

Airline Flight Management :

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id, From_location, To_Location, Month Name as "Month_Name" and average price as "Average_Price"

Display the records sorted in ascending order based on flight id and then by Month Name.

15 rows

```
select f.flight_id,f.from_location,f.to_location,monthname(fd.flight_departure_date) as  
Month_name,avg(fd.price) as Average_price from air_flight f join air_flight_details fd  
on f.flight_id=fd.flight_id group by f.flight_id,Month_name order by f.flight_id,Month_name;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	MONTH_NAME	AVERAGE_PRICE
1011	HYDERABAD	CHENNAI	APRIL	4614.000000
1011	HYDERABAD	CHENNAI	MAY	3855.500000
1262	HYDERABAD	CHENNAI	MAY	3444.500000
1265	CHENNAI	HYDERABAD	APRIL	4086.000000
1265	CHENNAI	HYDERABAD	MAY	3303.666667
289	CHENNAI	KOCHI	MAY	3257.750000
3004	BENGALURU	CHENNAI	MAY	3319.666667

3013	CHENNAI	BENGALURU	MAY	3257.750000
3148	CHENNAI	BENGALURU	JUNE	2773.000000
3148	CHENNAI	BENGALURU	MAY	3052.000000
3241	CHENNAI	KOCHI	MAY	3303.666667
3244	KOCHI	CHENNAI	MAY	3371.500000
3307	BENGALURU	CHENNAI	MAY	3309.000000
916	CHENNAI	HYDERABAD	APRIL	4086.000000
916	CHENNAI	HYDERABAD	MAY	3570.666667

2. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets"

Display the records sorted in ascending order based on customer's first name.

1 row

```
select apf.profile_id, apf.first_name, apf.address, count(ati.ticket_id) as No_of_Tickets
from air_passenger_profile apf
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id having
```

```
count(ati.ticket_id) <=all
```

```
(select count(ati.ticket_id) from air_passenger_profile apf
```

```
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id) order by
```

```
first_name;
```

PROFILE_ID	FIRST_NAME	ADDRESS	NO_OF_TICKETS
PFL008	GANESH	45 3RD ST,HYDERABAD- 24	1

3. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as "Month_Name" and number of flight services as "No_of_Services".

Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight.

The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name

9 rows

```
select af.from_location,af.to_location,monthname(afd.flight_departure_date)
as Month_Name,
count(afd.flight_departure_date) as No_of_Services from air_flight af join
air_flight_details afd
on af.flight_id=afd.flight_id group by
af.from_location,af.to_location,month_name order by
from_location,to_location,month_name;
```

FROM_LOCATION	TO_LOCATION	MONTH_NAME	NO_OF_SERVICES
BENGALURU	CHENNAI	MAY	7

CHENNAI	BENGALURU	JUNE	1
CHENNAI	BENGALURU	MAY	6
CHENNAI	HYDERABAD	APRIL	2
CHENNAI	HYDERABAD	MAY	6
CHENNAI	KOCHI	MAY	7
HYDERABAD	CHENNAI	APRIL	1
HYDERABAD	CHENNAI	MAY	4
KOCHI	CHENNAI	MAY	2

4. Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets booked as "No_of_Tickets"

Display the records in ascending order based on customer's first name.

1 row

```

select app.profile_id,app.first_name,app.address,count(ati.ticket_id) as No_of_Tickets
from air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on ati.flight_id=af.flight_id
where af.airline_name= 'ABC Airlines' group by app.profile_id

```

```

having count(ati.ticket_id) >= all (select count(ati.ticket_id) from air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on ati.flight_id=af.flight_id
where af.airline_name= 'ABC Airlines' group by app.profile_id) order by app.first_name;

```

PROFILE_ID	FIRST_NAME	ADDRESS	NO_OF_TICKETS
PFL009	RAM	119 2ND CROSS ST,ERNAKULAM-12	8

5. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id, first_name, last_name, Flight_Id, Departure_Date and number of tickets booked as "No_of_Tickets".

Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

3 rows

```

select
ati.profile_id, app.first_name, app.last_name, ati.flight_id, ati.flight_departure_date, count(ati.ticket_id)
as No_of_Tickets from air_ticket_info ati join air_passenger_profile app on ati.profile_id=
app.profile_id join air_flight af on ati.flight_id=af.flight_id
where af.from_location='chennai' and af.to_location='hyderabad' group by ati.profile_id,
ati.flight_id, ati.flight_departure_date order by
ati.profile_id,
ati.flight_id, ati.flight_departure_date;

```

PROFILE_ID	FIRST_NAME	LAST_NAME	FLIGHT_ID	FLIGHT_DEPARTURE_DATE	NO_OF_TICKETS
------------	------------	-----------	-----------	-----------------------	---------------

PFL001	LATHA	SANKAR	1265	2013-04-29	1
PFL004	AARTHI	RAMESH	1265	2013-05-29	1
PFL005	SIVA	KUMAR	916	2013-05-06	2

6. Write a query to display flight id, from location, to location and ticket price of flights whose departure is in the month of april.

3 rows

Display the records sorted in ascending order based on flight id and then by from location.

```
select af.flight_id, af.from_location, af.to_location, afd.price from
air_flight af
join air_flight_details afd on af.flight_id=afd.flight_id
where monthname(afd.flight_departure_date)='april' order by
flight_id, from_location;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	PRICE
1011	HYDERABAD	CHENNAI	4614.00
1265	CHENNAI	HYDERABAD	4086.00
916	CHENNAI	HYDERABAD	4086.00

7. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight_id, from_location, to_location and Average price as "Price".

Display the records sorted in ascending order based on flight id and then by from_location and then by to_location.

11 rows

```
select af.flight_id,af.from_location,af.to_location,avg(afd.price)
from air_flight af join air_flight_details afd
on af.flight_id=afd.flight_id group by af.flight_id,af.from_location,af.to_location
order by af.flight_id,af.from_location,af.to_location;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	PRICE
1011	HYDERABAD	CHENNAI	4108.333333
1262	HYDERABAD	CHENNAI	3444.500000
1265	CHENNAI	HYDERABAD	3499.250000
289	CHENNAI	KOCHI	3257.750000
3004	BENGALURU	CHENNAI	3319.666667
3013	CHENNAI	BENGALURU	3257.750000
3148	CHENNAI	BENGALURU	2959.000000
3241	CHENNAI	KOCHI	3303.666667
3244	KOCHI	CHENNAI	3371.500000
3307	BENGALURU	CHENNAI	3309.000000
916	CHENNAI	HYDERABAD	3699.500000

8. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (combine first_name & last_name with comma in b/w), address of the customer.

Give an alias to the name as customer_name.

Hint: Query should fetch unique customers irrespective of multiple tickets booked.

Display the records sorted in ascending order based on profile id.

3 rows

```
select app.profile_id, concat(app.first_name,',',app.last_name) as customer_name,app.address
from air_passenger_profile app join air_ticket_info ati on app.profile_id=ati.profile_id
join air_flight af on ati.flight_id=af.flight_id where af.from_location='chennai'
and af.to_location='hyderabad' group by app.profile_id order by app.profile_id;
```

ROFILE_ID	CUSTOMER_NAME	ADDRESS
PFL001	LATHA,SANKAR	123 BROAD CROSS ST,CHENNAI-48
PFL004	AARTHI,RAMESH	343 6TH STREET,HYDERABAD- 76
PFL005	SIVA,KUMAR	125 8TH STREET,CHENNAI-46

9. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets.

In case of multiple records, display the records sorted in ascending order based on profile id.

2 rows


```
select profile_id from air_ticket_info group by profile_id having
count(ticket_id) >= all (select count(ticket_id)

from air_ticket_info group by profile_id) order by profile_id;
```

PROFILE_ID
PFL002
PFL007

10. Write a query to display the total number of tickets as “No_of_Tickets” booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets.

Display only the flights in which atleast 1 ticket is booked.

Display the records sorted in ascending order based on flight id.

7 rows

```
select af.flight_id,af.from_location,af.to_location,count(ati.ticket_id) as
No_of_Tickets
from air_flight af join air_ticket_info ati on af.flight_id=ati.flight_id
group by af.flight_id having count(ati.ticket_id) >= 1;
```

IGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_TICKETS
1011	HYDERABAD	CHENNAI	4
1262	HYDERABAD	CHENNAI	1
1265	CHENNAI	HYDERABAD	2

3004	BENGALURU	CHENNAI	3
3148	CHENNAI	BENGALURU	7
3244	KOCHI	CHENNAI	7
916	CHENNAI	HYDERABAD	2

11. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as "No_of_Services" and the cost as "Total_Price" in the same order.

Order the result by Total Price in descending order and then by flight_id in descending order.

Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

11 rows

```
select af.flight_id, count(afd.flight_departure_date) as No_of_Services, sum(afd.price) as
Total_Price from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id
group by flight_id
order by total_price desc, flight_id desc;
```

FLIGHT_ID	NO_OF_SERVICES	TOTAL_PRICE
916	4	14798.00
1265	4	13997.00

3307	4	13236.00
3013	4	13031.00
289	4	13031.00
1011	3	12325.00
3004	3	9959.00
3241	3	9911.00
3148	3	8877.00
1262	2	6889.00
3244	2	6743.00

12. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as "No_of_Passengers" in the same order.

Display the records sorted in ascending order based on flight id and then by flight departure date.

9 rows

```

SELECT flight_id,
       flight_departure_date,
       COUNT(ticket_id) AS No_of_Passengers
FROM   air_ticket_info
GROUP BY flight_id,

```

flight_departure_date

ORDER BY flight_id, flight_departure_date;t

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	NO_OF_PASSENGERS
1011	2013-05-09	4
1262	2013-05-20	1
1265	2013-04-29	1
1265	2013-05-29	1
3004	2013-05-02	3
3148	2013-05-21	2
3148	2013-06-01	5
3244	2013-05-03	7
916	2013-05-06	2

13. Write a query to display profile id of passenger(s) who booked minimum number of tickets.

In case of multiple records, display the records sorted in ascending order based on profile id.

1 row

```
select profile_id from air_ticket_info group by profile_id having count(profile_id) <= all
```

```
(select count(profile_id) from air_ticket_info group by profile_id) order by profile_id;
```

PROFILE_ID
PFL008

14. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI.

Display the records sorted in ascending order based on profile id.

4 rows

```
select distinct ati.profile_id, app.first_name, app.mobile_number, app.email_id
from air_ticket_info
ati join air_passenger_profile app on ati.profile_id=app.profile_id join air_flight af
on ati.flight_id=af.flight_id
where af.from_location='hyderabad' and af.to_location='chennai' order by profile_id;
```

PROFILE_ID	FIRST_NAME	MOBILE_NUMBER	EMAIL_ID
PFL001	LATHA	9876543210	LATHA@GMAIL.COM
PFL004	AARTHI	9595652530	AARTHI@GMAIL.COM
PFL005	SIVA	9884416986	SIVA@GMAIL.COM
PFL008	GANESH	9375237890	GANESH@GMAIL.COM

15. Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger's profile_id, first_name, last_name, flight_id, flight_departure_date, actual departure time, actual arrival time, delayed departure time as "Delayed_Departure_Time", delayed arrival time as "Delayed_Arrival_Time". Hint: Distinct Profile ID should be displayed irrespective of multiple tickets booked by the same profile.

Display the records sorted in ascending order based on passenger's profile id.

1 row

```
select distinct app.profile_id, app.first_name, app.last_name, ati.flight_id, ati.flight_departure_date,
af.departure_time, af.arrival_time, af.departure_time, ADDTIME(af.departure_time, '1:00:00') as
Delayed_Departure_Time,
ADDTIME(af.arrival_time, '1:00:00') as Delayed_Arrival_Time from air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id join air_flight af on
ati.flight_id=af.flight_id where ati.flight_departure_date='2013-05-06' order by app.profile_id;
```

PROFILE_ ID	FIRST _NAME	LAST_NAME	FLIGHT _ID	FLIGHT_ DEPARTURE _DATE	DEPARTURE_TIME	ARRIVAL _TIME
PFL005	SIVA	KUMAR	916	2013-05-06	19:55:00	21:00:00

DELAYED_DEPARTURE_TIME	DELAYED_ARRIVAL_TIME
20:55:00	22:00:00

16. Write a query to display the number of tickets as "No_of_Tickets" booked by Kochi Customers. The Query should display the Profile_Id, First_Name, Base_Location and number of tickets booked.

Hint: Use String functions to get the base location of customer from their Address and give alias name as "Base_Location"

Display the records sorted in ascending order based on customer first name.

2 rows

```
select
ap.profile_id,ap.first_name,substring_index(substring_index(ap.address,',',-1),'-',1)
as base_location,count(at.ticket_id) as No_of_Tickets from
air_passenger_profile ap join air_ticket_info at
on at.profile_id=ap.profile_id
where substring_index(substring_index(ap.address,',',-1),'-',1) ='kochi'
group by ap.profile_id order by first_name
```

PROFILE_ID	FIRST_NAME	BASE_LOCATION	NO_OF_TICKETS
PFL003	AMIT	KOCHI	3
PFL006	RAMESH	KOCHI	4

17. Write a query to display the flight_id, from_location, to_location, number of Services as "No_of_Services" offered in the month of May.

Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

Display the records sorted in ascending order based on flight id.

11 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date)
as No_of_Services from air_flight af join air_flight_details afd
```

on af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05'

group by flight_id order by flight_id;

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES
1011	HYDERABAD	CHENNAI	2
1262	HYDERABAD	CHENNAI	2
1265	CHENNAI	HYDERABAD	3
289	CHENNAI	KOCHI	4
3004	BENGALURU	CHENNAI	3
3013	CHENNAI	BENGALURU	4
3148	CHENNAI	BENGALURU	2
3241	CHENNAI	KOCHI	3
3244	KOCHI	CHENNAI	2
3307	BENGALURU	CHENNAI	4
916	CHENNAI	HYDERABAD	3

18. Write a query to display profile id, last name, mobile number and email id of passengers whose base location is chennai.

Display the records sorted in ascending order based on profile id.

2 rows

```
select profile_id,last_name,mobile_number,email_id from air_passenger_profile where  
substring_index(substring_index(address,',',-1),'-',1)='chennai'  
order by profile_id;
```

PROFILE_ID	LAST_NAME	MOBILE_NUMBER	EMAIL_ID
PFL001	SANKAR	9876543210	LATHA@GMAIL.COM
PFL005	KUMAR	9884416986	SIVA@GMAIL.COM

18. Write a query to display number of flights between 6.00 AM and 6.00 PM from chennai. Hint Use FLIGHT_COUNT as alias name.

1 row

```
select count(flight_id) as FLIGHT_COUNT from air_flight where departure_time between  
'6:00:00' and '18:00:00' and from_location='chennai';;
```

FLIGHT_COUNT
3

19. Write a query to display unique profile id, first name, email id and contact number of passenger(s) who travelled on flight with id 3148. Display the records sorted in ascending order based on first name.

2 rows

```
select distinct app.profile_id,app.first_name,app.email_id,app.mobile_number from  
air_passenger_profile app
```

```
join air_ticket_info ati on app.profile_id=ati.profile_id
```

```
where ati.flight_id= 3148 group by app.first_name order by app.first_name;
```

PROFILE_ID	FIRST_NAME	EMAIL_ID	MOBILE_NUMBER
PFL002	ARUN	ARUN@AOL.COM	8094564243
PFL007	GAYATHRI	GAYATHRI@GMAIL.COM	8073245678

20. Write a query to display the flights available in Morning, Afternoon, Evening & Night. The Query should display the Flight_Id, From_Location, To_Location, Departure_Time, time of service as "Time_of_Service".

Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning, 12:00:01 to 18:00:00 Hrs - Afternoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - Night

Display the records sorted in ascending order based on flight id.

11 rows

```
select flight_id,from_location,to_location,departure_time,  
case when departure_time between '05:00:01' and '12:00:00' then 'Morning'  
when departure_time between '12:00:01' and '18:00:00' then 'Afternoon'  
when departure_time between '18:00:01' and '24:00:00' then 'Evening'  
when departure_time between '00:00:01' and '05:00:00' then 'Night'  
end as Time_of_Service  
from air_flight order by flight_id;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	DEPARTURE_TIME	TIME_OF_SERVICE
1011	HYDERABAD	CHENNAI	12:30:00	AFTERNOON
1262	HYDERABAD	CHENNAI	06:00:00	MORNING
1265	CHENNAI	HYDERABAD	21:25:00	EVENING
289	CHENNAI	KOCHI	08:40:00	MORNING
3004	BENGALURU	CHENNAI	09:05:00	MORNING
3013	CHENNAI	BENGALURU	07:40:00	MORNING
3148	CHENNAI	BENGALURU	20:15:00	EVENING
3241	CHENNAI	KOCHI	10:40:00	MORNING
3244	KOCHI	CHENNAI	21:10:00	EVENING
3307	BENGALURU	CHENNAI	18:45:00	EVENING
916	CHENNAI	HYDERABAD	19:55:00	EVENING

21. Please follow instructions given below.

Write a query to display flight id, departure date, flight type of all flights. Flight type can be identified based on the following rules : if ticket price is less than 3000 then 'AIR PASSENGER', ticket price between 3000 and less than 4000 'AIR BUS' and ticket price between 4000 and greater than 4000 then 'EXECUTIVE PASSENGER'. Hint use FLIGHT_TYPE as alias name.

Display the records sorted in ascending order based on flight_id and then by departure date.

36 rows

```
select flight_id,flight_departure_date,  
  
case when price<3000 then 'AIR PASSENGER'  
  
when price>=3000 and price<=4000 then 'AIR BUS'  
  
when price>4000 then 'EXECUTIVE PASSENGER'  
  
end as FLIGHT_TYPE from air_flight_details order by flight_id,flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	FLIGHT_TYPE
1011	2013-04-30	EXECUTIVE PASSENGER
1011	2013-05-09	EXECUTIVE PASSENGER
1011	2013-05-21	AIR BUS
1262	2013-05-20	AIR BUS
1262	2013-05-29	AIR BUS
1265	2013-04-29	EXECUTIVE PASSENGER
1265	2013-05-14	AIR BUS
1265	2013-05-18	EXECUTIVE PASSENGER
1265	2013-05-29	AIR PASSENGER

289	2013-05-06	AIR BUS
289	2013-05-08	AIR BUS
289	2013-05-20	AIR BUS
289	2013-05-31	AIR PASSENGER
3004	2013-05-02	AIR BUS
3004	2013-05-19	AIR BUS
3004	2013-05-24	AIR BUS
3013	2013-05-04	AIR BUS
3013	2013-05-06	AIR BUS
3013	2013-05-22	AIR BUS
3013	2013-05-30	AIR PASSENGER
3148	2013-05-16	AIR BUS
3148	2013-05-21	AIR BUS
3148	2013-06-01	AIR PASSENGER

3241	2013-05-01	EXECUTIVE PASSENGER
3241	2013-05-13	AIR BUS
3241	2013-05-27	AIR PASSENGER
3244	2013-05-03	AIR BUS
3244	2013-05-15	AIR BUS
3307	2013-05-03	AIR BUS
3307	2013-05-03	AIR BUS
3307	2013-05-23	AIR BUS
3307	2013-05-29	AIR BUS
916	2013-04-28	EXECUTIVE PASSENGER
916	2013-05-01	EXECUTIVE PASSENGER
916	2013-05-06	AIR BUS
916	2013-05-12	AIR BUS

22.Please follow instructions given below.

Write a query to display the credit card type and no of credit cards used on the same type. Display the records sorted in ascending order based on credit card type.

Hint: Use CARD_COUNT AS Alias name for no of cards.

3 rows

```
SELECT CARD_TYPE,count(card_type) CARD_COUNT FROM air_credit_card_details group by CARD_TYPE
order by CARD_TYPE;
```

CARD_TYPE	CARD_COUNT
GOLD	3
INSTANT	2
PLATINIUM	3

23.Please follow instructions given below.

Write a Query to display serial no, first name,mobile number,email id of all the passengers who holds email address from gmail.com.

The Serial No will be the last three digits of profile ID.

Hint: Use SERIAL_NO as Alias name for serial number.

Display the records sorted in ascending order based on name.

6 rows

```
select substring(profile_id,4) as SERIAL_NO,first_name,mobile_number,email_id
from air_passenger_profile where email_id like '%gmail.com' order by first_name;
```

SERIAL_NO	FIRST_NAME	MOBILE_NUMBER	EMAIL_ID
004	AARTHI	9595652530	AARTHI@GMAIL.COM
008	GANESH	9375237890	GANESH@GMAIL.COM
007	GAYATHRI	8073245678	GAYATHRI@GMAIL.COM
001	LATHA	9876543210	LATHA@GMAIL.COM
006	RAMESH	9432198760	RAMESH@GMAIL.COM
005	SIVA	9884416986	SIVA@GMAIL.COM

24. Please follow instructions given below.

Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight_id, from_location, to_location, least number of Services as "No_of_Services" Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight

If there are multiple flights, display them sorted in ascending order based on flight id.

4 rows

```

select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as
No_of_Services from air_flight af join air_flight_details afd on
af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id
having count(afd.flight_departure_date)
<= all (select count(afd.flight_departure_date) from air_flight af join air_flight_details afd on
af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id)
order by af.flight_id;

```


LIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES
1011	HYDERABAD	CHENNAI	2
1262	HYDERABAD	CHENNAI	2
3148	CHENNAI	BENGALURU	2
3244	KOCHI	CHENNAI	2

25. Please follow instructions given below.

Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as "No_of_Flights".

Hint: Get the distinct from location and to location.

Display the records sorted in ascending order based on from location.

4 rows

```
select distinct from_location, count(to_location) as No_of_Flights from
air_flight

group by from_location order by from_location;
```

FROM_LOCATION	NO_OF_FLIGHTS
BENGALURU	2
CHENNAI	6
HYDERABAD	2

KOCHI	1
-------	---

26. Please follow instructions given below.

Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight_id, from_location, To_location, flight_departure_date and the number of passengers as "No_of_Passengers".

Hint: The Number of passengers inclusive of all the tickets booked with single profile id.

Display the records sorted in ascending order based on flight id and then by flight departure date.

9 rows

```
select af.flight_id,af.from_location,af.to_location,ati.flight_departure_date,count(ati.ticket_id)
as No_of_Passengers from air_flight af join air_ticket_info ati on af.flight_id=ati.flight_id
group by af.flight_id,ati.flight_departure_date order by af.flight_id,ati.flight_departure_date;
```

flight_id	from_location	to_location	flight_departure_date	No_of_Passengers
1011	HYDERABAD	CHENNAI	2013-05-09	3
1262	HYDERABAD	CHENNAI	2013-05-20	1
1265	CHENNAI	HYDERABAD	2013-04-29	1
1265	CHENNAI	HYDERABAD	2013-05-29	1
3004	BENGALURU	CHENNAI	2013-05-02	3
3148	CHENNAI	BENGALURU	2013-05-21	1
3148	CHENNAI	BENGALURU	2013-06-01	3
3244	KOCHI	CHENNAI	2013-05-03	7
916	CHENNAI	HYDERABAD	2013-05-06	2

27. Please follow instructions given below.

Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight_Id, From_Location, To_Location, Total_Seats, seats booked as "No_of_Seats_Booked".

Display the records sorted in ascending order based on flight id and then by No_of_Seats_Booked.

1 row

```
select af.flight_id,af.from_location,af.to_location,af.total_seats,(af.total_seats-afd.available_seats)
as No_of_Seats_Booked from air_flight af join air_flight_details afd on af.flight_id=
afd.flight_id where (af.total_seats-afd.available_seats)>(af.total_seats*0.1) group by flight_id order by
flight_id,No_of_Seats_Booked;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	TOTAL_SEATS	NO_OF_SEATS_BOOKED
3244	KOCHI	CHENNAI	50	7

28.Please follow instructions given below.

Write a query to display the Flight_Id, Flight_Departure_Date, From_Location,To_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

Display the records sorted in ascending order based on flight id and then by flight departure date.

14 rows

```
select af.flight_id,afd.flight_departure_date,af.from_location,af.to_location,af.duration
from air_flight af join air_flight_details afd on af.flight_id=afd.flight_id
where duration<'1:10:00' group by af.flight_id,afd.flight_departure_date
order by af.flight_id,afd.flight_departure_date;
```

FLIGHT_ID	FLIGHT_DEPARTURE_DATE	FROM_LOCATION	TO_LOCATION	DURATION
3013	2013-05-04	CHENNAI	BENGALURU	01:05:00

3013	2013-05-06	CHENNAI	BENGALURU	01:05:00
3013	2013-05-22	CHENNAI	BENGALURU	01:05:00
3013	2013-05-30	CHENNAI	BENGALURU	01:05:00
3148	2013-05-16	CHENNAI	BENGALURU	01:05:00
3148	2013-05-21	CHENNAI	BENGALURU	01:05:00
3148	2013-06-01	CHENNAI	BENGALURU	01:05:00
3307	2013-05-03	BENGALURU	CHENNAI	01:00:00
3307	2013-05-23	BENGALURU	CHENNAI	01:00:00
3307	2013-05-29	BENGALURU	CHENNAI	01:00:00
916	2013-04-28	CHENNAI	HYDERABAD	01:05:00
916	2013-05-01	CHENNAI	HYDERABAD	01:05:00
916	2013-05-06	CHENNAI	HYDERABAD	01:05:00
916	2013-05-12	CHENNAI	HYDERABAD	01:05:00

29. Please follow instructions given below.

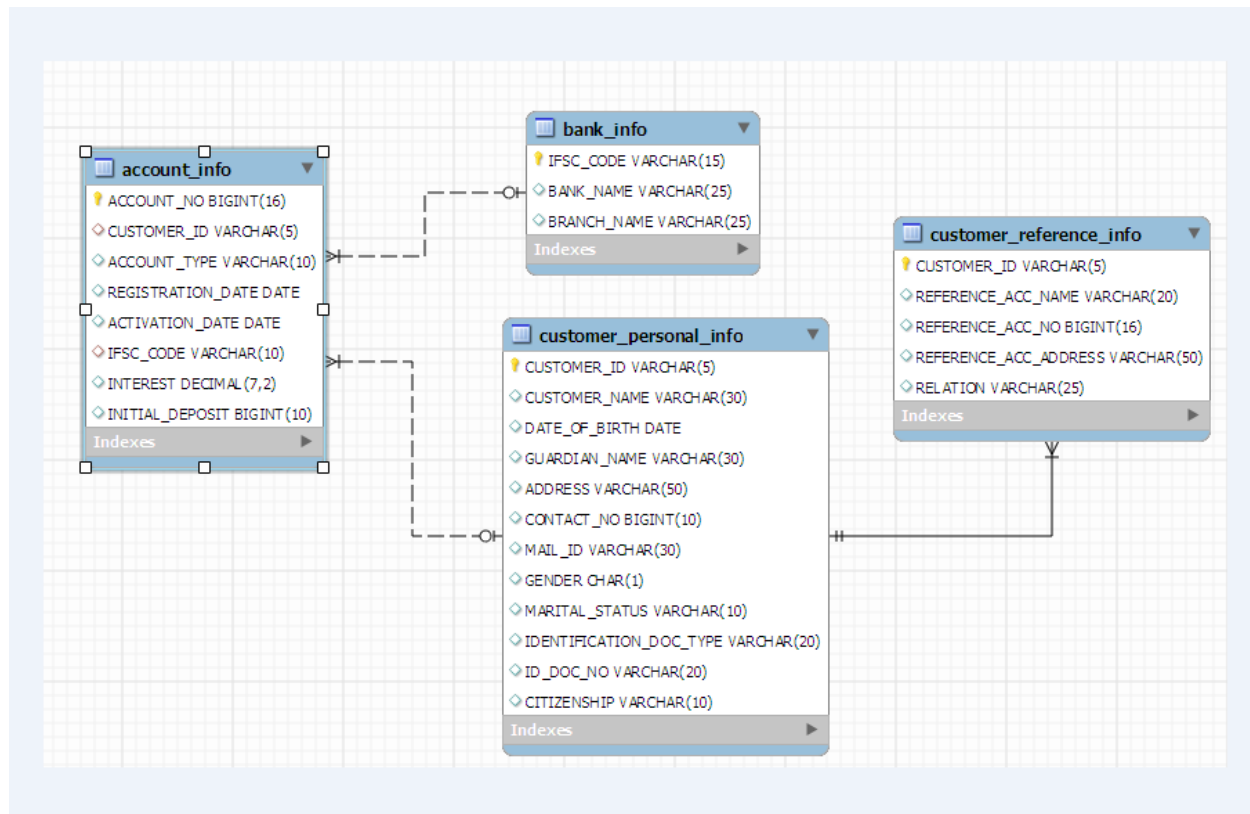
Write a query to display the flight_id, from_location, to_location, number of services as "No_of_Services", average ticket price as "Average_Price" whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

4 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as  
No_of_Services,  
avg(afd.price) as Average_Price from air_flight af join air_flight_details afd  
on af.flight_id=afd.flight_id group by af.flight_id having avg(afd.price)>  
(select avg(afd.price) from air_flight_details afd) order by afd.price;
```

FLIGHT_ID	FROM_LOCATION	TO_LOCATION	NO_OF_SERVICES	AVERAGE
1262	HYDERABAD	CHENNAI	2	3444.5000
1265	CHENNAI	HYDERABAD	4	3499.2500
916	CHENNAI	HYDERABAD	4	3699.5000
1011	HYDERABAD	CHENNAI	3	4108.3333

BANK MANGEMENT SYSTEM



DDL COMMANDS

```
create DATABASE BMS_DB;
```

```
use BMS_DB;
```

```
-- CUSTOMER_PERSONAL_INFO
```

```
CREATE TABLE CUSTOMER_PERSONAL_INFO
```

```
(
```

```
  CUSTOMER_ID VARCHAR(5),
```

```
  CUSTOMER_NAME VARCHAR(30),
```

```
DATE_OF_BIRTH DATE,  
GUARDIAN_NAME VARCHAR(30),  
ADDRESS VARCHAR(50),  
CONTACT_NO BIGINT(10),  
MAIL_ID VARCHAR(30),  
GENDER CHAR(1),  
MARITAL_STATUS VARCHAR(10),  
IDENTIFICATION_DOC_TYPE VARCHAR(20),  
ID_DOC_NO VARCHAR(20),  
CITIZENSHIP VARCHAR(10),  
CONSTRAINT CUST_PERS_INFO_PK PRIMARY KEY(CUSTOMER_ID)  
);
```

```
-- CUSTOMER_REFERENCE_INFO
```

```
CREATE TABLE CUSTOMER_REFERENCE_INFO  
(  
    CUSTOMER_ID VARCHAR(5),  
    REFERENCE_ACC_NAME VARCHAR(20),  
    REFERENCE_ACC_NO BIGINT(16),  
    REFERENCE_ACC_ADDRESS VARCHAR(50),  
    RELATION VARCHAR(25),  
    CONSTRAINT CUST_REF_INFO_PK PRIMARY KEY(CUSTOMER_ID),  
    CONSTRAINT CUST_REF_INFO_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES  
    CUSTOMER_PERSONAL_INFO(CUSTOMER_ID)  
);
```

```
-- BANK_INFO
```

```
CREATE TABLE BANK_INFO
(
    IFSC_CODE VARCHAR(15),
    BANK_NAME VARCHAR(25),
    BRANCH_NAME VARCHAR(25),
    CONSTRAINT BANK_INFO_PK PRIMARY KEY(IFSC_CODE)
);
```

-- ACCOUNT_INFO

```
CREATE TABLE ACCOUNT_INFO
(
    ACCOUNT_NO BIGINT(16),
    CUSTOMER_ID VARCHAR(5),
    ACCOUNT_TYPE VARCHAR(10),
    REGISTRATION_DATE DATE,
    ACTIVATION_DATE DATE,
    IFSC_CODE VARCHAR(10),
    INTEREST DECIMAL(7,2),
    INITIAL_DEPOSIT BIGINT(10),
    CONSTRAINT ACC_INFO_PK PRIMARY KEY(ACCOUNT_NO),
    CONSTRAINT ACC_INFO_PERS_FK FOREIGN KEY(CUSTOMER_ID) REFERENCES
    CUSTOMER_PERSONAL_INFO(CUSTOMER_ID),
    CONSTRAINT ACC_INFO_BANK_FK FOREIGN KEY(IFSC_CODE) REFERENCES BANK_INFO(IFSC_CODE)
);
```


DML COMMANDS

-- BANK_INFO

```
INSERT INTO
BANK_INFO(IFSC_CODE,BANK_NAME,BRANCH_NAME)VALUES('HDVL0012','HDFC','VALASARAVAKKAM')
;
```

```
INSERT INTO BANK_INFO(IFSC_CODE,BANK_NAME,BRANCH_NAME)
VALUES('SBITN0123','SBI','TNAGAR');
```

```
INSERT INTO BANK_INFO(IFSC_CODE,BANK_NAME,BRANCH_NAME)
VALUES('ICITN0232','ICICI','TNAGAR');
```

```
INSERT INTO BANK_INFO(IFSC_CODE,BANK_NAME,BRANCH_NAME)
VALUES('ICIPG0242','ICICI','PERUNGUDI');
```

```
INSERT INTO BANK_INFO(IFSC_CODE,BANK_NAME,BRANCH_NAME)
VALUES('SBISD0113','SBI','SAIDAPET');
```

-- CUSTOMER_PERSONAL_INFO

```
INSERT INTO
CUSTOMER_PERSONAL_INFO(CUSTOMER_ID,CUSTOMER_NAME,DATE_OF_BIRTH,GUARDIAN_NAME,A
DDRESS,CONTACT_NO,MAIL_ID,GENDER,MARITAL_STATUS,IDENTIFICATION_DOC_TYPE,ID_DOC_NO,CIT
IZENSHIP) VALUES('C-001','JOHN','1984-05-03','PETER','NO.14, ST.MARKS
ROAD,BANGALORE','9734526719','JOHN_123@gmail.com','M','SINGLE','PASSPORT','PASS123','INDIAN');
```

```
INSERT INTO
CUSTOMER_PERSONAL_INFO(CUSTOMER_ID,CUSTOMER_NAME,DATE_OF_BIRTH,GUARDIAN_NAME,A
DDRESS,CONTACT_NO,MAIL_ID,GENDER,MARITAL_STATUS,IDENTIFICATION_DOC_TYPE,ID_DOC_NO,CIT
IZENSHIP) VALUES('C-002','JAMES','1984-08-06','GEORGE','NO.18, MG
ROAD,BANGALORE','9237893481','JAMES_123@gmail.com','M','MARRIED','PASSPORT','PASS124','INDIAN
');
```

```
INSERT INTO
CUSTOMER_PERSONAL_INFO(CUSTOMER_ID,CUSTOMER_NAME,DATE_OF_BIRTH,GUARDIAN_NAME,A
DDRESS,CONTACT_NO,MAIL_ID,GENDER,MARITAL_STATUS,IDENTIFICATION_DOC_TYPE,ID_DOC_NO,CIT
IZENSHIP) VALUES('C-003','SUNITHA','1984-11-06','VINOD','NO.21, GM
ROAD,CHENNAI','9438978389','SUNITHA_123@gmail.com','F','SINGLE','VOTER-ID','GMV1234','INDIAN');
```

```
INSERT INTO
CUSTOMER_PERSONAL_INFO(CUSTOMER_ID,CUSTOMER_NAME,DATE_OF_BIRTH,GUARDIAN_NAME,A
```

```
DDRESS,CONTACT_NO,MAIL_ID,GENDER,MARITAL_STATUS,IDENTIFICATION_DOC_TYPE,ID_DOC_NO,CITIZENSHIP) VALUES('C-004','RAMESH','1985-12-11','KRISHNAN','NO.14,LB ROAD,CHENNAI',9235234534,'RAMESH_123@gmail.com','M','MARRIED','PASSPORT','PASS125','INDIAN');
```

```
INSERT INTO  
CUSTOMER_PERSONAL_INFO(CUSTOMER_ID,CUSTOMER_NAME,DATE_OF_BIRTH,GUARDIAN_NAME,ADDRESS,CONTACT_NO,MAIL_ID,GENDER,MARITAL_STATUS,IDENTIFICATION_DOC_TYPE,ID_DOC_NO,CITIZENSHIP) VALUES('C-005','KUMAR','1983-04-26','KIRAN','NO.18,MM ROAD,BANGALORE',9242342312,'KUMAR_123@gmail.com','M','SINGLE','PASSPORT','PASS126','INDIAN');
```

-- CUSTOMER_REFERENCE_INFO

```
INSERT INTO  
CUSTOMER_REFERENCE_INFO(CUSTOMER_ID,REFERENCE_ACC_NAME,REFERENCE_ACC_NO,REFERENCE_ACC_ADDRESS,RELATION) VALUES('C-001','RAM',0987654321122345,'NO.11,BRIGADE ROAD,BANGALORE','FRIEND');
```

```
INSERT INTO  
CUSTOMER_REFERENCE_INFO(CUSTOMER_ID,REFERENCE_ACC_NAME,REFERENCE_ACC_NO,REFERENCE_ACC_ADDRESS,RELATION) VALUES('C-002','RAGHUL',0987654321122346,'NO.21,CUNNGHAM ROAD,BANGALORE','FRIEND');
```

```
INSERT INTO  
CUSTOMER_REFERENCE_INFO(CUSTOMER_ID,REFERENCE_ACC_NAME,REFERENCE_ACC_NO,REFERENCE_ACC_ADDRESS,RELATION) VALUES('C-003','GOKUL',0987654321122357,'NO.12,OMR,CHENNAI','NEIGHBOUR');
```

```
INSERT INTO  
CUSTOMER_REFERENCE_INFO(CUSTOMER_ID,REFERENCE_ACC_NAME,REFERENCE_ACC_NO,REFERENCE_ACC_ADDRESS,RELATION) VALUES('C-004','RAHMAN',0987654321122348,'NO.35,ECR,CHENNAI','FRIEND');
```

```
INSERT INTO  
CUSTOMER_REFERENCE_INFO(CUSTOMER_ID,REFERENCE_ACC_NAME,REFERENCE_ACC_NO,REFERENCE_ACC_ADDRESS,RELATION) VALUES('C-005','VIVEK',0987654321122359,'NO.78,JAYA NAGAR,BANGALORE','NEIGHBOUR');
```

-- ACCOUNT_INFO

INSERT INTO

ACCOUNT_INFO(ACCOUNT_NO,CUSTOMER_ID,ACCOUNT_TYPE,REGISTRATION_DATE,ACTIVATION_DATE,IFSC_CODE,INTEREST, INITIAL_DEPOSIT) VALUES(1234567898765432,'C-001','SAVINGS','2012-02-23','2012-02-28','HDVL0012',5,10000);

INSERT INTO

ACCOUNT_INFO(ACCOUNT_NO,CUSTOMER_ID,ACCOUNT_TYPE,REGISTRATION_DATE,ACTIVATION_DATE,IFSC_CODE,INTEREST, INITIAL_DEPOSIT) VALUES(1234567898765433,'C-002','SALARY','2012-03-12','2012-03-17','SBITN0123',6,0);

INSERT INTO

ACCOUNT_INFO(ACCOUNT_NO,CUSTOMER_ID,ACCOUNT_TYPE,REGISTRATION_DATE,ACTIVATION_DATE,IFSC_CODE,INTEREST, INITIAL_DEPOSIT) VALUES(1234567898765434,'C-003','SAVINGS','2012-03-15','2012-03-20','ICITN0232',4,16000);

INSERT INTO

ACCOUNT_INFO(ACCOUNT_NO,CUSTOMER_ID,ACCOUNT_TYPE,REGISTRATION_DATE,ACTIVATION_DATE,IFSC_CODE,INTEREST, INITIAL_DEPOSIT) VALUES(1234567898765435,'C-004','SALARY','2012-04-05','2012-04-10','HDVL0012',7,0);

INSERT INTO

ACCOUNT_INFO(ACCOUNT_NO,CUSTOMER_ID,ACCOUNT_TYPE,REGISTRATION_DATE,ACTIVATION_DATE,IFSC_CODE,INTEREST, INITIAL_DEPOSIT) VALUES(1234567898765436,'C-005','SAVINGS','2012-04-12','2012-04-17','SBISD0113',8,20000);

QUESTIONS

1. Write a query which will display the customer id, account type they hold, their account number and bank name.
2. Write a query which will display the customer id, account type and the account number of HDFC customers who registered after 12-JAN-2012 and before 04-APR-2012.
3. Write a query which will display the customer id, customer name, account no, account type and bank name where the customers hold the account.

4. Write a query which will display the customer id, customer name, gender, marital status along with the unique reference string and sort the records based on customer id in descending order.
 Hint: Generate unique reference string as mentioned below:
 CustomerName_Gender_MaritalStatus
 Example,
 C-005 KUMAR M SINGLE KUMAR_M_SINGLE
 Use "UNIQUE_REF_STRING" as alias name for displaying the unique reference string.
5. Write a query which will display the account number, customer id, registration date, initial deposit amount of the customer whose initial deposit amount is within the range of Rs.15000 to Rs.25000.
6. Write a query which will display customer id, customer name, date of birth, guardian name of the customers whose name starts with 'J'.
7. Write a query which will display customer id, account number and passcode.
 Hint: To generate passcode, join the last three digits of customer id and last four digit of account number.
 Example
 C-001 1234567898765432 0015432
 Use "PASSCODE" as alias name for displaying the passcode.
8. Write a query which will display the customer id, customer name, date of birth, Marital Status, Gender, Guardian name, contact no and email id of the customers whose gender is male 'M' and marital status is MARRIED.
9. Write a query which will display the customer id, customer name, guardian name, reference account holders name of the customers who are referenced / referred by their 'FRIEND'.
10. Write a query to display the customer id, account number and interest amount in the below format with INTEREST_AMT as alias name. Sort the result based on the INTEREST_AMT in ascending order.
 Example: \$5Hint: Need to prefix \$ to interest amount and round the result without decimals.
11. Write a query which will display the customer id, customer name, account no, account type, activation date, bank name whose account will be activated on '10-APR-2012'
12. Write a query which will display account number, customer id, customer name, bank name, branch name, ifsc code, citizenship, interest and initial deposit amount of all the customers.
13. Write a query which will display customer id, customer name, date of birth, guardian name, contact number, mail id and reference account holder's name of the customers who has submitted the passport as an identification document.
14. Write a query to display the customer id, customer name, account number, account type, initial deposit, interest who have deposited maximum amount in the bank.
15. Write a query to display the customer id, customer name, account number, account type, interest, bank name and initial deposit amount of the customers who are getting maximum interest rate.
16. Write a query to display the customer id, customer name, account no, bank name, contact no and mail id of the customers who are from BANGALORE.
17. Write a query which will display customer id, bank name, branch name, ifsc code, registration date, activation date of the customers whose activation date is in the month of march (March 1'st to March 31'st).

18. Write a query which will calculate the interest amount and display it along with customer id, customer name, account number, account type, interest, and initial deposit amount. Hint : Formula for interest amount, calculate: $((\text{interest}/100) * \text{initial deposit amt})$ with column name 'interest_amt' (alias)
19. Write a query to display the customer id, customer name, date of birth, guardian name, contact number, mail id, reference name who has been referenced by 'RAGHUL'.
20. Write a query which will display the customer id, customer name and contact number with ISD code of all customers in below mentioned format. Sort the result based on the customer id in descending order. Format for contact number is :
"+91-3digits-3digits-4digits"
Example: +91-924-234-2312
Use "CONTACT_ISD" as alias name.
21. Write a query which will display account number, account type, customer id, customer name, date of birth, guardian name, contact no, mail id , gender, reference account holders name, reference account holders account number, registration date, activation date, number of days between the registration date and activation date with alias name "NoofdaysforActivation", bank name, branch name and initial deposit for all the customers.
22. Write a query which will display customer id, customer name, guardian name, identification doc type, reference account holders name, account type, ifsc code, bank name and current balance for the customers who has only the savings account.
Hint: Formula for calculating current balance is add the intital deposit amount and interest and display without any decimals. Use "CURRENT_BALANCE" as alias name.
23. Write a query which will display the customer id, customer name, account number, account type, interest, initial deposit; check the initial deposit, if initial deposit is 20000 then display "high", if initial deposit is 16000 display 'moderate', if initial deposit is 10000 display 'average', if initial deposit is 5000 display 'low', if initial deposit is 0 display 'very low' otherwise display 'invalid' and sort by interest in descending order.
Hint: Name the column as "Deposit_Status" (alias). Strictly follow the lower case for strings in this query.
24. Write a query which will display customer id, customer name, account number, account type, bank name, ifsc code, initial deposit amount and new interest amount for the customers whose name starts with "J".
Hint: Formula for calculating "new interest amount" is if customers account type is savings then add 10 % on current interest amount to interest amount else display the current interest amount. Round the new interest amount to 2 decimals. Use "NEW_INTEREST" as alias name for displaying the new interest amount.
Example, Assume Jack has savings account and his current interest amount is 10.00, then the new interest amount is 11.00 i.e $(10 + (10 * 10/100))$.

25. Write query to display the customer id, customer name, account no, initial deposit, tax percentage as calculated below. Hint: If initial deposit = 0 then tax is '0%' If initial deposit <= 10000 then tax is '3%' If initial deposit > 10000 and initial deposit <= 20000 then tax is '5%' If initial deposit > 20000 and initial deposit <= 30000 then tax is '7%' If initial deposit > 30000 then tax is '10%' Use the alias name 'taxPercentage'

ANSWERS

- 1) select a.Customer_ID, a.account_type, a.account_no, b.bank_name from account_info a join bank_info b on(a.ifsc_code = b.ifsc_code);
- 2) select Customer_ID, account_type, account_no from account_info a join bank_info b on(a.ifsc_code = b.ifsc_code) where b.bank_name = 'HDFC' and registration_date between '2012-01-12' and '2012-04-04';
- 3) select a.Customer_ID, c.Customer_Name, a.account_no, a.account_type, b.bank_name from account_info a join bank_info b on(a.ifsc_code = b.ifsc_code) join customer_personal_info c on(a.customer_id = c.customer_id);
- 4) select Customer_ID, Customer_Name, gender, marital_status, concat(customer_name, '_', gender, '_', marital_status) UNIQUE_REF_STRING from customer_personal_info;
- 5) select account_no, customer_id, registration_date, initial_deposit from account_info where initial_deposit between 15000 and 25000;
- 6) select Customer_ID, Customer_Name, date_of_birth, guardian_name from customer_personal_info where guardian_name like 'j%';
- 7) select Customer_ID, account_no, concat(substr(customer_id,3,5),substr(account_no,14,16)) passcode from account_info;
- 8) select Customer_ID, Customer_Name, date_of_birth, marital_status, gender, guardian_name, contact_no, mail_id from customer_personal_info where gender = 'm' and marital_status = 'married';
- 9) select c.customer_id, c.Customer_Name, c.guardian_name, r.reference_acc_name from customer_personal_info c join customer_reference_info r on(c.customer_id = r.customer_id) where relation = 'friend';
- 10) select customer_id, account_no, concat('\$',round(interest)) INTEREST_AMT from account_info group by interest;
- 11) select i.Customer_ID, i.Customer_Name, a.account_no, a.account_type, a.activation_date, b.bank_name from customer_personal_info i join account_info a on(i.customer_id = a.customer_id) join bank_info b on(a.ifsc_code = b.ifsc_code)

where a.activation_date = '2012-04-10';

12) select a.account_no, a.customer_id, c.customer_name, b.bank_name, b.branch_name, b.ifsc_code, c.citizenship, a.interest, a.initial_deposit from account_info a

join bank_info b on(a.ifsc_code = b.ifsc_code)

join customer_personal_info c on(a.customer_id = c.customer_id);

13) select a.customer_id, a.customer_name, a.date_of_birth, a.guardian_name, a.contact_no, a.mail_id, b.reference_acc_name from customer_personal_info a join customer_reference_info b on(a.customer_id = b.customer_id) where a.identification_doc_type = 'passport';

14) select b.customer_id, a.customer_name, b.account_no, b.account_type, b.initial_deposit, b.interest from customer_personal_info a join account_info b on(a.customer_id = b.customer_id) where b.initial_deposit = (select max(initial_deposit) from account_info);

15) select b.customer_id, a.customer_name, b.account_no, b.account_type, b.interest, c.bank_name, b.initial_deposit from account_info b join customer_personal_info a on(a.customer_id = b.customer_id) join bank_info c on(b.ifsc_code = c.ifsc_code) where b.interest = (select max(interest) from account_info);

16) select a.Customer_ID, a.customer_name, b.account_no, c.bank_name, a.contact_no, a.mail_id from customer_personal_info a join account_info b on(a.customer_id = b.customer_id) join bank_info c on(b.ifsc_code = c.ifsc_code) where a.address like '%bangalore';

17) select b.customer_id, a.bank_name, a.branch_name, a.ifsc_code, b.registration_date, b.activation_date from bank_info a join account_info b on(a.ifsc_code = b.ifsc_code) where b.activation_date like '%-03-%';

18) select a.customer_id, a.customer_name, b.account_no, b.account_type, b.interest, b.initial_deposit, ((b.interest/100)*b.initial_deposit) interest_amt from customer_personal_info a join account_info b on(a.customer_id = b.customer_id);

19) select a.customer_id, a.customer_name, a.date_of_birth, a.guardian_name, a.contact_no, a.mail_id, b.reference_acc_name from customer_personal_info a join customer_reference_info b on(a.customer_id = b.customer_id) where reference_acc_name = 'raghul';

20) select Customer_ID, Customer_Name, concat('+91-', substr(contact_no, 1, 3), '-', substr(contact_no, 4, 3), '-', substr(contact_no, 7, 4)) CONTACT_ISD from customer_personal_info;

21) select a.ACCOUNT_NO, a.ACCOUNT_TYPE, a.CUSTOMER_ID, b.CUSTOMER_NAME, b.DATE_OF_BIRTH, b.GUARDIAN_NAME, b.CONTACT_NO, b.MAIL_ID, b.GENDER, c.REFERENCE_ACC_NAME, c.REFERENCE_ACC_NO, a.REGISTRATION_DATE,

```

a.ACTIVATION_DATE, d.BANK_NAME, d.BRANCH_NAME, a.INITIAL_DEPOSIT,
(a.ACTIVATION_DATE-a.REGISTRATION_DATE) NoOfDaysForActivation from
account_info a join customer_personal_info b on(a.customer_id = b.customer_id)
join bank_info d on(a.ifsc_code = d.ifsc_code) join customer_reference_info c
on(b.customer_id = c.customer_id);
22) select a.CUSTOMER_ID, a.CUSTOMER_NAME, a.GUARDIAN_NAME,
a.IDENTIFICATION_DOC_TYPE, b.REFERENCE_ACC_NAME, c.ACCOUNT_TYPE,
c.IFSC_CODE, d.BANK_NAME,
round(c.initial_deposit+((c.interest/100)*c.initial_deposit)) current_balance from
customer_personal_info a join customer_reference_info b on(b.customer_id =
a.customer_id) join account_info c on(a.customer_id = c.customer_id)
join bank_info d on(c.ifsc_code = d.ifsc_code);
23) select a.CUSTOMER_ID, b.CUSTOMER_NAME, a.ACCOUNT_NO, a.ACCOUNT_TYPE,
a.INTEREST, CASE WHEN INITIAL_DEPOSIT = 20000 then 'high'
WHEN INITIAL_DEPOSIT = 16000 then 'moderate' WHEN INITIAL_DEPOSIT = 10000 THEN
'average' when INITIAL_DEPOSIT = 5000 then 'low' when initial_deposit = 0 then 'very
low' END as Deposit_Status from account_info a
join customer_personal_info b on(a.customer_id = b.customer_id);
24) select a.CUSTOMER_ID, b.CUSTOMER_NAME, a.ACCOUNT_NO, a.ACCOUNT_TYPE,
c.BANK_NAME, c.IFSC_CODE, a.INITIAL_DEPOSIT, if(ACCOUNT_TYPE = 'savings',
round(a.interest+(a.interest*(a.interest/100)),2), a.interest) as NEW_INTEREST
from account_info a join customer_personal_info b on(a.customer_id = b.customer_id)
join bank_info c on(a.ifsc_code = c.ifsc_code)
where CUSTOMER_NAME like 'j%';
25) select a.CUSTOMER_ID, b.customer_name, a.account_no, a.INITIAL_DEPOSIT,
case
when a.INITIAL_DEPOSIT = 0 then '0%'
when a.INITIAL_DEPOSIT <= 10000 then '3%'
when a.INITIAL_DEPOSIT > 10000 && a.INITIAL_DEPOSIT <= 20000 then '5%'
when a.INITIAL_DEPOSIT > 20000 && a.INITIAL_DEPOSIT <= 30000 then '7%'
when a.INITIAL_DEPOSIT > 30000 then '10%'
END as taxPercentage from account_info a
join customer_personal_info b on(a.customer_id = b.customer_id);

```


Bank Management System Queries:

1. Please follow instructions given below.

Write a query to display account number, customer's number, customer's firstname, lastname, account opening date.

Display the records sorted in ascending order based on account number.

```
SELECT account_number, am.customer_number, firstname, lastname, account_opening_date
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
ORDER BY account_number;
```

2. Please follow instructions given below.

Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```
SELECT count(customer_number) Cust_Count
FROM customer_master
WHERE customer_city='Delhi'
```

3. Please follow instructions given below.

Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number, firstname, account_number
```

```
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE extract(day from account_opening_date)>15
ORDER BY am.customer_number, account_number
```

4.Please follow instructions given below.

Write a query to display customer number, customer's first name, account number where the account status is terminated.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number,firstname, account_number
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE account_status='Terminated'
ORDER BY am.customer_number, account_number
```

5.Please follow instructions given below.

Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions.

Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type,count(transaction_number) Trans_Count
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
WHERE customer_number like '%001'
```

GROUP BY transaction_type

ORDER BY transaction_type

6. Please follow instructions given below.

Write a query to display the number of customers who have registration but no account in the bank.

Give the alias name as Count_Customer for number of customers.

```
SELECT count(customer_number) Count_Customer
FROM customer_master
WHERE customer_number NOT IN (SELECT customer_number FROM account_master)
```

7. Please follow instructions given below.

Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT td.account_number, opening_balance+sum(transaction_amount) Deposit_Amount
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
WHERE transaction_type='deposit'
GROUP BY account_number
ORDER BY account_number
```

8. Please follow instructions given below.

Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```
select branch_master.branch_city, count(account_master.account_number) as No_of_Accounts from  
branch_master left join account_master on account_master.branch_id=branch_master.branch_id  
  
group by branch_master.branch_city order by branch_city;
```

9.Please follow instructions given below.

Write a query to display the firstname of the customers who have more than 1 account. Display the records in sorted order based on firstname.

```
select firstname  
  
FROM customer_master cm INNER JOIN account_master am  
  
ON cm.customer_number=am.customer_number  
  
group by firstname  
  
having count(account_number)>1  
  
order by firstname;
```

10.Please follow instructions given below.

Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch.

Display the records sorted in order based on customer number.

```
SELECT Id.customer_number, firstname, lastname  
  
FROM customer_master cm INNER JOIN loan_details Id  
  
ON cm.customer_number=Id.customer_number
```

GROUP BY customer_number

HAVING count(branch_id)>1

ORDER BY customer_number

11.Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different.

Display the records sorted in ascending order based on customer number.

```
select customer_master.customer_number, firstname, customer_city, branch_city
from account_master inner join customer_master on account_master.customer_number =
customer_master.customer_number
inner join branch_master on account_master.branch_id = branch_master.branch_id
where customer_city != branch_city order by customer_master.customer_number;
```

12.Please follow instructions given below.

Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(Id.customer_number) Count
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number
WHERE cm.customer_number NOT IN ( SELECT customer_number FROM account_master)
```

13.Please follow instructions given below.

Write a query to display the account number who has done the highest transaction.

For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed.

In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT td.account_number
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number
having count(td.transaction_number)>=ALL
(SELECT count(td.transaction_number)
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number) order by am.account_number;
```

14. Please follow instructions given below.

Write a query to show the branch name, branch city where we have the maximum customers.

For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore.

In case of multiple records, display the records sorted in ascending order based on branch name.

```
select branch_name, branch_city
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name
having count(customer_number)>=ALL
```

```
(select count(customer_number)
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name) order by branch_name;
```

15. Please follow instructions given below.

Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well.

For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011.

Display the records sorted in ascending order based on account number.

```
SELECT td.account_number, sum(CASE WHEN transaction_type='Deposit' THEN transaction_amount
END)
+(SELECT opening_balance FROM account_master am2 where
am2.account_number=am.account_number) Deposit,
sum(CASE WHEN transaction_type='Withdrawal' THEN transaction_amount END) Withdrawal
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
GROUP BY td.account_number
HAVING Withdrawal > Deposit
ORDER BY am.account_number
```

16. Please follow instructions given below.

Write a query to show the balance amount for account number that ends with 001.

Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount.

For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000) – (500+500+1000).

```
SELECT (SUM(CASE WHEN transaction_type='Deposit'
THEN transaction_amount END)) -
(SUM(CASE WHEN transaction_type='Withdrawal'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001') AS Balance_Amount
FROM transaction_details where account_number like '%001'
```

17.Please follow instructions given below.

Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT cm.customer_number,firstname, am.account_number,count(transaction_number)
Count_Trans
FROM customer_master cm inner JOIN account_master am
ON cm.customer_number=am.customer_number
INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by am.account_number order by cm.customer_number, am.account_number
```

18.Please follow instructions given below.

Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```
SELECT firstname  
  
FROM customer_master INNER JOIN account_master  
  
ON customer_master.customer_number=account_master.customer_number  
  
GROUP BY firstname  
  
having count(firstname)>=2 order by firstname;
```

19.Please follow instructions given below.

Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches.

For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011.

So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT ld.customer_number, firstname, lastname  
  
FROM customer_master cm INNER JOIN loan_details ld  
  
ON cm.customer_number=ld.customer_number  
  
group by customer_number  
  
having count(branch_id)>=2 and sum(loan_amount)>=All(select sum(loan_amount) from loan_details  
group by customer_number)
```

20.Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans..

Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT Id.customer_number, firstname,branch_id, loan_amount
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number order by cm.customer_number, branch_id,
loan_amount
```

21.Please follow instructions given below.

Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.

Display the records sorted in ascending order based on city name.

```
SELECT branch_city, count(branch_id) Count_Branch
FROM branch_master
GROUP BY branch_city
ORDER BY branch_city
```

22.Please follow instructions given below.

Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active.

Display the records sorted in ascending order based on account id /account number.

```
SELECT account_number, firstname, lastname
FROM customer_master cm INNER JOIN account_master am
```

ON cm.customer_number=am.customer_number

WHERE account_status='Active'

ORDER BY account_number

23.Please follow instructions given below.

Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name.

Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,coalesce(middlename,lastname) Middle_Name
```

```
FROM customer_master order by customer_number
```

24.Please follow instructions given below.

Write a query to display the customer number , firstname, customer's date of birth . Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth
```

```
FROM customer_master order by year(customer_date_of_birth), firstname;
```

25.Please follow instructions given below.

Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student.

Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT firstname, customer_city,account_number
```

FROM customer_master cm INNER JOIN account_master am

ON cm.customer_number=am.customer_number

**WHERE occupation !='Service' and occupation != 'Student' and occupation != 'Business' order by
firstname, account_number**

Bank Management System Queries:

1. Please follow instructions given below.

Write a query to display account number, customer's number, customer's firstname, lastname, account opening date.

Display the records sorted in ascending order based on account number.

```
SELECT account_number, am.customer_number, firstname, lastname, account_opening_date
FROM customer_master cm JOIN account_master am
ON cm.customer_number=am.customer_number
ORDER BY account_number;
```

ACCOUNT_NUMBER	CUSTOMER_NUMBER	FIRSTNAME	LASTNAME	ACCOUNT_OPENING_DATE
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27
A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09

A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2.Please follow instructions given below.

Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```
SELECT count(customer_number) Cust_Count
FROM customer_master
WHERE customer_city='Delhi'
```

CUST_COUNT
4

3.Please follow instructions given below.

Write a query to display the customer number, customer firstname,account number for the customer's whose accounts were created after 15th of any month.

Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT am.customer_number, firstname, account_number
FROM customer_master cm JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE day(account_opening_date)>15
```

ORDER BY am.customer_number, account_number

CUSTOMER_NUMBER	FIRSTNAMEMiddle_Name	ACCOUNT_NUMBER
C00002	AVINASH	A00004
C00003	RAHUL	A00003
C00003	RAHUL	A00009
C00006	CHITRESH	A00005

4.Please follow instructions given below.

Write a query to display customer number, customer's first name, account number where the account status is terminated.

Display the records sorted in ascending order based on customer number and then by account number.

SELECT am.customer_number,firstname, account_number

FROM customer_master cm JOIN account_master am

ON cm.customer_number=am.customer_number

WHERE account_status='Terminated'

ORDER BY am.customer_number, account_number

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Please follow instructions given below.

Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions.

Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type, count(transaction_number) Trans_Count
```

```
FROM account_master am JOIN transaction_details td
```

```
ON am.account_number=td.account_number
```

```
WHERE customer_number like '%001'
```

```
GROUP BY transaction_type
```

```
ORDER BY transaction_type
```

TRANSACTION_TYPE	TRANS_COUNT
DEPOSIT	3
WITHDRAWAL	3

6. Please follow instructions given below.

Write a query to display the number of customers who have registration but no account in the bank.

Give the alias name as Count_Customer for number of customers.

```
SELECT count(customer_number) Count_Customer
```

```
FROM customer_master
```


WHERE customer_number NOT IN (SELECT customer_number FROM account_master)

COUNT_CUSTOMER
4

7.Please follow instructions given below.

Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT td.account_number, opening_balance+sum(transaction_amount) Deposit_Amount
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
WHERE transaction_type='deposit'
GROUP BY account_number
ORDER BY account_number
```

ACCOUNT_NUMBER	DEPOSIT_AMOUNT
A00001	10000
A00002	6000
A00007	17000

8.Please follow instructions given below.

Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```
select branch_master.branch_city, count(account_master.account_number) as No_of_Accounts from  
branch_master left join account_master on account_master.branch_id=branch_master.branch_id
```

```
group by branch_master.branch_city order by branch_city;
```

BRANCH_CITY	NO_OF_ACCOUNTS
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9.Please follow instructions given below.

Write a query to display the firstname of the customers who have more than 1 account. Display the records in sorted order based on firstname.

```
select firstname
```

```
FROM customer_master cm INNER JOIN account_master am
```

```
ON cm.customer_number=am.customer_number
```

```
group by firstname
```

```
having count(account_number)>1
```

```
order by firstname;
```

FIRSTNAME
AMIT
AVINASH
RAHUL
RAMESH

10.Please follow instructions given below.

Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch.

Display the records sorted in order based on customer number.

```

SELECT Id.customer_number, firstname, lastname
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number
GROUP BY customer_number
HAVING count(branch_id)>1
ORDER BY customer_number

```

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11.Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different.

Display the records sorted in ascending order based on customer number.

```
select customer_master.customer_number, firstname, customer_city, branch_city
from account_master inner join customer_master on account_master.customer_number =
customer_master.customer_number
inner join branch_master on account_master.branch_id = branch_master.branch_id
where customer_city != branch_city order by customer_master.customer_number;
```

CUSTOMER_NUMBER	FIRSTNAME	CUSTOMER_CITY	BRANCH_CITY
C00002	AVINASH	DELHI	MUMBAI
C00003	RAHUL	DELHI	MUMBAI
C00007	AMIT	MUMBAI	DELHI

12.Please follow instructions given below.

Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(Id.customer_number) Count
FROM customer_master cm INNER JOIN loan_details Id
ON cm.customer_number=Id.customer_number
WHERE cm.customer_number NOT IN ( SELECT customer_number FROM account_master)
```

(Or)

```
select count(customer_number) as Count from customer_master where customer_number not in
(select customer_number from account_master) and customer_number in
```

(select customer_number from loan_details);

COUNT
2

13. Please follow instructions given below.

Write a query to display the account number who has done the highest transaction.

For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed.

In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT td.account_number
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number
having count(td.transaction_number)>=ALL
(SELECT count(td.transaction_number)
FROM account_master am INNER JOIN transaction_details td
ON am.account_number=td.account_number
group by td.account_number) order by am.account_number;
```

ACCOUNT_NUMBER
A00001

14. Please follow instructions given below.

Write a query to show the branch name, branch city where we have the maximum customers.

For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore.

In case of multiple records, display the records sorted in ascending order based on branch name.

```
select branch_name,branch_city
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name
having count(customer_number)>=ALL
(select count(customer_number)
FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name) order by branch_name;
```

BRANCH_NAME	BRANCH_CITY
ASAF ALI ROAD	DELHI

15. Please follow instructions given below.

Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well.

For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011.

Display the records sorted in ascending order based on account number.

```
select am.account_number,opening_balance+sum(case when transaction_type='Deposit' then
transaction_amount end) as Deposit,sum(case when transaction_type='withdrawal' then
transaction_amount end) as Withdrawal from account_master am join transaction_details td
on am.account_number=td.account_number group by am.account_number having
Withdrawal>Deposit;
```

ACCOUNT_NUMBER	DEPOSIT	WITHDRAWAL
A00001	10000	12000
A00002	6000	7000

16.Please follow instructions given below.

Write a query to show the balance amount for account number that ends with 001.

Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount.

For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000) – (500+500+1000).

```
SELECT (SUM(CASE WHEN transaction_type='Deposit'
THEN transaction_amount END)) -
(SUM(CASE WHEN transaction_type='Withdrawal'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001') AS Balance_Amount
FROM transaction_details where account_number like '%001'
```

BALANCE_AMOUNT
-2000

17.Please follow instructions given below.

Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT cm. customer_number,firstname, am.account_number,count(transaction_number)
Count_Trans
```

```
FROM customer_master cm inner JOIN account_master am
```

```
ON cm.customer_number=am.customer_number
```

```
INNER JOIN transaction_details td
```

```
ON am.account_number=td.account_number
```

```
group by am.account_number order by cm.customer_number, am.account_number
```

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER	COUNT_TRANS
C00001	RAMESH	A00001	6
C00002	AVINASH	A00002	3
C00007	AMIT	A00007	3

18.Please follow instructions given below.

Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.


```
SELECT firstname  
  
FROM customer_master INNER JOIN account_master  
  
ON customer_master.customer_number=account_master.customer_number  
  
GROUP BY firstname  
  
having count(firstname)>=2 order by firstname;
```

FIRSTNAME
AMIT
AVINASH
RAHUL
RAMESH

19.Please follow instructions given below.

Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches.

For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012

Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is

600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011.

So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT Id.customer_number, firstname, lastname
```

FROM customer_master cm INNER JOIN loan_details ld

ON cm.customer_number=ld.customer_number

group by customer_number

**having count(branch_id)>=2 and sum(loan_amount)>=All(select sum(loan_amount) from loan_details
group by customer_number)**

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00002	AVINASH	MINHA

20.Please follow instructions given below.

Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans..

Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

SELECT ld.customer_number, firstname,branch_id, loan_amount

FROM customer_master cm INNER JOIN loan_details ld

**ON cm.customer_number=ld.customer_number order by cm.customer_number, branch_id,
loan_amount**

CUSTOMER_NUMBER	FIRSTNAME	BRANCH_ID	LOAN_AMOUNT
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000

C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000
C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21.Please follow instructions given below.

Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch.

Display the records sorted in ascending order based on city name.

SELECT branch_city, count(branch_id) Count_Branch

FROM branch_master

GROUP BY branch_city

ORDER BY branch_city

BRANCH_CITY	COUNT_BRANCH
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22.Please follow instructions given below.

Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active.

Display the records sorted in ascending order based on account id /account number.

```
SELECT account_number, firstname, lastname
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE account_status='Active'
ORDER BY account_number
```

ACCOUNT_NUMBER	FIRSTNAME	LASTNAME
A00001	RAMESH	SHARMA
A00002	AVINASH	MINHA
A00003	RAHUL	RASTOGI
A00004	AVINASH	MINHA
A00005	CHITRESH	BARWE
A00007	AMIT	BORKAR
A00010	PARUL	GANDHI

23.Please follow instructions given below.

Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name.

Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,coalesce(middlename,lastname) Middle_Name  
FROM customer_master order by customer_number
```

CUSTOMER_NUMBER	FIRSTNAME	MIDDLE_NAME
C00001	RAMESH	CHANDRA
C00002	AVINASH	SUNDER
C00003	RAHUL	RASTOGI
C00004	PARUL	GANDHI
C00005	NAVEEN	CHANDRA
C00006	CHITRESH	BARWE
C00007	AMIT	KUMAR
C00008	NISHA	DAMLE
C00009	ABHISHEK	DUTTA
C00010	SHANKAR	NAIR

24.Please follow instructions given below.

Write a query to display the customer number , firstname, customer's date of birth . Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth
FROM customer_master order by year(customer_date_of_birth), firstname;
```

CUSTOMER_NUMBER	FIRSTNAME	CUSTOMER_DATE_OF_BIRTH
C00009	ABHISHEK	1973-05-22
C00002	AVINASH	1974-10-16
C00008	NISHA	1975-12-03
C00005	NAVEEN	1976-09-19
C00004	PARUL	1976-11-03
C00001	RAMESH	1976-12-06
C00010	SHANKAR	1976-07-12
C00007	AMIT	1981-09-06
C00003	RAHUL	1981-09-26
C00006	CHITRESH	1992-11-06

25.Please follow instructions given below.

Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student.

Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT firstname, customer_city, account_number
FROM customer_master cm INNER JOIN account_master am
ON cm.customer_number=am.customer_number
WHERE occupation != 'Service' and occupation != 'Student' and occupation != 'Business' order by
firstname, account_number
```

FIRSTNAME	CUSTOMER_CITY	ACCOUNT_NUMBER
PARUL	DELHI	A00010

```
CREATE TABLE customer_master(  
    CUSTOMER_NUMBER VARCHAR(6),  
    FIRSTNAME VARCHAR(30),  
  
    middlename VARCHAR(30),  
    lastname VARCHAR(30),  
    CUSTOMER_CITY VARCHAR(15),  
  
    CUSTOMER_CONTACT_NO VARCHAR(10),  
    occupation VARCHAR(10),  
    CUSTOMER_DATE_OF_BIRTH DATE,  
  
    CONSTRAINT customer_custid_pk PRIMARY KEY (CUSTOMER_NUMBER)  
);
```

```
CREATE TABLE branch_master(  
    branch_id VARCHAR(6),  
  
    branch_name VARCHAR(30),  
    branch_city VARCHAR(30),  
  
    CONSTRAINT branch_bid_pk PRIMARY KEY (branch_id)  
);
```

```
CREATE TABLE account_master  
(account_number VARCHAR(255),  
    customer_number VARCHAR(255),  
  
    branch_id VARCHAR(255),  
    opening_balance INT(20),  
    account_opening_date DATE,  
    account_type VARCHAR(10),
```



```
account_status VARCHAR(10),  
    PRIMARY KEY (account_number),  
    FOREIGN KEY (customer_number) references customer_master(customer_number),  
  
    FOREIGN KEY (branch_id) references branch_master(branch_id)  
);
```

```
CREATE TABLE transaction_details(  
    transaction_number VARCHAR(6),  
  
    account_number VARCHAR(6),  
    date_of_transaction DATE,  
    medium_of_transaction VARCHAR(20),  
  
    transaction_type VARCHAR(20),  
    transaction_amount INT(7),  
  
    CONSTRAINT transaction_details_tnumber_pk PRIMARY KEY (transaction_number),  
  
    CONSTRAINT transaction_details_acnumber_fk FOREIGN KEY (account_number)  
  
    REFERENCES account_master (account_number)  
);
```

```
create table loan_details(customer_number varchar(255),  
branch_id varchar(255),  
loan_amount bigint(20),  
foreign key(customer_number) references customer_master(customer_number));
```

```
insert into customer_master values('C00001',      'RAMESH',  'CHANDRA',  
'SHARMA',  'DELHI',      '9543198345', 'SERVICE'  , '1976-12-06');
```

```
insert into customer_master values('C00002',      'AVINASH',  'SUNDER',  
'MINHA',   'DELHI',      '9876532109' , 'SERVICE',  '1974-10-16');
```

```
insert into customer_master values('C00003',      'RAHUL',  
'NULL','RASTOGI',  'DELHI',      '9765178901', 'STUDENT',  '1981-09-26');
```

```
insert into customer_master values('C00004',      'PARUL',  
'NULL','GANDHI',   'DELHI',      '9876532109' , 'HOUSEWIFE',  
      '1976-11-03');
```

```
insert into customer_master values('C00005',  
'NAVEEN'    , 'CHANDRA', 'AEDEKAR',  'MUMBAI',      '8976523190', 'SERVICE'   , '1976-  
09-19');
```

```
insert into customer_master  
values('C00006',      'CHITRESH', 'NULL','BARWE',      'MUMBAI',      '7651298321',  
      'STUDENT'   , '1992-11-06');
```

```
insert into customer_master  
values('C00007',      'AMIT' , 'KUMAR',      'BORKAR',      'MUMBAI',      '9875189761',  
      'STUDENT',  '1981-09-06');
```

```
insert into customer_master  
values('C00008',      'NISHA',      NULL, 'DAMLE',      'MUMBAI',      '7954198761',  
      'SERVICE',  '1975-12-03');
```

```
insert into customer_master  
values('C00009',      'ABHISHEK', NULL, 'DUTTA',      'KOLKATA'   , '9856198761',  
      'SERVICE'   , '1973-05-22');
```

```
insert into customer_master  
values('C00010',      'SHANKAR'  , NULL, 'NAIR', 'CHENNAI',      '8765489076', 'SERVICE',
```

'1976-07-12');

insert into branch_master values('B00001', 'ASAF ALI ROAD','DELHI');

insert into branch_master values('B00002','NEW DELHI MAIN BRANCH','DELHI');

insert into branch_master values('B00003' , 'DELHI CANTT', 'DELHI');

insert into branch_master values('B00004' , 'JASOLA', 'DELHI');

insert into branch_master values('B00005' , 'MAHIM' , 'MUMBAI');

insert into branch_master values('B00006' , 'VILE PARLE', 'MUMBAI');

insert into branch_master values('B00007', 'MANDVI' , 'MUMBAI');

insert into branch_master values('B00008' , 'JADAVPUR', 'KOLKATA');

insert into branch_master values('B00009' , 'KODAMBAKKAM', 'CHENNAI');

insert into account_master values('A00001' , 'C00001' , 'B00001' , 1000 , '2012-12-15',
'SAVING', 'ACTIVE');

insert into account_master values('A00002' , 'C00002' , 'B00001' , 1000 , '2012-06-12'
, 'SAVING', 'ACTIVE');

```
insert into account_master values('A00003' , 'C00003',      'B00002',      1000      , '2012-05-17'
                                , 'SAVING',      'ACTIVE');
```

```
insert into account_master values('A00004' , 'C00002',      'B00005',      1000      , '2013-01-27'
                                , 'SAVING      ', 'ACTIVE');
```

```
insert into account_master values('A00005' , 'C00006',      'B00006',      1000      , '2012-12-17'
                                , 'SAVING', 'ACTIVE');
```

```
insert into account_master values('A00006' , 'C00007',      'B00007',      1000      , '2010-08-12'
                                , 'SAVING      ', 'SUSPENDED');
```

```
insert into account_master values('A00007' , 'C00007',      'B00001',      1000      , '2012-10-02'
                                , 'SAVING      ', 'ACTIVE');
```

```
insert into account_master values('A00008' , 'C00001'      , 'B00003',      1000      , '2009-11-09'
                                , 'SAVING      ', 'TERMINATED');
```

```
insert into account_master values('A00009' , 'C00003',      'B00007',      1000      , '2008-11-30'
                                , 'SAVING',      'TERMINATED');
```

```
insert into account_master values('A00010' , 'C00004',      'B00002',      1000      , '2013-03-01'
                                , 'SAVING',      'ACTIVE');
```

```
insert into transaction_details values('T00001',      'A00001',      '2013-01-01', 'CHEQUE',
                                'DEPOSIT',      2000);
```

```
insert into transaction_details values
('T00002'      , 'A00001'      , '2013-02-01' , 'CASH'      , 'WITHDRAWAL',      1000);
```

```
insert into transaction_details values
('T00003', 'A00002', '2013-01-01','CASH','DEPOSIT', 2000);
```

```
insert into transaction_details values('T00004', 'A00002', '2013-02-01', 'CASH'
, 'DEPOSIT', 3000);
```

```
insert into transaction_details values('T00005', 'A00007', '2013-01-11', 'CASH'
, 'DEPOSIT', 7000);
```

```
insert into transaction_details values('T00006', 'A00007', '2013-01-13', 'CASH'
, 'DEPOSIT', 9000);
```

```
insert into transaction_details values('T00007', 'A00001', '2013-03-13', 'CASH'
, 'DEPOSIT', 4000);
```

```
insert into transaction_details values('T00008', 'A00001', '2013-03-14', 'CHEQUE'
, 'DEPOSIT', 3000);
```

```
insert into transaction_details values('T00009', 'A00001', '2013-03-21', 'CASH'
, 'WITHDRAWAL', 9000);
```

```
insert into transaction_details values('T00010', 'A00001', '2013-03-22', 'CASH'
, 'WITHDRAWAL', 2000);
```

```
insert into transaction_details values('T00011', 'A00002', '2013-03-25', 'CASH'
, 'WITHDRAWAL', 7000);
```

```
insert into transaction_details values('T00012', 'A00007', '2013-03-26', 'CASH'
, 'WITHDRAWAL', 2000);
```

```
insert into Loan_details values('C00001', 'B00001', 100000);
```

```
insert into Loan_details values('C00002', 'B00002', 200000);
```

```
insert into Loan_details values('C00009', 'B00008', 400000);
```

```
insert into Loan_details values('C00010', 'B00009', 500000);
```

```
insert into Loan_details values('C00001', 'B00003', 600000);
```

```
insert into Loan_details values('C00002', 'B00001', 600000);
```


Item Loan Database Queries

1. Please follow instructions given below.

Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

3 rows

```
SELECT item_category , count(item_id) Count_category  
  
FROM item_master  
  
GROUP BY item_category order by count_category DESC;
```


Filter:	<input type="text"/>	
	item_category	Count_category
▶	furniture	15
	Crockery	4
	Stationary	3

2. Please follow instructions given below.

Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

1 row

```
SELECT count(employee_id) AS No_of_Employees  
  
FROM employee_master  
  
WHERE department= 'HR'
```

Filter:	<input type="text"/>	
	No_of_Employees	
▶	2	

3. Please follow instructions given below.

Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

1 row

```
select employee_id, employee_name, designation, department from employee_master
where employee_id
not in (select employee_id from employee_issue_details) order by employee_id;
```

	employee_id	employee_name	designation	department
▶	E00005	Radica	Manager	HR
★	NULL	NULL	NULL	NULL

4. Please follow instructions given below.

Write a query to display the employee id, employee name who was issued an item of highest valuation.

In case of multiple records, display the records sorted in ascending order based on employee id.

[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

1 row

```
select em.employee_id, em.employee_name from employee_master em join employee_issue_details
eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id
and im.item_valuation>=all(select im.item_valuation from employee_master em
join employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id)
order by employee_id;
```

	employee_id	employee_name
▶	E00004	Zuben

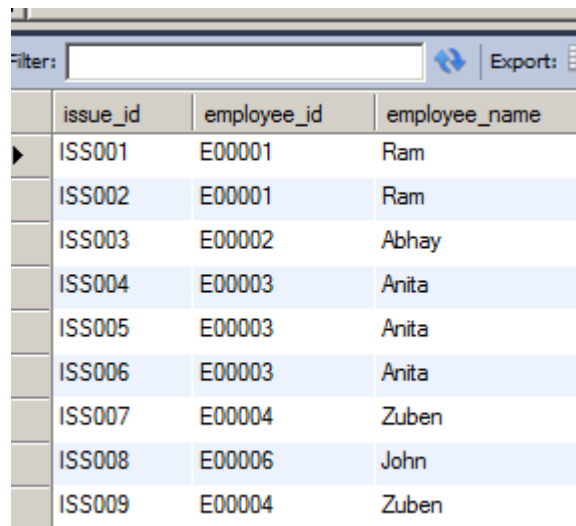
5. Please follow instructions given below.

Write a query to display issue_id, employee_id, employee_name.

Display the records sorted in ascending order based on issue id.

9 rows

```
select eid.issue_id, eid.employee_id, em.employee_name from employee_issue_details eid join  
employee_master em on eid.employee_id = em.employee_id group by eid.issue_id, eid.employee_id
```



	issue_id	employee_id	employee_name
▶	ISS001	E00001	Ram
	ISS002	E00001	Ram
	ISS003	E00002	Abhay
	ISS004	E00003	Anita
	ISS005	E00003	Anita
	ISS006	E00003	Anita
	ISS007	E00004	Zuben
	ISS008	E00006	John
	ISS009	E00004	Zuben

```
order by eid.issue_id;
```

6. Please follow instructions given below.

Write a query to display employee id, employee name who don't have loan cards.

Display the records sorted in ascending order based on employee id.

3 rows

```
SELECT employee_id, employee_name  
FROM employee_master  
WHERE employee_id NOT IN ( SELECT employee_id FROM employee_card_details )  
order by employee_id;
```

Filter:		
	employee_id	employee_name
▶	E00004	Zuben
	E00005	Radica
	E00006	John
*	NULL	NULL

7. Please follow instructions given below.

Write a query to count the number of cards issued to an employee "Ram". Give the count an alias name as No_of_Cards.

1 row

```
select count(eid.loan_id) as No_of_Cards from employee_card_details eid join employee_master em
on eid.employee_id=em.employee_id where em.employee_name='Ram'
```

Filter:	
	No_of_Cards
▶	3

8. Please follow instructions given below.

Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

1 row

```
select count(ecd.employee_id) as Count_Stationary from employee_card_details ecd
join loan_card_master lcm on ecd.loan_id=lcm.loan_id where lcm.loan_type='Stationary'
```

Filter:	
	Count_stationary
▶	3

9. Please follow instructions given below.

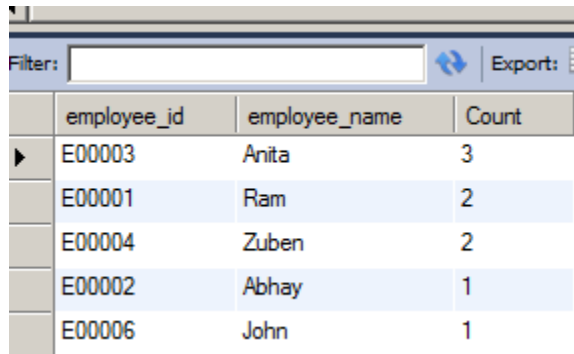
Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then by employee id in ascending order. Consider only employees who have been issued atleast 1 item.

5 rows

```
select em.employee_id,em.employee_name,count(eid.item_id) as Count from employee_master em
join
```

```
employee_issue_details eid on em.employee_id=eid.employee_id group by em.employee_id having
```

```
count(eid.item_id)>=1 order by Count desc,employee_id asc;
```



employee_id	employee_name	Count
E00003	Anita	3
E00001	Ram	2
E00004	Zuben	2
E00002	Abhay	1
E00006	John	1

10. Please follow instructions given below.

Write a query to display the employee id, employee name who was issued an item of minimum valuation.

In case of multiple records, display them sorted in ascending order based on employee id.

[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

2 rows

```
select em.employee_id,em.employee_name from employee_master em join employee_issue_details
eid
```

```
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id
```

```
and im.item_valuation<=all (select im.item_valuation from employee_master em join
employee_issue_details eid
```

```
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id) order by
employee_id;
```

employee_id	employee_name
E00002	Abhay
E00003	Anita

11.Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION.

Display the records sorted in ascending order based on employee id.

Consider only employees who have been issued atleast 1 item.

5 rows

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION
from employee_master em
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id group by em.employee_id having count(im.item_valuation)>=1
order by em.employee_id;
```

employee_id	employee_name	TOTAL_VALUATION
E00001	Ram	7000.00
E00002	Abhay	1500.00
E00003	Anita	15500.00
E00004	Zuben	25500.00
E00006	John	4500.00

12.Please follow instructions given below.

Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date.

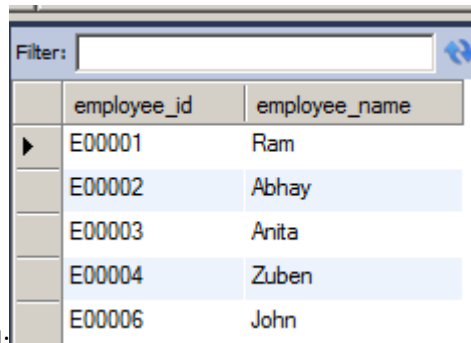
Display the records only if it is more than 365 Days.

Display the records sorted in ascending order based on employee id.

5 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
```

```
on em.employee_id=eid.employee_id where datediff(return_date,issue_date)>365 order by
```



employee_id	employee_name
E00001	Ram
E00002	Abhay
E00003	Anita
E00004	Zuben
E00006	John

```
employee_id;
```

13.Please follow instructions given below.

Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT_ITEMS.

Display the records sorted in ascending order on employee id.

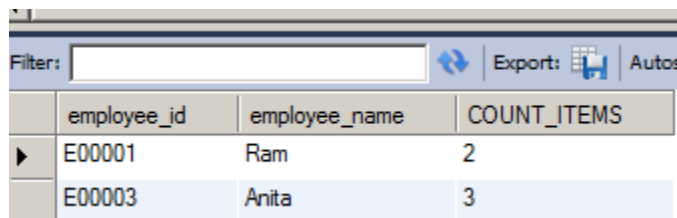
2 rows

```
select em.employee_id,em.employee_name,count(im.item_id) as COUNT_ITEMS from
employee_master em
```

```
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
```

```
on eid.item_id=im.item_id where item_category='furniture' group by employee_id having
```

```
count(COUNT_ITEMS)>1 order by employee_id;
```



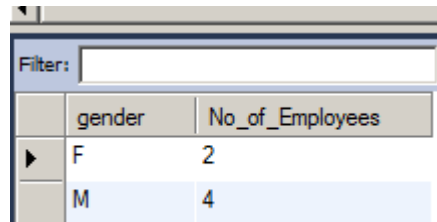
employee_id	employee_name	COUNT_ITEMS
E00001	Ram	2
E00003	Anita	3

14.Please follow instructions given below.

Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

2 rows

select gender,count(employee_id) as No_of_Employees from employee_master group by



gender	No_of_Employees
F	2
M	4

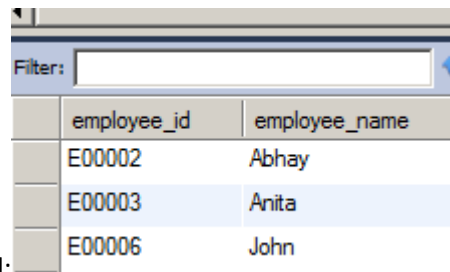
gender order by gender;

15.Please follow instructions given below.

Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

3 rows

select employee_id,employee_name from employee_master where year(date_of_joining)>2005



employee_id	employee_name
E00002	Abhay
E00003	Anita
E00006	John

order by employee_id;

16.Please follow instructions given below.

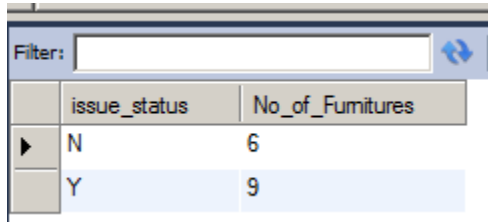
Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No_of_Furnitures.

Display the records sorted in ascending order based on issue_status.

2 rows

```
select issue_status,count(item_id) as No_of_Furnitures from item_master where  
item_category='furniture' group by issue_status order by
```

```
issue_status;
```



	issue_status	No_of_Furnitures
▶	N	6
	Y	9

17.Please follow instructions given below.




Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.

16 rows

```
select item_category,item_make,item_description,count(item_id) as No_of_Items from
```

```
item_master im group by item_category,item_make,item_description order by
```

```
item_category,item_make,item_description;
```

Filter: <input type="text"/>						Export: 	Autosize: 
	item_category	item_make	item_description	No_of_Items			
▶	Crockery	Bonechina	Dining Set	1			
	Crockery	Bonechina	Tea Set	1			
	Crockery	Glass	Dining Set	1			
	Crockery	Glass	Tea Set	1			
	furniture	Steel	Cupboard	2			
	furniture	Steel	Side Table	1			
	furniture	Steel	Single Bed	2			
	furniture	Steel	Tea Table	2			
	furniture	Wooden	Dining Chair	1			
	furniture	Wooden	Dining Table	1			
	furniture	Wooden	Double Bed	2			
	furniture	Wooden	Side Table	1			
	furniture	Wooden	Sofa	1			
	furniture	Wooden	Tea Table	2			
	Stationary	Plastic	Pen	2			
	Stationary	Wooden	Pencil	1			

18. Please follow instructions given below.

Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.




1 row

```
select em.employee_id,em.employee_name,im.item_id,im.item_description from employee_master em
join
```

```
employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
```

```
eid.item_id=im.item_id where year(eid.issue_date)=2013 and month(eid.issue_date)=01 order by
```

```
em.employee_id,im.item_id;
```

Filter:			Export: 	Autosize: 
	employee_id	employee_name	item_id	item_description
▶	E00002	Abhay	I00005	Side Table

19. Please follow instructions given below.

Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.

Give the alias name for category count as COUNT_CATEGORY.

Display the records sorted in ascending order based on employee id.

1 row

```
select em.employee_id,em.employee_name,count(distinct im.item_category) as COUNT_CATEGORY
from employee_master em
```

```
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
```

```
on eid.item_id=im.item_id group by em.employee_id having COUNT_CATEGORY>=2
```

```
order by em.employee_id;
```

Filter:		Export:	Autosize:
	employee_id	employee_name	COUNT_CATEGORY
▶	E00004	Zuben	2

20. Please follow instructions given below.

Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.

14 rows

```
select item_id,item_description from item_master where item_id not in (select item_id
```

```
from employee_issue_details) order by item_id;
```

Filter:		
	item_id	item_description
▶	I00002	Dining Table
	I00003	Tea Table
	I00006	Tea Table
	I00009	Sofa
	I00011	Cupboard
	I00013	Double Bed
	I00014	Single Bed
	I00015	Single Bed
	I00016	Tea Set
	I00017	Tea Set
	I00019	Dining Set
	I00020	Pencil
	I00021	Pen
	I00022	Pen
▼	NULL	NULL

21. Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation for the employees who have issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issued item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

1 row

```
select em.employee_id, em.employee_name, sum(im.item_valuation) as TOTAL_VALUATION from
employee_master em
```

```
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
```

```
eid.item_id=im.item_id group by em.employee_id having sum(im.item_valuation) <= all
```

```
(select sum(im.item_valuation) from employee_master em
```

```
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
```

```
eid.item_id=im.item_id group by em.employee_id) order by employee_id;
```

filter:		Export:	Autosize:
employee_id	employee_name	TOTAL_VALUATION	
E00002	Abhay	1500.00	

22. Please follow instructions given below.

Write a query to display the employee id, employee name, card issue date and card valid date.

Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE.

[Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.]

```

SELECT ecd.employee_id, employee_name,
card_issue_date, if(lcd.duration_in_years=0, 'NO-VALIDITY DATE', date_add(ec.card_issue_date, interval
duration_in_years year)) as CARD_VALIDITY_DATE
FROM employee_master em INNER JOIN
employee_card_details ecd
ON em.employee_id=ecd.employee_id
INNER JOIN loan_card_master lcd
ON ecd.loan_id=lcd.loan_id
order by employee_name, CARD_VALID_DATE;

```

Filter:		Export:	Autosize:
employee_id	employee_name	card_issue_date	CARD_VALID_DATE
E00002	Abhay	2007-02-01	2012-02-01
E00002	Abhay	2007-03-11	No Validity Date
E00003	Anita	2007-04-15	2008-04-15
E00003	Anita	2007-04-15	2012-04-15
E00003	Anita	2007-04-15	No Validity Date
E00001	Ram	2002-12-14	2003-12-14
E00001	Ram	2000-01-01	2005-01-01
E00001	Ram	2000-01-01	No Validity Date

23. Please follow instructions given below.

Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

3 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid on
```

```
em.employee_id=eid.employee_id where em.employee_id not in
```

```
(select employee_id from employee_issue_details where year(issue_date)=2013)
```

```
order by employee_id;
```

Filter:	
employee_id	employee_name
E00001	Ram
E00003	Anita
E00006	John

24. Please follow instructions given below.

Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.

9 rows

```
select eid.issue_id,em.employee_id,em.employee_name,im.item_id,im.item_description,eid.issue_date
from employee_issue_details eid join employee_master em on eid.employee_id=em.employee_id
join item_master im on eid.item_id=im.item_id order by eid.issue_date desc,eid.issue_id;
```




Iter:		Export:		Autosize:	
issue_id	employee_id	employee_name	item_id	item_description	issue_date
ISS009	E00004	Zuben	I00018	Dining Set	2013-04-18
ISS007	E00004	Zuben	I00012	Double Bed	2013-04-14
ISS003	E00002	Abhay	I00005	Side Table	2013-01-03
ISS008	E00006	John	I00018	Dining Set	2012-08-18
ISS006	E00003	Anita	I00010	Cupboard	2012-03-14
ISS001	E00001	Ram	I00001	Tea Table	2012-02-03
ISS002	E00001	Ram	I00004	Side Table	2012-02-03
ISS004	E00003	Anita	I00007	Dining Chair	2010-07-04
ISS005	E00003	Anita	I00008	Tea Table	2010-07-04

25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

1 row

```
select em.employee_id,em.employee_name,sum(im.item_valuation) as TOTAL_VALUATION
from employee_master em join employee_issue_details eid on em.employee_id=eid.employee_id
join item_master im on eid.item_id=im.item_id group by em.employee_id having
sum(im.item_valuation)
>= all (select sum(im.item_valuation) from employee_master em join employee_issue_details eid on
em.employee_id=eid.employee_id
join item_master im on eid.item_id=im.item_id group by em.employee_id);;
```

Iter: <input type="text"/>				Export: 	Autosize: 
employee_id	employee_name	TOTAL_VALUATION			
E00004	Zuben	25500.00			

```

CREATE TABLE loan_card_master
(
    loan_id                int(6)          PRIMARY KEY,
    loan_type              varchar(20),
    duration_in_years      int(2)
);

```

```

CREATE TABLE employee_master
(
    employee_id            varchar(10)       PRIMARY KEY,
    employee_name          varchar(30),
    designation            varchar(255),

    department            varchar(255),
    gender                varchar(6),
    date_of_birth          date,
    date_of_joining        date
);

```

```

CREATE TABLE item_master
(
    item_id                varchar(10)       PRIMARY KEY,
    item_description       varchar(50),
    issue_status           varchar(10),
    item_make              varchar(20),
    item_category          varchar(20),
    item_valuation         double(7,2)
);

```

```

CREATE TABLE employee_card_details
(
    employee_id            varchar(10)       REFERENCES employee_master,
    loan_id                varchar(10)       REFERENCES loan_card_master,
    card_issue_date        date
);

```

```

CREATE TABLE employee_issue_details
(
    issue_id               varchar(10)       PRIMARY KEY,
    employee_id            varchar(10)       REFERENCES employee_master,
    item_id                varchar(10)       REFERENCES item_master,
    issue_date             date,
    return_date            date
);

```

);

```
insert into loan_card_master
values('00001','stationary',5);
insert into loan_card_master
values('00002','recurring',0);
insert into loan_card_master
values('00003','Crockery',1);
```

```
insert into employee_master
values('E00001','Ram','Manager','Finance','M','1973-12-01','2001-01-01');
```

```
insert into employee_master
values('E00002','Abhay','Assistant Manager',
'Finance','M','1976-01-01','2006-12-01');
```

```
insert into employee_master
values('E00003','Anita','Senior Executive','Marketing','F','1977-05-12','2007-03-21');
```

```
insert into employee_master
values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');
```

```
insert into employee_master
values('E00005','Radica','Manager','HR','F','1976-07-22','2004-01-23');
```

```
insert into employee_master
values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');
```

```
insert into item_master
values ('I00001','Tea Table','Y','Wooden','furniture',5000);
```

```
insert into item_master
values ('I00002','Dining Table','N','Wooden','furniture',15000);
```

```
insert into item_master
values ('I00003','Tea Table','N','Steel','furniture',6000);
```

```
insert into item_master
values ('I00004','Side Table','Y','Wooden','furniture',2000);
```



```
insert into item_master  
values ('I00005','Side Table','Y','Steel','furniture',1500);
```

```
insert into item_master  
values ('I00006','Tea Table','N','Steel','furniture',7000);
```

```
insert into item_master  
values ('I00007','Dining Chair','Y','Wooden','furniture',1500);
```

```
insert into item_master  
values ('I00008','Tea Table','Y','Wooden','furniture',4000);
```

```
insert into item_master  
values ('I00009','Sofa','N','Wooden','furniture',18000);
```

```
insert into item_master  
values ('I00010','Cupboard','Y','Steel','furniture',10000);
```

```
insert into item_master  
values ('I00011','Cupboard','N','Steel','furniture',14000);
```

```
insert into item_master  
values ('I00012','Double Bed','Y','Wooden','furniture',21000);
```

```
insert into item_master  
values ('I00013','Double Bed','Y','Wooden','furniture',20000);
```

```
insert into item_master  
values ('I00014','Single Bed','Y','Steel','furniture',10000);
```

```
insert into item_master  
values ('I00015','Single Bed','N','Steel','furniture',10000);
```

```
insert into item_master  
values ('I00016','Tea Set','Y','Glass','Crockery',3000);
```

```
insert into item_master  
values ('I00017','Tea Set','Y','Bonechina','Crockery',4000);
```

```
insert into item_master  
values ('I00018','Dining Set','Y','Glass','Crockery',4500);
```

```
insert into item_master  
values ('I00019','Dining Set','N','Bonechina','Crockery',5000);
```

```
insert into item_master  
values ('I00020','Pencil','Y','Wooden','Stationary',5);
```

```
insert into item_master  
values ('I00021','Pen','Y','Plastic','Stationary',100);
```

```
insert into item_master  
values ('I00022','Pen','N','Plastic','Stationary',200);
```

```
insert into employee_card_details  
values('E00001','00001','2000-01-01');
```

```
insert into employee_card_details  
values('E00001','00002','2000-01-01');
```

```
insert into employee_card_details  
values('E00001','00003','2002-12-14');
```

```
insert into employee_card_details  
values('E00002','00001','2007-02-01');
```

```
insert into employee_card_details  
values('E00002','00002','2007-03-11');
```

```
insert into employee_card_details  
values('E00003','00001','2007-04-15');
```

```
insert into employee_card_details  
values('E00003','00002','2007-04-15');
```

```
insert into employee_card_details  
values('E00003','00003','2007-04-15');
```

```
insert into employee_issue_details
values('ISS001','E00001','I00001','2012-02-03','2014-02-03');
insert into employee_issue_details
values('ISS002','E00001','I00004','2012-02-03','2020-02-03');
```

```
insert into employee_issue_details
values('ISS003','E00002','I00005','2013-01-03','2015-01-03');
insert into employee_issue_details
values('ISS004','E00003','I00007','2010-07-04','2012-07-04');
```

```
insert into employee_issue_details
values('ISS005','E00003','I00008','2010-07-04','2012-08-05');
```

```
insert into employee_issue_details
values('ISS006','E00003','I00010','2012-03-14','2012-06-15');
```

```
insert into employee_issue_details
values('ISS007','E00004','I00012','2013-04-14','2016-04-14');
```

```
insert into employee_issue_details
values('ISS008','E00006','I00018','2012-08-18','2019-04-17');
```

```
insert into employee_issue_details
values('ISS009','E00004','I00018','2013-04-18','2013-05-18');
```

Question Text	Choice1	Choice2	Choice3	Choice4	Choice5	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5
What are the major elements in an object model?	Abstraction, Encapsulation and persistence	Hierarchy, concurrency and typing	Abstraction, encapsulation and hierarchy	Typing		0	0	1	0	
Which of the following is not a type of object oriented abstraction?	Abstraction of data	Abstraction of function	Abstraction of structure	Abstraction of name		0	0	0	1	
A pure virtual function or pure virtual method is a virtual function that is required to be implemented by a derived class that is abstract.	FALSE	TRUE				1	0			
Wrapping up of data & functions together in a class is known as ____.	Overloading	Data Abstraction	Polymorphism	Encapsulation		0	0	0	1	
Including only necessary details and ignoring additional details while defining a class is known as ____.	Overloading	Data Abstraction	Polymorphism	Encapsulation		0	1	0	0	
Preventing direct access of data-members of the class from outside world is known as ____.	Polymorphism	Inheritance	Data Hiding	scope resolution.		0	0	1	0	
State the object oriented languages	C++	Java	Eiffel	All of the above		0	0	0	1	
What is a reference?	an operator	a reference is an alias for an object	used to rename an object	None of these		0	1	0	0	
A library function exit() causes an exit from	the loop in which it occurs	the block in which it occurs	the function in which it occurs	the program in which it occurs		0	0	0	1	
In Object-oriented programming, the problem is divided into ____.	classes & objects	functions	structures	modules		1	0	0	0	
A class is ____ datatype.	primitive	derived	user-defined	All of these		0	0	1	0	
A class is a collection of ____ and ____.	data-members & member functions	data-members, member functions and main()	data-members, member functions, main() and	None of these		1	0	0	0	

			include statements							
An object is	a variable of class datatype	same as a class.	just like a global variable	collection of data-members alone		1	0	0	0	
In OOPS unit of data is called as	Bits	Blocks	Structures	Targets		0	0	1	0	
There is no difference between an object and an instance.State true or false	FALSE	TRUE				1	0			
A pure virtual function or pure virtual method is a virtual function that is required to be implemented by a derived class that is abstract.	FALSE	TRUE				1	0			
Creating a new class using one or more existing classes is known as _____.	Polymorphism	Encapsulation	overloading	inheritance		0	0	0	1	
Ability of an operator or function call to take different forms is known as _____.	Polymorphism	Encapsulation	overloading	inheritance		1	0	0	0	
If a class C is derived from class B, which is derived from class A, all through public inheritance, then a class C member function can access	protected and public data only in C and B.	protected and public data only in C	private data in A and B.	protected data in A and B		0	0	0	1	
RunTime Polymorphism is achieved by _____	friend function	virtual function	operator overloading	function overloading		0	1	0	0	
Which of the statements is true in a protected derivation of a derived class from a base class?	Private members of the base class become protected members of the derived class	Protected members of the base class become public members of the derived class	Public members of the base class become protected members of the derived class	Protected derivation does not affect private and protected members of the derived class.		0	0	1	0	
Mechanism of deriving a class from another derived class is known as_____	Polymorphism	Single Inheritance	Multilevel Inheritance	Message Passing		0	0	1	0	
Which of the following statements is NOT valid about operator overloading?	Only existing operators can be overloaded.	Overloaded operator must have at least one operand of its class type.	The overloaded operators follow the syntax rules of the original operator.	none of the above.		0	0	0	1	
Which of the following is the valid class declaration header for the derived class d with base classes b1 and b2?	class d : public b1, public b2	class d : class b1, class b2	class d : public b1, b2	class d : b1, b2		1	0	0	0	
A class defined within another class is:	Nested class	Inheritance	Containership	Encapsulation		1	0	0	0	
The major goal of inheritance in c++ is:	To facilitate the conversion of data types.	To help modular programming.	To extend the capabilities of a class	To hide the details of base class.		0	0	1	0	

The following can be declared as friend in a class	an object	a class	a public data member	a private data member		0	1	0	0	
Which of the following operator can be overloaded through friend function?	->	()	equal to	*		0	0	0	1	
A class cannot inherit members from more than one class. (State whether true or false)	TRUE	FALSE				0	1			
By default, all members of a class have _____ access for all its members	Public	Protected	No access	private		0	0	0	1	
Functions can be declared with default values in parameters. We use default keyword to specify the value of such parameters State whether the statement is true or false	TRUE	FALSE				0	1			
Overloaded functions are	Very long functions that can hardly run	One function containing another one or more functions inside it.	Two or more functions with the same name but different number of parameters or type.	None of the listed options		0	0	1	0	
Identify the correct statement regarding scope of variables	Global variables are declared in a separate file and accessible from any program.	Local variables are declared inside a function and accessible within the function only.	Global variables are declared inside a function and accessible from anywhere in program.	Local variables are declared in the function that can be accessible outside from any other functions.		0	1	0	0	
You can use C++ as a procedural, as well as an object-oriented, language	TRUE	FALSE				1	0			
When the compiler cannot differentiate between two overloaded constructors, they are called	overloaded	destructured	ambiguous	dubious		0	0	1	0	
To be called object-oriented, a programming language must allow which of the following features	Overloading	polymorphism	inheritance	All of the above		0	0	0	1	
Header files in C++ often have the file extension _____	.H	.HE	.HEA	.HEAD		1	0	0	0	
When a child class function is called, the compiler looks first for a matching function name in the _____	class of the object using the function name	immediate ancestor class	base class	descendant class		1	0	0	0	

Paying attention to the important properties while ignoring inessential details is known as	selectiveness	polymorphism	abstraction	summarizing		0	0	1	0	
A base class may also be called a	child class	subclass	derived class	parent class		0	0	0	1	
Which of the following statements is correct?	Base class pointer cannot point to derived class.	Derived class pointer cannot point to base class.	Pointer to derived class cannot be created.	Pointer to base class cannot be created.		0	1	0	0	
Which of the following is not the member of class?	Static function	Friend function	Const function	Virtual function		0	1	0	0	
How many instances of an abstract class can be created?	1	5	13	0		0	0	0	1	
Which of the following concepts of OOPS means exposing only necessary information to client?	Encapsulation	Abstraction	Data hiding	Data binding		0	0	1	0	
Which of the following is not a feature of C++ ?	Operator overloading	Namespaces	Inheritance	Reflection		0	0	0	1	
Overloading the function operator	requires a class with an overloaded operator.	requires a class with an overloaded [] operator.	allows you to create objects that act syntactically like functions.	usually make use of a constructor that takes arguments.		1	0	0	0	
How many access specifiers are present in C++ programming class?	1	2	3	4		0	0	1	0	
Which of the following is a valid class declaration?	class A { int x; };	class B { }	class B { }	object A { int x; };		1	0	0	0	
To overload an operator _____ keyword must be used along with the operator to be overloaded.	Over	Overload	void	Operator		0	0	0	1	
When an object has many forms, it has _____.	Scalability	Inheritance	Polymorphism	Encapsulation		0	0	1	0	
By polymorphism of a subsystem we mean	it should be reusable	it should have polymorphic data types	it should accept generic commands and interpret appropriately	None of the listed options		0	0	1	0	
All objects have (i) attributes (ii) states (iii)a set of operations (iv) a unique identity	iii	ii, iii, iv	i, iii, iv	i, ii, iii, iv		0	0	0	1	
Which of the following operator can not be overloaded ?	Scope resolution operator	Equality operator	Assignment Operator	None of the listed options		1	0	0	0	
Which of the following statement is correct?	C++ allows static type checking.	C++ allows dynamic type checking.	C++ allows static member function be of type const.	None of the listed options	C++ allows global member function	1	1	0	0	0

					be of type const.						
Which of the following ways are legal to access a class data member using this pointer?	this->x	this.x	*this.x	*this-x		1	0	0	0		
Which special character is used to mark the end of class?	;	:	#	\$		1	0	0	0		
Procedure oriented Programs are called as	Structured programming	Object oriented programming	Functional programming	None of the listed options		1	0	0	0		
A _____ is an abstract idea that can be represented with data structures and functions.	class	object	loop	data type		1	0	0	0		
Automatic Initialization of object is carried out using a special member function called _____	friend	casting	reference parameter	constructor.		0	0	0	1		
In C++ a class can allow non-member functions and other classes to access its own private data, by making them as _____.	private	protected	Friend	public		0	0	1	0		
In c++ _____ Operator is used for Dynamic memory allocation	Scope resolution	Conditional	New	Membership access		0	0	1	0		
The advantages of OOP are , 1. increased programming productivity 2. decreased maintenance costs. 3. less time to execute 4. easy to understand	1& 3	1& 2	3& 4	2& 3		0	1	0	0		
State True or False 1. Public data members can be accessed directly in the main function without an object. 2. Constructors can be overloaded.	1-F, 2-F	1-F, 2-T	1-T, 2-T	1-T, 2-F		0	1	0	0		

Match the following. A) Self review B) Formal review C) Informal review 1. Conducted by one or	A - 1, B - 2, C - 3	A - 2, B - 3, C - 1	A - 3, B - 2, C - 1	A - 3, B - 1, C - 2	A - 2, B - 1, C - 3	0	0	1	0	0	
---	---------------------	---------------------	---------------------	---------------------	---------------------	---	---	---	---	---	--

more peers in the team 2. Conducted by one or more reviewers or SME 3. Conducted by the author himself											
Review of Test case Artifact is done with the help of?	Checklist	Self review	Peer review	Author	Reviewer	1	0	0	0	0	
What are the possible causes for ending up into 0.1 % defective application?	Misunderstood requirements	Defective code	Less knowledge on development language	Developers tend to neglect test approach to the developed product.	Lack of domain knowledge	1	1	0	1	1	
In causal analysis which attributes among below assist in analyzing the effect?	Reason	Cause	Test Approach	Requirement gathering	Failures	1	1	0	0	0	
Software testing ensures which of the below?	Usage of design architecture	Use of proper test approach	Proper causal analysis	Requirement satisfaction and usage of best design architecture	None of the above	0	0	0	1	0	
State whether true or false. Selenium tools helps to develop Automated test scripts	TRUE	FALSE				1	0				
Test environment check up is part of _____ .	Test Scenario	Test Execution	Test Design	Test Development	None of the above	0	1	0	0	0	
State whether true or false. QC is used for logging the outcome of the test execution.	TRUE	FALSE				1	0				
Which of the following map the corresponding phases from SDLC with STLC.	Requirement Analysis - Test Planning Design and Code - Test Design Testing - Component Integration testing and System testing	Requirement Analysis - Test Design Design and Code - Test Planning Testing - Component Integration testing and System testing	Requirement Analysis - Test Planning Design and Code - Test Design Testing - Unit Testing	Requirement Analysis - Test Planning Design and Code - Unit Testing Testing - Component Integration testing and System testing	None of the above	1	0	0	0	0	

Which of the statements is applicable to software testing?	Helps in identifying defects	Helps prevent the defects	Helps to provide a reliable system	Helps to identify completeness of the software	None of the above	1	1	1	1	0	
State whether True or False. Iterative model is an example of a methodology used for software development.	TRUE	FALSE				1	0				
State whether True or False. Each SDLC model follows the same life cycle in order to ensure success in the process of software development	TRUE	FALSE				1	0				
Which statements are applicable to V model?	Includes verification on right arm and validation on left arm of the V shape	Integration test plan is prepared based on detailed design phase	System test plan is prepared based on user requirements	Acceptance testing is the last phase	System testing is done after integration testing	0	0	0	1	1	
State whether True or False. Test Design is done after requirement analysis and before test execution	TRUE	FALSE				1	0				
State whether True or False. Test Design involves the activity of prioritizing the test cases	TRUE	FALSE				0	1				
State whether True or False. Test Design process involves only Test development process	TRUE	FALSE				0	1				
State whether True or False. Unit testing is done - To test the smallest piece of code Is done on source code	TRUE	FALSE				1	0				
State whether True or False. Regression test cases are	TRUE	FALSE				1	0				

identified in Test development process											
Which statements are applicable to Test Scenarios?	Done after Requirement elicitation	Allows ease of review by developers	Helps in identifying defects	Functions to bridge the requirement analysis and test development	Used to create a clear and simple flow of a complex system	0	0	1	1	1	
Which statement is correct with regards to Pre Condition?	These are required to be set for performing the activity to achieve the goal	These are required to be verified by tester after the activity is performed	These are required to be verified by developer after the activity is performed	These are required to be set for planning the activity to achieve the goal	None of the above	1	0	0	0	0	
Which statement is correct with regards to Post Condition?	These are required to be set for performing the activity to achieve the goal	These are required to be verified by tester after the activity is performed	These are required to be verified by developer after the activity is performed	These are required to be set for planning the activity to achieve the goal	None of the above	0	1	0	0	0	
Which of the following is not a step involved to arrive at a test case?	Identify test conditions	Identify input variables, different options for the input variables	Combine scenarios with test conditions	Combine on split test cases for different flows	None of the above	0	0	0	1	0	
The process of creating complete set of test cases is called?	Test Scenario	Test Case	Test Development	Test Execution	None of the above	0	0	1	0	0	
State whether True or False. While writing Test scenarios we can replace requirement ID with use case name.	TRUE	FALSE				0	1				

Question Text	Choice1	Choice2	Choice3	Choice4	Choice5	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5
State whether True or False. A use case can result into more than one scenario.	TRUE	FALSE				1	0			
Test conditions can be valid or invalid (State True or False)	TRUE	FALSE				1	0			
Alternate flows can be	TRUE	FALSE				0	1			

tested by themselves (State True or false)										
Test scenarios have to be written with the consideration of ?	Business rules	Functional standards	Non functional standards	All of the above	None of the listed options	1	1	1	0	0
Test Scenarios have case specific data assigned to them (State True or False)	TRUE	FALSE				0	1			
Test data preparation data is done during _____ ?	Test Scenario identification process	Test Development process	Test Execution process	Test condition defining process		0	1	0	0	
An input field takes the birth year of the user ranging from 1960 to 1995. The boundary values for testing this field are?	0,1960,1995	1960, 1995, 1996	1959, 1960, 1961, 1994, 1995, 1996	0, 1959, 1960, 1961, 1994, 1995, 1996	1959, 1960, 1994, 1995	0	0	1	0	0
State whether True or False. Testers should be involved in reviewing documents as soon as drafts are available in the development cycle.	TRUE	FALSE				1	0			
A procedure used to derive and or select test cases is called?	Requirement Analysis	Test Planning	Test Design	Test Execution	Test Design and Execution	0	0	1	0	0
Testing during the design stage involves?	Examining the design documents	Reading drafts of the planning documents	Integration Testing	1 and 3	None of the above	1	0	0	0	0
State whether true or false. Informal review is done after formal review.	TRUE	FALSE				0	1			
Review report is created in which type of review?	Self review	Informal review	Formal review	All of the above	None of the above	0	0	1	0	0

For a given set of boundaries, how many boundary values are possible?	2	4	6	8	None of the above	0	0	1	0	0
We derive _____ by using the test design techniques	Test Scenario	Test condition	Test case	All of the above	None of the above	0	0	1	0	0
Which of the following statements is/are true?	Test scenario involves the expected results.	Test case includes the method of how the test would be performed.	Test scenario define the setup to perform the tests	Test case includes the steps to execute.	Test cases are developed from Test conditions.	0	1	0	1	0
Which is the correct order to be followed for a Build Verification Process?	<p>A. Build the compiled code into software</p> <p>B. Add the release notes</p> <p>C. Perform Smoke/ Sanity Test</p> <p>D. Test Execution</p>	<p>A. Review the code</p> <p>B. Build the compiled code into software</p> <p>C. Perform Smoke/ Sanity Test</p> <p>D. Test Execution</p>	<p>A. Build the compiled code into software</p> <p>B. Add the release notes</p> <p>C. Perform Smoke/ Sanity Test</p> <p>D. Rebuild the compiled code after bug fixing</p> <p>E. Update the release notes</p> <p>F. Perform Smoke/ Sanity Test</p> <p>G. Perform Test Execution if Smoke/ Sanity test is a pass</p>	<p>A. Build the compiled code into software</p> <p>B. Add the release notes</p> <p>C. Perform Smoke/ Sanity Test</p> <p>D. Test Execution</p> <p>E. Rebuild the compiled code after defect fixing</p> <p>F. Update the release notes</p> <p>G. Perform Smoke/ Sanity Test</p> <p>H. Perform Test Execution if Smoke/</p>	None of the above	0	0	1	0	0

				Sanity test is a pass						
The conditions that need to be verified by the tester after the activity is performed are called _____?	Pre condition	Post condition	Triggers	Exceptions		0	1	0	0	
A defect is found after retest. What are all the possible stages this defect may undergo?	Open, Fixed, Reopen, Closed	Reopen, Fixed, Closed	Deferred, Open, Fixed, Reopen, Closed	Reopen, Fixed		0	1	0	0	
What are the action items if an application does not behave as expected?	Update status of the defect	Log defect	Retest	Execute next test step of same test case		1	1	0	0	
Which is not a major task of test implementation and execution?	Develop and prioritizing test cases, creating test data, writing test procedures and optionally, preparing test harness and writing automated test scripts	Logging the outcome of test execution and recording the identities and versions of the software under test, test tools and testware	Verifying that the test environment has been set up correctly	Checking test logs against the exit criteria specified in test planning	3 and 4	0	0	0	1	0
What are the subsequent states that a new defect can undergo?	Rejected	Open	Deferred	Fixed	Closed	1	1	1	0	0
State whether True or False. Triage meeting is done before fixing the defect.	TRUE	FALSE				1	0			
State whether True or False. Developer has to ensure that the pre requisite of each test case are met.	True	FALSE				0	1			
State whether True or False. Release notes are prepared	TRUE	FALSE				0	1			

by developer/ development team.										
Which of the below is not an activity involved in Test execution process?	Buil d verification process	Test data setup	Test case execution	Defect Tracking	Retesting of defects	0	1	0	0	0

Question Text	Choice1	Choice2	Choice3	Choice 4	Choice 5	Gr ad e1	Gr ad e2	Gr ad e3	Gr ad e4	Gr ad e5
In requirements validation the requirements model is reviewed to ensure its technical feasibility. State True/False	TRUE	FALSE				0	1			
Software engineering aims at developing	Reliable Software	Cost Effective Software	Reliable and cost effective Software	None Of Above		0	0	1	0	
Software Engineering approach is used to achieve	Better performance of h/w	Error free s/w	Reusable software	Quality software product		0	0	0	1	
The best way to conduct a requirements validation review is to	send them to the design team and see if they have any concerns	use a checklist of questions to examine each requirement	have the customer look over the requirements	examine the system model for errors		0	1	0	0	
Project risk factor is considered in	Water fall	Spiral	Prototype	All of the above		0	1	0	0	
Management of software development is dependent upon	People	Product	Process	All of the above		0	0	0	1	
Milestones are used to	Know the cost of the project	Know the status of the project	Know the user expectations	None of the above		0	1	0	0	
The review is one of the methods of V&V. The other methods are	Inspecti on	Walkthroug h	Testing	All of the above		0	0	0	1	
Which of the following is not Risk characteristic	Inheren t in every project	Neither intrinsically good not bad	Something to fear but not something to manage	Probabil ity of loss		0	0	1	0	
The Prototype is a	Workin g model of existing system	Mini model of existing system	Mini model of processe d system	None of the above		1	0	0	0	

Which is not the responsibility of customer/ user of the software	Plan how and by whom each acceptance activity will be performed	Prepare the acceptance plan	Prepare resource plan	Plan resources for providing information on which to base acceptance decisions		0	0	1	0	
Software Engineering is the systematic approach to the development, operation, maintenance and retirement of software. This definition is given by_____	IEEE	Bauer	Boehm	Charles Babbage		1	0	0	0	
Software engineering umbrella activities are only applied during the initial phases of software development projects. State True or False	TRUE	FALSE				0	1			
Which of the items listed below is not one of the software engineering layers	Process	Manufacturing	Methods	Tools		0	1	0	0	
A stakeholder is anyone who will purchase the completed software system under development. State True/False	TRUE	FALSE				0	1			
Major component of Risk Analysis are	The probability that the negative event will occur	The potential loss is very high	The potential loss or impact associated with the event	A and C.		0	0	0	1	
Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind. State True/False	TRUE	FALSE				1	0			
Which phase is not available in s/w life cycle	Coding	Design	Specifications	Installation & Maintenance		0	0	0	1	
The work products produced during requirement elicitation will vary	size of the product being built	size of the budget	software process being used	stakeholders needs		1	0	0	0	
The term module in the design phase refers to	Functions	Procedures	Sub programs	All of the above		0	0	0	1	
Which of the following is the correct definition for DFD	The modern version of flowchart	Mainly used at systems specification stages	The primary output of the system design phase	All the above		0	0	1	0	
In system design, we do following	Hardware design after software	Software design after hardware	Parallel hardware and software design	No hardware design needed		0	0	1	0	

In object oriented design of software , objects have	attributes and names only	operations and names only	attribute s, name and operations	None of above		0	0	1	0	
Informational cohesion is a realization of	data abstraction	structured programming	Modularity	Concurrency		1	0	0	0	
Software is a product and can be manufactured using the same technologies used for other engineering artifacts. State True or False	TRUE	FALSE				0	1			
Object-oriented analysis techniques can be used to identify and refine user task objects and actions without any need to refer to the user voice. State True/False	TRUE	FALSE				0	1			
Which of these criteria are useful in assessing the effectiveness of a particular design notation	size	maintainability	simplicity	modularity	b,c and d	0	0	0	0	1
Which of these is a graphical notation for depicting procedural detail	decision table	process diagram	flowchart	ER diagram		0	0	1	0	
Which of the following comments about object oriented design of software, is not true	Objects inherit the properties of class	Classes are defined based on the attributes of objects	an object can belong to two classes	classes are always different		0	0	1	0	
The entity relationship diagram	depicts relationships between data objects	indicates system reactions to external events	depicts functions that transform the data flow	indicates how data are transformed by the system		1	0	0	0	
The data flow diagram must be augmented by descriptive text in order to describe the functional requirements for a software product. State True/False	TRUE	FALSE				1	0			
Which is not a software life cycle model	Water fall	Spiral	Prototype	Capability Maturity Model		0	0	0	1	
If requirements are understandable, easy, defined, which model is best suited	Water fall	Spiral	Prototype	None		1	0	0	0	
If requirements are frequently changing, which model is best suited	Water fall	Spiral	Prototype	RAD		0	0	1	0	
A data model consists of the following information	Data Object	The attributes that describe data object	Relationship that connect data object to one another	All of the above		0	0	0	1	
The incremental model of software development is	A good approach when a working core product	A reasonable approach when requirements are well defined	The best approach to use for projects with large development	A revolutionary model that is not used		1	0	0	0	

	is require d quickly		teams.	for comme rcial produc ts						
The prototyping model of software development is	The best approach to use for projects with large development teams	A risky model that rarely produces a meaningful product	A useful approach when a customer cannot define requirements clearly	A reasonable approach when requirements are well defined		0	0	1	0	
Which of following is not a UML diagram used creating a system analysis model	Dataflow diagram	Class diagram	Activity diagram	State diagram		0	0	1	0	
Control flow diagrams are	needed to model event driven systems .	required for all systems.	used in place of data flow diagrams.	useful for modeling real-time systems.	both a and d	0	0	0	1	0
The object relationship pair of data model is represented graphically by using	Data flow diagram	Flow chart	Entity relationship diagram	All of the above		0	0	1	0	
Using software process improvement model will help a company	To decrease development time	To meet schedule	To decrease the defect rate	To increase profitability	all of them	0	0	0	0	1
Data structure suitable for the application is discussed in ?	data design	architectural design	procedural design	interface design		1	0	0	0	
Process models are described as agile because they	eliminate the need for cumbersome documentation	make extensive use of prototype creation	do not waste development time on planning activities	emphasize maneuverability and adaptability		0	0	0	1	
Software processes can be constructed out of pre-existing software patterns to best meet the needs of a software project. State True or False	TRUE	FALSE				1	0			
The following s/w process model can be represented schematically as a series of major technical activities and there associated state	Incremental model	Component assembly	Concurrent development model	All of the above		0	0	1	0	
Which one is the most important feature of spiral model	Quality management	Risk Management	Performance Management	Efficiency management		0	1	0	0	
To produce a good quality product, process should be	Complex	Efficient	Rigorous	None		0	1	0	0	

If Quality Control and Quality Assurance are compared	Both are literally the same	QA is a higher activity in the management Hierarchy	QC is a higher activity in the management Hierarchy	QA is done by the client and QC is done by the software vendor		0	0	1	0	
Who is essentially responsible for the quality of a product	Customer	QA Manager	Development Manager			0	0	1		
What is used to measure the characteristics of the documentation and code	Process metrics	Product metrics	Software Quality metrics	None of the above		0	1	0	0	
What are the qualities of a good s/w	Reusability	Portability	Inter Operability	All The Above		0	0	0	1	
A key concept of quality control is that all work products	are delivered on time and under budget	have measurable specifications for process outputs	are thoroughly tested before delivery to the customer	have complete documentation		0	1	0	0	
Software safety is a quality assurance activity that focuses on hazards that	affect the reliability of a software component	may result from user input errors	prevent profitable marketing of the final product	may cause an entire system to fail		0	0	0	1	
What exactly Baseline means	A single software product that may or may not fully support a business function	A quantitative measure of the current level of performance	A test or analysis conducted after an application is moved into production	None of the above		0	1	0	0	
What is configuration management in software engineering	overall management of the design of the system	management of the configurable components in a system	the identification of the configuration of a system at discreet points in time to control changes to the configuration	in object-oriented programming, the management of objects that control the configuration of some other		0	0	1	0	

				function(s) in the system						
Which of the following tasks is not part of software configuration management?	change control	version control	reporting	statistical quality control		0	0	0	1	
Which of these are valid software configuration items?	documentation	software tools	test data	executable programs	all of the above	0	0	0	0	1
The primary purpose of configuration status reporting is to	evaluate the performance of software developers and organizations	make sure that change information is communicated to all affected parties	allow revision of project schedules and cost estimates by project managers	none of the above		0	1	0	0	
A new _____ is defined when major changes have been made to one or more configuration objects.	variant	entity	item	version		0	0	0	1	
In software quality assurance work there is no difference between software verification and software validation. State True/False	TRUE	FALSE				0	1			
People who perform software quality assurance must look at the software from the customer's perspective.	TRUE	FALSE				1	0			
Variation control in the context of software engineering involves controlling variation in the	processes applied	product quality attributes	resources expended	all of the above		0	0	0	1	
The goal of quality assurance is to provide management with the data needed to determine which software engineers are producing the most defects.	TRUE	FALSE				0	1			
The purpose of software reviews is to uncover errors in work products so they can be removed before moving on to the next phase of development.	TRUE	FALSE				1	0			
The ability to track relationships and changes to configuration objects is one of the most important features of the SCM repository.	TRUE	FALSE				1	0			
A basic configuration object is a _____ created by a software engineer during some phase of the software development process.	program data structure	unit of information	a software component	all of the above		0	0	1	0	
When software configuration management is a formal activity, the software configuration audit is conducted by the	quality assurance group	development team	senior managers	testing specialists		1	0	0	0	

Question Text	Choice1	Choice2	Choice3	Choice4	Choice5	Grade1	Grade2	Grade3	Grade4	Grade5
---------------	---------	---------	---------	---------	---------	--------	--------	--------	--------	--------

<p>Statement 1: A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.</p> <p>Statement 2: A subquery can be nested inside the WHERE or HAVING clause of an outer SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery.</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
<p>A query is called correlated subquery when both the inner query and the outer query are interdependent.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				1	0			
<p>Statement 1: If a subquery is not dependent on the outer query it is called a non-correlated subquery.</p> <p>Statement 2: Subqueries cannot be used with the comparison</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		1	0	0	0	

operators. Which of the above statements are TRUE?										
An index helps speed up SELECT queries and WHERE clauses, but it slows down data input, with UPDATE and INSERT statements. State whether the above statement is TRUE or FALSE.	TRUE	FALSE				1	0			
Which of the given options are TRUE?	An inline view exists only inside of the FROM clause as a run-time result set.	A subquery exists only inside of the FROM clause as a run-time	An inline view exists only inside of the WHERE clause as a run-time result set.	All listed options		1	0	0	0	
To remove duplicate rows from the result set of a SELECT use the following keyword:	NO DUPLICATE	UNIQUE	DISTINCT	None of the listed options		0	0	1	0	
Which of the following can add a row to a table?	Add	Insert	Update	Alter		0	1	0	0	
Which SQL statement is used to insert a new data in a database?	INSERT INTO	UPDATE	ADD	INSERT NEW		1	0	0	0	
In a LIKE clause, you can ask for any value ending in "qpt" by writing	LIKE %qpt	LIKE *ton	LIKE ton\$	LIKE ^.*ton\$		1	0	0	0	
In a LIKE clause, you can ask for any 6 letter value by writing?	LIKE ??????	LIKE .{6} Answer 5: LIKE ^.{6}\$	LIKE (that's six dots)	LIKE _____ (that's six underscore characters)		0	0	0	1	
The result of a SELECT	TRUE	FALSE				1	0			

statement can contain duplicate rows.										
A table may be joined to itself.	TRUE	FALSE				1	0			
Which of the following is not a valid aggregate function?	COUNT	MIN	MAX	COMPUTE		0	0	0	1	
What SQL clause is used to restrict the rows returned by a query?	AND	WHERE	HAVING	FROM		0	1	0	0	
Primary Key does allow the Null Values. where as in Unique key doesn't accept the Null values. State whether the statement is true or false	TRUE	FALSE				0	1			
Which of the following commands should be used to create a database named "student"?	CREATE ?I student	CREATE DATABASE student	DATABASE /student	DATABSE student		0	1	0	0	
Which one will delete the table data as well as table structure?	TRUNCATE	DROP	REMOVE	DISTINCT		0	1	0	0	
A SELECT command without a WHERE clause returns?	All the records from a table that match the previous WHERE clause	All the records from a table, or information about all the records	SELECT is invalid without a WHERE clause	Nothing		0	1	0	0	
What does the ALTER TABLE clause do?	The SQL ALTER TABLE clause is used to insert data into database table.	The SQL ALTER TABLE deletes data from database table.	The SQL ALTER TABLE clause modifies a table definition by altering, adding, or deleting table columns and/or constraints.	The SQL ALTER TABLE clause is used to delete a database table		0	0	1	0	
Can you use combination of GROUP BY clause,HAVING clause and WHERE clause SQL	TRUE	FALSE				1	0			

clauses in one SQL statement?										
What is a primary key?	The primary key is a column that can have NULL values.	The primary key is a column or combination of columns whose values uniquely identify each row in the table.	The primary key column is a column or combination of columns whose values can be non-unique.			0	1	0		
What is the purpose of the SQL AS clause?	The AS clause defines a search condition	The AS SQL clause is used to change the name of a column in the result set or to assign a name to a derived column.	The AS clause is used with the JOIN clause only.			0	1	0		
Which two are true about aggregate functions?(Choose two)	You can use aggregate functions in any clause of a SELECT statement.	You can use aggregate functions only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.	You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns.	You can pass column names, expressions, constants, or functions as parameters to an aggregate function.		0	0	1	1	
Which clause should you use to exclude group results?	WHERE	HAVING	RESTRICT	GROUP BY		0	1	0	0	
Which of the following SQL statements is correct?	SELECT CustomerName, COUNT(CustomerName) FROM Orders ORDER BY CustomerName	SELECT CustomerName, COUNT(CustomerName) FROM Orders	SELECT CustomerName, COUNT(CustomerName) FROM Orders GROUP BY CustomerName			0	0	1		
The SQL DROP TABLE clause is	create a new table in the database	delete a table from the database	modify an existing table in a database			0	1	0		

used to...										
We refer to a join as a self-join when?	we are joining table to itself	we are using left and right join together	we are joining more than 2 tables			1	0	0		
The INNER JOIN clause...	returns all rows from 2 tables	returns all rows that have matching value in the field on which the 2 tables are joined.	returns only the rows from the first table, which have non-matching values with the second table in the field on which the 2 tables are joined.			0	1	0		
If table A have 10 rows and table B have 5 rows, how many rows will be returned if you perform a cartesian join on those two tables?	5	50	10	15		0	1	0	0	
Which syntax would be used to retrieve all rows in both the EMPLOYEES and DEPARTMENTS tables, even when there is no match?	Outer join	Inner join	Self join	Natural join		1	0	0	0	
The main reason that constraints are added to a table is:	Constraints add a level of complexity	Constraints ensure data integrity	Constraints gives programmers job security	None of the listed options		0	1	0	0	
To automatically delete rows in a child table when a parent record is deleted use:	ON DELETE SET NULL	ON DELETE ORPHAN	ON DELETE CASCADE	None of the listed options		0	0	1	0	
A table can have more	TRUE	FALSE				1	0			

than one UNIQUE key constraint. True or False?										
A column defined as NOT NULL can have a DEFAULT value of NULL. True or False?	TRUE	FALSE				0	1			
A table must have at least one not null constraint and one unique constraint. True or False?	TRUE	FALSE				0	1			
The _____ join is the ANSI-standard syntax used to generate a Cartesian product.	NATURAL	ALL	FULL	CROSS		0	0	0	1	
In the relational model, relationships between relations or tables are created by using:	composite keys.	determinants.	candidate keys	foreign keys.		0	0	0	1	
Which two statements are true regarding the ORDER BY clause? (Choose two)	The sort is in ascending order by default.	The ORDER BY clause comes last in the SELECT statement.	The sort is in descending order by default	The ORDER BY clause is executed on the client side		1	1	0	0	
What is true about joining tables through an equijoin?	You can join a maximum of two tables through an equijoin.	You can join a maximum of two columns through an equijoin.	You can join n tables (all having single column primary keys) in a SQL statement by specifying a minimum of n-1 join	All listed options		0	0	1	0	

			conditions.							
The CUSTOMERS table has these columns: CUSTOMER_ID NUMBER(4) NOT NULL CUSTOMER_NAME VARCHAR2(100) NOT NULL STREET_ADDRESS VARCHAR2(150) CITY_ADDRESS VARCHAR2(50) STATE_ADDRESS VARCHAR2(50) PROVINCE_ADDRESS VARCHAR2(50) COUNTRY_ADDRESS VARCHAR2(50) POSTAL_CODE VARCHAR2(12) CUSTOMER_PHONE VARCHAR2(20) A sale is being advertised to the customers in France. Which WHERE clause identifies	WHERE lower(country_address) = 'france'	WHERE lower(country_address) = "france"	WHERE lower(country_address) IS 'france'	None	1	0	0	0		

customers that are located in France?										
SQL can be used to:	Modify the database	create database structures only.	query database data only.	All the listed operation can be done by SQL.		0	0	0	1	
Examine the structure of the EMPLOYEES table: EMPLOYEE_ID NUMBER Primary Key FIRST_NAME VARCHAR2(25) LAST_NAME VARCHAR2(25) HIRE_DATE DATE Which UPDATE statement is valid?	UPDATE employees SET first_name = 'John', SET last_name = 'Smith' WHERE employee_id = 180;	UPDATE employees SET first_name = 'John', last_name = 'Smith' WHERE employee_id = 180;	UPDATE employees SET first_name = 'John' AND last_name = 'Smith' WHERE employee_id = 180;	UPDATE employees SET first_name = 'John' SET last_name = 'Smith' WHERE employee_id = 180;		0	1	0	0	
The SQL WHERE clause:	limits the column data that are returned.	limits the row data are returned.	limits the rows & columns returned	NONE		0	1	0	0	
ON UPDATE CASCADE ensures which of the following?	Normalization	data Integrity	Materialized View	None		0	1	0	0	
Which of the following is valid SQL for an Index?	CREATE INDEX ID;	CHANGE INDEX ID;	ADD INDEX ID;	REMOVE INDEX ID;		1	0	0	0	

Question Text	Choice1	Choice2	Choice3	Choice4	Choice5	Grade1	Grade2	Grade3	Grade4	Grade5
---------------	---------	---------	---------	---------	---------	--------	--------	--------	--------	--------

Which of the given options are TRUE about 'varchar' datatype?	Holds a variable length string (can contain letters, numbers, and special characters).	Its maximum size is specified in parenthesis .	All listed options	None of the listed options		0	0	1	0	
Which of the given options are TRUE about TCL?	TCL contains the commands which are required for Transaction Management.	TCL consists of 2 commands: COMMIT and ROLLBACK	All listed options	None of the listed options		0	0	1	0	
Which of the given options are TRUE regarding 'Constraints'?	Constraints are used to limit the type of data that can go into a table	The NOT NULL constraint enforces a column to NOT accept NULL values.	All listed options	None of the listed options		0	0	1	0	
Statement 1: 'AND' Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE; otherwise returns UNKNOWN. Statement 2: 'EXISTS' returns FALSE if a sub-query returns at least one row. Which of the above statements are TRUE?	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		1	0	0	0	
Statement 1: 'UNION' returns all distinct rows selected by either query. Statement 2: 'INTERSECT' returns all distinct rows selected by both queries. Which of the above statements is TRUE?	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
Which operator returns all distinct rows selected by the first query but not the second?	UNION	MINUS	INTERSECT	UNION ALL		0	1	0	0	

Which of the given options is TRUE?	COUNT function is used to count the number of columns in a database table.	SUM function allows selecting the total for a numeric column.	All listed options	None of the listed options		0	1	0	0	
Which of the given options return rows when there is at least one match in both tables?	JOIN	WHERE	GROUP BY	ORDER BY		1	0	0	0	
Which type of join does not require each record in the two joined tables to have a matching record?	Inner join	Outer Join	Self join	Equi Join		0	1	0	0	
Statement 1: Clustered index physically rearranges the data that users inserts in your tables. Statement 2: There can be 2000 non-clustered index per table. Which of the above statement are TRUE?	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		1	0	0	0	
What is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.	Semicolon	Colon	Comma	All listed options		1	0	0	0	
DDL part of SQL does which of the following?	allows database tables to be created or deleted	Defines indexes (keys)	Specifies links between tables, and imposes constraints between tables	All listed options		0	0	0	1	
ANSI is the official U.S. representative to the International Organization for Standardization (ISO). State whether the above statement is true or false	TRUE	FALSE				1	0			
Which statement is used to query the database and retrieve selected data that match the criteria that you	INSERT	RETRIEVE	SELECT	UPDATE		0	0	1	0	

specify?										
<p>Statement1: Data types specify what the type of data can be for that particular column</p> <p>Statement 2: Varchar is a datatype in SQL</p> <p>Which of the above statements is TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
<p>Statement 1: The DELETE statement is used to delete columns in a table.</p> <p>Statement 2: The UPDATE statement is used to update existing records in a table.</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	1	0	0	
<p>Statement 1: DCL contains the commands which protect data from unauthorized access.</p> <p>Statement 2: DCL consists of 2 commands: COMMIT and ROLLBACK</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		1	0	0	0	
<p>Statement 1: GRANT, DENY and REVOKE are DCL commands</p> <p>Statement 2: CREATE, ALTER, DROP, TRUNCATE are DDL commands</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
<p>Which of the given options are TRUE regarding 'Constraints'?</p>	The UNIQUE constraint uniquely identifies each record in a database table.	A PRIMARY KEY constraint does not automatically have a UNIQUE constraint defined on it.	All listed options	None of the listed options		1	0	0	0	

You can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table. State whether the above statement is TRUE or FALSE.	TRUE	FALSE				1	0			
Statement 1: Each table can have only ONE primary key per table Statement 2: A primary key column can contain NULL values Which of the above statements are TRUE?	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		1	0	0	0	
Statement 1: A FOREIGN KEY in one table points to a PRIMARY KEY in another table. Statement 2: If you define a CHECK constraint on a single column it allows only certain values for this column. Which of the given options are TRUE?	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
Statement 1: Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement. Statement 2: Arithmetic operators manipulate numeric operands. Which of the above statements are TRUE?	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
Statement 1: If you want to select rows that satisfy at least one of the given conditions, you can use the logical operator, AND. Statement 2: <> Checks if the value of two operands are equal or not, if values are not equal	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	1	0	0	

<p>then condition becomes true.</p> <p>Which of the above statements are TRUE?</p>										
<p>Statement 1: SQL aggregate functions return a single value, calculated from values in a column.</p> <p>Statement 2: AVG() returns the average value</p> <p>Which of the above statements is TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
<p>LIKE clause is used to compare a value to similar values using logical operators. State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				0	1			
<p>The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				1	0			
<p>The HAVING clause places conditions on the selected columns, whereas the WHERE clause places conditions on groups created by the GROUP BY clause.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				0	1			
<p>Which of the given options is TRUE about LIKE clause?</p>	The percent sign represents zero, one, or multiple characters, when used with LIKE clause.	The underscore represents a single number or character.	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
<p>GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				1	0			

<p>Statement 1: Numeric functions accept numeric input and return string values.</p> <p>Statement 2: Single-row functions return a single result row for every row of a queried table or view.</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	All of the above	None of the listed options		0	1	0	0	
<p>The percent sign and the underscore cannot be used in combinations, when using LIKE clause.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				0	1			
<p>The ROUND() function is used to round a numeric field to the nearest hundred.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				0	1			
<p>SQL joins are used to query data from two or more tables, based on ____.</p>	a relationship between certain columns in tables	a relationship between certain rows in tables.	All listed options	None of the listed options		1	0	0	0	
<p>Which of the given options return all rows from the left table, even if there are no matches in the right table?</p>	JOIN	LEFT JOIN	RIGHT JOIN	CROSS JOIN		0	1	0	0	
<p>A Self Join is a type of sql join which is used to join a table to itself, particularly when the table has a FOREIGN KEY that references its own PRIMARY KEY.</p> <p>State whether the above statement is TRUE or FALSE.</p>	TRUE	FALSE				1	0			

<p>Statement 1: CROSS JOIN returns the Cartesian product of the sets of rows from the joined tables.</p> <p>Statement 2: You can have multiple conditions for the ON clause just like you can in a WHERE clause.</p> <p>Which of the above statements is TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
<p>Statement 1: In case of Natural Joins, common columns are columns that have the same number of rows in both tables.</p> <p>Statement 2: JOIN ON syntax is much more readable and maintainable than the natural join syntax.</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	1	0	0	
<p>Statement 1: The FULL OUTER JOIN will return all rows, as long as there's matching data in one of the tables.</p> <p>Statement 2: FULL OUTER JOIN includes all the rows from both the participating tables and does not select either the LEFT or RIGHT table from the JOIN key word.</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
Which type of join combines the results of both left and right outer joins?	Inner join	Cross Join	Full Outer Join	All of the above		0	0	1	0	
<p>You cannot add a subquery to a SELECT clause as a column expression in the SELECT list.</p> <p>State whether the</p>	TRUE	FALSE				0	1			

above statement is TRUE or FALSE.										
<p>Statement 1: A view can be accessed with the use of SQL SELECT statement like a table.</p> <p>Statement 2: A view can be made up by selecting data from more than one tables.</p> <p>Which of the above statements are TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		0	0	1	0	
View can be removed using which command?	DELETE VIEW	DROP VIEW	REMOVE VIEW	All listed options		0	1	0	0	
<p>Statement 1: The SQL subquery is a SELECT query that is embedded in the main SELECT statement.</p> <p>Statement 2: A subquery cannot return more than one rows</p> <p>Which of the above statements is TRUE?</p>	Only statement 1	Only statement 2	Both statement 1 and statement 2	None of the listed options		1	0	0	0	

Question Text	Choice1	Choice2	Choice 3	Choice 4	Choice5	Grade1	Grade2	Grade3	Grade4	Grade5
Able and Bill are two friends. Able is carrying silver and bill is carrying gold. They prepare a mixture by mixing metals in proportions 7:2 and 7:11 respectively. If equal quantities of the two metals are melted to form an alloy , the proportion of silver and gold in the alloy will be?	5:07:00 AM	5:08:00 AM	7:05:00 AM	9:04:00 AM		0	0	1	0	
Spring provides the following mechanisms of instantiating a bean	using the constructors	using static factory methods	using a factory bean	using BeanFactory	using Application Context	1	1	1	0	0

Global action is not going to stop climate change. The world needs to look harder at how to live with it.										
http://www.youtube.com/watch?v=qn7Z6P22Hfw&t=30s	The video showcases only 1 online assessment company.	The video showcases 3 online assessment companies.				0	1			
The author wants men to give women the right to vote because women	have been subjected to only domestic chores till date.	need liberation from the discrimination of the weaker gender.	are capable of maintaining peace and order	are better rulers.	None of the above	0	0	1	0	0
#FTB# interface, #FTB# annotation and #FTB# bean attribute can be used to specify destruction lifecycle callback on a bean.	DisposableBean, PostDestroy, destroy-method					1				

1) MySQL runs on which operating systems?

- a) Linux and Mac OS-X only
- b) Any operating system at all
- c) **Unix, Linux, Windows and others**
- d) Unix and Linux only

2) To remove duplicate rows from the result set of a SELECT use the following keyword:

- a) NO DUPLICATE
- b) UNIQUE
- c) **DISTINCT**
- d) None of the above

3) Which of the following can add a row to a table?

- a) Add

b) Insert

c) Update

d) Alter

4) To use MySQL on your computer, you'll need?

a) FTP and Telnet

b) Some sort of client program to access the databases

c) A Browser

d) Perl, PHP or Java

5) Which SQL statement is used to insert a new data in a database?

a) INSERT INTO

b) UPDATE

c) ADD

d) INSERT NEW

6) In a LIKE clause, you can could ask for any value ending in "qpt" by writing

a) LIKE %qpt

b) LIKE *ton

c) LIKE ton\$

d) LIKE ^. *ton\$

7) A NULL value is treated as a blank or 0.

a) True

b) Fasle

c) None of the above

8) MySQL is

a) A Programming language

b) A Programming language

c) A technique for writing reliable programs

d) A Relational Database Management System

9) In a LIKE clause, you can ask for any 6 letter value by writing?

a) LIKE ??????

b) LIKE .{6} Answer 5: LIKE ^.{6}\$

c) LIKE (that's six dots)

d) LIKE _____ (that's six underscore characters)

10) The result of a **SELECT** statement can contain duplicate rows.

a) False

b) True

c) None of the above

11) Which function used to get the current time in mysql?

a) getTime()

b) Time()

c) NOW()

12) A table may be joined to itself.

a) True

b) false

c) None of the above

13) Which of the following is not a valid aggregate function?

a) COUNT

b) MIN

c) MAX

d) COMPUTE

14) **mysql_pconnect()**

is used to make a persistent connection to the database which means a SQL link that do not close when the execution of your script ends.

a) True

b) False

15) What SQL clause is used to **restrict the rows** returned by a query?

a) AND

b) WHERE

c) HAVING

d) FROM

16) Which of the following is used to delete an entire MYSQL database?

a) mysql_drop_database

b) mysql_drop_entiredb

c) mysql_drop_db

d) mysql_drop_dbase

17) MySQL supports the complete SQL99 standard

a) false

b) true

18) **Primary Key** does allow the Null Values. where as in **Unique key** doesn't accept the Null values.

Question:

True or False ?

a) False

b) True

19)How much character are allowed to create database name?

a) 55

b) 72

c) 64

d) 40

20) Which of the following commands should be used to create a database named "student"?

- a) CREATE ?I student
 - b) CREATE DATABASE student
 - c) DATABASE /student
 - d) DATABSE student
-

21) Which one will delete the table data as well as table structure?

- a) TRUNCATE
 - b) DROP
-

22) The USE command?

- a) Is used to load code from another file
 - b) Has been deprecated and should be avoided for security reasons
 - c) Is a pseudonym for the SELECT command
 - d) Should be used to choose the database you want to use once you've connected to MySQL
-

23) Given an employees table as follows:

emp_id	emp_name
1	Brush
2	Jerrin

what value will be return by below query ?
Select count(*) from employees

- a) 3
 - b) 2
 - c) 1
 - d) none of the above
-

24) The main MySQL program that does all the data handling is called?

- a) mysql.exe
 - b) mysql
 - c) mysqld
 - d) httpd
-

25) A SELECT command without a WHERE clause returns?

- a) All the records from a table that match the previous WHERE clause
 - b) All the records from a table, or information about all the records
 - c) SELECT is invalid without a WHERE clause
 - d) Nothing
-

26) MySQL Access security is controlled through?

- a) The ID that the user logged into the server through, and priveliges set up for that account.
 - b) MySQL login accounts, and priveliges set for each account
 - c) The normal login security is sufficient for MySQL, and it does not have any extra controls of its own.
 - d) A table of valid IP addresses, and priveliges set up for each IP address
-

27) In a SELECT with a GROUP BY clause, a WHERE clause, and a HAVING clause, the WHERE conditions are applied before the HAVING conditions.

- a) True
- b) False
- c) Either True or False
- d) None of the above

QUESTION NO: 1

A table is successfully created by executing the following statement:

```
CREATE TABLE numbers (  
    double_number double,  
    decimal_number decimal(2,1)  
)
```

One row is successfully inserted into the numbers table. At this point, the table contains the following data:

```
+-----+-----+  
| double_number | decimal_number |  
+-----+-----+  
| 1.5 | 2.5 |  
+-----+-----+
```

The row is updated by executing the following statement:

```
UPDATE numbers  
SET double_number = double_number + 0.25,  
    decimal_number = decimal_number + 0.01
```

Which values are now stored in the double_number and decimal_number columns of the updated row? Select the best response.

- A. 1.8 and 2.5
- B. 1.75 and 2.5

- C. 1.8 and 2.51
- D. 1.75 and 2.51

Answer: B

QUESTION NO: 2

Which of the following statements can be used to list all databases that are accessible to the current user? Select the best response.

- A. LIST DATABASES
- B. SHOW DATABASES
- C. DISPLAY DATABASES
- D. VIEW DATABASES

Answer: B

QUESTION NO: 3

Which of the following statements will discard the existing database called world? Select the best response.

- A. DELETE DATABASE world
- B. DROP DATABASE world
- C. REMOVE DATABASE world
- D. TRUNCATE DATABASE world

Answer: B

QUESTION NO: 4

Which statement can be used to list all columns in the City table? Select the best response.

- A. DISPLAY COLUMNS FROM City
- B. SHOW COLUMNS FROM City
- C. SHOW COLUMNS LIKE 'City'
- D. SHOW City COLUMNS

Answer: B

QUESTION NO: 5

The default database contains a table called City. Which of the following statements may be executed to obtain a statement that could be used to (re-)create the City table? Select the best response.

- A. DESCRIBE City
- B. DESCRIBE TABLE City
- C. SHOW TABLE City
- D. SHOW CREATE TABLE City

Answer: D

QUESTION NO: 6

A MySQL table has ...

Select the best response.

- A.** zero or more columns, and zero or more rows.
- B.** zero or more columns, and one or more rows.
- C.** one or more columns, and zero or more rows.
- D.** one or more columns, and one or more rows.

Answer: C

QUESTION NO: 7

Which part of a SELECT statement specifies the tables from which data is to be retrieved? Select the best response.

- A.** The SELECT list.
- B.** The FROM clause.
- C.** The WHERE clause.
- D.** The LIMIT clause.

Answer: B

QUESTION NO: 8

Which of the following statements best describes the purpose of the SQL WHERE clause? In SQL statements, the WHERE clause specifies ...

Select the best response.

- A.** the tables from which data is to be retrieved.
- B.** a condition to filter for only specific rows.
- C.** a condition to filter for only specific groups defined by a GROUP BY clause.
- D.** a number to limit the number of rows that is operated upon by the statement.

Answer: B

QUESTION NO: 9

The table Country contains the following rows:

+-----+-----+	
Name Population	
+-----+-----+	
Nauru 12000	
Turks and Caicos Islands 17000	
Tuvalu 12000	
Wallis and Futuna 15000	
+-----+-----+	

Which of the following statements will return all rows in the table, sorted by the value in the Population column? Select the best response.

- A.** SELECT Name, Population ASC
FROM Country
- B.** SELECT Name, ORDER BY Population
FROM Country
- C.** SELECT Name, Population
FROM Country
GROUP BY Population ASC
- D.** SELECT Name, Population
FROM CountryORDER BY
Population

Answer: D

QUESTION NO: 10

In the context of database transactions, the atomicity property guarantees that...

Select the best response.

- A.** during a transaction, rows are processed one at a time.
- B.** all statements that are executed inside a transaction are immediately committed.
- C.** all statements that are executed inside a transaction are committed or rolled back as one unit.
- D.** other transactions cannot see the changes made in other ongoing uncommitted transactions.

Answer: C

QUESTION NO: 11

The following output describes the table City:

Field	Type	Null	Key	Default	Extra
CountryCode	char(3)	NO	PRI		
CityName	char(35)	NO	PRI		

The following output describes the table Country:

Field	Type	Null	Key	Default	Extra
CountryCode	char(3)	NO	PRI		
CountryName	char(52)	NO			
Continent	varchar(10)	YES		NULL	

The tables are related through the CountryCode column.

You need to retrieve all cities and list each CityName with the CountryName of only the corresponding country. Is this possible using the following query?

```
SELECT CityName,CountryName
```

```
FROM Country
```

```
INNER JOIN City
```

Select the best response.

- A. Yes.
- B. No, you can't do that in one statement.
- C. No, the tables are listed in the wrong order.
- D. No, the statement needs a condition to match related rows.

Answer: D

QUESTION NO: 12

Is it possible to save the result of a SELECT statement into a file using an SQL statement? Select the best response.

- A. No, not with SQL alone.
- B. Yes, by using the FILE() function.
- C. Yes, by using the INTO OUTFILE clause.
- D. Yes, by using the LOAD DATA INFILE clause.

Answer: C

QUESTION NO: 13

The Country table exists in the default database. In the same database, you need to create a new table called Country_Copy that is to contain the same columns as the Country table, as well as all of the data in the Country table. Which of the following statements can be used to create the Country_Copy table? Select the best response.

- A. CREATE TABLE Country_Copy SELECT * FROM Country
- B. INSERT INTO Country_Copy SELECT * FROM Country
- C. CREATE TABLE Country_Copy LIKE Country
- D. COPY TABLE Country TO Country_Copy

Answer: A

QUESTION NO: 14

The following output describes the table Country:

+-----+-----+-----+-----+-----+					
Field	Type	Null	Key	Default	Extra
+-----+-----+-----+-----+-----+					
Code	char(3)	NO	PRI		
Name	char(53)	NO			
Population	int(11)	YES		NULL	
+-----+-----+-----+-----+-----+					

You want to discard the rows in the Country table for which the value in the Population column is less than 5000 (and retain any other rows). Which of the following statements can be used to do that? Select the best response.

- A. DROP Country WHERE Population < 5000
- B. DROP FROM Country WHERE Population < 5000
- C. DELETE FROM Country WHERE Population < 5000
- D. DELETE SELECT * FROM Country WHERE Population < 5000

Answer: C

QUESTION NO: 15

The table Product contains exactly one row:

+-----+-----+-----+		
Name	Price	Discount
+-----+-----+-----+		
bread	1.00	NULL
+-----+-----+-----+		

Which of the options best matches the result returned by the following query:

SELECT Price - Price * Discount

FROM Product

Select the best response.

- A. +-----+
| Price - Price * Discount |
+-----+
| NULL |
+-----+
- B. +-----+
| Price - Price * Discount |
+-----+
| 0 |
+-----+
- C. +-----+
| Price - Price * Discount |
+-----+
| 0.00 |
+-----+
- D. +-----+
| Price - Price * Discount |
+-----+
| 1.00 |

+-----+

Answer: A

QUESTION NO: 16

Which of the following statements best describes the meaning of NULL? Select the best response.

- A. NULL denotes an empty set. It is used to indicate that a query does not return any rows.
- B. NULL denotes the default value for a data type or column.
- C. NULL denotes a missing or unknown value.
- D. In a string context, NULL is exactly the same as " - the empty string; in a numerical context, NULL is exactly the same as 0 - zero.

Answer: C

QUESTION NO: 17

You need to create a view called CountryDensity based on the following query:

```
SELECT Code, Name, Population / SurfaceArea As Density
FROM Country
```

Which of the following statements will create this view?

Select the best response.

- A. INSERT
INTO CountryDensity
SELECT Code, Name, Population / SurfaceArea As Density
FROM Country
- B. CREATE TABLE CountryDensity
AS
SELECT Code, Name, Population / SurfaceArea As Density
FROM Country
- C. CREATE VIEW CountryDensity
AS
SELECT Code, Name, Population / SurfaceArea As Density
FROM Country
- D. CREATE CountryDensity
AS
SELECT Code, Name, Population / SurfaceArea As Density
FROM Country

Answer: C

QUESTION NO: 18

Assuming that the table Country exists, which of the following statements can be used to discard the data and structure of the Country table? Select the best response.

- A.** TRUNCATE TABLE Country
- B.** DELETE TABLE Country
- C.** REMOVE TABLE Country
- D.** DROP TABLE Country

Answer: D

QUESTION NO: 19

What is the effect of the ROLLBACK statement?

Select the best response.

- A.** Issuing a ROLLBACK statement will undo all changes on transactional tables performed since the beginning of the session.
- B.** Issuing a ROLLBACK statement will undo all changes on transactional tables performed since the beginning of the transaction.
- C.** Issuing a ROLLBACK statement will undo all changes made by the previous statement.
- D.** Issuing a ROLLBACK statement will undo the effect of the previous COMMIT statement.

Answer: B

QUESTION NO: 20

You need to add a char(35) column called LocalName to the existing table City. Which of the following statements may be used to achieve this? Select the best response.

- A.** CREATE COLUMN LocalName char(35) FOR City
- B.** INSERT INTO City COLUMNS LocalName char(35)
- C.** ALTER TABLE City INSERT LocalName char(35)
- D.** ALTER TABLE City ADD LocalName char(35)

Answer: D

QUESTION NO: 21

Which of the following statements can be used to remove the SurfaceArea column from the Country table? Select the best response.

- A. DELETE SurfaceArea FROM Country
- B. DROP SurfaceArea FROM Country
- C. ALTER TABLE Country DROP SurfaceArea
- D. ALTER TABLE Country DELETE SurfaceArea

Answer: C

QUESTION NO: 22

LOAD DATA INFILE ...

Select the best response.

- A. is a statement to load data from a text file into a table.
- B. is a statement that allows one to recreate an entire database from a text file.
- C. is an SQL statement for loading data into a file.
- D. loads an SQL script into the mysql command line client.

Answer: A

QUESTION NO: 23

The following output describes the table Country:

+-----+-----+-----+-----+-----+-----+						
Field	Type	Null	Key	Default	Extra	
+-----+-----+-----+-----+-----+-----+						
Code	char(3)	PRI				
Name	char(52)					
Population	int(11)		0			
LocalName	char(45)					
Capital	int(11)	YES	NULL			

+-----+-----+-----+-----+-----+

5 rows in set (0.00 sec)

The following output describes the table City:

+-----+-----+-----+-----+-----+

| Field | Type | Null | Key | Default | Extra |

+-----+-----+-----+-----+-----+

| Id | int(11) | | PRI | NULL | auto_increment |

| Name | char(35) | | | |

| Country | char(3) | | | |

+-----+-----+-----+-----+-----+

3 rows in set (0.00 sec)

The following SQL statements are all syntactically correct, yet one of them will result in an error when executed. Which one? Select the best response.

A. SELECT Name, Name

FROM Country

INNER JOIN City

ON Capital = Id

B. SELECT Country, Country

FROM Country

INNER JOIN City

ON Capital = Id

C. SELECT Country, Id

FROM Country

INNER JOIN City

ON Capital = Id

D. SELECT Country.Name, Id

FROM Country

INNER JOIN City

ON Capital = Id

Answer: A

QUESTION NO: 24

After starting a transaction and executing a statement, you accidentally execute ROLLBACK instead of COMMIT. Is there any way to commit the entered statement?

Select the best response.

- A. You should execute COMMIT immediately.
- B. You should execute CANCEL ROLLBACK and then COMMIT.
- C. You should execute REPEAT TRANSACTION and then COMMIT.
- D. There is no way to do this. You have to repeat your transaction.

Answer: D

QUESTION NO: 25

The Cities table contains the following rows:

+-----+-----+	
Country	City
+-----+-----+	
USA	Seattle
Germany	Berlin
USA	New York
Sweden	Stockholm
+-----+-----+	

What will be the result of executing the following query?

```
SELECT Country, City
FROM Cities
ORDER BY Country, City
```

Select the best response.

- A.

+-----+-----+	
Country	City
+-----+-----+	
USA	Seattle
Germany	Berlin
Sweden	Stockholm
USA	New York
+-----+-----+	
- B.

+-----+-----+	
Country	City
+-----+-----+	
Germany	Berlin
USA	New York
USA	Seattle
Sweden	Stockholm
+-----+-----+	

C. +-----+-----+
 | Country | City |
 +-----+-----+
Germany	Berlin
Sweden	Stockholm
USA	New York
USA	Seattle
 +-----+-----+

D. +-----+-----+
 | Country | City |
 +-----+-----+
Germany	Berlin
Sweden	Stockholm
USA	Seattle
USA	New York
 +-----+-----+

Answer: C

QUESTION NO: 26

Assume that the database yellow exists and that no database contains a table called circle. You execute the following statement:

```
CREATE TABLE yellow.circle(x INT, y INT, r INT)
```

Which of the following options best describes the effect of executing this CREATE TABLE statement? Select the best response.

- A.** The table circle is created in the default database.
- B.** The table yellow.circle is created in the default database.
- C.** The table circle is created in the database yellow.
- D.** Executing the statement fails because yellow.circle is not a valid table name.

Answer: C

QUESTION NO: 27

Which result will be returned after executing the following statement?

```
SELECT NULL = NULL
```

Select the best response.

A. +-----+
 | NULL = NULL |

+-----+

| 0 |

+-----+

B. +-----+

| NULL = NULL |

+-----+

| 1 |

+-----+

C. +-----+

| NULL = NULL |

+-----+

| TRUE |

+-----+

D. +-----+

| NULL = NULL |

+-----+

| NULL |

+-----+

Answer: D

QUESTION NO: 28

The friends table has the following table structure and data:

```
mysql> SELECT * FROM Friends;
```

```
+-----+-----+
```

```
| Id | Name |
```

```
+-----+-----+
```

```
| 1 | Tom |
```

```
| 2 | Matt |
```

```
| 3 | David |
```

```
| 4 | Monty |
```

```
+-----+-----+
```

Which query could be used to retrieve a result similar to the one shown here:

```
+-----+
```

```
| Name |
```

```
+-----+
```

```
| Matt |
```

```
| Monty |
```

```
+-----+
```

Select the best response.

- A.** SELECT Name
FROM Friends
WHERE Id IN ('Matt','Monty')
- B.** SELECT Name
FROM Friends
WHERE Name = 'Matt'
AND Name = 'Monty'
- C.** SELECT Name
FROM Friends

```
WHERE Name = 'Matt'  
OR Name = 'Monty'
```


D. SELECT Name
FROM Friends matt, Friends monty
WHERE matt.name = 'Matt'
AND monty.name = 'Monty'

Answer: C

QUESTION NO: 29

Two rows are inserted into the empty table CountryLanguage:

```
+-----+-----+  
| CountryCode | Language |  
+-----+-----+  
| NLD | Papiamentu |  
| NLD | Sranantonga |  
+-----+-----+
```

Is it possible that a single statement was used to insert these rows?

Select the best response.

A. Yes, using this statement:

```
INSERT INTO CountryLanguage (CountryCode,Language)  
VALUES ('NLD','Papiamentu'),  
('NLD','Sranantonga')
```

B. Yes, using this statement:

```
INSERT INTO CountryLanguage (CountryCode,Language)  
VALUES ('NLD','Papiamentu')  
AND ('NLD','Sranantonga')
```

C. Yes, using this statement:

```
INSERT INTO CountryLanguage (CountryCode,Language)  
VALUES ('NLD','Papiamentu')  
VALUES ('NLD','Sranantonga')
```

D. No, you need at least two statements, like this:

```
INSERT INTO CountryLanguage (CountryCode,Language)  
VALUES ('NLD','Papiamentu')  
and then
```

```
INSERT INTO CountryLanguage (CountryCode,Language)  
VALUES ('NLD','Sranantonga')
```

Answer: A

QUESTION NO: 30

How many PRIMARY KEYs can be defined for a given table?

Select the best response.

- A. At most one PRIMARY KEY may be defined.
- B. Exactly one PRIMARY KEY must be defined.
- C. At least one PRIMARY KEY must be defined.
- D. For each column, at most one PRIMARY KEY may be defined.

Answer: A

QUESTION NO: 31

The following output describes the table City:

+-----+-----+-----+-----+-----+					
Field	Type	Null	Key	Default	Extra
+-----+-----+-----+-----+-----+					
Name	char(35)	NO	PRI		
Population	int(10) unsigned	YES		NULL	
Country	char(35)	NO	PRI		
+-----+-----+-----+-----+-----+					

Which of the following statements can be used to add a row for the city called 'Urk' in the country called 'The Netherlands'? Select the best response.

- A. INSERT INTO City('Urk',,'The Netherlands')
- B. INSERT INTO City VALUES ('Urk','The Netherlands')
- C. INSERT INTO City VALUES ('Urk',,'The Netherlands')
- D. INSERT INTO City(Name,Country) VALUES ('Urk','The Netherlands')

Answer: D

QUESTION NO: 32

The following output describes the table Country:

```
+-----+-----+-----+-----+-----+
```

```
| Field | Type | Null | Key | Default | Extra |
```

```
+-----+-----+-----+-----+-----+
```

```
| Code | char(3) | | PRI | | |
```

```
| Name | char(52) | | | |
```

```
| Capital | int(11) | | | |
```

```
+-----+-----+-----+-----+-----+
```

The following output describes the table City:

```
+-----+-----+-----+-----+-----+
```

```
| Field | Type | Null | Key | Default | Extra |
```

```
+-----+-----+-----+-----+-----+
```

```
| Id | int(11) | | PRI | NULL | auto_increment |
```

```
| Name | char(35) | | | |
```

```
| Population | int(11) | | | 0 | |
```

```
+-----+-----+-----+-----+-----+
```

The tables are related: Capital in Country references Id in City. You need to get a list of countries that contains the name of the country as well as the name of the country's capital. Which of the following statements can be used to do that? Select the best response.

A. SELECT Country.Name, Capital
FROM Country
INNER JOIN City

ON Capital = City.Id
B. SELECT Country.Name, City.Name
FROM Country
INNER JOIN City
C. SELECT Country.Name, City.Name
FROM Country INNER JOIN City ON Capital = City.Id
D. SELECT Country.Name, Capital.Name
FROM Country

Answer: C

QUESTION NO: 33

Given the following tables: mysql> DESCRIBE Country;

Field	Type	Null	Key	Default	Extra
Code	char(3)	NO	PRI		
Name	char(52)	NO			
Capital	int(11)	YES		NULL	

mysql> DESCRIBE CountryLanguage;

Field	Type	Null	Key	Default	Extra
Country	char(3)	NO	PRI		
Language	char(30)	NO	PRI		
Percentage	float(3,1)	NO		0.0	

The tables are related through Code in Country and Country in CountryLanguage. You want to obtain a list with the names of only those countries where English is spoken. Which of the following queries can be used to obtain such a list? Select the best response.

A. SELECT Country.Name
FROM Country
INNER JOIN CountryLanguage WHERE Language = 'English' B.
SELECT Country.Name FROM Country
INNER JOIN CountryLanguage
ON Country.Code = CountryLanguage.Country
WHERE Language = 'English'
C. SELECT Country
FROM Country
INNER JOIN CountryLanguage
ON Country.Code = CountryLanguage.Country
WHERE Language = 'English'
D. SELECT Country
FROM Language
WHERE CountryLanguage = 'English'

Answer: B

QUESTION NO: 34

The following output describes the table City:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRIMARY	0	
Name	char(35)	YES		NULL	
Population	int(10)	NO		0	

Which of the following statements will discard all data in the table without discarding the table structure? Select the best response.

- A. DELETE City
- B. DELETE FROM City
- C. DROP City
- D. DROP TABLE City

Answer: B

QUESTION NO: 35

The table keywords contains the following rows:

article_id	keyword
1	Linux
1	MySQL
1	Windows
2	Linux
2	MySQL
3	Linux
3	Windows
4	MySQL

8 rows in set (0.00 sec)

You want to retrieve all article_id values for those articles that are associated with the keyword 'MySQL' as well as the keyword 'Linux'. Which of the following statements can be used to achieve that? Select the best response.

- A. SELECT DISTINCT article_id
FROM keywords

WHERE keyword = 'MySQL' OR keyword = 'Linux'

B. SELECT article_id

FROM keywords

WHERE keyword = 'MySQL'

AND keyword = 'Linux' **C.** SELECT k2.article_id FROM keywords

AS k1

INNER JOIN keywords AS k2

ON k1.article_id = k2.article_id WHERE k1.keyword = 'MySQL' AND

k2.keyword = 'Linux'

D. You cannot do it in a single statement.

Answer: C

QUESTION NO: 36

When executing DELETE FROM articles LIMIT 10 Which rows will be deleted? Select the best response.

A. All the rows in the table.

B. The first 10 rows from the table sorted by primary key. **C.** The last 10 rows from the table sorted by primary key. **D.** The first 10 rows found by the server.

Answer: D

QUESTION NO: 37

In the context of MySQL client/server architecture, the role of the client program is to ... Select the best response.

A. initiate client/server communication.

B. send requests to the server to perform data manipulation.

C. send commands to control server behavior.

D. all of the above.

Answer: D

QUESTION NO: 38

In the context of MySQL client/server architecture, the role of the server program is to ... Select the best response.

A. receive and process commands and queries.

B. send SQL queries to client programs and receive result sets.

C. ensure that only one client may access a piece of data at any one time.

D. all of the above.

Answer: A

QUESTION NO: 39

The City table is created by executing the following statement: CREATE TABLE City (
ID int NOT NULL AUTO_INCREMENT, Name char(35) NOT NULL,
CountryCode char(3) NOT NULL, District char(20),
Population int NOT NULL, PRIMARY KEY (ID)
)

Which of the following statements can be used to ensure that no NULL values can be entered for the District column? Select the best response.

- A. UPDATE City SET District = NOT NULL
- B. UPDATE City MODIFY District NOT NULL
- C. ALTER TABLE City SET District NOT NULL
- D. ALTER TABLE City CHANGE District District char(20) NOT NULL

Answer: D

QUESTION NO: 40

A database management system is ... Select the best response.

- A. a computer hardware component where data is physically stored.
- B. a particular kind of computer program that stores and retrieves data on behalf of other applications.
- C. a particular kind of computer program that allows end-users to enter SQL statements.
- D. a collection of files that stores database data.

Answer: B

QUESTION NO: 41

A VIEW is ...

Select the best response.

- A. a temporary table.
- B. a special type of query that combines the data from multiple tables.
- C. a particular type of table that derives its structure and content from a query.
- D. another name for the output obtained by executing a SHOW statement.

Answer: C Explanation:

QUESTION NO: 42

The following output describes the structure of the Product table:

Field	Type	Null	Key	Default	Extra
Name	varchar(32)	NO			
Price	decimal(5,2)	NO			
Size	int(11)	YES		NULL	

Which of the following queries can be used to find all rows in the Product table for which the Size column contains the NULL value?
Select the best response.

- A. SELECT * FROM Product WHERE Size = 0
- B. SELECT * FROM Product WHERE Size = NULL
- C. SELECT * FROM Product WHERE Size IS NULL
- D. SELECT * FROM Product WHERE Size IS 'NULL'

Answer: C

QUESTION NO: 43

What is the main reason for adding indexes to tables? Select the best response.

- A. Only indexed columns may be used in expressions.
- B. Indexes enforce referential integrity.
- C. Indexes can speed up execution of queries.
- D. Indexes can speed up table maintenance tasks.

Answer: C Explanation:

QUESTION NO: 44

The following output describes the City table:

Field	Type	Null	Key	Default	Extra
Name	char(35)	NO	PRI		
CountryCode	char(3)	NO	PRI		
District	char(20)	YES		NULL	

The following statement is used to return all rows in the table: SELECT CountryCode,Name FROM

City

In what order are the rows returned? Select the best response.

- A. By CountryCode; then by Name.
- B. By Name; then by CountryCode; then by District.
- C. No guarantee can be made about the order.
- D. The rows are returned in the same order as they were added to the table

Answer: C

QUESTION NO: 45

The following output lists the contents of the City table:

+-----+-----+	
Name District	
+-----+-----+	
Dallas Texas	
New York New York	
Chicago Illinois	
Los Angeles California	
Houston Texas	
+-----+-----+	

Which result will be returned by executing the following statement? SELECT District, Count(District)

FROM City

GROUP BY District

Select the best response.

- A. +-----+-----+
| District | Count(District) |
+-----+-----+
| California | 1 |
| Illinois | 1 |
| New York | 1 |
| Texas | 1 |
+-----+-----+
- B. +-----+-----+
| District | Count(District) |
+-----+-----+
| California | 1 |
| Illinois | 1 |
| New York | 1 |
| Texas | 2 |
+-----+-----+
- C. +-----+-----+
| District | Count(District) |
+-----+-----+
| California | 1 |
| Illinois | 1 |
| New York | 1 |
| Texas | 1 |
| Texas | 2 |
+-----+-----+
- D. +-----+-----+
| District | Count(District) |
+-----+-----+
| California | 1 |
| Illinois | 1 |
| New York | 1 |
| Texas | 2 |
| Texas | 2 |

+-----+-----+

Answer: B

QUESTION NO: 46

Which of the following activities would imply using a join in a query? Select the best response.

- A.** Aggregating data from a given table.
- B.** Making particular groups of the rows in a table.
- C.** Making a list of all rows from a given table followed by all rows from another table.
- D.** Making a list of rows that combine data from a given table with data from another table.

Answer: D

QUESTION NO: 47

What is the purpose of the mysqldump program? Select the best response.

- A.** To migrate a non-MySQL database to a MySQL database.
- B.** To export MySQL databases to a text file.
- C.** To make a binary backup of a MySQL database.
- D.** To convert the binary log into a human readable format.

Answer: B

QUESTION NO: 48

What is the purpose of the mysqlimport program? Select the best response.

- A.** To import log files into a MySQL database table.
- B.** To import data from a binary log into a MySQL database table.
- C.** To import data from a text file into a MySQL database table.
- D.** To import tables from a non-MySQL database into a MySQL database.

Answer: C

QUESTION NO: 49

What is the effect of using the keyword LOCAL with the LOAD DATA INFILE statement? Select the best response.

- A.** With LOCAL, the server will request the file from the client host. Without LOCAL, the server will perform the operation using a file located on the server host.
- B.** With LOCAL, the server will perform the operation using a file located on the server host. Without LOCAL, the server will request the file from the client host.
- C.** The keyword LOCAL is optional. The server always performs the operation using a file located on the server host.
- D.** The keyword LOCAL is optional. The server always requests the file from the client host.

Answer: A

QUESTION NO: 50

Three UPDATE statements have been executed within one transaction. The transaction is still uncommitted when the connection between the server and the client issuing the commands is closed. What will happen to the transaction? Select the best response.

- A.** All changes are committed.
- B.** All changes are rolled back.
- C.** If the connection was closed normally at the clients' request, the changes are committed. If the connection closed abnormally, the changes are rolled back.
- D.** The changes are neither committed nor rolled back. The entire session state, including the pending changes are saved separately by the server, and the session is restored when the client reconnects.

Answer: B

MOBILE

ddl

create table Distributor

(

Distributor_ID varchar(10) ,

Distributor_Name varchar(20),

Address varchar(100),

Mobile decimal(23,0),

Email varchar(30),

constraint pk_distributor primary key(Distributor_ID)

);

create table Mobile_Master

(

IME_No varchar(10),

Model_Name varchar(20),

Manufacturer varchar(20),

Date_Of_Manufac date,

Warranty_in_Years int,

Price decimal(7,2),

Distributor_ID varchar(10),

constraint pk_ime primary key(IME_No),

foreign key(Distributor_ID) references Distributor(Distributor_ID)

);

create table Mobile_Specification

(

IME_No varchar(10),

Dimension varchar(20),

Weight varchar(20),

Display_Type varchar(20),

Display_Size varchar(20),

Internal_mem_in_MB int,

Memory_Card_Capacity_GB int,

Network_3G varchar(5),

GPRS varchar(5),

Bluetooth varchar(5),

Camera varchar(5),

Camera_Quality varchar(5) ,

OS varchar(20),

Battery_Life_Hrs int ,

constraint fk_ime foreign key(IME_No) references Mobile_Master(IME_No)

);

```
create table Customer_Info
```

```
(
```

```
Customer_ID varchar(10) ,
```

```
Customer_Name varchar(20),
```

```
Address varchar(100),
```

```
Mobile int,
```

```
Email varchar(30),
```

```
constraint pk_customer primary key(Customer_ID)
```

```
);
```

```
create table Sales_Info
```

```
(
```

```
SalesId int,
```

```
Sales_Date date,
```

```
IME_No varchar(10),
```

```
Price int,
```

```
Discount int,
```

```
Net_Amount int,
```

```
Customer_ID varchar(10),
```

```
constraint fk_sales primary key(SalesId),foreign key(Customer_ID) references Customer_Info(Customer_ID),  
foreign key(IME_No)
```

```
references Mobile_Master(IME_No)
```

);

dml

```
insert into distributor values('d01','sujit koley','hooghly',9051296438,'sujit9@gmail.com');
```

```
insert into distributor values('d02','chiranjib das','midnapure',9051297438,'chiru9@gmail.com');
```

```
insert into distributor values('d03','joydip das','kolkata',9051299438,'joy9@gmail.com');
```

```
insert into distributor values('d04','pappu barik','hooghly',9058296438,'pappu9@gmail.com');
```

```
insert into mobile_master values('ime01','n_series','nokia','2011-02-23',2,12000,'d01');
```

```
insert into mobile_master values('ime02','guru','samsung','2011-03-23',2,4000,'d03');
```

```
insert into mobile_master values('ime03','sII','samsung','2011-01-23',2,23000,'d04');
```

```
insert into mobile_master values('ime04','galaxy','samsung','2011-02-23',2,11000,'d03');
```

```
insert into mobile_master values('ime05','andro','nokia','2011-02-23',2,12000,'d04');
```

```
insert into mobile_master values('ime06','sI','samsung','2011-03-23',2,23000,'d01');
```

```
insert into mobile_master values('ime07','n_series','nokia','2011-02-23',2,14000,'d03');
```

```
insert into mobile_master values('ime08','n_series','nokia','2011-03-23',2,15000,'d02');
```

```
insert into mobile_master values('ime09','adrosmart','sony','2011-02-23',2,16000,'d04');
```

```
insert into mobile_master values('ime10','galaxy','samsung','2011-02-23',2,27000,'d02');
```

```
insert into mobile_specification values('ime01','2','','100 gm','led','5"',1000,4000,'yes','yes','yes','yes','2
mp','windows 7',24 );
```

```
insert into mobile_specification values('ime02','1','','150 gm','led','4"',1000,8000,'no','no','no','no','2
mp','sybian',20 );
```

```
insert into mobile_specification values('ime03','2','','110 gm','lcd','7"',1000,16000,'yes','yes','yes','yes','2
mp','android',20 );
```

```
insert into mobile_specification values('ime04','2','','125 gm','led','5"',1000,4000,'yes','yes','yes','yes','2
mp','android',20 );
```

```
insert into mobile_specification values('ime05','2','','135 gm','lcd','5"',1000,4000,'yes','yes','yes','yes','2
mp','android',24 );
```

```
insert into mobile_specification values('ime06','2','','145 gm','lcd','6"',1000,8000,'yes','yes','yes','yes','2
mp','android',24 );
```

```
insert into mobile_specification values('ime07','2','','200 gm','lcd','5"',1000,4000,'yes','yes','yes','yes','2
mp','windows 7',7 );
```

```
insert into mobile_specification values('ime08','2','','175 gm','lcd','5"',1000,4000,'yes','yes','yes','yes','2
mp','windows 7',24 );
```

```
insert into mobile_specification values('ime09','2','','109 gm','led','4"',1000,4000,'yes','yes','yes','yes','2
mp','android',7 );
```

```
insert into mobile_specification values('ime10','2','','123 gm','led','5"',1000,8000,'yes','yes','yes','yes','2
mp','android',28 );
```

```
insert into customer_info values('c01','soumik pal','kolkata',905125345,'soumik9@gmail.com');
```

insert into customer_info values('c02','gourav das','kolkata',905125349,'gourav9@gmail.com');

insert into customer_info values('c03','sanjib chatterjee','kolkata',905125745,'sanjib9@gmail.com');

insert into customer_info values('c04','samar roy','nodia',905125349,'samar9@gmail.com');

insert into customer_info values('c05','krish mahoto','kolkata',905125395,'krish9@gmail.com');

insert into sales_info values (1,'2012-12-23','ime01','12000',500,11500,'c02');

insert into sales_info values (2,'2012-11-23','ime02','4000',500,3500,'c01');

insert into sales_info values (3,'2012-10-23','ime01','12000',500,11500,'c03');

insert into sales_info values (4,'2012-12-29','ime03','23000',1000,22000,'c01');

insert into sales_info values (5,'2012-12-28','ime06','23000',1000,22000,'c05');

insert into sales_info values (6,'2012-12-26','ime07','14000',500,13500,'c04');

insert into sales_info values (7,'2012-12-25','ime08','15000',500,14500,'c02');

insert into sales_info values (8,'2012-12-24','ime09','16000',1000,15000,'c04');

ques & ans

Simple Questions:

Problem # 1: WAQ to Display the mobile details such as IMENO, Model Name produced by the manufacturer "Nokia".

Solution: select IME_NO,Model_Name from mobile_master where manufacturer='Nokia';

Problem # 2: WAQ to display IMENO, Model Name,Manufacturer,Camera Quality of mobiles whose camera quality is 5MP.

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,m2.camera_quality from mobile_master m1 join mobile_specification m2 on m1.ime_no=m2.ime_no where m2.camera_quality='5MP';

Problem # 3: WAQ to display Model Name,Quantity sold on the date 25-APR-12.

Solution: select model_name,count(ime_no) from sales_info where sales_date='23-APR-12' group by model_name;

Problem # 4: WAQ to display distributor id ,mobile supply details such as mobile model name, quantity supplied in sorted order of distributor id.

Solution: select Distributor_ID,Model_Name,count(Model_Name) from Mobile_Master group by Distributor_ID,Model_Name order by Distributor_id;

Problem # 5: WAQ to display the IMENO,model name,manufacturer,price and discount of all mobiles regardless of whether the mobile is sold or not.

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,m1.price,s.discount from mobile_master m1 left outer join sales_info s on m1.ime_no=s.ime_no;

Problem # 6: WAQ to display the distributor details such as distributor name,mobile number and email of the model 'Nokia 1100'.

Solution: select Distributor_Name,Mobile from Distributor where Distributor_Id=(select distributor_id from mobile_master where model_name='Nokia 1100');

Problem # 7: WAQ to display the Ime No and Model Name of mobiles which are not sold(Hint : use minus operator)

Solution: select ime_no ,model_name from mobile_master minus select ime_no ,model_name from sales_info;

Problem # 8: WAQ to display the Ime No and Model Name of mobiles which are sold(Hint : use intersect operator)

Solution: select ime_no ,model_name from mobile_master intersect select ime_no ,model_name from sales_info;

Problem # 9: WAQ to display the ImeNO, Model Name,Manufacturer, Price and NewPrice of all mobiles.

(Hint : find new price as 10% of the price with column name "New Price")

Solution: select ime_no,model_name,manufacturer,price,price+(price*10/100) "New Price" from mobile_master;

Problem # 10: WAQ to display mobile model, manufacturer and price for the mobiles having a price range from 8500 to 25300.

Solution: select model_name,manufacturer,price from mobile_master where price between 8500 and 25300;

Average Questions:

Problem # 1: WAQ to display the Model Name,Manufacturer, Price , Warranty , Internal memory, memory card capacity,gprs support,bluetooth,camera quality and OS for the mobile with IME NO "MC1000104" .

Solution: select

m1.model_name,m1.manufacturer,m1.warranty_in_years,m1.price,m2.Internal_mem_in_MB,m2.Memory_Card_Capacity_GB, m2.GPRS,m2.Bluetooth,m2.Camera_Quality,m2.OS from mobile_master m1 join mobile_specification m2 on m1.IME_No=m2.IME_No where m1.IME_no='MC1000104';

Problem # 2: WAQ to display IMENO, Model Name,Manufacturer,Price ,GPRS information,Memory card support of mobiles which has GPRS support with memory card capacity 16GB or above.

Solution: select

m1.ime_no,m1.model_name,m1.manufacturer,m1.price,m2.gprs,m2.Memory_Card_Capacity_GB from mobile_master m1 join mobile_specification m2 on m1.ime_no=m2.ime_no where m2.GPRS='Yes' and m2.Memory_Card_Capacity_GB>=16;

Problem # 3: WAQ to display the customer name ,mobile purchase details such as IMENO,Model Name ,Purchase Date,Net amount paid in sorted order of customer name.

Solution: select c1.Customer_Name,m1.IME_NO,m1.Model_Name,m1.Sales_Date,m1.Net_Amount from Customer_Info c1 join Sales_info m1 on m1.Customer_ID=c1.Customer_ID order by c1.Customer_Name;

Problem # 4: WAQ to display the distributor details such as distributor id ,name ,address,contact no who has

supplied the maximum number of mobiles.

Solution: select distributor_id,distributor_name,address,mobile,email from distributor where distributor_id=(select distributor_id from mobile_master having count(distributor_id)=(select max(count(distributor_id)) from mobile_master group by distributor_id) group by distributor_id);

Problem # 5: WAQ to display the IMENO,model name,manufacturer,price and discount of all mobiles regardless of whether the mobile is sold or not.

[Hint: If not sold, display discount as "Not Sold"]

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,m1.price,nvl(to_char(m2.discount),'Not Sold') "discount" from mobile_master m1 left outer join sales_info s on m1.ime_no=s.ime_no;

Problem # 6: WAQ to display the report containing the sales date and total sales amount of the dates between 20-APR-12 and 25-APR-12.

(Hint : total sales amount column should be displayed as "Total Sales Amount")

Solution: select sales_date,sum(net_amount) "Total Sales Amount" from sales_info where sales_date between '20-APR-12' and '25-APR-12' group by sales_date;

Problem # 7: WAQ to display mobile imeno,model name,manufacturer and price of the mobiles which are having the longest battery life.

Solution: select ime_no,model_name,manufacturer,price from mobile_master where ime_no in(select ime_no from mobile_specification where battery_life_hrs=(select max(battery_life_hrs) from mobile_specification));

Problem # 8: WAQ to display the ImeNO, Model Name,Manufacturer, Price of mobiles having the maximum price.

Solution: select ime_no,model_name,manufacturer,price from mobile_master where ime_no in(select ime_no from mobile_master where price=(select max(price) from mobile_master));

Problem # 9: WAQ to display the customer details such as Customer ID,Customer Name, Address, Total Purchase amount.

Solution: select c1.Customer_ID,c1.Customer_Name,c1.Address,(select sum(Net_Amount) from sales_info where Customer_id=c1.Customer_ID) "Total Purchase Amount" from Customer_info c1;

Problem # 10: WAQ to display the most costly mobile information such as mobile model, manufacturer and price manufactured by "Samsung".

Solution: s

Complex Questions:

•Problem # 1: WAQ to display the customer details such as Customer ID,Customer Name, Address and Total Purchase amount having the maximum purchase amount.

Solution: select Customer_ID,Customer_Name,Address from customer_info where customer_id=(select customer_id from sales_info having sum(Net_Amount)=(select max(sum(net_amount)) from sales_info group by customer_id) group by customer_id);

Problem # 2: WAQ to determine whether the mobile with ime no "MC1000105" is been sold out or not and display the model name,sales status.(Hint: If sold display status as "Sold Out" with column name "Sales Status").

Solution: select model_name,(select 'Sold Out' from sales_info where ime_no='MC1000105') "Sales Status" from mobile_master where ime_no='MC1000105' ;

Problem # 3: WAQ to display the mobile information such as ime no,model name,manufacturer ,distributor id,distributor name and price supplied by the distributor named 'AXA Ltd' .

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,d1.distributor_id,d1.distributor_name,m1.price from mobile_master m1 join distributor d1 on m1.distributor_id=d1.distributor_id and

d1.distributor_id=(select distributor_id from distributor where distributor_name='AXA Ltd');

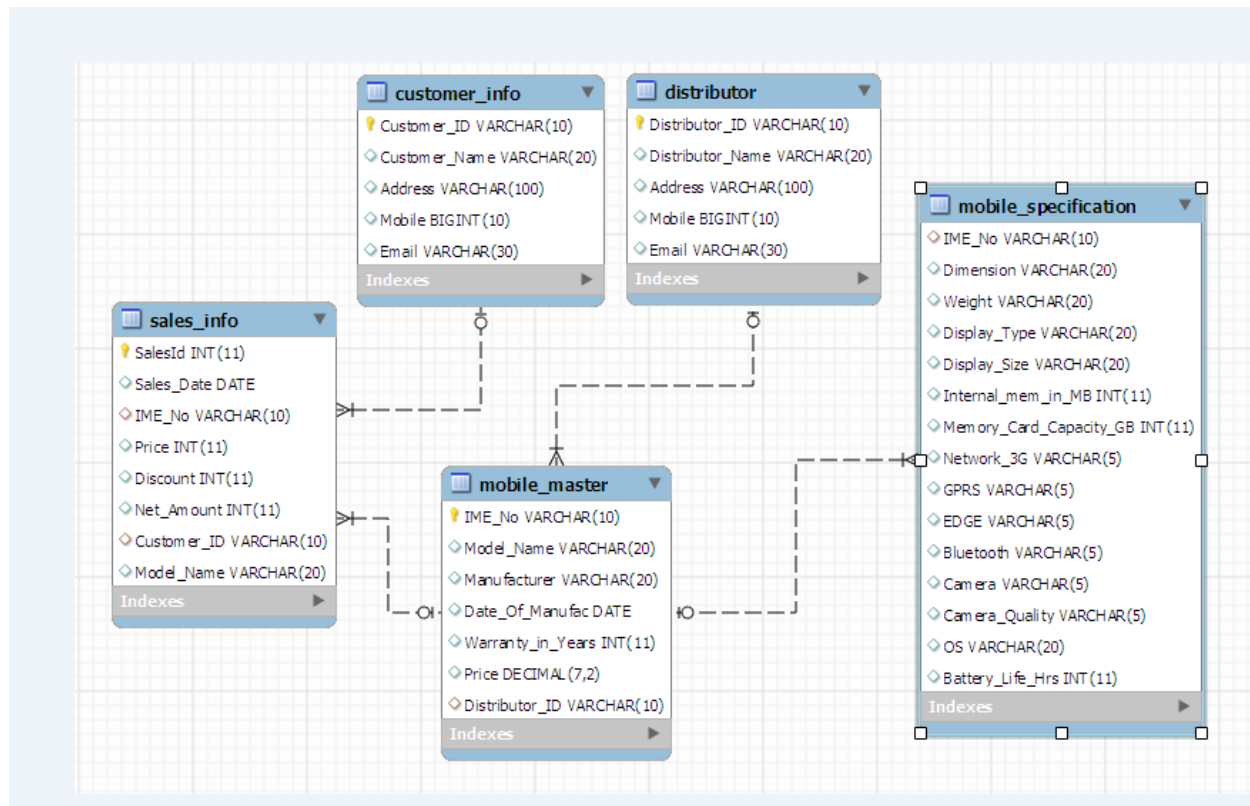
Problem # 4: WAQ to display distributor details who supplies mobile with the following specifications such as 3G Network, Android OS, 5 MP Camera.

Solution: select distributor_id,distributor_name,address,mobile from distributor where distributor_id IN (select distributor_id from mobile_master where ime_no IN (select ime_no from mobile_specification where network_3g='Yes' and os LIKE '%Android%' and camera_quality='3.5MP'));

Problem # 5: WAQ to Display the maximum sold mobile model name and manufacturer .

Solution: select distinct model_name,manufacturer from mobile_master where model_name=(select model_name from sales_info having count(model_name)=(select max(count(model_name))from sales_info group by model_name)group by model_name)

MOBILE MANAGEMENT SYSTEM



DDL COMMANDS

-- Distributor Info table

```
CREATE DATABASE mobile_db;
```

```
USE mobile_db;
```

```
create table Distributor
```

```
(
```

```
Distributor_ID varchar(10) ,
```

```
Distributor_Name varchar(20),
```

```
Address varchar(100),
```

```
Mobile BIGINT(10),
```

```
Email varchar(30), constraint pk_distributor primary key(Distributor_ID) );
```

-- Mobile master table

create table Mobile_Master

(

IME_No varchar(10), Model_Name varchar(20), Manufacturer varchar(20), Date_Of_Manufac date,

Warranty_in_Years INT, Price DECIMAL(7,2), Distributor_ID varchar(10),

**constraint pk_ime primary key(IME_No), foreign key(Distributor_ID) references
Distributor(Distributor_ID)**

);

-- Mobile specification table

create table Mobile_Specification

(

IME_No varchar(10), Dimension varchar(20), Weight varchar(20), Display_Type varchar(20),

Display_Size varchar(20),

**Internal_mem_in_MB INT, Memory_Card_Capacity_GB INT, Network_3G varchar(5), GPRS varchar(5),
EDGE varchar(5), Bluetooth varchar(5),**

Camera varchar(5), Camera_Quality varchar(5) , OS varchar(20), Battery_Life_Hrs INT ,

constraint fk_ime foreign key(IME_No) references Mobile_Master(IME_No)

);

-- Customer Information table

create table Customer_Info

(

Customer_ID varchar(10) ,

Customer_Name varchar(20),

Address varchar(100),

Mobile BIGINT(10),

Email varchar(30),constraint pk_customer primary key(Customer_ID));

-- Sales information table

create table Sales_Info

(

SalesId INT,

Sales_Datedate,IME_No varchar(10),Price INT,DiscountINT,Net_AmountINT,Customer_ID
varchar(10),Model_Name varchar(20),

constraint fk_sales primary key(SalesId),foreign key(Customer_ID) references
Customer_Info(Customer_ID), foreign key(IME_No) references Mobile_Master(IME_No));

DML COMMANDS

-- Data for Distributor table

insert into Distributor values('D001','AXA Ltd','4th Floor CDG
Avenue',9897874563,'axaltd@gmail.com');

insert into Distributor values('D002','KK Ditributors','Rajiv Gandhi
Salai,Chennai',9112225632,'kandk@rocketmail.com');

insert into Distributor values('D003','Vasanth Traders','NDR
Building,Chennai',9844555883,'vasanth@gmail.com');

-- Data for Mobile_Master table

insert into Mobile_mastervalues('MC1000100','Nokia C5-03','Nokia','2005-05-11',1,9500,'D001');

insert into Mobile_mastervalues('MC1000102','Nokia Lumia','Nokia','2011-08-16',2,35350,'D001');

insert into Mobile_mastervalues('MC1000103','Samsung GalaxyTAB','Samsung','2010-05-
06',2,32338,'D001');

insert into Mobile_mastervalues('MC1000104','Samsung Galaxy Y','Samsung','2010-05-
19',1,24569,'D001');

insert into Mobile_mastervalues('MC1000105','Nokia 5230','Nokia','2003-03-09',1,6123,'D002');

insert into Mobile_mastervalues('MC1000106','Samsung C3010','Samsung','2005-04-19',1,4000,'D002');

insert into Mobile_mastervalues('MC1000107','Sony Experia','Sony Erricson','2011-05-30',1,16500,'D003');

insert into Mobile_mastervalues('MC1000108','Nokia 1100','Nokia','2001-03-01',1,2100,'D003');

insert into Mobile_mastervalues('MC1000109','Nokia C5-03','Nokia','05-03-09',1,9500,'D001');

insert into Mobile_mastervalues('MC1000110','Samsung GalaxyTAB','Samsung','2010-06-05',2,32338,'D002');

insert into Mobile_mastervalues('MC1000111','Nokia C5-03','Nokia','2005-03-09',1,9500,'D001');

-- Data for Mobile_Specification

insert into Mobile_Specificationvalues('MC1000100','105.8 x 51 x 13.8 mm','93g','TFT touchscreen','360 x 640 pixels',128,16,'Yes','Yes','Yes','Yes','Yes','5MP','Symbian OS v9.4',600);

insert into Mobile_Specificationvalues('MC1000102','116.5 x 61.2 mm','142g','AMOLED touchscreen','480 x 800 pixels',512,16,'Yes','Yes','Yes','Yes','Yes','8MP','Windows 7.5 Mango',210);

insert into Mobile_Specificationvalues('MC1000103','256.6x175.3 x 9.7 mm','581g','PLS TFT touchscreen','800 x 1280 pixels',1024,32,'Yes','Yes','Yes','Yes','Yes','3.5MP','Android OS, v4.0.3',320);

insert into Mobile_Specificationvalues('MC1000104','109.8 x 60 x 12 mm','109g','TFT touchscreen','240 x 320 pixels',160,32,'Yes','Yes','Yes','Yes','Yes','3.5MP','Android OS, v2.3',360);

insert into Mobile_Specificationvalues('MC1000105','108 x 43.5 x 10.5 mm','78g','TFT, 256K colors','240 x 320 pixels',30,8,'No','Yes','Yes','Yes','Yes','2MP','Nokia OS',406);

insert into Mobile_Specificationvalues('MC1000106','76 x 43.5 x 7.5 mm','82g','TFT,65K colors','120 x 120 pixels',10,2,'No','Yes','Yes','Yes','Yes','VGA','SOS',1200);

insert into Mobile_Specificationvalues('MC1000107','256.6x175.3 x 9.7 mm','95g','TFT touchscreen','240 x 320 pixels',160,16,'Yes','Yes','Yes','Yes','Yes','5MP','Android OS, v2.3',390);

insert into Mobile_Specificationvalues('MC1000108','76 x 43.5 x 7.5 mm','82g','TFT,65K colors','120 x 120 pixels',10,2,'No','No','No','No','No','VGA','Nokia OS',1200);

-- Data for Customer_Info table

```
insert into Customer_Infovalues('C001','Shyam',' No.3/5,serene flats Chennai-96',9962100341,'shyam23@gmail.com');
```

```
insert into Customer_Infovalues('C002','Asha',' No.6/2, Gandhi Nagar, Adyar Chennai-20',8952100123,'Asha46@rediffmail.com');
```

```
insert into Customer_Infovalues('C003','Mano',' No.12, Camp Road,Tambaram Chennai-80',9841400341,'mano_sundar@gmail.com');
```

```
insert into Customer_Infovalues('C004','Naresh',' No.46,Lotus Garden,Kilpauk Chennai-32',8962122346,'Naresh12@yahoo.co.in');
```

-- Data for Sales_Info table

```
insert into Sales_Infovalues(1001,'2012-04-20','MC1000100',9500,2000,7500,'C001','Nokia C5-03');
```

```
insert into Sales_Infovalues(1002,'2012-04-21','MC1000102',35350,350,35000,'C001','Nokia Lumia');
```

```
insert into Sales_Infovalues(1003,'2012-04-21','MC1000103',32338,338,32000,'C002','Samsung GalaxyTAB');
```

```
insert into Sales_Infovalues(1004,'2012-04-22','MC1000104',32338,338,32000,'C003','Samsung Galaxy Y');
```

```
insert into Sales_Infovalues(1005,'2012-04-23','MC1000105',6123,123,6000,'C004','Nokia 5230');
```

```
insert into Sales_Infovalues(1006,'2012-04-23','MC1000108',2100,100,2000,'C003','Nokia 1100');
```

```
insert into Sales_Infovalues(1007,'2012-04-23','MC1000109',2100,100,2000,'C003','Nokia C5-03');
```

```
insert into Sales_Infovalues(1008,'2012-04-23','MC1000111',2100,100,2000,'C003','Nokia C5-03');
```

QUESTIONS AND ANSWERS

Simple Questions:

Problem # 1: WAQ to Display the mobile details such as IMENO, Model Name produced by the manufacturer "Nokia".

Solution: select IME_NO,Model_Name from mobile_master where manufacturer='Nokia';

Problem # 2: WAQ to display IMENO, Model Name,Manufacturer,Camera Quality of mobiles whose camera quality is 5MP.

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,m2.camera_quality from mobile_master m1 join mobile_specification m2 on m1.ime_no=m2.ime_no where m2.camera_quality='5MP';

Problem # 3: WAQ to display Model Name,Quantity sold on the date 25-APR-12.

Solution: select model_name,count(ime_no) from sales_info where sales_date='23-APR-12' group by model_name;

Problem # 4: WAQ to display distributor id ,mobile supply details such as mobile model name, quantity supplied in sorted order of distributor id.

Solution: select Distributor_ID,Model_Name,count(Model_Name) from Mobile_Master group by Distributor_ID,Model_Name order by Distributor_id;

Problem # 5: WAQ to display the IMENO,modelname,manufacturer,price and discount of all mobiles regardless of whether the mobile is sold or not.

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,m1.price,s.discount from mobile_master m1 left outer join sales_infos on m1.ime_no=s.ime_no;

Problem # 6: WAQ to display the distributor details such as distributor name,mobile number and email of the model 'Nokia 1100'.

Solution: select Distributor_Name,Mobile from Distributor where Distributor_Id=(select distributor_id from mobile_master where model_name='Nokia 1100');

Problem # 7: WAQ to display the Ime No and Model Name of mobiles which are not sold(Hint : use minus operator)

Solution: select ime_no ,model_name from mobile_master minus select ime_no ,model_name from sales_info;

Problem # 8: WAQ to display the Ime No and Model Name of mobiles which are sold(Hint : use intersect operator)

Solution: select ime_no ,model_name from mobile_master intersect select ime_no ,model_name from sales_info;

Problem # 9: WAQ to display the ImeNO, Model Name,Manufacturer, Price and NewPrice of all mobiles.

(Hint : find new price as 10% of the price with column name "New Price")

Solution: select ime_no,model_name,manufacturer,price,price+(price*10/100) "New Price" from mobile_master;

Problem # 10: WAQ to display mobile model, manufacturer and price for the mobiles having a price range from 8500 to 25300.

Solution: select model_name,manufacturer,price from mobile_master where price between 8500 and 25300;

Average Questions:

Problem # 1: WAQ to display the Model Name,Manufacturer, Price , Warranty , Internal memory, memory card capacity,gprssupport,bluetooth,camera quality and OS for the mobile with IME NO "MC1000104" .

Solution: select
m1.model_name,m1.manufacturer,m1.warranty_in_years,m1.price,m2.Internal_mem_in_MB,m2.Memory_Card_Capacity_GB, m2.GPRS,m2.Bluetooth,m2.Camera_Quality,m2.OS from mobile_master m1 join mobile_specification m2 on m1.IME_No=m2.IME_No where m1.IME_no='MC1000104';

Problem # 2: WAQ to display IMENO, Model Name,Manufacturer,Price ,GPRS information,Memory card support of mobiles which has GPRS support with memory card capacity 16GB or above.

Solution: select
m1.ime_no,m1.model_name,m1.manufacturer,m1.price,m2.gprs,m2.Memory_Card_Capacity_GB from mobile_master m1 join mobile_specification m2 on m1.ime_no=m2.ime_no where m2.GPRS='Yes' and m2.Memory_Card_Capacity_GB>=16;

Problem # 3:WAQ to display the customer name ,mobile purchase details such as IMENO,Model Name ,Purchase Date,Net amount paid in sorted order of customer name.

Solution: select c1.Customer_Name,m1.IME_NO,m1.Model_Name,m1.Sales_Date,m1.Net_Amount from Customer_Info c1 join Sales_info m1 on m1.Customer_ID=c1.Customer_ID order by c1.Customer_Name;

Problem # 4: WAQ to display the distributor details such as distributor id ,name ,address,contact no who has supplied the maximum number of mobiles.

Solution: select distributor_id,distributor_name,address,mobile,email from distributor where distributor_id=(select distributor_id from mobile_master having count(distributor_id)=(select max(count(distributor_id)) from mobile_master group by distributor_id) group by distributor_id);

Problem # 5: WAQ to display the IMENO,modelname,manufacturer,price and discount of all mobiles regardless of whether the mobile is sold or not.

[Hint: If not sold, display discount as "Not Sold"]

Solution: select m1.ime_no,m1.model_name,m1.manufacturer,m1.price,nvl(to_char(m2.discount),'Not Sold') "discount" from mobile_master m1 left outer join sales_infos on m1.ime_no=s.ime_no;

Problem # 6: WAQ to display the report containing the sales date and total sales amount of the dates between 20-APR-12 and 25-APR-12.

(Hint : total sales amount column should be displayed as "Total Sales Amount")

Solution: select sales_date,sum(net_amount) "Total Sales Amount" from sales_info

where sales_date between '20-APR-12' and '25-APR-12' group by sales_date;

Problem # 7: WAQ to display mobile imeno,modelname,manufacturer and price of the mobiles which are having the longest battery life.

Solution: select ime_no,model_name,manufacturer,price from mobile_master where ime_no in(select ime_no from mobile_specification where battery_life_hrs=(select max(battery_life_hrs) from mobile_specification));

Problem # 8: WAQ to display the ImeNO, Model Name,Manufacturer, Price of mobiles having the maximum price.

Solution: select ime_no,model_name,manufacturer,price from mobile_master where ime_no in(select ime_no from mobile_master where price=(select max(price) from mobile_master));

Problem # 9: WAQ to display the customer details such as Customer ID,Customer Name, Address, Total Purchase amount.

Solution: select c1.Customer_ID,c1.Customer_Name,c1.Address,(select sum(Net_Amount) from sales_info where Customer_id=c1.Customer_ID) "Total Purchase Amount" from Customer_info c1;

Problem # 10: WAQ to display the most costly mobile information such as mobile model, manufacturer and price manufactured by "Samsung".

Solution: s

Complex Questions:

•**Problem # 1:** WAQ to display the customer details such as Customer ID,Customer Name, Address and Total Purchase amount having the maximum purchase amount.

Solution: select Customer_ID,Customer_Name,Address from customer_info where customer_id=(select customer_id from sales_info having sum(Net_Amount)=(select max(sum(net_amount)) from sales_info group by customer_id) group by customer_id);

Problem # 2: WAQ to determine whether the mobile with ime no "MC1000105" is been sold out or not and display the model name,sales status.(Hint: If sold display status as "Sold Out" with column name "Sales Status").

Solution: select model_name,(select 'Sold Out' from sales_info where ime_no='MC1000105')"Sales Status" from mobile_master where ime_no='MC1000105' ;

Problem # 3: WAQ to display the mobile information such as imeno,modelname,manufacturer ,distributor id ,distributor name and price supplied by the distributor named 'AXA Ltd' .

Solution: select
m1.ime_no,m1.model_name,m1.manufacturer,d1.distributor_id,d1.distributor_name,m1.price from
mobile_master m1 join distributor d1 on m1.distributor_id=d1.distributor_id and
d1.distributor_id=(select distributor_id from distributor where distributor_name='AXA Ltd');

Problem # 4: WAQ to display distributor details who supplies mobile with the following specifications such as 3G Network, Android OS, 5 MP Camera.

Solution: select distributor_id,distributor_name,address,mobile from distributor where distributor_id
IN (select distributor_id from mobile_master where ime_no IN (select ime_no from mobile_specification
where network_3g='Yes' and os LIKE '%Android%' and camera_quality='3.5MP'));

Problem # 5: WAQ to Display the maximum sold mobile model name and manufacturer .

Solution: select distinct model_name,manufacturer from mobile_master where model_name=(select
model_name from sales_info having count(model_name)=(select max(count(model_name))from
sales_info group by model_name)group by model_name)

```
CREATE DATABASE PAY;  
USE PAY;
```

```
CREATE TABLE ELEAVE (LEAVE_CATEGORY VARCHAR (1), CL NUMERIC, EL NUMERIC, ML  
NUMERIC, CONSTRAINT PK_LEAVE PRIMARY KEY (LEAVE_CATEGORY));
```

```
CREATE TABLE SALARY(EMPLOYEE_CATEGORY VARCHAR(1), BASIC  
NUMERIC(7,2),TRAVELLING_ALLOWANCE NUMERIC(6,2),DEARNNESS_ALLOWANCE  
NUMERIC(6,2),HOUSE_RENT_ALLOWANCE NUMERIC(6,2),LOCATION_ALLOWANCE  
NUMERIC(6,2),PROVIDENT_FUND NUMERIC(6,2),MEDICAL_ALLOWANCE  
NUMERIC(6,2),PROFTAX NUMERIC(6,2),INSURANCE NUMERIC(6,2), CONSTRAINT PK_SALARY  
PRIMARY KEY(EMPLOYEE_CATEGORY));
```

```
CREATE TABLE DEPARTMENT (DEPTID VARCHAR (10), DEPTNAME VARCHAR (20), LOCATION  
VARCHAR (20), CONSTRAINT PK_DEPARTMENT PRIMARY KEY (DEPTID));
```

```
CREATE TABLE EMPLOYEE (EMPID VARCHAR (10), EMPNAME VARCHAR (20), DEPTID VARCHAR  
(10), JOINING_DT DATE, DOB DATE, YRS_OF_EXP NUMERIC,EMPLOYEE_CATEGORY  
VARCHAR(1), LEAVE_CATEGORY VARCHAR(1), CONSTRAINT PK_EMPLOYEE PRIMARY  
KEY(EMPID), CONSTRAINT fK_DID FOREIGN KEY(DEPTID) REFERENCES  
DEPARTMENT(DEPTID),CONSTRAINT FK_ECAT FOREIGN KEY(EMPLOYEE_CATEGORY)  
REFERENCES SALARY(EMPLOYEE_CATEGORY),CONSTRAINT FK_LEAV FOREIGN  
KEY(LEAVE_CATEGORY) REFERENCES ELEAVE(LEAVE_CATEGORY));
```

```
CREATE TABLE EMPLOYEE_LEAVE (EMPID VARCHAR (10), FROM_DATE DATE, TO_DATE DATE,  
TOTAL_LEAVES NUMERIC, LEAVE_TYPE VARCHAR (5), CONSTRAINT FK_EMPLOYEE_LEAVE  
FOREIGN KEY (EMPID) REFERENCES EMPLOYEE(EMPID));
```

```
CREATE TABLE PAYROLL (TRANSNO NUMERIC, EMPID VARCHAR (10), MONTH VARCHAR (10),  
YEAR NUMERIC (4), TOTALEARNING NUMERIC,TOTALDEDUCTION NUMERIC, LOPAMOUNT  
NUMERIC, NETPAY NUMERIC, CONSTRAINT EMP_PAY PRIMARY KEY (TRANSNO));
```

```
INSERT INTO SALARY VALUES ('A',6000,1400,1200,1750,900,750,1450,260,1000);  
INSERT INTO SALARY VALUES ('B',5500,1100,900,1450,700,650,1250,230,900);  
INSERT INTO SALARY VALUES ('C',5000,1000,800,1350,650,500,1050,190,700);  
INSERT INTO SALARY VALUES ('D',4000,900,750,1150,450,400,750,120,600);  
INSERT INTO ELEAVE VALUES ('X',18,5,10);  
INSERT INTO ELEAVE VALUES ('Y',15,3,8);  
INSERT INTO ELEAVE VALUES ('Z',12,3,7);  
INSERT INTO DEPARTMENT VALUES ('D001','IT','BANGALORE');  
INSERT INTO DEPARTMENT VALUES ('D002','SALES','CHENNAI');
```

```

INSERT INTO DEPARTMENT VALUES ('D003','HR','COCHIN');
INSERT INTO DEPARTMENT VALUES ('D004','TRANSPORT','DELHI');
INSERT INTO EMPLOYEE VALUES ('E001','RAM','D001','2001-03-1','1979-06-9',11,'A','X');
INSERT INTO EMPLOYEE VALUES ('E002','DEV','D001','2011-6-20','1987-03-8',1,'D','Z');
INSERT INTO EMPLOYEE VALUES ('E003','SAM','D001','11-4-11','84-11-4',1,'D','Z');
INSERT INTO EMPLOYEE VALUES ('E004','STEVE',NULL,NULL,'1985-1-3',NULL,NULL,NULL);
INSERT INTO EMPLOYEE VALUES ('E005','OLGA','D002','2002-8-27','2012-6-9',5,'B','Y');
INSERT INTO EMPLOYEE VALUES ('E006','SANGEETHA','D003','2010-7-05','1982-9-2',7,'B','Y');
INSERT INTO EMPLOYEE VALUES ('E007','PRAKASH','D003','2012-6-9','1883-7-7',3,'D','Z');
INSERT INTO EMPLOYEE VALUES ('E008','SANA',NULL,NULL,NULL,NULL,NULL,NULL);
INSERT INTO EMPLOYEE VALUES ('E009','MANO','D002','2008-11-01','1988-05-12',4,'C','Y');
INSERT INTO EMPLOYEE VALUES ('E010','DINU','D001','2006-03-11','1982-03-08',6,'B','Y');
INSERT INTO EMPLOYEE_LEAVE VALUES ('E001','2012-1-1','2012-1-7',6,'CL');
INSERT INTO EMPLOYEE_LEAVE VALUES ('E002','2012-2-1','2012-2-2',1,'CL');
INSERT INTO EMPLOYEE_LEAVE VALUES ('E003','2012-3-1','2012-4-1',31,'ML');
INSERT INTO EMPLOYEE_LEAVE VALUES ('E004','2012-4-4','2012-4-5',1,'OH');
INSERT INTO EMPLOYEE_LEAVE VALUES ('E005','2012-5-5','2012-5-8',3,'EL');
INSERT INTO EMPLOYEE_LEAVE VALUES ('E001','2012-6-10','2012-6-12',2,'CL');
INSERT INTO PAYROLL VALUES (1,'E001','APR',2012,12700,2010,0,10690);
INSERT INTO PAYROLL VALUES (2,'E002','MAR',2012,8000,1120,0,6880);
INSERT INTO PAYROLL VALUES (3,'E003','APR',2012,8000,1120,3360,4640);
INSERT INTO PAYROLL VALUES (4,'E005','JAN',2012,10700,1780,0,8920);
INSERT INTO PAYROLL VALUES (5,'E006','JAN',2012,10700,1780,500,8420);

```

```

ALTER TABLE PAYROLL ADD FOREIGN KEY(EMPID) REFERENCES EMPLOYEE(EMPID);

```

questions and answers

Payroll Schema – Simple Questions

Problem # 1: Write a query to display Employee ID, Employee Name, Department ID and Department Name of all employees who has a department assigned.

```

àselect e.empid,e.empname,e.deptid,d.deptname from employee e, department d where e.deptid=d.deptid and e.deptid is not null;

```

Problem # 2: Write a query to display the Employee ID, Employee Name, Basic Pay of all employees who are in employee category 'A'

```

à select e.empid,e.empname,s.basic from employee e,salary s where e.employee_category=s.employee_category and s.employee_category='A';

```

Problem # 3: Write a query to display the Employee ID, Employee Name, Department ID and Department Name of all employees who has a department assigned and department location is 'CHENNAI'.

```

à select e.empid,e.empname,e.deptid,d.deptname from employee e,department d where e.deptid is not null and

```

e.deptid=d.deptid and d.location='Chennai';

Problem # 4: Write a query to display the employee ID and employee name of the employees who have not been assigned a department yet.

à select empid,empname from employee where deptid is null;

Problem # 5: Write a query to display the employee ID, employee name and joining date of the employees who joined before 2005.

à select empid,empname,joining_dt from employee where joining_dt<'2005-1-1';

Problem # 6: write a query to display employee name and date of joining for all employees.(Date should be displayed in the format “23/JANUARY/2012” with Alias “ JOINING _DATE” in select statement)

à select empname,date_format(joining_dt,'%d' '/' '%M' '/' '%Y') as JOINING _DATE from employee;

Problem # 7: Write a query to display the employee ID, employee name and joining date of the employees who joined between Jan 1 2005 and Dec 31'st 2010

à select empid,empname,joining_dt from employee where joining_dt between '2005-1-1' and '2010-31-12';

Problem # 8: Write a query to display the employee ID, employee name and joining date of the employees who joined in MARCH.

à select empid,empname,joining_dt from employee where extract(month from joining_dt)=3;

Problem # 9: Write a query to display all employee names which begins with 'R'.

à select empname from employee where empname like 'R%';

Problem # 10: Write a query to display the first five employees name in the employee table and the respective row number (use ROWNUM for identifying the first five records)

à select @rowno:=@rowno+1 as ROWNUM,empname from employee,(select @rowno:=0) r limit 0,5;

Payroll Schema – Average Questions

Problem # 1: Write a query to display the EmployeeID, Employee Name,Net Pay of an employee whose ID is "E001" for the month of APRIL

à Write a query to display the EmployeeID, Employee Name,Net Pay of an employee whose ID is "E001" for

the month of APRIL

Problem # 2: Write a query to display the department id and no of employees in each department sorted by department id. (Exclude department with null values).

à select d.deptid,count(e.empname) from department d,employee e where e.deptid=d.deptid group by d.deptid order by deptid;

Problem # 3: Write a query to display the EmployeeID, Employee Name and the total number of leaves each employee has taken with "Total_Leaves" as alias.

Hint: For Example, if employee "E001" has taken 2 days leave on January and 3 days leave of February then his total number of leaves will be 5 days. Similarly display the total number of leaves for all employees.

à select e.empid,e.empname,sum(el.total_leaves) from employee e, employee_leave el where e.empid=el.empid group by el.empid;

Problem # 4: Write a query to display the EmployeeID, Employee Name, DOB and Age in Years without decimals with alias name "Age".

Hint: Formula for age calculation is Age = current date- dob/12, round this to the nearest whole number.

à select empid,empname,dob,round(((datediff(current_date,dob)/30)/12) as Age from employee;

Problem # 5: Write a query to display employee id, employee name of all employees who doesn't have LOP amount for the month of APR and year 2012.

à select e.empid,e.empname from employee e, payroll p where e.empid=p.empid and p.lopamount=0 and p.month='apr' and p.year=2012;

Problem # 6: Write a query to display employee name, professional tax, netpay of employees with employee category 'A'

à select e.empid,e.empname,s.proftax,p.netpay from employee e,salary s,payroll p where e.employee_category=s.employee_category and e.employee_category='a' and e.empid=p.empid;

Problem # 7: Write a query to display employee id, employee name,department id who are having netpay in the range 10000 – 20000

à select e.empid,e.empname,d.deptid from employee e,department d,payroll p where e.empid=p.empid and p.netpay between 10000 and 20000 group by p.empid;

Problem # 8: Write a query to display employee names whose total deduction is more than 2000 for the month of APRIL.

à Write a query to display employee names whose total deduction is more than 2000 for the month of APRIL.

Problem # 9: Write a query to display employee id, employee name, department id, department name of all employees regardless of whether an employee is assigned a department or not.

à Write a query to display employee id, employee name, department id, department name of all employees regardless of whether an employee is assigned a department or not.

Problem # 10: Write a query to display Employee ID, Employee Name, Department ID, Years of Experience and Employee Category of the employees who have availed leaves more than 10 days.

Hint: Use the total_leaves column to check the leave condition for more than ten days.

àselect empid,empname,deptid,yrs_of_exp,employee_category from employee where empid in (select empid from employee_leave group by empid having sum(total_leaves)>10);

Payroll Schema – Complex Questions

Problem # 1: Write a query to display employee id, employee name and remaining casual leaves (alias- RemainingLeaves) for the employee with employee id "E002". Based on the total causal leaves available, subtract the number of causal leaves he has availed to get the remaining leaves.

Hint: CL – Causal leave.

à select (l.cl-sum(el.total_leaves)) as RemainingLeaves from employee_leave el,leave l where empid='E002' and leave_type='CL' and l.Leave_category=(select leave_category from employee where empid='E002');

Problem # 2: Write a query to display employee id, employee name and total number of leaves he can take (hint: with “EligibleLeave” as alias). This should be retrieved for all the employees. Sum all the EL, ML and EL leaves for the each employee’s category to get the total leaves.

Hint:

EMPLOYEE_INFO table has Employee’s leave Category. For example employee “E001” belong to “X” leave category.

LEAVE_INFO table has the Leave Category and number of CL, EL and ML available for them. For example, Employee E001 belongs to X category and he has 18 days of CL and 5 days of EL and 10 days of ML that he can avail.

So, E001's eligible leave would be 33 days which is sum of all his leaves. Similarly calculate for all employees.

à select e.empid,e.empname,(l.cl+l.el+l.ml) as EligibleLeaves from employee e,leave l where
l.leave_category=e.leave_category;

Problem # 3: Write a Query to display employee id, employee name, department id, department name, net pay of all employees who have drawn the highest salary (net pay) in the month of APRIL 2012.

Hint: For example if there are 10 employees where 3 employees have got a salary of 1000 which is the highest salary of the employee in the month of April all the three records needs to be displayed.

à select e.empid,e.empname,e.deptid,d.deptname,p.netpay from employee e,department d,payroll p where
e.empid=p.empid

and p.netpay=(select max(netpay) from payroll where month='apr' and year='2012') group by e.empid;

Problem # 4: Write a query to display employee id, employee name, basic pay and tax percentage for all employees. Use "TaxPercentage" as alias. Display the Tax percentage for all employees based on the following criteria: (If Basic Pay <= 4000 then tax percentage should be 10%, basic <= 5000 then 20%, basic<=6000 then 30% basic > 6000 then 40%).

```
DROP DATABASE sms_db;
```

```
CREATE DATABASE sms_db;
```

```
USE sms_db;
```

```
CREATE TABLE student
```

```
(  
  sid char(4) primary key,  
  sname varchar(20),  
  sdob date,  
  scity varchar(20),  
  squal varchar(20),  
  semail varchar(30),  
  sphone varchar(20)  
);
```

```
CREATE TABLE course
```

```
(  
  courseid char(4) primary key,  
  coursename varchar(40),  
  coursecategory varchar(20),  
  coursefees decimal(10,2),  
  courseduration int  
);
```

```
CREATE TABLE batch
```

```
(  
  batchid char(4) primary key,  
  bsdate datetime ,  
  bstrength int,  
  courseid char(4),  
  foreign key(courseid) references course(courseid)  
);
```

```
CREATE TABLE enrollment
```

```
(  
  batchid char(4),  
  sid char(4) ,  
  edate date,  
  primary key(batchid,sid),  
  foreign key(sid) references student(sid),  
  foreign key(batchid) references batch(batchid)  
);
```

dml

insert into student

values('s001','rajesh','1980-12-17','kolkata','graduate','rajesh@gmail.com','09830978900');

insert into student

values('s002','john','1949-1-7','hyderabad','postgraduate','john@yahoo.com','9833978933');

insert into student

values('s003','kunal','1967-2-3','pune','postgraduate','kunal@gmail.com','09830922900');

insert into student

values('s004','maya','1990-12-17','kolkata','graduate','maya.com','09830765900');

insert into student

values('s005','jadeja','1940-1-23','kolkata','postgraduate','jadeja@yahoo.com','09837865432');

insert into student

values('s006','suman','1995-6-17','kolkata','undergraduate','suman@gmail.com','0983097890');

insert into student

values('s007','soha','1990-7-17','mumbai','undergraduate',null,null);

insert into student

values('s008','thapa','1980-8-17','assam','graduate','thapa@gmail.com','19830978900');

insert into student

values('s009','hira','1954-9-17','mumbai','postgraduate','hira@gmail.com','09234097890');

insert into student

values('s010','akash','1977-1-27','kolkata','postgraduate','akash@gmail.com',null);

insert into student

values('s011','amir','1992-1-1','delhi','undergraduate','amir@gmail.com','09831118900');

insert into student

values('s012','ramesh','1980-12-17','kolkata','graduate','ramesh@yahoo.com','09830918900');

insert into student

values('s013','suresh','1980-3-22','kolkata','graduate','suresh@gmail.com','09830978912');

insert into student

values('s014','amir','1945-1-13','delhi','postgraduate','amir123@rediffmail.com','29830978900');

insert into student

values('s015','esha','1981-10-30','mumbai','graduate','esha@gmail.com','09831378900');

insert into student

values('s016','gopichand','1966-5-7','assam','postgraduate','gopi@gmail.com','09831918100');

insert into student

values('s017','sonali','1995-11-11','mumbai','undergraduate','sonali@gmail.com','09855978900');

insert into student

values('s018','lisa','1983-1-31','delhi','graduate','lisa@gmail.com','09832978923');

insert into student

values('s019','smith','1980-12-17','pune','graduate','smith@yahoo.com','09831111900');

insert into student

values('s020','rajesh','1994-7-8','pune','graduate','rajesh@gmail.com','09830978900');

insert into course values('c001','sql server','compse',1000,40);

insert into course values('c002','compmat','civileng',3000,120);

```
insert into course values('c003','biomaths','biotech',4000,160);
insert into course values('c004','word','compsc',500,8);
insert into course values('c005','photo','compsc',800,8);
```

```
insert into batch values('b001','2013-02-01 09:30',10, 'c001');
insert into batch values('b002','2013-03-01 09:30',10, 'c002');
insert into batch values('b003','2013-01-01 09:30',10, 'c003');
insert into batch values('b004','2013-03-31 09:30',10, 'c003');
insert into batch values('b005','2013-04-04 09:30',10, 'c005');
insert into batch values('b006','2013-01-27 09:30',10, 'c002');
insert into batch values('b007','2012-11-30 09:30',10, 'c004');
insert into batch values('b008','2013-01-28 09:30',10, 'c002');
insert into batch values('b009','2013-02-16 09:30',10,'c001');
insert into batch values('b010','2012-12-12 09:30',10, 'c003');
```

```
insert into enrollment values('b001','s001','2013-01-01');
insert into enrollment values('b001','s002','2013-01-31');
insert into enrollment values('b001','s003','2013-01-11');
insert into enrollment values('b001','s004','2013-02-02');
insert into enrollment values('b001','s005','2013-01-01');
insert into enrollment values('b001','s006','2013-01-01');
insert into enrollment values('b001','s007','2013-01-01');
insert into enrollment values('b001','s008','2013-01-01');
insert into enrollment values('b001','s009','2013-01-01');
```

```
insert into enrollment values('b002','s010','2013-02-01');
insert into enrollment values('b002','s012','2013-02-27');
insert into enrollment values('b002','s014','2013-01-21');
insert into enrollment values('b002','s016','2013-01-12');
insert into enrollment values('b002','s017','2013-02-15');
```

```
insert into enrollment values('b003','s018','2013-12-11');
insert into enrollment values('b003','s019','2013-02-27');
insert into enrollment values('b003','s020','2013-01-21');
insert into enrollment values('b003','s013','2013-01-01');
insert into enrollment values('b003','s007','2013-12-15');
insert into enrollment values('b003','s008','2013-11-25');
```

```
insert into enrollment values('b004','s001','2013-02-11');
insert into enrollment values('b004','s003','2013-02-27');
insert into enrollment values('b004','s006','2013-01-21');
insert into enrollment values('b004','s009','2013-03-01');
```

```
insert into enrollment values('b005','s001','2013-02-11');
insert into enrollment values('b005','s003','2013-02-27');
```

insert into enrollment values('b005','s006','2013-03-21');
insert into enrollment values('b005','s009','2013-04-01');

insert into enrollment values('b006','s001','2013-01-11');
insert into enrollment values('b006','s003','2012-12-27');
insert into enrollment values('b006','s006','2013-01-11');
insert into enrollment values('b006','s009','2013-01-01');
insert into enrollment values('b006','s007','2013-01-13');
insert into enrollment values('b006','s002','2012-12-17');
insert into enrollment values('b006','s008','2013-01-21');
insert into enrollment values('b006','s005','2013-01-01');

insert into enrollment values('b007','s001','2012-11-11');
insert into enrollment values('b007','s002','2012-11-11');
insert into enrollment values('b007','s003','2012-11-21');
insert into enrollment values('b007','s004','2012-11-13');
insert into enrollment values('b007','s007','2012-10-13');
insert into enrollment values('b007','s010','2012-10-17');
insert into enrollment values('b007','s009','2012-12-01');

insert into enrollment values('b008','s011','2012-11-11');
insert into enrollment values('b008','s012','2012-11-11');
insert into enrollment values('b008','s013','2012-11-21');
insert into enrollment values('b008','s014','2012-11-13');
insert into enrollment values('b008','s017','2012-10-13');
insert into enrollment values('b008','s020','2012-10-17');
insert into enrollment values('b008','s019','2012-12-01');

insert into enrollment values('b009','s001','2012-11-11');
insert into enrollment values('b009','s012','2012-11-11');
insert into enrollment values('b009','s013','2012-11-21');
insert into enrollment values('b009','s004','2012-11-13');
insert into enrollment values('b009','s007','2012-10-13');
insert into enrollment values('b009','s010','2012-10-17');
insert into enrollment values('b009','s009','2012-12-01');

insert into enrollment values('b010','s011','2012-11-11');
insert into enrollment values('b010','s002','2012-11-11');
insert into enrollment values('b010','s003','2012-11-21');
insert into enrollment values('b010','s014','2012-11-13');
insert into enrollment values('b010','s017','2012-10-13');
insert into enrollment values('b010','s010','2012-10-17');
insert into enrollment values('b010','s009','2012-12-01');

Q&A

```
select * from batch;  
select * from course;  
select * from enrollment;  
select * from student;
```

1) Display all undergraduate student whose name starts with 'S' and is of length between 5 and 20

```
select sname from student where squal='undergraduate' and sname like 's%'  
and length(sname) between 5 and 20  
order by sname;
```

2) Display the student who are senior citizen (≥ 60).

```
select sname from student where datediff(current_date,date(sdob))/365 $\geq$ '60';
```

3) Display student who were born after 1st of June 1980.

```
select sname from student where sdob>'1980-06-01'
```

4) The student are suppose to only provide mobile numbers .All mobile numbers should start with zero followed by 10 digits. Display student name having invalid phone numbers.

```
select sname from student where sphone like '0%' or length(sphone)<11 or sphone is null  
order by sname;
```

5) All emails should have "@" anywhere after the first character and should end with ".com".
Display count of students having invalid email id.

```
select count(sname) from student where semail not like '%@%.com' or semail is null;
```

6) Display the name and email of student who have a Gmail account.

```
select sname,semail from student where semail like '%gmail%';
```

7) Display the above record but do not consider invalid email id.

```
select sname,semail from student where semail like '%@gmail%';
```

8) Display the qualification and the total number of students registered based on their qualifications.
(Alias use "totalStud" for total number of students)

select squal,count(sid) as totalstud from student group by squal

9) Display the full name of the month and the total number of students who are having their birthday in that month.

(Alias use "Month" and "Total")

select monthname(sdob),count(sid) from student group by month(sdob);

10) Display the student name that was born in a leap year ordering by student name and year of birth

select sname from student where year(sdob)%4=0 or year(sdob)%400=0 and year(sdob)%100!=0

11) Display student whose city is Kolkata as "HomeStudent" and others as "DistanceStudent" under a column "Remarks".

Also display the name and city of the student

select sname,scity,case
when scity='kolkata' then 'Home Student'
when scity!='kolkata' then 'DistanceStudent' end as 'Remarks'
from student;

12) Display batchid, coursename, batch start date, batch end date for all batches.

#(batch end date=batch start date +course duration).

select b.batchid,c.coursename,b.bsdate,date_add(b.bsdate,interval c.courseduration day) as bachendddate from
batch b ,course c where b.courseid=c.courseid;

13) Display all batchid having a difference of 10 hours and less between its starting and ending date.

select batch.batchid from batch join course on
batch.courseid=course.courseid
group by batch.batchid having date_diff(batch.bsdate,date_add(b.bsdate,interval c.courseduration day))<=10;

14) Display all batches having similar start date and strength.

select batch.batchid from batch, batch a,batch b where
a.bsdate=b.bsdate and a.bstrength=b.bstrength
and a.batchid!=b.batchid
group by batch.batchid;

15) Display student who enrolled for the batch after its start date.


```
select distinct sname from student join enrollment on
student.sid=enrollment.sid join batch on
enrollment.batchid=batch.batchid where batch.bsdate<enrollment.edate;
```

16) Display the studentid, studentname , totalfees for all student.

```
select student.sid,student.sname,sum(course.coursefees) as totalfees from student join enrollment on
student.sid=enrollment.sid join batch on
enrollment.batchid=batch.batchid join course on
batch.courseid=course.courseid
group by student.sid;
```

17) Display courses which are not being taught currently along with courses which are being taught.
#Also display the batchid for the courses currently running and null for non executing courses.

```
select b.batchid,c.coursename,case when
date_add(b.bsdate,interval c.courseduration day)>=current_date then b.batchid
else 'null' end as cou
from batch b join course c on b.courseid=c.courseid
group by c.coursename;
```

18) Display count of students having no contact information. (Either email or phone).

```
select count(student.sid) from student where semail is null or sphone is null;
```

19)) Display coursename having above average fees.

```
select course.coursename from course where course.coursefees>(select avg(coursefees) from course);
```

20)) Display coursename where fees are less than the average fees of its category

```
select coursename,coursefees from course c1 where c1.coursefees<(
select avg(c2.coursefees)from course c2 where c1.coursecategory=c2.coursecategory);
```

21)Display the coursename having the highest enrollment

```

select course.coursename from course join batch on
course.courseid=batch.courseid join enrollment on
batch.batchid=enrollment.batchid
group by course.courseid having count(enrollment.sid)>=all(select count(enrollment.sid) from course join
batch on
course.courseid=batch.courseid join enrollment on
batch.batchid=enrollment.batchid
group by course.courseid);

```

22) Display student name having duplicate email ids.

```

select distinct a.sname from student a,student b where
a.sid!=b.sid and a.semail = b.semail;

```

#23) Display student name having similar name but different email ids.

```

select distinct a.sname from student a,student b where
a.sname=b.sname and a.semail!=b.semail;

```

24) Display the student name, date of birth and their zodiac sign. Use Zodiac as alias

- a. Aries ??Mar 21-Apr 19
- b. Taurus?Apr 20 –May 20
- c. Gemini?May 21-Jun 20
- d. Cancer ??Jun21- Jul22
- e. Leo ??Jul 23- Aug 22
- f. Virgo ??Aug 23-Sept 22
- g. Libra ??Sept 23-Oct 22
- h. Scorpio ??Oct 23- Nov 21
- i. Sagittarius ??Nov 22-Dec 21
- j. Capricorn ??Dec 22- Jan 19
- k. Aquarius ?Jan 20 – Feb 18
- l. Pisces ?Feb 19- Mar 20.

```

select sname ,sdob,case
when date_format(sdob,'%m%d') between '03-21' and '04-19' then 'aries'

when date_format(sdob,'%m%d') between '04-20' and '05-20' then 'Taurus'

when date_format(sdob,'%m%d') between '05-21' and '06-20' then 'gemini'

when date_format(sdob,'%m%d') between '06-21' and '07-22' then 'Cancer'

```

```

when date_format(sdob,'%m%d') between '07-23' and '08-22' then 'Leo'

when date_format(sdob,'%m%d') between '08-23' and '09-22' then 'Vigo'

when date_format(sdob,'%m%d') between '09-23' and '10-22' then 'Libra'

when date_format(sdob,'%m%d') between '10-23' and '11-21' then 'Scorpio'


when date_format(sdob,'%m%d') between '11-22' and '12-21' then 'Sagittarius'

when date_format(sdob,'%m%d') between '12-22' and '01-19' then 'Capricorn '

when date_format(sdob,'%m%d') between '01-20' and '02-18' then 'Aquarius '

when date_format(sdob,'%m%d') between '02-19' and '03-20' then 'Pisces'


end as sign from student;

```

25) Display the course name fetching the 2nd highest revenue.

```

SELECT MAX(ASD), SDF FROM
(SELECT COURSE.COURSENAME AS SDF , COUNT(STUDENT.SID) AS CON ,
COURSE.COURSEFEES AS FEE,
COUNT(STUDENT.SID)* COURSE.COURSEFEES AS ASD
FROM COURSE
JOIN BATCH ON COURSE.COURSEID=BATCH.COURSEID
JOIN ENROLLMENT ON BATCH.BATCHID = ENROLLMENT.BATCHID
JOIN STUDENT ON ENROLLMENT.SID=STUDENT.SID
GROUP BY COURSE.COURSENAME, COURSE.COURSEFEES)A

WHERE ASD <> (SELECT MAX(ASD) FROM (SELECT COUNT(STUDENT.SID)*
COURSE.COURSEFEES AS ASD
FROM COURSE
JOIN BATCH ON COURSE.COURSEID=BATCH.COURSEID
JOIN ENROLLMENT ON BATCH.BATCHID = ENROLLMENT.BATCHID
JOIN STUDENT ON ENROLLMENT.SID=STUDENT.SID
GROUP BY COURSE.COURSENAME, COURSE.COURSEFEES) B)

```

26) Generate report which displays the batch number and the number of seats vacant. [Use Alias "Vacant"].

```
select batch.batchid,(batch.bstrength-count(enrollment.batchid)) as vacant from batch join enrollment on
batch.batchid=enrollment.batchid
group by batch.batchid;
```

27) Which among the following have the highest enrollment? (Graduate,Undergraduate or postgraduate).

Write query to display "Qualification" , "HighestEnrollmet" (Use Alias").

```
select distinct squal from student group by squal
having count(squal)>=all(select count(squal) from student group by squal)
```

28) Display student name, age[Alias] , coursename, batchid, batch_start_date, batch_end_date ,

#and enrollment date in the

#format 1st of Jan , 2012 .

```
select student.sname,datediff(current_date,student.sdob) as age,course.coursename,batch.batchid,batch.bsdate,
date_add(batch.bsdate,interval course.courseduration day) as enddate,
concat(cast(day(enrollment.edate)as char),'st of ', (cast(substr(monthname(enrollment.edate),1,3)as char)),' ',
cast(year(enrollment.edate) as char))as dates
from student join enrollment on
student.sid=enrollment.sid join batch on
enrollment.batchid=batch.batchid join course on
batch.courseid=course.courseid;
```

#29) Display report in the following format. Consider Sat & Sun as holiday.

#Week Enrollment

#Weekday 10

#Hoilday 05

```
select case when dayname(edate)='Monday' then 'Weekday'
```

```
when dayname(edate)='Tuesday' then 'Weekday'
```

```
when dayname(edate)='Wednesday' then 'Weekday'
```

```
when dayname(edate)='Thursday' then 'Weekday'
```

```
when dayname(edate)='Friday' then 'Weekday'
```

```
else 'holiday'
```

```
end as 'week', count(sid) as 'enrollment'
```

```
from `enrollment`
```

```
group by week desc;
```

```
create table CUSTOMER_MASTER
(
    CUSTOMER_ID Varchar(10),
    CUSTOMER_NAME Varchar(30) NOT NULL,
    CONTACT_NO BIGINT(20),
    CONTACT_ADD Varchar(100),
    DATE_OF_REGISTRATION Date NOT NULL,
    AGE Varchar(15) NOT NULL,
    Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID)
);
```

Create table MOVIES_MASTER

```
(
    MOVIE_ID Varchar(10),
    MOVIE_NAME Varchar(80) NOT NULL,
    RELEASE_DATE Varchar(30) NOT NULL,
    LANGUAGE Varchar(30),
    RATING int(2),
    DURATION_In_Minutes VARCHAR(10) NOT NULL,
    MOVIE_TYPE Varchar(100),
    MOVIE_CATEGORY VARCHAR(40) NOT NULL,
    DIRECTOR VARCHAR(60) NOT NULL,
    LEAD_Actor_name1 Varchar(50) NOT NULL,
    LEAD_Actor_name2 VARCHAR(60) NOT NULL,
    RENTAL_COST BIGINT(10),
    Constraint MT_cts4 PRIMARY KEY(MOVIE_ID)
);
```

Create table CUSTOMER_ISSUE_DETAILS

```
(  
    ISSUE_ID Varchar(10) NOT NULL,  
    CUSTOMER_ID Varchar(10) NOT NULL,  
    MOVIE_ID VARCHAR(10),  
    ISSUE_DATE Date NOT NULL,  
    RETURN_DATE Date NOT NULL,  
    ACTUAL_DATE_RETURN Date NOT NULL,  
    Constraint MT_cts5 PRIMARY KEY(ISSUE_ID),  
    Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References  
CUSTOMER_MASTER(CUSTOMER_ID),  
    Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID) References MOVIES_MASTER(MOVIE_ID)  
  
);
```

Create table LIBRARY_CARD_MASTER

```
(  
    CARD_ID Varchar(10),  
    DESCRIPTION Varchar(30) NOT NULL,  
    AMOUNT      BIGINT(50),  
    NUMBER_OF_YEARS bigint(10) NOT NULL,  
    Constraint MT_cts2 PRIMARY KEY(CARD_ID)  
  
);
```

Create table CUSTOMER_CARD_DETAILS

```
(
    CUSTOMER_ID Varchar(10),
    CARD_ID VARCHAR(10),
    ISSUE_DATE DATE NOT NULL,
    Constraint MT_cts3 PRIMARY KEY(CUSTOMER_ID),
    Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID) References
CUSTOMER_MASTER(CUSTOMER_ID),
    Constraint MT_CTS42 FOREIGN KEY(CARD_ID) References LIBRARY_CARD_MASTER(CARD_ID)
);
```

insert into customer_master values

```
('C00001','    NITIN  ','9830354218','A/122, KALKAJI','    2012-10-15',    22),
```

```
('C00002','    AGNESH      ','8923156781','9/1,ANDHERI EAST','    2012-11-01',    35),
```

```
('C00003','    T RAMACHANDRAN',' 9831289761',' 9/1,NANDABAKKAM',' 2012-11-02',    25),
```


('C00004', 'RAJIB MITRA', '9830356781', 'H/56, BLOCK1,JADAVPUR', '2012-11-21', 45),

('C00005', 'SHIV PRASAD', NULL, '2/2 PHASE II JAWAHAR NAGAR', '2012-12-25', 30),

('C00006', 'AJAY GHOSH', '8763478901', 'N/2,GANDHI COLONY DUM DUM', '2012-12-30', 20),

('C00007', 'GEETHA REDDY', '8976167890', 'AH 1/1 T NAGAR', '2012-12-31', 30),

('C00008', 'RIA NATRAJAN', '9856723190', 'A/B GANDHI COLONY', '2013-01-01', 45),

('C00009', 'RAJAN PILLAI', NULL, 'A 1/66 KODAMBAKKAM', '2013-01-02', 40),

('C00010', 'RAGHAV SINGH', '9675167890', 'A/6 NEHRU JAWAHAR NAGAR', '2013-03-02', 50),

('C00011', 'RAJ SEKHANRAN', '8423178906', 'A/1 MAYUR KUNJ', '2013-03-15', 25);

insert into movies_master values

('M00001',' DIE HARD',' 1998 ','ENGLISH' ',4' ',120' ',UNIVERSAL',' ACTION',
'JOHN MCTIERNAN' ',BRUCE WILLIS','BONNIE BEDELIA' ',100),

('M00002',' THE DARK KNIGHT',' 2008',' ENGLISH', 5 ',90 ',PARENTAL
GUIDENCE',' ACTION','
CHRISTOPHER NOLAN' ',
'CHRISTIAN BALE','HEATH LEDGER', 100),

('M00003',' THE MATRIX', 1999,' ENGLISH', 4, 120,' UNIVERSAL',' ACTION','
ANDY LARRY',' KEANU REEVES ',
'CARRIE-ANEE MOSS', 100),

('M00004',' INCEPTION', 2010 ',ENGLISH' ',5' ',120,' PARENTAL GUIDENCE
,',ACTION',
'CHRISTOPHER NOLAN', 'LEONARDO DICAPRIO','JOSEPH GORDAN' ',100),

('M00005','OFFICE SPACE', 1999 , 'ENGLISH', 4, 95 , 'UNIVERSAL ' , 'COMEDY',
,
'MIKE JUDGE', ' RON LIVINGSTON', ' JENNIFER ANISTON ', 100),

('M00006 ', 'YOUNG FRANKENSTEIN', ' 1974', ' ENGLISH', 4 , 130, ' UNIVERSAL', ' COMEDY',
'MEL BROOKS', ' GENE WILDER', ' TERI GARR', 100),

('M00007', ' SHAUN OF THE DEAD', 2004 , 'ENGLISH', 4 , 95, ' UNIVERSAL', ' COMEDY',
'EDGAR WRIGHT', ' SIMON PEGG ', 'KATE ASHFIELD' , 100),

('M00008', ' CASABLANCA', 1942, ' ENGLISH', 3, 120 , 'UNIVERSAL', ' ROMANCE', ' MICHAEL
CURTIZ', ' HUMPREY BOGART', ' INGRID BERGMAN', 1000),

('M00009 ', 'THE NOTEBOOK', 2004 , 'ENGLISH', 3 , 120 , 'PARENTAL GUIDENCE', ' ROMANCE ',
'NICK CASSAVETES', ' RYAN GOSLING', ' RACHEL MCADAMS', 100),

('M00010 ', 'GONE WITH THE WIND ', 1939 , 'ENGLISH', 3 , 120, ' PARENTAL GUIDENCE ', 'ROMANCE',
'VICTOR FLEMMING', ' CLARK GABLE', 'VIVIEN LEIGH ', 100),

('M00011 ', 'TITANIC', 1997 , 'ENGLISH ', 3 , 120 , 'PARENTAL GUIDENCE', ' ROMANCE'
, 'JAMES CAMERON', ' LEONARDO DICAPRIO ', 'KATE WINSLET', 100);

insert into customer_issue_details values

('I00001','C00001','M00001','2012-10-15','2012-10-17','2012-10-17');

insert into customer_issue_details values('I00002', 'C00002','M00002','2012-11-02','2012-11-04','2012-11-05');

insert into customer_issue_details values

('I00003','C00002','M00002','2012-12-02','2012-12-04','2012-12-03');

insert into customer_issue_details values('I00004','C00003','M00003','2012-11-02','2012-11-04','2012-11-10');

insert into customer_issue_details values('I00005','C00003','M00004','2012-11-10',
'2012-11-12','2012-11-12');

insert into customer_issue_details values('I00006','C00003','M00005',
'2012-11-12','2012-11-14','2012-11-14');

insert into customer_issue_details values('I00007','C00004','M00006', '2012-11-21',
'2012-11-23','2012-11-24');

insert into customer_issue_details values('I00008','C00010','M00008','2013-03-02',
'2013-03-04','2013-03-05');

insert into customer_issue_details values('I00009','C00011','M00010', '2013-03-16','2013-03-18',
'2013-03-18');

insert into customer_issue_details values('I00010','C00004', 'M00007',
'2012-11-25','2012-11-27','2012-11-27');

```
insert into customer_issue_details values('I00011','C00004'      , 'M00007','2012-11-28',  
'2012-11-30','2012-11-30');
```

```
insert into customer_issue_details values('I00012','C00001','M00001','2013-11-28','2013-11-30',  
'2013-11-30');
```

```
insert into customer_issue_details values('I00013','C00003','M00001','2012-12-03','2012-12-05',  
'2012-12-05');
```

```
insert into customer_issue_details values('I00014','C00003','M00010      ', '2013-01-02','2013-01-04'  
, '2013-01-05');
```

```
insert into customer_issue_details values('I00015','C00003','M00011      ', '2013-02-03','2013-02-05',  
'2013-02-06');
```

```
insert into customer_issue_details values('I00016','C00003','M00011      ', '2013-03-05','2013-03-07'  
, '2013-03-07');
```

```
insert into customer_issue_details values('I00017','C00003','M00008','2013-04-15','2013-04-17'  
, '2013-04-17');
```

```
insert into customer_issue_details values('I00018','C00002','M00010      ', '2015-01-15','2015-01-17'  
, '2015-01-17');
```

```
insert into customer_issue_details values('I00019','C00004','M00001','2012-11-15      ', '2012-11-  
17','2012-11-17');
```

```
insert into library_card_master values
```

('CRD001', 'SILVER CARD' ,1000, 1),

('CRD002', 'GOLD CARD' ,2000 ,2),

('CRD003', 'PLATINUM CARD' ,3000, 3),

('CRD004', 'DIAMOND CARD', 4000 ,5);

Insert into CUSTOMER_CARD_DETAILS Values('C00001','CRD001','2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('C00002','CRD002','2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('C00003','CRD002','2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('C00004','CRD003','2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('C00005','CRD003','2012-05-13');

Video Management database queries:

1. Please follow instructions given below.

Write a query to display movie names and number of times that movie is issued to customers. In case movies are never issued to customers display number of times as 0.

Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order).

The Alias name for the number of movies issued is ISSUE_COUNT.

11 rows

```
select mm.movie_name, count(cid.issue_id) as ISSUE_COUNT
from movies_master mm left outer join customer_issue_details
cid on mm.movie_id=cid.movie_id group by mm.movie_name
order by ISSUE_COUNT desc,mm.movie_name asc;
```

MOVIE_NAME	ISSUE_COUNT
DIE HARD	4
GONE WITH THE WIND	3
CASABLANCA	2
SHAUN OF THE DEAD	2
THE DARK KNIGHT	2
TITANIC	2

INCEPTION	1
OFFICE SPACE	1
THE MATRIX	1
YOUNG FRANKENSTEIN	1
THE NOTEBOOK	0

2. Please follow instructions given below.

Write a query to display id,name,age,contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

4 rows

select customer_id,customer_name,age,

ifnull(concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',substring(contact_no,7,4)),'N/A')

as CONTACT_ISD from customer_master where age>25 and year(date_of_registration)=2012

order by age,customer_name;

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
C00007	GEETHA REDDY	30	+91-897-616-7890
C00005	SHIV PRASAD	30	N/A

C00002	AGNESH	35	+91-892-315-6781
C00004	RAJIB MITRA	45	+91-983-035-6781

3.Please follow instructions given below.

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

Hint: Use NO_OF_MOVIES as alias name for number of movies.

3 rows

Ans:

select movie_category,count(movie_id) as NO_OF_MOVIES from movies_master group by movie_category

order by NO_OF_MOVIES desc,movie_category asc;

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4.Please follow instructions given below.

Write a query to display the number of customers having card with description "Gold card".
Hint: Use CUSTOMER_COUNT as alias name for number of customers

1 row

select count(ccd.customer_id) as CUSTOMER_COUNT from customer_card_details ccd join

library_card_master lcd on ccd.card_id=lcd.card_id where lcd.description='Gold Card';

CUSTOMER_COUNT
2

4.Please follow instructions given below.

Write a query to display the customer id, customer name, year of registration,library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

Use REGISTERED_YEAR as alias name for year of registration.

5 rows

```
select cm.customer_id,cm.customer_name,year(cm.date_of_registration) as
REGISTERED_YEAR,ccd.card_id,ccd.issue_date

from customer_master cm join customer_card_details ccd on cm.customer_id=ccd.customer_id

order by cm.customer_name desc;
```

CUSTOMER_ID	CUSTOMER_NAME	REGISTERED_YEAR	CARD_ID	ISSUE_DATE
C00003	T RAMACHANDRAN	2012	CRD002	2012-11-02
C00005	SHIV PRASAD	2012	CRD003	2012-12-26
C00004	RAJIB MITRA	2012	CRD003	2012-11-21
C00001	NITIN	2012	CRD001	2012-10-15
C00002	AGNESH	2012	CRD002	2012-12-01

5. Please follow instructions given below.

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

Display the records sorted in ascending order based on customer name.

2 rows

```
select cid.issue_id,cid.customer_id,cm.customer_name from customer_issue_details cid join
customer_master cm on cid.customer_id=cm.customer_id where cm.customer_name like 'R%'
and cid.actual_date_return>cid.return_date order by cm.customer_name;
```

ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME
I00008	C00010	RAGHAV SINGH
I00007	C00004	RAJIB MITRA

6. Please follow instructions given below.

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount.
Example Assume 500 is the amount then dollar value will be \$10.

Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar.

Display the records in ascending order based on customer name.

```
SELECT ccd.customer_id, customer_name, ccd.card_id, description,concat('$',round(amount/52.42,0))
AMOUNT_DOLLAR FROM customer_master cm INNER JOIN customer_card_details ccd ON
cm.customer_id=ccd.customer_id INNER JOIN library_card_master lcm ON ccd.card_id=lcm.card_id
```

**INNER JOIN customer_issue_details cid ON cid.customer_id = cm.customer_id WHERE
cm.date_of_registration=cid.issue_date order by customer_name;**

CUSTOMER_ID	CUSTOMER_NAME	CARD_ID	DESCRIPTION	AMOUNT_DOLLAR
C00001	NITIN	CRD001	SILVER CARD	\$19
C00004	RAJIB MITRA	CRD003	PLATINUM CARD	\$57
C00003	T RAMACHANDRAN	CRD002	GOLD CARD	\$38

7.Please follow instructions given below.

Write a query to display the customer id, customer name,contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use CUSTOMER_NAME as alias name for customer name.
Display the details sorted in ascending order based on customer name.

**SELECT customer_id , upper(customer_name) CUSTOMER_NAME,contact_no,contact_address FROM
customer_master WHERE customer_id NOT IN (select customer_id from customer_card_details) AND
customer_id IN (SELECT customer_id from customer_issue_details) and contact_address like
'%Nagar' order by customer_name ;**

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADDRESS
C00010	RAGHAV SINGH	9675167890	A/6 NEHRU JAWAHAR NAGAR

8.Please follow instructions given below.

Write a query to display the movie id, movie name,release year,director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.

```
select movie_id,movie_name , release_year ,director_name from movies_master where
lead_actor_name1 in(select lead_actor_name1 from(select
lead_actor_name1,count(movie_id) ct from movies_master group by lead_actor_name1)t where
t.ct>=all(select count(movie_id) from movies_master
group by lead_actor_name1))order by movie_name;
```

MOVIE_ID	MOVIE_NAME	RELEASE_YEAR	DIRECTOR_NAME
M00004	INCEPTION	2010	CHRISTOPHER NOLAN
M00011	TITANIC	1997	JAMES CAMERON

9.Please follow instructions given below.

Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.

Hint: Use MOVIE_COUNT as alias name for number of movies issued.

11 rows

```
select cm.customer_name,count(cid.movie_id) as MOVIE_COUNT from customer_master cm left join
customer_issue_details cid on cm.customer_id=cid.customer_id group by cm.customer_name order by
cm.customer_name;
```

CUSTOMER_NAME	MOVIE_COUNT
---------------	-------------

AGNESH	3
AJAY GHOSH	0
GEETHA REDDY	0
NITIN	2
RAGHAV SINGH	1
RAJ SEKHANRAN	1
RAJAN PILLAI	0
RAJIB MITRA	4
RIA NATRAJAN	0
SHIV PRASAD	0
T RAMACHANDRAN	8

10. Please follow instructions given below.

Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.

For Example Assume the issue id is I00005 then the serial number is 05

Hint: Alias name for serial number is 'SERIAL_NO'

19 rows

```
select substring(cid.issue_id,5,2) as  
SERIAL_NO,cid.issue_id,cid.customer_id,cm.customer_name,mm.movie_id,mm.movie_name  
  
from customer_issue_details cid join customer_master cm on cm.customer_id=cid.customer_id  
  
join movies_master mm on cid.movie_id=mm.movie_id group by  
SERIAL_NO,cid.customer_id,mm.movie_id  
  
order by SERIAL_NO;
```

SERIAL_NO	ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME	MOVIE_ID	MOVIE_NAME
01	I00001	C00001	NITIN	M00001	DIE HARD
02	I00002	C00002	AGNESH	M00002	THE DARK KNIGHT
03	I00003	C00002	AGNESH	M00002	THE DARK KNIGHT
04	I00004	C00003	T RAMACHANDRAN	M00003	THE MATRIX
05	I00005	C00003	T RAMACHANDRAN	M00004	INCEPTION
06	I00006	C00003	T RAMACHANDRAN	M00005	OFFICE SPACE
07	I00007	C00004	RAJIB MITRA	M00006	YOUNG FRANKENSTEIN
08	I00008	C00010	RAGHAV SINGH	M00008	CASABLANCA

09	I00009	C00011	RAJ SEKHANRAN	M00010	GONE WITH THE WIND
10	I00010	C00004	RAJIB MITRA	M00007	SHAUN OF THE DEAD
11	I00011	C00004	RAJIB MITRA	M00007	SHAUN OF THE DEAD
12	I00012	C00001	NITIN	M00001	DIE HARD
13	I00013	C00003	T RAMACHANDRAN	M00001	DIE HARD
14	I00014	C00003	T RAMACHANDRAN	M00010	GONE WITH THE WIND
15	I00015	C00003	T RAMACHANDRAN	M00011	TITANIC
16	I00016	C00003	T RAMACHANDRAN	M00011	TITANIC
17	I00017	C00003	T RAMACHANDRAN	M00008	CASABLANCA
18	I00018	C00002	AGNESH	M00010	GONE WITH THE WIND
19	I00019	C00004	RAJIB MITRA	M00001	DIE HARD





11. Please follow instructions given below.

Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

7 rows

```
select cid.issue_id, cid.issue_date, cid.customer_id, cm.customer_name, cm.contact_no
from customer_issue_details cid join customer_master cm on cid.customer_id = cm.customer_id
where year(issue_date) = 2013 group by issue_id, issue_date, customer_id order by
issue_date desc;
```

ISSUE_ID	ISSUE_DATE	CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO
I00012	2013-11-28	C00001	NITIN	9830354218
I00017	2013-04-15	C00003	T RAMACHANDRAN	9831289761
I00009	2013-03-16	C00011	RAJ SEKHANRAN	8423178906
I00016	2013-03-05	C00003	T RAMACHANDRAN	9831289761
I00008	2013-03-02	C00010	RAGHAV SINGH	9675167890
I00015	2013-02-03	C00003	T RAMACHANDRAN	9831289761
I00014	2013-01-02	C00003	T RAMACHANDRAN	9831289761

12.Please follow instructions given below.

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format.LEAD_ACTOR_ONE space ambersant space LEAD_ACTOR_TWO.

Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria"Hint:Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.

1 row

```
select movie_id,movie_name,concat(lead_actor_name1,' & ',lead_actor_name2) as ACTORS
from movies_master where movie_id
not in (select movie_id from customer_issue_details) order by
movie_name;
```

MOVIE_I D	MOVIE_NAME	ACTORS
M00009	THE NOTEBOOK	RYAN GOSLING & RACHEL MCADAMS

13.Please follow instructions given below.

Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

2 rows

```
SELECT upper(director_name) DIRECTOR_NAME,movie_name,lead_actor_name1 FROM
movies_master WHERE director_name in (SELECT director_name FROM movies_master GROUP BY
director_name HAVING count(movie_id)>1) order by director_name, movie_name desc;
```

DIRECTOR_NAME	MOVIE_NAME	LEAD_ACTOR_NAME1
CHRISTOPHER NOLAN	THE DARK KNIGHT	CHRISTIAN BALE
CHRISTOPHER NOLAN	INCEPTION	LEONARDO DICAPRIO

14.Please follow instructions given below.

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
 Hint:Use NO_OF_CUSTOMERS as alias name for number of customers.

1 row

**select count(customer_id) as NO_OF_CUSTOMERS from customer_master where
year(date_of_registration)
=2012 and contact_no != 'NULL'**

NO_OF_CUSTOMERS
6

15.Please follow instructions given below.

Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.

11 rows

**select cm.customer_name,ifnull(cm.contact_no,cm.contact_add) as
CONTACT_DETAILS,lcd.card_id,lcd.description from customer_master cm
left join customer_card_details ccd on cm.customer_id=ccd.customer_id**

left join library_card_master lcd on ccd.card_id=lcd.card_id group by
customer_name,description,CONTACT_DETAILS

order by customer_name;

CUSTOMER_NAME	CONTACT_DETAILS	CARD_ID	DESCRIPTION
AGNESH	8923156781	CRD002	GOLD CARD
AJAY GHOSH	8763478901	NULL	NULL
GEETHA REDDY	8976167890	NULL	NULL
NITIN	9830354218	CRD001	SILVER CARD
RAGHAV SINGH	9675167890	NULL	NULL
RAJ SEKHANRAN	8423178906	NULL	NULL
RAJAN PILLAI	A 1/66 KODAMBAKKAM	NULL	NULL
RAJIB MITRA	9830356781	CRD003	PLATINUM CARD
RIA NATRAJAN	9856723190	NULL	NULL
SHIV PRASAD	2/2 PHASE II, JAWAHAR NAGAR	CRD003	PLATINUM CARD
T RAMACHANDRAN	9831289761	CRD002	GOLD CARD

16. Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in descending order. For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of John. Hint: Use NO_OF_TIMES as alias name for number of times

4 rows

```
select cm.customer_id,cm.customer_name,count(cid.movie_id) as NO_OF_TIMES from  
customer_master
```

```
cm join customer_issue_details cid on cm.customer_id=cid.customer_id group by  
customer_id,movie_id having
```

```
count(movie_id)>1 order by customer_name desc;
```

CUSTOMER_ID	CUSTOMER_NAME	NO_OF_TIMES
C00003	T RAMACHANDRAN	2
C00004	RAJIB MITRA	2
C00001	NITIN	2
C00002	AGNESH	2

17. Please follow instructions given below.

Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.

Hint: Use NO_OF_MOVIES as alias name for number of movies column.

Hint: Use CONTACT_ISD as alias name for contact number.

Display the records sorted in ascending order based on customer name and then by movie category.

5 rows

```
select cid.customer_id,cm.customer_name,  
concat('+91-',substring(cm.contact_no,1,3),'-',substring(cm.contact_no,4,3),'-',  
substring(cm.contact_no,7,4)) as CONTACT_ISD,  
mm.movie_category,count(mm.movie_category) as NO_OF_MOVIES from customer_master  
cm join customer_issue_details cid  
on cm.customer_id=cid.customer_id join movies_master mm on cid.movie_id=mm.movie_id  
group by mm.movie_category,cm.customer_name having count(movie_category)>1  
order by cm.customer_name,mm.movie_category;
```

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_ISD	MOVIE_CATEGORY	NO_OF_MOVIES
C00002	AGNESH	+91-892-315-6781	ACTION	2
C00001	NITIN	+91-983-035-4218	ACTION	2
C00004	RAJIB MITRA	+91-983-035-6781	COMEDY	3
C00003	T RAMACHANDRAN	+91-983-128-9761	ACTION	3
C00003	T RAMACHANDRAN	+91-983-128-9761	ROMANCE	4

18. Please follow instructions given below.

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

3 rows

```
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) >= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by cm.customer_name)
union all
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) <= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by cm.customer_name)
```

CUSTOMER_ID	CUSTOMER_NAME
C00003	T RAMACHANDRAN

C00010	RAGHAV SINGH
C00011	RAJ SEKHANRAN

19.Please follow instructions given below.

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO_OF_TIMES as alias name

Display the records in ascending order based on customer name.

1 row

```
select cm.customer_id,cm.customer_name,count(mm.movie_id) as NO_OF_TIMES from
customer_master cm
```

```
join customer_issue_details cid on cm.customer_id=cid.customer_id join
```

```
movies_master mm on cid.movie_id=mm.movie_id where mm.movie_category='comedy' group by
customer_id
```

```
order by customer_name>1;
```

CUSTOMER_ID	CUSTOMER_NAME	NO_OF_TIMES
C00004	RAJIB MITRA	3

20.Please follow instructions given below.

Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST. Display the records sorted in ascending order based on customer id

6 rows

```
select cid.customer_id,sum(mm.rental_cost) as TOTAL_COST from customer_issue_details cid  
join movies_master mm  
on cid.movie_id=mm.movie_id group by customer_id order by customer_id;
```

	customer_id	TOTAL_COST
▶	C00001	200
	C00002	300
	C00003	1700
	C00004	400
	C00010	1000
	C00011	100

MOVIE

CREATE DATABASE video;USE video;

Create table CUSTOMER_MASTER

(CUSTOMER_ID Varchar(10),CUSTOMER_NAME Varchar(30) NOT NULL,CONTACT_NO BIGINT(10),CONTACT_ADD Varchar(20),DATE_OF_REGISTRATION Date NOT NULL,AGE Varchar(15)NOT NULL,Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID));

Create table LIBRARY_CARD_MASTER

(CARD_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT BIGINT(50),NUMBER_OF_YEARS bigint(10) NOT NULL,Constraint MT_cts2 PRIMARY KEY(CARD_ID));

Create table MOVIES_MASTER

(MOVIE_ID Varchar(10), MOVIE_NAME Varchar(50) NOT NULL,RELEASE_DATE Varchar(30) NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION VARCHAR(10) NOT NULL, MOVIE_TYPE Varchar(3),MOVIE_CATEGORY VARCHAR(20) NOT NULL,DIRECTOR VARCHAR(20) NOT NULL, LEAD_ROLE_1 Varchar(3) NOT NULL,LEAD_ROLE_2 VARCHAR(4) NOT NULL,RENT_COST BIGINT(10),Constraint MT_cts4 PRIMARY KEY(MOVIE_ID));

Create table CUSTOMER_CARD_DETAILS

(CUSTOMER_ID Varchar(10),CARD_ID VARCHAR(10),ISSUE_DATE DATE NOT NULL,Constraint MT_cts3 PRIMARY KEY(CUSTOMER_ID),Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID) References CUSTOMER_MASTER(CUSTOMER_ID),Constraint MT_CTS42 FOREIGN KEY(CARD_ID) References LIBRARY_CARD_MASTER(CARD_ID));

Create table CUSTOMER_ISSUE_DETAILS

(ISSUE_ID Varchar(10) NOT NULL,CUSTOMER_ID Varchar(10) NOT NULL,MOVIE_ID VARCHAR(10), ISSUE_DATE Date NOT NULL,RETURN_DATE Date NOT NULL,

ACTUAL_DATE_RETURN Date NOT NULL,Constraint MT_cts5 PRIMARY KEY(ISSUE_ID),Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References CUSTOMER_MASTER(CUSTOMER_ID), Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID) References MOVIES_MASTER(MOVIE_ID));

Insert into CUSTOMER_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS007', 'NAME7',
3210987654,'ADD7', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS008', 'NAME8',
2109876543,'ADD8', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS009', 'NAME9',
NULL,'ADD9', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS010', 'NAM10',
9934567890,'ADD10', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS011', 'NAM11',
9875678910,'ADD11', '2013-02-12', '21');

Insert into LIBRARY_CARD_MASTER Values('CR001', 'Silver', 200, 5);

Insert into LIBRARY_CARD_MASTER Values('CR002', 'Gold', 400, 9);

Insert into LIBRARY_CARD_MASTER Values('CR003', 'Platinum', 600, 8);

Insert into LIBRARY_CARD_MASTER Values('CR004', 'VISA', 800, 7);

Insert into LIBRARY_CARD_MASTER Values('CR005', 'CREDIT', 1200, 6);

Insert into MOVIES_MASTER Values('MV001', 'DIEHARD', '2012-05-
13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR1','L1','L2',100);

Insert into MOVIES_MASTER Values('MV002', 'THE MATRIX', '2012-05-
13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR2','L1','L2',100);

Insert into MOVIES_MASTER Values('MV003', 'INCEPTION', '2012-05-
13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR3','L15','L2',100);

Insert into MOVIES_MASTER Values('MV004', 'DARK KNIGHT', '2012-05-
13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR4','L15','L2',100);

Insert into MOVIES_MASTER Values('MV005', 'OFFICE S', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR5','L12','L24',100);

Insert into MOVIES_MASTER Values('MV006', 'SHAWN OF DEAD', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR6','L1','L25',100);

Insert into MOVIES_MASTER Values('MV007', 'YOUNG FRANKEN', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR7','L1','L2',100);

Insert into MOVIES_MASTER Values('MV008', 'CAS', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR8','L1','L2',100);

Insert into MOVIES_MASTER Values('MV009', 'GWW', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR9','L1','L2',100);

Insert into MOVIES_MASTER Values('MV010', 'TITANIC', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR10','L1','L2',100);

Insert into MOVIES_MASTER Values('MV011', 'THE NOTE BOOK', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR11','L1','L2',100);

Insert into CUSTOMER_CARD_DETAILS Values('CUS001', 'CR001', '2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS002', 'CR002', '2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS003', 'CR002', '2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS004', 'CR003', '2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS005', 'CR003', '2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS001', 'CUS001',
'MV001', '2012-05-13', '2012-05-13','2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS002', 'CUS001',
'MV001', '2012-05-01', '2012-05-16','2012-05-16');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS003', 'CUS002',
'MV004', '2012-05-02', '2012-05-06','2012-05-16');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS004', 'CUS002',
'MV004', '2012-04-03', '2012-04-16','2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS005', 'CUS002',
'MV009', '2012-04-04', '2012-04-16','2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS006', 'CUS003',
'MV002', '2012-03-30', '2012-04-15','2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS007', 'CUS003',
'MV003', '2012-04-20', '2012-05-05','2012-05-05');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS008', 'CUS003',
'MV005', '2012-04-21', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS009', 'CUS003',
'MV001', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS010', 'CUS003',
'MV009', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS011', 'CUS003',
'MV010', '2012-04-23', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS012', 'CUS003',
'MV010', '2012-04-24', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS013', 'CUS003',
'MV008', '2012-04-25', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS014', 'CUS004',
'MV007', '2012-04-26', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS015', 'CUS004',
'MV006', '2012-04-27', '2012-05-07','2012-05-25');

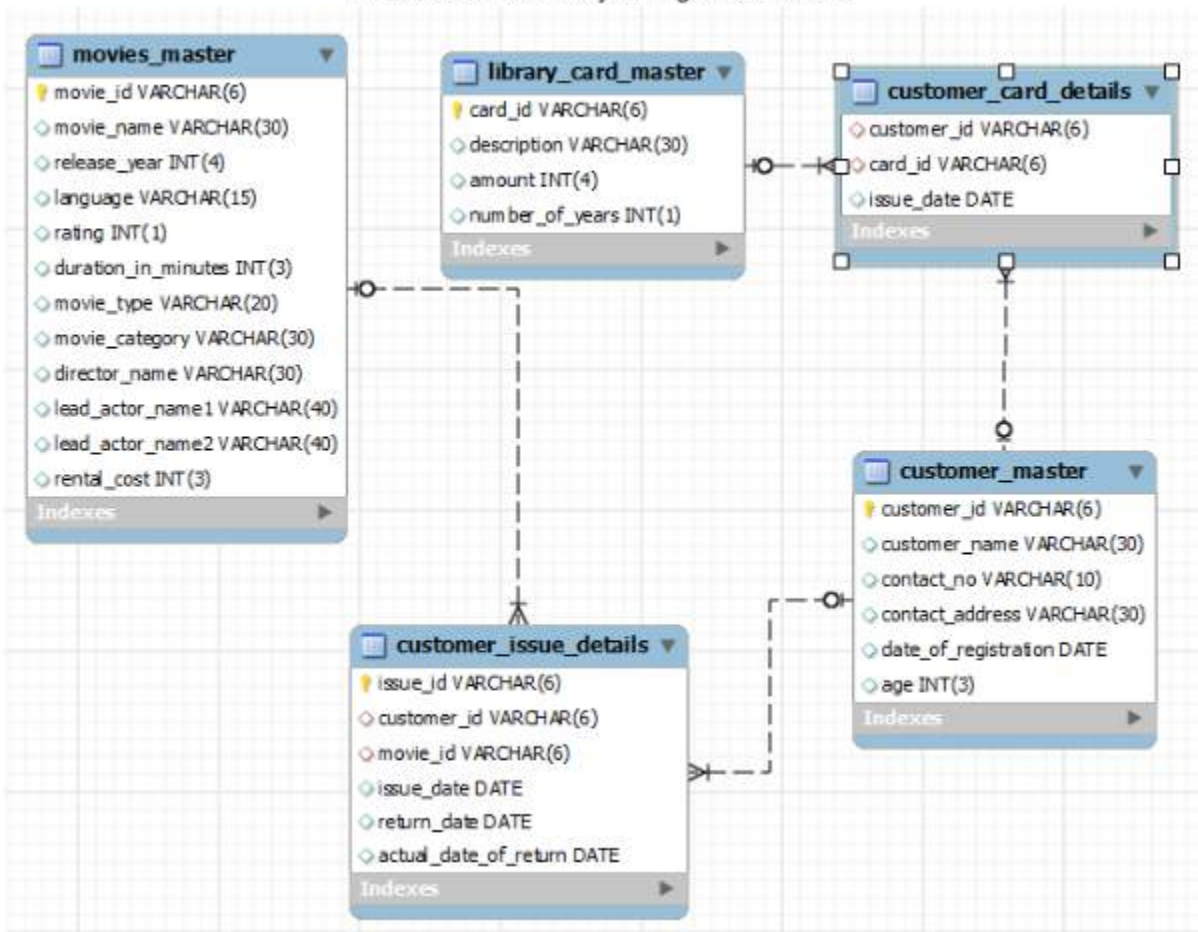
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS016', 'CUS004',
'MV006', '2012-04-28', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS017', 'CUS004',
'MV001', '2012-04-29', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS018', 'CUS010',
'MV008', '2012-04-24', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS019', 'CUS011',
'MV009', '2012-04-27', '2012-05-07','2012-05-25');

ANSI SQL Video Library Management Schema



MOVIE MASTER

[illegible]

LEAD_ROLE_2	RENT_COST
L2	100
L2	100
L2	100
L2	100
L24	100
L25	100
L2	100
L2	100
L2	100
L2	100
L2	100
NULL	NULL

CUSTOMER MASTER

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADD	DATE_OF_REGISTRATION	AGE
CUS001	AMIT	9876543210	ADD1	2012-02-12	21
CUS002	ABDHUL	8765432109	ADD2	2012-02-12	21
CUS003	GAYAN	7654321098	ADD3	2012-02-12	21
CUS004	RADHA	6543210987	ADD4	2012-02-12	21
CUS005	GURU	NULL	ADD5	2012-02-12	21
CUS006	MOHAN	4321098765	ADD6	2012-02-12	21
CUS007	NAME7	3210987654	ADD7	2012-02-12	21
CUS008	NAME8	2109876543	ADD8	2013-02-12	21
CUS009	NAME9	NULL	ADD9	2013-02-12	21
CUS010	NAM10	9934567890	ADD10	2013-02-12	21
CUS011	NAM11	9875678910	ADD11	2013-02-12	21
NULL	NULL	NULL	NULL	NULL	NULL

LIBRARY CARD MASTER

CARD_ID	DESCRIPTION	AMOUNT	NUMBER_OF_YEARS
CR001	Silver	200	5
CR002	Gold	400	9
CR003	Platinum	600	8
CR004	VISA	800	7
CR005	CREDIT	1200	6
NULL	NULL	NULL	NULL

CUSTOMER CARD DETAILS

CUSTOMER_ID	CARD_ID	ISSUE_DATE
CUS001	CR001	2012-05-13
CUS002	CR002	2012-05-13
CUS003	CR002	2013-05-13
CUS004	CR003	2013-05-13
CUS005	CR003	2012-05-13
NULL	NULL	NULL

CUSTOMER ISSUE DETAILS

ISSUE_ID	CUSTOMER_ID	MOVIE_ID	ISSUE_DATE	RETURN_DATE	ACTUAL_DATE_RETURN
IS001	CUS001	MV001	2012-05-13	2012-05-13	2012-05-13
IS002	CUS001	MV001	2012-05-01	2012-05-16	2012-05-16
IS003	CUS002	MV004	2012-05-02	2012-05-06	2012-05-16
IS004	CUS002	MV004	2012-04-03	2012-04-16	2012-04-20
IS005	CUS002	MV009	2012-04-04	2012-04-16	2012-04-20
IS006	CUS003	MV002	2012-03-30	2012-04-15	2012-04-20
IS007	CUS003	MV003	2012-04-20	2012-05-05	2012-05-05
IS008	CUS003	MV005	2012-04-21	2012-05-07	2012-05-25
IS009	CUS003	MV001	2012-04-22	2012-05-07	2012-05-25
IS010	CUS003	MV009	2012-04-22	2012-05-07	2012-05-25
IS011	CUS003	MV010	2012-04-23	2012-05-07	2012-05-25
IS012	CUS003	MV010	2012-04-24	2012-05-07	2012-05-25
IS013	CUS003	MV008	2012-04-25	2012-05-07	2012-05-25
IS014	CUS004	MV007	2012-04-26	2012-05-07	2012-05-25
IS015	CUS004	MV006	2012-04-27	2012-05-07	2012-05-25
IS016	CUS004	MV006	2012-04-28	2012-05-07	2012-05-25
IS017	CUS004	MV001	2012-04-29	2012-05-07	2012-05-25
IS018	CUS010	MV008	2012-04-24	2012-05-07	2012-05-25
IS019	CUS011	MV009	2012-04-27	2012-05-07	2012-05-25
NULL	NULL	NULL	NULL	NULL	NULL

1. Write a query to display movie names and number of times that movie is issued to customers. In case movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order). The Alias name for the number of movies issued is ISSUE_COUNT.

```

SELECT m.MOVIE_NAME,count(ISSUE_ID) ISSUE_COUNT FROM
movies_master m LEFT JOIN customer_issue_details c ON
m.MOVIE_ID=c.MOVIE_ID

GROUP BY m.movie_name

ORDER BY ISSUE_COUNT DESC,MOVIE_NAME;

```

MOVIE_NAME	ISSUE_COUNT
DIEHARD	4
GWW	3
CAS	2
DARK KNIGHT	2
SHAWN OF DEAD	2
TITANIC	2
INCEPTION	1
OFFICE S	1
THE MATRIX	1
YOUNG FRANKEN	1
THE NOTE BOOK	0

2. Write a query to display id,name,age,contact no of customers whose age is greater than 25 and and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

```

SELECT CUSTOMER_ID,CUSTOMER_NAME,AGE,ifnull(
concat('+91-',substring(contact_no,1,3),'-',
substring(contact_no,4,3),'-',substring(contact_no,7)), 'N/A')
CONTACT_ISD

FROM customer_master WHERE age>25 and
year(date_of_registration)='2012'

```

ORDER BY age,CUSTOMER_NAME;

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
-------------	---------------	-----	-------------

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO_OF_MOVIES as alias name for number of movies.

```
SELECT MOVIE_CATEGORY,count(MOVIE_ID) NO_OF_MOVIES FROM
movies_master GROUP BY MOVIE_CATEGORY
ORDER BY NO_OF_MOVIES DESC,MOVIE_CATEGORY;
```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Write a query to display the number of customers having card with description “Gold card”. Hint: Use CUSTOMER_COUNT as alias name for number of customers

```
SELECT count(c.customer_id) CUSTOMER_COUNT FROM
library_card_master l JOIN customer_card_details c ON
l.CARD_ID=c.CARD_ID
WHERE description='Gold';
```

CUSTOMER_COUNT
2

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who

hold library card. Display the records sorted by customer name in descending order. Use REGISTERED_YEAR as alias name for year of registration.

```
SELECT c.customer_id,c.customer_name,  
year(c.DATE_OF_REGISTRATION)  
REGISTERED_YEAR,cd.card_id,cd.issue_date FROM  
customer_master c JOIN customer_card_details cd ON  
c.customer_id=cd.customer_id  
ORDER BY CUSTOMER_NAME DESC;
```

customer_id	customer_name	REGISTERED_YEAR	card_id	issue_date
CUS004	RADHA	2012	CR003	2013-05-13
CUS005	GURU	2012	CR003	2012-05-13
CUS003	GAYAN	2012	CR002	2013-05-13
CUS001	AMIT	2012	CR001	2012-05-13
CUS002	ABDHUL	2012	CR002	2012-05-13

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order by customer name.

```
SELECT ci.issue_id,ci.CUSTOMER_ID,c.CUSTOMER_NAME FROM  
customer_master c JOIN customer_issue_details ci ON  
c.customer_id=ci.customer_id  
WHERE customer_name LIKE 'R%' and  
ci.actual_date_return>ci.return_date  
ORDER BY customer_name;
```

issue_id	CUSTOMER_ID	CUSTOMER_NAME
IS014	CUS004	RADHA
IS015	CUS004	RADHA
IS016	CUS004	RADHA
IS017	CUS004	RADHA

7. Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10. Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name.

```
SELECT c.CUSTOMER_ID,c.CUSTOMER_NAME,l.card_id,l.DESCRPTION,
concat('$',round(amount/52.42)) AMOUNT_DOLLAR FROM
customer_master c JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id
JOIN customer_card_details cc ON cc.customer_id=c.customer_id
JOIN library_card_master l ON cc.card_id=l.card_id
WHERE c.DATE_OF_REGISTRATION=ci.issue_date
ORDER BY customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	card_id	DESCRIPTION	AMOUNT_DOLLAR
-------------	---------------	---------	-------------	---------------

8. Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from

**library without library card and whose address ends with 'Nagar'.
Display customer name in upper case. Hint: Use CUSTOMER_NAME as
alias name for customer name. Display the details sorted in ascending
order based on customer name.**

```
SELECT CUSTOMER_ID,upper(CUSTOMER_NAME)
CUSTOMER_NAME,contact_no,contact_add FROM
customer_master WHERE contact_add LIKE '%Nagar' and
customer_id NOT IN (SELECT customer_id FROM
customer_card_details)
and customer_id IN (SELECT customer_id FROM
customer_issue_details)
ORDER BY CUSTOMER_NAME;
```

CUSTOMER_ID	CUSTOMER_NAME	contact_no	contact_add
-------------	---------------	------------	-------------

**9. Write a query to display the movie id, movie name, release
year, director name of movies acted by the lead actor1 who acted
maximum number of movies. Display the records sorted in ascending
order based on movie name.**

```
SELECT movie_id,movie_name,release_date,director FROM
movies_master
WHERE lead_role_1 IN(SELECT lead_role_1 FROM
(SELECT lead_role_1,count(movie_id)ct FROM movies_master
GROUP BY lead_role_1)t WHERE t.ct>=ALL(SELECT count(movie_id)
FROM movies_master GROUP BY lead_role_1)) ORDER BY
movie_name;
```


movie_id	movie_name	release_date	director
MV008	CAS	2012-05-13	DIR8
MV001	DIEHARD	2012-05-13	DIR1
MV009	GWW	2012-05-13	DIR9
MV006	SHAWN OF DEAD	2012-05-13	DIR6
MV002	THE MATRIX	2012-05-13	DIR2
MV011	THE NOTE BOOK	2012-05-13	DIR11
MV010	TITANIC	2012-05-13	DIR10
MV007	YOUNG FRANK...	2012-05-13	DIR7

10. Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.
**
Hint: Use MOVIE_COUNT as alias name for number of movies issued.**

```
SELECT c.customer_name, count(ci.movie_id) MOVIE_COUNT FROM
customer_master c LEFT JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id
GROUP BY c.customer_id ORDER BY c.customer_name;
```

customer_name	MOVIE_COUNT
ABDHUL	3
AMIT	2
GAYAN	8
GURU	0
MOHAN	0
NAM10	1
NAM11	1
NAME7	0
NAME8	0
NAME9	0
RADHA	4

11. Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number. Serial number can be generated from the issue id , that is last two characters of issue id is the serial number. For Example Assume the issue id is I00005 then the serial number is 05 Hint: Alias name for serial number is 'SERIAL_NO'

```
SELECT substring(ci.issue_id,-2)
SERIAL_NO,ci.issue_id,c.customer_id,c.customer_name,
m.movie_id,m.movie_name FROM customer_master c JOIN
customer_issue_details ci
ON c.customer_id=ci.customer_id JOIN movies_master m ON
m.movie_id=ci.movie_id
ORDER BY SERIAL_NO;
```

SERIAL_NO	issue_id	customer_id	customer_name	movie_id	movie_name
01	IS001	CUS001	AMIT	MV001	DIEHARD
02	IS002	CUS001	AMIT	MV001	DIEHARD
03	IS003	CUS002	ABDHUL	MV004	DARK KNIGHT
04	IS004	CUS002	ABDHUL	MV004	DARK KNIGHT
05	IS005	CUS002	ABDHUL	MV009	GWW
06	IS006	CUS003	GAYAN	MV002	THE MATRIX
07	IS007	CUS003	GAYAN	MV003	INCEPTION
08	IS008	CUS003	GAYAN	MV005	OFFICE S
09	IS009	CUS003	GAYAN	MV001	DIEHARD
10	IS010	CUS003	GAYAN	MV009	GWW
11	IS011	CUS003	GAYAN	MV010	TITANIC
12	IS012	CUS003	GAYAN	MV010	TITANIC
13	IS013	CUS003	GAYAN	MV008	CAS
14	IS014	CUS004	RADHA	MV007	YOUNG FRAN...
15	IS015	CUS004	RADHA	MV006	SHAWN OF D...
16	IS016	CUS004	RADHA	MV006	SHAWN OF D...
17	IS017	CUS004	RADHA	MV001	DIEHARD
18	IS018	CUS010	NAM10	MV008	CAS
19	IS019	CUS011	NAM11	MV009	GWW

12. Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

```

SELECT
ci.issue_id,ci.issue_date,c.customer_id,c.customer_name,c.contact_no
FROM
customer_master c JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id
and year(ci.issue_date)='2013' ORDER BY ci.issue_date DESC;

```

issue_id	issue_date	customer_id	customer_name	contact_no
----------	------------	-------------	---------------	------------

**13. Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format. LEAD_ACTOR_ONE space ambersant space LEAD_ACTOR_TWO. Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria" Hint: Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.**

```
SELECT movie_id, movie_name, concat(lead_role_1, ' & ', lead_role_2)
ACTOR FROM movies_master
```

```
WHERE movie_id NOT IN (SELECT movie_id FROM
customer_issue_details) ORDER BY movie_name;
```

movie_id	movie_name	ACTOR
MV011	THE NOTE BOOK	L1 & L2

14. Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

```
SELECT upper(director) DIRECTOR_NAME, movie_name, lead_role_1
FROM movies_master
```

GROUP BY director HAVING count(movie_id)>1 ORDER BY
director,movie_name DESC;

DIRECTOR_NAME	movie_name	lead_role_1
---------------	------------	-------------

**15. Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
 Hint: Use NO_OF_CUSTOMERS as alias name for number of customers.**

```
SELECT count(customer_id) NO_OF_CUSTOMER FROM  
customer_master  
  
WHERE contact_no is not null and year(date_of_registration)='2012';
```

NO_OF_CUSTOMER
6

16. Write a query to display the customer's name, contact number, library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.

```
SELECT c.customer_name,ifnull(c.contact_no,c.contact_add)  
CONTACT_DETAILS,l.card_id,l.description FROM  
  
customer_master c LEFT JOIN customer_card_details cc ON  
c.customer_id=cc.customer_id  
  
LEFT JOIN library_card_master l ON l.card_id=cc.card_id  
  
ORDER BY customer_name;
```

customer_name	CONTACT_DETAILS	card_id	description
ABDHUL	8765432109	CR002	Gold
AMIT	9876543210	CR001	Silver
GAYAN	7654321098	CR002	Gold
GURU	ADD5	CR003	Platinum
MOHAN	4321098765	NULL	NULL
NAM10	9934567890	NULL	NULL
NAM11	9875678910	NULL	NULL
NAME7	3210987654	NULL	NULL
NAME8	2109876543	NULL	NULL
NAME9	ADD9	NULL	NULL
RADHA	6543210987	CR003	Platinum

17. Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO_OF_TIMES as alias name for number of times

```
SELECT ci.customer_id,c.customer_name,count(ci.movie_id)
NO_OF_TIMES FROM
```

```
customer_issue_details ci JOIN customer_master c ON
c.customer_id=ci.customer_id
```

```
GROUP BY ci.customer_id,ci.movie_id HAVING count(movie_id)>1
```

```
ORDER BY customer_name DESC;
```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	2
CUS003	GAYAN	2
CUS001	AMIT	2
CUS002	ABDHUL	2

18. Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format. Hint: Use NO_OF_MOVIES as alias name for number of movies column. Hint: Use CONTACT_ISD as alias name for contact number. Display the records sorted in ascending order based on customer name and then by movie category.

```
SELECT c.customer_id,c.customer_name,concat('+91-
',substring(c.contact_no,1,3),'-',
substring(c.contact_no,4,3),'-',substring(c.contact_no,7)) CONTACT_ISD
,m.movie_category,count(cc.movie_id) NO_OF_MOVIES FROM
customer_master c JOIN customer_issue_details cc
ON c.customer_id=cc.customer_id JOIN movies_master m ON
m.movie_id=cc.movie_id
GROUP BY c.customer_id,m.movie_category HAVING
count(cc.movie_id)>1
ORDER BY customer_name,movie_category;
```

customer_id	customer_name	CONTACT_ISD	movie_category	NO_OF_MOVIES
CUS002	ABDHUL	+91-876-543-2109	ACTION	2
CUS001	AMIT	+91-987-654-3210	ACTION	2
CUS003	GAYAN	+91-765-432-1098	ACTION	3
CUS003	GAYAN	+91-765-432-1098	ROMANCE	4
CUS004	RADHA	+91-654-321-0987	COMEDY	3

19. Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies. For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The

name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

```
SELECT cid.customer_id , customer_name FROM customer_master cm
JOIN customer_issue_details cid ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)>=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id)
UNION
SELECT cid.customer_id , customer_name FROM
customer_master cm JOIN customer_issue_details cid
ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)<=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id) ORDER BY customer_name;
```


customer_id	customer_name
CUS003	GAYAN
CUS010	NAM10
CUS011	NAM11

20. Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once. Hint: Use NO_OF_TIMES as alias name Display the records in ascending order based on customer name.

```
SELECT c.customer_id,c.customer_name,count(m.movie_id)
NO_OF_TIMES FROM
customer_master c JOIN customer_issue_details cc ON
c.customer_id=cc.customer_id
JOIN movies_master m ON m.movie_id=cc.movie_id
WHERE m.movie_category='Comedy'
GROUP BY c.customer_id HAVING count(m.movie_id)>1
ORDER BY customer_name;
```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	3

21. Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST. Display the records sorted in ascending order based on customer id

```
SELECT cid.customer_id, sum(m.rent_cost) TOTAL_COST FROM
customer_issue_details cid JOIN movies_master mm ON
cid.movie_id=mm.movie_id GROUP BY cid.customer_id order by
customer_id;
```

customer_id	TOTAL_COST
CUS001	200
CUS002	300
CUS003	800
CUS004	400
CUS010	100
CUS011	100

LOAN

BANK

```
create database bank;
```

```
use bank;
```

```
CREATE TABLE customer_master(  
CUSTOMER_NUMBER VARCHAR(6),  
FIRSTNAME VARCHAR(30),  
middlename VARCHAR(30),  
lastname VARCHAR(30),  
CUSTOMER_CITY VARCHAR(15),  
CUSTOMER_CONTACT_NO VARCHAR(10),  
occupation VARCHAR(10),  
CUSTOMER_DATE_OF_BIRTH DATE,  
CONSTRAINT customer_custid_pk PRIMARY KEY  
(CUSTOMER_NUMBER));
```

```
CREATE TABLE branch_master(  
branch_id VARCHAR(6),
```

branch_name VARCHAR(30),
branch_city VARCHAR(30),
CONSTRAINT branch_bid_pk PRIMARY KEY (branch_id));

CREATE TABLE **account_master**
(account_number VARCHAR(255),
customer_number VARCHAR(255),
branch_id VARCHAR(255),
opening_balance INT(20),
account_opening_date DATE,
account_type VARCHAR(10),
account_status VARCHAR(10),
PRIMARY KEY (account_number),
FOREIGN KEY (customer_number) references
customer_master(customer_number),
FOREIGN KEY (branch_id) references branch_master(branch_id));

CREATE TABLE **transaction_details**(
transaction_number VARCHAR(6),
account_number VARCHAR(6),

```
date_of_transaction DATE,  
medium_of_transaction VARCHAR(20),  
transaction_type VARCHAR(20),  
transaction_amount INT(7),  
CONSTRAINT transaction_details_tnumber_pk PRIMARY KEY  
(transaction_number),  
CONSTRAINT transaction_details_acnumber_fk FOREIGN KEY  
(account_number)  
REFERENCES account_master (account_number));
```

```
CREATE TABLE loan_details  
(customer_number varchar(255),  
branch_id varchar(255),  
loan_amount bigint(20),  
foreign key(customer_number) references  
customer_master(customer_number));
```

```
insert into customer_master values('C00001',    'RAMESH',  
    'CHANDRA',    'SHARMA', 'DELHI',    '9543198345', 'SERVICE'  
    , '1976-12-06');  
  
insert into customer_master values('C00002',    'AVINASH', 'SUNDER',  
    'MINHA', 'DELHI',    '9876532109'    , 'SERVICE', '1974-10-16');
```

```
insert into customer_master values('C00003',    'RAHUL',  'NULL',  
    'RASTOGI', 'DELHI',    '9765178901',  'STUDENT',    '1981-09-  
26');
```

```
insert into customer_master values('C00004',    'PARUL',  'NULL',  
    'GANDHI', 'DELHI',    '9876532109'  , 'HOUSEWIFE', '1976-11-  
03');
```

```
insert into customer_master values('C00005',    'NAVEEN'  
    , 'CHANDRA',    'AEDEKAR',    'MUMBAI', '8976523190',  
    'SERVICE'  , '1976-09-19');
```

```
insert into customer_master values('C00006',    'CHITRESH',  
    'NULL',    'BARWE', 'MUMBAI', '7651298321',  'STUDENT'  
    , '1992-11-06');
```

```
insert into customer_master values('C00007',    'AMIT'    , 'KUMAR',  
    'BORKAR', 'MUMBAI', '9875189761',  'STUDENT',    '1981-09-  
06');
```

```
insert into customer_master values('C00008',    'NISHA',  NULL,  
    'DAMLE',  'MUMBAI', '7954198761',  'SERVICE', '1975-12-03');
```

```
insert into customer_master values('C00009',    'ABHISHEK',  
    NULL,    'DUTTA',  'KOLKATA', '9856198761',  'SERVICE'  
    , '1973-05-22');
```

```
insert into customer_master values('C00010', 'SHANKAR'    , NULL,  
    'NAIR',    'CHENNAI', '8765489076',  'SERVICE', '1976-07-12');
```

```
insert into branch_master values('B00001', 'ASAF ALI ROAD', 'DELHI');
```

```
insert into branch_master values('B00002', 'NEW DELHI MAIN  
BRANCH', 'DELHI');
```

```
insert into branch_master values('B00003' , 'DELHI CANTT', 'DELHI');
insert into branch_master values('B00004' , 'JASOLA', 'DELHI');
insert into branch_master values('B00005' , 'MAHIM' , 'MUMBAI');
insert into branch_master values('B00006' , 'VILE PARLE',
    'MUMBAI');
insert into branch_master values('B00007', 'MANDVI' , 'MUMBAI');
insert into branch_master values('B00008' , 'JADAVPUR',
    'KOLKATA');
insert into branch_master values('B00009' , 'KODAMBAKKAM',
    'CHENNAI');
```

```
insert into account_master values('A00001' , 'C00001', 'B00001', 1000
    , '2012-12-15', 'SAVING', 'ACTIVE');
insert into account_master values('A00002' , 'C00002', 'B00001', 1000
    , '2012-06-12' , 'SAVING', 'ACTIVE');
insert into account_master values('A00003' , 'C00003', 'B00002', 1000
    , '2012-05-17' , 'SAVING', 'ACTIVE');
insert into account_master values('A00004' , 'C00002', 'B00005', 1000
    , '2013-01-27' , 'SAVING' , 'ACTIVE');
insert into account_master values('A00005' , 'C00006', 'B00006', 1000
    , '2012-12-17' , 'SAVING', 'ACTIVE');
insert into account_master values('A00006' , 'C00007', 'B00007', 1000
    , '2010-08-12' , 'SAVING' , 'SUSPENDED');
```

```
insert into account_master values('A00007' , 'C00007', 'B00001', 1000
    , '2012-10-02'    , 'SAVING'  , 'ACTIVE');
```

```
insert into account_master values('A00008' , 'C00001', 'B00003', 1000
    , '2009-11-09'    , 'SAVING'  , 'TERMINATED');
```

```
insert into account_master values('A00009' , 'C00003', 'B00007', 1000
    , '2008-11-30'    , 'SAVING'  , 'TERMINATED');
```

```
insert into account_master values('A00010' , 'C00004', 'B00002', 1000
    , '2013-03-01'    , 'SAVING'  , 'ACTIVE');
```

```
insert into transaction_details values('T00001', 'A00001', '2013-01-
01', 'CHEQUE', 'DEPOSIT', 2000);
```

```
insert into transaction_details values('T00002' , 'A00001' , '2013-02-
01' , 'CASH' , 'WITHDRAWAL', 1000);
```

```
insert into transaction_details values('T00003', 'A00002' , '2013-
01-01', 'CASH' , 'DEPOSIT', 2000);
```

```
insert into transaction_details values('T00004', 'A00002', '2013-02-
01' , 'CASH' , 'DEPOSIT', 3000);
```

```
insert into transaction_details values('T00005', 'A00007', '2013-01-
11', 'CASH' , 'DEPOSIT', 7000);
```

```
insert into transaction_details values('T00006', 'A00007', '2013-01-
13', 'CASH' , 'DEPOSIT', 9000);
```

```
insert into transaction_details values('T00007', 'A00001', '2013-03-
13', 'CASH' , 'DEPOSIT' , 4000);
```

```
insert into transaction_details values('T00008', 'A00001', '2013-03-
14', 'CHEQUE' , 'DEPOSIT' , 3000);
```



```
insert into transaction_details values('T00009', 'A00001', '2013-03-21', 'CASH', 'WITHDRAWAL', 9000);
```

```
insert into transaction_details values('T00010', 'A00001', '2013-03-22', 'CASH', 'WITHDRAWAL', 2000);
```

```
insert into transaction_details values('T00011', 'A00002', '2013-03-25', 'CASH', 'WITHDRAWAL', 7000);
```

```
insert into transaction_details values('T00012', 'A00007', '2013-03-26', 'CASH', 'WITHDRAWAL', 2000);
```

```
insert into Loan_details values('C00001', 'B00001', 100000);
```

```
insert into Loan_details values('C00002', 'B00002', 200000);
```

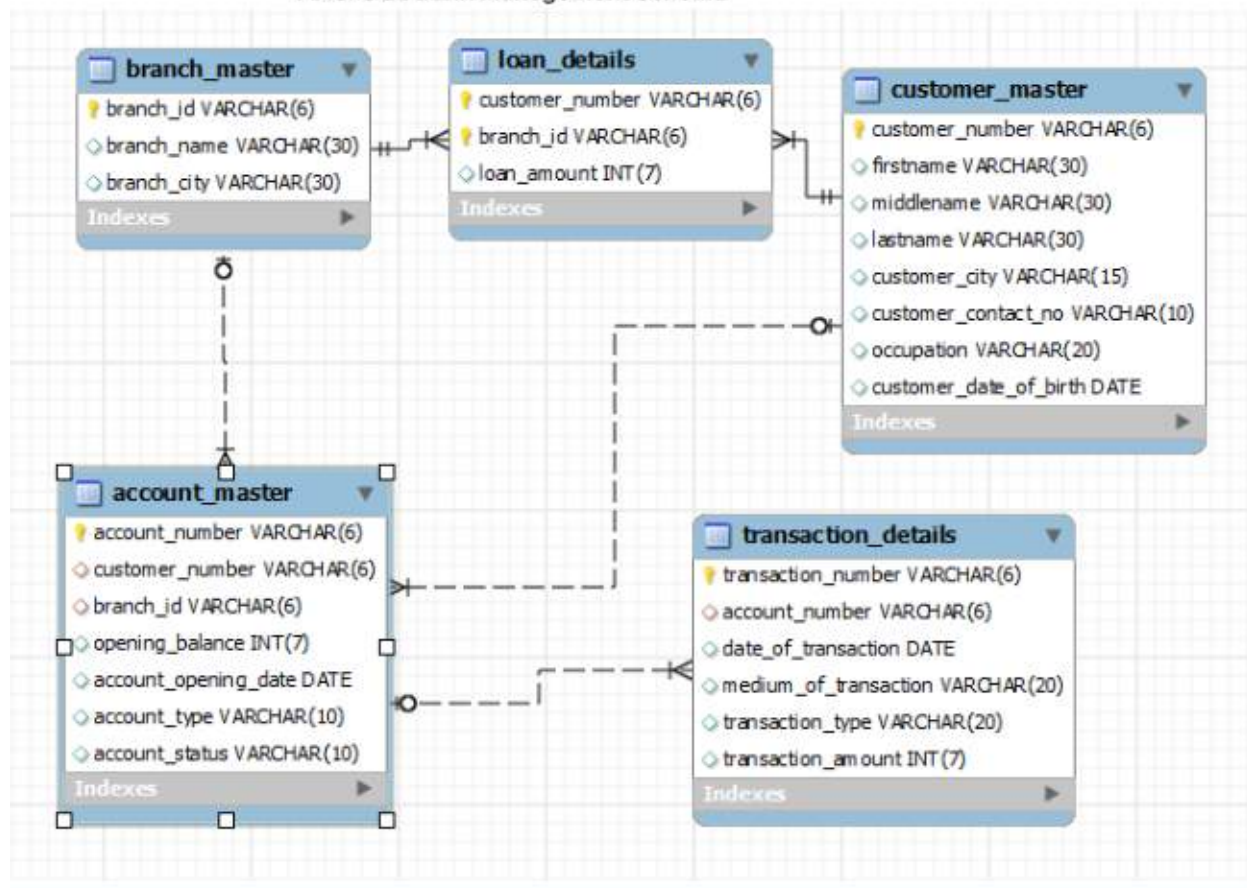
```
insert into Loan_details values('C00009', 'B00008', 400000);
```

```
insert into Loan_details values('C00010', 'B00009', 500000);
```

```
insert into Loan_details values('C00001', 'B00003', 600000);
```

```
insert into Loan_details values('C00002', 'B00001', 600000);
```

ANSI SQL Bank Management Schema



CUSTOMER MASTER

[illegible]

ACCOUNT MASTER

account_number	customer_number	branch_id	opening_balance	account_opening_date	account_type	account_status
A00001	C00001	B00001	1000	2012-12-15	SAVING	ACTIVE
A00002	C00002	B00001	1000	2012-06-12	SAVING	ACTIVE
A00003	C00003	B00002	1000	2012-05-17	SAVING	ACTIVE
A00004	C00002	B00005	1000	2013-01-27	SAVING	ACTIVE
A00005	C00006	B00006	1000	2012-12-17	SAVING	ACTIVE
A00006	C00007	B00007	1000	2010-08-12	SAVING	SUSPENDED
A00007	C00007	B00001	1000	2012-10-02	SAVING	ACTIVE
A00008	C00001	B00003	1000	2009-11-09	SAVING	TERMINATED
A00009	C00003	B00007	1000	2008-11-30	SAVING	TERMINATED
A00010	C00004	B00002	1000	2013-03-01	SAVING	ACTIVE
NULL	NULL	NULL	NULL	NULL	NULL	NULL

BRANCH MASTER

branch_id	branch_name	branch_city
B00001	ASAF ALI ROAD	DELHI
B00002	NEW DELHI MAIN BRANCH	DELHI
B00003	DELHI CANTT	DELHI
B00004	JASOLA	DELHI
B00005	MAHIM	MUMBAI
B00006	VILE PARLE	MUMBAI
B00007	MANDVI	MUMBAI
B00008	JADAVPUR	KOLKATA
B00009	KODAMBAKKAM	CHENNAI
NULL	NULL	NULL

LOAN DETAILS

customer_number	branch_id	loan_amount
C00001	B00001	100000
C00002	B00002	200000
C00009	B00008	400000
C00010	B00009	500000
C00001	B00003	600000
C00002	B00001	600000

TRANSACTION DETAILS

transaction_number	account_number	date_of_transaction	medium_of_transaction	transaction_type	transaction_amount
T00001	A00001	2013-01-01	CHEQUE	DEPOSIT	2000
T00002	A00001	2013-02-01	CASH	WITHDRAWAL	1000
T00003	A00002	2013-01-01	CASH	DEPOSIT	2000
T00004	A00002	2013-02-01	CASH	DEPOSIT	3000
T00005	A00007	2013-01-11	CASH	DEPOSIT	7000
T00006	A00007	2013-01-13	CASH	DEPOSIT	9000
T00007	A00001	2013-03-13	CASH	DEPOSIT	4000
T00008	A00001	2013-03-14	CHEQUE	DEPOSIT	3000
T00009	A00001	2013-03-21	CASH	WITHDRAWAL	9000
T00010	A00001	2013-03-22	CASH	WITHDRAWAL	2000
T00011	A00002	2013-03-25	CASH	WITHDRAWAL	7000
T00012	A00007	2013-03-26	CASH	WITHDRAWAL	2000
NULL	NULL	NULL	NULL	NULL	NULL

QUERIES

1. Write a query to display account number, customer's number, customer's firstname, lastname, account opening date. Display the records sorted in ascending order based on account number.

SELECT

a.account_number,c.customer_number,c.firstname,c.lastname,a.account_number

FROM customer_master c JOIN account_master a ON
c.customer_number=a.customer_number

ORDER BY a.account_number;

account_number	customer_number	firstname	lastname	account_opening_date
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27
A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09
A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2. Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

SELECT count(customer_number) Cust_Count FROM customer_master
WHERE customer_city='Delhi';

cust_count
4

3. Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number FROM
account_master a join customer_master c ON
c.customer_number=a.customer_number WHERE
day(a.account_opening_date)>'15' ORDER BY
c.customer_number,a.account_number;
```

customer_number	firstname	account_number
C00002	AVINASH	A00004
C00003	RAHUL	A00003
C00003	RAHUL	A00009
C00006	CHITRESH	A00005

4. Write a query to display customer number, customer's first name, account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number
FROM account_master a JOIN customer_master c
ON c.customer_number=a.customer_number
```

WHERE a.account_status='Terminated'

ORDER BY c.customer_number,a.account_number;

customer_number	firstname	account_number
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions. Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type,count(transaction_number) Trans_Count
FROM account_master am JOIN transaction_details td
ON am.account_number=td.account_number
WHERE customer_number like '%001'
GROUP BY transaction_type
ORDER BY transaction_type;
```

transaction_type	Trans_count
DEPOSIT	3
WITHDRAWAL	3

6. Write a query to display the number of customers who have registration but no account in the bank. Give the alias name as Count_Customer for number of customers.

```
SELECT      count(customer_number)      Count_Customer      FROM
customer_master

WHERE customer_number NOT IN (SELECT customer_number FROM
account_master);
```

Count_customer
4

7. Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT
a.account_number,a.opening_balance+sum(t.transaction_amount)
FROM  account_master  a  JOIN  transaction_details  t  ON
a.account_number=t.account_number

WHERE t.transaction_type='Deposit' GROUP BY t.account_number;
```

account_number	Deposit_Amount
A00001	10000
A00002	6000
A00007	17000

8. Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```
SELECT      branch.branch_city,      count(account.account_number)
No_of_Accounts

FROM branch_master LEFT JOIN account_master

ON account.branch_id=branch.branch_id

GROUP BY branch.branch_city ORDER BY branch_city;
```

branch_city	No_of_accounts
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9. Write a query to display the firstname of the customers who have more than 1 account. Display the records in sorted order based on firstname.

```
SELECT c.firstname FROM

customer_master      c      JOIN      account_master      a      ON

a.customer_number=c.customer_number

GROUP BY a.customer_number HAVING count(a.account_number)>1;
```

firstname
AMIT
AVINASH
RAHUL
RAMESH

10. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.

```
SELECT c.customer_number,c.firstname,c.lastname FROM
customer_master      c      JOIN      loan_details      l      ON
c.customer_number=l.customer_number
GROUP BY l.customer_number HAVING count(l.branch_id)>1
ORDER BY c.customer_number;
```

customer_number	firstname	lastname
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11. Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different. Display the records sorted in ascending order based on customer number.

```
SELECT  c.customer_number,c.firstname,c.customer_city,b.branch_city
FROM
Customer_master      c      JOIN      Account_master      a      ON
c.customer_number=a.customer_number
JOIN Branch_master b ON b.branch_id=a.branch_id
WHERE b.branch_city<>c.customer_city
ORDER BY c.customer_number;
```

customer_number	firstname	customer_city	branch_city
C00002	AVINASH	DELHI	MUMBAI
C00003	RAHUL	DELHI	MUMBAI
C00007	AMIT	MUMBAI	DELHI

12. Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(c.customer_number)Count FROM customer_master c
JOIN loan_details l
```

```
ON c.customer_number=l.customer_number
```

```
WHERE c.customer_number NOT IN (SELECT customer_number FROM
account_master);
```

Count
2

13. Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT account_number FROM transaction_details
```

```
GROUP BY account_number
```

```
HAVING count(transaction_number)>=ALL
```

```
(SELECT count(transaction_number) FROM transaction_details
```

```
GROUP BY account_number) ORDER BY account_number;
```

account_number
A00001

14. Write a query to show the branch name,branch city where we have the maximum customers. For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
SELECT b.branch_name,b.branch_city FROM
Branch_master b JOIN account a ON a.branch_id=b.branch_id
GROUP BY a.branch_id HAVING count(a.customer_number)>=ALL
(SELECT count(customer_number) FROM
Account_master GROUP BY branch_id)
ORDER BY b.branch_name;
```

branch_name	branch_city
ASAF ALI ROAD	DELHI

15. Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well. For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011. Display the records sorted in ascending order based on account number.

```

SELECT td.account_number,
sum(CASE WHEN transaction_type='Deposit' THEN transaction_amount
END)
+(SELECT opening_balance
FROM account_master where account_number=td.account_number)
Deposit,
sum(CASE WHEN transaction_type='Withdrawal' THEN
transaction_amount END) Withdrawal
FROM transaction_details td
GROUP BY td.account_number
HAVING Withdrawal > Deposit
ORDER BY td.account_number;

```

(or)

```

SELECT          ifnull(t1.account_number,t2.account_number)
account_number,
t2.d Deposit,ifnull(t1.w,0) Withdrawal FROM
(SELECT account_number,transaction_type,sum(transaction_amount)
w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT
a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account_master a JOIN transaction_details t ON
a.account_number=t.account_number

```

```

WHERE t.transaction_type='Deposit'GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.w,0)>t2.d
ORDER BY account_number;

```

account_number	Deposit	Withdrawal
A00001	10000	12000
A00002	6000	7000

16. Write a query to show the balance amount for account number that ends with 001. Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount. For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000) – (500+500+1000).

```

SELECT ifnull((SUM(CASE WHEN transaction_type='Deposit'
THEN transaction_amount END)) -
(SUM(CASE WHEN transaction_type='Withdrawal'
THEN transaction_amount END)))+(select opening_balance
from account_master where account_number like '%001'),(SUM(CASE
WHEN transaction_type='Deposit'
THEN transaction_amount END)))+(select opening_balance
from account_master where account_number like '%001')) AS
Balance_Amount
FROM transaction_details where account_number like '%001';

```

(or)

```

SELECT          ifnull(t1.account_number,t2.account_number)
account_number,
t2.d-ifnull(t1.w,0) Balance_Amount FROM
(SELECT  account_number,transaction_type,sum(transaction_amount)
w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT
a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM  account      a      JOIN      transaction_details      t      ON
a.account_number=t.account_number
WHERE t.transaction_type='Deposit'GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.account_number,t2.account_number) LIKE '%001'
ORDER BY account_number;

```

account_number	Balance_Amount
A00001	-2000

17. Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```

SELECT          c.customer_number,c.firstname,t.account_number,
count(t.account_number) Count_Trans

```

```
FROM transaction_details t JOIN account_master a ON  
a.account_number=t.account_number
```

```
JOIN customer c ON c.customer_number=a.customer_number
```

```
GROUP BY t.account_number ORDER BY c.customer_number,  
a.account_number;
```

customer_number	firstname	account_number	Count_Trans
C00001	RAMESH	A00001	6
C00002	AVINASH	A00002	3
C00007	AMIT	A00007	3

18. Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```
SELECT c.firstname FROM
```

```
Customer_master c JOIN account_master a ON  
c.customer_number=a.customer_number
```

```
GROUP BY a.customer_number HAVING count(a.account_number)>1
```

```
ORDER BY c.firstname;
```

firstname
AMIT
AVINASH
RAHUL
RAMESH

19. Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches. For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan

amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011. So total loan taken is 300000. So loan taken by C00012 is more than C00013.

```
SELECT Id.customer_number, firstname, lastname
FROM customer_master cm JOIN loan_details Id
ON cm.customer_number=Id.customer_number
GROUP BY customer_number
HAVING count(branch_id)>=2 AND sum(loan_amount)>=
ALL(SELECT sum(loan_amount) FROM loan GROUP BY
customer_number);
```

customer_number	firstname	lastname
C00002	AVINASH	MINHA

20. Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans. Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT c.customer_number,c.firstname,l.branch_id,l.loan_amount
FROM
Customer_master c JOIN loan_details l ON
c.customer_number=l.customer_number
ORDER BY c.customer_number,l.branch_id,l.loan_amount;
```

customer_number	firstname	branch_id	loan_amount
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000
C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000
C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21. Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch. Display the records sorted in ascending order based on city name.

```
SELECT branch_city,count(branch_id) Count_Branch FROM
Branch_master GROUP BY branch_city
ORDER BY branch_city;
```

branch_city	Count_Branch
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22. Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active. Display the records sorted in ascending order based on account id /account number.

```
SELECT a.account_number,c.firstname,c.lastname FROM
Customer_master      c      JOIN      account_master      a      ON
c.customer_number=a.customer_number      and
a.account_status='Active'
ORDER BY a.account_number;
```

account_number	firstname	lastname
A00001	RAMESH	SHARMA
A00002	AVINASH	MINHA
A00003	RAHUL	RASTOGI
A00004	AVINASH	MINHA
A00005	CHITRESH	BARWE
A00007	AMIT	BORKAR
A00010	PARUL	GANDHI

23. Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name. Display the records sorted in ascending order based on customer number.

```
SELECT      customer_number,firstname,ifnull(middlename,lastname)
Middle_name FROM
```

```
Customer_master ORDER BY customer_number;
```

customer_number	firstname	Middle_name
C00001	RAMESH	CHANDRA
C00002	AVINASH	SUNDER
C00003	RAHUL	NULL
C00004	PARUL	NULL
C00005	NAVEEN	CHANDRA
C00006	CHITRESH	NULL
C00007	AMIT	KUMAR
C00008	NISHA	DAMLE
C00009	ABHISHEK	DUTTA
C00010	SHANKAR	NAIR

24. Write a query to display the customer number , firstname, customer's date of birth . Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth FROM
```

Customer_master

ORDER

BY

year(customer_date_of_birth),customer_number;

customer_number	firstname	customer_date_of_birth
C00009	ABHISHEK	1973-05-22
C00002	AVINASH	1974-10-16
C00008	NISHA	1975-12-03
C00001	RAMESH	1976-12-06
C00004	PARUL	1976-11-03
C00005	NAVEEN	1976-09-19
C00010	SHANKAR	1976-07-12
C00003	RAHUL	1981-09-26
C00007	AMIT	1981-09-06
C00006	CHITRESH	1992-11-06

25. Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student. Display the records sorted in ascending order based on customer first name and then by account number.

SELECT c.firstname,c.customer_city,a.account_number FROM

Customer_master c JOIN account_master a ON
a.customer_number=c.customer_number

WHERE c.occupation NOT IN ('Service','Student','Business')

ORDER BY c.firstname,a.account_number;

firstname	customer_city	account_number
PARUL	DELHI	A00010

AIRLINES

```
create database flight;
```

```
use flight;
```

```
CREATE TABLEair_credit_card_details
```

```
(
```

```
profile_id VARCHAR(10)    NOT NULL,
```

```
card_number    BIGINT,
```

```
card_type VARCHAR(45),
```

```
expiration_month INT,
```

```
expiration_year INT
```

```
);
```

```
CREATE TABLEair_passenger_profile
```

```
(
```

```
profile_id VARCHAR(10) NOT NULL ,
```

```
password VARCHAR(45) NULL ,
```

```
first_name VARCHAR(45) NULL ,
```

```
last_name VARCHAR(45) NULL ,
```

```
address VARCHAR(45) NULL ,
```

```
mobile_number BIGINT NULL ,
```

```
email_id VARCHAR(45) NULL
```

```
);
```

```
CREATE TABLE air_ticket_info
(
ticket_id VARCHAR(45) NOT NULL ,
profile_id VARCHAR(10) NULL ,
flight_id VARCHAR(45) NULL ,
flight_departure_date DATE NULL ,
status VARCHAR(45) NULL
);
```

```
CREATE TABLE air_flight_details
(
flight_id VARCHAR(45) NOT NULL ,
flight_departure_date DATE NULL ,
price DECIMAL(10,2) NULL ,
available_seats INT NULL
);
```

```
CREATE TABLE air_flight
(
flight_id VARCHAR(45) NOT NULL ,
airline_id VARCHAR(45) NULL ,
```

```
airline_name VARCHAR(45) NULL ,  
from_location VARCHAR(45) NULL ,  
to_location VARCHAR(45) NULL ,  
departure_time TIME NULL ,  
arrival_time TIME NULL ,  
duration TIME NULL ,  
total_seats INT NULL  
);
```

```
INSERT INTO air_credit_card_details VALUES  
(1, 622098761234, 'debit', 5, 2013),  
(2, 652362563625, 'credit', 1, 2013),  
(1, 765432345678, 'credit', 2, 2013),  
(3, 654378561234, 'debit', 6, 2013),  
(4, 625417895623, 'debit', 2, 2013),  
(5, 865478956325, 'debit', 3, 2013),  
(6, 789563521457, 'credit', 4, 2013),  
(2, 543267895432, 'credit', 8, 2013),  
(1, 256369856321, 'debit', 1, 2013);
```

INSERT INTO air_flight VALUES

(3173, 'MH370', 'abc', 'hyderabad', 'chennai', '06:30:00',
'07:15:00', '0:45:00', 100),

(3178, 'MH17', 'def', 'chennai', 'hyderabad', '08:00:00',
'09:00:00', '1:00:00', 200),

(3172, 'AR342', 'fgh', 'kolkata', 'chennai', '11:30:00', '13:00:00',
'1:30:00', 100),

(3071, 'JT564', 'jkl', 'chennai', 'delhi', '08:00:00', '10:00:00',
'2:00:00', 100),

(3170, 'DT345', 'xyz', 'delhi', 'kolkata', '21:00:00', '22:30:00',
'1:30:00', 100),

(3175, 'MJ654', 'abc', 'chennai', 'hyderabad', '15:00:00',
'16:00:00', '1:00:00', 200),

(3176, 'MH370', 'def', 'kochi', 'chennai', '18:00:00', '19:05:00',
'1:05:00', 100),

(3177, 'MH45', 'fgh', 'delhi', 'kochi', '19:00:00',
'21:00:00', '2:00:00', 200),

(3174, 'MH321', 'xyz', 'kolkata', 'delhi', '0:00:00', '2:00:00',
'2:00:00', 100),

(3179, 'JT435', 'abc', 'chennai', 'kolkata', '14:00:00', '15:00:00',
'1:00:00', 100),

(3180, 'JT456', 'ijk', 'kolkata', 'kochi', '5:00:00', '5:45:00',
'0:45:00', 200);

INSERT INTO air_flight_details VALUES

(3170, '2013-02-14', 1000, 10),
(3171, '2013-03-15', 5000, 0),
(3172, '2013-02-05', 3000, 32),
(3173, '2013-04-07', 2000, 12),
(3174, '2013-04-05', 3800, 3),
(3175, '2013-05-25', 3500, 10),
(3176, '2013-03-14', 8000, 2),
(3177, '2013-06-15', 1500, 0),
(3178, '2013-05-06', 3000, 5),
(3179, '2013-04-03', 4000, 15),
(3180, '2013-04-02', 3000, 14);

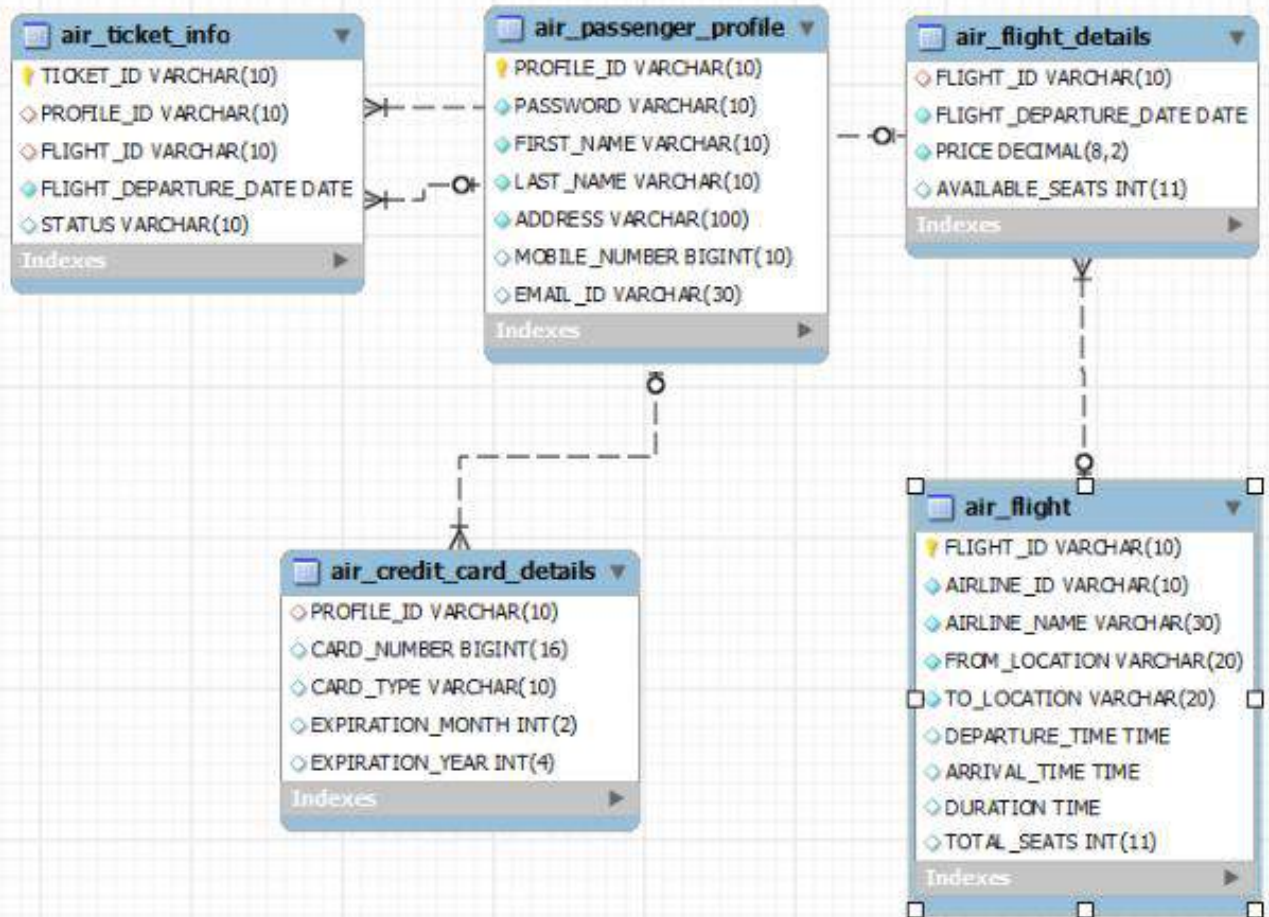
INSERT INTO air_ticket_info VALUES

(1, 1, 3178, '2013-05-06', 'delayed'),
(2, 5, 3179, '2013-04-03', 'on time'),
(2, 4, 3180, '2013-04-02', 'on time'),
(1, 2, 3177, '2013-06-15', 'on time'),
(1, 3, 3176, '2013-03-14', 'on time'),
(3, 1, 3171, '2013-03-15', 'on time'),
(4, 4, 3172, '2013-02-06', 'delayed'),

```
(5, 2, 3178, '2013-06-05', 'on time'),  
(4, 3, 3171, '2013-03-15', 'on time'),  
(5, 1, 3175, '2013-05-25', 'on time'),  
(6, 3, 3177, '2013-06-15', 'on time');
```

INSERT INTO air_passenger_profile VALUES

```
(1, 'godbless', 'John', 'Stuart', 'Street 21, Near Bus Stop-  
Hyderabad-432126', 9865263251, 'john@gmail.com'),  
(2, 'heyyaa', 'Robert', 'Clive', 'Sector 3, Technopolis-Kolkata-  
700102', 9733015875, 'robert@yahoo.com'),  
(3, 'hello123', 'Raj', 'Sharma', 'House No. 3, Anna Nagar-Kochi-  
452314', 9775470232, 'raj3452@hotmail.com'),  
(4, 'yesboss', 'Sanjay', 'Mittal', '21 Cauunaught Place-Delhi-  
144985', 9856856321, 'sanjay@yahoo.com'),  
(5, 'imhere', 'Tony', 'Stark', '51A, Greams Lane-  
Chennai-144587', 9832015785, 'tony@gmail.com');
```



AIR TICKET INFO

ticket_id	profile_id	flight_id	flight_departure_date	status
1	1	3178	2013-05-06	delayed
2	5	3179	2013-04-03	on time
2	4	3180	2013-04-02	on time
1	2	3177	2013-06-15	on time
1	3	3176	2013-03-14	on time
3	1	3171	2013-03-15	on time
4	4	3172	2013-02-06	delayed
5	2	3178	2013-06-05	on time
4	3	3171	2013-03-15	on time
5	1	3175	2013-05-25	on time
6	3	3177	2013-06-15	on time

AIR PASSENGER DETAILS

profile_id	password	first_name	last_name	address	mobile_number	email_id
1	godbless	John	Stuart	Street 21, Near Bus Stop-Hyderabad-432126	9865263251	john@gmail.com
2	heyyaa	Robert	Clive	Sector 3, Technopolis-Kolkata-700102	9733015875	robert@yahoo.com
3	hello123	Raj	Shama	House No. 3, Anna Nagar-Kochi-452314	9775470232	raj3452@hotmail...
4	yesboss	Sanjay	Mittal	21 Cauunaught Place-Delhi-144985	9856856321	sanjay@yahoo.c...
5	imhere	Tony	Stark	51A, Greams Lane-Chennai-144587	9832015785	tony@gmail.com

AIR FLIGHT DETAILS

flight_id	flight_departure_date	price	available_seats
3170	2013-02-14	1000.00	10
3171	2013-03-15	5000.00	0
3172	2013-02-05	3000.00	32
3173	2013-04-07	2000.00	12
3174	2013-04-05	3800.00	3
3175	2013-05-25	3500.00	10
3176	2013-03-14	8000.00	2
3177	2013-06-15	1500.00	0
3178	2013-05-06	3000.00	5
3179	2013-04-03	4000.00	15
3180	2013-04-02	3000.00	14

AIR CREDIT CARD DETAILS

profile_id	card_number	card_type	expiration_month	expiration_year
1	622098761234	debit	5	2013
2	652362563625	credit	1	2013
1	765432345678	credit	2	2013
3	654378561234	debit	6	2013
4	625417895623	debit	2	2013
5	865478956325	debit	3	2013
6	789563521457	credit	4	2013
2	543267895432	credit	8	2013
1	256369856321	debit	1	2013

AIR FLIGHT

flight_id	airline_id	airline_name	from_location	to_location	departure_time	arrival_time	duration	total_seats
3170	DT345	xyz	delhi	kolkata	21:00:00	22:30:00	01:30:00	100
3171	JT564	jkl	chennai	delhi	08:00:00	10:00:00	02:00:00	100
3172	AR342	fgh	kolkata	chennai	11:30:00	13:00:00	01:30:00	100
3173	MH370	abc	hyderabad	chennai	06:30:00	07:15:00	00:45:00	100
3174	MH321	xyz	kolkata	delhi	00:00:00	02:00:00	02:00:00	100
3175	MJ654	abc	chennai	hyderabad	15:00:00	16:00:00	01:00:00	200
3176	MH370	def	kochi	chennai	18:00:00	19:05:00	01:05:00	100
3177	MH45	fgh	delhi	kochi	19:00:00	21:00:00	02:00:00	200
3178	MH17	def	chennai	hyderabad	08:00:00	09:00:00	01:00:00	200
3179	JT435	abc	chennai	kolkata	14:00:00	15:00:00	01:00:00	100
3180	JT456	ijk	kolkata	kochi	05:00:00	05:45:00	00:45:00	200

QUERIES

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id,From_location,To_Location,Month Name as “Month_Name” and average price as “Average_Price”. Display the records sorted in ascending order based on flight id and then by Month Name.

```
SELECT f.flight_id,f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
AVG(price) Average_Price FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id WHERE f.airline_name = 'abc'
GROUP BY f.flight_id,f.from_location,f.to_location,Month_Name
ORDER BY f.flight_id, Month_Name;
```

flight_id	from_location	to_location	Month_Name	Average_Price
3173	hyderabad	chennai	April	2000.000000
3175	chennai	hyderabad	May	3500.000000
3179	chennai	kolkata	April	4000.000000

2. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as “Month_Name” and number of flight services as “No_of_Services”. Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight. The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name.

```
SELECT f.from_location,f.to_location,  
monthname(af.flight_departure_date) Month_Name,  
count(af.flight_departure_date) No_of_Services  
FROM air_flight f JOIN air_flight_details af  
ON f.flight_id = af.flight_id  
GROUP BY f.from_location,f.to_location,Month_Name  
ORDER BY f.from_location,f.to_Location,Month_Name;
```

from_location	to_location	Month_Name	No_of_Services
chennai	delhi	March	1
chennai	hyderabad	May	2
chennai	kolkata	April	1
delhi	kochi	June	1
delhi	kolkata	February	1
hyderabad	chennai	April	1
kochi	chennai	March	1
kolkata	chennai	February	1
kolkata	delhi	April	1
kolkata	kochi	April	1

3. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id,

customer's first_name, Address and Number of tickets booked as "No_of_Tickets".Display the records sorted in ascending order based on customer's first name.

```
SELECT      ap.profile_id,ap.first_name,ap.address,count(ati.ticket_id)
No_of_Tickets FROM
air_passenger_profile  ap      JOIN      air_ticket_info      ati      ON
ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id and af.airline_name='abc'
GROUP      BY      ap.profile_id,ap.first_name,ap.address      HAVING
count(ati.ticket_id)<=ALL
(SELECT count(ticket_id)
FROM air_ticket_info GROUP BY profile_id)
ORDER BY ap.first_name;
```

profile_id	first_name	address	No_of_Tickets
1	John	Street 21, Near Bus Stop-Hyderabad-432126	1
5	Tony	51A, Greams Lane-Chennai-144587	1

4. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as "No_of_Tickets".Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

```
SELECT
ap.profile_id,ap.first_name,ap.last_name,af.flight_id,ati.flight_departur
e_date,
count(ati.profile_id) No_of_Tickets FROM
```



```

air_ticket_info    ati    JOIN    air_passenger_profile    ap    ON
ap.profile_id=ati.profile_id

JOIN air_flight af ON af.flight_id=ati.flight_id

WHERE af.from_location='Chennai' and af.to_location='Hyderabad'

GROUP BY ati.flight_id,ati.profile_id

ORDER BY ap.profile_id,af.flight_id,ati.flight_departure_date;

```

profile_id	first_name	last_name	flight_id	flight_departure_date	No_of_Tickets
1	John	Stuart	3175	2013-05-25	1
1	John	Stuart	3178	2013-05-06	1
2	Robert	Clive	3178	2013-06-05	1

5. Write a query to display flight id,from location, to location and ticket price of flights whose departure is in the month of april.Display the records sorted in ascending order based on flight id and then by from location.

```

SELECT af.flight_id,af.from_location,af.to_location,afd.price FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
and month(afd.flight_departure_date)='04'

ORDER BY af.flight_id,af.from_location;

```

flight_id	from_location	to_location	price
3173	hyderabad	chennai	2000.00
3174	kolkata	delhi	3800.00
3179	chennai	kolkata	4000.00
3180	kolkata	kochi	3000.00

6. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight_id,

from_location, to_location and Average price as “Price”. Display the records sorted in ascending order based on flight id and then by from_location and then by to_location.

```
SELECT      af.flight_id,af.from_location,af.to_location,avg(afd.price)
Average_Price FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id
ORDER BY af.flight_id,af.from_location,af.to_location;
```

flight_id	from_location	to_location	Average_Price
3170	delhi	kolkata	1000.000000
3171	chennai	delhi	5000.000000
3172	kolkata	chennai	3000.000000
3173	hyderabad	chennai	2000.000000
3174	kolkata	delhi	3800.000000
3175	chennai	hyderabad	3500.000000
3176	kochi	chennai	8000.000000
3177	delhi	kochi	1500.000000
3178	chennai	hyderabad	3000.000000
3179	chennai	kolkata	4000.000000
3180	kolkata	kochi	3000.000000

7. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (combine first_name & last_name with comma in b/w), address of the customer. Give an alias to the name as customer_name. Hint: Query should fetch unique customers irrespective of multiple tickets booked. Display the records sorted in ascending order based on profile id.

```

SELECT          ap.profile_id,concat(ap.first_name,',',ap.last_name)
customer_name,ap.address FROM
air_passenger_profile    ap    JOIN    air_ticket_info    ati    ON
ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.profile_id
ORDER BY ap.profile_id;

```

profile_id	Customer_name	address
1	John,Stuart	Street 21, Near Bus Stop-Hyderabad-432126
2	Robert,Clive	Sector 3, Technopolis-Kolkata-700102

8. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets.In case of multiple records, display the records sorted in ascending order based on profile id.

```

SELECT profile_id FROM air_ticket_info
group by profile_id
having count(ticket_id)>=all(select count(ticket_id)
from air_ticket_info
group by profile_id) order by profile_id;

```

profile_id
1
3

9. Write a query to display the total number of tickets as “No_of_Tickets” booked in each flight in ABC Airlines. The Query

should display the flight_id, from_location, to_location and the number of tickets. Display only the flights in which atleast 1 ticket is booked. Display the records sorted in ascending order based on flight id.

```
SELECT f.flight_id,f.from_location,f.to_location,COUNT(t.ticket_id) AS  
No_of_Tickets  
FROM air_ticket_info t JOIN air_flight f  
ON f.flight_id = t.flight_id where AIRLINE_NAME = 'abc' GROUP by  
f.flight_id,f.from_location,f.to_location  
having count(t.ticket_id)>=1  
ORDER by f.flight_id;
```

flight_id	from_location	to_location	No_of_Tickets
3175	chennai	hyderabad	1
3179	chennai	kolkata	1

10. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as “No_of_Services” and the cost as “Total_Price” in the same order. Order the result by Total Price in descending order and then by flight_id in descending order. Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

```
SELECT flight_id, count(flight_departure_date)  
No_of_services, sum(price) Total_Price FROM  
air_flight_details GROUP BY flight_id  
ORDER BY Total_price DESC, flight_id DESC;
```

flight_id	No_of_services	Total_Price
3176	1	8000.00
3171	1	5000.00
3179	1	4000.00
3174	1	3800.00
3175	1	3500.00
3180	1	3000.00
3178	1	3000.00
3172	1	3000.00
3173	1	2000.00
3177	1	1500.00
3170	1	1000.00

11. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as “No_of_Passengers” in the same order. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT flight_id, flight_departure_date, count(ticket_id)
No_of_passengers FROM
air_ticket_info GROUP BY flight_id, flight_departure_date
ORDER BY flight_id, flight_departure_date;
```

flight_id	flight_departure_date	No_of_passengers
3171	2013-03-15	2
3172	2013-02-06	1
3175	2013-05-25	1
3176	2013-03-14	1
3177	2013-06-15	2
3178	2013-05-06	1
3178	2013-06-05	1
3179	2013-04-03	1
3180	2013-04-02	1

12. Write a query to display profile id of passenger(s) who booked minimum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.

```
SELECT profile_id FROM air_ticket_info
GROUP BY profile_id HAVING count(ticket_id)<=ALL
(SELECT count(ticket_id) FROM air_ticket_info GROUP BY profile_id)
ORDER BY profile_id;
```

profile_id
5

13. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI. Display the records sorted in ascending order based on profile id.

```
SELECT DISTINCT
ap.profile_id,ap.first_name,ap.mobile_number,ap.email_id FROM
air_passenger_profile    ap    JOIN    air_ticket_info    ati    ON
ap.profile_id=ati.profile_id
JOIN air_flight af ON ati.flight_id=af.flight_id
WHERE af.from_location='Hyderabad' and af.to_location='Chennai'
ORDER BY profile_id;
```

profile_id	first_name	mobile_number	email_id
------------	------------	---------------	----------

14. Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger's profile_id, first_name,last_name, flight_id, flight_departure_date, actual departure time , actual arrival time , delayed departure time as "Delayed_Departure_Time", delayed arrival time as "Delayed_Arrival_Time" Hint: Distinct Profile ID should be displayed irrespective of multiple tickets booked by the same profile.Display the records sorted in ascending order based on passenger's profile id.

```
SELECT DISTINCT
ap.profile_id,ap.first_name,ap.last_name,ati.flight_id,ati.flight_departu
re_date,
af.departure_time,af.arrival_time,
addtime(af.departure_time,'01:00:00') Delayed_Deaparuture_Time,
addtime(af.arrival_time,'01:00:00') Delayed_Arrival_Time FROM
air_passenger_profile ap JOIN air_ticket_info ati ON
ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
and ati.flight_departure_date='2013-05-06'
ORDER BY profile_id;
```

profile_id	first_name	last_name	flight_id	flight_departure_date	departure_time	arrival_time	Delayed_Departure_Time	Delayed_Arrival_Time
1	John	Stuart	3178	2013-05-06	08:00:00	09:00:00	09:00:00	10:00:00

15. Write a query to display the number of tickets as “No_of_Tickets” booked by Kochi Customers. The Query should display the Profile_Id, First_Name, Base_Location and number of tickets booked. Hint: Use String functions to get the base location of customer from their Address and give alias name as “Base_Location” Display the records sorted in ascending order based on customer first name.

```
SELECT ap.profile_id,ap.first_name,
substring_index(substring_index(ap.address,'-',2),'-',-1) Base_Location,
count(ati.ticket_id) No_of_Tickets
FROM
air_passenger_profile    ap    JOIN    air_ticket_info    ati    ON
ati.profile_id=ap.profile_id
WHERE ap.address LIKE '%Kochi%'
ORDER BY ap.first_name;
```

profile_id	first_name	Base_Location	No_of_Tickets
3	Raj	Kochi	3

16. Write a query to display the flight_id, from_location, to_location, number of Services as “No_of_Services” offered in the month of May.

```
SELECT
af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_
date) No_of_services
FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
```


WHERE month(flight_departure_date)='05'

GROUP BY af.flight_id,af.from_location,af.to_location

ORDER BY af.flight_id;

flight_id	from_location	to_location	No_of_services
3175	chennai	hyderabad	1
3178	chennai	hyderabad	1

17. Write a query to display profile id,last name,mobile number and email id of passengers whose base location is chennai.Display the records sorted in ascending order based on profile id.

SELECT profile_id, last_name, mobile_number, email_id

FROM air_passenger_profile

WHERE address LIKE '%Chennai%'

ORDER BY profile_id;

profile_id	last_name	mobile_number	email_id
5	Stark	9832015785	tony@gmail.com

18. Write a query to display number of flights between 6.00 AM and 6.00 PM from chennai. Hint Use FLIGHT_COUNT as alias name.

SELECT count(flight_id) FLIGHT_COUNT FROM air_flight

WHERE from_location='CHENNAI'

AND departure_time BETWEEN '06:00:00' AND '18:00:00';

FLIGHT_COUNT
4

19. Write a query to display unique profile id,first name , email id and contact number of passenger(s) who travelled on flight with id 3178. Display the records sorted in ascending order based on first name.

```

SELECT                                                    DISTINCT
ap.profile_id,ap.first_name,ap.email_id,ap.mobile_number FROM
air_passenger_profile    ap    JOIN    air_ticket_info    ati    ON
ap.profile_id=ati.profile_id
WHERE ati.flight_id='3178'
ORDER BY ap.first_name;

```

profile_id	first_name	email_id	mobile_number
1	John	john@gmail.com	9865263251
2	Robert	robert@yahoo.com	9733015875

20. Write a query to display flight id,departure date,flight type of all flights. Flight type can be identified based on the following rules : if ticket price is less than 3000 then 'AIR PASSENGER',ticket price between 3000 and less than 4000 'AIR BUS' and ticket price between 4000 and greater than 4000 then 'EXECUTIVE PASSENGER'. Hint use FLIGHT_TYPE as alias name.Display the records sorted in ascendeing order based on flight_id and then by departure date.

```

SELECT flight_id,flight_departure_date,
case when price<3000 then 'AIR PASSENGER'
      when price>=3000 and price<4000 then 'AIR BUS'
      when price>=4000 then 'EXECUTIVE PASSENGER'
end FLIGHT_TYPE FROM air_flight_details
ORDER BY flight_id,flight_departure_date;

```

flight_id	flight_departure_date	FLIGHT_TYPE
3170	2013-02-14	AIR PASSENGER
3171	2013-03-15	EXECUTIVE PASSENGER
3172	2013-02-05	AIR BUS
3173	2013-04-07	AIR PASSENGER
3174	2013-04-05	AIR BUS
3175	2013-05-25	AIR BUS
3176	2013-03-14	EXECUTIVE PASSENGER
3177	2013-06-15	AIR PASSENGER
3178	2013-05-06	AIR BUS
3179	2013-04-03	EXECUTIVE PASSENGER
3180	2013-04-02	AIR BUS

21. Write a query to display the credit card type and no of credit cards used on the same type. Display the records sorted in ascending order based on credit card type.Hint: Use CARD_COUNT AS Alias name for no of cards.

```
SELECT    card_type,      count(card_type)    Card_Count    FROM
air_credit_card_details

GROUP BY card_type ORDER BY card_type;
```

card_type	Card_Count
credit	4
debit	5

22. Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com. The Serial No will be the last three digits of profile ID. Hint: Use SERIAL_NO as Alias name for serial number. Display the records sorted in ascending order based on name.

```
SELECT                                     substring(profile_id,-3)
SERIAL_NO,first_name,mobile_number,email_id FROM
air_passenger_profile
```

WHERE email_id LIKE '%@gmail.com'

ORDER BY first_name;

SERIAL_NO	first_name	mobile_number	email_id
	John	9865263251	john@gmail.com
	Tony	9832015785	tony@gmail.com

23. Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight_id, from_location, to_location, least number of Services as “No_of_Services” Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight if there are multiple flights, display them sorted in ascending order based on flight id.

```
SELECT afd.flight_id,af.from_location,af.to_location,count(afd.flight_id)
No_of_Services
```

```
FROM    air_flight_details    afd    JOIN    air_flight    af    ON
af.flight_id=afd.flight_id
```

```
WHERE monthname(afd.flight_departure_date)='May'
```

```
GROUP BY afd.flight_departure_date HAVING count(afd.flight_id) <=
```

```
ALL(SELECT count(flight_id) FROM air_flight_details
```

```
WHERE monthname(flight_departure_date)='May'
```

```
GROUP BY flight_departure_date)
```

```
ORDER BY flight_id;
```

flight_id	from_location	to_location	No_of_Services
3175	chennai	hyderabad	1
3178	chennai	hyderabad	1

24. Write a query to display the flights available in Morning, AfterNoon, Evening& Night. The Query should display the Flight_Id, From_Location, To_Location , Departure_Time, time of service as "Time_of_Service". Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning, 12:00:01 to 18:00:00 Hrs - AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - NightDisplay the records sorted in ascending order based on flight id.

```
SELECT flight_id,from_location,to_location,Departure_Time,  
CASE  
WHEN departure_time BETWEEN ('05:00:01') AND ('12:00:00')  
THEN 'Morning'  
WHEN departure_time BETWEEN ('12:00:01') AND ('18:00:00')  
THEN 'AfterNoon'  
WHEN departure_time BETWEEN ('18:00:01') AND ('24:00:00')  
THEN 'Evening'  
WHEN departure_time='00:00:00'  
THEN 'Evening'  
WHEN departure_time BETWEEN ('00:00:01') AND ('05:00:00')  
THEN 'Night'  
END Time_of_Service  
FROM air_flight
```

order by flight_id;

flight_id	from_location	to_location	Departure_Time	Time_of_Service
3170	delhi	kolkata	21:00:00	Evening
3171	chennai	delhi	08:00:00	Moming
3172	kolkata	chennai	11:30:00	Moming
3173	hyderabad	chennai	06:30:00	Moming
3174	kolkata	delhi	00:00:00	Evening
3175	chennai	hyderabad	15:00:00	AfterNoon
3176	kochi	chennai	18:00:00	AfterNoon
3177	delhi	kochi	19:00:00	Evening
3178	chennai	hyderabad	08:00:00	Moming
3179	chennai	kolkata	14:00:00	AfterNoon
3180	kolkata	kochi	05:00:00	Night

25. Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No_of_Flights”. Hint: Get the distinct from location and to location. Display the records sorted in ascending order based on from location.

```
SELECT from_location, count(flight_id) No_of_Flights FROM  
air_flight GROUP BY from_location  
ORDER BY from_location;
```

from_location	No_of_Flights
chennai	4
delhi	2
hyderabad	1
kochi	1
kolkata	3

26. Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight_id, from_location, To_location, flight_departure_date and the number of passengers as “No_of_Passengers”. Hint: The Number of

passengers inclusive of all the tickets booked with single profile id. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT
af.flight_id,af.from_location,af.to_location,ati.flight_departure_date,
count(ati.ticket_id) No_of_Passengers FROM
air_flight af JOIN air_ticket_info ati ON af.flight_id=ati.flight_id
GROUP BY
af.flight_id,af.from_location,af.to_location,ati.flight_departure_date
ORDER BY af.flight_id,ati.flight_departure_date;
```

flight_id	from_location	to_location	flight_departure_date	No_of_Passengers
3171	chennai	delhi	2013-03-15	2
3172	kolkata	chennai	2013-02-06	1
3175	chennai	hyderabad	2013-05-25	1
3176	kochi	chennai	2013-03-14	1
3177	delhi	kochi	2013-06-15	2
3178	chennai	hyderabad	2013-05-06	1
3178	chennai	hyderabad	2013-06-05	1
3179	chennai	kolkata	2013-04-03	1
3180	kolkata	kochi	2013-04-02	1

27. Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight_Id, From_Location, To_Location, Total_Seats, seats booked as "No_of_Seats_Booked". Display the records sorted in ascending order based on flight id and then by No_of_Seats_Booked.

```
SELECT af.flight_id,af.from_location,af.to_location,af.total_seats,
(af.total_seats-afd.available_seats) No_of_Seats_Booked FROM
air_flight_details afd JOIN air_flight af ON afd.flight_id=af.flight_id
```

WHERE (af.total_seats-afd.available_seats)>(af.total_seats*0.1)

ORDER BY flight_id,No_of_Seats_Booked;

flight_id	from_location	to_location	total_seats	No_of_Seats_Booked
3170	delhi	kolkata	100	90
3171	chennai	delhi	100	100
3172	kolkata	chennai	100	68
3173	hyderabad	chennai	100	88
3174	kolkata	delhi	100	97
3175	chennai	hyderabad	200	190
3176	kochi	chennai	100	98
3177	delhi	kochi	200	200
3178	chennai	hyderabad	200	195
3179	chennai	kolkata	100	85
3180	kolkata	kochi	200	186

28. Write a query to display the Flight_Id, Flight_Departure_Date, From_Location,To_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

SELECT

af.flight_Id,afd.flight_Departure_Date,af.From_Location,af.To_Location
,af.duration

FROM air_flight af JOIN air_flight_details afd ON
af.flight_id=afd.flight_id

WHERE af.duration<'01:10:00';

flight_Id	flight_Departure_Date	From_Location	To_Location	duration
3173	2013-04-07	hyderabad	chennai	00:45:00
3175	2013-05-25	chennai	hyderabad	01:00:00
3176	2013-03-14	kochi	chennai	01:05:00
3178	2013-05-06	chennai	hyderabad	01:00:00
3179	2013-04-03	chennai	kolkata	01:00:00
3180	2013-04-02	kolkata	kochi	00:45:00

29. Write a query to display the flight_id, from_location,to_location,number of services as “No_of_Services” , average ticket price as “Average_Price” whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

```
SELECT afd.flight_id,af.from_location,af.to_location,  
count(afd.flight_departure_date)      No_of_Service,      avg(price)  
Average_Price  
FROM    air_flight    af    JOIN    air_flight_details    afd    ON  
af.flight_id=afd.flight_id  
GROUP BY af.flight_id,af.from_location,af.to_location  
HAVING avg(price)>(SELECT avg(price) FROM air_flight_details)  
ORDER BY average_price;
```

flight_id	from_location	to_location	No_of_Service	Average_Price
3175	chennai	hyderabad	1	3500.000000
3174	kolkata	delhi	1	3800.000000
3179	chennai	kolkata	1	4000.000000
3171	chennai	delhi	1	5000.000000
3176	kochi	chennai	1	8000.000000

MOVIE

CREATE DATABASE video;USE video;

Create table CUSTOMER_MASTER

(CUSTOMER_ID Varchar(10),CUSTOMER_NAME Varchar(30) NOT NULL,CONTACT_NO BIGINT(10),CONTACT_ADD Varchar(20),DATE_OF_REGISTRATION Date NOT NULL,AGE Varchar(15)NOT NULL,Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID));

Create table LIBRARY_CARD_MASTER

(CARD_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT BIGINT(50),NUMBER_OF_YEARS bigint(10) NOT NULL,Constraint MT_cts2 PRIMARY KEY(CARD_ID));

Create table MOVIES_MASTER

(MOVIE_ID Varchar(10), MOVIE_NAME Varchar(50) NOT NULL,RELEASE_DATE Varchar(30) NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION VARCHAR(10) NOT NULL, MOVIE_TYPE Varchar(3),MOVIE_CATEGORY VARCHAR(20) NOT NULL,DIRECTOR VARCHAR(20) NOT NULL, LEAD_ROLE_1 Varchar(3) NOT NULL,LEAD_ROLE_2 VARCHAR(4) NOT NULL,RENT_COST BIGINT(10),Constraint MT_cts4 PRIMARY KEY(MOVIE_ID));

Create table CUSTOMER_CARD_DETAILS

(CUSTOMER_ID Varchar(10),CARD_ID VARCHAR(10),ISSUE_DATE DATE NOT NULL,Constraint MT_cts3 PRIMARY KEY(CUSTOMER_ID),Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID) References CUSTOMER_MASTER(CUSTOMER_ID),Constraint MT_CTS42 FOREIGN KEY(CARD_ID) References LIBRARY_CARD_MASTER(CARD_ID));

Create table CUSTOMER_ISSUE_DETAILS

(ISSUE_ID Varchar(10) NOT NULL,CUSTOMER_ID Varchar(10) NOT NULL,MOVIE_ID VARCHAR(10), ISSUE_DATE Date NOT NULL,RETURN_DATE Date NOT NULL,

ACTUAL_DATE_RETURN Date NOT NULL,Constraint MT_cts5 PRIMARY KEY(ISSUE_ID),Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References CUSTOMER_MASTER(CUSTOMER_ID), Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID) References MOVIES_MASTER(MOVIE_ID));

Insert into CUSTOMER_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS007', 'NAME7',
3210987654,'ADD7', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS008', 'NAME8',
2109876543,'ADD8', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS009', 'NAME9',
NULL,'ADD9', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS010', 'NAM10',
9934567890,'ADD10', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS011', 'NAM11',
9875678910,'ADD11', '2013-02-12', '21');

Insert into LIBRARY_CARD_MASTER Values('CR001', 'Silver', 200, 5);

Insert into LIBRARY_CARD_MASTER Values('CR002', 'Gold', 400, 9);

Insert into LIBRARY_CARD_MASTER Values('CR003', 'Platinum', 600, 8);

Insert into LIBRARY_CARD_MASTER Values('CR004', 'VISA', 800, 7);

Insert into LIBRARY_CARD_MASTER Values('CR005', 'CREDIT', 1200, 6);

Insert into MOVIES_MASTER Values('MV001', 'DIEHARD', '2012-05-
13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR1','L1','L2',100);

Insert into MOVIES_MASTER Values('MV002', 'THE MATRIX', '2012-05-
13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR2','L1','L2',100);

Insert into MOVIES_MASTER Values('MV003', 'INCEPTION', '2012-05-
13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR3','L15','L2',100);

Insert into MOVIES_MASTER Values('MV004', 'DARK KNIGHT', '2012-05-
13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR4','L15','L2',100);

Insert into MOVIES_MASTER Values('MV005', 'OFFICE S', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR5','L12','L24',100);

Insert into MOVIES_MASTER Values('MV006', 'SHAWN OF DEAD', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR6','L1','L25',100);

Insert into MOVIES_MASTER Values('MV007', 'YOUNG FRANKEN', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR7','L1','L2',100);

Insert into MOVIES_MASTER Values('MV008', 'CAS', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR8','L1','L2',100);

Insert into MOVIES_MASTER Values('MV009', 'GWW', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR9','L1','L2',100);

Insert into MOVIES_MASTER Values('MV010', 'TITANIC', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR10','L1','L2',100);

Insert into MOVIES_MASTER Values('MV011', 'THE NOTE BOOK', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR11','L1','L2',100);

Insert into CUSTOMER_CARD_DETAILS Values('CUS001', 'CR001', '2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS002', 'CR002', '2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS003', 'CR002', '2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS004', 'CR003', '2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS005', 'CR003', '2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS001', 'CUS001', 'MV001', '2012-05-13', '2012-05-13','2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS002', 'CUS001', 'MV001', '2012-05-01', '2012-05-16','2012-05-16');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS003', 'CUS002', 'MV004', '2012-05-02', '2012-05-06','2012-05-16');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS004', 'CUS002', 'MV004', '2012-04-03', '2012-04-16','2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS005', 'CUS002', 'MV009', '2012-04-04', '2012-04-16','2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS006', 'CUS003', 'MV002', '2012-03-30', '2012-04-15','2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS007', 'CUS003', 'MV003', '2012-04-20', '2012-05-05','2012-05-05');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS008', 'CUS003', 'MV005', '2012-04-21', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS009', 'CUS003', 'MV001', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS010', 'CUS003', 'MV009', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS011', 'CUS003', 'MV010', '2012-04-23', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS012', 'CUS003', 'MV010', '2012-04-24', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS013', 'CUS003',
'MV008', '2012-04-25', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS014', 'CUS004',
'MV007', '2012-04-26', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS015', 'CUS004',
'MV006', '2012-04-27', '2012-05-07','2012-05-25');

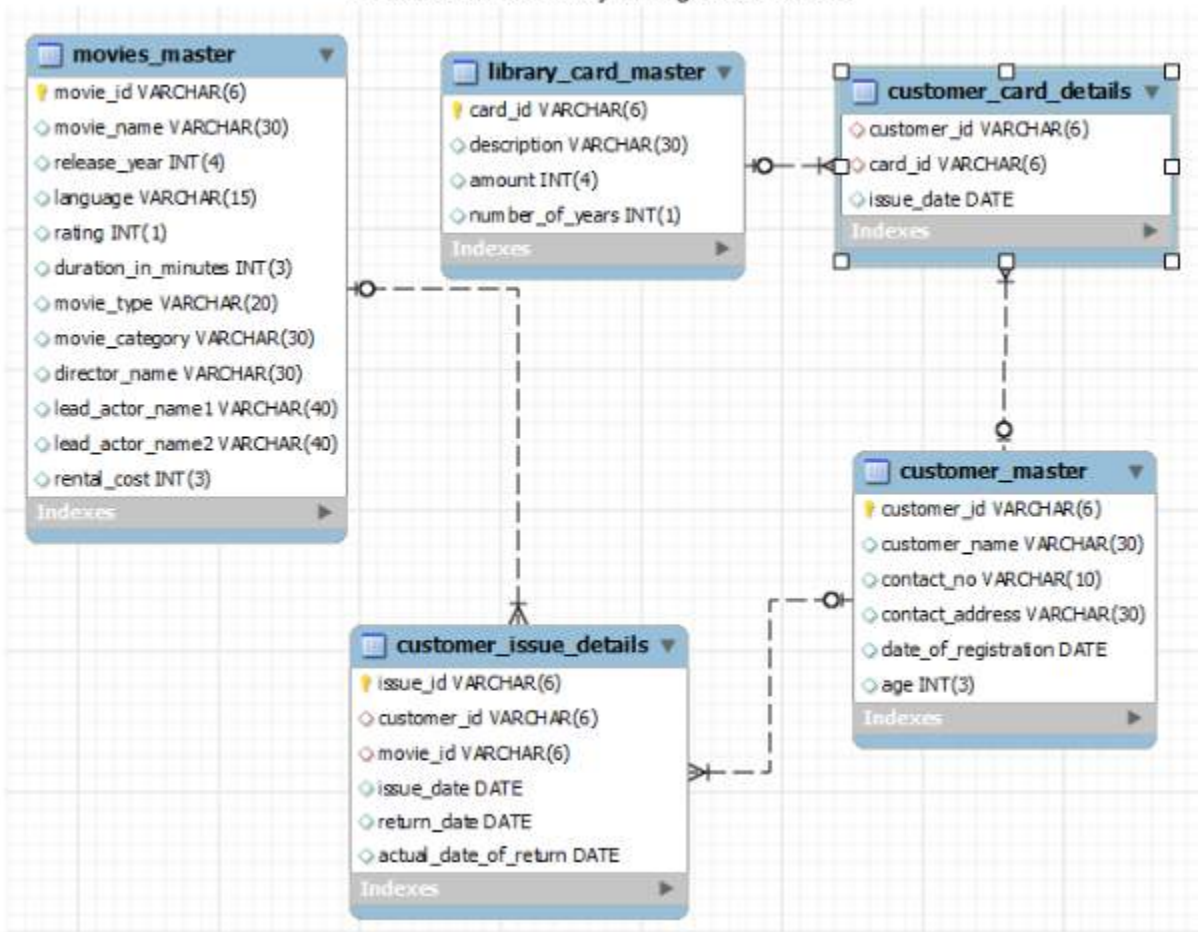
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS016', 'CUS004',
'MV006', '2012-04-28', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS017', 'CUS004',
'MV001', '2012-04-29', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS018', 'CUS010',
'MV008', '2012-04-24', '2012-05-07','2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS019', 'CUS011',
'MV009', '2012-04-27', '2012-05-07','2012-05-25');

ANSI SQL Video Library Management Schema



MOVIE MASTER

[illegible]

LEAD_ROLE_2	RENT_COST
L2	100
L2	100
L2	100
L2	100
L24	100
L25	100
L2	100
L2	100
L2	100
L2	100
L2	100
NULL	NULL

CUSTOMER MASTER

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADD	DATE_OF_REGISTRATION	AGE
CUS001	AMIT	9876543210	ADD1	2012-02-12	21
CUS002	ABDHUL	8765432109	ADD2	2012-02-12	21
CUS003	GAYAN	7654321098	ADD3	2012-02-12	21
CUS004	RADHA	6543210987	ADD4	2012-02-12	21
CUS005	GURU	NULL	ADD5	2012-02-12	21
CUS006	MOHAN	4321098765	ADD6	2012-02-12	21
CUS007	NAME7	3210987654	ADD7	2012-02-12	21
CUS008	NAME8	2109876543	ADD8	2013-02-12	21
CUS009	NAME9	NULL	ADD9	2013-02-12	21
CUS010	NAM10	9934567890	ADD10	2013-02-12	21
CUS011	NAM11	9875678910	ADD11	2013-02-12	21
NULL	NULL	NULL	NULL	NULL	NULL

LIBRARY CARD MASTER

CARD_ID	DESCRIPTION	AMOUNT	NUMBER_OF_YEARS
CR001	Silver	200	5
CR002	Gold	400	9
CR003	Platinum	600	8
CR004	VISA	800	7
CR005	CREDIT	1200	6
NULL	NULL	NULL	NULL

CUSTOMER CARD DETAILS

CUSTOMER_ID	CARD_ID	ISSUE_DATE
CUS001	CR001	2012-05-13
CUS002	CR002	2012-05-13
CUS003	CR002	2013-05-13
CUS004	CR003	2013-05-13
CUS005	CR003	2012-05-13
NULL	NULL	NULL

CUSTOMER ISSUE DETAILS

ISSUE_ID	CUSTOMER_ID	MOVIE_ID	ISSUE_DATE	RETURN_DATE	ACTUAL_DATE_RETURN
IS001	CUS001	MV001	2012-05-13	2012-05-13	2012-05-13
IS002	CUS001	MV001	2012-05-01	2012-05-16	2012-05-16
IS003	CUS002	MV004	2012-05-02	2012-05-06	2012-05-16
IS004	CUS002	MV004	2012-04-03	2012-04-16	2012-04-20
IS005	CUS002	MV009	2012-04-04	2012-04-16	2012-04-20
IS006	CUS003	MV002	2012-03-30	2012-04-15	2012-04-20
IS007	CUS003	MV003	2012-04-20	2012-05-05	2012-05-05
IS008	CUS003	MV005	2012-04-21	2012-05-07	2012-05-25
IS009	CUS003	MV001	2012-04-22	2012-05-07	2012-05-25
IS010	CUS003	MV009	2012-04-22	2012-05-07	2012-05-25
IS011	CUS003	MV010	2012-04-23	2012-05-07	2012-05-25
IS012	CUS003	MV010	2012-04-24	2012-05-07	2012-05-25
IS013	CUS003	MV008	2012-04-25	2012-05-07	2012-05-25
IS014	CUS004	MV007	2012-04-26	2012-05-07	2012-05-25
IS015	CUS004	MV006	2012-04-27	2012-05-07	2012-05-25
IS016	CUS004	MV006	2012-04-28	2012-05-07	2012-05-25
IS017	CUS004	MV001	2012-04-29	2012-05-07	2012-05-25
IS018	CUS010	MV008	2012-04-24	2012-05-07	2012-05-25
IS019	CUS011	MV009	2012-04-27	2012-05-07	2012-05-25
NULL	NULL	NULL	NULL	NULL	NULL

1. Write a query to display movie names and number of times that movie is issued to customers. In case movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order). The Alias name for the number of movies issued is ISSUE_COUNT.

```

SELECT m.MOVIE_NAME,count(ISSUE_ID) ISSUE_COUNT FROM
movies_master m LEFT JOIN customer_issue_details c ON
m.MOVIE_ID=c.MOVIE_ID

GROUP BY m.movie_name

ORDER BY ISSUE_COUNT DESC,MOVIE_NAME;

```

MOVIE_NAME	ISSUE_COUNT
DIEHARD	4
GWW	3
CAS	2
DARK KNIGHT	2
SHAWN OF DEAD	2
TITANIC	2
INCEPTION	1
OFFICE S	1
THE MATRIX	1
YOUNG FRANKEN	1
THE NOTE BOOK	0

2. Write a query to display id,name,age,contact no of customers whose age is greater than 25 and and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

```

SELECT CUSTOMER_ID,CUSTOMER_NAME,AGE,ifnull(
concat('+91-',substring(contact_no,1,3),'-',
substring(contact_no,4,3),'-',substring(contact_no,7)), 'N/A')
CONTACT_ISD

FROM customer_master WHERE age>25 and
year(date_of_registration)='2012'

```

ORDER BY age,CUSTOMER_NAME;

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
-------------	---------------	-----	-------------

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO_OF_MOVIES as alias name for number of movies.

```
SELECT MOVIE_CATEGORY,count(MOVIE_ID) NO_OF_MOVIES FROM
movies_master GROUP BY MOVIE_CATEGORY
ORDER BY NO_OF_MOVIES DESC,MOVIE_CATEGORY;
```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Write a query to display the number of customers having card with description "Gold card". Hint: Use CUSTOMER_COUNT as alias name for number of customers

```
SELECT count(c.customer_id) CUSTOMER_COUNT FROM
library_card_master l JOIN customer_card_details c ON
l.CARD_ID=c.CARD_ID
WHERE description='Gold';
```

CUSTOMER_COUNT
2

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who

hold library card. Display the records sorted by customer name in descending order. Use REGISTERED_YEAR as alias name for year of registration.

```
SELECT c.customer_id,c.customer_name,  
year(c.DATE_OF_REGISTRATION)  
REGISTERED_YEAR,cd.card_id,cd.issue_date FROM  
customer_master c JOIN customer_card_details cd ON  
c.customer_id=cd.customer_id  
ORDER BY CUSTOMER_NAME DESC;
```

customer_id	customer_name	REGISTERED_YEAR	card_id	issue_date
CUS004	RADHA	2012	CR003	2013-05-13
CUS005	GURU	2012	CR003	2012-05-13
CUS003	GAYAN	2012	CR002	2013-05-13
CUS001	AMIT	2012	CR001	2012-05-13
CUS002	ABDHUL	2012	CR002	2012-05-13

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order by customer name.

```
SELECT ci.issue_id,ci.CUSTOMER_ID,c.CUSTOMER_NAME FROM  
customer_master c JOIN customer_issue_details ci ON  
c.customer_id=ci.customer_id  
WHERE customer_name LIKE 'R%' and  
ci.actual_date_return>ci.return_date  
ORDER BY customer_name;
```

issue_id	CUSTOMER_ID	CUSTOMER_NAME
IS014	CUS004	RADHA
IS015	CUS004	RADHA
IS016	CUS004	RADHA
IS017	CUS004	RADHA

7. Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10. Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name.

```
SELECT c.CUSTOMER_ID,c.CUSTOMER_NAME,l.card_id,l.DESCRPTION,
concat('$',round(amount/52.42)) AMOUNT_DOLLAR FROM
customer_master c JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id
JOIN customer_card_details cc ON cc.customer_id=c.customer_id
JOIN library_card_master l ON cc.card_id=l.card_id
WHERE c.DATE_OF_REGISTRATION=ci.issue_date
ORDER BY customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	card_id	DESCRIPTION	AMOUNT_DOLLAR
-------------	---------------	---------	-------------	---------------

8. Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from

**library without library card and whose address ends with 'Nagar'.
Display customer name in upper case. Hint: Use CUSTOMER_NAME as
alias name for customer name. Display the details sorted in ascending
order based on customer name.**

```
SELECT CUSTOMER_ID,upper(CUSTOMER_NAME)  
CUSTOMER_NAME,contact_no,contact_add FROM  
customer_master WHERE contact_add LIKE '%Nagar' and  
customer_id NOT IN (SELECT customer_id FROM  
customer_card_details)  
and customer_id IN (SELECT customer_id FROM  
customer_issue_details)  
ORDER BY CUSTOMER_NAME;
```

CUSTOMER_ID	CUSTOMER_NAME	contact_no	contact_add
-------------	---------------	------------	-------------

**9. Write a query to display the movie id, movie name, release
year, director name of movies acted by the leadactor1 who acted
maximum number of movies .Display the records sorted in ascending
order based on movie name.**

```
SELECT movie_id,movie_name,release_date,director FROM  
movies_master  
WHERE lead_role_1 IN(SELECT lead_role_1 FROM  
(SELECT lead_role_1,count(movie_id)ct FROM movies_master  
GROUP BY lead_role_1)t WHERE t.ct>=ALL(SELECT count(movie_id)  
FROM movies_master GROUP BY lead_role_1)) ORDER BY  
movie_name;
```

movie_id	movie_name	release_date	director
MV008	CAS	2012-05-13	DIR8
MV001	DIEHARD	2012-05-13	DIR1
MV009	GWW	2012-05-13	DIR9
MV006	SHAWN OF DEAD	2012-05-13	DIR6
MV002	THE MATRIX	2012-05-13	DIR2
MV011	THE NOTE BOOK	2012-05-13	DIR11
MV010	TITANIC	2012-05-13	DIR10
MV007	YOUNG FRANK...	2012-05-13	DIR7

10. Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.
**
Hint: Use MOVIE_COUNT as alias name for number of movies issued.**

```
SELECT c.customer_name, count(ci.movie_id) MOVIE_COUNT FROM
customer_master c LEFT JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id

GROUP BY c.customer_id ORDER BY c.customer_name;
```

customer_name	MOVIE_COUNT
ABDHUL	3
AMIT	2
GAYAN	8
GURU	0
MOHAN	0
NAM10	1
NAM11	1
NAME7	0
NAME8	0
NAME9	0
RADHA	4

11. Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number. Serial number can be generated from the issue id , that is last two characters of issue id is the serial number. For Example Assume the issue id is I00005 then the serial number is 05 Hint: Alias name for serial number is 'SERIAL_NO'

```
SELECT substring(ci.issue_id,-2)
SERIAL_NO,ci.issue_id,c.customer_id,c.customer_name,
m.movie_id,m.movie_name FROM customer_master c JOIN
customer_issue_details ci
ON c.customer_id=ci.customer_id JOIN movies_master m ON
m.movie_id=ci.movie_id
ORDER BY SERIAL_NO;
```

SERIAL_NO	issue_id	customer_id	customer_name	movie_id	movie_name
01	IS001	CUS001	AMIT	MV001	DIEHARD
02	IS002	CUS001	AMIT	MV001	DIEHARD
03	IS003	CUS002	ABDHUL	MV004	DARK KNIGHT
04	IS004	CUS002	ABDHUL	MV004	DARK KNIGHT
05	IS005	CUS002	ABDHUL	MV009	GWW
06	IS006	CUS003	GAYAN	MV002	THE MATRIX
07	IS007	CUS003	GAYAN	MV003	INCEPTION
08	IS008	CUS003	GAYAN	MV005	OFFICE S
09	IS009	CUS003	GAYAN	MV001	DIEHARD
10	IS010	CUS003	GAYAN	MV009	GWW
11	IS011	CUS003	GAYAN	MV010	TITANIC
12	IS012	CUS003	GAYAN	MV010	TITANIC
13	IS013	CUS003	GAYAN	MV008	CAS
14	IS014	CUS004	RADHA	MV007	YOUNG FRAN...
15	IS015	CUS004	RADHA	MV006	SHAWN OF D...
16	IS016	CUS004	RADHA	MV006	SHAWN OF D...
17	IS017	CUS004	RADHA	MV001	DIEHARD
18	IS018	CUS010	NAM10	MV008	CAS
19	IS019	CUS011	NAM11	MV009	GWW

12. Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

```

SELECT
ci.issue_id,ci.issue_date,c.customer_id,c.customer_name,c.contact_no
FROM
customer_master c JOIN customer_issue_details ci ON
c.customer_id=ci.customer_id
and year(ci.issue_date)='2013' ORDER BY ci.issue_date DESC;

```

issue_id	issue_date	customer_id	customer_name	contact_no
----------	------------	-------------	---------------	------------

**13. Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format. LEAD_ACTOR_ONE space ambersant space LEAD_ACTOR_TWO. Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria" Hint: Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.**

```
SELECT movie_id, movie_name, concat(lead_role_1, ' & ', lead_role_2)
ACTOR FROM movies_master
```

```
WHERE movie_id NOT IN (SELECT movie_id FROM
customer_issue_details) ORDER BY movie_name;
```

movie_id	movie_name	ACTOR
MV011	THE NOTE BOOK	L1 & L2

14. Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

```
SELECT upper(director) DIRECTOR_NAME, movie_name, lead_role_1
FROM movies_master
```

GROUP BY director HAVING count(movie_id)>1 ORDER BY
director,movie_name DESC;

DIRECTOR_NAME	movie_name	lead_role_1
---------------	------------	-------------

**15. Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
 Hint: Use NO_OF_CUSTOMERS as alias name for number of customers.**

```
SELECT count(customer_id) NO_OF_CUSTOMER FROM  
customer_master
```

```
WHERE contact_no is not null and year(date_of_registration)='2012';
```

NO_OF_CUSTOMER
6

16. Write a query to display the customer's name, contact number, library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.

```
SELECT c.customer_name,ifnull(c.contact_no,c.contact_add)  
CONTACT_DETAILS,l.card_id,l.description FROM
```

```
customer_master c LEFT JOIN customer_card_details cc ON  
c.customer_id=cc.customer_id
```

```
LEFT JOIN library_card_master l ON l.card_id=cc.card_id
```

```
ORDER BY customer_name;
```

customer_name	CONTACT_DETAILS	card_id	description
ABDHUL	8765432109	CR002	Gold
AMIT	9876543210	CR001	Silver
GAYAN	7654321098	CR002	Gold
GURU	ADD5	CR003	Platinum
MOHAN	4321098765	NULL	NULL
NAM10	9934567890	NULL	NULL
NAM11	9875678910	NULL	NULL
NAME7	3210987654	NULL	NULL
NAME8	2109876543	NULL	NULL
NAME9	ADD9	NULL	NULL
RADHA	6543210987	CR003	Platinum

17. Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO_OF_TIMES as alias name for number of times

```
SELECT ci.customer_id,c.customer_name,count(ci.movie_id)
NO_OF_TIMES FROM
```

```
customer_issue_details ci JOIN customer_master c ON
c.customer_id=ci.customer_id
```

```
GROUP BY ci.customer_id,ci.movie_id HAVING count(movie_id)>1
```

```
ORDER BY customer_name DESC;
```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	2
CUS003	GAYAN	2
CUS001	AMIT	2
CUS002	ABDHUL	2

18. Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format. Hint: Use NO_OF_MOVIES as alias name for number of movies column. Hint: Use CONTACT_ISD as alias name for contact number. Display the records sorted in ascending order based on customer name and then by movie category.

```
SELECT c.customer_id,c.customer_name,concat('+91-
',substring(c.contact_no,1,3),'-',
substring(c.contact_no,4,3),'-',substring(c.contact_no,7)) CONTACT_ISD
,m.movie_category,count(cc.movie_id) NO_OF_MOVIES FROM
customer_master c JOIN customer_issue_details cc
ON c.customer_id=cc.customer_id JOIN movies_master m ON
m.movie_id=cc.movie_id
GROUP BY c.customer_id,m.movie_category HAVING
count(cc.movie_id)>1
ORDER BY customer_name,movie_category;
```

customer_id	customer_name	CONTACT_ISD	movie_category	NO_OF_MOVIES
CUS002	ABDHUL	+91-876-543-2109	ACTION	2
CUS001	AMIT	+91-987-654-3210	ACTION	2
CUS003	GAYAN	+91-765-432-1098	ACTION	3
CUS003	GAYAN	+91-765-432-1098	ROMANCE	4
CUS004	RADHA	+91-654-321-0987	COMEDY	3

19. Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies. For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The

name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

```
SELECT cid.customer_id , customer_name FROM customer_master cm
JOIN customer_issue_details cid ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)>=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id)
UNION
SELECT cid.customer_id , customer_name FROM
customer_master cm JOIN customer_issue_details cid
ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)<=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id) ORDER BY customer_name;
```

customer_id	customer_name
CUS003	GAYAN
CUS010	NAM10
CUS011	NAM11

20. Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once. Hint: Use NO_OF_TIMES as alias name Display the records in ascending order based on customer name.

```
SELECT c.customer_id,c.customer_name,count(m.movie_id)
NO_OF_TIMES FROM
customer_master c JOIN customer_issue_details cc ON
c.customer_id=cc.customer_id
JOIN movies_master m ON m.movie_id=cc.movie_id
WHERE m.movie_category='Comedy'
GROUP BY c.customer_id HAVING count(m.movie_id)>1
ORDER BY customer_name;
```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	3

21. Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST. Display the records sorted in ascending order based on customer id

```
SELECT cid.customer_id, sum(m.rent_cost) TOTAL_COST FROM
customer_issue_details cid JOIN movies_master mm ON
cid.movie_id=mm.movie_id GROUP BY cid.customer_id order by
customer_id;
```


customer_id	TOTAL_COST
CUS001	200
CUS002	300
CUS003	800
CUS004	400
CUS010	100
CUS011	100

LOAN

```
create database loan;
```

```
use loan;
```

```
CREATE TABLE loan_card_master
```

```
(  
    loan_id    varchar(6) PRIMARY KEY,  
    loan_type  varchar(15),  
    duration_in_years  int(2)  
);
```

```
CREATE TABLE employee_master
```

```
(  
    employee_id      varchar(6) PRIMARY KEY,  
    employee_name     varchar(20),  
    designation       varchar(25),  
department          varchar(25),  
    gender            char(1),  
    date_of_birth     date,  
    date_of_joining   date  
);
```

```
CREATE TABLE item_master
```

```
(
    item_id    varchar(6) PRIMARY KEY,
    item_description    varchar(25),
    issue_status        char(1),
    item_make           varchar(25),
    item_category       varchar(20),
    item_valuation    int(6)
);
```

CREATE TABLE employee_card_details

```
(
    employee_id        varchar(6) REFERENCES
    employee_master,
    loan_id            varchar(6) REFERENCES    loan_card_master,
    card_issue_date    date
);
```

CREATE TABLE employee_issue_details

```
(
    issue_id           varchar(6) PRIMARY KEY,
    employee_id        varchar(6) REFERENCES
    employee_master,
    item_id            varchar(6) REFERENCES    item_master,
```

```
        issue_date      date,  
        return_date     date  
);
```

```
insert into loan_card_master values('L00001','Furniture',5);  
insert into loan_card_master values('L00002','Stationary',0);  
insert into loan_card_master values('L00003','Crockery',1);
```

```
insert into employee_issue_details  
values('ISS001','E00001','I00001','2012-02-03','2014-02-03');  
  
insert into employee_issue_details  
values('ISS002','E00001','I00004','2012-02-03','2020-02-03');  
  
insert into employee_issue_details  
values('ISS003','E00002','I00005','2013-01-03','2015-01-03');  
  
insert into employee_issue_details  
values('ISS004','E00003','I00007','2010-07-04','2012-07-04');  
  
insert into employee_issue_details  
values('ISS005','E00003','I00008','2010-07-04','2012-08-05');  
  
insert into employee_issue_details  
values('ISS006','E00003','I00010','2012-03-14','2012-06-15');  
  
insert into employee_issue_details  
values('ISS007','E00004','I00012','2013-04-14','2016-04-14');
```

```
insert into employee_issue_details  
values('ISS008','E00006','I00018','2012-08-18','2019-04-17');
```

```
insert into employee_issue_details  
values('ISS009','E00004','I00018','2013-04-18','2013-05-18');
```

```
insert into employee_master  
values('E00001','Ram','Manager','Finance','M','1973-12-01','2000-01-01');
```

```
insert into employee_master values('E00002','Abhay','Assistant  
Manager','Finance','M','1976-01-01','2006-12-01');
```

```
insert into employee_master values('E00003','Anita','Senior  
Executive','Marketing','F','1977-05-12','2007-03-21');
```

```
insert into employee_master  
values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');
```

```
insert into employee_master  
values('E00005','Radhica','Manager','HR','F','1976-07-22','2004-01-23');
```

```
insert into employee_master  
values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');
```

```
insert into employee_card_details values('E00001','L00001','2000-01-01');
```

```
insert into employee_card_details values('E00001','L00002','2000-01-01');
```

```
insert into employee_card_details values('E00001','L00003','2002-12-14');
```

```
insert into employee_card_details values('E00002','L00001','2007-02-01');
```

```
insert into employee_card_details values('E00002','L00002','2007-03-11');
```

```
insert into employee_card_details values('E00003','L00001','2007-04-15');
```

```
insert into employee_card_details values('E00003','L00002','2007-04-15');
```

```
insert into employee_card_details values('E00003','L00003','2007-04-15');
```

```
INSERT INTO item_master VALUES ('I00001','Tea  
Table','Y','Wooden','Furniture',5000);
```

```
INSERT INTO item_master VALUES ('I00002','Dinning  
Table','N','Wooden','Furniture',15000);
```

```
INSERT INTO item_master VALUES ('I00003','Tea  
Table','N','Steel','Furniture',6000);
```

```
INSERT INTO item_master VALUES ('I00004','Side  
Table','Y','Wooden','Furniture',2000);
```

```
INSERT INTO item_master VALUES ('I00005','Side  
Table','Y','Steel','Furniture',1500);
```

```

INSERT INTO item_master VALUES      ('I00006','Tea
Table','N','Steel','Furniture',7000);

INSERT INTO item_master VALUES      ('I00007','Dinning
Chair','Y','Wooden','Furniture',1500);

INSERT INTO item_master VALUES      ('I00008','Tea
Table','Y','Wooden','Furniture',4000);

INSERT INTO item_master VALUES
('I00009','Sofa','N','Wooden','Furniture',18000);

INSERT INTO item_master VALUES
      ('I00010','Cupboard','Y','Steel','Furniture',10000);

INSERT INTO item_master VALUES
      ('I00011','Cupboard','N','Steel','Furniture',14000);

INSERT INTO item_master VALUES      ('I00012','Double
Bed','Y','Wooden','Furniture',21000);

INSERT INTO item_master VALUES      ('I00013','Double
Bed','Y','Wooden','Furniture',20000);

INSERT INTO item_master VALUES      ('I00014','Single
Bed','Y','Steel','Furniture',10000);

INSERT INTO item_master VALUES      ('I00015','Single
Bed','N','Steel','Furniture',10000);

INSERT INTO item_master VALUES      ('I00016','Tea
Set','Y','Glass','Crockery',3000);

INSERT INTO item_master VALUES      ('I00017','Tea
Set','Y','Bonechina','Crockery',4000);

INSERT INTO item_master VALUES      ('I00018','Dinning
Set','Y','Glass','Crockery',4500);

```

```
INSERT INTO item_master VALUES      ('I00019','Dinning
Set','N','Bonechina','Crockery',5000);
```

```
INSERT INTO item_master VALUES
      ('I00020','Pencil','Y','Wooden','Stationary',5);
```

```
INSERT INTO item_master VALUES
      ('I00021','Pen','Y','Plastic','Stationary',100);
```

```
INSERT INTO item_master VALUES
      ('I00022','Pen','N','Plastic','Stationary',200);
```

LOAN CARD MASTER

loan_id	loan_type	duration_in_years
L00001	Furniture	5
L00002	Stationary	0
L00003	Crockery	1
NULL	NULL	NULL

EMPLOYEE CARD DETAILS

employee_id	loan_id	card_issue_date
E00001	L00001	2000-01-01
E00001	L00002	2000-01-01
E00001	L00003	2002-12-14
E00002	L00001	2007-02-01
E00002	L00002	2007-03-11
E00003	L00001	2007-04-15
E00003	L00002	2007-04-15
E00003	L00003	2007-04-15

EMPLOYEE ISSUE DETAILS

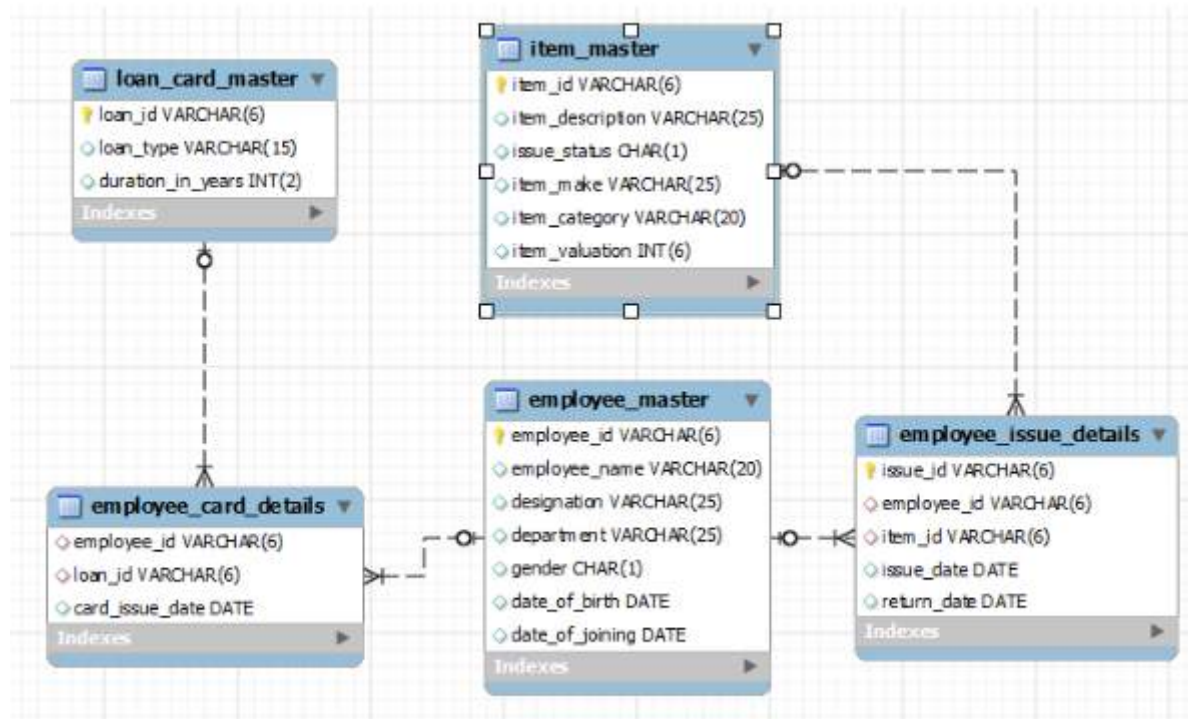
issue_id	employee_id	item_id	issue_date	return_date
ISS001	E00001	I00001	2012-02-03	2014-02-03
ISS002	E00001	I00004	2012-02-03	2020-02-03
ISS003	E00002	I00005	2013-01-03	2015-01-03
ISS004	E00003	I00007	2010-07-04	2012-07-04
ISS005	E00003	I00008	2010-07-04	2012-08-05
ISS006	E00003	I00010	2012-03-14	2012-06-15
ISS007	E00004	I00012	2013-04-14	2016-04-14
ISS008	E00006	I00018	2012-08-18	2019-04-17
ISS009	E00004	I00018	2013-04-18	2013-05-18
NULL	NULL	NULL	NULL	NULL

EMPLOYEE MASTER

employee_id	employee_name	designation	department	gender	date_of_birth	date_of_joining
E00001	Ram	Manager	Finance	M	1973-12-01	2000-01-01
E00002	Abhay	Assistant Manager	Finance	M	1976-01-01	2006-12-01
E00003	Anita	Senior Executive	Marketing	F	1977-05-12	2007-03-21
E00004	Zuben	Manager	Marketing	M	1974-10-12	2003-07-23
E00005	Radhica	Manager	HR	F	1976-07-22	2004-01-23
E00006	John	Executive	HR	M	1983-11-08	2010-05-17
NULL	NULL	NULL	NULL	NULL	NULL	NULL

ITEM MASTER

item_id	item_description	issue_status	item_make	item_category	item_valuation
I00001	Tea Table	Y	Wooden	Furniture	5000
I00002	Dinning Table	N	Wooden	Furniture	15000
I00003	Tea Table	N	Steel	Furniture	6000
I00004	Side Table	Y	Wooden	Furniture	2000
I00005	Side Table	Y	Steel	Furniture	1500
I00006	Tea Table	N	Steel	Furniture	7000
I00007	Dinning Chair	Y	Wooden	Furniture	1500
I00008	Tea Table	Y	Wooden	Furniture	4000
I00009	Sofa	N	Wooden	Furniture	18000
I00010	Cupboard	Y	Steel	Furniture	10000
I00011	Cupboard	N	Steel	Furniture	14000
I00012	Double Bed	Y	Wooden	Furniture	21000
I00013	Double Bed	Y	Wooden	Furniture	20000
I00014	Single Bed	Y	Steel	Furniture	10000
I00015	Single Bed	N	Steel	Furniture	10000
I00016	Tea Set	Y	Glass	Crockery	3000
I00017	Tea Set	Y	Bonechina	Crockery	4000
I00018	Dinning Set	Y	Glass	Crockery	4500
I00019	Dinning Set	N	Bonechina	Crockery	5000
I00020	Pencil	Y	Wooden	Stationary	5
I00021	Pen	Y	Plastic	Stationary	100
I00022	Pen	N	Plastic	Stationary	200
NULL	NULL	NULL	NULL	NULL	NULL



1. Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

```
SELECT item_category,count(item_id) Count_category FROM
item_master GROUP BY item_category ORDER BY Count_category
DESC;
```

item_category	Count_category
Furniture	15
Crockery	4
Stationary	3

2. Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

```
SELECT count(employee_id) No_of_Employees FROM
employee_master WHERE department='HR';
```

No_of_Employees
2

3. Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id,employee_name,designation,department FROM
employee_master
```

```
WHERE employee_id NOT IN (SELECT employee_id FROM
employee_issue_details)
```

```
ORDER BY employee_id;
```

employee_id	employee_name	designation	department
E00005	Radhica	Manager	HR
NULL	NULL	NULL	NULL

4. Write a query to display the employee id, employee name who was issued an item of highest valuation. In case of multiple records, display the records sorted in ascending order based on employee id.[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

```
SELECT employee_id,employee_name FROM employee_master
```

```
WHERE employee_id IN(SELECT employee_id FROM
employee_issue_details
```

```
WHERE item_id IN (SELECT item_id FROM item_master
```

```
WHERE item_valuation=(SELECT max(item_valuation) FROM
```

```
item_master i JOIN employee_issue_details e ON
i.item_id=e.item_id)))
```

employee_id	employee_name
E00004	Zuben
NULL	NULL

5. Write a query to display issue_id, employee_id, employee_name. Display the records sorted in ascending order based on issue id.

```
SELECT eid.issue_id, eid.employee_id, em.employee_name
FROM employee_master em JOIN employee_issue_details eid
ON em.employee_id=eid.employee_id ORDER BY eid.issue_id;
```

issue_id	employee_id	employee_name
ISS001	E00001	Ram
ISS002	E00001	Ram
ISS003	E00002	Abhay
ISS004	E00003	Anita
ISS005	E00003	Anita
ISS006	E00003	Anita
ISS007	E00004	Zuben
ISS008	E00006	John
ISS009	E00004	Zuben

6. Write a query to display employee id, employee name who don't have loan cards. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id, employee_name FROM employee_master
WHERE employee_id NOT IN(SELECT employee_id FROM
employee_card_details);
```

employee_id	employee_name
E00004	Zuben
E00005	Radhica
E00006	John
NULL	NULL

7. Write a query to count the number of cards issued to an employee "Ram". Give the count an alias name as No_of_Cards.

```
SELECT count(loan_id) No_of_Cards FROM
employee_card_details WHERE employee_id IN
(SELECT employee_id FROM employee_master WHERE
employee_name='Ram');
```

(or)

```
SELECT count(loan_id) No_of_Cards FROM
employee_card_details c JOIN employee_master e
ON c.employee_id = e.employee_id
WHERE e.employee_name= 'Ram';
```

No_of_Cards
3

8. Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

```
SELECT count(e.employee_id) Count_Stationary
FROM employee_card_details e JOIN loan_card_master l
ON e.loan_id=l.loan_id WHERE l.loan_type='Stationary';
```

Count_Stationary
3

9. Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then

```
SELECT e.employee_id,employee_name,count(e.item_id) Count FROM  
employee_issue_details e JOIN employee_master em ON  
e.employee_id=em.employee_id  
  
GROUP BY e.employee_id ORDER BY count DESC,e.employee_id;
```

employee_id	employee_name	Count
E00003	Anita	3
E00001	Ram	2
E00004	Zuben	2
E00002	Abhay	1
E00006	John	1

10. Write a query to display the employee id, employee name who was issued an item of minimum valuation.In case of multiple records, display them sorted in ascending order based on employee id.[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

```
SELECT employee_id,employee_name FROM employee_master  
  
WHERE employee_id IN(SELECT employee_id FROM  
employee_issue_details  
  
WHERE item_id IN (SELECT item_id FROM item_master  
  
WHERE item_valuation=(SELECT min(item_valuation) FROM  
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)))  
  
ORDER BY employee_id;
```


employee_id	employee_name
E00002	Abhay
E00003	Anita
NULL	NULL

11. Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION. Display the records sorted in ascending order based on employee id. Consider only employees who have been issued atleast 1 item.

```
SELECT      e.employee_id,em.employee_name,sum(i.item_valuation)
TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id ORDER BY employee_id;
```

employee_id	employee_name	TOTAL_VALUATION
E00001	Ram	7000
E00002	Abhay	1500
E00003	Anita	15500
E00004	Zuben	25500
E00006	John	4500

12. Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days. Display the records sorted in ascending order based on employee id.

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON
e.employee_id=ei.employee_id
```

WHERE datediff(ei.return_date,ei.issue_date)>365

ORDER BY employee_id;

employee_id	employee_name
E00001	Ram
E00002	Abhay
E00003	Anita
E00004	Zuben
E00006	John

13. Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT_ITEMS. Display the records sorted in ascending order on employee id.

```
SELECT          e.employee_id,e.employee_name,count(ei.item_id)
COUNT_ITEMS FROM
```

```
employee_master e JOIN employee_issue_details ei ON
e.employee_id=ei.employee_id
```

```
JOIN item_master i ON ei.item_id=i.item_id
```

```
WHERE i.item_category='Furniture'
```

```
GROUP BY ei.employee_id HAVING count(ei.item_id)>1;
```

employee_id	employee_name	COUNT_ITEMS
E00001	Ram	2
E00003	Anita	3

14. Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

```
SELECT gender,count(employee_id) FROM employee_master
```

GROUP BY gender ORDER BY gender;

gender	count(employee_id)
F	2
M	4

15. Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id,employee_name FROM employee_master  
WHERE year(date_of_joining)>'2005'  
ORDER BY employee_id;
```

employee_id	employee_name
E00002	Abhay
E00003	Anita
E00006	John
NULL	NULL

16. Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No_of_Furnitures. Display the records sorted in ascending order based on issue_status.

```
SELECT issue_status,count(item_id) No_of_Furniture FROM  
item_master WHERE item_category='Furniture'  
GROUP BY issue_status ORDER BY issue_status;
```

issue_status	No_of_Furniture
N	6
Y	9

17. Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items. Display the

records in ascending order based on Item Category, then by item make and then by item description.

```
SELECT      item_category,item_make,item_description,count(item_id)
No_of_items FROM
item_master GROUP BY item_category,item_make,item_description
ORDER BY item_category,item_make,item_description;
```

item_category	item_make	item_description	No_of_items
Crockery	Bonechina	Dinning Set	1
Crockery	Bonechina	Tea Set	1
Crockery	Glass	Dinning Set	1
Crockery	Glass	Tea Set	1
Furniture	Steel	Cupboard	2
Furniture	Steel	Side Table	1
Furniture	Steel	Single Bed	2
Furniture	Steel	Tea Table	2
Furniture	Wooden	Dinning Chair	1
Furniture	Wooden	Dinning Table	1
Furniture	Wooden	Double Bed	2
Furniture	Wooden	Side Table	1
Furniture	Wooden	Sofa	1
Furniture	Wooden	Tea Table	2
Stationary	Plastic	Pen	2
Stationary	Wooden	Pencil	1

18. Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

```
SELECT      e.employee_id,employee_name,i.item_id,i.item_description
FROM
employee_master      e      JOIN      employee_issue_details      ei      ON
e.employee_id=ei.employee_id
```

JOIN item_master i ON i.item_id=ei.item_id

WHERE month(ei.issue_date)='01' and year(ei.issue_date)='2013'

ORDER BY employee_id,item_id;

employee_id	employee_name	item_id	item_description
E00002	Abhay	I00005	Side Table

19. Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories. Give the alias name for category count as COUNT_CATEGORY. Display the records sorted in ascending order based on employee id.

SELECT ei.employee_id,e.employee_name,count(DISTINCT
i.item_category) COUNT_CATEGORY FROM

employee_master e JOIN employee_issue_details ei ON
e.employee_id=ei.employee_id

JOIN item_master i ON i.item_id=ei.item_id

GROUP BY ei.employee_id

HAVING COUNT_CATEGORY>=2

ORDER BY employee_id;

employee_id	employee_name	COUNT_CATEGORY
E00004	Zuben	2

20. Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.

SELECT item_id, item_description FROM item_master

WHERE item_id NOT IN (SELECT item_id from employee_issue_details)

ORDER BY item_id;

item_id	item_description
I00002	Dinning Table
I00003	Tea Table
I00006	Tea Table
I00009	Sofa
I00011	Cupboard
I00013	Double Bed
I00014	Single Bed
I00015	Single Bed
I00016	Tea Set
I00017	Tea Set
I00019	Dinning Set
I00020	Pencil
I00021	Pen
I00022	Pen
NULL	NULL

21. Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION. [Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

```
SELECT      e.employee_id,em.employee_name,sum(i.item_valuation)
TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id HAVING sum(i.item_valuation)<=ALL(
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM
```

```

item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id);

```

employee_id	employee_name	TOTAL_VALUATION
E00002	Abhay	1500

22. Write a query to display the employee id, employee name, card issue date and card valid date. Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE. [Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.]

```

SELECT e.employee_id,e.employee_name,card_issue_date,
case
when l.duration_in_years>0 then date_add(ec.card_issue_date,interval
l.duration_in_years year)
when l.duration_in_years=0 then 'No Validity Date' end
CARD_VALID_DATE
FROM
employee_master e JOIN employee_card_details ec ON
e.employee_id=ec.employee_id
JOIN loan_card_master l ON l.loan_id=ec.loan_id
ORDER BY employee_name,CARD_VALID_DATE;

```

employee_id	employee_name	card_issue_date	CARD_VALID_DATE
E00002	Abhay	2007-02-01	2012-02-01
E00002	Abhay	2007-03-11	No Validity Date
E00003	Anita	2007-04-15	2008-04-15
E00003	Anita	2007-04-15	2012-04-15
E00003	Anita	2007-04-15	No Validity Date
E00001	Ram	2002-12-14	2003-12-14
E00001	Ram	2000-01-01	2005-01-01
E00001	Ram	2000-01-01	No Validity Date

23. Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON
e.employee_id=ei.employee_id
WHERE e.employee_id NOT IN (SELECT employee_id FROM
employee_issue_details
WHERE year(issue_date)='2013')
ORDER BY employee_id;
```

employee_id	employee_name
E00001	Ram
E00003	Anita
E00006	John

24. Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.


```

SELECT  issue_id,  eid.employee_id,  employee_name,  im.item_id,
item_description,issue_date

FROM  employee_issue_details eid JOIN  employee_master em ON
eid.employee_id=em.employee_id

JOIN item_master im ON eid.item_id=im.item_id

ORDER BY issue_date DESC, issue_id;

```

issue_id	employee_id	employee_name	item_id	item_description	issue_date
ISS009	E00004	Zuben	I00018	Dinning Set	2013-04-18
ISS007	E00004	Zuben	I00012	Double Bed	2013-04-14
ISS003	E00002	Abhay	I00005	Side Table	2013-01-03
ISS008	E00006	John	I00018	Dinning Set	2012-08-18
ISS006	E00003	Anita	I00010	Cupboard	2012-03-14
ISS001	E00001	Ram	I00001	Tea Table	2012-02-03
ISS002	E00001	Ram	I00004	Side Table	2012-02-03
ISS004	E00003	Anita	I00007	Dinning Chair	2010-07-04
ISS005	E00003	Anita	I00008	Tea Table	2010-07-04

25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

```

SELECT      e.employee_id,em.employee_name,sum(i.item_valuation)
TOTAL_VALUATION FROM

item_master i JOIN employee_issue_details e ON e.item_id=i.item_id

JOIN employee_master em ON em.employee_id=e.employee_id

GROUP BY e.employee_id HAVING sum(i.item_valuation)>=ALL(

```

```
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM  
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id  
JOIN employee_master em ON em.employee_id=e.employee_id  
GROUP BY e.employee_id);
```

employee_id	employee_name	TOTAL_VALUATION
E00004	Zuben	25500

BANK

```
create database bank;
```

```
use bank;
```

```
CREATE TABLE customer_master(  
CUSTOMER_NUMBER VARCHAR(6),  
FIRSTNAME VARCHAR(30),  
middlename VARCHAR(30),  
lastname VARCHAR(30),  
CUSTOMER_CITY VARCHAR(15),  
CUSTOMER_CONTACT_NO VARCHAR(10),  
occupation VARCHAR(10),  
CUSTOMER_DATE_OF_BIRTH DATE,  
CONSTRAINT customer_custid_pk PRIMARY KEY (CUSTOMER_NUMBER));
```

```
CREATE TABLE branch_master(  
branch_id VARCHAR(6),  
branch_name VARCHAR(30),  
branch_city VARCHAR(30),  
CONSTRAINT branch_bid_pk PRIMARY KEY (branch_id));
```

```
CREATE TABLE account_master
(account_number VARCHAR(255),
customer_number VARCHAR(255),
branch_id VARCHAR(255),
opening_balance INT(20),
account_opening_date DATE,
account_type VARCHAR(10),
account_status VARCHAR(10),
PRIMARY KEY (account_number),
FOREIGN KEY (customer_number) references customer_master(customer_number),
FOREIGN KEY (branch_id) references branch_master(branch_id));
```

```
CREATE TABLE transaction_details(
transaction_number VARCHAR(6),
account_number VARCHAR(6),
date_of_transaction DATE,
medium_of_transaction VARCHAR(20),
transaction_type VARCHAR(20),
transaction_amount INT(7),
CONSTRAINT transaction_details_tnumber_pk PRIMARY KEY (transaction_number),
CONSTRAINT transaction_details_acnumber_fk FOREIGN KEY (account_number)
REFERENCES account_master (account_number));
```

```
CREATE TABLE loan_details
(customer_number varchar(255),
branch_id varchar(255),
loan_amount bigint(20),
foreign key(customer_number) references customer_master(customer_number));
```

```

insert into customer_master values('C00001', 'RAMESH', 'CHANDRA', 'SHARMA', 'DELHI',
    '9543198345', 'SERVICE', '1976-12-06');

insert into customer_master values('C00002', 'AVINASH', 'SUNDER', 'MINHA', 'DELHI',
    '9876532109', 'SERVICE', '1974-10-16');

insert into customer_master values('C00003', 'RAHUL', 'NULL', 'RASTOGI', 'DELHI',
    '9765178901', 'STUDENT', '1981-09-26');

insert into customer_master values('C00004', 'PARUL', 'NULL', 'GANDHI', 'DELHI',
    '9876532109', 'HOUSEWIFE', '1976-11-03');

insert into customer_master values('C00005', 'NAVEEN', 'CHANDRA', 'AEDEKAR',
    'MUMBAI', '8976523190', 'SERVICE', '1976-09-19');

insert into customer_master values('C00006', 'CHITRESH', 'NULL', 'BARWE', 'MUMBAI',
    '7651298321', 'STUDENT', '1992-11-06');

insert into customer_master values('C00007', 'AMIT', 'KUMAR', 'BORKAR', 'MUMBAI',
    '9875189761', 'STUDENT', '1981-09-06');

insert into customer_master values('C00008', 'NISHA', 'NULL', 'DAMLE', 'MUMBAI',
    '7954198761', 'SERVICE', '1975-12-03');

insert into customer_master values('C00009', 'ABHISHEK', 'NULL', 'DUTTA', 'KOLKATA',
    '9856198761', 'SERVICE', '1973-05-22');

insert into customer_master values('C00010', 'SHANKAR', 'NULL', 'NAIR', 'CHENNAI', '8765489076',
    'SERVICE', '1976-07-12');


insert into branch_master values('B00001', 'ASAF ALI ROAD', 'DELHI');

insert into branch_master values('B00002', 'NEW DELHI MAIN BRANCH', 'DELHI');

insert into branch_master values('B00003', 'DELHI CANTT', 'DELHI');

insert into branch_master values('B00004', 'JASOLA', 'DELHI');

insert into branch_master values('B00005', 'MAHIM', 'MUMBAI');

insert into branch_master values('B00006', 'VILE PARLE', 'MUMBAI');

insert into branch_master values('B00007', 'MANDVI', 'MUMBAI');

insert into branch_master values('B00008', 'JADAVPUR', 'KOLKATA');

insert into branch_master values('B00009', 'KODAMBAKKAM', 'CHENNAI');

```

```

insert into account_master values('A00001','C00001','B00001',1000    ,'2012-12-15', 'SAVING',
    'ACTIVE');

insert into account_master values('A00002'    ,'C00002','B00001',1000,'2012-06-12'    ,'SAVING',
    'ACTIVE');

insert into account_master values('A00003'    ,'C00003',    'B00002',    1000    ,'2012-05-17'
    ,'SAVING',    'ACTIVE');

insert into account_master values('A00004'    ,'C00002',    'B00005',    1000    ,'2013-01-27'
    ,'SAVING    ','ACTIVE');

insert into account_master values('A00005'    ,'C00006',    'B00006',    1000    ,'2012-12-17'
    ,'SAVING','ACTIVE');

insert into account_master values('A00006'    ,'C00007',    'B00007',    1000    ,'2010-08-12'
    ,'SAVING    ','SUSPENDED');

insert into account_master values('A00007'    ,'C00007',    'B00001',    1000    ,'2012-10-02'
    ,'SAVING    ','ACTIVE');

insert into account_master values('A00008'    ,'C00001','B00003',    1000    ,'2009-11-09'
    ,'SAVING    ','TERMINATED');

insert into account_master values('A00009'    ,'C00003',    'B00007',    1000    ,'2008-11-30'
    ,'SAVING',    'TERMINATED');

insert into account_master values('A00010'    ,'C00004',    'B00002',    1000    ,'2013-03-01'
    ,'SAVING',    'ACTIVE');


insert into transaction_details values('T00001', 'A00001',    '2013-01-01', 'CHEQUE',
    'DEPOSIT',    2000);

insert into transaction_details values('T00002' , 'A00001'    ,'2013-02-01'    ,'CASH'
    ,'WITHDRAWAL',    1000);

insert into transaction_details values('T00003', 'A00002'    ,'2013-01-01', 'CASH' , 'DEPOSIT',
    2000);

insert into transaction_details values('T00004', 'A00002',    '2013-02-01', 'CASH'    ,'DEPOSIT',
    3000);

insert into transaction_details values('T00005', 'A00007',    '2013-01-11', 'CASH' , 'DEPOSIT',
    7000);

insert into transaction_details values('T00006', 'A00007',    '2013-01-13', 'CASH' , 'DEPOSIT',
    9000);

insert into transaction_details values('T00007', 'A00001',    '2013-03-13', 'CASH' , 'DEPOSIT'
    ,4000);

```

```
insert into transaction_details values('T00008', 'A00001', '2013-03-14', 'CHEQUE',  
    , 'DEPOSIT' ,3000);
```

```
insert into transaction_details values('T00009', 'A00001', '2013-03-21', 'CASH',  
    , 'WITHDRAWAL' ,9000);
```

```
insert into transaction_details values('T00010', 'A00001', '2013-03-22', 'CASH',  
    , 'WITHDRAWAL' ,2000);
```

```
insert into transaction_details values('T00011', 'A00002', '2013-03-25', 'CASH',  
    , 'WITHDRAWAL' ,7000);
```

```
insert into transaction_details values('T00012', 'A00007', '2013-03-26', 'CASH',  
    , 'WITHDRAWAL' ,2000);
```

```
insert into Loan_details values('C00001', 'B00001', 100000);
```

```
insert into Loan_details values('C00002', 'B00002', 200000);
```

```
insert into Loan_details values('C00009', 'B00008', 400000);
```

```
insert into Loan_details values('C00010', 'B00009', 500000);
```

```
insert into Loan_details values('C00001', 'B00003', 600000);
```

```
insert into Loan_details values('C00002', 'B00001', 600000);
```

```

    erDiagram
        branch_master ||--o{ account_master : "has"
        loan_details ||--o{ customer_master : "has"
        loan_details ||--o{ account_master : "has"
        loan_details ||--o{ transaction_details : "has"
        customer_master ||--o{ account_master : "has"
        customer_master ||--o{ transaction_details : "has"

        branch_master {
            VARCHAR(6) branch_id PK
            VARCHAR(30) branch_name
            VARCHAR(30) branch_city
        }
        loan_details {
            VARCHAR(6) customer_number PK
            VARCHAR(6) branch_id PK
            INT(7) loan_amount
        }
        customer_master {
            VARCHAR(6) customer_number PK
            VARCHAR(30) firstname
            VARCHAR(30) middlename
            VARCHAR(30) lastname
            VARCHAR(15) customer_city
            VARCHAR(10) customer_contact_no
            DATE customer_date_of_birth
        }
        account_master {
            VARCHAR(6) account_number PK
            VARCHAR(6) customer_number PK
            VARCHAR(6) branch_id PK
            INT(7) opening_balance
            DATE account_opening_date
            VARCHAR(10) account_type
            VARCHAR(10) account_status
        }
        transaction_details {
            VARCHAR(6) transaction_number PK
            VARCHAR(6) account_number PK
            DATE date_of_transaction
            VARCHAR(20) medium_of_transaction
            VARCHAR(20) transaction_type
            INT(7) transaction_amount
        }

```

The diagram illustrates the following tables and their attributes:

- branch_master**
 - branch_id VARCHAR(6) (Primary Key)
 - branch_name VARCHAR(30)
 - branch_city VARCHAR(30)
- loan_details**
 - customer_number VARCHAR(6) (Primary Key)
 - branch_id VARCHAR(6) (Primary Key)
 - loan_amount INT(7)
- customer_master**
 - customer_number VARCHAR(6) (Primary Key)
 - firstname VARCHAR(30)
 - middlename VARCHAR(30)
 - lastname VARCHAR(30)
 - customer_city VARCHAR(15)
 - customer_contact_no VARCHAR(10)
 - customer_date_of_birth DATE
- account_master**
 - account_number VARCHAR(6) (Primary Key)
 - customer_number VARCHAR(6) (Primary Key)
 - branch_id VARCHAR(6) (Primary Key)
 - opening_balance INT(7)
 - account_opening_date DATE
 - account_type VARCHAR(10)
 - account_status VARCHAR(10)
- transaction_details**
 - transaction_number VARCHAR(6) (Primary Key)
 - account_number VARCHAR(6) (Primary Key)
 - date_of_transaction DATE
 - medium_of_transaction VARCHAR(20)
 - transaction_type VARCHAR(20)
 - transaction_amount INT(7)

Relationships are indicated by dashed lines with crow's foot notation:

- branch_master** to **account_master**: One-to-Many relationship (1 to 0..1).
- loan_details** to **customer_master**: One-to-Many relationship (1 to 0..1).
- loan_details** to **account_master**: One-to-Many relationship (1 to 0..1).
- loan_details** to **transaction_details**: One-to-Many relationship (1 to 0..1).
- customer_master** to **account_master**: One-to-Many relationship (1 to 0..1).
- customer_master** to **transaction_details**: One-to-Many relationship (1 to 0..1).

[illegible]

ACCOUNT MASTER

account_number	customer_number	branch_id	opening_balance	account_opening_date	account_type	account_status
A00001	C00001	B00001	1000	2012-12-15	SAVING	ACTIVE
A00002	C00002	B00001	1000	2012-06-12	SAVING	ACTIVE
A00003	C00003	B00002	1000	2012-05-17	SAVING	ACTIVE
A00004	C00002	B00005	1000	2013-01-27	SAVING	ACTIVE
A00005	C00006	B00006	1000	2012-12-17	SAVING	ACTIVE
A00006	C00007	B00007	1000	2010-08-12	SAVING	SUSPENDED
A00007	C00007	B00001	1000	2012-10-02	SAVING	ACTIVE
A00008	C00001	B00003	1000	2009-11-09	SAVING	TERMINATED
A00009	C00003	B00007	1000	2008-11-30	SAVING	TERMINATED
A00010	C00004	B00002	1000	2013-03-01	SAVING	ACTIVE
NULL	NULL	NULL	NULL	NULL	NULL	NULL

BRANCH MASTER

branch_id	branch_name	branch_city
B00001	ASAF ALI ROAD	DELHI
B00002	NEW DELHI MAIN BRANCH	DELHI
B00003	DELHI CANTT	DELHI
B00004	JASOLA	DELHI
B00005	MAHIM	MUMBAI
B00006	VILE PARLE	MUMBAI
B00007	MANDVI	MUMBAI
B00008	JADAVPUR	KOLKATA
B00009	KODAMBAKKAM	CHENNAI
NULL	NULL	NULL

LOAN DETAILS

customer_number	branch_id	loan_amount
C00001	B00001	100000
C00002	B00002	200000
C00009	B00008	400000
C00010	B00009	500000
C00001	B00003	600000
C00002	B00001	600000

TRANSACTION DETAILS

transaction_number	account_number	date_of_transaction	medium_of_transaction	transaction_type	transaction_amount
T00001	A00001	2013-01-01	CHEQUE	DEPOSIT	2000
T00002	A00001	2013-02-01	CASH	WITHDRAWAL	1000
T00003	A00002	2013-01-01	CASH	DEPOSIT	2000
T00004	A00002	2013-02-01	CASH	DEPOSIT	3000
T00005	A00007	2013-01-11	CASH	DEPOSIT	7000
T00006	A00007	2013-01-13	CASH	DEPOSIT	9000
T00007	A00001	2013-03-13	CASH	DEPOSIT	4000
T00008	A00001	2013-03-14	CHEQUE	DEPOSIT	3000
T00009	A00001	2013-03-21	CASH	WITHDRAWAL	9000
T00010	A00001	2013-03-22	CASH	WITHDRAWAL	2000
T00011	A00002	2013-03-25	CASH	WITHDRAWAL	7000
T00012	A00007	2013-03-26	CASH	WITHDRAWAL	2000
NULL	NULL	NULL	NULL	NULL	NULL

QUERIES

1. Write a query to display account number, customer's number, customer's firstname, lastname, account opening date. Display the records sorted in ascending order based on account number.

```
SELECT a.account_number,c.customer_number,c.firstname,c.lastname,a.account_number
FROM      customer_master      c      JOIN      account_master      a      ON
c.customer_number=a.customer_number
ORDER BY a.account_number;
```

account_number	customer_number	firstname	lastname	account_opening_date
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27
A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09
A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2. Write a query to display the number of customer's from Delhi. Give the count an alias name of Cust_Count.

```
SELECT count(customer_number) Cust_Count FROM customer_master WHERE customer_city='Delhi';
```

cust_count
4

3. Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number FROM account_master a join
customer_master c ON c.customer_number=a.customer_number WHERE
day(a.account_opening_date)>'15' ORDER BY c.customer_number,a.account_number;
```

customer_number	firstname	account_number
C00002	AVINASH	A00004
C00003	RAHUL	A00003
C00003	RAHUL	A00009
C00006	CHITRESH	A00005

4. Write a query to display customer number, customer's first name, account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number
FROM account_master a JOIN customer_master c
ON c.customer_number=a.customer_number
WHERE a.account_status='Terminated'
ORDER BY c.customer_number,a.account_number;
```

customer_number	firstname	account_number
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Write a query to display the total number of withdrawals and total number of deposits being done by customer whose customer number ends with 001. The query should display transaction type and the number of transactions. Give an alias name as Trans_Count for number of transactions. Display the records sorted in ascending order based on transaction type.

```
SELECT transaction_type,count(transaction_number) Trans_Count
FROM account_master am JOIN transaction_details td
ON am.account_number=td.account_number
WHERE customer_number like '%001'
GROUP BY transaction_type
ORDER BY transaction_type;
```

transaction_type	Trans_count
DEPOSIT	3
WITHDRAWAL	3

6. Write a query to display the number of customers who have registration but no account in the bank. Give the alias name as Count_Customer for number of customers.

```
SELECT count(customer_number) Count_Customer FROM customer_master
WHERE customer_number NOT IN (SELECT customer_number FROM account_master);
```

Count_customer
4

7. Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit_Amount. Display the records in sorted order based on account number.

```
SELECT a.account_number,a.opening_balance+sum(t.transaction_amount)
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number;
```

account_number	Deposit_Amount
A00001	10000
A00002	6000
A00007	17000

8. Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No_of_Accounts.For the branch city where we don't have any accounts opened display 0. Display the records in sorted order based on branch city.

```
SELECT branch.branch_city, count(account.account_number) No_of_Accounts
FROM branch_master LEFT JOIN account_master
ON account.branch_id=branch.branch_id
GROUP BY branch.branch_city ORDER BY branch_city;
```

branch_city	No_of_accounts
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9. Write a query to display the firstname of the customers who have more than 1 account. Display the records in sorted order based on firstname.

```
SELECT c.firstname FROM  
customer_master c JOIN account_master a ON a.customer_number=c.customer_number  
GROUP BY a.customer_number HAVING count(a.account_number)>1;
```

firstname
AMIT
AVINASH
RAHUL
RAMESH

10. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.

```
SELECT c.customer_number,c.firstname,c.lastname FROM  
customer_master c JOIN loan_details l ON c.customer_number=l.customer_number  
GROUP BY l.customer_number HAVING count(l.branch_id)>1  
ORDER BY c.customer_number;
```

customer_number	firstname	lastname
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11. Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different. Display the records sorted in ascending order based on customer number.

```
SELECT c.customer_number,c.firstname,c.customer_city,b.branch_city FROM  
Customer_master c JOIN Account_master a ON c.customer_number=a.customer_number  
JOIN Branch_master b ON b.branch_id=a.branch_id  
WHERE b.branch_city<>c.customer_city  
ORDER BY c.customer_number;
```

customer_number	firstname	customer_city	branch_city
C00002	AVINASH	DELHI	MUMBAI
C00003	RAHUL	DELHI	MUMBAI
C00007	AMIT	MUMBAI	DELHI

12. Write a query to display the number of clients who have asked for loans but they don't have any account in the bank though they are registered customers. Give the count an alias name of Count.

```
SELECT count(c.customer_number)Count FROM customer_master c JOIN loan_details l
ON c.customer_number=l.customer_number
WHERE c.customer_number NOT IN (SELECT customer_number FROM account_master);
```

Count
2

13. Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT account_number FROM transaction_details
GROUP BY account_number
HAVING count(transaction_number)>=ALL
(SELECT count(transaction_number) FROM transaction_details
GROUP BY account_number) ORDER BY account_number;
```

account_number
A00001

14. Write a query to show the branch name,branch city where we have the maximum customers. For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
SELECT b.branch_name,b.branch_city FROM
Branch_master b JOIN account a ON a.branch_id=b.branch_id
GROUP BY a.branch_id HAVING count(a.customer_number)>=ALL
(SELECT count(customer_number) FROM
Account_master GROUP BY branch_id)
ORDER BY b.branch_name;
```

branch_name	branch_city
ASAF ALI ROAD	DELHI

15. Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well. For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011. Display the records sorted in ascending order based on account number.

```
SELECT td.account_number,
sum(CASE WHEN transaction_type='Deposit' THEN transaction_amount END)
+(SELECT opening_balance
FROM account_master where account_number=td.account_number) Deposit,
sum(CASE WHEN transaction_type='Withdrawal' THEN transaction_amount END) Withdrawal
FROM transaction_details td
GROUP BY td.account_number
HAVING Withdrawal > Deposit
ORDER BY td.account_number;
```

(or)

```
SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d Deposit,ifnull(t1.w,0) Withdrawal FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit'GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.w,0)>t2.d
ORDER BY account_number;
```

account_number	Deposit	Withdrawal
A00001	10000	12000
A00002	6000	7000

16. Write a query to show the balance amount for account number that ends with 001. Note: Balance amount includes account opening balance also. Give alias name as Balance_Amount. For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000) – (500+500+1000).

```
SELECT ifnull((SUM(CASE WHEN transaction_type='Deposit'
THEN transaction_amount END)) -
(SUM(CASE WHEN transaction_type='Withdrawal'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001'),(SUM(CASE WHEN
transaction_type='Deposit'
THEN transaction_amount END))+(select opening_balance
from account_master where account_number like '%001')) AS Balance_Amount
FROM transaction_details where account_number like '%001';
```

(or)

```
SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d-ifnull(t1.w,0) Balance_Amount FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit'GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.account_number,t2.account_number) LIKE '%001'
ORDER BY account_number;
```

account_number	Balance_Amount
A00001	-2000

17. Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as

Count_Trans. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,t.account_number, count(t.account_number) Count_Trans
FROM transaction_details t JOIN account_master a ON a.account_number=t.account_number
JOIN customer c ON c.customer_number=a.customer_number
GROUP BY t.account_number ORDER BY c.customer_number, a.account_number;
```

customer_number	firstname	account_number	Count_Trans
C00001	RAMESH	A00001	6
C00002	AVINASH	A00002	3
C00007	AMIT	A00007	3

18. Write a query to display the customer's firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.

```
SELECT c.firstname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1
ORDER BY c.firstname;
```

firstname
AMIT
AVINASH
RAHUL
RAMESH

19. Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches. For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011. So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT Id.customer_number, firstname, lastname
FROM customer_master cm JOIN loan_details Id
ON cm.customer_number=Id.customer_number
GROUP BY customer_number
HAVING count(branch_id)>=2 AND sum(loan_amount)>=
```

ALL(SELECT sum(loan_amount) FROM loan GROUP BY customer_number);

customer_number	firstname	lastname
C00002	AVINASH	MINHA

20. Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans. Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT c.customer_number,c.firstname,l.branch_id,l.loan_amount FROM  
Customer_master c JOIN loan_details l ON c.customer_number=l.customer_number  
ORDER BY c.customer_number,l.branch_id,l.loan_amount;
```

customer_number	firstname	branch_id	loan_amount
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000
C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000
C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21. Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count_Branch. Display the records sorted in ascending order based on city name.

```
SELECT branch_city,count(branch_id) Count_Branch FROM  
Branch_master GROUP BY branch_city  
ORDER BY branch_city;
```

branch_city	Count_Branch
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22. Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active. Display the records sorted in ascending order based on account id /account number.

```
SELECT a.account_number,c.firstname,c.lastname FROM  
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number and  
a.account_status='Active'
```

ORDER BY a.account_number;

account_number	firstname	lastname
A00001	RAMESH	SHARMA
A00002	AVINASH	MINHA
A00003	RAHUL	RASTOGI
A00004	AVINASH	MINHA
A00005	CHITRESH	BARWE
A00007	AMIT	BORKAR
A00010	PARUL	GANDHI

23. Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle_Name. Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,ifnull(middlename,lastname) Middle_name FROM
```

```
Customer_master ORDER BY customer_number;
```

customer_number	firstname	Middle_name
C00001	RAMESH	CHANDRA
C00002	AVINASH	SUNDER
C00003	RAHUL	NULL
C00004	PARUL	NULL
C00005	NAVEEN	CHANDRA
C00006	CHITRESH	NULL
C00007	AMIT	KUMAR
C00008	NISHA	DAMLE
C00009	ABHISHEK	DUTTA
C00010	SHANKAR	NAIR

24. Write a query to display the customer number , firstname, customer's date of birth . Display the records sorted in ascending order of date of birth year and within that sort by firstname in ascending order.

```
SELECT customer_number,firstname,customer_date_of_birth FROM
```

```
Customer_master ORDER BY year(customer_date_of_birth),customer_number;
```

customer_number	firstname	customer_date_of_birth
C00009	ABHISHEK	1973-05-22
C00002	AVINASH	1974-10-16
C00008	NISHA	1975-12-03
C00001	RAMESH	1976-12-06
C00004	PARUL	1976-11-03
C00005	NAVEEN	1976-09-19
C00010	SHANKAR	1976-07-12
C00003	RAHUL	1981-09-26
C00007	AMIT	1981-09-06
C00006	CHITRESH	1992-11-06

25. Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student. Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT c.firstname,c.customer_city,a.account_number FROM
```

```
Customer_master c JOIN account_master a ON a.customer_number=c.customer_number
```

```
WHERE c.occupation NOT IN ('Service','Student','Business')
```

```
ORDER BY c.firstname,a.account_number;
```

firstname	customer_city	account_number
PARUL	DELHI	A00010

AIRLINES

```
create database flight;
```

```
use flight;
```

```
CREATE TABLEair_credit_card_details
```

```
(
```

```
profile_id VARCHAR(10) NOT NULL,
```

```
card_number BIGINT,  
card_type VARCHAR(45),  
expiration_month INT,  
expiration_year INT  
);
```

```
CREATE TABLE air_passenger_profile  
(  
profile_id VARCHAR(10) NOT NULL ,  
password VARCHAR(45) NULL ,  
first_name VARCHAR(45) NULL ,  
last_name VARCHAR(45) NULL ,  
address VARCHAR(45) NULL ,  
mobile_number BIGINT NULL ,  
email_id VARCHAR(45) NULL  
);
```

```
CREATE TABLE air_ticket_info  
(  
ticket_id VARCHAR(45) NOT NULL ,  
profile_id VARCHAR(10) NULL ,  
flight_id VARCHAR(45) NULL ,  
flight_departure_date DATE NULL ,  
status VARCHAR(45) NULL  
);
```

```
CREATE TABLE air_flight_details  
(  
flight_id VARCHAR(45) NOT NULL ,
```

```
flight_departure_date DATE NULL ,  
price DECIMAL(10,2) NULL ,  
available_seats INT NULL  
);
```

```
CREATE TABLEair_flight  
(  
flight_id VARCHAR(45) NOT NULL ,  
airline_id VARCHAR(45) NULL ,  
airline_name VARCHAR(45) NULL ,  
from_location VARCHAR(45) NULL ,  
to_location VARCHAR(45) NULL ,  
departure_time TIME NULL ,  
arrival_time TIME NULL ,  
duration TIME NULL ,  
total_seats INT NULL  
);
```

```
INSERT INTO air_credit_card_details VALUES  
(1, 622098761234, 'debit', 5, 2013),  
(2, 652362563625, 'credit', 1, 2013),  
(1, 765432345678, 'credit', 2, 2013),  
(3, 654378561234, 'debit', 6, 2013),  
(4, 625417895623, 'debit', 2, 2013),  
(5, 865478956325, 'debit', 3, 2013),  
(6, 789563521457, 'credit', 4, 2013),  
(2, 543267895432, 'credit', 8, 2013),  
(1, 256369856321, 'debit', 1, 2013);
```

INSERT INTO air_flight VALUES

```
(3173, 'MH370',      'abc', 'hyderabad',  'chennai',      '06:30:00',    '07:15:00',
      '0:45:00',    100),
(3178, 'MH17',      'def', 'chennai',      'hyderabad',    '08:00:00',    '09:00:00',
      '1:00:00',    200),
(3172, 'AR342',      'fgh', 'kolkata',      'chennai',      '11:30:00',    '13:00:00',
      '1:30:00',    100),
(3071, 'JT564', 'jkl', 'chennai',      'delhi', '08:00:00',    '10:00:00',    '2:00:00',    100),
(3170, 'DT345',      'xyz', 'delhi', 'kolkata',      '21:00:00',    '22:30:00',    '1:30:00',
      100),
(3175, 'MJ654',      'abc', 'chennai',      'hyderabad',    '15:00:00',    '16:00:00',
      '1:00:00',    200),
(3176, 'MH370',      'def', 'kochi', 'chennai',      '18:00:00',    '19:05:00',    '1:05:00',
      100),
(3177, 'MH45',      'fgh', 'delhi', 'kochi', '19:00:00',    '21:00:00',    '2:00:00',    200),
(3174, 'MH321',      'xyz', 'kolkata',      'delhi', '0:00:00',     '2:00:00',     '2:00:00',
      100),
(3179, 'JT435', 'abc', 'chennai',      'kolkata',      '14:00:00',    '15:00:00',    '1:00:00',
      100),
(3180, 'JT456', 'ijk', 'kolkata',      'kochi', '5:00:00',     '5:45:00',     '0:45:00',    200);
```

INSERT INTO air_flight_details VALUES

```
(3170, '2013-02-14', 1000, 10),
(3171, '2013-03-15', 5000, 0),
(3172, '2013-02-05', 3000, 32),
(3173, '2013-04-07', 2000, 12),
(3174, '2013-04-05', 3800, 3),
(3175, '2013-05-25', 3500, 10),
(3176, '2013-03-14', 8000, 2),
```



```
(3177, '2013-06-15', 1500, 0),  
(3178, '2013-05-06', 3000, 5),  
(3179, '2013-04-03', 4000, 15),  
(3180, '2013-04-02', 3000, 14);
```

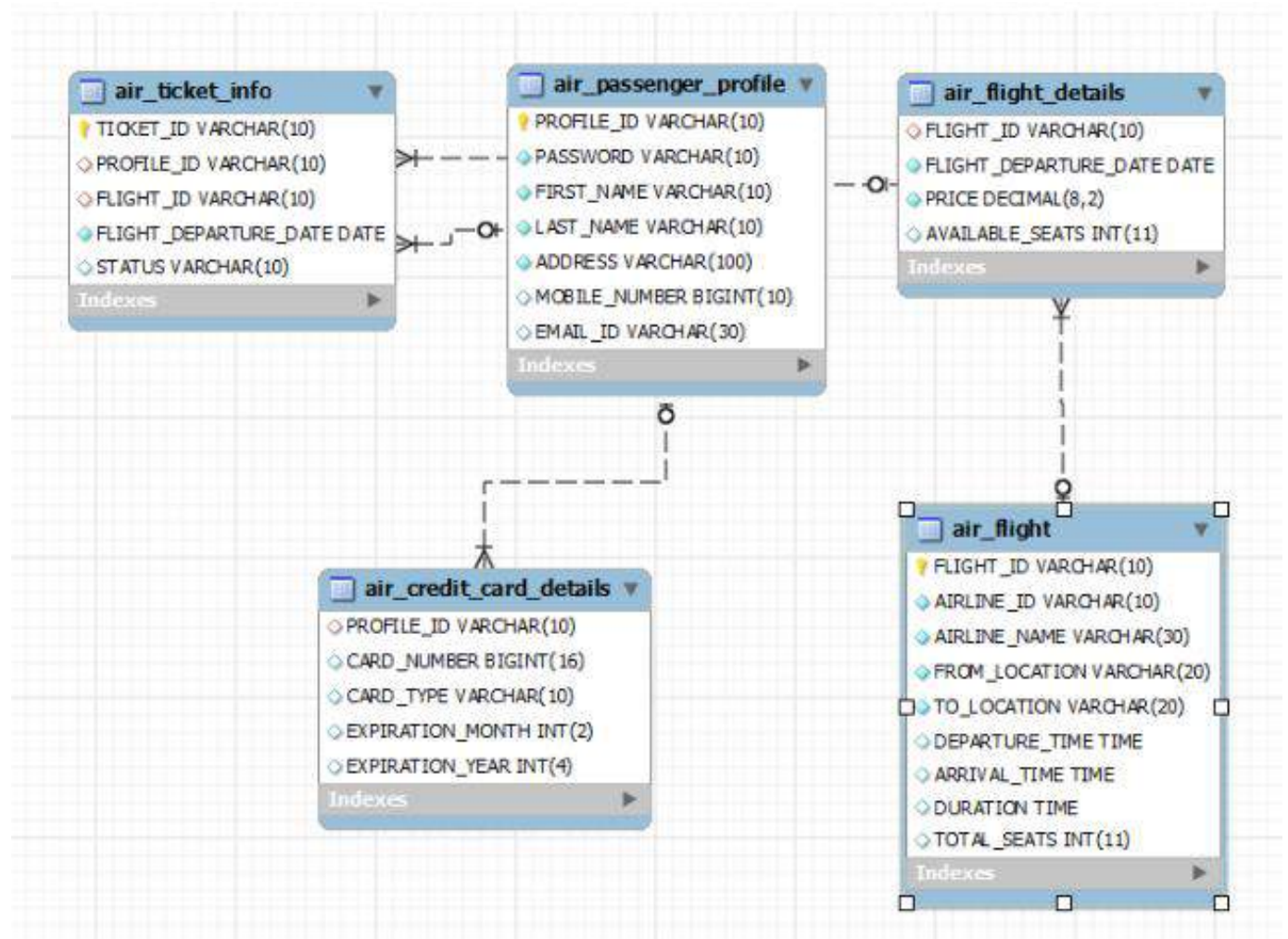
INSERT INTO air_ticket_info VALUES

```
(1, 1, 3178, '2013-05-06', 'delayed'),  
(2, 5, 3179, '2013-04-03', 'on time'),  
(2, 4, 3180, '2013-04-02', 'on time'),  
(1, 2, 3177, '2013-06-15', 'on time'),  
(1, 3, 3176, '2013-03-14', 'on time'),  
(3, 1, 3171, '2013-03-15', 'on time'),  
(4, 4, 3172, '2013-02-06', 'delayed'),  
(5, 2, 3178, '2013-06-05', 'on time'),  
(4, 3, 3171, '2013-03-15', 'on time'),  
(5, 1, 3175, '2013-05-25', 'on time'),  
(6, 3, 3177, '2013-06-15', 'on time');
```

INSERT INTO air_passenger_profile VALUES

```
(1, 'godbless', 'John', 'Stuart', 'Street 21, Near Bus Stop-Hyderabad-432126',  
9865263251, 'john@gmail.com'),  
(2, 'heyyaa', 'Robert', 'Clive', 'Sector 3, Technopolis-Kolkata-700102',  
9733015875, 'robert@yahoo.com'),  
(3, 'hello123', 'Raj', 'Sharma', 'House No. 3, Anna Nagar-Kochi-452314',  
9775470232, 'raj3452@hotmail.com'),  
(4, 'yesboss', 'Sanjay', 'Mittal', '21 Cauunaught Place-Delhi-144985',  
9856856321, 'sanjay@yahoo.com');
```

(5, 'imhere', 'Tony', 'Stark', '51A, Greems Lane-Chennai-144587',
9832015785, 'tony@gmail.com');



AIR TICKET INFO

ticket_id	profile_id	flight_id	flight_departure_date	status
1	1	3178	2013-05-06	delayed
2	5	3179	2013-04-03	on time
2	4	3180	2013-04-02	on time
1	2	3177	2013-06-15	on time
1	3	3176	2013-03-14	on time
3	1	3171	2013-03-15	on time
4	4	3172	2013-02-06	delayed
5	2	3178	2013-06-05	on time
4	3	3171	2013-03-15	on time
5	1	3175	2013-05-25	on time
6	3	3177	2013-06-15	on time

AIR PASSENGER DETAILS

profile_id	password	first_name	last_name	address	mobile_number	email_id
1	godbless	John	Stuart	Street 21, Near Bus Stop-Hyderabad-432126	9865263251	john@gmail.com
2	heyyaa	Robert	Clive	Sector 3, Technopolis-Kolkata-700102	9733015875	robert@yahoo.com
3	hello123	Raj	Shama	House No. 3, Anna Nagar-Kochi-452314	9775470232	raj3452@hotmail...
4	yesboss	Sanjay	Mittal	21 Cauunaught Place-Delhi-144985	9856856321	sanjay@yahoo.c...
5	imhere	Tony	Stark	51A, Greams Lane-Chennai-144587	9832015785	tony@gmail.com

AIR FLIGHT DETAILS

flight_id	flight_departure_date	price	available_seats
3170	2013-02-14	1000.00	10
3171	2013-03-15	5000.00	0
3172	2013-02-05	3000.00	32
3173	2013-04-07	2000.00	12
3174	2013-04-05	3800.00	3
3175	2013-05-25	3500.00	10
3176	2013-03-14	8000.00	2
3177	2013-06-15	1500.00	0
3178	2013-05-06	3000.00	5
3179	2013-04-03	4000.00	15
3180	2013-04-02	3000.00	14

AIR CREDIT CARD DETAILS

profile_id	card_number	card_type	expiration_month	expiration_year
1	622098761234	debit	5	2013
2	652362563625	credit	1	2013
1	765432345678	credit	2	2013
3	654378561234	debit	6	2013
4	625417895623	debit	2	2013
5	865478956325	debit	3	2013
6	789563521457	credit	4	2013
2	543267895432	credit	8	2013
1	256369856321	debit	1	2013

AIR FLIGHT

flight_id	airline_id	airline_name	from_location	to_location	departure_time	arrival_time	duration	total_seats
3170	DT345	xyz	delhi	kolkata	21:00:00	22:30:00	01:30:00	100
3171	JT564	jkl	chennai	delhi	08:00:00	10:00:00	02:00:00	100
3172	AR342	fgh	kolkata	chennai	11:30:00	13:00:00	01:30:00	100
3173	MH370	abc	hyderabad	chennai	06:30:00	07:15:00	00:45:00	100
3174	MH321	xyz	kolkata	delhi	00:00:00	02:00:00	02:00:00	100
3175	MJ654	abc	chennai	hyderabad	15:00:00	16:00:00	01:00:00	200
3176	MH370	def	kochi	chennai	18:00:00	19:05:00	01:05:00	100
3177	MH45	fgh	delhi	kochi	19:00:00	21:00:00	02:00:00	200
3178	MH17	def	chennai	hyderabad	08:00:00	09:00:00	01:00:00	200
3179	JT435	abc	chennai	kolkata	14:00:00	15:00:00	01:00:00	100
3180	JT456	ijk	kolkata	kochi	05:00:00	05:45:00	00:45:00	200

QUERIES

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id,From_location,To_Location,Month Name as "Month_Name" and average price as "Average_Price". Display the records sorted in ascending order based on flight id and then by Month Name.

```
SELECT f.flight_id,f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
AVG(price) Average_Price FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id WHERE f.airline_name = 'abc'
```

GROUP BY f.flight_id,f.from_location,f.to_location,Month_Name

ORDER BY f.flight_id, Month_Name;

flight_id	from_location	to_location	Month_Name	Average_Price
3173	hyderabad	chennai	April	2000.000000
3175	chennai	hyderabad	May	3500.000000
3179	chennai	kolkata	April	4000.000000

2. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as "Month_Name" and number of flight services as "No_of_Services". Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight. The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name.

```
SELECT f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
count(af.flight_departure_date) No_of_Services
FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id
GROUP BY f.from_location,f.to_location,Month_Name
ORDER BY f.from_location,f.to_Location,Month_Name;
```

from_location	to_location	Month_Name	No_of_Services
chennai	delhi	March	1
chennai	hyderabad	May	2
chennai	kolkata	April	1
delhi	kochi	June	1
delhi	kolkata	February	1
hyderabad	chennai	April	1
kochi	chennai	March	1
kolkata	chennai	February	1
kolkata	delhi	April	1
kolkata	kochi	April	1

3. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer's first_name, Address and Number of tickets

booked as “No_of_Tickets” Display the records sorted in ascending order based on customer's first name.

```
SELECT ap.profile_id,ap.first_name,ap.address,count(ati.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id and af.airline_name='abc'
GROUP BY ap.profile_id,ap.first_name,ap.address HAVING count(ati.ticket_id)<=ALL
(SELECT count(ticket_id)
FROM air_ticket_info GROUP BY profile_id)
ORDER BY ap.first_name;
```

profile_id	first_name	address	No_of_Tickets
1	John	Street 21, Near Bus Stop-Hyderabad-432126	1
5	Tony	51A, Greams Lane-Chennai-144587	1

4. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as “No_of_Tickets”.Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

```
SELECT ap.profile_id,ap.first_name,ap.last_name,af.flight_id,ati.flight_departure_date,
count(ati.profile_id) No_of_Tickets FROM
air_ticket_info ati JOIN air_passenger_profile ap ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.flight_id,ati.profile_id
ORDER BY ap.profile_id,af.flight_id,ati.flight_departure_date;
```

profile_id	first_name	last_name	flight_id	flight_departure_date	No_of_Tickets
1	John	Stuart	3175	2013-05-25	1
1	John	Stuart	3178	2013-05-06	1
2	Robert	Clive	3178	2013-06-05	1

5. Write a query to display flight id,from location, to location and ticket price of flights whose departure is in the month of april.Display the records sorted in ascending order based on flight id and then by from location.

```
SELECT af.flight_id,af.from_location,af.to_location,afd.price FROM
```

```
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
and month(afd.flight_departure_date)='04'

ORDER BY af.flight_id,af.from_location;
```

flight_id	from_location	to_location	price
3173	hyderabad	chennai	2000.00
3174	kolkata	delhi	3800.00
3179	chennai	kolkata	4000.00
3180	kolkata	kochi	3000.00

6. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight_id, from_location, to_location and Average price as "Price". Display the records sorted in ascending order based on flight id and then by from_location and then by to_location.

```
SELECT af.flight_id,af.from_location,af.to_location,avg(afd.price) Average_Price FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id

GROUP BY af.flight_id

ORDER BY af.flight_id,af.from_location,af.to_location;
```

flight_id	from_location	to_location	Average_Price
3170	delhi	kolkata	1000.000000
3171	chennai	delhi	5000.000000
3172	kolkata	chennai	3000.000000
3173	hyderabad	chennai	2000.000000
3174	kolkata	delhi	3800.000000
3175	chennai	hyderabad	3500.000000
3176	kochi	chennai	8000.000000
3177	delhi	kochi	1500.000000
3178	chennai	hyderabad	3000.000000
3179	chennai	kolkata	4000.000000
3180	kolkata	kochi	3000.000000

7. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile_id, customer_name (combine first_name & last_name with comma in

b/w), address of the customer. Give an alias to the name as customer_name. Hint: Query should fetch unique customers irrespective of multiple tickets booked. Display the records sorted in ascending order based on profile id.

```
SELECT ap.profile_id,concat(ap.first_name,',',ap.last_name) customer_name,ap.address FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.profile_id
ORDER BY ap.profile_id;
```

profile_id	Customer_name	address
1	John,Stuart	Street 21, Near Bus Stop-Hyderabad-432126
2	Robert,Clive	Sector 3, Technopolis-Kolkata-700102

8. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.

```
SELECT profile_id FROM air_ticket_info
group by profile_id
having count(ticket_id)>=all(select count(ticket_id)
from air_ticket_info
group by profile_id) order by profile_id;
```

profile_id
1
3

9. Write a query to display the total number of tickets as “No_of_Tickets” booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets. Display only the flights in which atleast 1 ticket is booked. Display the records sorted in ascending order based on flight id.

```
SELECT f.flight_id,f.from_location,f.to_location,COUNT(t.ticket_id) AS No_of_Tickets
FROM air_ticket_info t JOIN air_flight f
ON f.flight_id = t.flight_id where AIRLINE_NAME = 'abc' GROUP by
f.flight_id,f.from_location,f.to_location
```


having count(t.ticket_id)>=1

ORDER by f.flight_id;

flight_id	from_location	to_location	No_of_Tickets
3175	chennai	hyderabad	1
3179	chennai	kolkata	1

10. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as “No_of_Services” and the cost as “Total_Price” in the same order. Order the result by Total Price in descending order and then by flight_id in descending order.Hint:The number of services can be calculated from the number of scheduled departure dates of the flight

SELECT flight_id,count(flight_departure_date) No_of_services,sum(price) Total_Price FROM

air_flight_details GROUP BY flight_id

ORDER BY Total_price DESC,flight_id DESC;

flight_id	No_of_services	Total_Price
3176	1	8000.00
3171	1	5000.00
3179	1	4000.00
3174	1	3800.00
3175	1	3500.00
3180	1	3000.00
3178	1	3000.00
3172	1	3000.00
3173	1	2000.00
3177	1	1500.00
3170	1	1000.00

11. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as “No_of_Passengers” in the same order.Display the records sorted in ascending order based on flight id and then by flight departure date.

SELECT flight_id,flight_departure_date,count(ticket_id) No_of_passengers FROM

air_ticket_info GROUP BY flight_id,flight_departure_date

ORDER BY flight_id,flight_departure_date;

flight_id	flight_departure_date	No_of_passengers
3171	2013-03-15	2
3172	2013-02-06	1
3175	2013-05-25	1
3176	2013-03-14	1
3177	2013-06-15	2
3178	2013-05-06	1
3178	2013-06-05	1
3179	2013-04-03	1
3180	2013-04-02	1

12. Write a query to display profile id of passenger(s) who booked minimum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.

```
SELECT profile_id FROM air_ticket_info
GROUP BY profile_id HAVING count(ticket_id)<=ALL
(SELECT count(ticket_id) FROM air_ticket_info GROUP BY profile_id)
ORDER BY profile_id;
```

profile_id
5

13. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI. Display the records sorted in ascending order based on profile id.

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.mobile_number,ap.email_id FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON ati.flight_id=af.flight_id
WHERE af.from_location='Hyderabad' and af.to_location='Chennai'
ORDER BY profile_id;
```

profile_id	first_name	mobile_number	email_id
------------	------------	---------------	----------

14. Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger's profile_id, first_name,last_name, flight_id, flight_departure_date, actual departure time , actual arrival time , delayed departure time as "Delayed_Departure_Time", delayed arrival time as "Delayed_Arrival_Time" Hint: Distinct Profile ID should be displayed irrespective of multiple tickets booked by the same profile.Display the records sorted in ascending order based on passenger's profile id.

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.last_name,ati.flight_id,ati.flight_departure_date,
af.departure_time,af.arrival_time,
addtime(af.departure_time,'01:00:00') Delayed_Departure_Time,
addtime(af.arrival_time,'01:00:00') Delayed_Arrival_Time FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
and ati.flight_departure_date='2013-05-06'
ORDER BY profile_id;
```

profile_id	first_name	last_name	flight_id	flight_departure_date	departure_time	arrival_time	Delayed_Departure_Time	Delayed_Arrival_Time
1	John	Stuart	3178	2013-05-06	08:00:00	09:00:00	09:00:00	10:00:00

15. Write a query to display the number of tickets as “No_of_Tickets” booked by Kochi Customers. The Query should display the Profile_Id, First_Name, Base_Location and number of tickets booked.Hint: Use String functions to get the base location of customer from their Address and give alias name as “Base_Location”Display the records sorted in ascending order based on customer first name.

```
SELECT ap.profile_id,ap.first_name,
substring_index(substring_index(ap.address,'-',2),'-',-1) Base_Location,
count(ati.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ati.profile_id=ap.profile_id
WHERE ap.address LIKE '%Kochi%'
ORDER BY ap.first_name;
```

profile_id	first_name	Base_Location	No_of_Tickets
3	Raj	Kochi	3

16. Write a query to display the flight_id, from_location, to_location, number of Services as “No_of_Services” offered in the month of May.

```
SELECT af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) No_of_services
FROM
```

```
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
```

```
WHERE month(flight_departure_date)='05'
```

```
GROUP BY af.flight_id,af.from_location,af.to_location
```

```
ORDER BY af.flight_id;
```

flight_id	from_location	to_location	No_of_services
3175	chennai	hyderabad	1
3178	chennai	hyderabad	1

17. Write a query to display profile id,last name,mobile number and email id of passengers whose base location is chennai.Display the records sorted in ascending order based on profile id.

```
SELECT profile_id, last_name, mobile_number, email_id
```

```
FROM air_passenger_profile
```

```
WHERE address LIKE '%Chennai%'
```

```
ORDER BY profile_id;
```

profile_id	last_name	mobile_number	email_id
5	Stark	9832015785	tony@gmail.com

18. Write a query to display number of flights between 6.00 AM and 6.00 PM from chennai. Hint Use FLIGHT_COUNT as alias name.

```
SELECT count(flight_id) FLIGHT_COUNT FROM air_flight
```

```
WHERE from_location='CHENNAI'
```

```
AND departure_time BETWEEN '06:00:00' AND '18:00:00';
```

FLIGHT_COUNT

19. Write a query to display unique profile id,first name , email id and contact number of passenger(s) who travelled on flight with id 3178. Display the records sorted in ascending order based on first name.

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.email_id,ap.mobile_number FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
WHERE ati.flight_id='3178'
ORDER BY ap.first_name;
```

profile_id	first_name	email_id	mobile_number
1	John	john@gmail.com	9865263251
2	Robert	robert@yahoo.com	9733015875

20. Write a query to display flight id,departure date,flight type of all flights. Flight type can be identified based on the following rules : if ticket price is less than 3000 then 'AIR PASSENGER',ticket price between 3000 and less than 4000 'AIR BUS' and ticket price between 4000 and greater than 4000 then 'EXECUTIVE PASSENGER'. Hint use FLIGHT_TYPE as alias name.Display the records sorted in ascendeing order based on flight_id and then by departure date.

```
SELECT flight_id,flight_departure_date,
case when price<3000 then 'AIR PASSENGER'
      when price>=3000 and price<4000 then 'AIR BUS'
      when price>=4000 then 'EXECUTIVE PASSENGER'
end FLIGHT_TYPE FROM air_flight_details
ORDER BY flight_id,flight_departure_date;
```

flight_id	flight_departure_date	FLIGHT_TYPE
3170	2013-02-14	AIR PASSENGER
3171	2013-03-15	EXECUTIVE PASSENGER
3172	2013-02-05	AIR BUS
3173	2013-04-07	AIR PASSENGER
3174	2013-04-05	AIR BUS
3175	2013-05-25	AIR BUS
3176	2013-03-14	EXECUTIVE PASSENGER
3177	2013-06-15	AIR PASSENGER
3178	2013-05-06	AIR BUS
3179	2013-04-03	EXECUTIVE PASSENGER
3180	2013-04-02	AIR BUS

21. Write a query to display the credit card type and no of credit cards used on the same type. Display the records sorted in ascending order based on credit card type. Hint: Use CARD_COUNT AS Alias name for no of cards.

```
SELECT card_type, count(card_type) Card_Count FROM air_credit_card_details
GROUP BY card_type ORDER BY card_type;
```

card_type	Card_Count
credit	4
debit	5

22. Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com. The Serial No will be the last three digits of profile ID. Hint: Use SERIAL_NO as Alias name for serial number. Display the records sorted in ascending order based on name.

```
SELECT substring(profile_id,-3) SERIAL_NO, first_name, mobile_number, email_id FROM
air_passenger_profile
WHERE email_id LIKE '%@gmail.com'
ORDER BY first_name;
```

SERIAL_NO	first_name	mobile_number	email_id
	John	9865263251	john@gmail.com
	Tony	9832015785	tony@gmail.com

23. Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight_id, from_location, to_location, least number of Services as "No_of_Services" Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight if there are multiple flights, display them sorted in ascending order based on flight id.

```
SELECT afd.flight_id, af.from_location, af.to_location, count(afd.flight_id) No_of_Services
FROM air_flight_details afd JOIN air_flight af ON af.flight_id=afd.flight_id
WHERE monthname(afd.flight_departure_date)='May'
GROUP BY afd.flight_departure_date HAVING count(afd.flight_id) <=
ALL(SELECT count(flight_id) FROM air_flight_details
WHERE monthname(flight_departure_date)='May'
GROUP BY flight_departure_date)
```

ORDER BY flight_id;

flight_id	from_location	to_location	No_of_Services
3175	chennai	hyderabad	1
3178	chennai	hyderabad	1

24. Write a query to display the flights available in Morning, AfterNoon, Evening& Night. The Query should display the Flight_Id, From_Location, To_Location , Departure_Time, time of service as "Time_of_Service". Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning, 12:00:01 to 18:00:00 Hrs -AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - NightDisplay the records sorted in ascending order based on flight id.

```
SELECT flight_id,from_location,to_location,Departure_Time,
CASE
WHEN departure_time BETWEEN ('05:00:01') AND ('12:00:00')
THEN 'Morning'
WHEN departure_time BETWEEN ('12:00:01') AND ('18:00:00')
THEN 'AfterNoon'
WHEN departure_time BETWEEN ('18:00:01') AND ('24:00:00')
THEN 'Evening'
WHEN departure_time='00:00:00'
THEN 'Evening'
WHEN departure_time BETWEEN ('00:00:01') AND ('05:00:00')
THEN 'Night'
END Time_of_Service
FROM air_flight
order by flight_id;
```

flight_id	from_location	to_location	Departure_Time	Time_of_Service
3170	delhi	kolkata	21:00:00	Evening
3171	chennai	delhi	08:00:00	Moming
3172	kolkata	chennai	11:30:00	Moming
3173	hyderabad	chennai	06:30:00	Moming
3174	kolkata	delhi	00:00:00	Evening
3175	chennai	hyderabad	15:00:00	AfterNoon
3176	kochi	chennai	18:00:00	AfterNoon
3177	delhi	kochi	19:00:00	Evening
3178	chennai	hyderabad	08:00:00	Moming
3179	chennai	kolkata	14:00:00	AfterNoon
3180	kolkata	kochi	05:00:00	Night

25. Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No_of_Flights”. Hint: Get the distinct from location and to location. Display the records sorted in ascending order based on from location.

```
SELECT from_location, count(flight_id) No_of_Flights FROM
air_flight GROUP BY from_location
ORDER BY from_location;
```

from_location	No_of_Flights
chennai	4
delhi	2
hyderabad	1
kochi	1
kolkata	3

26. Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight_id, from_location, To_location, flight_departure_date and the number of passengers as “No_of_Passengers”. Hint: The Number of passengers inclusive of all the tickets booked with single profile id. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT af.flight_id, af.from_location, af.to_location, ati.flight_departure_date,
count(ati.ticket_id) No_of_Passengers FROM
air_flight af JOIN air_ticket_info ati ON af.flight_id=ati.flight_id
GROUP BY af.flight_id, af.from_location, af.to_location, ati.flight_departure_date
```


ORDER BY af.flight_id,ati.flight_departure_date;

flight_id	from_location	to_location	flight_departure_date	No_of_Passengers
3171	chennai	delhi	2013-03-15	2
3172	kolkata	chennai	2013-02-06	1
3175	chennai	hyderabad	2013-05-25	1
3176	kochi	chennai	2013-03-14	1
3177	delhi	kochi	2013-06-15	2
3178	chennai	hyderabad	2013-05-06	1
3178	chennai	hyderabad	2013-06-05	1
3179	chennai	kolkata	2013-04-03	1
3180	kolkata	kochi	2013-04-02	1

27. Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight_Id, From_Location, To_Location,Total_Seats, seats booked as “No_of_Seats_Booked” .Display the records sorted in ascending order based on flight id and then by No_of_Seats_Booked.

```
SELECT af.flight_id,af.from_location,af.to_location,af.total_seats,
(af.total_seats-afd.available_seats) No_of_Seats_Booked FROM
air_flight_details afd JOIN air_flight af ON afd.flight_id=af.flight_id
WHERE (af.total_seats-afd.available_seats)>(af.total_seats*0.1)
ORDER BY flight_id,No_of_Seats_Booked;
```

flight_id	from_location	to_location	total_seats	No_of_Seats_Booked
3170	delhi	kolkata	100	90
3171	chennai	delhi	100	100
3172	kolkata	chennai	100	68
3173	hyderabad	chennai	100	88
3174	kolkata	delhi	100	97
3175	chennai	hyderabad	200	190
3176	kochi	chennai	100	98
3177	delhi	kochi	200	200
3178	chennai	hyderabad	200	195
3179	chennai	kolkata	100	85
3180	kolkata	kochi	200	186

28. Write a query to display the Flight_Id, Flight_Departure_Date, From_Location,To_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

```
SELECT af.flight_Id,afd.flight_Departure_Date,af.From_Location,af.To_Location,af.duration
```

```
FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
WHERE af.duration<'01:10:00';
```

flight_id	flight_Departure_Date	From_Location	To_Location	duration
3173	2013-04-07	hyderabad	chennai	00:45:00
3175	2013-05-25	chennai	hyderabad	01:00:00
3176	2013-03-14	kochi	chennai	01:05:00
3178	2013-05-06	chennai	hyderabad	01:00:00
3179	2013-04-03	chennai	kolkata	01:00:00
3180	2013-04-02	kolkata	kochi	00:45:00

29. Write a query to display the flight_id, from_location,to_location,number of services as “No_of_Services” , average ticket price as “Average_Price” whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

```
SELECT afd.flight_id,af.from_location,af.to_location,
count(afd.flight_departure_date) No_of_Service, avg(price) Average_Price
FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id,af.from_location,af.to_location
HAVING avg(price)>(SELECT avg(price) FROM air_flight_details)
ORDER BY average_price;
```

flight_id	from_location	to_location	No_of_Service	Average_Price
3175	chennai	hyderabad	1	3500.000000
3174	kolkata	delhi	1	3800.000000
3179	chennai	kolkata	1	4000.000000
3171	chennai	delhi	1	5000.000000
3176	kochi	chennai	1	8000.000000

MOVIE

CREATE DATABASE video;USE video;

Create table CUSTOMER_MASTER

(CUSTOMER_ID Varchar(10),CUSTOMER_NAME Varchar(30) NOT NULL,CONTACT_NO BIGINT(10),CONTACT_ADD Varchar(20),DATE_OF_REGISTRATION Date NOT NULL,AGE Varchar(15)NOT NULL,Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID));

Create table LIBRARY_CARD_MASTER

(CARD_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT BIGINT(50),NUMBER_OF_YEARS bigint(10) NOT NULL,Constraint MT_cts2 PRIMARY KEY(CARD_ID));

Create table MOVIES_MASTER

(MOVIE_ID Varchar(10), MOVIE_NAME Varchar(50) NOT NULL,RELEASE_DATE Varchar(30) NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION VARCHAR(10) NOT NULL, MOVIE_TYPE Varchar(3),MOVIE_CATEGORY VARCHAR(20) NOT NULL,DIRECTOR VARCHAR(20) NOT NULL,

LEAD_ROLE_1 Varchar(3) NOT NULL,LEAD_ROLE_2 VARCHAR(4) NOT NULL,RENT_COST BIGINT(10),Constraint MT_cts4 PRIMARY KEY(MOVIE_ID));

Create table CUSTOMER_CARD_DETAILS

(CUSTOMER_ID Varchar(10),CARD_ID VARCHAR(10),ISSUE_DATE DATE NOT NULL,Constraint MT_cts3 PRIMARY KEY(CUSTOMER_ID),Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID) References CUSTOMER_MASTER(CUSTOMER_ID),Constraint MT_CTS42 FOREIGN KEY(CARD_ID) References LIBRARY_CARD_MASTER(CARD_ID));

Create table CUSTOMER_ISSUE_DETAILS

(ISSUE_ID Varchar(10) NOT NULL,CUSTOMER_ID Varchar(10) NOT NULL,MOVIE_ID VARCHAR(10), ISSUE_DATE Date NOT NULL,RETURN_DATE Date NOT NULL,

ACTUAL_DATE_RETURN Date NOT NULL,Constraint MT_cts5 PRIMARY KEY(ISSUE_ID),Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References CUSTOMER_MASTER(CUSTOMER_ID), Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID) References MOVIES_MASTER(MOVIE_ID));

Insert into CUSTOMER_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS007', 'NAME7', 3210987654,'ADD7', '2012-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS008', 'NAME8', 2109876543,'ADD8', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS009', 'NAME9', NULL,'ADD9', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS010', 'NAM10', 9934567890,'ADD10', '2013-02-12', '21');

Insert into CUSTOMER_MASTER Values('CUS011', 'NAM11', 9875678910,'ADD11', '2013-02-12', '21');

Insert into LIBRARY_CARD_MASTER Values('CR001', 'Silver', 200, 5);

Insert into LIBRARY_CARD_MASTER Values('CR002', 'Gold', 400, 9);

Insert into LIBRARY_CARD_MASTER Values('CR003', 'Platinum', 600, 8);

Insert into LIBRARY_CARD_MASTER Values('CR004', 'VISA', 800, 7);

Insert into LIBRARY_CARD_MASTER Values('CR005', 'CREDIT', 1200, 6);

Insert into MOVIES_MASTER Values('MV001', 'DIEHARD', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR1','L1','L2',100);

Insert into MOVIES_MASTER Values('MV002', 'THE MATRIX', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR2','L1','L2',100);

Insert into MOVIES_MASTER Values('MV003', 'INCEPTION', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR3','L15','L2',100);

Insert into MOVIES_MASTER Values('MV004', 'DARK KNIGHT', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR4','L15','L2',100);

Insert into MOVIES_MASTER Values('MV005', 'OFFICE S', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR5','L12','L24',100);

Insert into MOVIES_MASTER Values('MV006', 'SHAWN OF DEAD', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR6','L1','L25',100);

Insert into MOVIES_MASTER Values('MV007', 'YOUNG FRANKEN', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR7','L1','L2',100);

Insert into MOVIES_MASTER Values('MV008', 'CAS', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR8','L1','L2',100);

Insert into MOVIES_MASTER Values('MV009', 'GWW', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR9','L1','L2',100);

Insert into MOVIES_MASTER Values('MV010', 'TITANIC', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR10','L1','L2',100);

Insert into MOVIES_MASTER Values('MV011', 'THE NOTE BOOK', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR11','L1','L2',100);

Insert into CUSTOMER_CARD_DETAILS Values('CUS001', 'CR001', '2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS002', 'CR002', '2012-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS003', 'CR002', '2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS004', 'CR003', '2013-05-13');

Insert into CUSTOMER_CARD_DETAILS Values('CUS005', 'CR003', '2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS001', 'CUS001', 'MV001', '2012-05-13', '2012-05-13','2012-05-13');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS002', 'CUS001', 'MV001', '2012-05-01', '2012-05-16','2012-05-16');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS003', 'CUS002', 'MV004', '2012-05-02', '2012-05-06', '2012-05-16');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS004', 'CUS002', 'MV004', '2012-04-03', '2012-04-16', '2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS005', 'CUS002', 'MV009', '2012-04-04', '2012-04-16', '2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS006', 'CUS003', 'MV002', '2012-03-30', '2012-04-15', '2012-04-20');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS007', 'CUS003', 'MV003', '2012-04-20', '2012-05-05', '2012-05-05');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS008', 'CUS003', 'MV005', '2012-04-21', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS009', 'CUS003', 'MV001', '2012-04-22', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS010', 'CUS003', 'MV009', '2012-04-22', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS011', 'CUS003', 'MV010', '2012-04-23', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS012', 'CUS003', 'MV010', '2012-04-24', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS013', 'CUS003', 'MV008', '2012-04-25', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS014', 'CUS004', 'MV007', '2012-04-26', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS015', 'CUS004', 'MV006', '2012-04-27', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS016', 'CUS004', 'MV006', '2012-04-28', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS017', 'CUS004', 'MV001', '2012-04-29', '2012-05-07', '2012-05-25');

Insert into CUSTOMER_ISSUE_DETAILS Values ('IS018', 'CUS010', 'MV008', '2012-04-24', '2012-05-07', '2012-05-25');



LEAD_ROLE_2	RENT_COST
L2	100
L2	100
L2	100
L2	100
L24	100
L25	100
L2	100
L2	100
L2	100
L2	100
L2	100
NULL	NULL

CUSTOMER MASTER

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADD	DATE_OF_REGISTRATION	AGE
CUS001	AMIT	9876543210	ADD1	2012-02-12	21
CUS002	ABDHUL	8765432109	ADD2	2012-02-12	21
CUS003	GAYAN	7654321098	ADD3	2012-02-12	21
CUS004	RADHA	6543210987	ADD4	2012-02-12	21
CUS005	GURU	NULL	ADD5	2012-02-12	21
CUS006	MOHAN	4321098765	ADD6	2012-02-12	21
CUS007	NAME7	3210987654	ADD7	2012-02-12	21
CUS008	NAME8	2109876543	ADD8	2013-02-12	21
CUS009	NAME9	NULL	ADD9	2013-02-12	21
CUS010	NAM10	9934567890	ADD10	2013-02-12	21
CUS011	NAM11	9875678910	ADD11	2013-02-12	21
NULL	NULL	NULL	NULL	NULL	NULL

LIBRARY CARD MASTER

CARD_ID	DESCRIPTION	AMOUNT	NUMBER_OF_YEARS
CR001	Silver	200	5
CR002	Gold	400	9
CR003	Platinum	600	8
CR004	VISA	800	7
CR005	CREDIT	1200	6
NULL	NULL	NULL	NULL

CUSTOMER CARD DETAILS

CUSTOMER_ID	CARD_ID	ISSUE_DATE
CUS001	CR001	2012-05-13
CUS002	CR002	2012-05-13
CUS003	CR002	2013-05-13
CUS004	CR003	2013-05-13
CUS005	CR003	2012-05-13
NULL	NULL	NULL

CUSTOMER ISSUE DETAILS

ISSUE_ID	CUSTOMER_ID	MOVIE_ID	ISSUE_DATE	RETURN_DATE	ACTUAL_DATE_RETURN
IS001	CUS001	MV001	2012-05-13	2012-05-13	2012-05-13
IS002	CUS001	MV001	2012-05-01	2012-05-16	2012-05-16
IS003	CUS002	MV004	2012-05-02	2012-05-06	2012-05-16
IS004	CUS002	MV004	2012-04-03	2012-04-16	2012-04-20
IS005	CUS002	MV009	2012-04-04	2012-04-16	2012-04-20
IS006	CUS003	MV002	2012-03-30	2012-04-15	2012-04-20
IS007	CUS003	MV003	2012-04-20	2012-05-05	2012-05-05
IS008	CUS003	MV005	2012-04-21	2012-05-07	2012-05-25
IS009	CUS003	MV001	2012-04-22	2012-05-07	2012-05-25
IS010	CUS003	MV009	2012-04-22	2012-05-07	2012-05-25
IS011	CUS003	MV010	2012-04-23	2012-05-07	2012-05-25
IS012	CUS003	MV010	2012-04-24	2012-05-07	2012-05-25
IS013	CUS003	MV008	2012-04-25	2012-05-07	2012-05-25
IS014	CUS004	MV007	2012-04-26	2012-05-07	2012-05-25
IS015	CUS004	MV006	2012-04-27	2012-05-07	2012-05-25
IS016	CUS004	MV006	2012-04-28	2012-05-07	2012-05-25
IS017	CUS004	MV001	2012-04-29	2012-05-07	2012-05-25
IS018	CUS010	MV008	2012-04-24	2012-05-07	2012-05-25
IS019	CUS011	MV009	2012-04-27	2012-05-07	2012-05-25
NULL	NULL	NULL	NULL	NULL	NULL

1. Write a query to display movie names and number of times that movie is issued to customers. In case movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order). The Alias name for the number of movies issued is ISSUE_COUNT.

```
SELECT m.MOVIE_NAME, count(ISSUE_ID) ISSUE_COUNT FROM
movies_master m LEFT JOIN customer_issue_details c ON m.MOVIE_ID=c.MOVIE_ID
GROUP BY m.movie_name
```

ORDER BY ISSUE_COUNT DESC,MOVIE_NAME;

MOVIE_NAME	ISSUE_COUNT
DIEHARD	4
GWW	3
CAS	2
DARK KNIGHT	2
SHAWN OF DEAD	2
TITANIC	2
INCEPTION	1
OFFICE S	1
THE MATRIX	1
YOUNG FRANKEN	1
THE NOTE BOOK	0

2. Write a query to display id,name,age,contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

```
SELECT CUSTOMER_ID,CUSTOMER_NAME,AGE,ifnull(
concat('+91-',substring(contact_no,1,3)),'-',
substring(contact_no,4,3)),'-',substring(contact_no,7)), 'N/A') CONTACT_ISD
FROM customer_master WHERE age>25 and year(date_of_registration)='2012'
ORDER BY age,CUSTOMER_NAME;
```

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
-------------	---------------	-----	-------------

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO_OF_MOVIES as alias name for number of movies.

```
SELECT MOVIE_CATEGORY,count(MOVIE_ID) NO_OF_MOVIES FROM
movies_master GROUP BY MOVIE_CATEGORY
ORDER BY NO_OF_MOVIES DESC,MOVIE_CATEGORY;
```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Write a query to display the number of customers having card with description "Gold card".

**
Hint: Use CUSTOMER_COUNT as alias name for number of customers**

```
SELECT count(c.customer_id) CUSTOMER_COUNT FROM
library_card_master l JOIN customer_card_details c ON l.CARD_ID=c.CARD_ID
WHERE description='Gold';
```

CUSTOMER_COUNT
2

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order. Use REGISTERED_YEAR as alias name for year of registration.

```
SELECT c.customer_id,c.customer_name,
year(c.DATE_OF_REGISTRATION) REGISTERED_YEAR,cd.card_id,cd.issue_date FROM
customer_master c JOIN customer_card_details cd ON c.customer_id=cd.customer_id
ORDER BY CUSTOMER_NAME DESC;
```

customer_id	customer_name	REGISTERED_YEAR	card_id	issue_date
CUS004	RADHA	2012	CR003	2013-05-13
CUS005	GURU	2012	CR003	2012-05-13
CUS003	GAYAN	2012	CR002	2013-05-13
CUS001	AMIT	2012	CR001	2012-05-13
CUS002	ABDHUL	2012	CR002	2012-05-13

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order by customer name.

```
SELECT ci.issue_id,ci.CUSTOMER_ID,c.CUSTOMER_NAME FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
WHERE customer_name LIKE 'R%' and ci.actual_date_return>ci.return_date
```

ORDER BY customer_name;

issue_id	CUSTOMER_ID	CUSTOMER_NAME
IS014	CUS004	RADHA
IS015	CUS004	RADHA
IS016	CUS004	RADHA
IS017	CUS004	RADHA

7. Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10. Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name.

```
SELECT c.CUSTOMER_ID,c.CUSTOMER_NAME,l.card_id,l.DESRIPTION,
concat('$',round(amount/52.42)) AMOUNT_DOLLAR FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
JOIN customer_card_details cc ON cc.customer_id=c.customer_id
JOIN library_card_master l ON cc.card_id=l.card_id
WHERE c.DATE_OF_REGISTRATION=ci.issue_date
ORDER BY customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	card_id	DESCRIPTION	AMOUNT_DOLLAR
-------------	---------------	---------	-------------	---------------

8. Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'. Display customer name in upper case. Hint: Use CUSTOMER_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.

```
SELECT CUSTOMER_ID,upper(CUSTOMER_NAME) CUSTOMER_NAME,contact_no,contact_add
FROM
customer_master WHERE contact_add LIKE '%Nagar' and
customer_id NOT IN (SELECT customer_id FROM customer_card_details)
and customer_id IN (SELECT customer_id FROM customer_issue_details)
```

ORDER BY CUSTOMER_NAME;

CUSTOMER_ID	CUSTOMER_NAME	contact_no	contact_add
-------------	---------------	------------	-------------

9. Write a query to display the movie id, movie name, release year, director name of movies acted by the lead actor 1 who acted maximum number of movies. Display the records sorted in ascending order based on movie name.

```
SELECT movie_id, movie_name, release_date, director FROM movies_master
WHERE lead_role_1 IN (SELECT lead_role_1 FROM
(SELECT lead_role_1, count(movie_id) ct FROM movies_master
GROUP BY lead_role_1) t WHERE t.ct >= ALL (SELECT count(movie_id)
FROM movies_master GROUP BY lead_role_1)) ORDER BY movie_name;
```

movie_id	movie_name	release_date	director
MV008	CAS	2012-05-13	DIR8
MV001	DIEHARD	2012-05-13	DIR1
MV009	GWW	2012-05-13	DIR9
MV006	SHAWN OF DEAD	2012-05-13	DIR6
MV002	THE MATRIX	2012-05-13	DIR2
MV011	THE NOTE BOOK	2012-05-13	DIR11
MV010	TITANIC	2012-05-13	DIR10
MV007	YOUNG FRANK...	2012-05-13	DIR7

**10. Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0.
Hint: Use MOVIE_COUNT as alias name for number of movies issued.**

```
SELECT c.customer_name, count(ci.movie_id) MOVIE_COUNT FROM
customer_master c LEFT JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
GROUP BY c.customer_id ORDER BY c.customer_name;
```

customer_name	MOVIE_COUNT
ABDHUL	3
AMIT	2
GAYAN	8
GURU	0
MOHAN	0
NAM10	1
NAM11	1
NAME7	0
NAME8	0
NAME9	0
RADHA	4

11. Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number. Serial number can be generated from the issue id , that is last two characters of issue id is the serial number. For Example Assume the issue id is I00005 then the serial number is 05 Hint: Alias name for serial number is 'SERIAL_NO'

```
SELECT substring(ci.issue_id,-2) SERIAL_NO,ci.issue_id,c.customer_id,c.customer_name,
m.movie_id,m.movie_name FROM customer_master c JOIN customer_issue_details ci
ON c.customer_id=ci.customer_id JOIN movies_master m ON m.movie_id=ci.movie_id
ORDER BY SERIAL_NO;
```

SERIAL_NO	issue_id	customer_id	customer_name	movie_id	movie_name
01	IS001	CUS001	AMIT	MV001	DIEHARD
02	IS002	CUS001	AMIT	MV001	DIEHARD
03	IS003	CUS002	ABDHUL	MV004	DARK KNIGHT
04	IS004	CUS002	ABDHUL	MV004	DARK KNIGHT
05	IS005	CUS002	ABDHUL	MV009	GWW
06	IS006	CUS003	GAYAN	MV002	THE MATRIX
07	IS007	CUS003	GAYAN	MV003	INCEPTION
08	IS008	CUS003	GAYAN	MV005	OFFICE S
09	IS009	CUS003	GAYAN	MV001	DIEHARD
10	IS010	CUS003	GAYAN	MV009	GWW
11	IS011	CUS003	GAYAN	MV010	TITANIC
12	IS012	CUS003	GAYAN	MV010	TITANIC
13	IS013	CUS003	GAYAN	MV008	CAS
14	IS014	CUS004	RADHA	MV007	YOUNG FRAN...
15	IS015	CUS004	RADHA	MV006	SHAWN OF D...
16	IS016	CUS004	RADHA	MV006	SHAWN OF D...
17	IS017	CUS004	RADHA	MV001	DIEHARD
18	IS018	CUS010	NAM10	MV008	CAS
19	IS019	CUS011	NAM11	MV009	GWW

12. Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

```
SELECT ci.issue_id, ci.issue_date, c.customer_id, c.customer_name, c.contact_no FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
and year(ci.issue_date)='2013' ORDER BY ci.issue_date DESC;
```

issue_id	issue_date	customer_id	customer_name	contact_no
----------	------------	-------------	---------------	------------

**13. Write a query to display movie id, movie name and actor names of movies which are not issued to any customers.
 Actors Name to be displayed in the below format. LEAD_ACTOR_ONE space ambersant space LEAD_ACTOR_TWO. Example: Assume lead**

actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria"Hint:Use ACTORS as alias name for actors name.
 Display the records in ascending order based on movie name.

```
SELECT movie_id,movie_name,concat(lead_role_1,' & ',lead_role_2) ACTOR FROM
movies_master
```

```
WHERE movie_id NOT IN (SELECT movie_id FROM customer_issue_details) ORDER BY
movie_name;
```

movie_id	movie_name	ACTOR
MV011	THE NOTE BOOK	L1 & L2

14.Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.

```
SELECT upper(director) DIRECTOR_NAME,movie_name,lead_role_1 FROM movies_master
GROUP BY director HAVING count(movie_id)>1 ORDER BY director,movie_name DESC;
```

DIRECTOR_NAME	movie_name	lead_role_1
---------------	------------	-------------

15.Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number.
 Hint:Use NO_OF_CUSTOMERS as alias name for number of customers.

```
SELECT count(customer_id) NO_OF_CUSTOMER FROM customer_master
WHERE contact_no is not null and year(date_of_registration)='2012';
```

NO_OF_CUSTOMER
6

16.Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.


```

SELECT c.customer_name,ifnull(c.contact_no,c.contact_add)
CONTACT_DETAILS,l.card_id,l.description FROM
customer_master c LEFT JOIN customer_card_details cc ON c.customer_id=cc.customer_id
LEFT JOIN library_card_master l ON l.card_id=cc.card_id
ORDER BY customer_name;

```

customer_name	CONTACT_DETAILS	card_id	description
ABDHUL	8765432109	CR002	Gold
AMIT	9876543210	CR001	Silver
GAYAN	7654321098	CR002	Gold
GURU	ADD5	CR003	Platinum
MOHAN	4321098765	NULL	NULL
NAM10	9934567890	NULL	NULL
NAM11	9875678910	NULL	NULL
NAME7	3210987654	NULL	NULL
NAME8	2109876543	NULL	NULL
NAME9	ADD9	NULL	NULL
RADHA	6543210987	CR003	Platinum

17. Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO_OF_TIMES as alias name for number of times

```

SELECT ci.customer_id,c.customer_name,count(ci.movie_id) NO_OF_TIMES FROM
customer_issue_details ci JOIN customer_master c ON c.customer_id=ci.customer_id
GROUP BY ci.customer_id,ci.movie_id HAVING count(movie_id)>1
ORDER BY customer_name DESC;

```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	2
CUS003	GAYAN	2
CUS001	AMIT	2
CUS002	ABDHUL	2

18. Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-

456-2345" format. Hint:Use NO_OF_MOVIES as alias name for number of movies column. Hint:Use CONTACT_ISD as alias name for contact number. Display the records sorted in ascending order based on customer name and then by movie category.

```
SELECT c.customer_id,c.customer_name,concat('+91-',substring(c.contact_no,1,3),'-',
substring(c.contact_no,4,3),'-',substring(c.contact_no,7)) CONTACT_ISD
,m.movie_category,count(cc.movie_id) NO_OF_MOVIES FROM customer_master c JOIN
customer_issue_details cc
ON c.customer_id=cc.customer_id JOIN movies_master m ON m.movie_id=cc.movie_id
GROUP BY c.customer_id,m.movie_category HAVING count(cc.movie_id)>1
ORDER BY customer_name,movie_category;
```

customer_id	customer_name	CONTACT_ISD	movie_category	NO_OF_MOVIES
CUS002	ABDHUL	+91-876-543-2109	ACTION	2
CUS001	AMIT	+91-987-654-3210	ACTION	2
CUS003	GAYAN	+91-765-432-1098	ACTION	3
CUS003	GAYAN	+91-765-432-1098	ROMANCE	4
CUS004	RADHA	+91-654-321-0987	COMEDY	3

19. Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies. For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

```
SELECT cid.customer_id , customer_name FROM customer_master cm JOIN
customer_issue_details cid ON cm.customer_id=cid.customer_id

GROUP BY customer_id , customer_name

HAVING count(movie_id)>=ALL(SELECT count(movie_id)

FROM customer_issue_details

GROUP BY customer_id)

UNION
```

```

SELECT cid.customer_id , customer_name FROM
customer_master cm JOIN customer_issue_details cid
ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)<=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id) ORDER BY customer_name;

```

customer_id	customer_name
CUS003	GAYAN
CUS010	NAM10
CUS011	NAM11

20. Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once. Hint: Use NO_OF_TIMES as alias name Display the records in ascending order based on customer name.

```

SELECT c.customer_id,c.customer_name,count(m.movie_id) NO_OF_TIMES FROM
customer_master c JOIN customer_issue_details cc ON c.customer_id=cc.customer_id
JOIN movies_master m ON m.movie_id=cc.movie_id
WHERE m.movie_category='Comedy'
GROUP BY c.customer_id HAVING count(m.movie_id)>1
ORDER BY customer_name;

```

customer_id	customer_name	NO_OF_TIMES
CUS004	RADHA	3

21. Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST. Display the records sorted in ascending order based on customer id

```

SELECT cid.customer_id, sum(m.rent_cost) TOTAL_COST FROM customer_issue_details cid
JOIN movies_master mm ON cid.movie_id=mm.movie_id GROUP BY cid.customer_id order by
customer_id;

```

customer_id	TOTAL_COST
CUS001	200
CUS002	300
CUS003	800
CUS004	400
CUS010	100
CUS011	100

LOAN

```
create database loan;
```

```
use loan;
```

```
CREATE TABLE loan_card_master
```

```
(  
    loan_id      varchar(6)    PRIMARY KEY,  
    loan_type    varchar(15),  
    duration_in_years  int(2)  
);
```

```
CREATE TABLE employee_master
```

```
(  
    employee_id    varchar(6)    PRIMARY KEY,  
    employee_name  varchar(20),  
    designation     varchar(25),  
    department     varchar(25),  
    gender         char(1),  
    date_of_birth  date,  
    date_of_joining  date  
);
```

```
CREATE TABLE item_master
```

```
(  
    item_id      varchar(6)    PRIMARY KEY,  
    item_description  varchar(25),
```

```
        issue_status      char(1),
        item_make         varchar(25),
        item_category     varchar(20),
        item_valuation    int(6)
);
```

```
CREATE TABLE employee_card_details
```

```
(
    employee_id          varchar(6)    REFERENCES employee_master,
    loan_id              varchar(6)    REFERENCES loan_card_master,
    card_issue_date      date
);
```

```
CREATE TABLE employee_issue_details
```

```
(
    issue_id             varchar(6)    PRIMARY KEY,
    employee_id          varchar(6)    REFERENCES employee_master,
    item_id              varchar(6)    REFERENCES item_master,
    issue_date           date,
    return_date          date
);
```

```
insert into loan_card_master values('L00001','Furniture',5);
```

```
insert into loan_card_master values('L00002','Stationary',0);
```

```
insert into loan_card_master values('L00003','Crockery',1);
```

```
insert into employee_issue_details values('ISS001','E00001','I00001','2012-02-03','2014-02-03');
insert into employee_issue_details values('ISS002','E00001','I00004','2012-02-03','2020-02-03');
insert into employee_issue_details values('ISS003','E00002','I00005','2013-01-03','2015-01-03');
insert into employee_issue_details values('ISS004','E00003','I00007','2010-07-04','2012-07-04');
insert into employee_issue_details values('ISS005','E00003','I00008','2010-07-04','2012-08-05');
insert into employee_issue_details values('ISS006','E00003','I00010','2012-03-14','2012-06-15');
insert into employee_issue_details values('ISS007','E00004','I00012','2013-04-14','2016-04-14');
insert into employee_issue_details values('ISS008','E00006','I00018','2012-08-18','2019-04-17');
insert into employee_issue_details values('ISS009','E00004','I00018','2013-04-18','2013-05-18');
```

```
insert into employee_master values('E00001','Ram','Manager','Finance','M','1973-12-01','2000-01-01');
```

```
insert into employee_master values('E00002','Abhay','Assistant Manager','Finance','M','1976-01-01','2006-12-01');
```

```
insert into employee_master values('E00003','Anita','Senior Executive','Marketing','F','1977-05-12','2007-03-21');
```

```
insert into employee_master values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');
```

```
insert into employee_master values('E00005','Radhica','Manager','HR','F','1976-07-22','2004-01-23');
```

```
insert into employee_master values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');
```

```
insert into employee_card_details values('E00001','L00001','2000-01-01');
```

```
insert into employee_card_details values('E00001','L00002','2000-01-01');
```

```
insert into employee_card_details values('E00001','L00003','2002-12-14');
```

```
insert into employee_card_details values('E00002','L00001','2007-02-01');
```

insert into employee_card_details values('E00002','L00002','2007-03-11');

insert into employee_card_details values('E00003','L00001','2007-04-15');

insert into employee_card_details values('E00003','L00002','2007-04-15');

insert into employee_card_details values('E00003','L00003','2007-04-15');

INSERT INTO item_master VALUES ('I00001','Tea Table','Y','Wooden','Furniture',5000);

INSERT INTO item_master VALUES ('I00002','Dinning Table','N','Wooden','Furniture',15000);

INSERT INTO item_master VALUES ('I00003','Tea Table','N','Steel','Furniture',6000);

INSERT INTO item_master VALUES ('I00004','Side Table','Y','Wooden','Furniture',2000);

INSERT INTO item_master VALUES ('I00005','Side Table','Y','Steel','Furniture',1500);

INSERT INTO item_master VALUES ('I00006','Tea Table','N','Steel','Furniture',7000);

INSERT INTO item_master VALUES ('I00007','Dinning Chair','Y','Wooden','Furniture',1500);

INSERT INTO item_master VALUES ('I00008','Tea Table','Y','Wooden','Furniture',4000);

INSERT INTO item_master VALUES ('I00009','Sofa','N','Wooden','Furniture',18000);

INSERT INTO item_master VALUES ('I00010','Cupboard','Y','Steel','Furniture',10000);

INSERT INTO item_master VALUES ('I00011','Cupboard','N','Steel','Furniture',14000);

INSERT INTO item_master VALUES ('I00012','Double Bed','Y','Wooden','Furniture',21000);

INSERT INTO item_master VALUES ('I00013','Double Bed','Y','Wooden','Furniture',20000);

INSERT INTO item_master VALUES ('I00014','Single Bed','Y','Steel','Furniture',10000);

INSERT INTO item_master VALUES ('I00015','Single Bed','N','Steel','Furniture',10000);

INSERT INTO item_master VALUES ('I00016','Tea Set','Y','Glass','Crockery',3000);

INSERT INTO item_master VALUES ('I00017','Tea Set','Y','Bonechina','Crockery',4000);

INSERT INTO item_master VALUES ('I00018','Dinning Set','Y','Glass','Crockery',4500);

INSERT INTO item_master VALUES ('I00019','Dinning Set','N','Bonechina','Crockery',5000);

INSERT INTO item_master VALUES ('I00020','Pencil','Y','Wooden','Stationary',5);

INSERT INTO item_master VALUES ('I00021','Pen','Y','Plastic','Stationary',100);

INSERT INTO item_master VALUES ('I00022','Pen','N','Plastic','Stationary',200);

LOAN CARD MASTER

loan_id	loan_type	duration_in_years
L00001	Furniture	5
L00002	Stationary	0
L00003	Crockery	1
NULL	NULL	NULL

EMPLOYEE CARD DETAILS

employee_id	loan_id	card_issue_date
E00001	L00001	2000-01-01
E00001	L00002	2000-01-01
E00001	L00003	2002-12-14
E00002	L00001	2007-02-01
E00002	L00002	2007-03-11
E00003	L00001	2007-04-15
E00003	L00002	2007-04-15
E00003	L00003	2007-04-15

EMPLOYEE ISSUE DETAILS

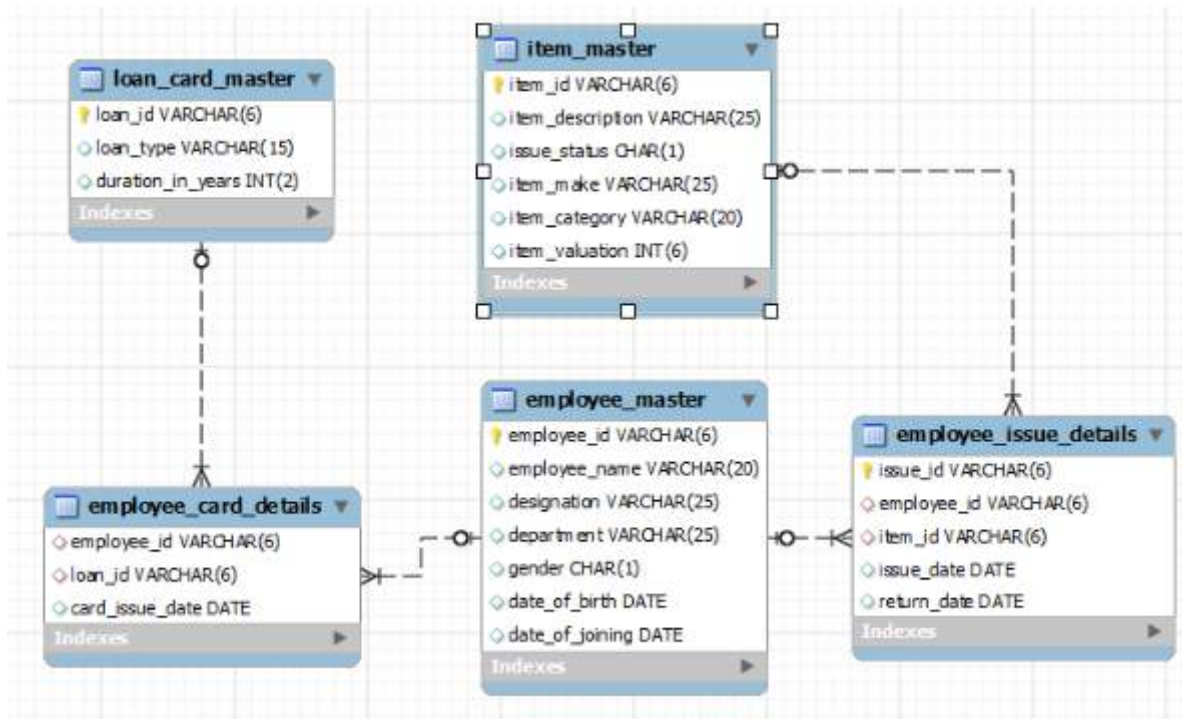
issue_id	employee_id	item_id	issue_date	return_date
ISS001	E00001	I00001	2012-02-03	2014-02-03
ISS002	E00001	I00004	2012-02-03	2020-02-03
ISS003	E00002	I00005	2013-01-03	2015-01-03
ISS004	E00003	I00007	2010-07-04	2012-07-04
ISS005	E00003	I00008	2010-07-04	2012-08-05
ISS006	E00003	I00010	2012-03-14	2012-06-15
ISS007	E00004	I00012	2013-04-14	2016-04-14
ISS008	E00006	I00018	2012-08-18	2019-04-17
ISS009	E00004	I00018	2013-04-18	2013-05-18
NULL	NULL	NULL	NULL	NULL

EMPLOYEE MASTER

employee_id	employee_name	designation	department	gender	date_of_birth	date_of_joining
E00001	Ram	Manager	Finance	M	1973-12-01	2000-01-01
E00002	Abhay	Assistant Manager	Finance	M	1976-01-01	2006-12-01
E00003	Anita	Senior Executive	Marketing	F	1977-05-12	2007-03-21
E00004	Zuben	Manager	Marketing	M	1974-10-12	2003-07-23
E00005	Radhica	Manager	HR	F	1976-07-22	2004-01-23
E00006	John	Executive	HR	M	1983-11-08	2010-05-17
NULL	NULL	NULL	NULL	NULL	NULL	NULL

ITEM MASTER

item_id	item_description	issue_status	item_make	item_category	item_valuation
I00001	Tea Table	Y	Wooden	Furniture	5000
I00002	Dinning Table	N	Wooden	Furniture	15000
I00003	Tea Table	N	Steel	Furniture	6000
I00004	Side Table	Y	Wooden	Furniture	2000
I00005	Side Table	Y	Steel	Furniture	1500
I00006	Tea Table	N	Steel	Furniture	7000
I00007	Dinning Chair	Y	Wooden	Furniture	1500
I00008	Tea Table	Y	Wooden	Furniture	4000
I00009	Sofa	N	Wooden	Furniture	18000
I00010	Cupboard	Y	Steel	Furniture	10000
I00011	Cupboard	N	Steel	Furniture	14000
I00012	Double Bed	Y	Wooden	Furniture	21000
I00013	Double Bed	Y	Wooden	Furniture	20000
I00014	Single Bed	Y	Steel	Furniture	10000
I00015	Single Bed	N	Steel	Furniture	10000
I00016	Tea Set	Y	Glass	Crockery	3000
I00017	Tea Set	Y	Bonechina	Crockery	4000
I00018	Dinning Set	Y	Glass	Crockery	4500
I00019	Dinning Set	N	Bonechina	Crockery	5000
I00020	Pencil	Y	Wooden	Stationary	5
I00021	Pen	Y	Plastic	Stationary	100
I00022	Pen	N	Plastic	Stationary	200
NULL	NULL	NULL	NULL	NULL	NULL



1. Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

```
SELECT item_category,count(item_id) Count_category FROM
item_master GROUP BY item_category ORDER BY Count_category DESC;
```

item_category	Count_category
Furniture	15
Crockery	4
Stationary	3

2. Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

```
SELECT count(employee_id) No_of_Employees FROM
employee_master WHERE department='HR';
```

No_of_Employees
2

3. Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id,employee_name,designation,department FROM employee_master
```

WHERE employee_id NOT IN (SELECT employee_id FROM employee_issue_details)

ORDER BY employee_id;

employee_id	employee_name	designation	department
E00005	Radhica	Manager	HR
NULL	NULL	NULL	NULL

4. Write a query to display the employee id, employee name who was issued an item of highest valuation. In case of multiple records, display the records sorted in ascending order based on employee id.[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

SELECT employee_id,employee_name FROM employee_master

WHERE employee_id IN(SELECT employee_id FROM employee_issue_details

WHERE item_id IN (SELECT item_id FROM item_master

WHERE item_valuation=(SELECT max(item_valuation) FROM

item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)));

employee_id	employee_name
E00004	Zuben
NULL	NULL

5. Write a query to display issue_id, employee_id, employee_name. Display the records sorted in ascending order based on issue id.

SELECT eid.issue_id, eid.employee_id, em.employee_name

FROM employee_master em JOIN employee_issue_details eid

ON em.employee_id=eid.employee_id ORDER BY eid.issue_id;

issue_id	employee_id	employee_name
ISS001	E00001	Ram
ISS002	E00001	Ram
ISS003	E00002	Abhay
ISS004	E00003	Anita
ISS005	E00003	Anita
ISS006	E00003	Anita
ISS007	E00004	Zuben
ISS008	E00006	John
ISS009	E00004	Zuben

6. Write a query to display employee id, employee name who don't have loan cards. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id, employee_name FROM employee_master
```

```
WHERE employee_id NOT IN (SELECT employee_id FROM employee_card_details);
```

employee_id	employee_name
E00004	Zuben
E00005	Radhica
E00006	John
NULL	NULL

7. Write a query to count the number of cards issued to an employee "Ram". Give the count an alias name as No_of_Cards.

```
SELECT count(loan_id) No_of_Cards FROM
```

```
employee_card_details WHERE employee_id IN
```

```
(SELECT employee_id FROM employee_master WHERE employee_name='Ram');
```

(or)

```
SELECT count(loan_id) No_of_Cards FROM
```

```
employee_card_details c JOIN employee_master e
```

```
ON c.employee_id = e.employee_id
```

```
WHERE e.employee_name = 'Ram';
```

No_of_Cards
3

8. Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

```
SELECT count(e.employee_id) Count_Stationary
```

```
FROM employee_card_details e JOIN loan_card_master l
```

```
ON e.loan_id=l.loan_id WHERE l.loan_type='Stationary';
```

Count_Stationary
3

9. Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then

```
SELECT e.employee_id, employee_name, count(e.item_id) Count FROM
```

employee_issue_details e JOIN employee_master em ON e.employee_id=em.employee_id
 GROUP BY e.employee_id ORDER BY count DESC,e.employee_id;

employee_id	employee_name	Count
E00003	Anita	3
E00001	Ram	2
E00004	Zuben	2
E00002	Abhay	1
E00006	John	1

10. Write a query to display the employee id, employee name who was issued an item of minimum valuation. In case of multiple records, display them sorted in ascending order based on employee id. [Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

```
SELECT employee_id, employee_name FROM employee_master
WHERE employee_id IN (SELECT employee_id FROM employee_issue_details
WHERE item_id IN (SELECT item_id FROM item_master
WHERE item_valuation = (SELECT min(item_valuation) FROM
item_master i JOIN employee_issue_details e ON i.item_id = e.item_id)))
ORDER BY employee_id;
```

employee_id	employee_name
E00002	Abhay
E00003	Anita
NULL	NULL

11. Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION. Display the records sorted in ascending order based on employee id. Consider only employees who have been issued atleast 1 item.

```
SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id = i.item_id
JOIN employee_master em ON em.employee_id = e.employee_id
GROUP BY e.employee_id ORDER BY employee_id;
```

employee_id	employee_name	TOTAL_VALUATION
E00001	Ram	7000
E00002	Abhay	1500
E00003	Anita	15500
E00004	Zuben	25500
E00006	John	4500

12. Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days.Display the records sorted in ascending order based on employee id.

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE datediff(ei.return_date,ei.issue_date)>365
ORDER BY employee_id;
```

employee_id	employee_name
E00001	Ram
E00002	Abhay
E00003	Anita
E00004	Zuben
E00006	John

13. Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT_ITEMS.Display the records sorted in ascending order on employee id.

```
SELECT e.employee_id,e.employee_name,count(ei.item_id) COUNT_ITEMS FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON ei.item_id=i.item_id
WHERE i.item_category='Furniture'
GROUP BY ei.employee_id HAVING count(ei.item_id)>1;
```

employee_id	employee_name	COUNT_ITEMS
E00001	Ram	2
E00003	Anita	3

14. Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

```
SELECT gender,count(employee_id) FROM employee_master
```

```
GROUP BY gender ORDER BY gender;
```

gender	count(employee_id)
F	2
M	4

15. Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

```
SELECT employee_id,employee_name FROM employee_master
```

```
WHERE year(date_of_joining)>'2005'
```

```
ORDER BY employee_id;
```

employee_id	employee_name
E00002	Abhay
E00003	Anita
E00006	John
NULL	NULL

16. Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No_of_Furnitures. Display the records sorted in ascending order based on issue_status.

```
SELECT issue_status,count(item_id) No_of_Furniture FROM
```

```
item_master WHERE item_category='Furniture'
```

```
GROUP BY issue_status ORDER BY issue_status;
```

issue_status	No_of_Furniture
N	6
Y	9

17. Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.

```
SELECT item_category,item_make,item_description,count(item_id) No_of_items FROM
```

```
item_master GROUP BY item_category,item_make,item_description
```

```
ORDER BY item_category,item_make,item_description;
```

item_category	item_make	item_description	No_of_items
Crockery	Bonechina	Dinning Set	1
Crockery	Bonechina	Tea Set	1
Crockery	Glass	Dinning Set	1
Crockery	Glass	Tea Set	1
Furniture	Steel	Cupboard	2
Furniture	Steel	Side Table	1
Furniture	Steel	Single Bed	2
Furniture	Steel	Tea Table	2
Furniture	Wooden	Dinning Chair	1
Furniture	Wooden	Dinning Table	1
Furniture	Wooden	Double Bed	2
Furniture	Wooden	Side Table	1
Furniture	Wooden	Sofa	1
Furniture	Wooden	Tea Table	2
Stationary	Plastic	Pen	2
Stationary	Wooden	Pencil	1

18. Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

```
SELECT e.employee_id,employee_name,i.item_id,i.item_description FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
WHERE month(ei.issue_date)='01' and year(ei.issue_date)='2013'
ORDER BY employee_id,item_id;
```

employee_id	employee_name	item_id	item_description
E00002	Abhay	I00005	Side Table

19. Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories. Give the alias name for category count as COUNT_CATEGORY. Display the records sorted in ascending order based on employee id.

```
SELECT ei.employee_id,e.employee_name,count(DISTINCT i.item_category) COUNT_CATEGORY FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
```

GROUP BY ei.employee_id

HAVING COUNT_CATEGORY>=2

ORDER BY employee_id;

employee_id	employee_name	COUNT_CATEGORY
E00004	Zuben	2

20. Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.

SELECT item_id, item_description FROM item_master

WHERE item_id NOT IN (SELECT item_id from employee_issue_details)

ORDER BY item_id;

item_id	item_description
I00002	Dinning Table
I00003	Tea Table
I00006	Tea Table
I00009	Sofa
I00011	Cupboard
I00013	Double Bed
I00014	Single Bed
I00015	Single Bed
I00016	Tea Set
I00017	Tea Set
I00019	Dinning Set
I00020	Pencil
I00021	Pen
I00022	Pen
NULL	NULL

21. Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM

item_master i JOIN employee_issue_details e ON e.item_id=i.item_id

JOIN employee_master em ON em.employee_id=e.employee_id

```

GROUP BY e.employee_id HAVING sum(i.item_valuation)<=ALL(
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id);

```

employee_id	employee_name	TOTAL_VALUATION
E00002	Abhay	1500

22. Write a query to display the employee id, employee name, card issue date and card valid date. Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE. [Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.]

```

SELECT e.employee_id,e.employee_name,card_issue_date,
case
when l.duration_in_years>0 then date_add(ec.card_issue_date,interval l.duration_in_years year)
when l.duration_in_years=0 then 'No Validity Date' end CARD_VALID_DATE
FROM
employee_master e JOIN employee_card_details ec ON e.employee_id=ec.employee_id
JOIN loan_card_master l ON l.loan_id=ec.loan_id
ORDER BY employee_name,CARD_VALID_DATE;

```

employee_id	employee_name	card_issue_date	CARD_VALID_DATE
E00002	Abhay	2007-02-01	2012-02-01
E00002	Abhay	2007-03-11	No Validity Date
E00003	Anita	2007-04-15	2008-04-15
E00003	Anita	2007-04-15	2012-04-15
E00003	Anita	2007-04-15	No Validity Date
E00001	Ram	2002-12-14	2003-12-14
E00001	Ram	2000-01-01	2005-01-01
E00001	Ram	2000-01-01	No Validity Date

23. Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

```

SELECT DISTINCT e.employee_id,e.employee_name FROM

```

```

employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE e.employee_id NOT IN (SELECT employee_id FROM employee_issue_details
WHERE year(issue_date)='2013')
ORDER BY employee_id;

```

employee_id	employee_name
E00001	Ram
E00003	Anita
E00006	John

24. Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.

```

SELECT issue_id, eid.employee_id, employee_name, im.item_id, item_description, issue_date
FROM employee_issue_details eid JOIN employee_master em ON eid.employee_id=em.employee_id
JOIN item_master im ON eid.item_id=im.item_id
ORDER BY issue_date DESC, issue_id;

```

issue_id	employee_id	employee_name	item_id	item_description	issue_date
ISS009	E00004	Zuben	I00018	Dinning Set	2013-04-18
ISS007	E00004	Zuben	I00012	Double Bed	2013-04-14
ISS003	E00002	Abhay	I00005	Side Table	2013-01-03
ISS008	E00006	John	I00018	Dinning Set	2012-08-18
ISS006	E00003	Anita	I00010	Cupboard	2012-03-14
ISS001	E00001	Ram	I00001	Tea Table	2012-02-03
ISS002	E00001	Ram	I00004	Side Table	2012-02-03
ISS004	E00003	Anita	I00007	Dinning Chair	2010-07-04
ISS005	E00003	Anita	I00008	Tea Table	2010-07-04

25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

```

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id HAVING sum(i.item_valuation)>=ALL(

```

```
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM  
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id  
JOIN employee_master em ON em.employee_id=e.employee_id  
GROUP BY e.employee_id);
```

employee_id	employee_name	TOTAL_VALUATION
E00004	Zuben	25500

////////////////////.Event Hall-Average cost of booked halls

```
select CITY,round(avg(cost_perday),0) as AVERAGE_COST from t_hall_details
where hall_id in(Select hall_id from t_hall_booking) and capacity>150
group by CITY
order by average_cost;
```

////////////////////Event Hall-Average cost of not booked halls

```
SELECT
    City, ROUND(AVG(cost_perday)) 'Average_Cost'
FROM
    T_hall_details
WHERE
    Hall_id NOT IN (SELECT
        Hall_id
    FROM
        T_hall_booking)
    AND capacity > 100
GROUP BY city;
```

////////////////////Dream Home-Customer name details based on total cost

```
SELECT c.customer_name,SUM(f.cost_inlakh) FROM t_flat_booking b
JOIN t_flat_details f ON b.flat_no=f.flat_no
JOIN t_customer_details C ON c.customer_id=b.customer_id
WHERE LENGTH(c.customer_name)>'10'
GROUP BY c.customer_name
ORDER BY customer_name
```

////////////////////Hospital-Total fees received based on gender and shift

```
SELECT t_patient.gender, sum(t_doctor.fees) FEES_RECEIVED
```

```

FROM t_patient
JOIN t_doctor on t_doctor.doctor_id=t_patient.doctor_id
JOIN t_hospital on t_hospital.available_doctor=t_doctor.doctor_id
WHERE upper(t_hospital.shift_time)="MORNING"
GROUP BY t_patient.gender
ORDER BY t_patient.gender DESC;

```

//////////Insurance-List of Agents

```

select a.agent_id,p.policy_name,sum(p.policy_sum)as policy_sum from t_agent a
join t_member m on m.agent_id=a.agent_id
join t_policy p on p.policy_id=m.policy_id
group by a.agent_id,p.policy_name
having count(m.member_id) >=1
order by a.agent_id,p.policy_name,policy_sum;

```

//////////Minimum & Maximum Discount Amount

```

SELECT MIN(DISCOUNT_AMOUNT) AS MIN_DISCOUNT,
MAX(DISCOUNT_AMOUNT) AS MAX_DISCOUNT
FROM DISCOUNT_MASTER;

```

//////////Number of Appointments..

```

SELECT doctor_id, COUNT(app_number) as APPOINTMENT_COUNT
FROM appointment
GROUP BY doctor_id
ORDER BY doctor_id;

```

Student Details In Capital Case..

```

select student_id,upper(student_name) as NAME,department,phone_no
from student_details
where address='BANGALORE'
order by student_id;

```

//////////Pizza-Delivery Partner Details With Rating..

```

select partner_id,partner_name,concat(partner_id,substr(partner_name,1,4))

```


Name,if(rating>=9,'Excellent',if(rating>=7 and rating<9,'Good','Worst')) as FEEDBACK from delivery_partner order by partner_id;

//////////Pizza-Framing Customer password ..

Select concat (cust_name,cust_id)

As USERNAME

Concat (substring(cust_name,1,3),

Substring(cust_phone,-4,4) as PASSWORD

FROM customer

ORDER BY USERNAME;

//////////Pizza-Low cost and High cost pizza..

Select pizza_type, pizza_name from pizza

Where amount IN(select max(amount) from pizza) or

Amount IN(SELECT MIN(amount) from pizza)

LIMIT 2;

//////////Cricket-Average runs of players based on name..

select m.player_id, round(avg(m.player_runs)) as average_runs from t_match_score_card m

join t_player p

on p.player_id=m.player_id where player_name like 'S%'

group by m.player_id

order by average_runs desc;

//////////Car Pooling-Vehicle details .

select c.vehicle_model,c.vehicle_type,sum(ci.distance)from car c

join booking b on b.vehicle_no=c.vehicle_no

join city_locations ci on (ci.city1=b.pickup_from and ci.city2=b.drop_at)or(ci.city2=b.pickup_from and ci.city1=b.drop_at)

group by c.vehicle_type,c.vehicle_model

order by sum(ci.distance);

//////////Event Hall-Customer details with booking done..

SELECT customer_id

,customer_name

```

,mobile_no
FROM t_customer_details
WHERE length(customer_name) > 10
AND customer_id IN (
  SELECT customer_id
  FROM (
    SELECT customer_id
    ,count(hall_id)
    FROM t_hall_booking
    GROUP BY customer_id
    HAVING count(hall_id) > (
      SELECT count(h.hall_id)
      FROM t_hall_booking h
      INNER JOIN t_customer_details c ON c.customer_id = h.customer_id
      WHERE c.customer_name = 'Suman Singh'
      GROUP BY h.customer_id
    )
  ) AS T1
)
ORDER BY customer_name;

```

//////////Dream Home -Flat details based on year..

```

select a.flat_no FLAT_NO, b.size SIZE, b.area area
from t_flat_booking a
join t_flat_details b
on a.flat_no = b.flat_no
where year(a.registration_date)
in (select year(b.registration_date)
from t_customer_details a
join t_flat_booking b
on a.customer_id=b.customer_id

```

where upper(a.customer_name='Niraj Kumar'))

order by area asc,a.flat_no desc;

//////////Cricket-Player details..

SELECT DISTINCT p.PLAYER_ID,p.PLAYER_NAME,p.PLAYER_CITY from t_player p

join t_match_score_card s on p.player_id=s.player_id

join t_match_record r on r.match_id=s.match_id

WHERE s.waysof_dismissal='STUMPED' AND r.played_city='BANGALORE'

order by player_name desc;

//////////Room Details Based On Location..

select

ROOM_DETAILS.ROOM_ID,ROOM_DETAILS.ROOM_TYPE,ROOM_DETAILS.MEMBER_CAPACITY,ROOM
_DETAILS.ROOM_RENT

from ROOM_DETAILS

inner join HOSTEL_DETAILS

on ROOM_DETAILS.HOSTEL_ID=HOSTEL_DETAILS.HOSTEL_ID

where HOSTEL_DETAILS.LOCATION = 'PHASE-A'

order by ROOM_DETAILS.ROOM_ID;

//////////Patient Appointment details Based On Month..

SELECT DISTINCT(PATIENT_ID),P_FIRST_NAME,P_AGE,ADDRESS,CONTACT_NUMBER

FROM PATIENT

WHERE PATIENT_ID IN(SELECT PATIENT_ID FROM APPOINTMENT WHERE APP_DATE BETWEEN '2019-
06-01' AND '2019-06-31')

ORDER BY PATIENT_ID;

//////////cricket number of players in each city

select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from
t_player where player_city not in

(select distinct played_city from t_match_record)

group by player_city

order by NUMBER_OF_PLAYERS,PLAYER_CITY;

//////////Hospital-Maximum fees paid patient details..

```
select p.patient_name , d.doctor_name , d.fees as 'fees_paid' , h.shift_time as 'checkup_done'
from t_patient p
join t_doctor d on p.doctor_id = d.doctor_id
join t_hospital h on h.available_doctor = d.doctor_id
where d.fees>( select max(fees) from t_doctor where specialization = 'DERMA')
order by d.doctor_name , p.patient_name;
```

//////////Insurance-Agent details..

```
select count(b.member_id) as NUMBER_OF_MEMBERS,a.agent_name as AGENT_NAME
from t_agent a join t_member b
on a.agent_id=b.agent_id
where a.agent_name like 'S%' or a.agent_name like 's%'
group by a.agent_name
order by AGENT_NAME,NUMBER_OF_MEMBERS asc;
```

Concatenating Details ..

```
SELECT CONCAT(MOVIE_NAME," is a ",LANGUAGE," Movie") AS MOVIE_DETAILS FROM
MOVIE_MASTER
ORDER BY MOVIE_DETAILS DESC;
```

//////////Pizza-Highest Business Customer Details..

```
select pizza.cust_id,customer.cust_name,sum(pizza.amount) as Max_Amount from customer
join pizza on customer.cust_id=pizza.cust_id group by pizza.cust_id order by Max_Amount
desc limit 1;
```

//////////Pizza-Total Cost of Pizza Ordered..

```
select cust_id, pizza_name, count(cust_id) as 'Times taken', sum(amount) as 'Total cost'
from pizza
```

```
where amount > 1200 group by pizza_name , cust_id order by cust_id asc;
```

////////// Pizza-Extra Large Pizza ..

//////////Event Hall-Customer having Average payment..

```

select cd.customer_name,round(avg(hd.cost_perday)) as A from t_customer_details cd join
t_hall_booking hb
on cd.customer_id=hb.customer_id
join t_hall_details hd on hd.hall_id=hb.hall_id
group by cd.customer_name
having A>(select max(hdd.cost_perday)from t_hall_details hdd join
t_hall_booking hbb on hbb.hall_id=hdd.hall_id
join t_customer_details cdd on cdd.customer_id=hbb.customer_id
where cdd.customer_name='Suraj Kumar')
order by A desc;

```

//////////Car Pooling-Maximum time driven driver details ..

```

select b.driver_id, d.driver_name, count(driver_id) as MaxTimesDriven
from driver d
inner join booking b on d.id=b.driver_id
group by b.driver_id
having count(driver_id)>2
order by b.driver_id;

```

//////////Hostel-Insert Student Records

```

insert into Student_details values
('S1001','Varsha','ECE','1999-06-12','CHENNAI',9845712345,'varsha123@gmail.com'),
('S1002','William','ECE','1999-02-04','CALCUTTA',6845712345,'william123@gmail.com'),
('S1003','Basha','EEE','1999-06-14','DELHI',9945712345,'basha222@gmail.com'),
('S1004','Catherine','CSE','1998-08-16','DELHI',6785712345,'cathu123@gmail.com'),
('S1005','Kate','ECE','1999-06-30','BANGALORE',7685712345,'katedd@gmail.com'),
('S1006','Michel','ECE','1998-06-04','COIMBATORE',6645712345,'michel000@gmail.com');

```

//////////Movie - Modify the datatype..

```

ALTER TABLE CUSTOMER_MASTER MODIFY COLUMN PHONE_NO INT(10);

```

//////////Create Movie_Master table set1..

////////////////////Hospital- Change the datatype/column ..

alter table patient modify

contact_number int(10);

alter table patient change p_age patient_age int;

////////////////////Hospital- Add a new column set1..

alter table doctor add column dr_contact_number int(10);

////////////////////Pizza Store- Update PIZZA table discount1.2..

UPDATE pizza

set amount = (amount * 0.95)/100

Where pizza_type = " Extra Large" ;

////////////////////Pizza Store- Alter table-Foreign key 1.1..

**ALTER TABLE pizza ADD CONSTRAINT FK1 FOREIGN KEY (cust_id) REFERENCES
customer(cust_id);**

**ALTER TABLE pizza ADD CONSTRAINT FK2 FOREIGN KEY (partner_id) REFERENCES
delivery_partner(partner_id);**

////////////////////Pizza Store - Update PIZZA table 1.2..

////////////////////Pizza Store- Alter table Pizza1.1..

////////////////////Event Hall- Update the event date1.2..

////////////////////Event Hall -Update T_HALL_DETAILS table1.2..

////////////////////Event Hall-Alter T_HALL_BOOKING table1////////////////////1 ..

alter table t_hall_booking

modify hall_id varchar(10) not null;

alter table t_hall_booking

add foreign key(hall_id) references t_hall_details(hall_id);

alter table t_hall_booking

modify customer_id varchar(10) not null;

alter table t_hall_booking

add foreign key(customer_id) references t_customer_details(customer_id);

//////////Dream Home- Update t_flat_details table1.2..

//////////Dream Home- Alter table t_flat_booking1.1..

//////////Cricket-Update T_PLAYER table(1.2)..

update t_player

set total_wickets=case

when(player_city='BANGALORE' and player_name like 'A%')

THEN total_wickets+5

when(player_city='DELHI' and player_name like 'A%')

THEN total_wickets+7

ELSE total_wickets

END;

//////////Cricket -Alter T_MATCH_SCORE_CARD table(1.1)..

alter table t_match_score_card add foreign key (match_id) references t_match_record (match_id);

alter table t_match_score_card add foreign key (player_id) references t_player(player_id);

//////////Car Pooling - Update booking table1.2..

update booking

set fare=(select min(distance)*11 from city_locations ct

join booking b On b.pickup_from=ct.city1 AND b.drop_at=ct.city2);

//////////Car Pooling- Create BOOKING table 1.1

create table booking (

booking_no varchar(50),

pickup_from varchar(50),

drop_at varchar(50),

customer_id varchar(50),

vehicle_no varchar(50),

```
driver_id varchar(50),
fare decimal(7,2),
primary key (booking_no),
foreign key (customer_id) references customer(id),
foreign key (vehicle_no) references car(vehicle_no),
foreign key (driver_id) references driver(id)
);
```

//////////Hospital-Update T_DOCTOR table 1.2..

```
update t_doctor set fees=350
where specialization="ENT" and doctor_name like "J%";
update t_doctor set fees=600
where specialization="DERMA" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="SURGEON" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="ORTHO" and doctor_name like "J%";
```

//////////Hospital- Alter T_HOSPITAL table 1.1..

```
alter table t_hospital
add foreign key (available_doctor) references t_doctor(doctor_id);
```

//////////Insurance-Update Agent details(1.2)..

```
Update t_agent
set target_policy_sum=
case
when upper(agent_city)='PUNE' and upper(agent_id) like 'M%'
then 400000
when upper(agent_city)='CHENNAI' and upper(agent_id) like 'M%'
then 250000
else target_policy_sum
```


end;

//////////Insurance- Alter table-add constraint(1.1)..

alter table T_MEMBER

ADD foreign key(AGENT_ID) references T_AGENT(agent_id),

ADD foreign key(POLICY_ID) references T_POLICY(policy_id);

//////////Event Hall- Alter table Hall Booking 1.1..

alter table t_hall_booking

modify hall_id varchar(10) not null;

alter table t_hall_booking

add foreign key(hall_id) references t_hall_details(hall_id);

//////////Pizza Store-Update PIZZA table 1.2..

//////////Pizza Store- Alter table Pizza 1.1..

//////////Patient Appointment Details based on reason..

//////////2.Pizza-Highest Business Date..

//////////2. Event Hall-Number of booking customer wise ..

select distinct c.customer_id,c.customer_name,count(h.hall_id) as NO_OF_BOOKING

from t_customer_details c

right join t_hall_booking h

on c.customer_id = h.customer_id

where h.event_date like '2020%'

group by c.customer_id

having c.customer_name like 'S%'

order by 2;

//////////2. Dream Home-Maximum cost of flat

SELECT floor_no AS FLOOR_NO,MAX(cost_inlakh) AS MAX_PRICE FROM t_flat_details

GROUP BY floor_no

ORDER BY floor_no DESC

//////////2.Cricket-Number of players in each city..

```
select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from t_player where
player_city not in
(select distinct played_city from t_match_record)
group by player_city
order by NUMBER_OF_PLAYERS,PLAYER_CITY;
```

//////////2.Car pooling-Driver booking details based on name..

```
select a.booking_no,b.user_name,c.driver_name,a.pickup_from,a.drop_at,d.distance
from booking a join customer b on a.customer_id=b.id
      join driver c on a.driver_id=c.id
      join city_locations d on ((a.pickup_from=d.city1 and
a.drop_at=d.city2)or(a.pickup_from=d.city2 and a.drop_at=d.city1))
      where upper(c.driver_name)='JOE AMAL'
      order by d.distance
```

//////////2.Hospital-Number of doctors based on shift..

```
select h.shift_time as SHIFT_TIME,count(h.available_doctor) AS NUMBER_OF_DOCTORS from
t_hospital h
join t_doctor d on d.doctor_id=h.available_doctor
where specialization = 'SURGEON'
group by shift_time
having count(available_doctor)>=1
order by shift_time desc;
```

//////////Movie details based on Certification and Duration..

```
select movie_id,movie_name,director_name,language from movie_master where certification='U'
and duration>130
order by movie_id;
```

//////////Student-Room Details..

```
select s.student_id,student_name,department,DOJ,r.room_id,  
room_type from student_details s join admission_details a  
on s.student_id=a.student_id join room_details r  
on r.room_id=a.room_id order by 1;
```

//////////2.Pizza-Delivery partner details..

//////////2.Pizza-Highest Selling Pizza..

```
SELECT order_date, sum(amount) as Highest_Business  
FROM pizza  
GROUP BY order_date  
ORDER BY Highest_Business DESC  
LIMIT 1;
```

//////////2. Event Hall-Halls booked more than once ..

```
select a.hall_name, count(b.hall_id) as no_of_times_booked  
from t_hall_details a join t_hall_booking b on a.hall_id = b.hall_id  
group by a.hall_name  
having length(a.hall_name)>5 and count(b.hall_id)>1  
order by a.hall_name desc  
;
```

//////////2.Insurance-List of Policies..

```
select distinct p.policy_name, p.policy_type  
from t_policy p, t_member m  
where p.policy_id = m.policy_id  
and m.member_id >= '1'  
order by policy_name, policy_type asc;
```

////////////////////Event Hall-Average cost of booked halls

```
select CITY,round(avg(cost_perday),0) as AVERAGE_COST from t_hall_details
where hall_id in(Select hall_id from t_hall_booking) and capacity>150
group by CITY
order by average_cost;
```

////////////////////Event Hall-Average cost of not booked halls

```
SELECT
    City, ROUND(AVG(cost_perday)) 'Average_Cost'
FROM
    T_hall_details
WHERE
    Hall_id NOT IN (SELECT
        Hall_id
    FROM
        T_hall_booking)
    AND capacity > 100
GROUP BY city;
```

////////////////////Dream Home-Customer name details based on total cost

```
SELECT c.customer_name,SUM(f.cost_inlakh) FROM t_flat_booking b
JOIN t_flat_details f ON b.flat_no=f.flat_no
JOIN t_customer_details C ON c.customer_id=b.customer_id
WHERE LENGTH(c.customer_name)>'10'
GROUP BY c.customer_name
ORDER BY customer_name
```

//////////Hospital-Total fees received based on gender and shift

```
SELECT t_patient.gender, sum(t_doctor.fees) FEES_RECEIVED
FROM t_patient
JOIN t_doctor on t_doctor.doctor_id=t_patient.doctor_id
JOIN t_hospital on t_hospital.available_doctor=t_doctor.doctor_id
WHERE upper(t_hospital.shift_time)="MORNING"
GROUP BY t_patient.gender
ORDER BY t_patient.gender DESC;
```

//////////Insurance-List of Agents

```
select a.agent_id,p.policy_name,sum(p.policy_sum)as policy_sum from t_agent a
join t_member m on m.agent_id=a.agent_id
join t_policy p on p.policy_id=m.policy_id
group by a.agent_id,p.policy_name
having count(m.member_id) >=1
order by a.agent_id,p.policy_name,policy_sum;
```

//////////Minimum & Maximum Discount Amount

```
SELECT MIN(DISCOUNT_AMOUNT) AS MIN_DISCOUNT,
MAX(DISCOUNT_AMOUNT) AS MAX_DISCOUNT
FROM DISCOUNT_MASTER;
```

//////////Number of Appointments..

```
SELECT doctor_id, COUNT(app_number) as APPOINTMENT_COUNT
FROM appointment
GROUP BY doctor_id
ORDER BY doctor_id;
```

Student Details In Capital Case..

```
select student_id,upper(student_name) as NAME,department,phone_no
from student_details
where address='BANGALORE'
order by student_id;
```

//////////Pizza-Delivery Partner Details With Rating..

**select partner_id,partner_name,concat(partner_id,substr(partner_name,1,4))
Name,if(rating>=9,'Excellent',if(rating>=7 and rating<9,'Good','Worst')) as FEEDBACK from
delivery_partner order by partner_id;**

//////////Pizza-Framing Customer password ..

Select concat (cust_name,cust_id)

As USERNAME

Concat (substring(cust_name,1,3),

Substring(cust_phone,-4,4) as PASSWORD

FROM customer

ORDER BY USERNAME;

//////////Pizza-Low cost and High cost pizza..

Select pizza_type, pizza_name from pizza

Where amount IN(select max(amount) from pizza) or

Amount IN(SELECT MIN(amount) from pizza)

LIMIT 2;

//////////Cricket-Average runs of players based on name..

**select m.player_id, round(avg(m.player_runs)) as average_runs from t_match_score_card
m**

join t_player p

on p.player_id=m.player_id where player_name like 'S%'

group by m.player_id

order by average_runs desc;

//////////Car Pooling-Vehicle details .

select c.vehicle_model,c.vehicle_type,sum(ci.distance)from car c

join booking b on b.vehicle_no=c.vehicle_no

**join city_locations ci on (ci.city1=b.pickup_from and ci.city2=b.drop_at)or(ci.city2=b.pickup_from and
ci.city1=b.drop_at)**

group by c.vehicle_type,c.vehicle_model

order by sum(ci.distance);

//////////Event Hall-Customer details with booking done..

```
SELECT customer_id
,customer_name
,mobile_no
FROM t_customer_details
WHERE length(customer_name) > 10
AND customer_id IN (
  SELECT customer_id
  FROM (
    SELECT customer_id
    ,count(hall_id)
    FROM t_hall_booking
    GROUP BY customer_id
    HAVING count(hall_id) > (
      SELECT count(h.hall_id)
      FROM t_hall_booking h
      INNER JOIN t_customer_details c ON c.customer_id = h.customer_id
      WHERE c.customer_name = 'Suman Singh'
      GROUP BY h.customer_id
    )
  ) AS T1
)
ORDER BY customer_name;
```

//////////Dream Home -Flat details based on year..

```
select a.flat_no FLAT_NO, b.size SIZE, b.area area
from t_flat_booking a
join t_flat_details b
on a.flat_no = b.flat_no
where year(a.registration_date)
in (select year(b.registration_date)
```

```

from t_customer_details a
join t_flat_booking b
on a.customer_id=b.customer_id
where upper(a.customer_name='Niraj Kumar'))
order by area asc,a.flat_no desc;

```

/////////////////Cricket-Player details..

```

SELECT DISTINCT p.PLAYER_ID,p.PLAYER_NAME,p.PLAYER_CITY from t_player p
join t_match_score_card s on p.player_id=s.player_id
join t_match_record r on r.match_id=s.match_id
WHERE s.waysof_dismissal='STUMPED' AND r.played_city='BANGALORE'
order by player_name desc;

```

/////////////////Room Details Based On Location..

```

select
ROOM_DETAILS.ROOM_ID,ROOM_DETAILS.ROOM_TYPE,ROOM_DETAILS.MEMBER_CAPACITY,ROOM
_DETAILS.ROOM_RENT
from ROOM_DETAILS
    inner join HOSTEL_DETAILS
        on ROOM_DETAILS.HOSTEL_ID=HOSTEL_DETAILS.HOSTEL_ID
where HOSTEL_DETAILS.LOCATION = 'PHASE-A'
    order by ROOM_DETAILS.ROOM_ID;

```

/////////////////Patient Appointment details Based On Month..

```

SELECT DISTINCT(PATIENT_ID),P_FIRST_NAME,P_AGE,ADDRESS,CONTACT_NUMBER
FROM PATIENT
WHERE PATIENT_ID IN(SELECT PATIENT_ID FROM APPOINTMENT WHERE APP_DATE BETWEEN '2019-
06-01' AND '2019-06-31')
ORDER BY PATIENT_ID;

```

/////////////////cricket number of players in each city

```

select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from
t_player where player_city not in

```



```
(select distinct played_city from t_match_record)
group by player_city
order by NUMBER_OF_PLAYERS,PLAYER_CITY;
```

//////////Hospital-Maximum fees paid patient details..

```
select p.patient_name , d.doctor_name , d.fees as 'fees_paid' , h.shift_time as 'checkup_done'
from t_patient p
join t_doctor d on p.doctor_id = d.doctor_id
join t_hospital h on h.available_doctor = d.doctor_id
where d.fees>( select max(fees) from t_doctor where specialization = 'DERMA')
order by d.doctor_name , p.patient_name;
```

//////////Insurance-Agent details..

```
select count(b.member_id) as NUMBER_OF_MEMBERS,a.agent_name as AGENT_NAME
from t_agent a join t_member b
on a.agent_id=b.agent_id
where a.agent_name like 'S%' or a.agent_name like 's%'
group by a.agent_name
order by AGENT_NAME,NUMBER_OF_MEMBERS asc;
```

Concatenating Details ..

```
SELECT CONCAT(MOVIE_NAME," is a ",LANGUAGE," Movie") AS MOVIE_DETAILS FROM
MOVIE_MASTER
ORDER BY MOVIE_DETAILS DESC;
```

//////////Pizza-Highest Business Customer Details..

```
select pizza.cust_id,customer.cust_name,sum(pizza.amount) as Max_Amount from customer
join pizza on customer.cust_id=pizza.cust_id group by pizza.cust_id order by Max_Amount
desc limit 1;
```

//////////Pizza-Total Cost of Pizza Ordered..

```
select cust_id, pizza_name, count(cust_id) as 'Times taken', sum(amount) as 'Total cost'
from pizza
```

where amount > 1200 group by pizza_name , cust_id order by cust_id asc;

////////// Pizza-Extra Large Pizza ..

//////////Event Hall-Customer having Average payment..

**select cd.customer_name,round(avg(hd.cost_perday)) as A from t_customer_details cd join
t_hall_booking hb**

on cd.customer_id=hb.customer_id

join t_hall_details hd on hd.hall_id=hb.hall_id

group by cd.customer_name

having A>(select max(hdd.cost_perday)from t_hall_details hdd join

t_hall_booking hbb on hbb.hall_id=hdd.hall_id

join t_customer_details cdd on cdd.customer_id=hbb.customer_id

where cdd.customer_name='Suraj Kumar')

order by A desc;

//////////Car Pooling-Maximum time driven driver details ..

select b.driver_id, d.driver_name, count(driver_id) as MaxTimesDriven

from driver d

inner join booking b on d.id=b.driver_id

group by b.driver_id

having count(driver_id)>2

order by b.driver_id;

//////////Hostel-Insert Student Records

insert into Student_details values

('S1001','Varsha','ECE','1999-06-12','CHENNAI',9845712345,'varsha123@gmail.com'),

('S1002','William','ECE','1999-02-04','CALCUTTA',6845712345,'william123@gmail.com'),

('S1003','Basha','EEE','1999-06-14','DELHI',9945712345,'basha222@gmail.com'),

('S1004','Catherine','CSE','1998-08-16','DELHI',6785712345,'cathu123@gmail.com'),

('S1005','Kate','ECE','1999-06-30','BANGALORE',7685712345,'katedd@gmail.com'),

('S1006','Michel','ECE','1998-06-04','COIMBATORE',6645712345,'michel000@gmail.com');

//////////Movie - Modify the datatype..

ALTER TABLE CUSTOMER_MASTER MODIFY COLUMN PHONE_NO INT(10);

//////////Create Movie_Master table set1..

//////////Hospital- Change the datatype/column ..

alter table patient modify

contact_number int(10);

alter table patient change p_age patient_age int;

//////////Hospital- Add a new column set1..

alter table doctor add column dr_contact_number int(10);

//////////Pizza Store- Update PIZZA table discount1.2..

UPDATE pizza

set amount = (amount * 0.95)/100

Where pizza_type = " Extra Large" ;

//////////Pizza Store- Alter table-Foreign key 1.1..

**ALTER TABLE pizza ADD CONSTRAINT FK1 FOREIGN KEY (cust_id) REFERENCES
customer(cust_id);**

**ALTER TABLE pizza ADD CONSTRAINT FK2 FOREIGN KEY (partner_id) REFERENCES
delivery_partner(partner_id);**

//////////Pizza Store - Update PIZZA table 1.2..

//////////Pizza Store- Alter table Pizza1.1..

//////////Event Hall- Update the event date1.2..

//////////Event Hall -Update T_HALL_DETAILS table1.2..

//////////Event Hall-Alter T_HALL_BOOKING table1//////////1 ..

```
alter table t_hall_booking
modify hall_id varchar(10) not null;

alter table t_hall_booking
add foreign key(hall_id) references t_hall_details(hall_id);

alter table t_hall_booking
modify customer_id varchar(10) not null;

alter table t_hall_booking
add foreign key(customer_id) references t_customer_details(customer_id);
```

//////////Dream Home- Update t_flat_details table1.2..

//////////Dream Home- Alter table t_flat_booking1.1..

//////////Cricket-Update T_PLAYER table(1.2)..

```
update t_player
set total_wickets=case
when(player_city='BANGALORE' and player_name like 'A%')
THEN total_wickets+5
when(player_city='DELHI' and player_name like 'A%')
THEN total_wickets+7
ELSE total_wickets
END;
```

//////////Cricket -Alter T_MATCH_SCORE_CARD table(1.1)..

```
alter table t_match_score_card add foreign key (match_id) references t_match_record
(match_id);
```

```
alter table t_match_score_card add foreign key (player_id) references t_player(player_id);
```

//////////Car Pooling - Update booking table1.2..

```
update booking
set fare=(select min(distance)*11 from city_locations ct
join booking b On b.pickup_from=ct.city1 AND b.drop_at=ct.city2);
```

//////////Car Pooling- Create BOOKING table 1.1

```
create table booking (
```

```

booking_no varchar(50),
pickup_from varchar(50),
drop_at varchar(50),
customer_id varchar(50),
vehicle_no varchar(50),
driver_id varchar(50),
fare decimal(7,2),
primary key (booking_no),
foreign key (customer_id) references customer(id),
foreign key (vehicle_no) references car(vehicle_no),
foreign key (driver_id) references driver(id)
);
//////////Hospital-Update T_DOCTOR table 1.2..
update t_doctor set fees=350
where specialization="ENT" and doctor_name like "J%";
update t_doctor set fees=600
where specialization="DERMA" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="SURGEON" and doctor_name like "J%";
update t_doctor set fees=null
where specialization="ORTHO" and doctor_name like "J%";

//////////Hospital- Alter T_HOSPITAL table 1.1..
alter table t_hospital
add foreign key (available_doctor) references t_doctor(doctor_id);

//////////Insurance-Update Agent details(1.2)..
Update t_agent
set target_policy_sum=
case

```

```
when upper(agent_city)='PUNE' and upper(agent_id) like 'M%'
then 400000

when upper(agent_city)='CHENNAI' and upper(agent_id) like 'M%'
then 250000

else target_policy_sum

end;
```

//////////Insurance- Alter table-add constraint(1.1)..

```
alter table T_MEMBER

ADD foreign key(AGENT_ID) references T_AGENT(agent_id),

ADD foreign key(POLICY_ID) references T_POLICY(policy_id);
```

//////////Event Hall- Alter table Hall Booking 1.1..

```
alter table t_hall_booking

modify hall_id varchar(10) not null;

alter table t_hall_booking

add foreign key(hall_id) references t_hall_details(hall_id);
```

//////////Pizza Store-Update PIZZA table 1.2..

//////////Pizza Store- Alter table Pizza 1.1..

//////////Patient Appointment Details based on reason..

//////////2.Pizza-Highest Business Date..

//////////2. Event Hall-Number of booking customer wise ..

```
select distinct c.customer_id,c.customer_name,count(h.hall_id) as NO_OF_BOOKING
from t_customer_details c
right join t_hall_booking h
on c.customer_id = h.customer_id
where h.event_date like '2020%'
group by c.customer_id
```

having c.customer_name like 'S%'

order by 2;

//////////2. Dream Home-Maximum cost of flat

**SELECT floor_no AS FLOOR_NO,MAX(cost_inlakh) AS MAX_PRICE FROM t_flat_details
GROUP BY floor_no
ORDER BY floor_no DESC**

//////////2.Cricket-Number of players in each city..

**select player_city as PLAYER_CITY, count(player_id) as NUMBER_OF_PLAYERS from t_player where
player_city not in**

(select distinct played_city from t_match_record)

group by player_city

order by NUMBER_OF_PLAYERS,PLAYER_CITY;

//////////2.Car pooling-Driver booking details based on name..

**select a.booking_no,b.user_name,c.driver_name,a.pickup_from,a.drop_at,d.distance
from booking a join customer b on a.customer_id=b.id
join driver c on a.driver_id=c.id
join city_locations d on ((a.pickup_from=d.city1 and
a.drop_at=d.city2)or(a.pickup_from=d.city2 and a.drop_at=d.city1))
where upper(c.driver_name)='JOE AMAL'
order by d.distance**

//////////2.Hospital-Number of doctors based on shift..

**select h.shift_time as SHIFT_TIME,count(h.available_doctor) AS NUMBER_OF_DOCTORS from
t_hospital h**

join t_doctor d on d.doctor_id=h.available_doctor

where specialization = 'SURGEON'

group by shift_time

having count(available_doctor)>=1

order by shift_time desc;

//////////Movie details based on Certification and Duration..

```
select movie_id,movie_name,director_name,language from movie_master where certification='U'
and duration>130
order by movie_id;
```

//////////Student-Room Details..

```
select s.student_id,student_name,department,DOJ,r.room_id,
room_type from student_details s join admission_details a
on s.student_id=a.student_id join room_details r
on r.room_id=a.room_id order by 1;
```

//////////2.Pizza-Delivery partner details..

//////////2.Pizza-Highest Selling Pizza..

```
SELECT order_date, sum(amount) as Highest_Business
FROM pizza
GROUP BY order_date
ORDER BY Highest_Business DESC
LIMIT 1;
```

//////////2. Event Hall-Halls booked more than once ..

```
select a.hall_name, count(b.hall_id) as no_of_times_booked
from t_hall_details a join t_hall_booking b on a.hall_id = b.hall_id
group by a.hall_name
having length(a.hall_name)>5 and count(b.hall_id)>1
order by a.hall_name desc
;
```

//////////2.Insurance-List of Policies..

```
select distinct p.policy_name, p.policy_type
from t_policy p, t_member m
```



```
where p.policy_id = m.policy_id  
and m.member_id >= '1'  
order by policy_name, policy_type asc;
```