1.Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight\_Id,From\_location,To\_Location,Month Name as “Month\_Name” and average price as “Average\_Price”

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

15 rows

select f.flight\_id,f.from\_location,f.to\_location,monthname(fd.flight\_departure\_date) as

Month\_name,avg(fd.price) as Average\_price from air\_flight f join air\_flight\_details fd

on f.flight\_id=fd.flight\_id group by f.flight\_id,Month\_name order by f.flight\_id,Month\_name;

2.Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile\_id, customer’s first\_name, Address and Number of tickets booked as “No\_of\_Tickets”

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

1 row

select apf.profile\_id,apf.first\_name,apf.address,count(ati.ticket\_id) as No\_of\_Tickets

from air\_passenger\_profile apf

join air\_ticket\_info ati on apf.profile\_id=ati.profile\_id group by apf.profile\_id having

count(ati.ticket\_id) <=all

(select count(ati.ticket\_id) from air\_passenger\_profile apf

join air\_ticket\_info ati on apf.profile\_id=ati.profile\_id group by apf.profile\_id) order by

first\_name;

3.Write a query to display the number of flight services between locations in a month. The Query should display From\_Location, To\_Location, Month as “Month\_Name” and number of flight services as “No\_of\_Services”.

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

The records should be displayed in ascending order based on From\_Location and then by To\_Location and then by month name

9 rows

select af.from\_location,af.to\_location,monthname(afd.flight\_departure\_date) as Month\_Name,

count(afd.flight\_departure\_date) as No\_of\_Services from air\_flight af join air\_flight\_details afd

on af.flight\_id=afd.flight\_id group by af.from\_location,af.to\_location,month\_name order by

from\_location,to\_location,month\_name;

4.Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The Query should display profile\_id, customer’s first\_name, Address and Number of tickets booked as “No\_of\_Tickets”

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

1 row

select app.profile\_id,app.first\_name,app.address,count(ati.ticket\_id) as No\_of\_Tickets

from air\_passenger\_profile app

join air\_ticket\_info ati on app.profile\_id=ati.profile\_id join air\_flight af on ati.flight\_id=af.flight\_id

where af.airline\_name= ’ABC Airlines’ group by app.profile\_id

having count(ati.ticket\_id) >= all (select count(ati.ticket\_id) from air\_passenger\_profile app

join air\_ticket\_info ati on app.profile\_id=ati.profile\_id join air\_flight af on ati.flight\_id=af.flight\_id

where af.airline\_name= ’ABC Airlines’ group by app.profile\_id) order by app.first\_name;

5.Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile\_id,first\_name,last\_name, Flight\_Id , Departure\_Date and number of tickets booked as “No\_of\_Tickets”.

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

3 rows

select ati.profile\_id,app.first\_name,app.last\_name,ati.flight\_id,ati.flight\_departure\_date,count(ati.ticket\_id)

as No\_of\_Tickets from air\_ticket\_info ati join air\_passenger\_profile app on ati.profile\_id=

app.profile\_id join air\_flight af on ati.flight\_id=af.flight\_id

where af.from\_location='chennai' and af.to\_location='hyderabad' group by ati.profile\_id,

ati.flight\_id,ati.flight\_departure\_date order by

ati.profile\_id,

ati.flight\_id,ati.flight\_departure\_date;

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

3 rows

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

select af.flight\_id,af.from\_location,af.to\_location,afd.price from air\_flight af

join air\_flight\_details afd on af.flight\_id=afd.flight\_id

where monthname(afd.flight\_departure\_date)='april' order by flight\_id,from\_location;

7.Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight\_id, from\_location, to\_location and Average price as “Price”.

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

11 rows

select af.flight\_id,af.from\_location,af.to\_location,avg(afd.price)

from air\_flight af join air\_flight\_details afd

on af.flight\_id=afd.flight\_id group by af.flight\_id,af.from\_location,af.to\_location

order by af.flight\_id,af.from\_location,af.to\_location;

8.Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile\_id, customer\_name (combine first\_name & last\_name with comma in b/w), address of the customer.

Give an alias to the name as customer\_name.

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

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

3 rows

select app.profile\_id, concat(app.first\_name,',',app.last\_name) as customer\_name,app.address

from air\_passenger\_profile app join air\_ticket\_info ati on app.profile\_id=ati.profile\_id

join air\_flight af on ati.flight\_id=af.flight\_id where af.from\_location='chennai'

and af.to\_location='hyderabad' group by app.profile\_id order by app.profile\_id;

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

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

2 rows

select profile\_id from air\_ticket\_info group by profile\_id having count(ticket\_id) >= all (select count(ticket\_id)

from air\_ticket\_info group by profile\_id) order by profile\_id;

10.Write a query to display the total number of tickets as “No\_of\_Tickets” booked in each flight in ABC Airlines. The Query should display the flight\_id, from\_location, to\_location and the number of tickets.

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

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

7 rows

select af.flight\_id,af.from\_location,af.to\_location,count(ati.ticket\_id) as No\_of\_Tickets

from air\_flight af join air\_ticket\_info ati on af.flight\_id=ati.flight\_id

group by af.flight\_id having count(ati.ticket\_id) >= 1;

11.Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight\_id, number of services as “No\_of\_Services” and the cost as “Total\_Price” in the same order.

Order the result by Total Price in descending order and then by flight\_id in descending order.

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

11 rows

select af.flight\_id, count(afd.flight\_departure\_date) as No\_of\_Services, sum(afd.price) as

Total\_Price from air\_flight af join air\_flight\_details afd on af.flight\_id=afd.flight\_id

group by flight\_id

order by total\_price desc,flight\_id desc;

12.Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight\_id, flight\_departure\_date and the number of passengers as “No\_of\_Passengers” in the same order.

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

9 rows

SELECT flight\_id,

flight\_departure\_date,

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

FROM air\_ticket\_info

GROUP BY flight\_id,

flight\_departure\_date

ORDER BY flight\_id, flight\_departure\_date;t

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

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

1 row

select profile\_id from air\_ticket\_info group by profile\_id having count(profile\_id) <= all

(select count(profile\_id) from air\_ticket\_info group by profile\_id) order by profile\_id;

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

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

4 rows

select distinct ati.profile\_id,app.first\_name,app.mobile\_number,app.email\_id

from air\_ticket\_info

ati join air\_passenger\_profile app on ati.profile\_id=app.profile\_id join air\_flight af

on ati.flight\_id=af.flight\_id

where af.from\_location='hyderabad' and af.to\_location='chennai' order by profile\_id;

15.Write a query to intimate the passengers who are boarding Chennai to Hyderabad Flight on 6th May 2013 stating the delay of 1hr in the departure time. The Query should display the passenger’s profile\_id, first\_name,last\_name, flight\_id, flight\_departure\_date, actual departure time , actual arrival time , delayed departure time as "Delayed\_Departure\_Time", delayed arrival time as "Delayed\_Arrival\_Time" Hint: Distinct Profile ID should be displayed irrespective of multiple tickets booked by the same profile.

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

1 row

select distinct app.profile\_id,app.first\_name,app.last\_name,ati.flight\_id,ati.flight\_departure\_date,

af.departure\_time,af.arrival\_time, af.departure\_time ,ADDTIME(af.departure\_time,'1:00:00') as Delayed\_Departure\_Time,

ADDTIME(af.arrival\_time,'1:00:00') as Delayed\_Arrival\_Time from air\_passenger\_profile app

join air\_ticket\_info ati on app.profile\_id=ati.profile\_id join air\_flight af on

ati.flight\_id=af.flight\_id where ati.flight\_departure\_date='2013-05-06' order by app.profile\_id;

16.Write a query to display the number of tickets as “No\_of\_Tickets” booked by Kochi Customers. The Query should display the Profile\_Id, First\_Name, Base\_Location and number of tickets booked.

Hint: Use String functions to get the base location of customer from their Address and give alias name as “Base\_Location”

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

2 rows

select ap.profile\_id,ap.first\_name,substring\_index(substring\_index(ap.address,',',-1),'-',1)

as base\_location,count(at.ticket\_id) as No\_of\_Tickets from air\_passenger\_profile ap join air\_ticket\_info at

on at.profile\_id=ap.profile\_id

where substring\_index(substring\_index(ap.address,',',-1),'-',1) ='kochi'

group by ap.profile\_id order by first\_name;

17.Write a query to display the flight\_id, from\_location, to\_location, number of Services as “No\_of\_Services” offered in the month of May.

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

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

11 rows

select af.flight\_id,af.from\_location,af.to\_location,count(afd.flight\_departure\_date)

as No\_of\_Services from air\_flight af join air\_flight\_details afd

on af.flight\_id=afd.flight\_id where month(afd.flight\_departure\_date)='05'

group by flight\_id order by flight\_id;

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

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

2 rows

select profile\_id,last\_name,mobile\_number,email\_id from air\_passenger\_profile where

substring\_index(substring\_index(address,',',-1),'-',1)='chennai'

order by profile\_id;

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

1 row

select count(flight\_id) as FLIGHT\_COUNT from air\_flight where departure\_time between

'6:00:00' and '18:00:00' and from\_location='chennai';

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

2 rows

select distinct app.profile\_id,app.first\_name,app.email\_id,app.mobile\_number from air\_passenger\_profile app

join air\_ticket\_info ati on app.profile\_id=ati.profile\_id

where ati.flight\_id= 3148 group by app.first\_name order by app.first\_name;

20.Write a query to display the flights available in Morning, AfterNoon, Evening & Night. The Query should display the Flight\_Id, From\_Location, To\_Location , Departure\_Time, time of service as "Time\_of\_Service".

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

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

11 rows

select flight\_id,from\_location,to\_location,departure\_time,

case when departure\_time between '05:00:01' and '12:00:00' then 'Morning'

when departure\_time between '12:00:01' and '18:00:00' then 'Afternoon'

when departure\_time between '18:00:01' and '24:00:00' then 'Evening'

when departure\_time between '00:00:01' and '05:00:00' then 'Night'

end as Time\_of\_Service

from air\_flight order by flight\_id;

21.Please follow instructions given below.

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

Display the records sorted in ascendeing order based on flight\_id and then by departure date.

36 rows

select flight\_id,flight\_departure\_date,

case when price<3000 then 'AIR PASSENGER'

when price>=3000 and price<=4000 then 'AIR BUS'

when price>4000 then 'EXECUTIVE PASSENGER'

end as FLIGHT\_TYPE from air\_flight\_details order by flight\_id,flight\_departure\_date;

22.Please follow instructions given below.

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

Hint: Use CARD\_COUNT AS Alias name for no of cards.

3 rows

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

23.Please follow instructions given below.

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

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

Hint: Use SERIAL\_NO as Alias name for serial number.

Display the records sorted in ascending order based on name.

6 rows

select substring(profile\_id,4) as SERIAL\_NO,first\_name,mobile\_number,email\_id

from air\_passenger\_profile where email\_id like '%gmail.com' order by first\_name;

24.Please follow instructions given below.

Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight\_id, from\_location, to\_location, least number of Services as “No\_of\_Services” Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight

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

4 rows

select af.flight\_id,af.from\_location,af.to\_location,count(afd.flight\_departure\_date) as

No\_of\_Services from air\_flight af join air\_flight\_details afd on

af.flight\_id=afd.flight\_id where month(afd.flight\_departure\_date)='05' group by af.flight\_id

having count(afd.flight\_departure\_date)

<= all (select count(afd.flight\_departure\_date) from air\_flight af join air\_flight\_details afd on

af.flight\_id=afd.flight\_id where month(afd.flight\_departure\_date)='05' group by af.flight\_id)

order by af.flight\_id;

25.Please follow instructions given below.

Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No\_of\_Flights”.

Hint: Get the distinct from location and to location.

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

4 rows

select distinct from\_location,count(to\_location) as No\_of\_Flights from air\_flight

group by from\_location order by from\_location;

26.Please follow instructions given below.

Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight\_id,from\_location,To\_location, flight\_departure\_date and the number of passengers as “No\_of\_Passengers”.

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

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

9 rows

select af.flight\_id,af.from\_location,af.to\_location,ati.flight\_departure\_date,count(ati.ticket\_id)

as No\_of\_Passengers from air\_flight af join air\_ticket\_info ati on af.flight\_id=ati.flight\_id

group by af.flight\_id,ati.flight\_departure\_date order by af.flight\_id,ati.flight\_departure\_date;

27.Please follow instructions given below.

Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight\_Id, From\_Location, To\_Location,Total\_Seats, seats booked as “No\_of\_Seats\_Booked” .

Display the records sorted in ascending order based on flight id and then by No\_of\_Seats\_Booked.

1 row

select af.flight\_id,af.from\_location,af.to\_location,af.total\_seats,(af.total\_seats-afd.available\_seats)

as No\_of\_Seats\_Booked from air\_flight af join air\_flight\_details afd on af.flight\_id=

afd.flight\_id where (af.total\_seats-afd.available\_seats)>(af.total\_seats\*0.1) group by flight\_id order by

flight\_id,No\_of\_Seats\_Booked;

28.Please follow instructions given below.

Write a query to display the Flight\_Id, Flight\_Departure\_Date, From\_Location,To\_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.

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

14 rows

select af.flight\_id,afd.flight\_departure\_date,af.from\_location,af.to\_location,af.duration

from air\_flight af join air\_flight\_details afd on af.flight\_id=afd.flight\_id

where duration<'1:10:00' group by af.flight\_id,afd.flight\_departure\_date

order by af.flight\_id,afd.flight\_departure\_date;

29.Please follow instructions given below.

Write a query to display the flight\_id, from\_location,to\_location,number of services as “No\_of\_Services” , average ticket price as “Average\_Price” whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

4 rows

select af.flight\_id,af.from\_location,af.to\_location,count(afd.flight\_departure\_date) as No\_of\_Services,

avg(afd.price) as Average\_Price from air\_flight af join air\_flight\_details afd

on af.flight\_id=afd.flight\_id group by af.flight\_id having avg(afd.price)>

(select avg(afd.price) from air\_flight\_details afd) order by afd.price;

……………………………………………..

**BANK MANGEMENT SYSTEM**

**QUESTIONS**

|  |
| --- |
| 1. Write a query which will display the customer id, account type they hold, their account number and bank name. |
| 1. Write a query which will display the customer id, account type and the account number of HDFC customers who registered after 12-JAN-2012 and before 04-APR-2012. |
| 1. Write a query which will display the customer id, customer name, account no, account type and bank name where the customers hold the account. |
| 1. Write a query which will display the customer id, customer name, gender, marital status along with the unique reference string and sort the records based on customer id in descending order.  <b>Hint: Generate unique reference string as mentioned below:  CustomerName\_Gender\_MaritalStatus <b> Example,   C-005 KUMAR M SINGLE KUMAR\_M\_SINGLE  Use "UNIQUE\_REF\_STRING" as alias name for displaying the unique reference string. |
| 1. Write a query which will display the account number, customer id, registration date, initial deposit amount of the customer whose initial deposit amount is within the range of Rs.15000 to Rs.25000. |
| 1. Write a query which will display customer id, customer name, date of birth, guardian name of the customers whose name starts with 'J'. |
| 1. Write a query which will display customer id, account number and passcode.  Hint: To generate passcode, join the last three digits of customer id and last four digit of account number.  Example C-001 1234567898765432 0015432 Use "PASSCODE" as alias name for displaying the passcode. |
| 1. Write a query which will display the customer id, customer name, date of birth, Marital Status, Gender, Guardian name, contact no and email id of the customers whose gender is male 'M' and marital status is MARRIED. |
| 1. Write a query which will display the customer id, customer name, guardian name, reference account holders name of the customers who are referenced / referred by their 'FRIEND'. |
| 1. Write a query to display the customer id, account number and interest amount in the below format with INTEREST\_AMT as alias name. Sort the result based on the INTEREST\_AMT in ascending order. Example: $5Hint: Need to prefix $ to interest amount and round the result without decimals. |
| 1. Write a query which will display the customer id, customer name, account no, account type, activation date, bank name whose account will be activated on '10-APR-2012' |
| 1. Write a query which will display account number, customer id, customer name, bank name, branch name, ifsc code, citizenship, interest and initial deposit amount of all the customers. |
| 1. Write a query which will display customer id, customer name, date of birth, guardian name, contact number, mail id and reference account holder's name of the customers who has submitted the passport as an identification document. |
| 1. Write a query to display the customer id, customer name, account number, account type, initial deposit, interest who have deposited maximum amount in the bank. |
| 1. Write a query to display the customer id, customer name, account number, account type, interest, bank name and initial deposit amount of the customers who are getting maximum interest rate. |
| 1. Write a query to display the customer id, customer name, account no, bank name, contact no and mail id of the customers who are from BANGALORE. |
| 1. Write a query which will display customer id, bank name, branch name, ifsc code, registration date, activation date of the customers whose activation date is in the month of march (March 1'st to March 31'st). |
| 1. Write a query which will calculate the interest amount and display it along with customer id, customer name, account number, account type, interest, and initial deposit amount.Hint :Formula for interest amount, calculate: ((interest/100) \* initial deposit amt) with column name 'interest\_amt' (alias) |
| 1. Write a query to display the customer id, customer name, date of birth, guardian name, contact number, mail id, reference name who has been referenced by 'RAGHUL'. |
| 1. Write a query which will display the customer id, customer name and contact number with ISD code of all customers in below mentioned format. Sort the result based on the customer id in descending order. Format for contact number is :   "+91-3digits-3digits-4digits"  Example: +91-924-234-2312  Use "CONTACT\_ISD" as alias name. |
| 1. Write a query which will display account number, account type, customer id, customer name, date of birth, guardian name, contact no, mail id , gender, reference account holders name, reference account holders account number, registration date, activation date, number of days between the registration date and activation date with alias name "NoofdaysforActivation", bank name, branch name and initial deposit for all the customers. |
| 1. Write a query which will display customer id, customer name, guardian name, identification doc type, reference account holders name, account type, ifsc code, bank name and current balance for the customers who has only the savings account.  Hint: Formula for calculating current balance is add the intital deposit amount and interest and display without any decimals. Use "CURRENT\_BALANCE" as alias name. |
| 1. Write a query which will display the customer id, customer name, account number, account type, interest, initial deposit; check the initial deposit, if initial deposit is 20000 then display "high", if initial deposit is 16000 display 'moderate', if initial deposit is 10000 display 'average', if initial deposit is 5000 display 'low', if initial deposit is 0 display 'very low' otherwise display 'invalid' and sort by interest in descending order. Hint: Name the column as "Deposit\_Status" (alias). Strictly follow the lower case for strings in this query. |
| 1. Write a query which will display customer id, customer name, account number, account type, bank name, ifsc code, initial deposit amount and new interest amount for the customers whose name starts with "J".   Hint: Formula for calculating "new interest amount" is if customers account type is savings then add 10 % on current interest amount to interest amount else display the current interest amount. Round the new interest amount to 2 decimals. Use "NEW\_INTEREST" as alias name for displaying the new interest amount. Example, Assume Jack has savings account and his current interest amount is 10.00, then the new interest amount is 11.00 i.e (10 + (10 \* 10/100)). |
| 1. Write query to display the customer id, customer name, account no, initial deposit, tax percentage as calculated below.Hint: If initial deposit = 0 then tax is '0%'If initial deposit &lt;= 10000 then tax is '3%' If initial deposit &gt; 10000 and initial deposit &lt; 20000 then tax is '5%' If initial deposit &gt;= 20000 and initial deposit&lt;=30000 then tax is '7%' If initial deposit &gt; 30000 then tax is '10%' Use the alias name 'taxPercentage' |

**ANSWERS**

1. select a.Customer\_ID, a.account\_type, a.account\_no, b.bank\_name from account\_info a

join bank\_info b on(a.ifsc\_code = b.ifsc\_code);

1. select Customer\_ID, account\_type, account\_no from account\_info a

join bank\_info b on(a.ifsc\_code = b.ifsc\_code) where b.bank\_name = 'HDFC'

and registration\_date between '2012-01-12' and '2012-04-04';

1. select a.Customer\_ID, c.Customer\_Name, a.account\_no, a.account\_type, b.bank\_name from account\_info a

join bank\_info b on(a.ifsc\_code = b.ifsc\_code)

join customer\_personal\_info c on(a.customer\_id = c.customer\_id);

1. select Customer\_ID, Customer\_Name, gender, marital\_status, concat(customer\_name,'\_',gender,'\_',marital\_status)

UNIQUE\_REF\_STRING from customer\_personal\_info;

1. select account\_no, customer\_id, registration\_date, initial\_deposit from account\_info

where initial\_deposit between 15000 and 25000;

1. select Customer\_ID, Customer\_Name, date\_of\_birth, guardian\_name from customer\_personal\_info where guardian\_name like 'j%';
2. select Customer\_ID, account\_no, concat(substr(customer\_id,3,5),substr(account\_no,14,16)) passcode from account\_info;
3. select Customer\_ID, Customer\_Name, date\_of\_birth, marital\_status, gender, guardian\_name, contact\_no, mail\_id from customer\_personal\_info where gender = 'm' and marital\_status = 'married';
4. select c.customer\_id, c.Customer\_Name, c.guardian\_name, r.reference\_acc\_name from customer\_personal\_info c join customer\_reference\_info r on(c.customer\_id = r.customer\_id) where relation = 'friend';
5. select customer\_id, account\_no, concat('$',round(interest)) INTEREST\_AMT from account\_info group by interest;
6. select i.Customer\_ID, i.Customer\_Name, a.account\_no, a.account\_type, a.activation\_date, b.bank\_name

from customer\_personal\_info i join account\_info a on(i.customer\_id = a.customer\_id)

join bank\_info b on(a.ifsc\_code = b.ifsc\_code)

where a.activation\_date = '2012-04-10';

1. select a.account\_no, a.customer\_id, c.customer\_name, b.bank\_name, b.branch\_name, b.ifsc\_code, c.citizenship, a.interest, a.initial\_deposit from account\_info a

join bank\_info b on(a.ifsc\_code = b.ifsc\_code)

join customer\_personal\_info c on(a.customer\_id = c.customer\_id);

1. select a.customer\_id, a.customer\_name, a.date\_of\_birth, a.guardian\_name, a.contact\_no, a.mail\_id, b.reference\_acc\_name from customer\_personal\_info a join customer\_reference\_info b on(a.customer\_id = b.customer\_id) where a.identification\_doc\_type = 'passport';
2. select b.customer\_id, a.customer\_name, b.account\_no, b.account\_type, b.initial\_deposit, b.interest from customer\_personal\_info a join account\_info b on(a.customer\_id = b.customer\_id) where b.initial\_deposit = (select max(initial\_deposit) from account\_info);
3. select b.customer\_id, a.customer\_name, b.account\_no, b.account\_type, b.interest,c.bank\_name,b.initial\_deposit from account\_info b join customer\_personal\_info a on(a.customer\_id = b.customer\_id) join bank\_info c on(b.ifsc\_code = c.ifsc\_code) where b.interest = (select max(interest) from account\_info);
4. select a.Customer\_ID, a.customer\_name, b.account\_no, c.bank\_name, a.contact\_no, a.mail\_id from customer\_personal\_info a join account\_info b on(a.customer\_id = b.customer\_id) join bank\_info c on(b.ifsc\_code = c.ifsc\_code) where a.address like '%bangalore';
5. select b.customer\_id, a.bank\_name, a.branch\_name, a.ifsc\_code, b.registration\_date, b.activation\_date from bank\_info a join account\_info b on(a.ifsc\_code = b.ifsc\_code)

where b.activation\_date like '%-03-%';

1. select a.customer\_id, a.customer\_name, b.account\_no, b.account\_type, b.interest, b.initial\_deposit, ((b.interest/100)\*b.initial\_deposit) interest\_amt from customer\_personal\_info a join account\_info b on(a.customer\_id = b.customer\_id);
2. select a.customer\_id, a.customer\_name, a.date\_of\_birth, a.guardian\_name, a.contact\_no, a.mail\_id, b.reference\_acc\_name

from customer\_personal\_info a join customer\_reference\_info b on(a.customer\_id = b.customer\_id) where reference\_acc\_name = 'raghul';

1. select Customer\_ID, Customer\_Name, concat('+91-',substr(contact\_no,1,3),'-',substr(contact\_no,4,3),'-',substr(contact\_no,7,4)) CONTACT\_ISD from customer\_personal\_info;
2. select a.ACCOUNT\_NO, a.ACCOUNT\_TYPE, a.CUSTOMER\_ID, b.CUSTOMER\_NAME, b.DATE\_OF\_BIRTH, b.GUARDIAN\_NAME, b.CONTACT\_NO, b.MAIL\_ID, b.GENDER, c.REFERENCE\_ACC\_NAME, c.REFERENCE\_ACC\_NO, a.REGISTRATION\_DATE, a.ACTIVATION\_DATE, d.BANK\_NAME, d.BRANCH\_NAME, a.INITIAL\_DEPOSIT, (a.ACTIVATION\_DATE-a.REGISTRATION\_DATE) NoOfDaysForActivation from account\_info a join customer\_personal\_info b on(a.customer\_id = b.customer\_id)

join bank\_info d on(a.ifsc\_code = d.ifsc\_code) join customer\_reference\_info c on(b.customer\_id = c.customer\_id);

1. select a.CUSTOMER\_ID, a.CUSTOMER\_NAME, a.GUARDIAN\_NAME, a.IDENTIFICATION\_DOC\_TYPE, b.REFERENCE\_ACC\_NAME, c.ACCOUNT\_TYPE, c.IFSC\_CODE, d.BANK\_NAME, round(c.initial\_deposit+((c.interest/100)\*c.initial\_deposit)) current\_balance from customer\_personal\_info a join customer\_reference\_info b on(b.customer\_id = a.customer\_id) join account\_info c on(a.customer\_id = c.customer\_id)

join bank\_info d on(c.ifsc\_code = d.ifsc\_code);

1. select a.CUSTOMER\_ID, b.CUSTOMER\_NAME, a.ACCOUNT\_NO, a.ACCOUNT\_TYPE, a.INTEREST, CASE WHEN INITIAL\_DEPOSIT = 20000 then 'high'

WHEN INITIAL\_DEPOSIT = 16000 then 'moderate' WHEN INITIAL\_DEPOSIT = 10000 THEN 'average’ when INITIAL\_DEPOSIT = 5000 then 'low' when initial\_deposit = 0 then 'very low' END as Deposit\_Status from account\_info a

join customer\_personal\_info b on(a.customer\_id = b.customer\_id);

1. select a.CUSTOMER\_ID, b.CUSTOMER\_NAME, a.ACCOUNT\_NO, a.ACCOUNT\_TYPE, c.BANK\_NAME, c.IFSC\_CODE, a.INITIAL\_DEPOSIT, if(ACCOUNT\_TYPE = 'savings', round(a.interest+(a.interest\*(a.interest/100)),2), a.interest) as NEW\_INTEREST

from account\_info a join customer\_personal\_info b on(a.customer\_id = b.customer\_id)

join bank\_info c on(a.ifsc\_code = c.ifsc\_code)

where CUSTOMER\_NAME like 'j%';

1. select a.CUSTOMER\_ID, b.customer\_name, a.account\_no, a.INITIAL\_DEPOSIT,

case

when a.INITIAL\_DEPOSIT = 0 then '0%'

when a.INITIAL\_DEPOSIT <= 10000 then '3%'

when a.INITIAL\_DEPOSIT > 10000 && a.INITIAL\_DEPOSIT <= 20000 then '5%'

when a.INITIAL\_DEPOSIT > 20000 && a.INITIAL\_DEPOSIT <= 30000 then '7%'

when a.INITIAL\_DEPOSIT > 30000 then '10%'

END as taxPercentage from account\_info a

join customer\_personal\_info b on(a.customer\_id = b.customer\_id);

……………………………………………………………

**Bank Management System Queries:**

**1.Please follow instructions given below.**

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

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

**SELECT account\_number,am.customer\_number,firstname,lastname,account\_opening\_date**

**FROM customer\_master cm INNER JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**ORDER BY account\_number;**

**2.Please follow instructions given below.**

**Write a query to display the number of customer’s from Delhi. Give the count an alias name of Cust\_Count.**

**SELECT count(customer\_number) Cust\_Count**

**FROM customer\_master**

**WHERE customer\_city='Delhi'**

**3.Please follow instructions given below.**

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

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

**SELECT am.customer\_number, firstname, account\_number**

**FROM customer\_master cm INNER JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**WHERE extract(day from account\_opening\_date)>15**

**ORDER BY am.customer\_number, account\_number**

**4.Please follow instructions given below.**

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

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

**SELECT am.customer\_number,firstname, account\_number**

**FROM customer\_master cm INNER JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**WHERE account\_status='Terminated'**

**ORDER BY am.customer\_number, account\_number**

**5.Please follow instructions given below.**

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

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

**SELECT transaction\_type,count(transaction\_number) Trans\_Count**

**FROM account\_master am INNER JOIN transaction\_details td**

**ON am.account\_number=td.account\_number**

**WHERE customer\_number like '%001'**

**GROUP BY transaction\_type**

**ORDER BY transaction\_type**

**6.Please follow instructions given below.**

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

**Give the alias name as Count\_Customer for number of customers.**

**SELECT count(customer\_number) Count\_Customer**

**FROM customer\_master**

**WHERE customer\_number NOT IN (SELECT customer\_number FROM account\_master)**

**7.Please follow instructions given below.**

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

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

**FROM account\_master am INNER JOIN transaction\_details td**

**ON am.account\_number=td.account\_number**

**WHERE transaction\_type='deposit'**

**GROUP BY account\_number**

**ORDER BY account\_number**

**8.Please follow instructions given below.**

**Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No\_of\_Accounts.For the branch city where we don’t have any accounts opened display 0. Display the records in sorted order based on branch city.**

**select branch\_master.branch\_city, count(account\_master.account\_number) as No\_of\_Accounts from branch\_master left join account\_master on account\_master.branch\_id=branch\_master.branch\_id**

**group by branch\_master.branch\_city order by branch\_city;**

**9.Please follow instructions given below.**

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

**select firstname**

**FROM customer\_master cm INNER JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**group by firstname**

**having count(account\_number)>1**

**order by firstname;**

**10.Please follow instructions given below.**

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

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

**SELECT ld.customer\_number, firstname, lastname**

**FROM customer\_master cm INNER JOIN loan\_details ld**

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

**GROUP BY customer\_number**

**HAVING count(branch\_id)>1**

**ORDER BY customer\_number**

**11.Please follow instructions given below.**

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

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

**select customer\_master.customer\_number, firstname, customer\_city, branch\_city**

**from account\_master inner join customer\_master on account\_master.customer\_number = customer\_master.customer\_number**

**inner join branch\_master on account\_master.branch\_id = branch\_master.branch\_id**

**where customer\_city != branch\_city order by customer\_master.customer\_number;**

**12.Please follow instructions given below.**

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

**SELECT count(ld.customer\_number) Count**

**FROM customer\_master cm INNER JOIN loan\_details ld**

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

**WHERE cm.customer\_number NOT IN ( SELECT customer\_number FROM account\_master)**

**13.Please follow instructions given below.**

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

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

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

**SELECT td.account\_number**

**FROM account\_master am INNER JOIN transaction\_details td**

**ON am.account\_number=td.account\_number**

**group by td.account\_number**

**having count(td.transaction\_number)>=ALL**

**(SELECT count(td.transaction\_number)**

**FROM account\_master am INNER JOIN transaction\_details td**

**ON am.account\_number=td.account\_number**

**group by td.account\_number) order by am.account\_number;**

**14.Please follow instructions given below.**

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

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

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

**select branch\_name,branch\_city**

**FROM branch\_master INNER JOIN account\_master**

**ON branch\_master.branch\_id=account\_master.branch\_id**

**group by branch\_name**

**having count(customer\_number)>=ALL**

**(select count(customer\_number)**

**FROM branch\_master INNER JOIN account\_master**

**ON branch\_master.branch\_id=account\_master.branch\_id**

**group by branch\_name) order by branch\_name;**

**15.Please follow instructions given below.**

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

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

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

**SELECT td.account\_number,sum(CASE WHEN transaction\_type='Deposit' THEN transaction\_amount END)**

**+(SELECT opening\_balance FROM account\_master am2 where am2.account\_number=am.account\_number) Deposit,**

**sum(CASE WHEN transaction\_type='Withdrawal' THEN transaction\_amount END) Withdrawal**

**FROM account\_master am INNER JOIN transaction\_details td**

**ON am.account\_number=td.account\_number**

**GROUP BY td.account\_number**

**HAVING Withdrawal > Deposit**

**ORDER BY am.account\_number**

**16.Please follow instructions given below.**

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

**Note: Balance amount includes account opening balance also. Give alias name as Balance\_Amount.**

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

**SELECT (SUM(CASE WHEN transaction\_type='Deposit'**

**THEN transaction\_amount END)) -**

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

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

**from account\_master where account\_number like '%001') AS Balance\_Amount**

**FROM transaction\_details where account\_number like '%001'**

**17.Please follow instructions given below.**

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

**SELECT cm. customer\_number,firstname, am.account\_number,count(transaction\_number) Count\_Trans**

**FROM customer\_master cm inner JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**INNER JOIN transaction\_details td**

**ON am.account\_number=td.account\_number**

**group by am.account\_number order by cm.customer\_number, am.account\_number**

**18.Please follow instructions given below.**

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

**SELECT firstname**

**FROM customer\_master INNER JOIN account\_master**

**ON customer\_master.customer\_number=account\_master.customer\_number**

**GROUP BY firstname**

**having count(firstname)>=2 order by firstname;**

**19.Please follow instructions given below.**

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

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

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

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

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

**SELECT ld.customer\_number, firstname, lastname**

**FROM customer\_master cm INNER JOIN loan\_details ld**

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

**group by customer\_number**

**having count(branch\_id)>=2 and sum(loan\_amount)>=All(select sum(loan\_amount) from loan\_details group by customer\_number)**

**20.Please follow instructions given below.**

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

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

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

**FROM customer\_master cm INNER JOIN loan\_details ld**

**ON cm.customer\_number=ld.customer\_number order by cm.customer\_number, branch\_id, loan\_amount**

**21.Please follow instructions given below.**

**Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count\_Branch.**

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

**SELECT branch\_city, count(branch\_id) Count\_Branch**

**FROM branch\_master**

**GROUP BY branch\_city**

**ORDER BY branch\_city**

**22.Please follow instructions given below.**

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

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

**SELECT account\_number, firstname, lastname**

**FROM customer\_master cm INNER JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**WHERE account\_status='Active'**

**ORDER BY account\_number**

**23.Please follow instructions given below.**

**Write a query to display customer’s number, first name and middle name. For the customers who don’t have middle name, display their last name as middle name. Give the alias name as Middle\_Name.**

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

**SELECT customer\_number,firstname,coalesce(middlename,lastname) Middle\_Name**

**FROM customer\_master order by customer\_number**

**24.Please follow instructions given below.**

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

**SELECT customer\_number,firstname,customer\_date\_of\_birth**

**FROM customer\_master order by year(customer\_date\_of\_birth), firstname;**

**25.Please follow instructions given below.**

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

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

**SELECT firstname, customer\_city,account\_number**

**FROM customer\_master cm INNER JOIN account\_master am**

**ON cm.customer\_number=am.customer\_number**

**WHERE occupation !='Service' and occupation != 'Student' and occupation != 'Business' order by firstname, account\_number**

………………………………………….

#1) Write a query which will display the customer id, account type they hold, their account number and bank name.

select p.customer\_id,a.account\_type,a.account\_no,b.bank\_name from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code

order by p.customer\_id;

# 2) Write a query which will display the customer id, account type and the account number of HDFC customers who registered

#after 12-JAN-2012 and before 04-APR-2012.

select p.customer\_id,a.account\_type,a.account\_no from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code where b.bank\_name like 'HDFC' and

a.registration\_date>'2012-01-12' and a.registration\_date<'2012-04-04'

order by p.customer\_id;

# 3) Write a query which will display the customer id, customer name, account no, account type and bank name where the customers

#hold the account.

select p.customer\_id,p.customer\_name,a.account\_type,a.account\_no,b.bank\_name from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code

order by p.customer\_id;

# 4)Write a query which will display the customer id, customer name, gender, marital status along with the unique reference string

#and sort the records based on customer id in descending order. Hint:Generate unique reference string as mentioned below.

#CustomerName\_Gender\_MaritalStatus. Use ""UNIQUE\_REF\_STRING"" as alias name for displaying the unique reference string."

select customer\_id, customer\_name, gender, marital\_status,concat('customer\_name','\_','Gender','\_','MaritalStatus') as UNIQUE\_REF\_STRING

from customer\_personal\_info

order by customer\_id desc;

# 5) Write a query which will display the account number, customer id, registration date, initial deposit amount of the customer

#whose initial deposit amount is within the range of Rs.15000 to Rs.25000.

select a.account\_no,p.customer\_id,a.registration\_date,a.initial\_deposit from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id

where a.initial\_deposit between 15000 and 25000

order by p.customer\_id;

#6)Write a query which will display customer id,customer name,date of birth,guardian name of the customers whose name starts with 'J'.

select p.customer\_id,p.customer\_name,p.date\_of\_birth,p.guardian\_name from customer\_personal\_info p

where p.customer\_name like 'j%';

#7) Write a query which will display customer id, account number and passcode. Hint: To generate passcode, join the last three

#digits of customer id and last four digit of account number.

#Example: C-001 1234567898765432 0015432. Use ""PASSCODE"" as alias name for displaying the passcode."

select p.customer\_id,a.account\_no,concat(substr(p.customer\_id,3),substr(a.account\_no,13)) as passcode from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id

order by p.customer\_id;

#8) Write a query which will display the customer id, customer name, date of birth, Marital Status, Gender, Guardian name,

#contact no and email id of the customers whose gender is male 'M' and marital status is MARRIED.

select customer\_id, customer\_name, date\_of\_birth, Marital\_Status, Gender, Guardian\_name,contact\_no,mail\_id

from customer\_personal\_info where gender like 'M' and marital\_status like 'MARRIED'

order by customer\_id;

#9) Write a query which will display the customer id, customer name, guardian name, reference account holders name of the

#customers who are referenced / referred by their 'FRIEND'.

select p.customer\_id,p.customer\_name,p.guardian\_name,r.reference\_acc\_name from

customer\_personal\_info p join customer\_reference\_info r on

p.customer\_id=r.customer\_id where r.relation like 'friend';

#10) Write a query to display the customer id, account number and interest amount in the below format with INTEREST\_AMT as alias

#name.Sort the result based on the INTEREST\_AMT in ascending order.

#Hint: Need to prefix $ to interest amount and round the result without decimals.

select customer\_id,account\_no,concat('$',cast(round(interest) as char)) as INTEREST\_AMT from account\_info order by INTEREST\_AMT;

#11) Write a query which will display the customer id, customer name, account no, account type, activation date,bank name whose

#account will be activated on '10-APR-2012'

select p.customer\_id,p.customer\_name,a.account\_type,a.account\_no,a.activation\_date,b.bank\_name from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code where a.activation\_date like '2012-04-10';

#12) Write a query which will display account number, customer id, customer name, bank name, branch name, ifsc code,citizenship,

#interest and initial deposit amount of all the customers.

select a.account\_no,p.customer\_id,p.customer\_name,b.bank\_name,a.ifsc\_code,p.citizenship,a.interest,a.initial\_deposit from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code;

#13) Write a query which will display customer id, customer name, date of birth, guardian name, contact number,mail id and

#reference account holder's name of the customers who has submitted the passport as an identification document.

select p.customer\_id, p.customer\_name, p.date\_of\_birth,p.Guardian\_name,p.contact\_no,p.mail\_id,r.reference\_acc\_name from

customer\_personal\_info p join customer\_reference\_info r on

p.customer\_id=r.customer\_id where

p.identification\_doc\_type like 'passport';

#14) Write a query to display the customer id, customer name, account number, account type, initial deposit,interest who have

#deposited maximum amount in the bank.

select p.customer\_id,p.customer\_name,a.account\_type,a.initial\_deposit,a.interest from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id group by p.customer\_id having

max(a.initial\_deposit)>=(select max(initial\_deposit) from account\_info);

#15) Write a query to display the customer id, customer name, account number, account type, interest, bank name and initial deposit

#amount of the customers who are getting maximum interest rate

select p.customer\_id,p.customer\_name,a.account\_type,a.account\_no,a.initial\_deposit,a.interest,b.bank\_name from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code

group by p.customer\_id having max(a.interest)>=(select max(interest) from account\_info);

#16) Write a query to display the customer id, customer name, account no, bank name, contact no and mail id of the customers who

#are from BANGALORE.

select p.customer\_id, p.customer\_name, a.account\_no, b.bank\_name, p.contact\_no,p.mail\_id from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code where p.address like '%Banglore';

select a.customer\_id,b.customer\_name,a.account\_no,c.bank\_name,b.contact\_no,b.mail\_id from

account\_info a,customer\_personal\_info b,bank\_info c

where a.customer\_id=b.customer\_id and a.ifsc\_code=c.ifsc\_code and b.address like '%BANGALORE';

#17) Write a query which will display customer id, bank name, branch name, ifsc code, registration date,activation date of the

#customers whose activation date is in the month of march (March 1'st to March 31'st).

select p.customer\_id,b.bank\_name,b.branch\_name,a.ifsc\_code,a.registration\_date,a.activation\_date from

customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id join bank\_info b on

a.ifsc\_code=b.ifsc\_code where a.activation\_date like '%%%%-03-%%';

#18) Write a query which will calculate the interest amount and display it along with customer id, customer name, account number,

#account type, interest, and initial deposit amount.

#calculate: ((interest/100) \* initial deposit amt) with column name 'interest\_amt' (alias)

select p.customer\_id,p.customer\_name,a.account\_type,a.account\_no,a.initial\_deposit,a.interest,concat((a.interest/100)\*a.initial\_deposit) as interest\_amt

from customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id;

#19) Write a query to display the customer id, customer name, date of birth, guardian name, contact number, mail id,

#reference name who has been referenced by 'RAGHUL'

select p.customer\_id, p.customer\_name, p.date\_of\_birth,p.Guardian\_name,p.contact\_no,p.mail\_id,r.reference\_acc\_name from

customer\_personal\_info p join customer\_reference\_info r on

p.customer\_id=r.customer\_id where r.reference\_acc\_name like 'raghul';

#20) Write a query which will display the customer id, customer name and contact number with ISD code of all customers in below

#mentioned format. Sort the result based on the customer id in descending order.

#Format for contact number is : ""+91-3digits-3digits-4digits""Example: +91-924-234-2312Use ""CONTACT\_ISD"" as alias name."

select b.customer\_id,b.customer\_name,

concat('+91-',substr(cast(contact\_no as char),1,3),'-',substr(cast(contact\_no as char),4,3),'-',substr(cast(contact\_no as char),7)) as contact\_isd

from customer\_personal\_info b;

select b.customer\_id,b.customer\_name,

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

substring(contact\_no,7))) as contact\_ISD from customer\_personal\_info b;

#21) Write a query which will display account number, account type, customer id, customer name, date of birth, guardian name,

#contact no, mail id , gender, reference account holders name, reference account holders account number, registration date,

#activation date, number of days between the registration date and activation date with alias name "NoofdaysforActivation",

#bank name, branch name and initial deposit for all the customers.

select a.account\_no,a.account\_type,p.customer\_id,p.customer\_name,p.date\_of\_birth,p.guardian\_name,p.contact\_no,p.mail\_id,p.gender,

r.reference\_acc\_name,r.reference\_acc\_no,a.registration\_date,a.activation\_date,

datediff(a.activation\_date,a.registration\_date) as NoofdaysforActivation,b.bank\_name,b.branch\_name,a.initial\_deposit

from account\_info a,customer\_personal\_info p,customer\_reference\_info r,bank\_info b where

a.customer\_id=p.customer\_id and p.customer\_id=r.customer\_id and a.ifsc\_code=b.ifsc\_code;

#22) Write a query which will display customer id, customer name, guardian name, identification doc type,reference account holders

#name, account type, ifsc code, bank name and current balance for the customers who has only the savings account.

#Hint: Formula for calculating current balance is add the intital deposit amount and interest and display without any decimals.

#Use ""CURRENT\_BALANCE"" as alias name."

select p.customer\_id,p.customer\_name,p.guardian\_name,p.identification\_doc\_type,r.reference\_acc\_name,a.account\_type,a.ifsc\_code,

b.bank\_name,(a.interest+a.initial\_deposit) as CURRENT\_BALANCE

from account\_info a,customer\_personal\_info p,customer\_reference\_info r,

bank\_info b

where a.customer\_id=p.customer\_id and p.customer\_id=r.customer\_id and a.ifsc\_code=b.ifsc\_code and a.account\_type='SAVINGS';

#23) Write a query which will display the customer id, customer name, account number, account type, interest, initial deposit;

#check the initial deposit, if initial deposit is 20000 then display ""high"",if initial deposit is 16000 display 'moderate',

#if initial deposit is 10000 display 'average', if initial deposit is 5000 display 'low', if initial deposit is 0 display

#'very low' otherwise display 'invalid' and sort by interest in descending order.

#Hint: Name the column as ""Deposit\_Status"" (alias).

#Strictly follow the lower case for strings in this query."

select p.customer\_id,p.customer\_name,a.account\_no,a.account\_type,a.interest,a.initial\_deposit,

case

when a.initial\_deposit=20000 then 'High'

when a.initial\_deposit=16000 then 'Moderate'

when a.initial\_deposit=10000 then 'Average'

when a.initial\_deposit=5000 then 'Low'

when a.initial\_deposit=0 then 'Very Low'

else 'invalid'

end as Deposit\_status from account\_info a,customer\_personal\_info p where a.customer\_id=p.customer\_id

order by interest desc;

#24) Write a query which will display customer id, customer name, account number, account type, bank name, ifsc code,

#initial deposit amountand new interest amount for the customers whose name starts with ""J"".

#Hint: Formula for calculating ""new interest amount"" is

#if customers account type is savings then add 10 % on current interest amount to interest amount else display

#the current interest amount.Round the new interest amount to 2 decimals.<br/> Use ""NEW\_INTEREST"" as alias name for

#displaying the new interest amount.

select a.customer\_id,p.customer\_name,a.account\_no,a.account\_type,b.bank\_name,a.ifsc\_code,a.initial\_deposit,round(case

when a.account\_type like 'savings' then a.interest+(a.interest\*10/100)

when a.account\_type not like 'savings' then a.interest end,2) as NEW\_INTEREST

from account\_info a,customer\_personal\_info p,bank\_info b

where a.customer\_id=p.customer\_id and a.ifsc\_code=b.ifsc\_code and p.customer\_name like 'J%';

#25) Write query to display the customer id, customer name, account no, initial deposit, tax percentage as calculated below.

#Hint:If initial deposit = 0 then tax is '0%'.If initial deposit <= 10000 then tax is '3%'

#If initial deposit > 10000 and initial deposit < 20000 then tax is '5%'.

#If initial deposit >= 20000 andinitial deposit <=30000 then tax is '7%'.

#If initial deposit > 30000 then tax is '10%'.Use the alias name 'taxPercentage'

select p.customer\_id,p.customer\_name,a.account\_no,a.initial\_deposit,case

when a.initial\_deposit = 0 then '0%'

when a.initial\_deposit <= 10000 then '3%'

when a.initial\_deposit > 10000 and a.initial\_deposit < 20000 then '5%'

when a.initial\_deposit >= 20000 and a.initial\_deposit <=30000 then '7%'

when a.initial\_deposit > 30000 then '10%' end as 'taxPercentage'

from customer\_personal\_info p join account\_info a on

p.customer\_id=a.customer\_id;

……………………………………………………….

**Item Loan Database Queries**

1.Please follow instructions given below.

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

3 rows

**SELECT item\_category , count(item\_id) Count\_category**

**FROM item\_master**

**GROUP BY item\_category order by count\_category DESC;**

2.Please follow instructions given below.

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

1 row

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

FROM employee\_master

WHERE department= 'HR'

3.Please follow instructions given below.

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

1 row

select employee\_id,employee\_name,designation,department from employee\_master

where employee\_id

not in (select employee\_id from employee\_issue\_details) order by employee\_id;

4.Please follow instructions given below.

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

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

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

1 row

select em.employee\_id,em.employee\_name from employee\_master em join employee\_issue\_details eid

on em.employee\_id=eid.employee\_id join item\_master im on eid.item\_id=im.item\_id

and im.item\_valuation>=all(select im.item\_valuation from employee\_master em

join employee\_issue\_details eid

on em.employee\_id=eid.employee\_id join item\_master im on eid.item\_id=im.item\_id)

order by employee\_id;

5.Please follow instructions given below.

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

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

9 rows

select eid.issue\_id,eid.employee\_id,em.employee\_name from employee\_issue\_details eid join

employee\_master em on eid.employee\_id=em.employee\_id group by eid.issue\_id,eid.employee\_id

order by eid.issue\_id;

6.Please follow instructions given below.

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

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

3 rows

SELECT employee\_id, employee\_name

FROM employee\_master

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

order by employee\_id;

7.Please follow instructions given below.

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

1 row

select count(eid.loan\_id) as No\_of\_Cards from employee\_card\_details eid join employee\_master em

on eid.employee\_id=em.employee\_id where em.employee\_name='Ram'

8.Please follow instructions given below.

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

1 row

select count(ecd.employee\_id) as Count\_Stationary from employee\_card\_details ecd

join loan\_card\_master lcm on ecd.loan\_id=lcm.loan\_id where lcm.loan\_type='Stationary'

9.Please follow instructions given below.

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

5 rows

select em.employee\_id,em.employee\_name,count(eid.item\_id) as Count from employee\_master em join

employee\_issue\_details eid on em.employee\_id=eid.employee\_id group by em.employee\_id having

count(eid.item\_id)>=1 order by Count desc,employee\_id asc;

10.Please follow instructions given below.

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

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

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

2 rows

select em.employee\_id,em.employee\_name from employee\_master em join employee\_issue\_details eid

on em.employee\_id=eid.employee\_id join item\_master im on eid.item\_id=im.item\_id

and im.item\_valuation<=all (select im.item\_valuation from employee\_master em join employee\_issue\_details eid

on em.employee\_id=eid.employee\_id join item\_master im on eid.item\_id=im.item\_id) order by employee\_id;

11.Please follow instructions given below.

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

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

Consider only employees who have been issued atleast 1 item.

5 rows

select em.employee\_id,em.employee\_name,sum(im.item\_valuation) as TOTAL\_VALUATION

from employee\_master em

join employee\_issue\_details eid on em.employee\_id=eid.employee\_id join item\_master im

on eid.item\_id=im.item\_id group by em.employee\_id having count(im.item\_valuation)>=1

order by em.employee\_id;

12.Please follow instructions given below.

Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days.

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

5 rows

select distinct em.employee\_id,em.employee\_name from employee\_master em join employee\_issue\_details eid

on em.employee\_id=eid.employee\_id where datediff(return\_date,issue\_date)>365 order by

employee\_id;

13.Please follow instructions given below.

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

Display the records sorted in ascending order on employee id.

2 rows

select em.employee\_id,em.employee\_name,count(im.item\_id) as COUNT\_ITEMS from employee\_master em

join employee\_issue\_details eid on em.employee\_id=eid.employee\_id join item\_master im

on eid.item\_id=im.item\_id where item\_category='furniture' group by employee\_id having

count(COUNT\_ITEMS)>1 order by employee\_id;

14.Please follow instructions given below.

Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No\_of\_Employees. Display the records sorted in ascending order based on gender.

2 rows

select gender,count(employee\_id) as No\_of\_Employees from employee\_master group by

gender order by gender;

15.Please follow instructions given below.

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

3 rows

select employee\_id,employee\_name from employee\_master where year(date\_of\_joining)>2005

order by employee\_id;

16.Please follow instructions given below.

Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No\_of\_Furnitures.

Display the records sorted in ascending order based on issue\_status.

2 rows

select issue\_status,count(item\_id) as No\_of\_Furnitures from item\_master where item\_category='furniture' group by issue\_status order by

issue\_status;

17.Please follow instructions given below.

Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No\_of\_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.

16 rows

select item\_category,item\_make,item\_description,count(item\_id) as No\_of\_Items from

item\_master im group by item\_category,item\_make,item\_description order by

item\_category,item\_make,item\_description;

18.Please follow instructions given below.

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

1 row

select em.employee\_id,em.employee\_name,im.item\_id,im.item\_description from employee\_master em join

employee\_issue\_details eid on em.employee\_id=eid.employee\_id join item\_master im on

eid.item\_id=im.item\_id where year(eid.issue\_date)=2013 and month(eid.issue\_date)=01 order by

em.employee\_id,im.item\_id;

19.Please follow instructions given below.

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

Give the alias name for category count as COUNT\_CATEGORY.

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

1 row

select em.employee\_id,em.employee\_name,count(distinct im.item\_category) as COUNT\_CATEGORY from employee\_master em

join employee\_issue\_details eid on em.employee\_id=eid.employee\_id join item\_master im

on eid.item\_id=im.item\_id group by em.employee\_id having COUNT\_CATEGORY>=2

order by em.employee\_id;

20.Please follow instructions given below.

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

14 rows

select item\_id,item\_description from item\_master where item\_id not in (select item\_id

from employee\_issue\_details) order by item\_id;

21.Please follow instructions given below.

Write a query to display the employee id, employee name and&nbsp;&nbsp;total valuation&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. ]

1 row

select em.employee\_id,em.employee\_name,sum(im.item\_valuation) as TOTAL\_VALUATION from employee\_master em

join employee\_issue\_details eid on em.employee\_id=eid.employee\_id join item\_master im on

eid.item\_id=im.item\_id group by em.employee\_id having sum(im.item\_valuation) <= all

(select sum(im.item\_valuation) from employee\_master em

join employee\_issue\_details eid on em.employee\_id=eid.employee\_id join item\_master im on

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

22.Please follow instructions given below.

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

Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD\_VALID\_DATE.

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

SELECT ecd.employee\_id,employee\_name,

card\_issue\_date, if(lcd.duration\_in\_years=0, ‘NO-VALIDITY DATE’, date\_add(ec.card\_issue\_date, interval duration\_in\_years year)) as CARD\_VALIDITY\_DATE

FROM employee\_master em INNER JOIN

employee\_card\_details ecd

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

INNER JOIN loan\_card\_master lcd

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

order by employee\_name, CARD\_VALID\_DATE;

23.Please follow instructions given below.

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

3 rows

select distinct em.employee\_id,em.employee\_name from employee\_master em join employee\_issue\_details eid on

em.employee\_id=eid.employee\_id where em.employee\_id not in

(select employee\_id from employee\_issue\_details where year(issue\_date)=2013)

order by employee\_id;

24.Please follow instructions given below.

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

9 rows

select eid.issue\_id,em.employee\_id,em.employee\_name,im.item\_id,im.item\_description,eid.issue\_date

from employee\_issue\_details eid join employee\_master em on eid.employee\_id=em.employee\_id

join item\_master im on eid.item\_id=im.item\_id order by eid.issue\_date desc,eid.issue\_id;

25.Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product.&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. ]

1 row

select em.employee\_id,em.employee\_name,sum(im.item\_valuation) as TOTAL\_VALUATION

from employee\_master em join employee\_issue\_details eid on em.employee\_id=eid.employee\_id

join item\_master im on eid.item\_id=im.item\_id group by em.employee\_id having sum(im.item\_valuation)

>= all (select sum(im.item\_valuation) from employee\_master em join employee\_issue\_details eid on em.employee\_id=eid.employee\_id

join item\_master im on eid.item\_id=im.item\_id group by em.employee\_id);;

…………………………………………………………………………

**Problem # 1:** WAQ to Display the mobile details such as IMENO, Model Name produced by the manufacturer "Nokia".

Solution: select IME\_NO,Model\_Name from mobile\_master where manufacturer='Nokia';

**Problem # 2:** WAQ to display IMENO, Model Name,Manufacturer,Camerea Quality of mobiles whose camera quality is 5MP.

Solution: select m1.ime\_no,m1.model\_name,m1.manufacturer,m2.camera\_quality from mobile\_master m1 join mobile\_specification m2 on m1.ime\_no=m2.ime\_no where m2.camera\_quality='5MP';

**Problem # 3:** WAQ to display Model Name,Quantity sold on the date 25-APR-12.

Solution: select model\_name,count(ime\_no) from sales\_info where sales\_date='23-APR-12' group by model\_name;

**Problem # 4:** WAQ to display distributor id ,mobile supply details such as mobile model name, quantity supplied in sorted order of distributor id.

Solution: select Distributor\_ID,Model\_Name,count(Model\_Name) from Mobile\_Master group by Distributor\_ID,Model\_Name order by Distributor\_id;

**Problem # 5:** WAQ to display the IMENO,model name,manufacturer,price and discount of all mobiles regardless of whether the mobile is sold or not.

Solution: select m1.ime\_no,m1.model\_name,m1.manufacturer,m1.price,s.discount from mobile\_master m1 left outer join sales\_info s on m1.ime\_no=s.ime\_no;

**Problem # 6:** WAQ to display the distributor details such as distributor name,mobile number and email of the model 'Nokia 1100'.

Solution: select Distributor\_Name,Mobile from Distributor where Distributor\_Id=(select distributor\_id from mobile\_master where model\_name='Nokia 1100');

**Problem # 7:** WAQ to display the Ime No and Model Name of mobiles which are not sold(Hint : use minus operator)

Solution: select ime\_no ,model\_name from mobile\_master minus select ime\_no ,model\_name from sales\_info;

**Problem # 8:** WAQ to display the Ime No and Model Name of mobiles which are sold(Hint : use intersect operator)

Solution: select ime\_no ,model\_name from mobile\_master intersect select ime\_no ,model\_name from sales\_info;

**Problem # 9:** WAQ to display the ImeNO, Model Name,Manufacturer, Price and NewPrice of all mobiles.

(Hint : find new price as 10% of the price with column name "New Price")

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

**Problem # 10:** WAQ to display mobile model, manufacturer and price for the mobiles having a price range from 8500 to 25300.

Solution: select model\_name,manufacturer,price from mobile\_master where price between 8500 and 25300;

Average Questions:

**Problem # 1:** WAQ to display the Model Name,Manufacturer, Price , Warranty , Internal memory, memory card capacity,gprs support,bluetooth,camera quality and OS for the mobile with IME NO "MC1000104" .

Solution: select m1.model\_name,m1.manufacturer,m1.warranty\_in\_years,m1.price,m2.Internal\_mem\_in\_MB,m2.Memory\_Card\_Capacity\_GB, m2.GPRS,m2.Bluetooth,m2.Camera\_Quality,m2.OS from mobile\_master m1 join mobile\_specification m2 on m1.IME\_No=m2.IME\_No where m1.IME\_no='MC1000104';

**Problem # 2:** WAQ to display IMENO, Model Name,Manufacturer,Price ,GPRS information,Memory card support of mobiles which has GPRS support with memory card capacity 16GB or above.

Solution: select m1.ime\_no,m1.model\_name,m1.manufacturer,m1.price,m2.gprs,m2.Memory\_Card\_Capacity\_GB from mobile\_master m1 join mobile\_specification m2 on m1.ime\_no=m2.ime\_no where m2.GPRS='Yes' and m2.Memory\_Card\_Capacity\_GB>=16;

**Problem # 3:** WAQ to display the customer name ,mobile purchase details such as IMENO,Model Name ,Purchase Date,Net amount paid in sorted order of customer name.

Solution: select c1.Customer\_Name,m1.IME\_NO,m1.Model\_Name,m1.Sales\_Date,m1.Net\_Amount from Customer\_Info c1 join Sales\_info m1 on m1.Customer\_ID=c1.Customer\_ID order by c1.Customer\_Name;

**Problem # 4:** WAQ to display the distributor details such as distributor id ,name ,address,contact no who has supplied the maximum number of mobiles.

Solution: select distributor\_id,distributor\_name,address,mobile,email from distributor where distributor\_id=(select distributor\_id from mobile\_master having count(distributor\_id)=(select max(count(distributor\_id)) from mobile\_master group by distributor\_id ) group by distributor\_id);

**Problem # 5:** WAQ to display the IMENO,model name,manufacturer,price and discount of all mobiles regardless of whether the mobile is sold or not.

[Hint: If not sold, display discount as "Not Sold"]

Solution: select m1.ime\_no,m1.model\_name,m1.manufacturer,m1.price,nvl(to\_char(m2.discount),'Not Sold') "discount" from mobile\_master m1 left outer join sales\_info s on m1.ime\_no=s.ime\_no;

**Problem # 6:** WAQ to display the report containing the sales date and total sales amount of the dates between 20-APR-12 and 25-APR-12.

(Hint : total sales amount column should be displayed as "Total Sales Amount" )

Solution: select sales\_date,sum(net\_amount) "Total Sales Amount" from sales\_info

where sales\_date between '20-APR-12' and '25-APR-12' group by sales\_date;

**Problem # 7:** WAQ to display mobile imeno,model name,manufacturer and price of the mobiles which are having the longest battery life.

Solution: select ime\_no,model\_name,manufacturer,price from mobile\_master where ime\_no in(select ime\_no from mobile\_specification where battery\_life\_hrs=(select max(battery\_life\_hrs) from mobile\_specification));

**Problem # 8:** WAQ to display the ImeNO, Model Name,Manufacturer, Price of mobiles having the maximum price.

Solution: select ime\_no,model\_name,manufacturer,price from mobile\_master where ime\_no in(select ime\_no from mobile\_master where price=(select max(price) from mobile\_master));

**Problem # 9:** WAQ to display the customer details such as Customer ID,Customer Name, Address, Total Purchase amount.

Solution: select c1.Customer\_ID,c1.Customer\_Name,c1.Address,(select sum(Net\_Amount) from sales\_info where Customer\_id=c1.Customer\_ID) ”Total Purchase Amount" from Customer\_info c1;

**Problem # 10:** WAQ to display the most costly mobile information such as mobile model, manufacturer and price manufactured by "Samsung".

Solution: s

Complex Questions:

•**Problem # 1:** WAQ to display the customer details such as Customer ID,Customer Name, Address and Total Purchase amount having the maximum purchase amount.

Solution: select Customer\_ID,Customer\_Name,Address from customer\_info where customer\_id=(select customer\_id from sales\_info having sum(Net\_Amount)=(select max(sum(net\_amount)) from sales\_info group by customer\_id) group by customer\_id);

**Problem # 2:** WAQ to determine whether the mobile with ime no "MC1000105" is been sold out or not and display the model name,sales status.(Hint: If sold display status as "Sold Out" with column name "Sales Status").

Solution: select model\_name,(select 'Sold Out' from sales\_info where ime\_no='MC1000105')"Sales Status" from mobile\_master where ime\_no='MC1000105' ;

**Problem # 3:** WAQ to display the mobile information such as ime no,model name,manufacturer ,distributor id ,distributor name and price supplied by the distributor named 'AXA Ltd' .

Solution: select m1.ime\_no,m1.model\_name,m1.manufacturer,d1.distributor\_id,d1.distributor\_name,m1.price from mobile\_master m1 join distributor d1 on m1.distributor\_id=d1.distributor\_id and d1.distributor\_id=(select distributor\_id from distributor where distributor\_name='AXA Ltd');

**Problem # 4:** WAQ to display distributor details who supplies mobile with the following speficiations such as 3G Network, Android OS, 5 MP Camera.

Solution: select distributor\_id,distributor\_name,address,mobile from distributor where distributor\_id IN (select distributor\_id from mobile\_master where ime\_no IN (select ime\_no from mobile\_specification where network\_3g='Yes' and os LIKE '%Android%' and camera\_quality='3.5MP' ));

**Problem # 5:** WAQ to Display the maximum sold mobile model name and manufacturer .

Solution: select distinct model\_name,manufacturer from mobile\_master where model\_name=(select model\_name from sales\_info having count(model\_name)=(select max(count(model\_name))from sales\_info group by model\_name)group by model\_name)

……………………………………………

questions and answers

Payroll Schema – Simple Questions

Problem # 1: Write a query to display Employee ID, Employee Name, Department ID and Department Name of all employees who has a department assigned.

àselect e.empid,e.empname,e.deptid,d.deptname from employee e, department d where e.deptid=d.deptid and e.deptid is not null;

Problem # 2: Write a query to display the Employee ID, Employee Name, Basic Pay of all employees who are in employee category 'A'

à select e.empid,e.empname,s.basic from employee e,salary s where e.employee\_category=s.employee\_category and s.employee\_category='A';

Problem # 3: Write a query to display the Employee ID, Employee Name, Department ID and Department Name of all employees who has a department assigned and department location is ‘CHENNAI’.

à select e.empid,e.empname,e.deptid,d.deptname from employee e,department d where e.deptid is not null and e.deptid=d.deptid and d.location='Chennai';

Problem # 4: Write a query to display the employee ID and employee name of the employees who have not been assigned a department yet.

à select empid,empname from employee where deptid is null;

Problem # 5: Write a query to display the employee ID, employee name and joining date of the employees who joined before 2005.

à select empid,empname,joining\_dt from employee where joining\_dt<'2005-1-1';

Problem # 6: write a query to display employee name and date of joining for all employees.(Date should be displayed in the format “23/JANUARY/2012” with Alias “ JOINING\_DATE” in select statement)

à select empname,date\_format(joining\_dt,'%d' '/' '%M' '/' '%Y') as JOINING\_DATE from employee;

Problem # 7: Write a query to display the employee ID, employee name and joining date of the employees who joined between Jan 1 2005 and Dec 31’st 2010

à select empid,empname,joining\_dt from employee where joining\_dt between '2005-1-1' and '2010-31-12';

Problem # 8: Write a query to display the employee ID, employee name and joining date of the employees who joined in MARCH.

à select empid,empname,joining\_dt from employee where extract(month from joining\_dt)=3;

Problem # 9: Write a query to display all employee names which begins with 'R'.

à select empname from employee where empname like 'R%';

Problem # 10: Write a query to display the first five employees name in the employee table and the respective row number (use ROWNUM for identifying the first five records)

à select @rowno:=@rowno+1 as ROWNUM,empname from employee,(select @rowno:=0) r limit 0,5;

Payroll Schema – Average Questions

Problem # 1: Write a query to display the EmployeeID, Employee Name,Net Pay of an employee whose ID is "E001" for the month of APRIL

à Write a query to display the EmployeeID, Employee Name,Net Pay of an employee whose ID is "E001" for the month of APRIL

Problem # 2: Write a query to display the department id and no of employees in each department sorted by department id. (Exclude department with null values).

à select d.deptid,count(e.empname) from department d,employee e where e.deptid=d.deptid group by d.deptid order by deptid;

Problem # 3: Write a query to display the EmployeeID, Employee Name and the total number of leaves each employee has taken with “Total\_Leaves” as alias.

Hint: For Example, if employee “E001” has taken 2 days leave on January and 3 days leave of February then his total number of leaves will be 5 days. Similarly display the total number of leaves for all employees.

à select e.empid,e.empname,sum(el.total\_leaves) from employee e, employee\_leave el where e.empid=el.empid group by el.empid;

Problem # 4: Write a query to display the EmployeeID, Employee Name, DOB and Age in Years without decimals with alias name "Age".

Hint: Formula for age calculation is Age = current date- dob/12, round this to the nearest whole number.

à select empid,empname,dob,round((datediff(current\_date,dob)/30)/12) as Age from employee;

Problem # 5: Write a query to display employee id, employee name of all employees who doesn't have LOP amount for the month of APR and year 2012.

à select e.empid,e.empname from employee e, payroll p where e.empid=p.empid and p.lopamount=0 and p.month='apr' and p.year=2012;

Problem # 6: Write a query to display employee name, professional tax, netpay of employees with employee category 'A'

à select e.empid,e.empname,s.proftax,p.netpay from employee e,salary s,payroll p where e.employee\_category=s.employee\_category and e.employee\_category='a' and e.empid=p.empid;

Problem # 7: Write a query to display employee id, employee name,department id who are having netpay in the range 10000 – 20000

à select e.empid,e.empname,d.deptid from employee e,department d,payroll p where e.empid=p.empid and p.netpay between 10000 and 20000 group by p.empid;

Problem # 8: Write a query to display employee names whose total deduction is more than 2000 for the month of APRIL.

à Write a query to display employee names whose total deduction is more than 2000 for the month of APRIL.

Problem # 9: Write a query to display employee id, employee name, department id, department name of all employees regardless of whether an employee is assigned a department or not.

à Write a query to display employee id, employee name, department id, department name of all employees regardless of whether an employee is assigned a department or not.

Problem # 10: Write a query to display Employee ID, Employee Name, Department ID, Years of Experience and Employee Category of the employees who have availed leaves more than 10 days.

Hint: Use the total\_leaves column to check the leave condition for more than ten days.

àselect empid,empname,deptid,yrs\_of\_exp,employee\_category from employee where empid in (select empid from employee\_leave group by empid having sum(total\_leaves)>10);

Payroll Schema – Complex Questions

Problem # 1: Write a query to display employee id, employee name and remaining casual leaves (alias- RemainingLeaves) for the employee with employee id "E002". Based on the total causal leaves available, subtract the number of causal leaves he has availed to get the remaining leaves.

Hint: CL – Causal leave.

à select (l.cl-sum(el.total\_leaves)) as RemainingLeaves from employee\_leave el,eleave l where empid='E002' and leave\_type='CL' and l.Leave\_category=(select leave\_category from employee where empid='E002');

Problem # 2: Write a query to display employee id, employee name and total number of leaves he can take (hint: with “EligibleLeave” as alias). This should be retrieved for all the employees. Sum all the EL, ML and EL leaves for the each employee’s category to get the total leaves.

Hint:

EMPLOYEE\_INFO table has Employee’s leave Category. For example employee “E001” belong to “X” leave category.

LEAVE\_INFO table has the Leave Category and number of CL, EL and ML available for them. For example, Employee E001 belongs to X category and he has 18 days of CL and 5 days of EL and 10 days of ML that he can avail.

So, E001’s eligible leave would be 33 days which is sum of all his leaves. Similarly calculate for all employees.

à select e.empid,e.empname,(l.cl+l.el+l.ml) as EligibleLeaves from employee e,eleave l where l.leave\_category=e.leave\_category;

Problem # 3: Write a Query to display employee id, employee name, department id, department name, net pay of all employees who have drawn the highest salary (net pay) in the month of APRIL 2012.

Hint: For example if there are 10 employees where 3 employees have got a salary of 1000 which is the highest salary of the employee in the month of April all the three records needs to be displayed.

à select e.empid,e.empname,e.deptid,d.deptname,p.netpay from employee e,department d,payroll p where e.empid=p.empid

and p.netpay=(select max(netpay) from payroll where month='apr' and year='2012') group by e.empid;

Problem # 4: Write a query to display employee id, employee name, basic pay and tax percentage for all employees. Use “TaxPercentage” as alias. Display the Tax percentage for all employees based on the following criteria: (If Basic Pay <= 4000 then tax percentage should be 10%, basic <= 5000 then 20%, basic<=6000 then 30% basic > 6000 then 40%).

……………………………………………………

# 1) Display all undergraduate student whose name starts with ‘S’ and is of length between 5 and 20

select sname from student where squal='undergraduate' and sname like 's%'

and length(sname) between 5 and 20

order by sname;

# 2) Display the student who are senior citizen (>=60).

select sname from student where datediff(current\_date,date(sdob))/365>='60';

# 3) Display student who were born after 1st of June 1980.

select sname from student where sdob>'1980-06-01'

# 4) The student are suppose to only provide mobile numbers .All mobile numbers should start with zero followed by 10 digits. Display student name having invalid phone numbers.

select sname from student where sphone like '0%' or length(sphone)<11 or sphone is null

order by sname;

# 5) All emails should have “@” anywhere after the first character and should end with “.com”.

Display count of students having invalid email id.

select count(sname) from student where semail not like '%@%.com' or semail is null;

# 6) Display the name and email of student who have a Gmail account.

select sname,semail from student where semail like '%gmail%';

# 7) Display the above record but do not consider invalid email id.

select sname,semail from student where semail like'%@gmail%';

# 8) Display the qualification and the total number of students registered based on their qualifications.

(Alias use “totalStud” for total number of students)

select squal,count(sid) as totalstud from student group by squal

# 9) Display the full name of the month and the total number of students who are having their birthday in that month.

(Alias use “Month” and “Total”)

select monthname(sdob),count(sid) from student group by month(sdob);

# 10) Display the student name that was born in a leap year ordering by student name and year of birth

select sname from student where year(sdob)%4=0 or year(sdob)%400=0 and year(sdob)%100!=0

# 11)Display student whose city is Kolkata as “HomeStudent ” and others as “DistanceStudent” under a column “Remarks”.

Also display the name and city of the student

select sname,scity,case

when scity='kolkata' then 'Home Student'

when scity!='kolkata' then 'DistanceStudent' end as 'Remarks'

from student;

# 12) Display batchid, coursename, batch start date, batch end date for all batches.

#(batch end date=batch start date +course duration).

select b.batchid,c.coursename,b.bsdate,date\_add(b.bsdate,interval c.courseduration day) as bachenddate from

batch b ,course c where b.courseid=c.courseid;

# 13) Display all batchid having a difference of 10 hours and less between its starting and ending date.

select batch.batchid from batch join course on

batch.courseid=course.courseid

group by batch.batchid having date\_diff(batch.bsdate,date\_add(b.bsdate,interval c.courseduration day))<=10;

# 14) Display all batches having similar start date and strength.

select batch.batchid from batch, batch a,batch b where

a.bsdate=b.bsdate and a.bstrength=b.bstrength

and a.batchid!=b.batchid

group by batch.batchid;

# 15) Display student who enrolled for the batch after its start date.

select distinct sname from student join enrollment on

student.sid=enrollment.sid join batch on

enrollment.batchid=batch.batchid where batch.bsdate<enrollment.edate;

# 16) Display the studentid, studentname , totalfees for all student.

select student.sid,student.sname,sum(course.coursefees) as totalfees from student join enrollment on

student.sid=enrollment.sid join batch on

enrollment.batchid=batch.batchid join course on

batch.courseid=course.courseid

group by student.sid;

# 17) Display courses which are not being taught currently along with courses which are being taught.

#Also display the batchid for the courses currently running and null for non executing courses.

select b.batchid,c.coursename,case when

date\_add(b.bsdate,interval c.courseduration day)>=current\_date then b.batchid

else 'null' end as cou

from batch b join course c on b.courseid=c.courseid

group by c.coursename;

# 18) Display count of students having no contact information. (Either email or phone).

select count(student.sid) from student where semail is null or sphone is null;

# 19)) Display coursename having above average fees.

select course.coursename from course where course.coursefees>(select avg(coursefees) from course);

# 20)) Display coursename where fees are less than the average fees of its category

select coursename,coursefees from course c1 where c1.coursefees<(

select avg(c2.coursefees)from course c2 where c1.coursecategory=c2.coursecategory);

# 21)Display the coursename having the highest enrollment

select course.coursename from course join batch on

course.courseid=batch.courseid join enrollment on

batch.batchid=enrollment.batchid

group by course.courseid having count(enrollment.sid)>=all(select count(enrollment.sid) from course join batch on

course.courseid=batch.courseid join enrollment on

batch.batchid=enrollment.batchid

group by course.courseid);

# 22)Display student name having duplicate email ids.

select distinct a.sname from student a,student b where

a.sid!=b.sid and a.semail = b.semail;

#23) Display student name having similar name but different email ids.

select distinct a.sname from student a,student b where

a.sname=b.sname and a.semail!=b.semail;

# 24) Display the student name, date of birth and their zodiac sign. Use Zodiac as alias

a. Aries ??Mar 21-Apr 19

b. Taurus?Apr 20 –May 20

c. Gemini?May 21-Jun 20

d. Cancer ??Jun21- Jul22

e. Leo ??Jul 23- Aug 22

f. Virgo ??Aug 23-Sept 22

g. Libra ??Sept 23-Oct 22

h. Scorpio ??Oct 23- Nov 21

i. Sagittarius ??Nov 22-Dec 21

j. Capricorn ??Dec 22- Jan 19

k. Aquarius ?Jan 20 – Feb 18

l. Pisces ?Feb 19- Mar 20.

select sname ,sdob,case

when date\_format(sdob,'%m%d') between '03-21' and '04-19' then 'aries'

when date\_format(sdob,'%m%d') between '04-20' and '05-20' then 'Taurus'

when date\_format(sdob,'%m%d') between '05-21' and '06-20' then 'gemini'

when date\_format(sdob,'%m%d') between '06-21' and '07-22' then 'Cancer'

when date\_format(sdob,'%m%d') between '07-23' and '08-22' then 'Leo'

when date\_format(sdob,'%m%d') between '08-23' and '09-22' then 'Vigo'

when date\_format(sdob,'%m%d') between '09-23' and '10-22' then 'Libra'

when date\_format(sdob,'%m%d') between '10-23' and '11-21' then 'Scorpio'

when date\_format(sdob,'%m%d') between '11-22' and '12-21' then 'Sagittarius'

when date\_format(sdob,'%m%d') between '12-22' and '01-19' then 'Capricorn '

when date\_format(sdob,'%m%d') between '01-20' and '02-18' then 'Aquarius '

when date\_format(sdob,'%m%d') between '02-19' and '03-20' then 'Pisces'

end as sign from student;

# 25) Display the course name fetching the 2nd highest revenue.

SELECT MAX(ASD), SDF FROM

(SELECT COURSE.COURSENAME AS SDF , COUNT(STUDENT.SID) AS CON , COURSE.COURSEFEES AS FEE,

COUNT(STUDENT.SID)\* COURSE.COURSEFEES AS ASD

FROM COURSE

JOIN BATCH ON COURSE.COURSEID=BATCH.COURSEID

JOIN ENROLLMENT ON BATCH.BATCHID = ENROLLMENT.BATCHID

JOIN STUDENT ON ENROLLMENT.SID=STUDENT.SID

GROUP BY COURSE.COURSENAME, COURSE.COURSEFEES)A

WHERE ASD <> (SELECT MAX(ASD) FROM (SELECT COUNT(STUDENT.SID)\* COURSE.COURSEFEES AS ASD

FROM COURSE

JOIN BATCH ON COURSE.COURSEID=BATCH.COURSEID

JOIN ENROLLMENT ON BATCH.BATCHID = ENROLLMENT.BATCHID

JOIN STUDENT ON ENROLLMENT.SID=STUDENT.SID

GROUP BY COURSE.COURSENAME, COURSE.COURSEFEES) B)

# 26) Generate report which displays the batch number and the number of seats vacant. [Use Alias “Vacant”].

select batch.batchid,(batch.bstrength-count(enrollment.batchid)) as vacant from batch join enrollment on

batch.batchid=enrollment.batchid

group by batch.batchid;

# 27) Which among the following have the highest enrollment? (Graduate,Undergraduate or postgraduate).

# Write query to display “Qualification” , “HighestEnrollmet” (Use Alias”).

select distinct squal from student group by squal

having count(squal)>=all(select count(squal) from student group by squal)

# 28) Display student name, age[Alias] , coursename, batchid, batch\_start\_ date, batch\_ end \_date ,

#and enrollment date in the

#format 1st of Jan , 2012 .

select student.sname,datediff(current\_date,student.sdob) as age,course.coursename,batch.batchid,batch.bsdate,

date\_add(batch.bsdate,interval course.courseduration day) as enddate,

concat(cast(day(enrollment.edate)as char) ,'st of ' , (cast(substr(monthname(enrollment.edate),1,3)as char)),' , ', cast(year(enrollment.edate) as char))as dates

from student join enrollment on

student.sid=enrollment.sid join batch on

enrollment.batchid=batch.batchid join course on

batch.courseid=course.courseid;

#29) Display report in the following format. Consider Sat & Sun as holiday.

#Week Enrollment

#Weekday 10

#Hoilday 05

select case when dayname(edate)='Monday' then 'Weekday'

when dayname(edate)='Tuesday' then 'Weekday'

when dayname(edate)='Wednesday' then 'Weekday'

when dayname(edate)='Thursday' then 'Weekday'

when dayname(edate)='Friday' then 'Weekday'

else 'holiday'

end as 'week', count(sid) as 'enrollment'

from `enrollment`

group by week desc;

……………………………………………………………………………….

**Video Management database queries:**

1.Please follow instructions given below.

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

11 rows

**select mm.movie\_name, count(cid.issue\_id) as ISSUE\_COUNT**

**from movies\_master mm left outer join customer\_issue\_details**

**cid on mm.movie\_id=cid.movie\_id group by mm.movie\_name**

**order by ISSUE\_COUNT desc,mm.movie\_name asc;**

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.

4 rows

**select customer\_id,customer\_name,age,**

**ifnull(concat('+91-',substring(contact\_no,1,3),'-',substring(contact\_no,4,3),'-',substring(contact\_no,7,4)),'N/A')**

**as CONTACT\_ISD from customer\_master where age>25 and year(date\_of\_registration)=2012**

**order by age,customer\_name;**

3.Please follow instructions given below.

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

Hint: Use NO\_OF\_MOVIES as alias name for number of movies.

3 rows

Ans:

**select movie\_category,count(movie\_id) as NO\_OF\_MOVIES from movies\_master group by movie\_category**

**order by NO\_OF\_MOVIES desc,movie\_category asc;**

4.Please follow instructions given below.

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

1 row

**select count(ccd.customer\_id) as CUSTOMER\_COUNT from customer\_card\_details ccd join**

**library\_card\_master lcd on ccd.card\_id=lcd.card\_id where lcd.description='Gold Card';**

4.Please follow instructions given below.

Write a query to display the customer id, customer name, year of registration,library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

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

5 rows

select cm.customer\_id,cm.customer\_name,year(cm.date\_of\_registration) as REGISTERED\_YEAR,ccd.card\_id,ccd.issue\_date

from customer\_master cm join customer\_card\_details ccd on cm.customer\_id=ccd.customer\_id

order by cm.customer\_name desc;

5.Please follow instructions given below.

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

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

2 rows

**select cid.issue\_id,cid.customer\_id,cm.customer\_name from customer\_issue\_details cid join**

**customer\_master cm on cid.customer\_id=cm.customer\_id where cm.customer\_name like 'R%'**

**and cid.actual\_date\_return>cid.return\_date order by cm.customer\_name;**

6.Please follow instructions given below.

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT\_DOLLAR = amount/52.42 and round it to zero decimal places and display as $Amount. Example Assume 500 is the amount then dollar value will be $10.

Hint: Use AMOUNT\_DOLLAR as alias name for amount in dollar.

Display the records in ascending order based on customer name.

**SELECT ccd.customer\_id, customer\_name, ccd.card\_id, description,concat('$',round(amount/52.42,0)) AMOUNT\_DOLLAR FROM customer\_master cm INNER JOIN customer\_card\_details ccd ON cm.customer\_id=ccd.customer\_id INNER JOIN library\_card\_master lcm ON ccd.card\_id=lcm.card\_id INNER JOIN customer\_issue\_details cid ON cid.customer\_id = cm.customer\_id WHERE cm.date\_of\_registration=cid.issue\_date order by customer\_name;**

7.Please follow instructions given below.

Write a query to display the customer id, customer name,contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use CUSTOMER\_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.

**SELECT customer\_id , upper(customer\_name) CUSTOMER\_NAME,contact\_no,contact\_address FROM customer\_master WHERE customer\_id NOT IN ( select customer\_id from customer\_card\_details ) AND customer\_id IN ( SELECT customer\_id from customer\_issue\_details ) and contact\_address like '%Nagar' order by customer\_name ;**

8.Please follow instructions given below.

Write a query to display the movie id, movie name,release year,director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.

**select movie\_id,movie\_name , release\_year ,director\_name from movies\_master where lead\_actor\_name1 in(select lead\_actor\_name1 from(select**

**lead\_actor\_name1,count(movie\_id) ct from movies\_master group by lead\_actor\_name1)t where t.ct>=all(select count(movie\_id) from movies\_master**

**group by lead\_actor\_name1))order by movie\_name;**

9.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.

11 rows

select cm.customer\_name,count(cid.movie\_id) as MOVIE\_COUNT from customer\_master cm left join

customer\_issue\_details cid on cm.customer\_id=cid.customer\_id group by cm.customer\_name order by cm.customer\_name;

10.Please follow instructions given below.

Write a query to display serial number,issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

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

For Example Assume the issue id is I00005 then the serial number is 05

Hint: Alias name for serial number is 'SERIAL\_NO'

**19 rows**

**select substring(cid.issue\_id,5,2) as SERIAL\_NO,cid.issue\_id,cid.customer\_id,cm.customer\_name,mm.movie\_id,mm.movie\_name**

**from customer\_issue\_details cid join customer\_master cm on cm.customer\_id=cid.customer\_id**

**join movies\_master mm on cid.movie\_id=mm.movie\_id group by SERIAL\_NO,cid.customer\_id,mm.movie\_id**

**order by SERIAL\_NO;**

11.Please follow instructions given below.

Write a query to display the issue id,issue date, customer id, customer name and contact number for videos that are issued in the year 2013.Display the records in decending order based on issue date of the video.

7 rows

**select cid.issue\_id,cid.issue\_date,cid.customer\_id,cm.customer\_name,cm.contact\_no**

**from customer\_issue\_details cid join customer\_master cm on cid.customer\_id=cm.customer\_id**

**where year(issue\_date)=2013 group by issue\_id,issue\_date,customer\_id order by**

**issue\_date desc;**

12.Please follow instructions given below.

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <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.

1 row

**select movie\_id,movie\_name,concat(lead\_actor\_name1,' & ',lead\_actor\_name2) as ACTORS**

**from movies\_master where movie\_id**

**not in (select movie\_id from customer\_issue\_details) order by**

**movie\_name;**

13.Please follow instructions given below.

Write a query to display the director's name, movie name and lead\_actor\_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR\_NAME as alias name for director name column Display the records sorted in ascending order based on director\_name and then by movie\_name in descending order.

2 rows

**SELECT upper(director\_name) DIRECTOR\_NAME,movie\_name,lead\_actor\_name1 FROM movies\_master WHERE director\_name in (SELECT director\_name FROM movies\_master GROUP BY director\_name HAVING count(movie\_id)>1) order by director\_name, movie\_name desc;**

14.Please follow instructions given below.

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <br> Hint:Use NO\_OF\_CUSTOMERS as alias name for number of customers.

1 row

**select count(customer\_id) as NO\_OF\_CUSTOMERS from customer\_master where year(date\_of\_registration)**

**=2012 and contact\_no != 'NULL'**

15.Please follow instructions given below.

Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT\_DETAILS as alias name for customer contact.

11 rows

**select cm.customer\_name,ifnull(cm.contact\_no,cm.contact\_add) as CONTACT\_DETAILS,lcd.card\_id,lcd.description from customer\_master cm**

**left join customer\_card\_details ccd on cm.customer\_id=ccd.customer\_id**

**left join library\_card\_master lcd on ccd.card\_id=lcd.card\_id group by customer\_name,description,CONTACT\_DETAILS**

**order by customer\_name;**

16.Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO\_OF\_TIMES as alias name for number of times

4 rows

**select cm.customer\_id,cm.customer\_name,count(cid.movie\_id) as NO\_OF\_TIMES from customer\_master**

**cm join customer\_issue\_details cid on cm.customer\_id=cid.customer\_id group by customer\_id,movie\_id having**

**count(movie\_id)>1 order by customer\_name desc;**

17.Please follow instructions given below.

Write a query to display customer id, customer name,contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.&nbsp;

Hint:Use NO\_OF\_MOVIES as alias name for number of movies column.

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

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

5 rows

**select cid.customer\_id,cm.customer\_name,**

**concat('+91-',substring(cm.contact\_no,1,3),'-',substring(cm.contact\_no,4,3),'-',**

**substring(cm.contact\_no,7,4)) as CONTACT\_ISD,**

**mm.movie\_category,count(mm.movie\_category) as NO\_OF\_MOVIES from customer\_master**

**cm join customer\_issue\_details cid**

**on cm.customer\_id=cid.customer\_id join movies\_master mm on cid.movie\_id=mm.movie\_id**

**group by mm.movie\_category,cm.customer\_name having count(movie\_category)>1**

**order by cm.customer\_name,mm.movie\_category;**

18.Please follow instructions given below.

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

3 rows

**(select cm.customer\_id,cm.customer\_name from customer\_master cm**

**join customer\_issue\_details cid**

**on cm.customer\_id=cid.customer\_id group by cm.customer\_id**

**having count(cid.issue\_id) >= all (select count(cid.issue\_id) from customer\_master cm**

**join customer\_issue\_details cid**

**on cm.customer\_id=cid.customer\_id group by cm.customer\_id) order by cm.customer\_name)**

**union all**

**(select cm.customer\_id,cm.customer\_name from customer\_master cm**

**join customer\_issue\_details cid**

**on cm.customer\_id=cid.customer\_id group by cm.customer\_id**

**having count(cid.issue\_id) <= all (select count(cid.issue\_id) from customer\_master cm**

**join customer\_issue\_details cid**

**on cm.customer\_id=cid.customer\_id group by cm.customer\_id) order by cm.customer\_name)**

19.Please follow instructions given below.

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO\_OF\_TIMES as alias name

Display the records in ascending order based on customer name.

1 row

select cm.customer\_id,cm.customer\_name,count(mm.movie\_id) as NO\_OF\_TIMES from customer\_master cm

join customer\_issue\_details cid on cm.customer\_id=cid.customer\_id join

movies\_master mm on cid.movie\_id=mm.movie\_id where mm.movie\_category='comedy' group by customer\_id

order by customer\_name>1;

20.Please follow instructions given below.

Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL\_COST. Display the records sorted in ascending order based on customer id

6 rows

select cid.customer\_id,sum(mm.rental\_cost) as TOTAL\_COST from customer\_issue\_details cid

join movies\_master mm

on cid.movie\_id=mm.movie\_id group by customer\_id order by customer\_id;