1.Airline FilghtManagement Latest

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(fd.flight\_departure\_date),

avg(fd.price) as Average\_Price

FROM air\_flight\_detailsfd, air\_flight f

WHERE f.flight\_id=fd.flight\_id and f.airline\_name="ABC "

GROUP BY f.flight\_id, fd.flight\_departure\_date,f.from\_location, f.to\_location

order by f.flight\_id, fp.Month\_Name;

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.

SELECT p.profile\_id, p.first\_name, p.address, count(t.ticket\_id) as No\_of\_Tickets

FROM air\_passenger\_profile p, air\_flight f, air\_ticket\_info t

WHERE p.profile\_id=t.profile\_id

AND t.flight\_id=f.flight\_id

AND f.airline\_name='abc'

GROUP BY p.profile\_id

HAVING count(t.ticket\_id)

IN (SELECT min(c) FROM

(SELECT count(t.ticket\_id) c FROM air\_flight f, air\_ticket\_info t

WHERE t.flight\_id=f.flight\_id ANDf.airline\_name='abc'

GROUP BY t.profile\_id )a);

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

select f.from\_location,f.to\_location,monthname(t.flight\_departure\_date),

count(t.flight\_departure\_date) as No\_of\_Services from air\_flight f

join air\_flight\_details t on t.flight\_id=f.flight\_id

group by f.from\_location,f.to\_location

order by f.from\_location,f.to\_location,monthname(t.flight\_departure\_date) asc;

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.

SELECT p.profile\_id, p.first\_name, p.address, count(t.ticket\_id) as No\_of\_Tickets

FROM air\_passenger\_profile p, air\_flight f, air\_ticket\_info t

WHERE p.profile\_id=t.profile\_id

AND t.flight\_id=f.flight\_id

AND f.airline\_name='abc'

GROUP BY p.profile\_id

HAVING count(t.ticket\_id)

IN (SELECT max(c) FROM

(SELECT count(t.ticket\_id) c FROM air\_flight f, air\_ticket\_info t

WHERE t.flight\_id=f.flight\_id ANDf.airline\_name='abc' GROUP BY t.profile\_id )a);

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.

select p.profile\_id,p.first\_name,p.last\_name,t.flight\_id,t.flight\_departure\_date,

count(t.ticket\_id) as No\_of\_Tickets

fromair\_passenger\_profile p join air\_ticket\_info t

onp.profile\_id=t.profile\_id

joinair\_flight f

ont.flight\_id=f.flight\_id

wheref.from\_location="chennai"

andf.to\_location="hyderabad"

group by p.profile\_id,t.flight\_id,t.flight\_departure\_date

order by p.profile\_id,t.flight\_id,t.flight\_departure\_date;

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.

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

SELECT fd.flight\_id,af.FROM\_LOCATION,af.TO\_LOCATION, fd.price FROM air\_flight\_detailsfd join air\_flightaf on af.FLIGHT\_ID=fd.FLIGHT\_ID where substring(fd.flight\_departure\_date,6,2)='04' order by fd.flight\_id, af.FROM\_LOCATION;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

AVG(fd.Price) AS Price

FROM air\_flight f JOIN

air\_flight\_detailsfd

ON f.flight\_id = fd.flight\_id

GROUP BY f.flight\_id order by f.flight\_id, f.from\_location, f.to\_location;

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.

SELECT DISTINCT c.profile\_id,

CONCAT(c.first\_name, ',',c.last\_name) AS customer\_name,

c.address

FROM air\_passenger\_profile c

JOIN air\_ticket\_info t

ON c.profile\_id = t.profile\_id

JOIN air\_flight f

ON f.flight\_id = t.flight\_id

AND f.from\_location = 'Chennai'

AND f.to\_location = 'Hyderabad'

order by c.profile\_id;

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.

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;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

COUNT(t.ticket\_id) AS No\_of\_Tickets

FROM air\_ticket\_infot JOIN

air\_flight f

ON f.flight\_id = t.flight\_id

where AIRLINE\_NAME = 'ABC ‘

GROUP by f.flight\_id

ORDER by f.flight\_id;

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

SELECT flight\_id,

COUNT(flight\_departure\_date) AS No\_of\_Services,

SUM(price) AS Total\_Price

FROM air\_flight\_details

GROUP BY flight\_id order by total\_price DESC, flight\_id DESC;

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.

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;

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.

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;

14.Write a query to display unique passenger profile id,firstname,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 ti.PROFILE\_ID,pi.first\_name,pi.mobile\_number, pi.email\_id FROM air\_ticket\_infoti join air\_passenger\_profile pi on pi.profile\_id=ti.profile\_id where flight\_id in (SELECT FLIGHT\_ID FROM air\_flight where FROM\_LOCATION ='HYDERABAD' and to\_location ='CHENNAI') order by ti.profile\_id

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.

SELECT DISTINCT p.profile\_id,

p.first\_name,

p.last\_name,

t.flight\_id,

t.flight\_departure\_date,

f.departure\_time,

f.arrival\_time,

ADDTIME(f.departure\_time,'01:00:00') AS Delayed\_Departure\_Time,

ADDTIME(f.arrival\_time,'01:00:00') AS Delayed\_Arrival\_Time

FROM air\_passenger\_profile p

JOIN air\_ticket\_info t

ON p.profile\_id = t.profile\_id

AND t.flight\_departure\_date = '2013-05-06'

JOIN air\_flight f

ON t.flight\_id = f.flight\_id

AND f.from\_location = 'Chennai'

AND f.to\_location = 'Hyderabad'

order by p.profile\_id;

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.

SELECT cus.profile\_id,

cus.First\_name,

SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')- INSTR(cus.address,',')-1)

AS Base\_Location,

s.No\_of\_Tickets

FROM air\_passenger\_profilecus JOIN

(

SELECT profile\_id,

COUNT(ticket\_id) AS No\_of\_Tickets

FROM air\_ticket\_info

GROUP BY profile\_id

)s

ON s.profile\_id = cus.profile\_id

AND SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')- INSTR(cus.address,',')-1) = 'Kochi'

order by first\_name;

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.

selectf.flight\_id,

f.from\_location,

f.to\_location,

count(at.flight\_departure\_date) as Number\_of\_Services

fromair\_flight f join air\_ticket\_info at

onat.flight\_id=f.flight\_id where month(at.flight\_departure\_date)=05

group by f.flight\_id order by f.flight\_id

18.Write a query to display profile id,lastname,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;

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';

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.

SELECT distinct ti.PROFILE\_ID,pi.first\_name,pi.email\_id,pi.mobile\_number FROM air\_ticket\_infoti join air\_passenger\_profile pi on pi.profile\_id=ti.profile\_id where flight\_id=3148 order by pi.first\_name;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

f.Departure\_Time,

CASE WHEN f.departure\_time BETWEEN ('05:00:01') AND ('12:00:00')

THEN 'Morning'

WHEN f.departure\_time BETWEEN ('12:00:01') AND ('18:00:00')

THEN 'AfterNoon'

WHEN f.departure\_time BETWEEN ('18:00:01') AND ('24:00:00')

THEN 'Evening'

WHEN f.departure\_time BETWEEN ('00:00:01') AND ('05:00:00')

THEN 'Night'

END AS Time\_of\_Service

FROM air\_flight f

order by f.flight\_id;

21.Please follow instructions given below.

Write a query to display flight id,departuredate,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.

selectflight\_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;

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.

SELECT CARD\_TYPE,count(card\_type) CARD\_COUNT FROM air\_credit\_card\_details group by CARD\_TYPE order by CARD\_TYPE;

23.Please follow instructions given below.

Write a Query to display serial no, first name,mobilenumber,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,4) SERIAL\_NO,first\_name,mobile\_number,email\_id from air\_passenger\_profile where email\_id like '%@gmail.com' order by first\_name;

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.

selectf.flight\_id,

f.from\_location,

f.to\_location,

count(at.flight\_departure\_date) as Number\_of\_Services

fromair\_flight f join air\_ticket\_info at

onat.flight\_id=f.flight\_id where month(at.flight\_departure\_date)=05

group by f.flight\_id

having count(at.flight\_departure\_date)<=all

(select count(at.flight\_departure\_date) from air\_flight f

joinair\_ticket\_info at on at.flight\_id=f.flight\_id

where month(at.flight\_departure\_date)=05 group by f.flight\_id);

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.

SELECT f.from\_location ,

COUNT(f.flight\_id) AS No\_of\_Flights

FROM air\_flight f

JOIN

(

SELECT DISTINCT from\_location AS Location

FROM air\_flight

UNION

SELECT DISTINCT to\_location AS Location

FROM air\_flight

)a

ON f.from\_location = a.location

GROUP BY f.from\_location

ORDER BY f.from\_location;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

t.FLIGHT\_DEPARTURE\_DATE,

t.No\_of\_Passengers

FROM air\_flight f

JOIN

(

SELECT flight\_id,

FLIGHT\_DEPARTURE\_DATE,

COUNT(ticket\_id) AS No\_of\_Passengers

FROM air\_ticket\_info

GROUP BY flight\_id,FLIGHT\_DEPARTURE\_DATE

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

f.Total\_Seats,

(f.Total\_Seats)-(fd.Available\_Seats) AS No\_of\_Seats\_Booked

FROM air\_flight f JOIN air\_flight\_detailsfd

ON f.flight\_id = fd.flight\_id

AND (f.Total\_Seats)-(fd.Available\_seats) > (f.total\_seats \*0.1)

order by f.flight\_id, No\_of\_Seats\_Booked;

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.

SELECT distinct f.flight\_id,

fd.flight\_departure\_date,

f.from\_location,

f.to\_location,

f.duration

FROM air\_flight f JOIN air\_flight\_detailsfd

ON f.flight\_id = fd.flight\_id

WHERE f.duration< '01:10:00'

order by f.flight\_id, fd.flight\_departure\_date;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

a.No\_of\_Services,

a.Average\_Price

FROM air\_flight f JOIN

(

SELECT flight\_id,

COUNT(flight\_departure\_date) AS No\_of\_Services,

AVG(price) AS Average\_Price

FROM air\_flight\_details

GROUP BY flight\_id

HAVING AVG(price) > (SELECT AVG(price) FROM air\_flight\_details)

)a

ON f.flight\_id = a.flight\_id order by average\_price ASC

2.AIRLINE MANAGEmeNT OLD

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.

ANS: SELECT f.flight\_id,

f.from\_location,

f.to\_location,

fp.Month\_Name,

fp.Average\_Price

FROM air\_flight f,

(

SELECT flight\_id,

MONTHNAME(flight\_departure\_date) AS Month\_Name,

AVG(price) as Average\_Price

FROM air\_flight\_details

GROUP BY flight\_id,

MONTHNAME(flight\_departure\_date)

) fp

WHERE f.flight\_id = fp.flight\_id

andf.airline\_name = 'ABC AIRLINES'

order by f.flight\_id, fp.Month\_Name;

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.

SELECT cus.profile\_id,

cus.first\_name,

cus.address,

mintkt.No\_of\_Tickets

FROM air\_passenger\_profilecus,

(SELECT MIN(s.Tot\_No\_of\_Tickets) AS No\_of\_Tickets

FROM (SELECT profile\_id,

COUNT(ticket\_id) AS Tot\_No\_of\_Tickets

FROM air\_ticket\_info

GROUP BY profile\_id

) s

) mintkt,

( SELECTprofile\_id,

COUNT(ticket\_id) AS Tot\_No\_of\_Tickets

FROM air\_ticket\_info

GROUP BY profile\_id

) tottkt

WHERE mintkt.No\_of\_Tickets = tottkt.Tot\_No\_of\_Tickets

AND cus.profile\_id = tottkt.profile\_id

order by cus.first\_name;

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.

SELECT f.from\_location,

f.to\_location,

s.Month\_Name,

SUM(s.No\_of\_Services) AS No\_of\_Services

FROM air\_flight f JOIN

(SELECT flight\_id,

MONTHNAME(flight\_departure\_date) AS Month\_Name,

COUNT(flight\_departure\_date) AS No\_of\_Services

FROM air\_flight\_details

GROUP BY flight\_id,

MONTHNAME(flight\_departure\_date)

) s

ON s.flight\_id =f.flight\_id

GROUP BY f.from\_location,

f.to\_Location,

s.Month\_Name

order by f.from\_location,

f.to\_Location, s.Month\_Name;

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.

SELECT cus.profile\_id,

cus.first\_name,

cus.address,

maxtkt.No\_of\_Tickets as No\_of\_Tickets

FROM air\_passenger\_profilecus,

(SELECT MAX(s.Tot\_No\_of\_Tickets) AS No\_of\_Tickets

FROM (SELECT profile\_id,

COUNT(ticket\_id) AS Tot\_No\_of\_Tickets

FROM air\_ticket\_info

GROUP BY profile\_id

) s

) maxtkt,

( SELECTprofile\_id,

COUNT(ticket\_id) AS Tot\_No\_of\_Tickets

FROM air\_ticket\_info

GROUP BY profile\_id

) tottkt

WHERE maxtkt.No\_of\_Tickets = tottkt.Tot\_No\_of\_Tickets

AND cus.profile\_id = tottkt.profile\_id

order by cus.first\_name;

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.

SELECT cus.profile\_id,

cus.first\_name,

cus.last\_name,

tkt.flight\_id,

tkt.flight\_departure\_date,

tkt.No\_of\_Tickets

FROM air\_passenger\_profilecus JOIN

(

SELECT profile\_id, FLIGHT\_ID,FLIGHT\_DEPARTURE\_DATE,COUNT(ticket\_id) AS No\_of\_Tickets

FROM air\_ticket\_info

WHERE flight\_id IN (SELECT flight\_id

FROM air\_flight

WHERE from\_location = 'Chennai'

AND to\_location = 'Hyderabad'

)

GROUP BY profile\_id,flight\_id,flight\_departure\_date

)tkt

ON cus.profile\_id = tkt.profile\_id

order by cus.profile\_id, tkt.flight\_id, tkt.flight\_departure\_date;

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

SELECT fd.flight\_id,af.FROM\_LOCATION,af.TO\_LOCATION, fd.price FROM air\_flight\_detailsfd join air\_flightaf on af.FLIGHT\_ID=fd.FLIGHT\_ID where substring(fd.flight\_departure\_date,6,2)='04' order by fd.flight\_id, af.FROM\_LOCATION;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

AVG(fd.Price) AS Price

FROM air\_flight f JOIN

air\_flight\_detailsfd

ON f.flight\_id = fd.flight\_id

GROUP BY f.flight\_id order by f.flight\_id, f.from\_location, f.to\_location;

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.

SELECT DISTINCT c.profile\_id,

CONCAT(c.first\_name, ',',c.last\_name) AS customer\_name,

c.address

FROM air\_passenger\_profile c

JOIN air\_ticket\_info t

ON c.profile\_id = t.profile\_id

JOIN air\_flight f

ON f.flight\_id = t.flight\_id

AND f.from\_location = 'Chennai'

AND f.to\_location = 'Hyderabad'

order by c.profile\_id;

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.

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;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

COUNT(t.ticket\_id) AS No\_of\_Tickets

FROM air\_ticket\_infot JOIN

air\_flight f

ON f.flight\_id = t.flight\_id

where AIRLINE\_NAME = 'ABC AIRLINES'

GROUP by f.flight\_id

ORDER by f.flight\_id;

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

SELECT flight\_id,

COUNT(flight\_departure\_date) AS No\_of\_Services,

SUM(price) AS Total\_Price

FROM air\_flight\_details

GROUP BY flight\_id order by total\_price DESC, flight\_id DESC;

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.

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;

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.

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;

14. Write a query to display unique passenger profile id,firstname,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 ti.PROFILE\_ID,pi.first\_name,pi.mobile\_number, pi.email\_id FROM air\_ticket\_infoti join air\_passenger\_profile pi on pi.profile\_id=ti.profile\_id where flight\_id in (SELECT FLIGHT\_ID FROM air\_flight where FROM\_LOCATION ='HYDERABAD' and to\_location ='CHENNAI') order by ti.profile\_id

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.

SELECT DISTINCT p.profile\_id,

p.first\_name,

p.last\_name,

t.flight\_id,

t.flight\_departure\_date,

f.departure\_time,

f.arrival\_time,

ADDTIME(f.departure\_time,'01:00:00') AS Delayed\_Departure\_Time,

ADDTIME(f.arrival\_time,'01:00:00') AS Delayed\_Arrival\_Time

FROM air\_passenger\_profile p

JOIN air\_ticket\_info t

ON p.profile\_id = t.profile\_id

AND t.flight\_departure\_date = '2013-05-06'

JOIN air\_flight f

ON t.flight\_id = f.flight\_id

AND f.from\_location = 'Chennai'

AND f.to\_location = 'Hyderabad'

order by p.profile\_id;

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.

SELECT cus.profile\_id,

cus.First\_name,

SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')- INSTR(cus.address,',')-1)

AS Base\_Location,

s.No\_of\_Tickets

FROM air\_passenger\_profilecus JOIN

(

SELECT profile\_id,

COUNT(ticket\_id) AS No\_of\_Tickets

FROM air\_ticket\_info

GROUP BY profile\_id

)s

ON s.profile\_id = cus.profile\_id

AND SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')- INSTR(cus.address,',')-1) = 'Kochi'

order by first\_name;

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.

SELECT f.flight\_id,

f.from\_location,

f.to\_location,

s.No\_of\_Services

FROM air\_flight f JOIN

(

SELECT flight\_id,

COUNT(flight\_departure\_date) AS No\_of\_Services

FROM air\_flight\_details

WHERE MONTH(flight\_departure\_date) = 5

GROUP BY flight\_id

) s

ON f.flight\_id = s.flight\_id

order by f.flight\_id;

18. Write a query to display profile id,lastname,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;

19. 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';

20. 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.

SELECT distinct ti.PROFILE\_ID,pi.first\_name,pi.email\_id,pi.mobile\_number FROM air\_ticket\_infoti join air\_passenger\_profile pi on pi.profile\_id=ti.profile\_id where flight\_id=3148 order by pi.first\_name;

3.EMPLOYEE MANAGEMENT SYSTEM

1.Write a query to display the employee ID, first name of the supervisors who has the maximum number of direct reportees (employees).

SELECT s.emp\_id, s.first\_name, count(e.emp\_id) as 'NO\_OF\_REPORTEES'

FROM emp\_details e, emp\_details s

WHERE s.emp\_id=e.supervisor\_id

GROUP BY s.emp\_id

HAVING NO\_OF\_REPORTEES IN ( SELECT MAX(c)

FROM ( SELECT count(\*) c

FROM emp\_details e

GROUP BY supervisor\_id)a

);

2.Write a query to find all employees who report to an employee 'Rahul' and who are not in any project and display the employee id, employee first name, designation ID, join date and supervisor id of those employees.

SELECT e.emp\_id, e.first\_name, e.designation\_id, e.join\_date, e.supervisor\_id

FROM emp\_details e

WHERE e.supervisor\_id = (SELECT emp\_id FROM emp\_details WHERE first\_name='Rahul')

AND e.emp\_id NOT IN (SELECT DISTINCT emp\_id FROM project\_alloc);

3.Write a query to display employee id,first name, last name, salary and designation id of employees who is getting maximum salary in each designation.

SELECT e.emp\_id, e.first\_name, e.last\_name, e.salary, e.designation\_id

FROM emp\_details e, (select designation\_id, max(salary) max\_sal FROM emp\_details GROUP BY designation\_id)e1

WHERE e.designation\_id = e1.designation\_id AND e.salary=e1.max\_sal AND e.designation\_id!='VP'

GROUP BY e.designation\_id

ORDER BY e.designation\_id desc;

4.Write to query to find for each project, find the number of employees who have either an 'OOP' or 'TEST' skill and display the project name and number of employees.

Hint: Use "NO\_OF\_EMPLOYEE" as alias name for displaying the number of employees.

SELECT p.proj\_name, count(e.emp\_id) as 'NO\_OF\_EMPLOYEE', s.skill\_name

FROM project\_details p, project\_alloc a, skill s, emp\_details e

WHERE e.emp\_id=a.emp\_id AND p.proj\_id=a.proj\_id AND e.skill\_id=s.skill\_id AND s.skill\_name IN ('OOP', 'TEST')

GROUP BY p.proj\_name;

5.Write a query to find all peoples who have either a 'JAV' or 'DES' skills and either 'E1' or 'E2' proficiency levels and

display the employee ID, employee first name, designation ID, skill ID, proficiency level and years of experience of those peoples.

SELECT e.emp\_id, e.first\_name, e.designation\_id, e.skill\_id, s.proficiency\_level, s.year\_of\_experience

FROM emp\_details e, skill s

WHERE e.skill\_id=s.skill\_id AND s.skill\_name IN ('JAV', 'DES') AND s.proficiency\_level IN ('E1', 'E2');

6.Write a query to display the first name, last name, salary, bonus (calculated 3 percent bonus per salary value) and

weekly salary (including bonus) of all employees and round the weekly salary to remove the decimals. Sort the result based on the salary in descending order.

Hint: Formula for calculating bonus is salary\*.03 and for calculating weekly salary is (salary/52) + ((salary\*.03)/52)

Use "BONUS" as alias name for displaying bonus and "WEEKLY\_SALARY" as alias name for displaying the weekly salary with bonus.

SELECT e.first\_name, e.last\_name, e.salary, 0.3\*(e.salary) as 'BONUS', round(((e.salary/52) + ((e.salary\*.03)/52)),0) as 'WEEKLY\_SALARY'

FROM emp\_details e

ORDER BY e.salary DESC;

7.Write a query to display the employee id and employee name of employee who is a developer in google testing team.

Display the employee name in below format ie first character of first name dot last name.

Example Assume first name is "Ram" and last name is "Kumar" then employee name will be "R.Kumar".

Hint: Use "EMPLOYEE\_NAME" as alias name for displaying employee name.

Example,

emp\_id EMPLOYEE\_NAME

1000 R.Kumar

SELECT e.emp\_id, concat(left(e.first\_name,1), '.', e.last\_name) as 'EMPLOYEE\_NAME'

FROM emp\_details e, dept\_details d, role r, project\_alloc a

WHERE e.dept\_no=d.dept\_no

AND e.emp\_id=a.emp\_id

AND a.role\_id=r.role\_id

AND d.dept\_name='Facebook Developer'

AND r.role\_name='Developer';

8.Write a query to find the all employees who are not currently in projects and are whose skill is 'DES' and proficiency level is 'E1' and

display the employee ID, employee first name, skill id, proficiency level and experience in years of those employees.

Hint: Employees who are not currently in projects means they either got released from the project or not at all allocated to any project.

SELECT e.emp\_id, e.first\_name, s.skill\_id, s.proficiency\_level, s.year\_of\_experience

FROM emp\_details e, skill s

WHERE e.skill\_id=s.skill\_id

AND s.skill\_name='DES'

AND s.proficiency\_level='E1'

AND e.emp\_id NOT IN (SELECT emp\_id FROM project\_alloc);

9.Write a query to find all people who have a 'JAV' skill and 'E2' proficiency level and display the distinct employee first name,

project ID, role ID, skill ID, proficiency level and release date of those peoples who are currently in projects.

Hint: Employees who are currently in projects means they should not have got released.

SELECT DISTINCT e.first\_name, a.proj\_id, a.role\_id, s.skill\_id, s.proficiency\_level

FROM emp\_details e, skill s, project\_alloc a

WHERE e.skill\_id=s.skill\_id

AND e.emp\_id=a.emp\_id

AND s.skill\_name='JAV'

AND s.proficiency\_level='E2';

10.Write a query to display the designation id, designation name and total salary of the employees who joined the company after 1-JAN-2004 whose total salary is greater than 50000 for each designation.

Hint: Use TOTALSALARY as alias name for displaying the total salary of the employees for each designation.

SELECT ds.designation\_id, ds.designation\_name, sum(e.salary) as 'TOTAL\_SALARY'

FROM designation ds, emp\_details e

WHERE ds.designation\_id=e.designation\_id AND e.join\_date>'2004-01-01'

GROUP BY ds.designation\_id

HAVING TOTAL\_SALARY>50000;

11.Write a query to find the all employees who are currently in projects and whose designation is 'C2' and display the employee ID, employee first name, designation ID, join date and supervisor ID of the employees.

Hint: Employees who are currently in projects means they are not released from project.

SELECT e.emp\_id, e.first\_name, e.designation\_id, e.join\_date, e.supervisor\_id

FROM emp\_details e

WHERE e.designation\_id='C2'

AND e.emp\_id IN (SELECT DISTINCT emp\_id FROM project\_alloc);

12.Write a query to display the employee id, employee first name, last name, designation id, join date and supervisor id of all employees who report to an employee 'Ram' and joined the company between&nbsp; '01-JAN-07' and '01-JAN-09' and who are not currently in any project (got released from projects).

Hint: Employees who are not currently in projects means they either got released from the project or not at all allocated to any project.

SELECT e.emp\_id, e.first\_name, e.last\_name, e.designation\_id, e.join\_date, e.supervisor\_id

FROM emp\_details e

WHERE e.supervisor\_id IN (SELECT emp\_id FROM emp\_details WHERE first\_name='Ram')

AND e.join\_date BETWEEN '2002-01-01' and '2010-01-01'

AND e.emp\_id NOT IN (SELECT DISTINCT emp\_id FROM project\_alloc);

13.Write a query to display the employee ID, employee first name, supervisor first name, designation name, project ID, project name, role id and role name of all employees for all the current and previous projects of all employees reporting to an employee 'Ram'.

Hint: Use SUPERVISOR\_NAME as alias name for displaying the supervisor first name.

SELECT DISTINCT e.emp\_id, e.first\_name, s.first\_name as 'SUPERVISOR\_NAME', ds.designation\_name, p.proj\_id, p.proj\_name, r.role\_id, r.role\_name

FROM emp\_details e, emp\_details s, designation ds, project\_details p, project\_alloc a, role r

WHERE e.designation\_id=ds.designation\_id

AND e.emp\_id=a.emp\_id

AND a.proj\_id=p.proj\_id

AND a.role\_id=r.role\_id

AND e.supervisor\_id=s.emp\_id

AND s.first\_name='Ram';

14.Write a query to display the employee id, employee first name, project ID and project name of the employees who have worked on the same project in the most number of roles.

SELECT e.emp\_id, e.first\_name, p.proj\_id, p.proj\_name, count(a.role\_id)

FROM emp\_details e, project\_details p, project\_alloc a

WHERE e.emp\_id=a.emp\_id AND a.proj\_id=p.proj\_id

GROUP BY e.emp\_id

HAVING count(p.proj\_id) IN ( SELECT max(c)

FROM ( SELECT e.emp\_id, e.first\_name, count(p.proj\_id) c, p.proj\_name, count(a.role\_id)

FROM emp\_details e, project\_details p, project\_alloc a

WHERE e.emp\_id=a.emp\_id AND a.proj\_id=p.proj\_id

GROUP BY e.emp\_id

)a1

)

AND count(a.role\_id) IN ( SELECT max(c)

FROM ( SELECT e.emp\_id, e.first\_name, p.proj\_id, p.proj\_name, count(a.role\_id) c

FROM emp\_details e, project\_details p, project\_alloc a

WHERE e.emp\_id=a.emp\_id AND a.proj\_id=p.proj\_id

GROUP BY e.emp\_id

)a12

);

15.Write a query to display the name, join date of the employees whose join date is greater than 01-JAN-2004 and sort the result based on the employee name in ascending order.

Display the name as, First Character of first name, Dot, Last Name.

Example: Assume first name is "Ram" and last name is "Kumar" then display employee name as "R.Kumar"

Hint: Use the EMPLOYEE\_NAME as alias name for displaying the employees name.

SELECT concat(substr(e.first\_name, 1,1), '.', e.last\_name) as EMPLOYEE\_NAME, e.join\_date

FROM emp\_details e

WHERE e.join\_date>'2004-01-01'

ORDER BY e.first\_name ASC;

16.Write a query to display the distinct employee id, employee first name, designation name and supervisor ID of all employees who work for the project "Google Testing".

Hint: There can be more than one employee working in same project with multiple roles, display their details only once.

SELECT DISTINCT e.emp\_id, e.first\_name, ds.designation\_name, e.supervisor\_id, p.proj\_name

FROM emp\_details e, designation ds, project\_details p, project\_alloc a

WHERE e.designation\_id=ds.designation\_id

AND e.emp\_id=a.emp\_id

AND a.proj\_id=p.proj\_id

AND p.proj\_name='GMS'

GROUP BY e.emp\_id;

17.Write a query to find the employee Id, first name, designation name, project name and role name of the employees who are second level reportees of a given employee 'Mark'.

Example, Assume, employee 'A' reports to "Mark" and employee "B" and "C" reports to employee "A". Here, display the above details for employee "B" and "C".

SELECT e.emp\_id, e.first\_name, ds.designation\_name, p.proj\_name, r.role\_name

FROM emp\_details e, designation ds, project\_details p, project\_alloc a, role r

WHERE e.designation\_id=ds.designation\_id

AND e.emp\_id=a.emp\_id

AND p.proj\_id=a.proj\_id

AND a.role\_id=r.role\_id

AND e.emp\_id IN ( SELECT e.emp\_id

FROM emp\_details e, emp\_details l1, emp\_details l2

WHERE e.supervisor\_id=l1.emp\_id

AND l1.supervisor\_id=l2.emp\_id

AND l2.first\_name='John'

);

18.Write a query to find the employees who have not worked for any projects till now and who are second level reportees of a given employee 'Mark'. Display the employee Id, first name and designation name of the employees.

Hint: Display all the employee details who have not working/worked for any projects.

Example, Assume, employee 'A' reports to "Mark" and employee "B" and "C" reports to employee "A". Here, display the above details for employee "B" and "C".

SELECT e.emp\_id, e.first\_name, ds.designation\_name

FROM emp\_details e, designation ds

WHERE e.designation\_id=ds.designation\_id

AND e.emp\_id NOT IN ( SELECT emp\_id

FROM project\_alloc

)

AND e.emp\_id IN ( SELECT e.emp\_id

FROM emp\_details e, emp\_details l1, emp\_details l2

WHERE e.supervisor\_id=l1.emp\_id

AND l1.supervisor\_id=l2.emp\_id

AND l2.first\_name='John'

);

19.Write a query to display the designation id, total salary for designation C1, total salary for designation C2, total salary for designation C3, total salary for designation C4, total salary for designation VP and total salary for all designations in the below format.

20.Write a query to display employee id,first name, last name,designation name and skill name of all the employees irrespective of skills available. If no skills are available then display it as "N/A" for that particular employee. If an employee has 2 skills then display both the records. Sort the records based on designation name in descending order.

Hint:Use SKILL\_NAME as alias name for skill name column.

SELECT e.emp\_id, e.first\_name, e.last\_name, ds.designation\_name, ifnull(s.skill\_name, 'N/A') as 'SKILL\_NAME'

FROM (emp\_details e LEFT OUTER JOIN skill s ON e.skill\_id=s.skill\_id) LEFT OUTER JOIN designation ds ON e.designation\_id=ds.designation\_id

ORDER BY ds.designation\_name DESC;

21.Write a query to display the employee id, employee first name, designation name, supervisor id, supervisor first name and supervisor designation ID of all employees including employees who does not have supervisor and sort them by employee designation name in ascending order.

Hint: Use "SUPERVISOR\_NAME" as alias name for displaying the supervisor first name and "SUPERVISOR\_DESIGNATION" as alias name for displaying the supervisor designation ID

SELECT e.emp\_id, e.first\_name, ds.designation\_name, s.emp\_id, s.first\_name as 'SUPERVISOR\_NAME', s.designation\_id as 'SUPERVISOR\_DESIGNATION'

FROM (emp\_details e LEFT OUTER JOIN emp\_details s ON e.supervisor\_id=s.emp\_id) LEFT OUTER JOIN designation ds ON e.designation\_id=ds.designation\_id

ORDER BY ds.designation\_name ASC;

4.MOVIE SCHEMA

1.count the members who has gold cards

ans:select count(customer\_id) from customer\_card\_details where card\_id in

(select card\_id from library\_card\_master where description='gold

card');

2.display the name of member who issued movie and the count of the

movies issued and display 0 for the member who have not issued any

movie

ans:select customer\_name,count(movie\_id) count

from customer\_issue\_details a,customer\_master b

where a.customer\_id=b.customer\_id

group by b.customer\_id union

select customer\_name,0 as count

from customer\_master

where customer\_id not in

(select customer\_id from customer\_issue\_details);

3.display the name of the person starting with letter 'r' and category

is 'comedy'

ans:

select distinct a.customer\_name

from customer\_master a,customer\_issue\_details b,movies\_master c

where a.customer\_id=b.customer\_id and b.movie\_id=c.movie\_id and c.movie\_category='comedy'

and a.customer\_name like 'r%';

4.display id,name & total rent of customers for movie issued

ans:select a.customer\_id,customer\_name,count(a.movie\_id)\*c.rent\_cost rent

from customer\_issue\_details a,customer\_master b ,(select movie\_id,rent\_cost from movies\_master) c

where a.customer\_id=b.customer\_id and a.movie\_id=c.movie\_id

group by

b.customer\_id;

5.display id,name,card id,amount in $(amount/54.42) upto 0 decimals

ans:

select a.customer\_id,customer\_name,b.card\_id,round(amount/54.42) Amount

from customer\_master a,customer\_card\_details b,library\_card\_master c

where a.customer\_id=b.customer\_id and

b.card\_id=c.card\_id;

6.display id,name of customers who dont have library card but still

have issued the movie

ans:

select distinct b.customer\_id,customer\_name

from customer\_issue\_details a,customer\_master b

where a.customer\_id=b.customer\_id

and a.customer\_id not in (select

customer\_id from customer\_card\_details);

7.display the no.of customers with first letter 'r' and have paid fine

i.e actual return date is greater than return date

ans:

select count(b.customer\_id) count from (select customer\_id

from customer\_issue\_details

where return\_date>actual\_date\_return

group by customer\_id) b,customer\_master c

where b.customer\_id=c.customer\_id and

c.customer\_name like 'r%';

8.display customer name,customer id who have issued max and min no.of

movies issued

ans:

select customer\_name,a.customer\_id

from customer\_issue\_details a,customer\_master b

where a.customer\_id=b.customer\_id

group by a.customer\_id

having count(a.movie\_id)=(

select min(a.count) min from (select customer\_id,count(movie\_id) count

from customer\_issue\_details

group by customer\_id) a)

or count(a.movie\_id)=(

select max(b.count) min from (select customer\_id,count(movie\_id) count

from customer\_issue\_details

group by customer\_id) b);

9.display id,name,mobile num and description of all customers.if

mobile num is not available then display address as alias contact,for

those who does't have library cards display null as description

ans:

select a.customer\_id,a.customer\_name,coalesce (a.contact\_no,a.contact\_add) contact,description

from customer\_master a,library\_card\_master b,customer\_card\_details c

where a.customer\_id=c.customer\_id and c.card\_id=b.card\_id union

select a.customer\_id,a.customer\_name,coalesce(a.contact\_no,a.contact\_add) contact,null as description

from customer\_master a

where customer\_id not in (select customer\_id from customer\_card\_details) ;

10.display customer details and movie id for those who issued same movie more than one time OR

//dislay customer details who watched same movie more than once

ans:select m.customer\_id,customer\_name,movie\_id,count(i.movie\_id) as count\_of\_movie

from customer\_master m join customer\_issue\_details i

on m.customer\_id = i.customer\_id

group by movie\_id

having count(movie\_id)>1;

11.display customer information those who has library cards

ans:

select a.customer\_id,customer\_name

from customer\_card\_details a,customer\_master b,library\_card\_master c

where

a.customer\_id=b.customer\_id and a.card\_id=c.card\_id;

12.display the members who watch the movie but doesnt have card

ans:select distinct b.customer\_id,customer\_name

from customer\_issue\_details a,customer\_master b

where a.customer\_id=b.customer\_id

and a.customer\_id not in (select

customer\_id from customer\_card\_details);

13.display sr no as 2 digits of issue id,emp id,movie watched,video id

and sort by sr no

ans:

select substring(issue\_id,4) sr\_no,issue\_id as video\_id,customer\_id,movie\_id

from customer\_issue\_details order by

sr\_no;

14.display total revenue spent on videos by each customer

ans:select a.customer\_id,count(a.movie\_id)\*c.rent\_cost rent

from customer\_issue\_details a,customer\_master b ,(select

movie\_id,rent\_cost from movies\_master) c

where a.customer\_id=b.customer\_id and a.movie\_id=c.movie\_id

group by b.customer\_id union

select customer\_id,0 as rent

from customer\_master

where customer\_id not in(select

customer\_id from customer\_issue\_details);

15.display customer name in perfect order

i.e 1st letter in ucase remaining lcase

ans:

select concat(ucase(substring(customer\_name,1,1)),lcase(substring (customer\_name,2,5)) )

from customer\_master;

16.count how many times a movie issued and arrange them in desc order

and display 0 for the movie not issued

ans:

select \* from (select a.movie\_id,count(b.movie\_id) count

from movies\_master a,customer\_issue\_details b

where a.movie\_id=b.movie\_id

group by a.movie\_id union

select movie\_id ,0 as count from movies\_master

where movie\_id not in (select movie\_id from

customer\_issue\_details)) x order by movie\_id desc;

17.waq to display cus id and cus name and address as if phone num

presents display phone num otherwise address.

SELECT CUSTOMER\_ID,CUSTOMER\_NAME,coalesce(CAST(CONTACT\_NO AS CHAR),CAST(CONTACT\_ADD AS CHAR))

from customer\_master;

SELECT CUSTOMER\_ID,CUSTOMER\_NAME,(CASE

WHEN CAST(CONTACT\_NO AS CHAR) IS NULL THEN CAST(CONTACT\_ADD AS CHAR)

ELSE CAST(CONTACT\_NO AS CHAR)

END) AS CONTACT

FROM CUSTOMER\_MASTER;

18.waq that num of customers registered in 2012 year and provided contact num

use NO\_OF\_CUSTOMERS as alias name.

ANS:SELECT COUNT(\*) AS NO\_OF\_CUSTOMERS

FROM CUSTOMER\_MASTER

WHERE DATE\_OF\_REGISTRATION LIKE '%2012%' AND CONTACT\_NO IS NOT NULL;

19.display customer id, cus name,year of registration,library card id,card issue date

alias name registered\_year for year of registration.

ANS:SELECT seLect C.CUSTOMER\_ID,C.CUSTOMER\_NAME,year (C.DATE\_OF\_REGISTRATION) REGISTERED\_YEAR,L.CARD\_ID,D.ISSUE\_DATE

FROM CUSTOMER\_MASTER C,LIBRARY\_CARD\_MASTER L,CUSTOMER\_CARD\_DETAILS D

WHERE

C.CUSTOMER\_ID=D.CUSTOMER\_ID AND L.CARD\_ID=D.CARD\_ID;

20. (\*\*\*REPEATED)waq to display movie name and num of times movie issued to

customers..in case of no movie issued

to customers display 0.. use alias name as NO\_OF\_TIMES

SELECT M.MOVIE\_NAME,COUNT(\*) AS NO\_OF\_TIMES

FROM CUSTOMER\_ISSUE\_DETAILS D,MOVIES\_MASTER M,CUSTOMER\_MASTER C

WHERE M.MOVIE\_ID=D.MOVIE\_ID AND D.CUSTOMER\_ID=C.CUSTOMER\_ID

GROUP BY C.CUSTOMER\_ID

union

SELECT MOVIE\_NAME,0

FROM MOVIES\_MASTER

WHERE MOVIE\_NAME NOT IN (SELECT M.MOVIE\_NAME

FROM CUSTOMER\_ISSUE\_DETAILS D,MOVIES\_MASTER M,CUSTOMER\_MASTER C

WHERE M.MOVIE\_ID=D.MOVIE\_ID AND D.CUSTOMER\_ID=C.CUSTOMER\_ID

GROUP BY C.CUSTOMER\_ID);

21.waq to display customer id and customer name ,num of times movie issued to customer in comedy movie category

display only customers who has issude more than once

ANS:

SELECT M.CUSTOMER\_NAME,M.CUSTOMER\_ID,P.NO\_OF\_TIMES

FROM CUSTOMER\_MASTER M,

(SELECT I.CUSTOMER\_ID,COUNT(\*) AS NO\_OF\_TIMES

FROM CUSTOMER\_ISSUE\_DETAILS I,(

SELECT MOVIE\_ID,MOVIE\_CATEGORY

FROM MOVIES\_MASTER

WHERE MOVIE\_CATEGORY='COMEDY')T

WHERE I.MOVIE\_ID=T.MOVIE\_ID

GROUP BY I.CUSTOMER\_ID

HAVING COUNT(\*)>1)P

WHERE M.CUSTOMER\_ID=P.CUSTOMER\_ID;

22.(\*\*\*REPEATED)waq to display customerid and total rent paid by them.

use alias name as total\_cost.

ANS:select \* from movies\_master;

select c.customer\_id,sum(rent\_cost) as total\_cost

from customer\_master c,customer\_issue\_details d,movies\_master m

where c.customer\_id=d.customer\_id and m.movie\_id=d.movie\_id

group by customer\_id;

23.waq to display customerid,cusname,contactno,num of movies issued to customer based on category and category

display the customer who has issude for more than one movie from that caregory.

display phone num as "+91-987-654-3210".

ANS:select m.customer\_id,customer\_name,

concat('+91-',substring(contact\_no,1,3),'-',substring (contact\_no,4,3),'-',substring(contact\_no,7) )as contact\_no,

count(i.movie\_id) as no\_of\_movies,movie\_category

from customer\_master m join customer\_issue\_details i join movies\_master mv

on m.customer\_id = i.customer\_id and i.movie\_id = mv.movie\_id

group by m.customer\_id

having count(\*) >1;

5.MOBILE MANAGEMENT SCHEMA

1. Write a Query to Display the IME Number, Model Name of mobiles which is manufactured by "Nokia".

select IME\_NO, MODEL\_NAME from mobile\_master where manufacturer = 'Nokia';

2. Write a Query to display IME number, Model Name, Manufacturer and Camera Quality of mobiles whose camera quality is 5MP.

select m.IME\_NO, m.MODEL\_NAME, m.MANUFACTURER, s.CAMERA\_QUALITY from mobile\_master m join mobile\_specification s on(m.ime\_no = s.ime\_no)

where camera\_quality = '5MP';

3. "Write a Query to display the Mobile Model Name and respective number of mobiles sold on the date 23-Apr-2012 for each mobile model.

Hint: For example, if 2 ""Nokia 1100"" and 1 ""Nokia C5-03"" are sold on the date 23-Apr-2012 then display both the records. Use ""NoofMobilesSold"" as alias name for the number of mobiles field."

select model\_name, count(model\_name) NoOfMobilesSold from sales\_info

where sales\_date = '2012-04-23' group by model\_name;

4. "Write a Query to display the distributor id, mobile model name, number of mobiles of the particular model name supplied to the distributors group by model name and distributor id and sort by the distributor id.

Hint: For example, if 3 ""Nokia 1100"" and 1 ""Nokia C5-03"" are sold to one distributor then display both the records. Display the distributor id, model name and number of mobiles of a particular model name. Use ""NoofMobilesSupplied"" as alias name for the number of mobiles."

select d.distributor\_id, m.model\_name, count(model\_name) NoOfMobilesSupplied from distributor d join mobile\_master m on(d.distributor\_id = m.distributor\_id) group by m.model\_name, m.distributor\_id order by m.distributor\_id;

5. "Write a Query to display the IME number, model name, manufacturer, price and discount of all mobiles regardless of whether the mobile is sold or not. Hint: Fetch the price, IME no and model name from mobile\_master table.

Example: For the mobile model ""Samsung GalaxyTAB with IME NO ""MC1000103"" is sold and other with IME No ""MC1000110"" is not sold. Then both the mobiles details namely IME number, model name, manufacturer, price and discount needs to be displayed.

select DISTINCT m.ime\_no, m.model\_name, m.manufacturer, m.price, s.discount from mobile\_master m join sales\_info s;

6. Write a Query to display the distributor name, mobile number and email of distributors selling model 'Nokia 1100'.

select d.distributor\_name, d.mobile, d.email from distributor d join mobile\_master m

on(d.distributor\_id = m.distributor\_id) where model\_name = 'Nokia 1100';

7. Write a Query to display the IME Number and Model Name of mobiles which are not sold. Hint: The details of the sold mobiles are available in the "SALES\_INFO" table and the overall mobile models are available in the mobile\_master table.

select ime\_no, model\_name from mobile\_master where model\_name

NOT IN( select model\_name from sales\_info);

8. Write a Query to display the IME Number, Model Name and net amount of the mobile which has the highest net amount.

select m.ime\_no, m.model\_name, max(s.net\_amount) Net\_Amount from mobile\_master m

join sales\_info s on(m.ime\_no = s.ime\_no);

9. "Write a Query to display the IME Number, Model Name, Manufacturer, Price and New Price of all mobiles

select ime\_no, model\_name, manufacturer, price, price + (price \* 10/100) New\_Price from mobile\_master;

10. Write a Query to display mobile model name, manufacturer and price for the mobiles having a price range between 8500 and 25300

select model\_name, manufacturer, price from mobile\_master where price >= 8500 and price <= 25300 ;

11. Write a Query 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

select m.Model\_Name, m.Manufacturer, m.Price, m.Warranty\_in\_Years, s.Internal\_mem\_in\_MB, s.Memory\_Card\_Capacity\_GB, s.GPRS, s.Bluetooth,s.Camera\_Quality, s.OS from mobile\_master m join mobile\_specification s on(m.ime\_no = s.ime\_no) where m.ime\_no = 'MC1000104';

12. "Write a Query to display IME Number, Model Name, Manufacturer, Price ,GPRS information, Memory card capacity of mobiles which has GPRS support with memory card capacity 16GB or above.

Hint: For GPRS support use GPRS = “Yes”."

select m.IME\_No, m.Model\_Name, m.Manufacturer, m.Price, s.GPRS, s.Memory\_Card\_Capacity\_GB from mobile\_master m join mobile\_specification s on(m.ime\_no = s.ime\_no) where s.GPRS = 'yes' and s.Memory\_Card\_Capacity\_GB >= '16GB';

13. Write a Query to display the customer name, IME Number, Model Name, Sales Date and Net amount paid by the customer and sort by customer name in ascending order.

select c.Customer\_Name, s.IME\_No, s.Model\_Name, s.Sales\_Date, s.Net\_Amount from customer\_info c join sales\_info s on( c.Customer\_ID = s.Customer\_ID ) group by c.Customer\_Name;

14. "Write a Query to display the IME Number, 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""

select m.IME\_No, m.Model\_Name, m.Manufacturer, m.Price, IFNULL(s.Discount, 'NotSold') discount from mobile\_master m left outer join sales\_info s on(m.ime\_no = s.ime\_no);

15. Write a Query to display the sales date and total net amount of all the mobiles based on the sales date that are sold between 20-APR-12 and 25-APR-12. Hint: Total net amount column should be displayed as "TotalNetAmount" (alias)

select Sales\_Date, sum(Net\_Amount) TotalNetAmount from sales\_info where Sales\_Date between '2012-04-12' and '2012-04-25' group by sales\_date;

16. Write a Query to display mobile IME number, model name, manufacturer, price and battery life of the mobiles which are having the longest battery life. Hint: Use the field "battery\_life\_hrs" for calculating maximum battery life.

select m.IME\_No, m.Model\_Name, m.Manufacturer, m.Price, s.Battery\_Life\_Hrs

from mobile\_master m join mobile\_specification s on( m.ime\_no = s.ime\_no )

where s.Battery\_Life\_Hrs = (select max(Battery\_Life\_Hrs) from mobile\_specification);

17 Write a Query to display the IME Number, Model Name, Manufacturer and Price of the mobile which is having the maximum price. Hint: Assume 2 mobiles have highest price then both the mobiles should be displayed

select IME\_No, Model\_Name, Manufacturer, Price from mobile\_master

where price = (select max(price) from mobile\_master);

18 Write a Query to display the Customer ID, Customer Name, Address, Total net amount of each customer. For example, assume customer\_1 has purchased 2 mobiles such as "Nokia C5-03" and "Nokia Lumia" then sum the prices of both the mobiles and should be displayed against his customer id. Hint: Use Total\_Net\_Amount as alias.

select c.Customer\_ID, c.Customer\_Name, c.Address, sum(s.Net\_Amount) Total\_net\_amount

from customer\_info c join sales\_info s on(c.customer\_id = s.customer\_id)

19 Write a Query to display the unique mobile model, manufacturer and price of the mobile which has highest price and manufactured by "Samsung".

select DISTINCT Model\_Name, Manufacturer, max(Price) from mobile\_master

where Manufacturer = 'Samsung';

20 Write a Query to display the IME number, model name, manufacturer, distributor id, distributor name and price supplied by the distributor named 'AXA Ltd'.

select m.IME\_No, m.Model\_Name, m.Manufacturer, m.Distributor\_ID, d.Distributor\_Name, m.Price from mobile\_master m join distributor d on( m.distributor\_id = d.distributor\_id)

21 Write a Query to display the distributor id, name, address , mobile no, email of the distributor who has supplied the maximum number of mobiles. Hint: Get the maximum number of mobile provided by a distributor from mobile master and use it to get the details

select DISTINCT d.Distributor\_ID, d.Distributor\_Name, d.Address, d.Mobile, d.Email

from distributor d join mobile\_master m on(d.distributor\_id = m.distributor\_id)

where d.Distributor\_ID = (select min(m.Distributor\_ID) from mobile\_master m

join distributor d on(m.Distributor\_ID = d.distributor\_id));

22 "Write a Query to display the Customer ID, Customer Name and Address of the customers who have purchased the maximum amount.

select c.Customer\_ID, c.Customer\_Name, c.Address from customer\_info c

join sales\_info s on( c.customer\_id = s.customer\_id ) group by c.Customer\_Name

having sum(net\_amount) =

(select max(s) from (select sum(net\_amount) s from sales\_info group by customer\_id)t);

23 Write a Query to determine whether the mobile "Samsung GalaxyTAB" has been sold out or not and display the model name, ime\_no and sales status. If sold display status as "Sold Out" else display "-"with column name "SalesStatus" (alias).

select m.model\_name, m.ime\_no, IF(s.discount,'Sold','NotSold') SalesStatus from mobile\_master m left outer join sales\_info s on(m.ime\_no = s.ime\_no) where m.model\_name = 'Samsung GalaxyTAB';

24 Write a Query to display the distinct distributor id, distributor name, address, mobile of all the distributors who supplies mobile with the following specifications network should support 3G Network and OS should be Android and camera quality should be 3.5 MP Camera. Hint: The above specifications are found in the "Mobile\_Specification" table.

select DISTINCT d.distributor\_id, d.Distributor\_Name, d.Address, d.Mobile from distributor d

join mobile\_master m on(d.distributor\_id = m.distributor\_id)

join mobile\_specification s on(m.ime\_no = s.ime\_no)

where s.Network\_3G = 'yes' and os like 'Android%' and camera\_quality = '3.5MP';

25 "Write a Query to Display the unique mobile model name and manufacturer of the mobile which has highest sales. Hint: Get the maximum count of mobile model from Sales info and get the manufacturer using the mobile model.

select s.Model\_Name, m.Manufacturer from sales\_info s

join mobile\_master m on (s.ime\_no = m.ime\_no) group by s.model\_name having count(\*) =

(select max(cnt )from (select count(\*) cnt from sales\_info group by model\_name)t);

6.Bank Management Schema

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.

ANS: 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.

ANS: 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.

ANS: 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.

ANS: 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.

ANS: SELECT transaction\_type,count(transaction\_number) Trans\_Count

FROM account\_masteram 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.

ANS: 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.

ANS: SELECT td.account\_number, opening\_balance+sum(transaction\_amount) Deposit\_Amount

FROM account\_masteram 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.

ANS: 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.

ANS: selectfirstname

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.

ANS: SELECT ld.customer\_number, firstname, lastname

FROM customer\_master cm INNER JOIN loan\_detailsld

ON cm.customer\_number=ld.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.

ANS: selectcustomer\_master.customer\_number, firstname, customer\_city, branch\_city

fromaccount\_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

wherecustomer\_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.

ANS: SELECT count(ld.customer\_number) Count

FROM customer\_master cm INNER JOIN loan\_detailsld

ON cm.customer\_number=ld.customer\_number

WHERE cm.customer\_number NOT IN ( SELECTcustomer\_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.

ANS: SELECT td.account\_number

FROM account\_masteram 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\_masteram 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.

ANS: selectbranch\_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.

ANS: 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\_masteram 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).

ANS: SELECT (SUM(CASE WHEN transaction\_type='Deposit'

THEN transaction\_amount END)) -

(SUM(CASE WHEN transaction\_type='Withdrawal'

THEN transaction\_amount END))+(select opening\_balance

fromaccount\_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.

ANS: 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.

ANS: 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.

ANS: SELECT ld.customer\_number, firstname, lastname

FROM customer\_master cm INNER JOIN loan\_detailsld

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.

ANS: SELECT ld.customer\_number, firstname,branch\_id, loan\_amount

FROM customer\_master cm INNER JOIN loan\_detailsld

ON cm.customer\_number=ld.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.

ANS: 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’slastname for the customer’s whose account is Active.

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

ANS: 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.

ANS: 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.

ANS: 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 customersfirstname, 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.

ANS: 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

7. ITEM LOAN

1.Please follow instructions given below.

<br>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.

ANS: SELECT item\_category , count(item\_id) Count\_category

FROM item\_master

GROUP BY item\_category order by count\_category DESC

2. Please follow instructions given below.

<br>Write a query to display the number of employees in HR department. Give the alias name as No\_of\_Employees.

ANS: SELECT count(employee\_id) AS No\_of\_Employees

FROM employee\_master

WHERE department= 'HR'

3.Please follow instructions given below.

<br><br>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.

ANS: SELECT employee\_id, employee\_name, designation, department

FROM employee\_master WHERE employee\_id

NOT IN ( SELECTemployee\_id FROM employee\_issue\_details)

order by employee\_id;

4. Please follow instructions given below.

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

ANS: SELECT eid.employee\_id, employee\_name

FROM employee\_masterem INNER JOIN employee\_issue\_detailseid

ON em.employee\_id=eid.employee\_id

INNER JOIN item\_masterim

ON eid.item\_id=im.item\_id

WHERE item\_valuation=(SELECT max(item\_valuation)

FROM employee\_issue\_detailseid INNER JOIN item\_masterim

ON eid.item\_id=im.item\_id) order by eid.employee\_id;

5. Please follow instructions given below.

<br>Write a query to display issue\_id, employee\_id, employee\_name.

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

ANS: SELECT issue\_id, eid.employee\_id, employee\_name

FROM employee\_masterem INNER JOIN employee\_issue\_detailseid

ON em.employee\_id=eid.employee\_id order by issue\_id;

6. Please follow instructions given below.

<br>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.

ANS: SELECT employee\_id, employee\_name

FROM employee\_master

WHERE employee\_id NOT IN ( SELECTemployee\_id FROM employee\_card\_details )

order by employee\_id;

7.Please follow instructions given below.

<br>Write a query to count the number of cards issued to an employee “Ram”. Give the count an alias name as No\_of\_Cards.

ANS: SELECT count(loan\_id) AS No\_of\_Cards

FROM employee\_card\_details c

JOIN employee\_master e

ON c.employee\_id = e.employee\_id

WHERE e.employee\_name= 'Ram'

GROUP BY c.employee\_id

8.Please follow instructions given below.

<br>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.

ANS: SELECT count(employee\_id) Count\_stationary

FROM employee\_card\_detailsecd INNER JOIN loan\_card\_masterlcd

ON ecd.loan\_id=lcd.loan\_id

WHERE loan\_type='stationary'

9. Please follow instructions given below.

<br>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.

ANS: SELECT eid.employee\_id, employee\_name, count(item\_id) Count

FROM employee\_masterem INNER JOIN employee\_issue\_detailseid

ON em.employee\_id=eid.employee\_id

GROUP BY employee\_id order by count DESC, eid.employee\_id;

10. Please follow instructions given below.

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

<br>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.]

ANS: SELECT eid.employee\_id, employee\_name

FROM employee\_masterem INNER JOIN employee\_issue\_detailseid

ON em.employee\_id=eid.employee\_id

INNER JOIN item\_masterim

ON eid.item\_id=im.item\_id

WHERE item\_valuation=(SELECT min(item\_valuation)

FROM employee\_issue\_detailseid INNER JOIN item\_masterim

ON eid.item\_id=im.item\_id)

order by eid.employee\_id;

11. Please follow instructions given below.

<br>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.

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

<br>Consider only employees who have been issued atleast 1 item.

ANS: SELECT em.employee\_id, employee\_name, sum(item\_valuation) TOTAL\_VALUATION

FROM employee\_masteremINNER JOINemployee\_issue\_detailseid

ON em.employee\_id=eid.employee\_id

INNER JOIN item\_masterim

ON eid.item\_id=im.item\_id

GROUP BY eid.employee\_id

ORDER BY eid.employee\_id;

12. Please follow instructions given below.

<br>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.

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

ANS: SELECT DISTINCT eid.employee\_id,employee\_name

FROM employee\_issue\_detailseid JOIN employee\_masterem

ON eid.employee\_id=em.employee\_id

WHERE datediff(return\_date,issue\_date) > 365

order by eid.employee\_id;

13. Please follow instructions given below.

<br>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.

<br>Display the records sorted in ascending order on employee id.

ANS: SELECT eid.employee\_id, employee\_name, count(eid.item\_id) COUNT\_ITEMS

FROM employee\_issue\_detailseid JOIN item\_masterim

ON eid.item\_id=im.item\_id

JOIN employee\_masterem

ON eid.employee\_id=em.employee\_id

WHERE item\_category='furniture'

GROUP BY employee\_id

HAVING COUNT\_ITEMS>1

order by eid.employee\_id;

14. Please follow instructions given below.

<br>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.

ANS: SELECT gender , count(gender) No\_of\_Employees

FROM employee\_master

GROUP BY gender ORDER BY gender

15. Please follow instructions given below.

<br>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.

ANS: SELECT employee\_id, employee\_name

FROM employee\_master

WHERE extract(year from date\_of\_joining)>2005

order by employee\_id;

16. Please follow instructions given below.

<br>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.

ANS: SELECT issue\_status, count(item\_id) No\_of\_Furnitures

FROM item\_master

WHERE item\_category='Furniture'

GROUP BY issue\_status

ORDER BY issue\_status

17. Please follow instructions given below.

<br>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.

ANS: SELECT item\_category, item\_make,item\_description, count(item\_description) No\_of\_Items FROM item\_master

GROUP BY item\_category, item\_make, item\_description ORDER BY item\_category, item\_make,item\_description;

18. Please follow instructions given below.

<br>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.

ANS: SELECT eid.employee\_id, employee\_name, eid.item\_id, item\_description

FROM employee\_issue\_detailseid JOIN employee\_masterem

ON eid.employee\_id=em.employee\_id

JOIN item\_masterim

ON eid.item\_id=im.item\_id

WHERE extract(month from issue\_date)=1 and extract(year from issue\_date)=2013

ORDER BY eid.employee\_id, eid.item\_id;

19. Please follow instructions given below.

<br>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.

ANS: SELECT em.employee\_id,employee\_name, count(distinct item\_category) COUNT\_CATEGORY

FROM employee\_issue\_detailseid JOIN item\_masterim

ON eid.item\_id=im.item\_id

JOIN employee\_masterem

ON eid.employee\_id=em.employee\_id

GROUP BY employee\_id

HAVING COUNT\_CATEGORY>=2

ORDER BY employee\_id;

20. Please follow instructions given below.

<br>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.

ANS: SELECT item\_id, item\_description

FROM item\_master

WHERE item\_id

NOT IN ( SELECTitem\_id from employee\_issue\_details)

ORDER BY item\_id;

21. Please follow instructions given below.

<br>Write a query to display the employee id, employee name and&nbsp;&nbsp;totalvaluation&nbsp;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. ]

ANS: select employee\_issue\_details.employee\_id,employee\_master.employee\_name,sum(item\_master.item\_valuation)as TOTAL\_VALUATION from

employee\_issue\_details inner join item\_master

onitem\_master.item\_id = employee\_issue\_details.item\_id

inner join employee\_master

onemployee\_master.employee\_id=employee\_issue\_details.employee\_id

group by employee\_issue\_details.employee\_id

order by TOTAL\_VALUATION asc limit 1;

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'. ]

ANS: SELECT ecd.employee\_id,employee\_name,

card\_issue\_date, CASE duration\_in\_years

WHEN 0 THEN 'No Validity Date'

ELSE DATE\_ADD(card\_issue\_date, INTERVAL duration\_in\_years YEAR)

END CARD\_VALID\_DATE

FROM employee\_masterem INNER JOIN

employee\_card\_detailsecd

ON em.employee\_id=ecd.employee\_id

INNER JOIN loan\_card\_masterlcd

ON ecd.loan\_id=lcd.loan\_id

order by employee\_name, CARD\_VALID\_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.

ANS: SELECT DISTINCT a.employee\_id,b.employee\_name FROM employee\_issue\_details a JOIN

employee\_master b ON a.employee\_id=b.employee\_id WHERE a.employee\_id NOT IN(SELECT employee\_id

FROM employee\_issue\_details WHERE (EXTRACT(YEAR FROM issue\_date)=2013))

ORDER BY a.employee\_id;

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.

ANS: SELECT issue\_id, eid.employee\_id, employee\_name, im.item\_id, item\_description,issue\_date

FROM employee\_issue\_detailseid INNER JOIN employee\_masterem

ON eid.employee\_id=em.employee\_id

INNER JOIN item\_masterim

ON eid.item\_id=im.item\_id

ORDER BY issue\_datedesc, issue\_idasc;

25. Please follow instructions given below.

<br>Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product.&nbsp; Give the alias name for total valuation as TOTAL\_VALUATION.&nbsp;

<br>[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. ]

ANS: select employee\_issue\_details.employee\_id,employee\_master.employee\_name,sum(item\_master.item\_valuation)as TOTAL\_VALUATION from

employee\_issue\_details inner join item\_master

onitem\_master.item\_id = employee\_issue\_details.item\_id

inner join employee\_master

onemployee\_master.employee\_id=employee\_issue\_details.employee\_id

group by employee\_issue\_details.employee\_id

order by TOTAL\_VALUATION desc limit 1;

8. VIDEO LIBRARY

1.Please follow instructions given below.

Write a query to display movie names and number of times that movie is issued to customers. Incase 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.

ANS: SELECT movie\_name, count(cid.movie\_id) ISSUE\_COUNT FROM movies\_master mm LEFT JOIN customer\_issue\_detailscid ON mm.movie\_id=cid.movie\_id GROUP BY movie\_name ORDER BY Issue\_count DESC, movie\_name;

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 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.

ANS: select customer\_id,customer\_name, age, coalesce(concat('+91-',substring(contact\_no,1,3),'-',substring(contact\_no,4,3),'-',substring(contact\_no,7,4)),'N/A') CONTACT\_ISD from customer\_master where age>25 and substring(date\_of\_registration,1,4)=2012 order by age asc, customer\_name;

3. Please follow instructions given below.

<br>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.

<br/>Hint: Use NO\_OF\_MOVIES as alias name for number of movies.

ANS: SELECT movie\_category, count(movie\_id) NO\_OF\_MOVIES FROM movies\_master GROUP BY movie\_category order by no\_of\_moviesdesc, movie\_categoryasc;

4.Please follow instructions given below.

<br>Write a query to display the number of customers having card with description “Gold card”. <br/>Hint: Use CUSTOMER\_COUNT as alias name for number of customers

ANS: SELECT count(customer\_id) CUSTOMER\_COUNT FROM library\_card\_master lcm INNER JOIN customer\_card\_detailsccd ON lcm.card\_id=ccd.card\_id WHERE description='Gold Card';

5.Please follow instructions given below.<br>

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.

<br> Use REGISTERED\_YEAR as alias name for year of registration.

ANS: SELECT c.customer\_id, c.customer\_name, extract(year from 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 c.customer\_name DESC;

6. Please follow instructions given below.<br>

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.

<br>

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

ANS: SELECT issue\_id ,cid.customer\_id, customer\_name FROM customer\_issue\_detailscid INNER JOIN customer\_master cm ON cid.customer\_id=cm.customer\_id WHERE actual\_date\_of\_return>return\_date and customer\_name like 'R%' order by customer\_name;

7. 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.

ANS: 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\_detailsccd ON cm.customer\_id=ccd.customer\_id INNER JOIN library\_card\_master lcm ON ccd.card\_id=lcm.card\_id INNER JOIN customer\_issue\_detailscid ON cid.customer\_id = cm.customer\_id WHERE cm.date\_of\_registration=cid.issue\_date order by customer\_name;

8. 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.

ANS: 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 ;

9. Please follow instructions given below.

Write a query to display the movie id, movie name,releaseyear,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.

ANS: 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;

10. Please follow instructions given below.

<br>

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. <br>Hint: Use MOVIE\_COUNT as alias name for number of movies issued.

ANS: SELECT customer\_name, count(movie\_id) MOVIE\_COUNT FROM customer\_master cm LEFT OUTER JOIN customer\_issue\_detailscid ON cm.customer\_id=cid.customer\_id GROUP BY customer\_name ORDER BY customer\_name;

11. Please follow instructions given below.

<br>

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.

<br/>Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.

<br/>For Example Assume the issue id is I00005 then the serial number is 05

<br/>Hint: Alias name for serial number is 'SERIAL\_NO'<br/><br/><br>

ANS: SELECT substring(issue\_id,5,2) SERIAL\_NO,issue\_id, cid.customer\_id,customer\_name, mm.movie\_id, movie\_name FROM customer\_issue\_detailscid INNER JOIN customer\_master cm ON cid.customer\_id=cm.customer\_id INNER JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id order by SERIAL\_NO ASC;

12. Please follow instructions given below.

<br>

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 decending order based on issue date of the video.

ANS: SELECT issue\_id, issue\_date, cid.customer\_id, customer\_name,contact\_no FROM customer\_issue\_detailscid INNER JOIN customer\_master cm ON cid.customer\_id=cm.customer\_id where extract(year from issue\_date)=2013 order by issue\_date DESC

13. Please follow instructions given below.

<br>

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <br> 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. <br> Display the records in ascending order based on movie name.

ANS: SELECT movie\_id, movie\_name,concat(lead\_actor\_name1,' & ',lead\_actor\_name2) ACTORS FROM movies\_master WHERE movie\_id NOT IN ( SELECT movie\_id from customer\_issue\_details ) order by movie\_nameasc;

14.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.

ANS: 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\_namedesc;

15.Please follow instructions given below.

<br>

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. <br>Hint:Use NO\_OF\_CUSTOMERS as alias name for number of customers.

ANS: SELECT count(customer\_id) NO\_OF\_CUSTOMERS FROM customer\_master WHERE extract(year from date\_of\_registration)=2012 and contact\_no is not null

16.Please follow instructions given below.

<br>

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.

ANS: SELECT c.customer\_name,coalesce(c.contact\_no,c.contact\_address) CONTACT\_DETAILS,cd.card\_id,cd.description FROM customer\_master c left join customer\_card\_detailsccd on ccd.customer\_id=c.customer\_id left join library\_card\_master cd on cd.card\_id=ccd.card\_id order by customer\_name;

17.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 decendingorder 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

ANS: SELECT cid.customer\_id, customer\_name,count(movie\_id) NO\_OF\_TIMES FROM customer\_master cm INNER JOIN customer\_issue\_detailscid ON cm.customer\_id=cid.customer\_id group by customer\_id, customer\_name,movie\_id having count(movie\_id)>1 order by customer\_namedesc;

18. 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.&nbsp; <br>Hint:Use NO\_OF\_MOVIES as alias name for number of movies column.

<br>Hint:Use CONTACT\_ISD as alias name for contact number.

<br> Display the records sorted in ascending order based on customer name and then by movie category.

ANS: SELECT cid.customer\_id,customer\_name,concat('+91-',substring(contact\_no,1,3),'-',substring(contact\_no,4,3),'-',substring(contact\_no,7)) CONTACT\_ISD,movie\_category,count(movie\_category) NO\_OF\_MOVIES FROM customer\_issue\_detailscid INNER JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id INNER JOIN customer\_master cm ON cm.customer\_id=cid.customer\_id group by customer\_id,customer\_name,CONTACT\_ISD,movie\_category having count(movie\_category)>1 order by customer\_name, movie\_category;

19. 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.

ANS: (selectcid.customer\_id , customer\_name FROM customer\_master cm INNER JOIN customer\_issue\_detailscid

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)

UNION ALL

(selectcid.customer\_id , customer\_name FROM customer\_master cm INNER JOIN customer\_issue\_detailscid

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) ;

20. 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.

ANS: SELECT cid.customer\_id,customer\_name,count(cid.movie\_id) NO\_OF\_TIMES FROM customer\_master cm INNER JOIN customer\_issue\_detailscid ON cm.customer\_id=cid.customer\_id INNER JOIN movies\_master mm

ON cid.movie\_id=mm.movie\_id WHERE movie\_category='Comedy' GROUP BY customer\_name HAVING count(cid.customer\_id)>1 order by customer\_name;

21. 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

ANS: SELECT cid.customer\_id, sum(rental\_cost) TOTAL\_COST FROM customer\_issue\_detailscid INNER JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id GROUP BY customer\_id order by customer\_id;

9. FLIGHT MANAGEMENT

1. 1.Display passenger name who has a 'a' as second letter in their name.

select passname from passenger where passname like '\_a%';

2. Display the name of the youngest passenger.

select passname from passenger

where passdob=(select max(passdob) from passenger);

3. . Display the name of the passenger, date of birth and age.

select passname,passdob,

round(datediff(current\_date,passdob)/365) as age from passenger;

4. Display the number of flights leaving Kolkata.

select count(\*) from flight where flightsource='kol';

5. Display the name of city where the number of flights leaving and reaching is the same.

select a.flightsource from flight a

group by a.flightsource having count(a.flightsource)=

(select count(b.flightdest) from flight b

where a.flightsource=b.flightdest);

6.

Display the name of the city which has flight source but no destination.

select flightsource from flight where flightsource not in(select flightdest from flight);

7. Display the dates on which flight 1 and 4 is flying.

select flightdate from flight where flightid in(1,4);

8. Display the number of passenger in each flight.

Use column alias “PassCount”.

select count(a.passid) as PassCount,flightid from booking\_details a,booking b where a.bookingid=b.bookingid group by b.flightid;

9. Display the name and date of birth of passengers who are senior citizen (age>=60).

select passname,passdob from passenger where round((datediff(current\_date,passdob)/365))>=60;

10. Display the booking id having the highest number of passengers.

select bookingid from booking\_details group by bookingid having count(passid)=

(select count(passid) from booking\_details group by bookingid order by count(passid) desc limit 1);

11. Display the booking id (ticket) and the total cost for the booking. Use column alias “Total Fare”.

select a.bookingid,sum(b.ticketcost) as Total\_Fare from booking\_details a,flight b,booking c

where c.flightid=b.flightid and a.bookingid=c.bookingid group by a.bookingid;

12. Display the booking id (ticket) and the total cost for the booking. Use column alias “Total Fare”.

Consider giving a rebate of 50% to senior citizen (age>=60).

select a.bookingid,sum(

case

when datediff(current\_date,d.passdob)/365>=60

then b.ticketcost\*0.5

else b.ticketcost end) as Total\_fare

from booking\_details a,flight b,booking c,passenger d

where a.bookingid=c.bookingid and b.flightid=c.flightid and a.passid=d.passid group by a.bookingid;

13. Display the city receiving the maximum number of flights.

select flightdest from flight group by flightdest having count(flightid)=(select count(flightid)

from flight group by flightdest

order by count(flightid) desc limit 1);

14. Display the passenger’s name having more than 1 booking.

select passname from passenger where passid in(select passid from booking\_details group by passid having count(bookingid)>1);

15. Display flightid with no of booking.

select a.flightid,count(b.passid) from booking a,booking\_details b where a.bookingid=b.bookingid group by a.flightid;

16. Display the passenger (name only) who booked ticket on the day of flight for flight no 1.

select passname from passenger where passid in(select passid from booking\_details where bookingid in(select bookingid from booking where bookdate in(select date(a.flightdate) from flight a,booking b where a.flightid=b.flightid and date(a.flightdate)=b.bookdate and a.flightid=1)));

17. Display flights having the same source and destination.

select a.flightid from flight a,flight b where a.flightid!=b.flightid and a.flightsource=b.flightsource and a.flightdest=b.flightdest;

18. Display the record in the following format.

Column alias “Booking Summary”

#Hints:“ Ticket No:1 Flight id: 1 Total Passengers :3 Total Fare:6000”

#“Ticket No:2 Flight id: 3 Total Passengers :1 Total Fare :2500”

select concat('Ticket No:',cast(b.bookingid as char),'Flight id:',cast(b.flightid as char),'Total Passengers:',cast(count(c.passid) as char),'Total Fare:',cast(sum(a.ticketcost) as char)) as Booking\_Summary from flight a,booking b,booking\_details c where a.flightid=b.flightid and b.bookingid=c.bookingid group by a.flightid;

19. Flight No: 2 have been delayed for 4 hrs due to fog. Display flight id , flight date and a new column “flight new date”, which displays the new timing.

select flightid,flightdate,flightdate + interval 4 hour as flight\_new\_date from flight where flightid=2;

20. Display passenger name , date of birth sorted by the month of birth .

(Jan ? Dec).

select passname,passdob from passenger order by month(passdob);