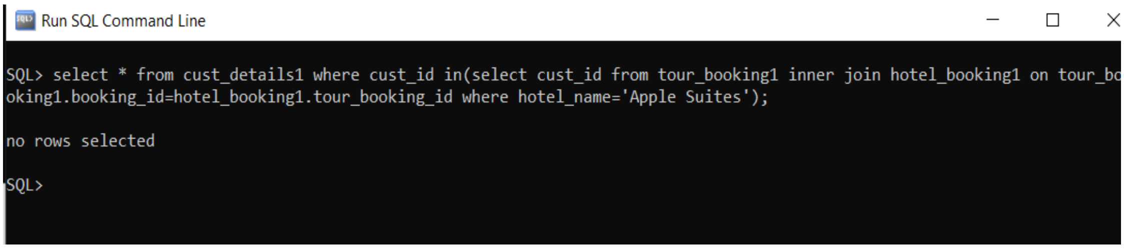
Use Join Query:

1. Find the cab and its driver details.



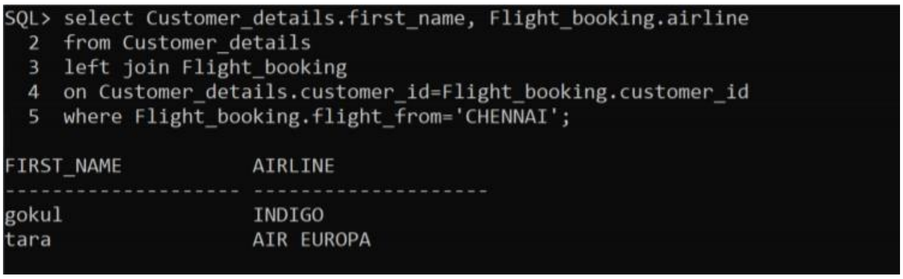
```
SQL> select * from cab_details1 inner join driver_details1 on cab_details1.cab_id=driver_details1.cab_id;

no rows selected

SQL>
```

2. Display all those customer details who has booked Hotel “Hotel Apple Suites” or any given name.



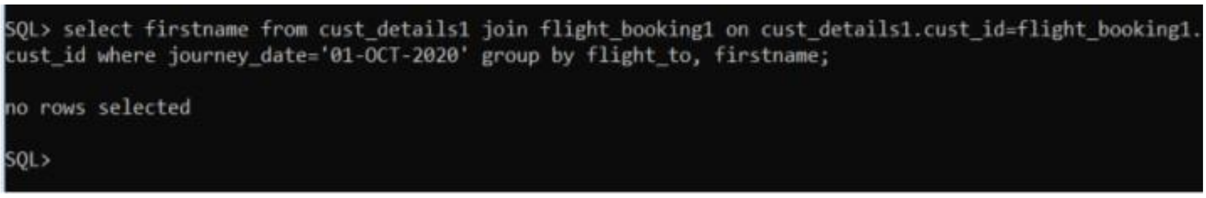
```
SQL> select * from cust_details1 where cust_id in(select cust_id from tour_booking1 inner join hotel_booking1 on tour_bo
oking1.booking_id=hotel_booking1.tour_booking_id where hotel_name='Apple Suites');

no rows selected

SQL>
```

3. Find the Passenger name and flight name who have booked the flight that starts from 'Chennai'.



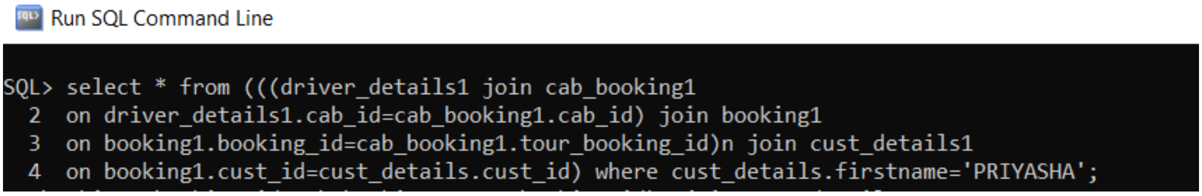
```
SQL> select Customer_details.first_name, Flight_booking.airline
  2  from Customer_details
  3  left join Flight_booking
  4  on Customer_details.customer_id=Flight_booking.customer_id
  5  where Flight_booking.flight_from='CHENNAI';

FIRST_NAME           AIRLINE
-------------------- --------------------
gokul                INDIGO
tara                 AIR EUROPA
```

4. Find the list of all customers who has booked the same destination for given date.

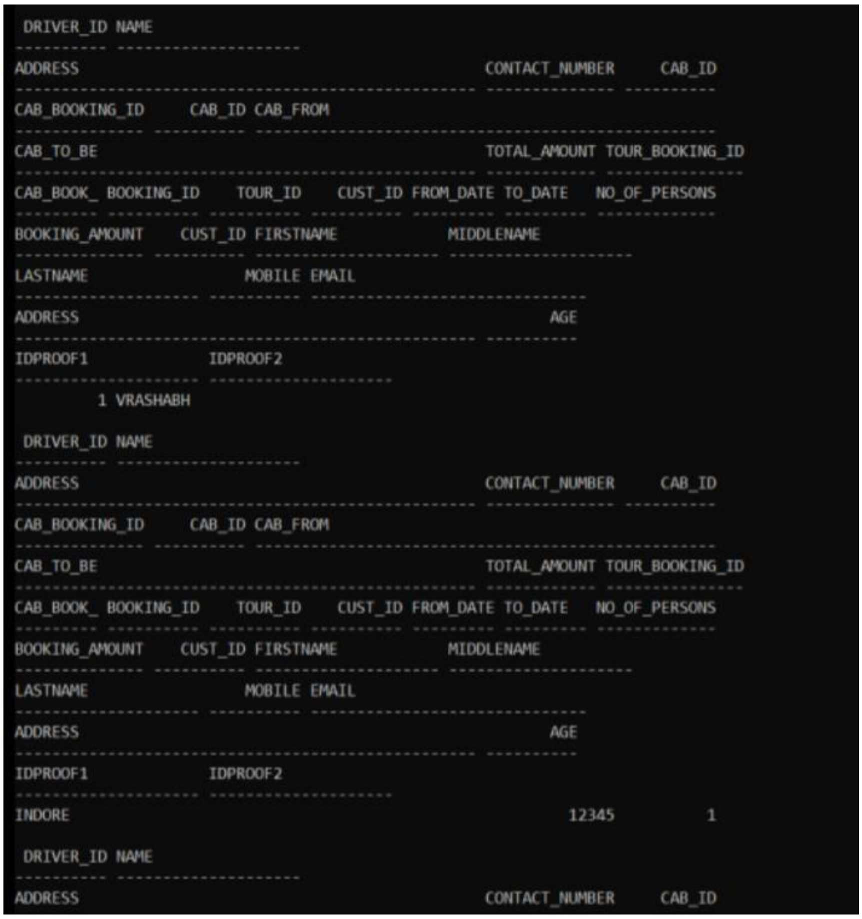


```
SQL> select firstname from cust_details1 join flight_booking1 on cust_details1.cust_id=flight_booking1.
cust_id where journey_date='01-OCT-2020' group by flight_to, firstname;

no rows selected

SQL>
```

5. Display the cab driver details who has attended the customer “XYZ”.



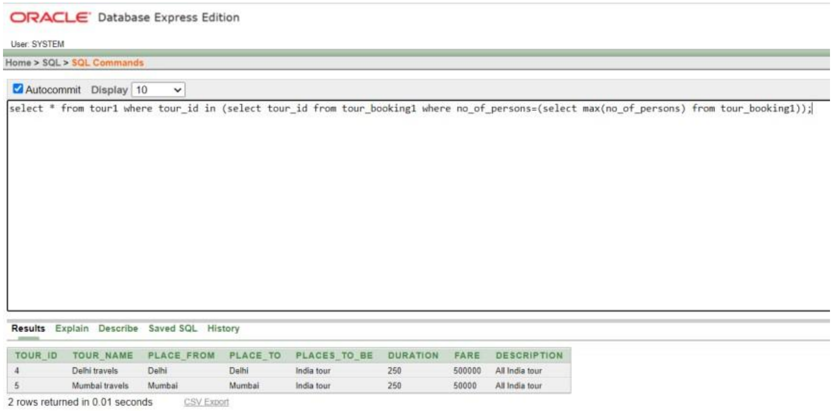
```
SQL> select * from (((driver_details1 join cab_booking1
  2  on driver_details1.cab_id=cab_booking1.cab_id) join booking1
  3  on booking1.booking_id=cab_booking1.tour_booking_id)n join cust_details1
  4  on booking1.cust_id=cust_details.cust_id) where cust_details.firstname='PRIYASHA';
```



```
 DRIVER_ID NAME
---------- --------------------
ADDRESS                                            CONTACT_NUMBER     CAB_ID
-------------------------------------------------- -------------- ----------
CAB_BOOKING_ID     CAB_ID CAB_FROM
-------------- ---------- --------------------------------------------------
CAB_TO_BE                                          TOTAL_AMOUNT TOUR_BOOKING_ID
-------------------------------------------------- ------------ ---------------
CAB_BOOK_ BOOKING_ID    TOUR_ID    CUST_ID FROM_DATE TO_DATE   NO_OF_PERSONS
--------- ---------- ---------- ---------- --------- --------- -------------
BOOKING_AMOUNT    CUST_ID FIRSTNAME            MIDDLENAME
-------------- ---------- -------------------- --------------------
LASTNAME                 MOBILE EMAIL
-------------------- ---------- ------------------------------
ADDRESS                                                   AGE
-------------------------------------------------- ----------
IDPROOF1             IDPROOF2
-------------------- --------------------
         1 VRASHABH

 DRIVER_ID NAME
---------- --------------------
ADDRESS                                            CONTACT_NUMBER     CAB_ID
-------------------------------------------------- -------------- ----------
CAB_BOOKING_ID     CAB_ID CAB_FROM
-------------- ---------- --------------------------------------------------
CAB_TO_BE                                          TOTAL_AMOUNT TOUR_BOOKING_ID
-------------------------------------------------- ------------ ---------------
CAB_BOOK_ BOOKING_ID    TOUR_ID    CUST_ID FROM_DATE TO_DATE   NO_OF_PERSONS
--------- ---------- ---------- ---------- --------- --------- -------------
BOOKING_AMOUNT    CUST_ID FIRSTNAME            MIDDLENAME
-------------- ---------- -------------------- --------------------
LASTNAME                 MOBILE EMAIL
-------------------- ---------- ------------------------------
ADDRESS                                                   AGE
-------------------------------------------------- ----------
IDPROOF1             IDPROOF2
-------------------- --------------------
INDORE                                                      12345          1

 DRIVER_ID NAME
---------- --------------------
ADDRESS                                            CONTACT_NUMBER     CAB_ID
```

QUESTION 3 - Complex queries(use groupby/groupby having/join/nested)

1 List the tour details for which the maximum numbers of peoples have booked.

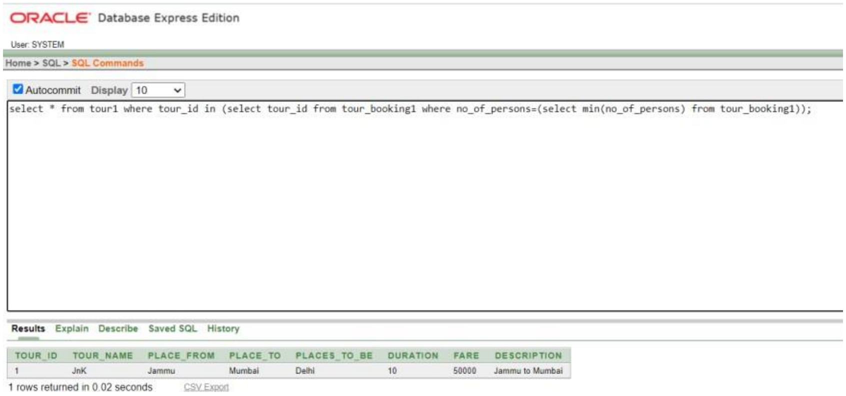

ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select * from tour1 where tour_id in (select tour_id from tour_booking1 where no_of_persons=(select max(no_of_persons) from tour_booking1));
```

TOUR_ID	TOUR_NAME	PLACE_FROM	PLACE_TO	PLACES_TO_BE	DURATION	FARE	DESCRIPTION
4	Delhi travels	Delhi	Delhi	India tour	250	500000	All India tour
5	Mumbai travels	Mumbai	Mumbai	India tour	250	50000	All India tour

2 rows returned in 0.01 seconds

2 List the tour details for which the minimum numbers of peoples have booked.

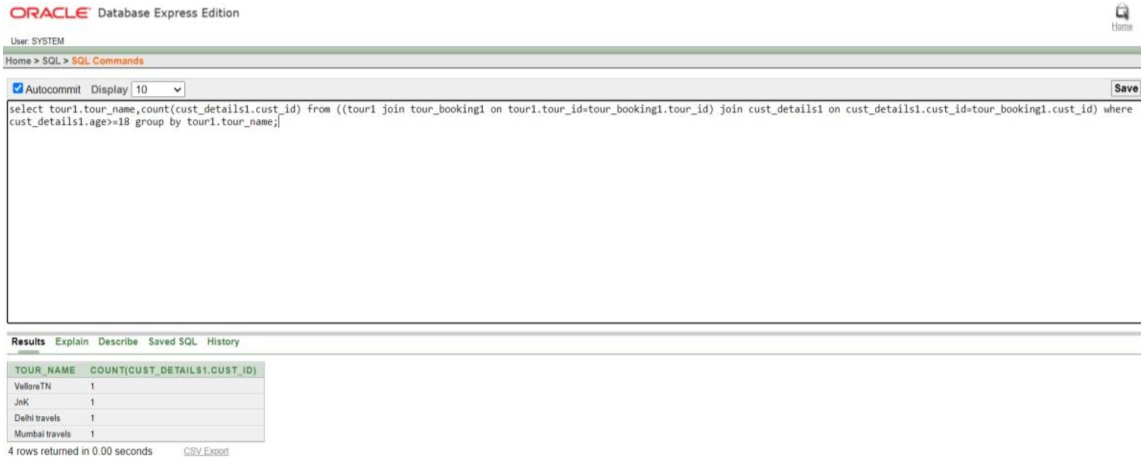

ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select * from tour1 where tour_id in (select tour_id from tour_booking1 where no_of_persons=(select min(no_of_persons) from tour_booking1));
```

TOUR_ID	TOUR_NAME	PLACE_FROM	PLACE_TO	PLACES_TO_BE	DURATION	FARE	DESCRIPTION
1	JnK	Jammu	Mumbai	Delhi	10	50000	Jammu to Mumbai

1 rows returned in 0.02 seconds

3. Count the total number of senior citizen for every tour name.

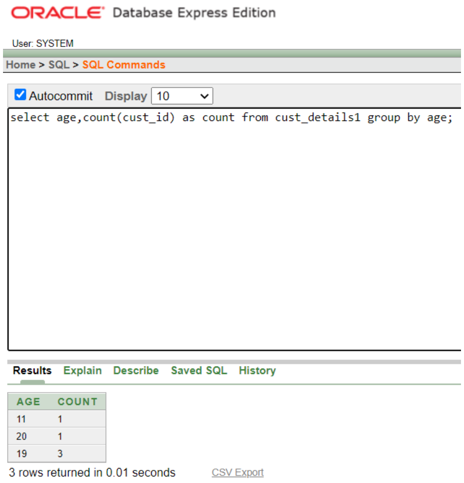

ORACLE Database Express Edition
Home
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Save
Results Explain Describe Saved SQL History
CSV Export

```
select tour1.tour_name,count(cust_details1.cust_id) from ((tour1 join tour_booking1 on tour1.tour_id=tour_booking1.tour_id) join cust_details1 on cust_details1.cust_id=tour_booking1.cust_id) where
cust_details1.age>=18 group by tour1.tour_name;
```

TOUR_NAME	COUNT(CUST_DETAILS1.CUST_ID)
VelloreTN	1
JnK	1
Delhi travels	1
Mumbai travels	1

4 rows returned in 0.00 seconds

4. Find the average age of the customer for every tour name.

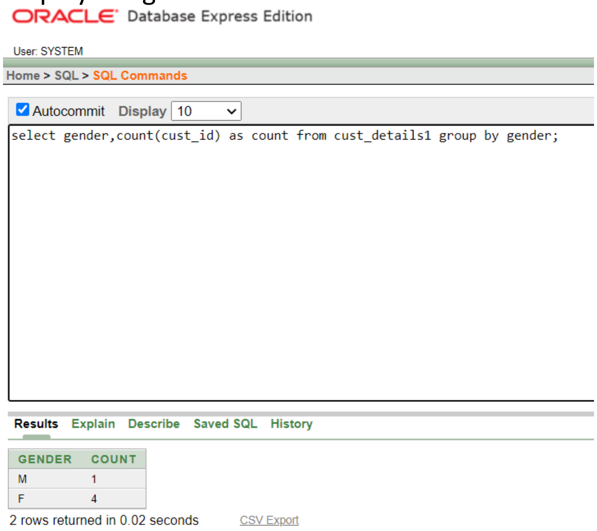

ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select age,count(cust_id) as count from cust_details1 group by age;
```

AGE	COUNT
11	1
20	1
19	3

3 rows returned in 0.01 seconds

5. Display the gender wise count of customers.

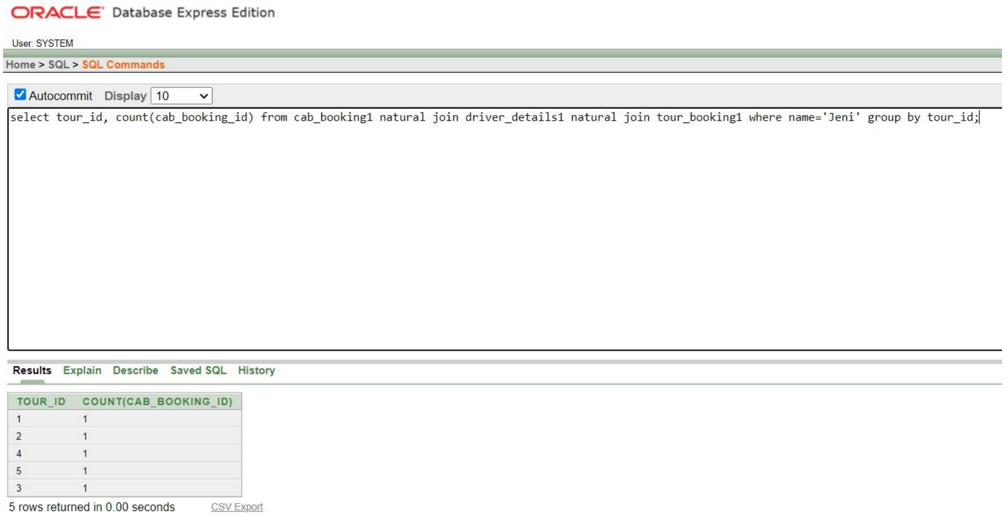

ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select gender,count(cust_id) as count from cust_details1 group by gender;
```

GENDER	COUNT
M	1
F	4

2 rows returned in 0.02 seconds

6. Count the number of drive given by the driver for a particular tour


ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
select tour_id, count(cab_booking_id) from cab_booking1 natural join driver_details1 natural join tour_booking1 where name='Jeni' group by tour_id;
```

TOUR_ID	COUNT(CAB_BOOKING_ID)
1	1
2	1
4	1
5	1
3	1

5 rows returned in 0.00 seconds