

Bank Management System Queries:

Bank Management System Queries:

1. Please follow instructions given below.

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

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

```

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

ACCOUNT_NUMBER	CUSTOMER_NUMBER	FIRSTNAME	LASTNAME	ACCOUNT_OPENING_DATE
A00001	C00001	RAMESH	SHARMA	2012-12-15
A00002	C00002	AVINASH	MINHA	2012-06-12
A00003	C00003	RAHUL	RASTOGI	2012-05-17
A00004	C00002	AVINASH	MINHA	2013-01-27
A00005	C00006	CHITRESH	BARWE	2012-12-17
A00006	C00007	AMIT	BORKAR	2010-08-12
A00007	C00007	AMIT	BORKAR	2012-10-02
A00008	C00001	RAMESH	SHARMA	2009-11-09
A00009	C00003	RAHUL	RASTOGI	2008-11-30
A00010	C00004	PARUL	GANDHI	2013-03-01

2. Please follow instructions given below.

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

```
SELECT count(customer_number) Cust_Count
```

```
FROM customer_master
```

```
WHERE customer_city='Delhi'
```

CUST_COUNT
4

3. Please follow instructions given below.

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

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

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

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

4. Please follow instructions given below.

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

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

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

CUSTOMER_NUMBER	FIRSTNAME	ACCOUNT_NUMBER
C00001	RAMESH	A00008
C00003	RAHUL	A00009

5. Please follow instructions given below.

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

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

```

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

TRANSACTION_TYPE	TRANS_COUNT
DEPOSIT	3
WITHDRAWAL	3

6. Please follow instructions given below.

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

Give the alias name as Count_Customer for number of customers.

```

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

```

COUNT_CUSTOMER
4

7. Please follow instructions given below.

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

```

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

```

ACCOUNT_NUMBER	DEPOSIT_AMOUNT
A00001	10000
A00002	6000
A00007	17000

8. Please follow instructions given below.

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

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

BRANCH_CITY	NO_OF_ACCOUNTS
CHENNAI	0
DELHI	6
KOLKATA	0
MUMBAI	4

9.Please follow instructions given below.

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

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

FIRSTNAME

AMIT
AVINASH
RAHUL
RAMESH

10. Please follow instructions given below.

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

Display the records sorted in order based on customer number.

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

```
FROM customer_master cm INNER JOIN loan_details ld
```

```
ON cm.customer_number=ld.customer_number
```

```
GROUP BY customer_number
```

```
HAVING count(branch_id)>1
```

```
ORDER BY customer_number
```

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00001	RAMESH	SHARMA
C00002	AVINASH	MINHA

11. Please follow instructions given below.

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

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

```

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

```

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

12. Please follow instructions given below.

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

```

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

```

(Or)

```

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

```

COUNT
2

13. Please follow instructions given below.

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

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

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

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

ACCOUNT_NUMBER
A00001

14. Please follow instructions given below.

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

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

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

```
select branch_name,branch_city
```

```

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

FROM branch_master INNER JOIN account_master
ON branch_master.branch_id=account_master.branch_id
group by branch_name) order by branch_name;

```

BRANCH_NAME	BRANCH_CITY
ASAF ALI ROAD	DELHI

15. Please follow instructions given below.

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

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

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

```

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

```

ACCOUNT_NUMBER	DEPOSIT	WITHDRAWAL
A00001	10000	12000
A00002	6000	7000

16. Please follow instructions given below.

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

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

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

```

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

BALANCE_AMOUNT
-2000

17. Please follow instructions given below.

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

```

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

FROM customer_master cm inner JOIN account_master am

ON cm.customer_number=am.customer_number

INNER JOIN transaction_details td

ON am.account_number=td.account_number

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

```

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

18. Please follow instructions given below.

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

```

SELECT firstname

FROM customer_master INNER JOIN account_master

ON customer_master.customer_number=account_master.customer_number

GROUP BY firstname

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

```

FIRSTNAME
AMIT
AVINASH

RAHUL
RAMESH

19. Please follow instructions given below.

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

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

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

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

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

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

CUSTOMER_NUMBER	FIRSTNAME	LASTNAME
C00002	AVINASH	MINHA

20. Please follow instructions given below.

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

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

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

CUSTOMER_NUMBER	FIRSTNAME	BRANCH_ID	LOAN_AMOUNT
C00001	RAMESH	B00001	100000
C00001	RAMESH	B00003	600000
C00002	AVINASH	B00001	600000
C00002	AVINASH	B00002	200000
C00009	ABHISHEK	B00008	400000
C00010	SHANKAR	B00009	500000

21. Please follow instructions given below.

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

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

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

BRANCH_CITY	COUNT_BRANCH
CHENNAI	1
DELHI	4
KOLKATA	1
MUMBAI	3

22. Please follow instructions given below.

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

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

```

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

```

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

A00010	PARUL	GANDHI
--------	-------	--------

23. Please follow instructions given below.

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

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

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

```
FROM customer_master order by customer_number
```

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

24. Please follow instructions given below.

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

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

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

25. Please follow instructions given below.

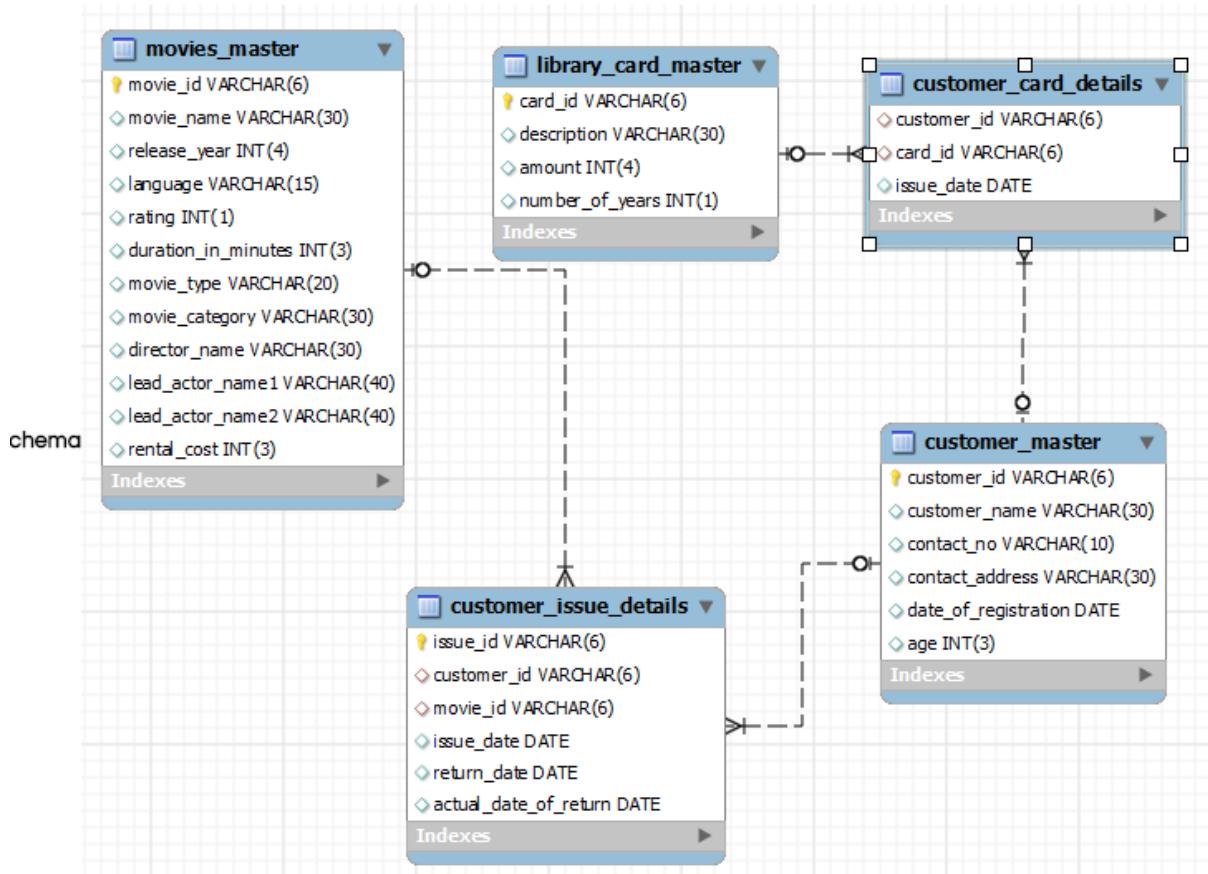
Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student.

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

```
SELECT firstname, customer_city,account_number  
FROM customer_master cm INNER JOIN account_master am  
ON cm.customer_number=am.customer_number
```

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

FIRSTNAME	CUSTOMER_CITY	ACCOUNT_NUMBER
PARUL	DELHI	A00010



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;
```

MOVIE_NAME	ISSUE_COUNT
DIE HARD	4
GONE WITH THE WIND	3
CASABLANCA	2
SHAUN OF THE DEAD	2
THE DARK KNIGHT	2
TITANIC	2
INCEPTION	1
OFFICE SPACE	1
THE MATRIX	1
YOUNG FRANKENSTEIN	1

THE NOTEBOOK	0
--------------	---

2. Please follow instructions given below.

Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

4 rows

```
select customer_id,customer_name,age,
ifnull(concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',
'substring(contact_no,7,4)),'N/A')

as CONTACT_ISD from customer_master where age>25 and year(date_of_registration)=2012

order by age,customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	AGE	CONTACT_ISD
C00007	GEETHA REDDY	30	+91-897-616-7890
C00005	SHIV PRASAD	30	N/A
C00002	AGNESH	35	+91-892-315-6781
C00004	RAJIB MITRA	45	+91-983-035-6781

3. Please follow instructions given below.

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

Hint: Use NO_OF_MOVIES as alias name for number of movies.

3 rows

Ans:

```

select movie_category,count(movie_id) as NO_OF_MOVIES from movies_master group by
movie_category

order by NO_OF_MOVIES desc,movie_category asc;

```

MOVIE_CATEGORY	NO_OF_MOVIES
ACTION	4
ROMANCE	4
COMEDY	3

4. Please follow instructions given below.

Write a query to display the number of customers having card with description “Gold card”.
 Hint:
Use CUSTOMER_COUNT as alias name for number of customers

1 row

```

select count(ccd.customer_id) as CUSTOMER_COUNT from customer_card_details ccd join
library_card_master lcd on ccd.card_id= lcd.card_id where lcd.description='Gold Card';

```

CUSTOMER_COUNT
2

4. Please follow instructions given below.

Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

Use REGISTERED_YEAR as alias name for year of registration.

5 rows

```

select cm.customer_id,cm.customer_name,year(cm.date_of_registration) as
REGISTERED_YEAR,ccd.card_id,ccd.issue_date
from customer_master cm join customer_card_details ccd on cm.customer_id=ccd.customer_id

```

order by cm.customer_name desc;

CUSTOMER_ID	CUSTOMER_NAME	REGISTERED_YEAR	CARD_ID	ISSUE_DATE
C00003	T RAMACHANDRAN	2012	CRD002	2012-11-02
C00005	SHIV PRASAD	2012	CRD003	2012-12-26
C00004	RAJIB MITRA	2012	CRD003	2012-11-21
C00001	NITIN	2012	CRD001	2012-10-15
C00002	AGNESH	2012	CRD002	2012-12-01

5. Please follow instructions given below.

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

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

2 rows

```
select cid.issue_id,cid.customer_id,cm.customer_name from customer_issue_details cid join
customer_master cm on cid.customer_id=cm.customer_id where cm.customer_name like 'R%'
and cid.actual_date_return>cid.return_date order by cm.customer_name;
```

ISSUE_ID	CUSTOMER_ID	CUSTOMER_NAME
I00008	C00010	RAGHAV SINGH
I00007	C00004	RAJIB MITRA

6. Please follow instructions given below.

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount.
Example Assume 500 is the amount then dollar value will be \$10.

Hint: Use AMOUNT_DOLLAR as alias name for amount in dollar.

Display the records in ascending order based on customer name.

```
SELECT ccd.customer_id, customer_name, ccd.card_id, description,concat('$',round(amount/52.42,0))  
AMOUNT_DOLLAR FROM customer_master cm INNER JOIN customer_card_details ccd ON  
cm.customer_id=ccd.customer_id INNER JOIN library_card_master lcm ON ccd.card_id=lcm.card_id  
INNER JOIN customer_issue_details cid ON cid.customer_id = cm.customer_id WHERE  
cm.date_of_registration=cid.issue_date order by customer_name;
```

CUSTOMER_ID	CUSTOMER_NAME	CARD_ID	DESCRIPTION	AMOUNT_DOLLAR
C00001	NITIN	CRD001	SILVER CARD	\$19
C00004	RAJIB MITRA	CRD003	PLATINUM CARD	\$57
C00003	T RAMACHANDRAN	CRD002	GOLD CARD	\$38

7. Please follow instructions given below.

Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use CUSTOMER_NAME as alias name for customer name.
Display the details sorted in ascending order based on customer name.

```
SELECT customer_id , upper(customer_name) CUSTOMER_NAME,contact_no,contact_address FROM  
customer_master WHERE customer_id NOT IN ( select customer_id from customer_card_details ) AND  
customer_id IN ( SELECT customer_id from customer_issue_details ) and contact_address like  
'%Nagar' order by customer_name ;
```

CUSTOMER_ID	CUSTOMER_NAME	CONTACT_NO	CONTACT_ADDRESS
C00010	RAGHAV SINGH	9675167890	A/6 NEHRU JAWAHAR NAGAR

8. Please follow instructions given below.

Write a query to display the movie id, movie name, release year, director name of movies acted by the lead actor 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 gmaster where
lead_actor_name1 in(select lead_actor_name1 from(select
lead_actor_name1, count(movie_id) ct from movies_master group by lead_actor_name1)t where
t.ct>=all(select count(movie_id) from movies_master
group by lead_actor_name1))order by movie_name;
```

| MOVIE_ID | MOVIE_NAME | RELEASE_YEAR | DIRECTOR_NAME     |
|----------|------------|--------------|-------------------|
| M00004   | INCEPTION  | 2010         | CHRISTOPHER NOLAN |
| M00011   | TITANIC    | 1997         | JAMES CAMERON     |

9. Please follow instructions given below.

<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;
```

| CUSTOMER_NAME  | MOVIE_COUNT |
|----------------|-------------|
| AGNESH         | 3           |
| AJAY GHOSH     | 0           |
| GEETHA REDDY   | 0           |
| NITIN          | 2           |
| RAGHAV SINGH   | 1           |
| RAJ SEKHANRAN  | 1           |
| RAJAN PILLAI   | 0           |
| RAJIB MITRA    | 4           |
| RIA NATRAJAN   | 0           |
| SHIV PRASAD    | 0           |
| T RAMACHANDRAN | 8           |

10. Please follow instructions given below.

Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

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

For Example Assume the issue id is I00005 then the serial number is 05

Hint: Alias name for serial number is 'SERIAL\_NO'

**19 rows**

```
select substring(cid.issue_id,5,2) as
SERIAL_NO,cid.issue_id,cid.customer_id,cm.customer_name,mm.movie_id,mm.movie_name
from customer_issue_details cid join customer_master cm on cm.customer_id=cid.customer_id
```

```
join movies_master mm on cid.movie_id=mm.movie_id group by
SERIAL_NO,cid.customer_id,mm.movie_id
```

```
order by SERIAL_NO;
```

| SERIAL_NO | ISSUE_ID | CUSTOMER_ID | CUSTOMER_NAME  | MOVIE_ID | MOVIE_NAME         |
|-----------|----------|-------------|----------------|----------|--------------------|
| 01        | I00001   | C00001      | NITIN          | M00001   | DIE HARD           |
| 02        | I00002   | C00002      | AGNESH         | M00002   | THE DARK KNIGHT    |
| 03        | I00003   | C00002      | AGNESH         | M00002   | THE DARK KNIGHT    |
| 04        | I00004   | C00003      | T RAMACHANDRAN | M00003   | THE MATRIX         |
| 05        | I00005   | C00003      | T RAMACHANDRAN | M00004   | INCEPTION          |
| 06        | I00006   | C00003      | T RAMACHANDRAN | M00005   | OFFICE SPACE       |
| 07        | I00007   | C00004      | RAJIB MITRA    | M00006   | YOUNG FRANKENSTEIN |
| 08        | I00008   | C00010      | RAGHAV SINGH   | M00008   | CASABLANCA         |
| 09        | I00009   | C00011      | RAJ SEKHANRAN  | M00010   | GONE WITH THE WIND |
| 10        | I00010   | C00004      | RAJIB MITRA    | M00007   | SHAUN OF THE DEAD  |
| 11        | I00011   | C00004      | RAJIB MITRA    | M00007   | SHAUN OF THE DEAD  |
| 12        | I00012   | C00001      | NITIN          | M00001   | DIE HARD           |
| 13        | I00013   | C00003      | T RAMACHANDRAN | M00001   | DIE HARD           |
| 14        | I00014   | C00003      | T RAMACHANDRAN | M00010   | GONE WITH THE WIND |
| 15        | I00015   | C00003      | T RAMACHANDRAN | M00011   | TITANIC            |
| 16        | I00016   | C00003      | T RAMACHANDRAN | M00011   | TITANIC            |
| 17        | I00017   | C00003      | T RAMACHANDRAN | M00008   | CASABLANCA         |

|    |        |        |             |        |                    |
|----|--------|--------|-------------|--------|--------------------|
| 18 | I00018 | C00002 | AGNESH      | M00010 | GONE WITH THE WIND |
| 19 | I00019 | C00004 | RAJIB MITRA | M00001 | DIE HARD           |
|    |        |        |             |        |                    |
|    |        |        |             |        |                    |

11. Please follow instructions given below.

Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

7 rows

```
select cid.issue_id,cid.issue_date,cid.customer_id,cm.customer_name,cm.contact_no
from customer_issue_details cid join customer_master cm on cid.customer_id=cm.customer_id
where year(issue_date)=2013 group by issue_id,issue_date,customer_id order by
issue_date desc;
```

| ISSUE_ID | ISSUE_DATE | CUSTOMER_ID | CUSTOMER_NAME  | CONTACT_NO |
|----------|------------|-------------|----------------|------------|
| I00012   | 2013-11-28 | C00001      | NITIN          | 9830354218 |
| I00017   | 2013-04-15 | C00003      | T RAMACHANDRAN | 9831289761 |
| I00009   | 2013-03-16 | C00011      | RAJ SEKHANRAN  | 8423178906 |
| I00016   | 2013-03-05 | C00003      | T RAMACHANDRAN | 9831289761 |
| I00008   | 2013-03-02 | C00010      | RAGHAV SINGH   | 9675167890 |
| I00015   | 2013-02-03 | C00003      | T RAMACHANDRAN | 9831289761 |
| I00014   | 2013-01-02 | C00003      | T RAMACHANDRAN | 9831289761 |

12. Please follow instructions given below.

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <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;
```

| MOVIE_ID | MOVIE_NAME   | ACTORS                        |
|----------|--------------|-------------------------------|
| M00009   | THE NOTEBOOK | RYAN GOSLING & RACHEL MCADAMS |

13. Please follow instructions given below.

Write a query to display the director's name, movie name and lead\_actor\_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR\_NAME as alias name for director name column Display the records sorted in ascending order based on director\_name and then by movie\_name in descending order.

2 rows

```
SELECT upper(director_name) DIRECTOR_NAME,movie_name,lead_actor_name1 FROM
movies_master WHERE director_name in (SELECT director_name FROM movies_master GROUP BY
director_name HAVING count(movie_id)>1) order by director_name, movie_name desc;
```

| DIRECTOR_NAME     | MOVIE_NAME      | LEAD_ACTOR_NAME1 |
|-------------------|-----------------|------------------|
| CHRISTOPHER NOLAN | THE DARK KNIGHT | CHRISTIAN BALE   |
| CHRISTOPHER       | INCEPTION       | LEONARDO         |

|       |  |          |
|-------|--|----------|
| NOLAN |  | DICAPRIO |
|-------|--|----------|

14. Please follow instructions given below.

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <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'
```

| NO_OF_CUSTOMERS |
|-----------------|
| 6               |

15. Please follow instructions given below.

Write a query to display the customer's name, contact number, library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT\_DETAILS as alias name for customer contact.

11 rows

```
select cm.customer_name,ifnull(cm.contact_no,cm.contact_add) as
CONTACT_DETAILS,lcd.card_id,lcd.description from customer_master cm
left join customer_card_details ccd on cm.customer_id=ccd.customer_id
left join library_card_master lcd on ccd.card_id=lcd.card_id group by
customer_name,description,CONTACT_DETAILS
order by customer_name;
```

| CUSTOMER_NAME | CONTACT_DETAILS | CARD_ID | DESCRIPTION |
|---------------|-----------------|---------|-------------|
| AGNESH        | 8923156781      | CRD002  | GOLD CARD   |

|                |                             |        |               |
|----------------|-----------------------------|--------|---------------|
| AJAY GHOSH     | 8763478901                  | NULL   | NULL          |
| GEETHA REDDY   | 8976167890                  | NULL   | NULL          |
| NITIN          | 9830354218                  | CRD001 | SILVER CARD   |
| RAGHAV SINGH   | 9675167890                  | NULL   | NULL          |
| RAJ SEKHANRAN  | 8423178906                  | NULL   | NULL          |
| RAJAN PILLAI   | A 1/66 KODAMBAKKAM          | NULL   | NULL          |
| RAJIB MITRA    | 9830356781                  | CRD003 | PLATINUM CARD |
| RIA NATRAJAN   | 9856723190                  | NULL   | NULL          |
| SHIV PRASAD    | 2/2 PHASE II, JAWAHAR NAGAR | CRD003 | PLATINUM CARD |
| T RAMACHANDRAN | 9831289761                  | CRD002 | GOLD CARD     |

16. Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in descending order. For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO\_OF\_TIMES as alias name for number of times

4 rows

```
select cm.customer_id, cm.customer_name, count(cid.movie_id) as NO_OF_TIMES from
customer_master
```

```
cm join customer_issue_details cid on cm.customer_id=cid.customer_id group by
customer_id, movie_id having
```

```
count(movie_id)>1 order by customer_name desc;
```

| CUSTOMER_ID | CUSTOMER_NAME  | NO_OF_TIMES |
|-------------|----------------|-------------|
| C00003      | T RAMACHANDRAN | 2           |
| C00004      | RAJIB MITRA    | 2           |
| C00001      | NITIN          | 2           |
| C00002      | AGNESH         | 2           |

17. Please follow instructions given below.

Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.

Hint: Use NO\_OF\_MOVIES as alias name for number of movies column.

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

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

5 rows

```
select cid.customer_id,cm.customer_name,
concat('+91-',substring(cm.contact_no,1,3),'-',substring(cm.contact_no,4,3),'-',
substring(cm.contact_no,7,4)) as CONTACT_ISD,
mm.movie_category,count(mm.movie_category) as NO_OF_MOVIES from customer_master
cm join customer_issue_details cid
on cm.customer_id=cid.customer_id join movies_master mm on cid.movie_id=mm.movie_id
group by mm.movie_category,cm.customer_name having count(movie_category)>1
order by cm.customer_name,mm.movie_category;
```

| CUSTOMER_ID | CUSTOMER_NAME  | CONTACT_ISD      | MOVIE_CATEGORY | NO_OF_MOVIES |
|-------------|----------------|------------------|----------------|--------------|
| C00003      | T RAMACHANDRAN | +91-876-456-2345 | COMEDY         | 2            |

|        |                |                  |         |   |
|--------|----------------|------------------|---------|---|
| C00002 | AGNESH         | +91-892-315-6781 | ACTION  | 2 |
| C00001 | NITIN          | +91-983-035-4218 | ACTION  | 2 |
| C00004 | RAJIB MITRA    | +91-983-035-6781 | COMEDY  | 3 |
| C00003 | T RAMACHANDRAN | +91-983-128-9761 | ACTION  | 3 |
| C00003 | T RAMACHANDRAN | +91-983-128-9761 | ROMANCE | 4 |

18. Please follow instructions given below.

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

3 rows

```
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) >= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by cm.customer_name)
union all
(select cm.customer_id,cm.customer_name from customer_master cm
```

```

join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) <= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by cm.customer_name)

```

| CUSTOMER_ID | CUSTOMER_NAME  |
|-------------|----------------|
| C00003      | T RAMACHANDRAN |
| C00010      | RAGHAV SINGH   |
| C00011      | RAJ SEKHANRAN  |

19. Please follow instructions given below.

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO\_OF\_TIMES as alias name

Display the records in ascending order based on customer name.

1 row

```

select cm.customer_id,cm.customer_name,count(mm.movie_id) as NO_OF_TIMES from
customer_master cm
join customer_issue_details cid on cm.customer_id=cid.customer_id join
movies_master mm on cid.movie_id=mm.movie_id where mm.movie_category='comedy' group by
customer_id
order by customer_name>1;

```

| CUSTOMER_ID | CUSTOMER_NAME | NO_OF_TIMES |
|-------------|---------------|-------------|
| C00004      | RAJIB MITRA   | 3           |

20. Please follow instructions given below.

Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL\_COST. Display the records sorted in ascending order based on customer id

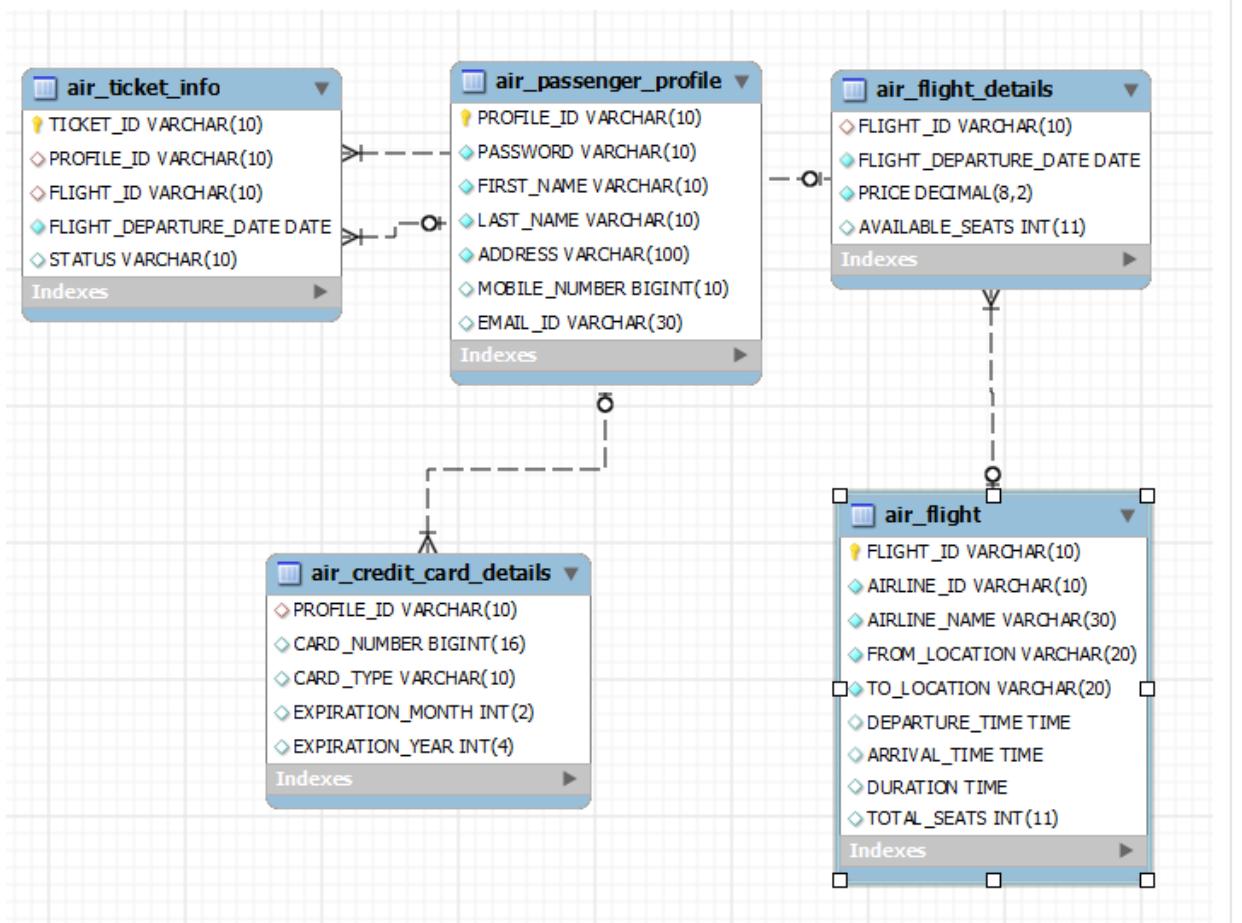
6 rows

```
select cid.customer_id,sum(mm.rental_cost) as TOTAL_COST from customer_issue_details cid
join movies_master mm
on cid.movie_id=mm.movie_id group by customer_id order by customer_id;
```

|   | customer_id | TOTAL_COST |
|---|-------------|------------|
| ▶ | C00001      | 200        |
|   | C00002      | 300        |
|   | C00003      | 1700       |
|   | C00004      | 400        |
|   | C00010      | 1000       |
|   | C00011      | 100        |

Airline Filght Management :

### ANSI SQL Airlines Management Schema



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

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

15 rows

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

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | MONTH_NAME | AVERAGE_PRICE |
|-----------|---------------|-------------|------------|---------------|
| 1011      | HYDERABAD     | CHENNAI     | APRIL      | 4614.000000   |
| 1011      | HYDERABAD     | CHENNAI     | MAY        | 3855.500000   |
| 1262      | HYDERABAD     | CHENNAI     | MAY        | 3444.500000   |
| 1265      | CHENNAI       | HYDERABAD   | APRIL      | 4086.000000   |
| 1265      | CHENNAI       | HYDERABAD   | MAY        | 3303.666667   |
| 289       | CHENNAI       | KOCHI       | MAY        | 3257.750000   |
| 3004      | BENGALURU     | CHENNAI     | MAY        | 3319.666667   |
| 3013      | CHENNAI       | BENGALURU   | MAY        | 3257.750000   |
| 3148      | CHENNAI       | BENGALURU   | JUNE       | 2773.000000   |
| 3148      | CHENNAI       | BENGALURU   | MAY        | 3052.000000   |
| 3241      | CHENNAI       | KOCHI       | MAY        | 3303.666667   |
| 3244      | KOCHI         | CHENNAI     | MAY        | 3371.500000   |
| 3307      | BENGALURU     | CHENNAI     | MAY        | 3309.000000   |
| 916       | CHENNAI       | HYDERABAD   | APRIL      | 4086.000000   |
| 916       | CHENNAI       | HYDERABAD   | MAY        | 3570.666667   |

2. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines.

The Query should display profile\_id, customer's first\_name, Address and Number of tickets booked as "No\_of\_Tickets"

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

1 row

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

| PROFILE_ID | FIRST_NAME | ADDRESS                    | NO_OF_TICKETS |
|------------|------------|----------------------------|---------------|
| PFL008     | GANESH     | 45 3RD<br>ST, HYDERABAD-24 | 1             |

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

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

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

9 rows

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

```
from_location,to_location,month_name;
```

| FROM_LOCATION | TO_LOCATION | MONTH_NAME | NO_OF_SERVICES |
|---------------|-------------|------------|----------------|
| BENGALURU     | CHENNAI     | MAY        | 7              |
| CHENNAI       | BENGALURU   | JUNE       | 1              |
| CHENNAI       | BENGALURU   | MAY        | 6              |
| CHENNAI       | HYDERABAD   | APRIL      | 2              |
| CHENNAI       | HYDERABAD   | MAY        | 6              |
| CHENNAI       | KOCHI       | MAY        | 7              |
| HYDERABAD     | CHENNAI     | APRIL      | 1              |
| HYDERABAD     | CHENNAI     | MAY        | 4              |
| KOCHI         | CHENNAI     | MAY        | 2              |

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

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

1 row

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

```

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

```

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

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

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

3 rows

```

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

```

| PROFILE_ID | FIRST_NAME | LAST_NAME | FLIGHT_ID | FLIGHT_DEPARTURE_DATE | NO_OF_TICKETS |
|------------|------------|-----------|-----------|-----------------------|---------------|
| PFL001     | LATHA      | SANKAR    | 1265      | 2013-04-29            | 1             |
| PFL004     | AARTHI     | RAMESH    | 1265      | 2013-05-29            | 1             |

|        |      |       |     |            |   |
|--------|------|-------|-----|------------|---|
| PFL005 | SIVA | KUMAR | 916 | 2013-05-06 | 2 |
|--------|------|-------|-----|------------|---|

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

3 rows

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

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

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

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

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

11 rows

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

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

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

Give an alias to the name as customer\_name.

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

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

3 rows

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

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

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

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

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

2 rows

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

| PROFILE_ID |
|------------|
| PFL002     |
| PFL007     |

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

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

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

7 rows

```

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

```

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | NO_OF_TICKETS |
|-----------|---------------|-------------|---------------|
| 1011      | HYDERABAD     | CHENNAI     | 4             |
| 1262      | HYDERABAD     | CHENNAI     | 1             |
| 1265      | CHENNAI       | HYDERABAD   | 2             |
| 3004      | BENGALURU     | CHENNAI     | 3             |
| 3148      | CHENNAI       | BENGALURU   | 7             |
| 3244      | KOCHI         | CHENNAI     | 7             |
| 916       | CHENNAI       | HYDERABAD   | 2             |

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

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

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

11 rows

```

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

```

| FLIGHT_ID | NO_OF_SERVICES | TOTAL_PRICE |
|-----------|----------------|-------------|
| 916       | 4              | 14798.00    |
| 1265      | 4              | 13997.00    |
| 3307      | 4              | 13236.00    |
| 3013      | 4              | 13031.00    |
| 289       | 4              | 13031.00    |
| 1011      | 3              | 12325.00    |
| 3004      | 3              | 9959.00     |
| 3241      | 3              | 9911.00     |
| 3148      | 3              | 8877.00     |
| 1262      | 2              | 6889.00     |
| 3244      | 2              | 6743.00     |

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

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

9 rows

```

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

flight\_departure\_date

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

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

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

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

1 row

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

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

| PROFILE_ID |
|------------|
| PFL008     |

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

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

4 rows

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

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

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

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

1 row

```
select distinct app.profile_id,app.first_name,app.last_name,ati.flight_id,ati.flight_departure_date,
```

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

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

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

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

| PROFILE_ID | FIRST_NAME | LAST_NAME | FLIGHT_ID | FLIGHT_DEPARTURE_DATE | DEPARTURE_TIME | ARRIVAL_TIME |
|------------|------------|-----------|-----------|-----------------------|----------------|--------------|
| PFL005     | SIVA       | KUMAR     | 916       | 2013-05-06            | 19:55:00       | 21:00:00     |

| DELAYED_DEPARTURE_TIME | DELAYED_ARRIVAL_TIME |
|------------------------|----------------------|
| 20:55:00               | 22:00:00             |

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

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

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

2 rows

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

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

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

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

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

11 rows

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

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | NO_OF_SERVICES |
|-----------|---------------|-------------|----------------|
| 1011      | HYDERABAD     | CHENNAI     | 2              |
| 1262      | HYDERABAD     | CHENNAI     | 2              |
| 1265      | CHENNAI       | HYDERABAD   | 3              |
| 289       | CHENNAI       | KOCHI       | 4              |
| 3004      | BENGALURU     | CHENNAI     | 3              |
| 3013      | CHENNAI       | BENGALURU   | 4              |

|      |           |           |   |
|------|-----------|-----------|---|
| 3148 | CHENNAI   | BENGALURU | 2 |
| 3241 | CHENNAI   | KOCHI     | 3 |
| 3244 | KOCHI     | CHENNAI   | 2 |
| 3307 | BENGALURU | CHENNAI   | 4 |
| 916  | CHENNAI   | HYDERABAD | 3 |

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

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

2 rows

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

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

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

1 row

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

|              |
|--------------|
| FLIGHT_COUNT |
| 3            |

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

2 rows

```
select distinct app.profile_id,app.first_name,app.email_id,app.mobile_number from
air_passenger_profile app
join air_ticket_info ati on app.profile_id=ati.profile_id
where ati.flight_id= 3148 group by app.first_name order by app.first_name;
```

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

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

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

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

11 rows

```
select flight_id,from_location,to_location,departure_time,
case when departure_time between '05:00:01' and '12:00:00' then 'Morning'
when departure_time between '12:00:01' and '18:00:00' then 'Afternoon'
```

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

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

end as Time\_of\_Service

from air\_flight order by flight\_id;

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

21. Please follow instructions given below.

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

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

36 rows

```

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

```

| FLIGHT_ID | FLIGHT_DEPARTURE_DATE | FLIGHT_TYPE         |
|-----------|-----------------------|---------------------|
| 1011      | 2013-04-30            | EXECUTIVE PASSENGER |
| 1011      | 2013-05-09            | EXECUTIVE PASSENGER |
| 1011      | 2013-05-21            | AIR BUS             |
| 1262      | 2013-05-20            | AIR BUS             |
| 1262      | 2013-05-29            | AIR BUS             |
| 1265      | 2013-04-29            | EXECUTIVE PASSENGER |
| 1265      | 2013-05-14            | AIR BUS             |
| 1265      | 2013-05-18            | EXECUTIVE PASSENGER |
| 1265      | 2013-05-29            | AIR PASSENGER       |
| 289       | 2013-05-06            | AIR BUS             |
| 289       | 2013-05-08            | AIR BUS             |
| 289       | 2013-05-20            | AIR BUS             |
| 289       | 2013-05-31            | AIR PASSENGER       |

|      |            |                        |
|------|------------|------------------------|
| 3004 | 2013-05-02 | AIR BUS                |
| 3004 | 2013-05-19 | AIR BUS                |
| 3004 | 2013-05-24 | AIR BUS                |
| 3013 | 2013-05-04 | AIR BUS                |
| 3013 | 2013-05-06 | AIR BUS                |
| 3013 | 2013-05-22 | AIR BUS                |
| 3013 | 2013-05-30 | AIR PASSENGER          |
| 3148 | 2013-05-16 | AIR BUS                |
| 3148 | 2013-05-21 | AIR BUS                |
| 3148 | 2013-06-01 | AIR PASSENGER          |
| 3241 | 2013-05-01 | EXECUTIVE<br>PASSENGER |
| 3241 | 2013-05-13 | AIR BUS                |
| 3241 | 2013-05-27 | AIR PASSENGER          |
| 3244 | 2013-05-03 | AIR BUS                |
| 3244 | 2013-05-15 | AIR BUS                |
| 3307 | 2013-05-03 | AIR BUS                |
| 3307 | 2013-05-03 | AIR BUS                |
| 3307 | 2013-05-23 | AIR BUS                |
| 3307 | 2013-05-29 | AIR BUS                |
| 916  | 2013-04-28 | EXECUTIVE              |

|     |            |                        |
|-----|------------|------------------------|
|     |            | PASSENGER              |
| 916 | 2013-05-01 | EXECUTIVE<br>PASSENGER |
| 916 | 2013-05-06 | AIR BUS                |
| 916 | 2013-05-12 | AIR BUS                |

22. Please follow instructions given below.

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

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

3 rows

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

| CARD_TYPE | CARD_COUNT |
|-----------|------------|
| GOLD      | 3          |
| INSTANT   | 2          |
| PLATINIUM | 3          |

23. Please follow instructions given below.

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

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

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

Display the records sorted in ascending order based on name.

6 rows

```
select substring(profile_id,4) as SERIAL_NO,first_name,mobile_number,email_id
```

```
from air_passenger_profile where email_id like '%gmail.com' order by first_name;
```

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

24. Please follow instructions given below.

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

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

4 rows

```
select af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) as
```

```
No_of_Services from air_flight af join air_flight_details afd on
```

```
af.flight_id=afd.flight_id where month(afd.flight_departure_date)='05' group by af.flight_id
```

```
having count(afd.flight_departure_date)
```

```
<= all (select count(afd.flight_departure_date) from air_flight af join air_flight_details afd on
```

```

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

order by af.flight_id;

```

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

25. Please follow instructions given below.

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

Hint: Get the distinct from location and to location.

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

4 rows

```

select distinct from_location, count(to_location) as No_of_Flights from
air_flight
group by from_location order by from_location;

```

| FROM_LOCATION | NO_OF_FLIGHTS |
|---------------|---------------|
| BENGALURU     | 2             |
| CHENNAI       | 6             |
| HYDERABAD     | 2             |
| KOCHI         | 1             |

**26.**Please follow instructions given below.

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

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

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

9 rows

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

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

**27.**Please follow instructions given below.

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

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

1 row

```
select af.flight_id,af.from_location,af.to_location,af.total_seats,(af.total_seats-afd.available_seats)
```

as No\_of\_Seats\_Booked from air\_flight af join air\_flight\_details afd on af.flight\_id=afd.flight\_id where (af.total\_seats-afd.available\_seats)>(af.total\_seats\*0.1) group by flight\_id order by flight\_id, No\_of\_Seats\_Booked;

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | TOTAL_SEATS | NO_OF_SEATS_BOOKED |
|-----------|---------------|-------------|-------------|--------------------|
| 3244      | KOCHI         | CHENNAI     | 50          | 7                  |

28. Please follow instructions given below.

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

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

14 rows

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

| FLIGHT_ID | FLIGHT_DEPARTURE_DATE | FROM_LOCATION | TO_LOCATION | DURATION |
|-----------|-----------------------|---------------|-------------|----------|
| 3013      | 2013-05-04            | CHENNAI       | BENGALURU   | 01:05:00 |
| 3013      | 2013-05-06            | CHENNAI       | BENGALURU   | 01:05:00 |
| 3013      | 2013-05-22            | CHENNAI       | BENGALURU   | 01:05:00 |
| 3013      | 2013-05-30            | CHENNAI       | BENGALURU   | 01:05:00 |

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

29. Please follow instructions given below.

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

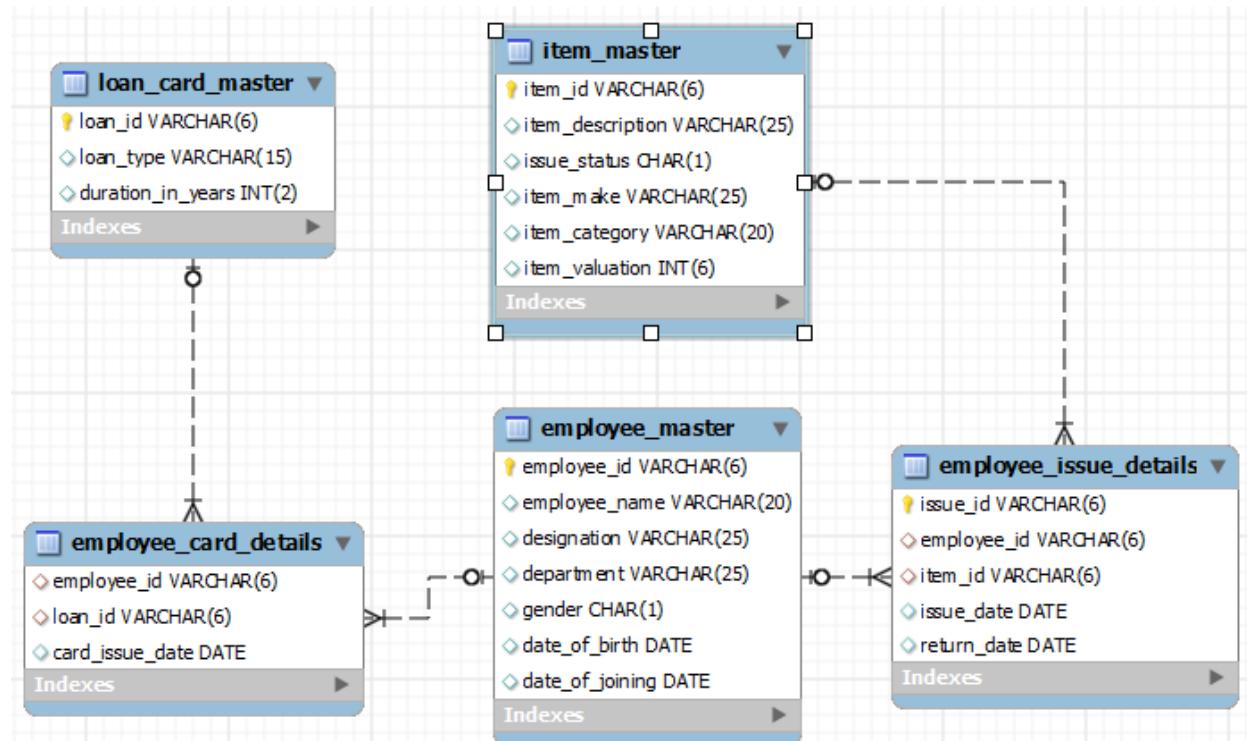
4 rows

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

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | NO_OF_SERVICES | AVERAGE    |
|-----------|---------------|-------------|----------------|------------|
| 1262      | HYDERABAD     | CHENNAI     | 2              | 3444.50000 |
| 1265      | CHENNAI       | HYDERABAD   | 4              | 3499.25000 |

|      |           |           |   |            |
|------|-----------|-----------|---|------------|
| 916  | CHENNAI   | HYDERABAD | 4 | 3699.50000 |
| 1011 | HYDERABAD | CHENNAI   | 3 | 4108.33333 |

ANSI SQL Item Loan Management System Schema



## Item Loan Database Queries

1. Please follow instructions given below.

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

3 rows

```

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

```

| Filter:       |                |
|---------------|----------------|
| item_category | Count_category |
| furniture     | 15             |
| Crockery      | 4              |
| Stationary    | 3              |

2. Please follow instructions given below.

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

1 row

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

| Filter:         |  |
|-----------------|--|
| No_of_Employees |  |
| 2               |  |

3. Please follow instructions given below.

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

1 row

```
select employee_id,employee_name,designation,department from employee_master
where employee_id
```

```
not in (select employee_id from employee_issue_details) order by employee_id;
```

|   | employee_id | employee_name | designation | department |
|---|-------------|---------------|-------------|------------|
| ▶ | E00005      | Radica        | Manager     | HR         |
| * | NULL        | NULL          | NULL        | NULL       |

4. Please follow instructions given below.

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

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

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

1 row

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

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00004      | Zuben         |

5. Please follow instructions given below.

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

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

9 rows

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

order by eid.issue\_id;

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

6. 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;
```

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00004      | Zuben         |
|   | E00005      | Radica        |
|   | E00006      | John          |
| * | NULL        | NULL          |

7. Please follow instructions given below.

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

1 row

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

| Filter: |             |
|---------|-------------|
|         | No_of_Cards |
| ▶       | 3           |

8. Please follow instructions given below.

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

1 row

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

| Filter: |                  |
|---------|------------------|
|         | Count_Stationary |
| ▶       | 3                |

9. Please follow instructions given below.

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

5 rows

```
select em.employee_id,em.employee_name,count(eid.item_id) as Count from employee_master em
join
employee_issue_details eid on em.employee_id=eid.employee_id group by em.employee_id having
```

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

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

10. Please follow instructions given below.

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

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

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

2 rows

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

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00002      | Abhay         |
|   | E00003      | Anita         |

11. Please follow instructions given below.

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

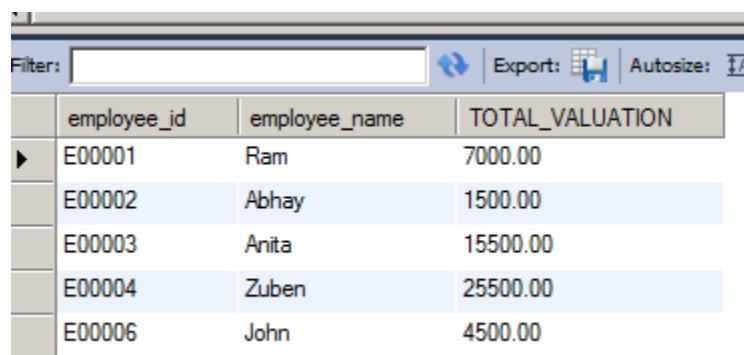
Display the records sorted in ascending order based on employee id.

Consider only employees who have been issued atleast 1 item.

5 rows

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

join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id group by em.employee_id having count(im.item_valuation)>=1
order by em.employee_id;
```



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

12. Please follow instructions given below.

Write a query to display distinct employee id, employee name who kept the item issued for more than a year.

Hint: Use Date time function to calculate the difference between item issue and return date.

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

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

5 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id where datediff(return_date,issue_date)>365 order by
```

employee\_id;

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

13. Please follow instructions given below.

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

Display the records sorted in ascending order on employee id.

2 rows

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

join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id where item_category='furniture' group by employee_id having
count(COUNT_ITEMS)>1 order by employee_id;
```

|   | employee_id | employee_name | COUNT_ITEMS |
|---|-------------|---------------|-------------|
| ▶ | E00001      | Ram           | 2           |
|   | E00003      | Anita         | 3           |

14. Please follow instructions given below.

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

2 rows

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

| Filter: |        |                 |
|---------|--------|-----------------|
|         | gender | No_of_Employees |
| ▶       | F      | 2               |
|         | M      | 4               |

```
gender order by gender;
```

15. Please follow instructions given below.

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

3 rows

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

| Filter: |             |               |
|---------|-------------|---------------|
|         | employee_id | employee_name |
| E00002  | Abhay       |               |
| E00003  | Anita       |               |
| E00006  | John        |               |

```
order by employee_id;
```

16. Please follow instructions given below.

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

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

2 rows

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

```
issue_status;
```

|   | issue_status | No_of_Furnitures |
|---|--------------|------------------|
| ▶ | N            | 6                |
|   | Y            | 9                |

17. Please follow instructions given below.

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

16 rows

```
select item_category,item_make,item_description,count(item_id) as No_of_Items from
item_master im group by item_category,item_make,item_description order by
item_category,item_make,item_description;
```

|   | item_category | item_make | item_description | No_of_Items |
|---|---------------|-----------|------------------|-------------|
| ▶ | Crockery      | Bonechina | Dining Set       | 1           |
|   | Crockery      | Bonechina | Tea Set          | 1           |
|   | Crockery      | Glass     | Dining Set       | 1           |
|   | Crockery      | Glass     | Tea Set          | 1           |
|   | furniture     | Steel     | Cupboard         | 2           |
|   | furniture     | Steel     | Side Table       | 1           |
|   | furniture     | Steel     | Single Bed       | 2           |
|   | furniture     | Steel     | Tea Table        | 2           |
|   | furniture     | Wooden    | Dining Chair     | 1           |
|   | furniture     | Wooden    | Dining Table     | 1           |
|   | furniture     | Wooden    | Double Bed       | 2           |
|   | furniture     | Wooden    | Side Table       | 1           |
|   | furniture     | Wooden    | Sofa             | 1           |
|   | furniture     | Wooden    | Tea Table        | 2           |
|   | Stationary    | Plastic   | Pen              | 2           |
|   | Stationary    | Wooden    | Pencil           | 1           |

18. Please follow instructions given below.

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

1 row

```
select em.employee_id,em.employee_name,im.item_id,im.item_description from employee_master em
join
employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
eid.item_id=im.item_id where year(eid.issue_date)=2013 and month(eid.issue_date)=01 order by
```

The screenshot shows a database query results window with a grid. The grid has four columns with headers: 'employee\_id', 'employee\_name', 'item\_id', and 'item\_description'. There is one data row with values: E00002, Abhay, I00005, and Side Table.

|   | employee_id | employee_name | item_id | item_description |
|---|-------------|---------------|---------|------------------|
| ▶ | E00002      | Abhay         | I00005  | Side Table       |

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;
```

Filter:  Export: Autosize:

|   | employee_id | employee_name | COUNT_CATEGORY |
|---|-------------|---------------|----------------|
| ▶ | E00004      | Zuben         | 2              |

20. Please follow instructions given below.

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

14 rows

```
select item_id,item_description from item_master where item_id not in (select item_id
from employee_issue_details) order by item_id;
```

Filter:

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

21. Please follow instructions given below.

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

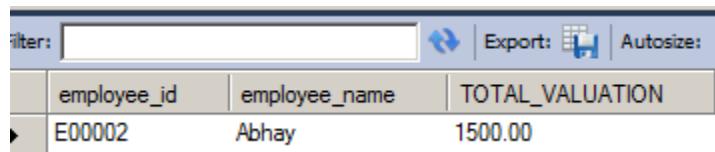
[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

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;
```



| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00002      | Abhay         | 1500.00         |

22. Please follow instructions given below.

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

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

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

```
SELECT ecd.employee_id,employee_name,
card_issue_date, if(lcd.duration_in_years=0, 'NO-VALIDITY DATE', date_add(ec.card_issue_date,
interval duration_in_years year)) as CARD_VALIDITY_DATE
FROM employee_master em INNER JOIN
employee_card_details ecd
```

```

ON em.employee_id=ecd.employee_id

INNER JOIN loan_card_master lcd

ON ecd.loan_id=lcd.loan_id

order by employee_name, CARD_VALID_DATE;

```

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

23. Please follow instructions given below.

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

3 rows

```

select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid on
em.employee_id=eid.employee_id where em.employee_id not in
(select employee_id from employee_issue_details where year(issue_date)=2013)
order by employee_id;

```

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00001      | Ram           |
|   | E00003      | Anita         |
|   | E00006      | John          |

24. Please follow instructions given below.

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

9 rows

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

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

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

<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);;
```

| Filter: |             |               | Export:         | Autosize: |
|---------|-------------|---------------|-----------------|-----------|
|         | employee_id | employee_name | TOTAL_VALUATION |           |
|         | E00004      | Zuben         | 25500.00        |           |

# BANK

```
create database bank;
```

```
use bank;
```

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

---

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

```
CREATE TABLE account_master
(account_number VARCHAR(255),
customer_number VARCHAR(255),
branch_id VARCHAR(255),
opening_balance INT(20),
account_opening_date DATE,
account_type VARCHAR(10),
account_status VARCHAR(10),
PRIMARY KEY (account_number),
FOREIGN KEY (customer_number) references customer_master(customer_number),
FOREIGN KEY (branch_id) references branch_master(branch_id));
```

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

---

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

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

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

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

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

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

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

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

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

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

insert into customer_master values('C00010','SHANKAR',NULL, 'NAIR', 'CHENNAI', '8765489076',
 'SERVICE', '1976-07-12');

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

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

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

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

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

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

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

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

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

---

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

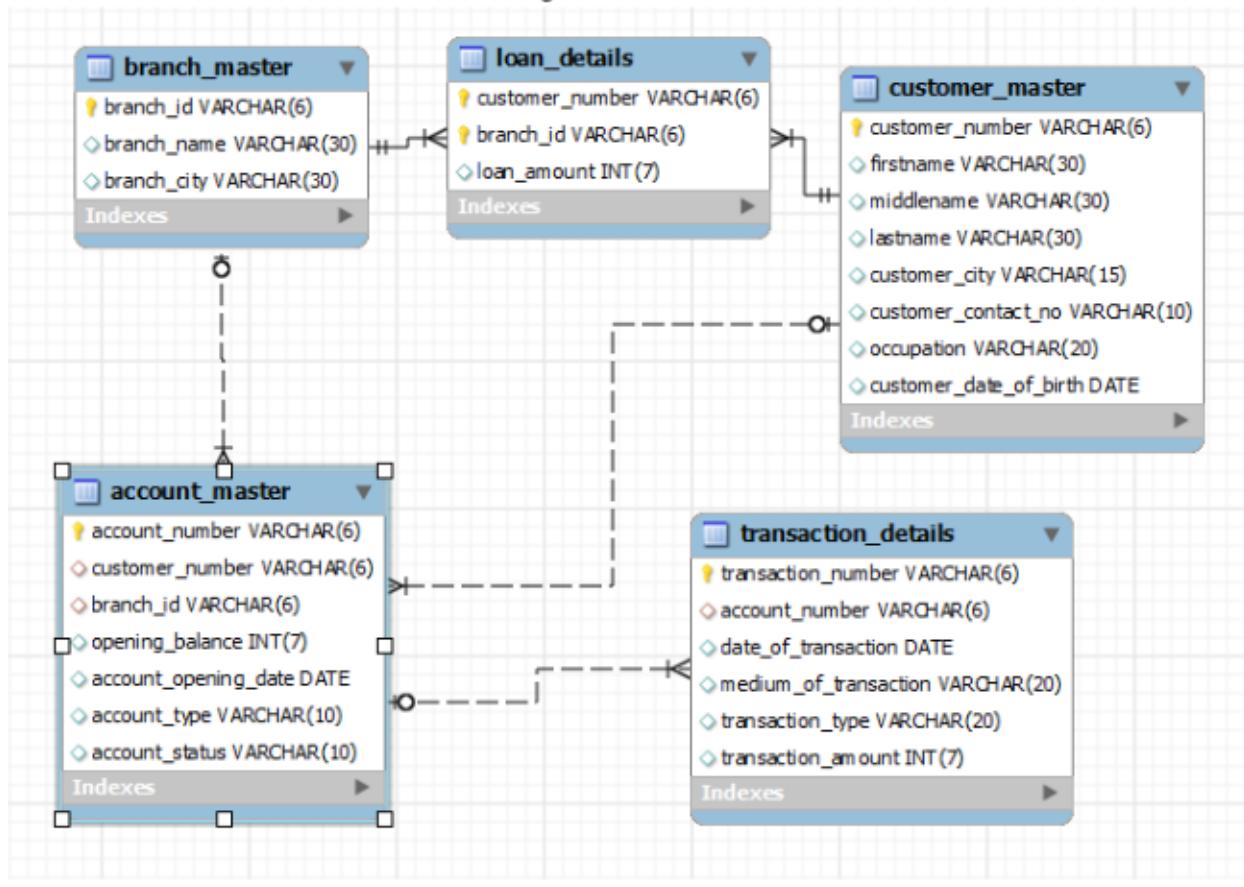
insert into transaction_details values('T00011', 'A00002', '2013-03-25', 'CASH'
 , 'WITHDRAWAL' ,7000);

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

insert into Loan_details values('C00001', 'B00001', 100000);
insert into Loan_details values('C00002', 'B00002', 200000);
insert into Loan_details values('C00009', 'B00008', 400000);
insert into Loan_details values('C00010', 'B00009', 500000);
insert into Loan_details values('C00001', 'B00003', 600000);
insert into Loan_details values('C00002', 'B00001', 600000);
```

---

ANSI SQL Bank Management Schema



## CUSTOMER MASTER

## ACCOUNT MASTER

| account_number | customer_number | branch_id | opening_balance | account_opening_date | account_type | account_status |
|----------------|-----------------|-----------|-----------------|----------------------|--------------|----------------|
| A00001         | C00001          | B00001    | 1000            | 2012-12-15           | SAVING       | ACTIVE         |
| A00002         | C00002          | B00001    | 1000            | 2012-06-12           | SAVING       | ACTIVE         |
| A00003         | C00003          | B00002    | 1000            | 2012-05-17           | SAVING       | ACTIVE         |
| A00004         | C00002          | B00005    | 1000            | 2013-01-27           | SAVING       | ACTIVE         |
| A00005         | C00006          | B00006    | 1000            | 2012-12-17           | SAVING       | ACTIVE         |
| A00006         | C00007          | B00007    | 1000            | 2010-08-12           | SAVING       | SUSPENDED      |
| A00007         | C00007          | B00001    | 1000            | 2012-10-02           | SAVING       | ACTIVE         |
| A00008         | C00001          | B00003    | 1000            | 2009-11-09           | SAVING       | TERMINATED     |
| A00009         | C00003          | B00007    | 1000            | 2008-11-30           | SAVING       | TERMINATED     |
| A00010         | C00004          | B00002    | 1000            | 2013-03-01           | SAVING       | ACTIVE         |
| NULL           | NULL            | NULL      | NULL            | NULL                 | NULL         | NULL           |

## BRANCH MASTER

| branch_id | branch_name           | branch_city |
|-----------|-----------------------|-------------|
| B00001    | ASAFA LI ROAD         | DELHI       |
| B00002    | NEW DELHI MAIN BRANCH | DELHI       |
| B00003    | DELHI CANTT           | DELHI       |
| B00004    | JASOLA                | DELHI       |
| B00005    | MAHIM                 | MUMBAI      |
| B00006    | VILE PARLE            | MUMBAI      |
| B00007    | MANDVI                | MUMBAI      |
| B00008    | JADAVPUR              | KOLKATA     |
| B00009    | KODAMBAKKAM           | CHENNAI     |
| NULL      | NULL                  | NULL        |

## LOAN DETAILS

| customer_number | branch_id | loan_amount |
|-----------------|-----------|-------------|
| C00001          | B00001    | 100000      |
| C00002          | B00002    | 200000      |
| C00009          | B00008    | 400000      |
| C00010          | B00009    | 500000      |
| C00001          | B00003    | 600000      |
| C00002          | B00001    | 600000      |

## TRANSACTION DETAILS

| transaction_number | account_number | date_of_transaction | medium_of_transaction | transaction_type | transaction_amount |
|--------------------|----------------|---------------------|-----------------------|------------------|--------------------|
| T00001             | A00001         | 2013-01-01          | CHEQUE                | DEPOSIT          | 2000               |
| T00002             | A00001         | 2013-02-01          | CASH                  | WITHDRAWAL       | 1000               |
| T00003             | A00002         | 2013-01-01          | CASH                  | DEPOSIT          | 2000               |
| T00004             | A00002         | 2013-02-01          | CASH                  | DEPOSIT          | 3000               |
| T00005             | A00007         | 2013-01-11          | CASH                  | DEPOSIT          | 7000               |
| T00006             | A00007         | 2013-01-13          | CASH                  | DEPOSIT          | 9000               |
| T00007             | A00001         | 2013-03-13          | CASH                  | DEPOSIT          | 4000               |
| T00008             | A00001         | 2013-03-14          | CHEQUE                | DEPOSIT          | 3000               |
| T00009             | A00001         | 2013-03-21          | CASH                  | WITHDRAWAL       | 9000               |
| T00010             | A00001         | 2013-03-22          | CASH                  | WITHDRAWAL       | 2000               |
| T00011             | A00002         | 2013-03-25          | CASH                  | WITHDRAWAL       | 7000               |
| T00012             | A00007         | 2013-03-26          | CASH                  | WITHDRAWAL       | 2000               |
| NULL               | NULL           | NULL                | NULL                  | NULL             | NULL               |

## QUERIES

1. Write a query to display account number, customer's number, customer's firstname, lastname, account opening date. Display the records sorted in ascending order based on account number.

```
SELECT a.account_number,c.customer_number,c.firstname,c.lastname,a.account_number
FROM customer_master c JOIN account_master a ON
c.customer_number=a.customer_number
ORDER BY a.account_number;
```

| account_number | customer_number | firstname | lastname | account_opening_date |
|----------------|-----------------|-----------|----------|----------------------|
| A00001         | C00001          | RAMESH    | SHARMA   | 2012-12-15           |
| A00002         | C00002          | AVINASH   | MINHA    | 2012-06-12           |
| A00003         | C00003          | RAHUL     | RASTOGI  | 2012-05-17           |
| A00004         | C00002          | AVINASH   | MINHA    | 2013-01-27           |
| A00005         | C00006          | CHITRESH  | BARWE    | 2012-12-17           |
| A00006         | C00007          | AMIT      | BORKAR   | 2010-08-12           |
| A00007         | C00007          | AMIT      | BORKAR   | 2012-10-02           |
| A00008         | C00001          | RAMESH    | SHARMA   | 2009-11-09           |
| A00009         | C00003          | RAHUL     | RASTOGI  | 2008-11-30           |
| A00010         | C00004          | PARUL     | GANDHI   | 2013-03-01           |

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

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

| cust_count |
|------------|
| 4          |

3. Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number FROM account_master a join
customer_master c ON c.customer_number=a.customer_number WHERE
day(a.account_opening_date)>'15' ORDER BY c.customer_number,a.account_number;
```

| customer_number | firstname | account_number |
|-----------------|-----------|----------------|
| C00002          | AVINASH   | A00004         |
| C00003          | RAHUL     | A00003         |
| C00003          | RAHUL     | A00009         |
| C00006          | CHITRESH  | A00005         |

**4. Write a query to display customer number, customer's first name, account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.**

```
SELECT c.customer_number,c.firstname,a.account_number
FROM account_master a JOIN customer_master c
ON c.customer_number=a.customer_number
WHERE a.account_status='Terminated'
ORDER BY c.customer_number,a.account_number;
```

| customer_number | firstname | account_number |
|-----------------|-----------|----------------|
| C00001          | RAMESH    | A00008         |
| C00003          | RAHUL     | A00009         |

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

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

| transaction_type | Trans_count |
|------------------|-------------|
| DEPOSIT          | 3           |
| WITHDRAWAL       | 3           |

**6. Write a query to display the number of customers who have registration but no account in the bank. Give the alias name as Count\_Customer for number of customers.**

```
SELECT count(customer_number) Count_Customer FROM customer_master
```

```
WHERE customer_number NOT IN (SELECT customer_number FROM account_master);
```

|                |
|----------------|
| Count_Customer |
|----------------|

|   |
|---|
| 4 |
|---|

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

```
SELECT a.account_number,a.opening_balance+sum(t.transaction_amount)
```

```
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
```

```
WHERE t.transaction_type='Deposit' GROUP BY t.account_number;
```

| account_number | Deposit_Amount |
|----------------|----------------|
| A00001         | 10000          |
| A00002         | 6000           |
| A00007         | 17000          |

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

```
SELECT branch.branch_city, count(account.account_number) No_of_Accounts
```

```
FROM branch_master LEFT JOIN account_master
```

```
ON account.branch_id=branch.branch_id
```

```
GROUP BY branch.branch_city ORDER BY branch_city;
```

| branch_city | No_of_accounts |
|-------------|----------------|
| CHENNAI     | 0              |
| DELHI       | 6              |
| KOLKATA     | 0              |
| MUMBAI      | 4              |

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

```
SELECT c.firstname FROM
customer_master c JOIN account_master a ON a.customer_number=c.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1;
```

firstname

AMIT

AVINASH

RAHUL

RAMESH

**10. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.**

```
SELECT c.customer_number,c.firstname,c.lastname FROM
customer_master c JOIN loan_details l ON c.customer_number=l.customer_number
GROUP BY l.customer_number HAVING count(l.branch_id)>1
ORDER BY c.customer_number;
```

| customer_number | firstname | lastname |
|-----------------|-----------|----------|
| C00001          | RAMESH    | SHARMA   |
| C00002          | AVINASH   | MINHA    |

**11. Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different. Display the records sorted in ascending order based on customer number.**

```
SELECT c.customer_number,c.firstname,c.customer_city,b.branch_city FROM
Customer_master c JOIN Account_master a ON c.customer_number=a.customer_number
JOIN Branch_master b ON b.branch_id=a.branch_id
WHERE b.branch_city<>c.customer_city
ORDER BY c.customer_number;
```

| customer_number | firstname | customer_city | branch_city |
|-----------------|-----------|---------------|-------------|
| C00002          | AVINASH   | DELHI         | MUMBAI      |
| C00003          | RAHUL     | DELHI         | MUMBAI      |
| C00007          | AMIT      | MUMBAI        | DELHI       |

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

```
SELECT count(c.customer_number)Count FROM customer_master c JOIN loan_details l
ON c.customer_number=l.customer_number
WHERE c.customer_number NOT IN (SELECT customer_number FROM account_master);
```

|       |
|-------|
| Count |
| 2     |

13. Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT account_number FROM transaction_details
GROUP BY account_number
HAVING count(transaction_number)>=ALL
(SELECT count(transaction_number) FROM transaction_details
GROUP BY account_number) ORDER BY account_number;
```

|                |
|----------------|
| account_number |
| A00001         |

14. Write a query to show the branch name,branch city where we have the maximum customers. For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
SELECT b.branch_name,b.branch_city FROM
Branch_master b JOIN account a ON a.branch_id=b.branch_id
GROUP BY a.branch_id HAVING count(a.customer_number)>=ALL
(SELECT count(customer_number) FROM
Account_master GROUP BY branch_id)
ORDER BY b.branch_name;
```

| branch_name   | branch_city |
|---------------|-------------|
| ASAF ALI ROAD | DELHI       |

15. Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well. For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011. Display the records sorted in ascending order based on account number.

```

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

```

(or)

```

SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d Deposit,ifnull(t1.w,0) Withdrawal FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.w,0)>t2.d
ORDER BY account_number;

```

| account_number | Deposit | Withdrawal |
|----------------|---------|------------|
| A00001         | 10000   | 12000      |
| A00002         | 6000    | 7000       |

**16. Write a query to show the balance amount for account number that ends with 001. Note: Balance amount includes account opening balance also. Give alias name as Balance\_Amount. For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000 ) – (500+500+1000).**

```

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

```

**(or)**

```

SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d-ifnull(t1.w,0) Balance_Amount FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.account_number,t2.account_number) LIKE '%001'
ORDER BY account_number;

```

| account_number | Balance_Amount |
|----------------|----------------|
| A00001         | -2000          |

**17. Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as**

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

```
SELECT c.customer_number,c.firstname,t.account_number, count(t.account_number) Count_Trans
FROM transaction_details t JOIN account_master a ON a.account_number=t.account_number
JOIN customer c ON c.customer_number=a.customer_number
GROUP BY t.account_number ORDER BY c.customer_number, a.account_number;
```

| customer_number | firstname | account_number | Count_Trans |
|-----------------|-----------|----------------|-------------|
| C00001          | RAMESH    | A00001         | 6           |
| C00002          | AVINASH   | A00002         | 3           |
| C00007          | AMIT      | A00007         | 3           |

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

```
SELECT c.firstname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1
ORDER BY c.firstname;
```

| firstname |
|-----------|
| AMIT      |
| AVINASH   |
| RAHUL     |
| RAMESH    |

**19.** Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches. For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011. So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT Id.customer_number,firstname,lastname
FROM customer_master cm JOIN loan_details Id
ON cm.customer_number=Id.customer_number
GROUP BY customer_number
HAVING count(branch_id)>=2 AND sum(loan_amount)>=
```

```
ALL(SELECT sum(loan_amount) FROM loan GROUP BY customer_number);
```

| customer_number | firstname | lastname |
|-----------------|-----------|----------|
| C00002          | AVINASH   | MINHA    |

20. Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans. Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT c.customer_number,c.firstname,l.branch_id,l.loan_amount FROM
Customer_master c JOIN loan_details l ON c.customer_number=l.customer_number
ORDER BY c.customer_number,l.branch_id,l.loan_amount;
```

| customer_number | firstname | branch_id | loan_amount |
|-----------------|-----------|-----------|-------------|
| C00001          | RAMESH    | B00001    | 100000      |
| C00001          | RAMESH    | B00003    | 600000      |
| C00002          | AVINASH   | B00001    | 600000      |
| C00002          | AVINASH   | B00002    | 200000      |
| C00009          | ABHISHEK  | B00008    | 400000      |
| C00010          | SHANKAR   | B00009    | 500000      |

21. Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count\_Branch. Display the records sorted in ascending order based on city name.

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

| branch_city | Count_Branch |
|-------------|--------------|
| CHENNAI     | 1            |
| DELHI       | 4            |
| KOLKATA     | 1            |
| MUMBAI      | 3            |

22. Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active. Display the records sorted in ascending order based on account id /account number.

```
SELECT a.account_number,c.firstname,c.lastname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number and
a.account_status='Active'
```

```
ORDER BY a.account_number;
```

| account_number | firstname | lastname |
|----------------|-----------|----------|
| A00001         | RAMESH    | SHARMA   |
| A00002         | AVINASH   | MINHA    |
| A00003         | RAHUL     | RASTOGI  |
| A00004         | AVINASH   | MINHA    |
| A00005         | CHITRESH  | BARWE    |
| A00007         | AMIT      | BORKAR   |
| A00010         | PARUL     | GANDHI   |

23. Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle\_Name. Display the records sorted in ascending order based on customer number.

```
SELECT customer_number,firstname,ifnull(middlename,lastname) Middle_name FROM
```

```
Customer_master ORDER BY customer_number;
```

| customer_number | firstname | Middle_name |
|-----------------|-----------|-------------|
| C00001          | RAMESH    | CHANDRA     |
| C00002          | AVINASH   | SUNDER      |
| C00003          | RAHUL     | NULL        |
| C00004          | PARUL     | NULL        |
| C00005          | NAVEEN    | CHANDRA     |
| C00006          | CHITRESH  | NULL        |
| C00007          | AMIT      | KUMAR       |
| C00008          | NISHA     | DAMLE       |
| C00009          | ABHISHEK  | DUTTA       |
| C00010          | SHANKAR   | NAIR        |

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

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

```
Customer_master ORDER BY year(customer_date_of_birth),customer_number;
```

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

25. Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student. Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT c.firstname,c.customer_city,a.account_number FROM
Customer_master c JOIN account_master a ON a.customer_number=c.customer_number
WHERE c.occupation NOT IN ('Service','Student','Business')
ORDER BY c.firstname,a.account_number;
```

| firstname | customer_city | account_number |
|-----------|---------------|----------------|
| PARUL     | DELHI         | A00010         |

# AIRLINES

```
create database flight;
```

```
use flight;
```

```
CREATE TABLE air_credit_card_details
(
profile_id VARCHAR(10) NOT NULL,
```

```
card_number BIGINT,
card_type VARCHAR(45),
expiration_month INT,
expiration_year INT
);
```

```
CREATE TABLE air_passenger_profile
(
profile_id VARCHAR(10) NOT NULL,
password VARCHAR(45) NULL,
first_name VARCHAR(45) NULL,
last_name VARCHAR(45) NULL,
address VARCHAR(45) NULL,
mobile_number BIGINT NULL,
email_id VARCHAR(45) NULL
);
```

```
CREATE TABLE air_ticket_info
(
ticket_id VARCHAR(45) NOT NULL,
profile_id VARCHAR(10) NULL,
flight_id VARCHAR(45) NULL,
flight_departure_date DATE NULL,
status VARCHAR(45) NULL
);
```

```
CREATE TABLE air_flight_details
(
flight_id VARCHAR(45) NOT NULL,
```

```
flight_departure_date DATE NULL ,
price DECIMAL(10,2) NULL ,
available_seats INT NULL
);
```

```
CREATE TABLE air_flight
(
flight_id VARCHAR(45) NOT NULL ,
airline_id VARCHAR(45) NULL ,
airline_name VARCHAR(45) NULL ,
from_location VARCHAR(45) NULL ,
to_location VARCHAR(45) NULL ,
departure_time TIME NULL ,
arrival_time TIME NULL ,
duration TIME NULL ,
total_seats INT NULL
);
```

```
INSERT INTO air_credit_card_details VALUES
(1, 622098761234, 'debit', 5, 2013),
(2, 652362563625, 'credit', 1, 2013),
(1, 765432345678, 'credit', 2, 2013),
(3, 654378561234, 'debit', 6, 2013),
(4, 625417895623, 'debit', 2, 2013),
(5, 865478956325, 'debit', 3, 2013),
(6, 789563521457, 'credit', 4, 2013),
(2, 543267895432, 'credit', 8, 2013),
(1, 256369856321, 'debit', 1, 2013);
```

INSERT INTO air\_flight VALUES

```
(3173, 'MH370', 'abc', 'hyderabad', 'chennai', '06:30:00', '07:15:00',
 '0:45:00', 100),
(3178, 'MH17', 'def', 'chennai', 'hyderabad', '08:00:00', '09:00:00',
 '1:00:00', 200),
(3172, 'AR342', 'fgh', 'kolkata', 'chennai', '11:30:00', '13:00:00',
 '1:30:00', 100),
(3071, 'JT564', 'jkl', 'chennai', 'delhi', '08:00:00', '10:00:00', '2:00:00',
 100),
(3170, 'DT345', 'xyz', 'delhi', 'kolkata', '21:00:00', '22:30:00', '1:30:00',
 100),
(3175, 'MJ654', 'abc', 'chennai', 'hyderabad', '15:00:00', '16:00:00',
 '1:00:00', 200),
(3176, 'MH370', 'def', 'kochi', 'chennai', '18:00:00', '19:05:00', '1:05:00',
 100),
(3177, 'MH45', 'fgh', 'delhi', 'kochi', '19:00:00', '21:00:00', '2:00:00',
 200),
(3174, 'MH321', 'xyz', 'kolkata', 'delhi', '0:00:00', '2:00:00', '2:00:00',
 100),
(3179, 'JT435', 'abc', 'chennai', 'kolkata', '14:00:00', '15:00:00', '1:00:00',
 100),
(3180, 'JT456', 'ijk', 'kolkata', 'kochi', '5:00:00', '5:45:00', '0:45:00',
 200);
```

INSERT INTO air\_flight\_details VALUES

```
(3170, '2013-02-14', 1000, 10),
(3171, '2013-03-15', 5000, 0),
(3172, '2013-02-05', 3000, 32),
(3173, '2013-04-07', 2000, 12),
(3174, '2013-04-05', 3800, 3),
(3175, '2013-05-25', 3500, 10),
(3176, '2013-03-14', 8000, 2),
```

```
(3177, '2013-06-15', 1500, 0),
(3178, '2013-05-06', 3000, 5),
(3179, '2013-04-03', 4000, 15),
(3180, '2013-04-02', 3000, 14);
```

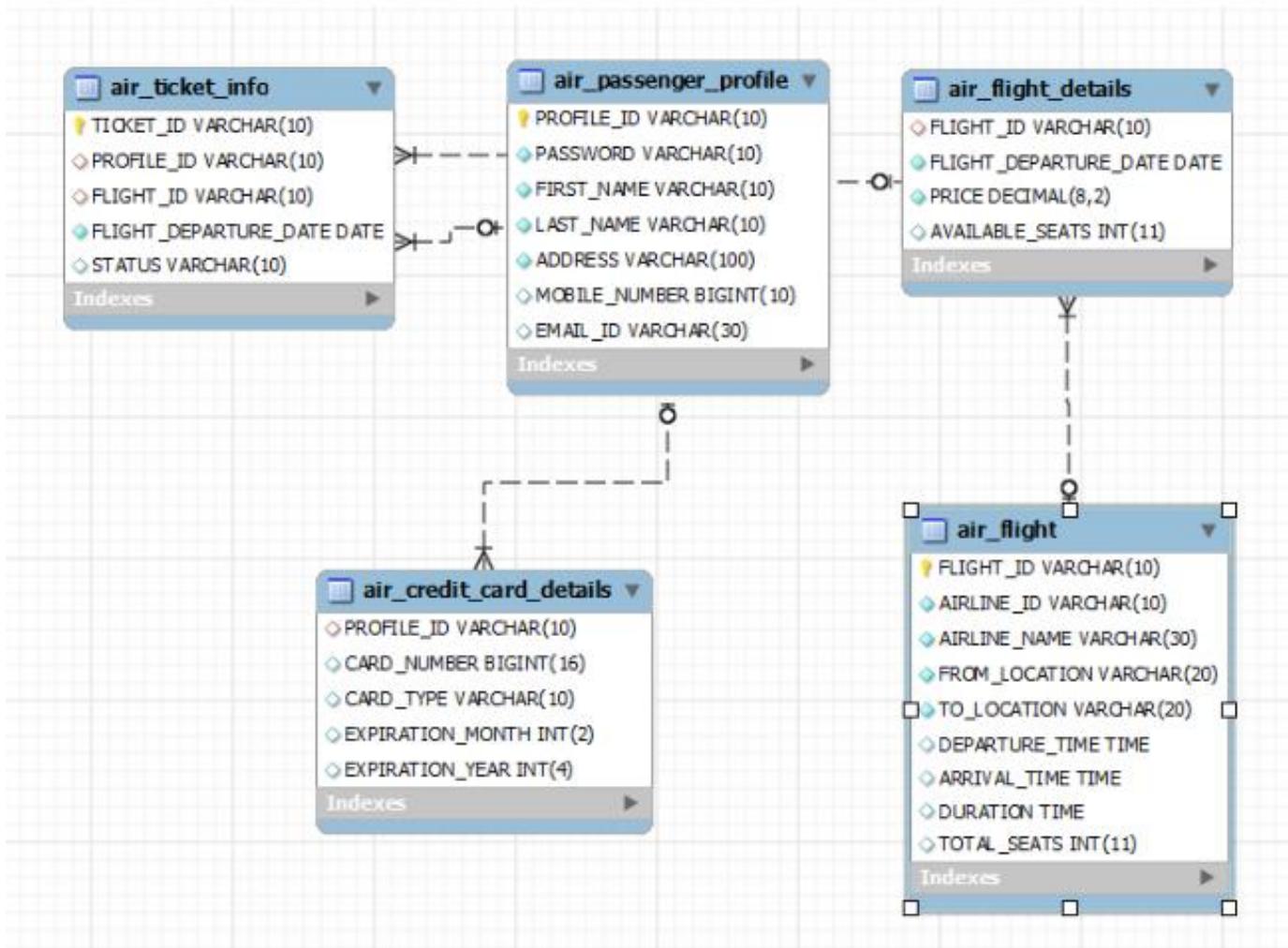
```
INSERT INTO air_ticket_info VALUES
```

```
(1, 1, 3178, '2013-05-06', 'delayed'),
(2, 5, 3179, '2013-04-03', 'on time'),
(2, 4, 3180, '2013-04-02', 'on time'),
(1, 2, 3177, '2013-06-15', 'on time'),
(1, 3, 3176, '2013-03-14', 'on time'),
(3, 1, 3171, '2013-03-15', 'on time'),
(4, 4, 3172, '2013-02-06', 'delayed'),
(5, 2, 3178, '2013-06-05', 'on time'),
(4, 3, 3171, '2013-03-15', 'on time'),
(5, 1, 3175, '2013-05-25', 'on time'),
(6, 3, 3177, '2013-06-15', 'on time');
```

```
INSERT INTO air_passenger_profile VALUES
```

```
(1, 'godbless', 'John', 'Stuart', 'Street 21, Near Bus Stop-Hyderabad-432126',
9865263251, 'john@gmail.com'),
(2, 'heyyaa', 'Robert', 'Clive', 'Sector 3, Technopolis-Kolkata-700102',
9733015875, 'robert@yahoo.com'),
(3, 'hello123', 'Raj', 'Sharma', 'House No. 3, Anna Nagar-Kochi-452314',
9775470232, 'raj3452@hotmail.com'),
(4, 'yesboss', 'Sanjay', 'Mittal', '21 Cauunaught Place-Delhi-144985',
9856856321, 'sanjay@yahoo.com'),
```

```
(5, 'imhere', 'Tony', 'Stark', '51A, Greams Lane-Chennai-144587',
9832015785, 'tony@gmail.com');
```



## AIR TICKET INFO

| ticket_id | profile_id | flight_id | flight_departure_date | status  |
|-----------|------------|-----------|-----------------------|---------|
| 1         | 1          | 3178      | 2013-05-06            | delayed |
| 2         | 5          | 3179      | 2013-04-03            | on time |
| 2         | 4          | 3180      | 2013-04-02            | on time |
| 1         | 2          | 3177      | 2013-06-15            | on time |
| 1         | 3          | 3176      | 2013-03-14            | on time |
| 3         | 1          | 3171      | 2013-03-15            | on time |
| 4         | 4          | 3172      | 2013-02-06            | delayed |
| 5         | 2          | 3178      | 2013-06-05            | on time |
| 4         | 3          | 3171      | 2013-03-15            | on time |
| 5         | 1          | 3175      | 2013-05-25            | on time |
| 6         | 3          | 3177      | 2013-06-15            | on time |

## AIR PASSENGER DETAILS

| profile_id | password | first_name | last_name | address                                   | mobile_number | email_id           |
|------------|----------|------------|-----------|-------------------------------------------|---------------|--------------------|
| 1          | godbless | John       | Stuart    | Street 21, Near Bus Stop-Hyderabad-432126 | 9865263251    | john@gmail.com     |
| 2          | heyyaa   | Robert     | Clive     | Sector 3, Technopolis-Kolkata-700102      | 9733015875    | robert@yahoo.com   |
| 3          | hello123 | Raj        | Sharma    | House No. 3, Anna Nagar-Kochi-452314      | 9775470232    | raj3452@hotmail... |
| 4          | yesboss  | Sanjay     | Mittal    | 21 Cauunaught Place-Delhi-144985          | 9856856321    | sanjay@yahoo.c...  |
| 5          | imhere   | Tony       | Stark     | 51A, Greams Lane-Chennai-144587           | 9832015785    | tony@gmail.com     |

## AIR FLIGHT DETAILS

| flight_id | flight_departure_date | price   | available_seats |
|-----------|-----------------------|---------|-----------------|
| 3170      | 2013-02-14            | 1000.00 | 10              |
| 3171      | 2013-03-15            | 5000.00 | 0               |
| 3172      | 2013-02-05            | 3000.00 | 32              |
| 3173      | 2013-04-07            | 2000.00 | 12              |
| 3174      | 2013-04-05            | 3800.00 | 3               |
| 3175      | 2013-05-25            | 3500.00 | 10              |
| 3176      | 2013-03-14            | 8000.00 | 2               |
| 3177      | 2013-06-15            | 1500.00 | 0               |
| 3178      | 2013-05-06            | 3000.00 | 5               |
| 3179      | 2013-04-03            | 4000.00 | 15              |
| 3180      | 2013-04-02            | 3000.00 | 14              |

## AIR CREDIT CARD DETAILS

| profile_id | card_number  | card_type | expiration_month | expiration_year |
|------------|--------------|-----------|------------------|-----------------|
| 1          | 622098761234 | debit     | 5                | 2013            |
| 2          | 652362563625 | credit    | 1                | 2013            |
| 1          | 765432345678 | credit    | 2                | 2013            |
| 3          | 654378561234 | debit     | 6                | 2013            |
| 4          | 625417895623 | debit     | 2                | 2013            |
| 5          | 865478956325 | debit     | 3                | 2013            |
| 6          | 789563521457 | credit    | 4                | 2013            |
| 2          | 543267895432 | credit    | 8                | 2013            |
| 1          | 256369856321 | debit     | 1                | 2013            |

## AIR FLIGHT

| flight_id | airline_id | airline_name | from_location | to_location | departure_time | arrival_time | duration | total_seats |
|-----------|------------|--------------|---------------|-------------|----------------|--------------|----------|-------------|
| 3170      | DT345      | xyz          | delhi         | kolkata     | 21:00:00       | 22:30:00     | 01:30:00 | 100         |
| 3171      | JT564      | JKL          | chennai       | delhi       | 08:00:00       | 10:00:00     | 02:00:00 | 100         |
| 3172      | AR342      | fgh          | kolkata       | chennai     | 11:30:00       | 13:00:00     | 01:30:00 | 100         |
| 3173      | MH370      | abc          | hyderabad     | chennai     | 06:30:00       | 07:15:00     | 00:45:00 | 100         |
| 3174      | MH321      | xyz          | kolkata       | delhi       | 00:00:00       | 02:00:00     | 02:00:00 | 100         |
| 3175      | MJ654      | abc          | chennai       | hyderabad   | 15:00:00       | 16:00:00     | 01:00:00 | 200         |
| 3176      | MH370      | def          | kochi         | chennai     | 18:00:00       | 19:05:00     | 01:05:00 | 100         |
| 3177      | MH45       | fgh          | delhi         | kochi       | 19:00:00       | 21:00:00     | 02:00:00 | 200         |
| 3178      | MH17       | def          | chennai       | hyderabad   | 08:00:00       | 09:00:00     | 01:00:00 | 200         |
| 3179      | JT435      | abc          | chennai       | kolkata     | 14:00:00       | 15:00:00     | 01:00:00 | 100         |
| 3180      | JT456      | ijk          | kolkata       | kochi       | 05:00:00       | 05:45:00     | 00:45:00 | 200         |

## QUERIES

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight\_Id,From\_Location,To\_Location,Month Name as "Month\_Name" and average price as "Average\_Price". Display the records sorted in ascending order based on flight id and then by Month Name.

```
SELECT f.flight_id,f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
AVG(price) Average_Price FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id WHERE f.airline_name = 'abc'
```

GROUP BY f.flight\_id,f.from\_location,f.to\_location,Month\_Name

ORDER BY f.flight\_id, Month\_Name;

| flight_id | from_location | to_location | Month_Name | Average_Price |
|-----------|---------------|-------------|------------|---------------|
| 3173      | hyderabad     | chennai     | April      | 2000.000000   |
| 3175      | chennai       | hyderabad   | May        | 3500.000000   |
| 3179      | chennai       | kolkata     | April      | 4000.000000   |

2. Write a query to display the number of flight services between locations in a month. The Query should display From\_Location, To\_Location, Month as "Month\_Name" and number of flight services as "No\_of\_Services". Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight. The records should be displayed in ascending order based on From\_Location and then by To\_Location and then by month name.

```
SELECT f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
count(af.flight_departure_date) No_of_Services
FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id
GROUP BY f.from_location,f.to_location,Month_Name
ORDER BY f.from_location,f.to_Location,Month_Name;
```

| from_location | to_location | Month_Name | No_of_Services |
|---------------|-------------|------------|----------------|
| chennai       | delhi       | March      | 1              |
| chennai       | hyderabad   | May        | 2              |
| chennai       | kolkata     | April      | 1              |
| delhi         | kochi       | June       | 1              |
| delhi         | kolkata     | February   | 1              |
| hyderabad     | chennai     | April      | 1              |
| kochi         | chennai     | March      | 1              |
| kolkata       | chennai     | February   | 1              |
| kolkata       | delhi       | April      | 1              |
| kolkata       | kochi       | April      | 1              |

3. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile\_id, customer's first\_name, Address and Number of tickets

**booked as “No\_of\_Tickets”Display the records sorted in ascending order based on customer's first name.**

```
SELECT ap.profile_id,ap.first_name,ap.address,count(ati.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id and af.airline_name='abc'
GROUP BY ap.profile_id,ap.first_name,ap.address HAVING count(ati.ticket_id)<=ALL
(SELECT count(ticket_id)
FROM air_ticket_info GROUP BY profile_id)
ORDER BY ap.first_name;
```

| profile_id | first_name | address                                   | No_of_Tickets |
|------------|------------|-------------------------------------------|---------------|
| 1          | John       | Street 21, Near Bus Stop-Hyderabad-432126 | 1             |
| 5          | Tony       | 51A, Greams Lane-Chennai-144587           | 1             |

**4. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile\_id,first\_name,last\_name, Flight\_Id , Departure\_Date and number of tickets booked as “No\_of\_Tickets”.Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.**

```
SELECT ap.profile_id,ap.first_name,ap.last_name,af.flight_id,ati.flight_departure_date,
count(ati.profile_id) No_of_Tickets FROM
air_ticket_info ati JOIN air_passenger_profile ap ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.flight_id,ati.profile_id
ORDER BY ap.profile_id,af.flight_id,ati.flight_departure_date;
```

| profile_id | first_name | last_name | flight_id | flight_departure_date | No_of_Tickets |
|------------|------------|-----------|-----------|-----------------------|---------------|
| 1          | John       | Stuart    | 3175      | 2013-05-25            | 1             |
| 1          | John       | Stuart    | 3178      | 2013-05-06            | 1             |
| 2          | Robert     | Clive     | 3178      | 2013-06-05            | 1             |

**5. Write a query to display flight id,from location, to location and ticket price of flights whose departure is in the month of april.Display the records sorted in ascending order based on flight id and then by from location.**

```
SELECT af.flight_id,af.from_location,af.to_location,afd.price FROM
```

```

air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
and month(afd.flight_departure_date)='04'
ORDER BY af.flight_id,af.from_location;

```

| flight_id | from_location | to_location | price   |
|-----------|---------------|-------------|---------|
| 3173      | hyderabad     | chennai     | 2000.00 |
| 3174      | kolkata       | delhi       | 3800.00 |
| 3179      | chennai       | kolkata     | 4000.00 |
| 3180      | kolkata       | kochi       | 3000.00 |

6. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight\_id, from\_location, to\_location and Average price as “Price”. Display the records sorted in ascending order based on flight id and then by from\_location and then by to\_location.

```

SELECT af.flight_id,af.from_location,af.to_location,avg(afd.price) Average_Price FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id
ORDER BY af.flight_id,af.from_location,af.to_location;

```

| flight_id | from_location | to_location | Average_Price |
|-----------|---------------|-------------|---------------|
| 3170      | delhi         | kolkata     | 1000.000000   |
| 3171      | chennai       | delhi       | 5000.000000   |
| 3172      | kolkata       | chennai     | 3000.000000   |
| 3173      | hyderabad     | chennai     | 2000.000000   |
| 3174      | kolkata       | delhi       | 3800.000000   |
| 3175      | chennai       | hyderabad   | 3500.000000   |
| 3176      | kochi         | chennai     | 8000.000000   |
| 3177      | delhi         | kochi       | 1500.000000   |
| 3178      | chennai       | hyderabad   | 3000.000000   |
| 3179      | chennai       | kolkata     | 4000.000000   |
| 3180      | kolkata       | kochi       | 3000.000000   |

7. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile\_id, customer\_name (combine first\_name & last\_name with comma in

**b/w), address of the customer. Give an alias to the name as customer\_name.Hint: Query should fetch unique customers irrespective of multiple tickets booked.Display the records sorted in ascending order based on profile id.**

```
SELECT ap.profile_id,concat(ap.first_name,',',ap.last_name) customer_name,ap.address FROM air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id JOIN air_flight af ON af.flight_id=ati.flight_id WHERE af.from_location='Chennai' and af.to_location='Hyderabad' GROUP BY ati.profile_id ORDER BY ap.profile_id;
```

| profile_id | Customer_name | address                                   |
|------------|---------------|-------------------------------------------|
| 1          | John,Stuart   | Street 21, Near Bus Stop-Hyderabad-432126 |
| 2          | Robert,Clive  | Sector 3, Technopolis-Kolkata-700102      |

**8. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets.In case of multiple records, display the records sorted in ascending order based on profile id.**

```
SELECT profile_id FROM air_ticket_info
group by profile_id
having count(ticket_id)>=all(select count(ticket_id)
from air_ticket_info
group by profile_id) order by profile_id;
```

| profile_id |
|------------|
| 1          |
| 3          |

**9. Write a query to display the total number of tickets as “No\_of\_Tickets” booked in each flight in ABC Airlines. The Query should display the flight\_id, from\_location, to\_location and the number of tickets. Display only the flights in which atleast 1 ticket is booked.Display the records sorted in ascending order based on flight id.**

```
SELECT f.flight_id,f.from_location,f.to_location,COUNT(t.ticket_id) AS No_of_Tickets
FROM air_ticket_info t JOIN air_flight f
ON f.flight_id = t.flight_id where AIRLINE_NAME = 'abc' GROUP by
f.flight_id,f.from_location,f.to_location
```

having count(t.ticket\_id)>=1

ORDER by f.flight\_id;

| flight_id | from_location | to_location | No_of_Tickets |
|-----------|---------------|-------------|---------------|
| 3175      | chennai       | hyderabad   | 1             |
| 3179      | chennai       | kolkata     | 1             |

**10.** Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight\_id, number of services as “No\_of\_Services” and the cost as “Total\_Price” in the same order. Order the result by Total Price in descending order and then by flight\_id in descending order. Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

```
SELECT flight_id,count(flight_departure_date) No_of_services,sum(price) Total_Price FROM
air_flight_details GROUP BY flight_id

ORDER BY Total_price DESC,flight_id DESC;
```

| flight_id | No_of_services | Total_Price |
|-----------|----------------|-------------|
| 3176      | 1              | 8000.00     |
| 3171      | 1              | 5000.00     |
| 3179      | 1              | 4000.00     |
| 3174      | 1              | 3800.00     |
| 3175      | 1              | 3500.00     |
| 3180      | 1              | 3000.00     |
| 3178      | 1              | 3000.00     |
| 3172      | 1              | 3000.00     |
| 3173      | 1              | 2000.00     |
| 3177      | 1              | 1500.00     |
| 3170      | 1              | 1000.00     |

**11.** Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight\_id, flight\_departure\_date and the number of passengers as “No\_of\_Passengers” in the same order. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT flight_id,flight_departure_date,count(ticket_id) No_of_passengerS FROM
air_ticket_info GROUP BY flight_id,flight_departure_date

ORDER BY flight_id,flight_departure_date;
```

| flight_id | flight_departure_date | No_of_passengers |
|-----------|-----------------------|------------------|
| 3171      | 2013-03-15            | 2                |
| 3172      | 2013-02-06            | 1                |
| 3175      | 2013-05-25            | 1                |
| 3176      | 2013-03-14            | 1                |
| 3177      | 2013-06-15            | 2                |
| 3178      | 2013-05-06            | 1                |
| 3178      | 2013-06-05            | 1                |
| 3179      | 2013-04-03            | 1                |
| 3180      | 2013-04-02            | 1                |

**12. Write a query to display profile id of passenger(s) who booked minimum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.**

```
SELECT profile_id FROM air_ticket_info
GROUP BY profile_id HAVING count(ticket_id)<=ALL
(SELECT count(ticket_id) FROM air_ticket_info GROUP BY profile_id)
ORDER BY profile_id;
```

| profile_id |
|------------|
| 5          |

**13. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI. Display the records sorted in ascending order based on profile id.**

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.mobile_number,ap.email_id FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON ati.flight_id=af.flight_id
WHERE af.from_location='Hyderabad' and af.to_location='Chennai'
ORDER BY profile_id;
```

| profile_id | first_name | mobile_number | email_id |
|------------|------------|---------------|----------|
|            |            |               |          |

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

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.last_name,ati.flight_id,ati.flight_departure_date,
af.departure_time,af.arrival_time,
addtime(af.departure_time,'01:00:00') Delayed_Departure_Time,
addtime(af.arrival_time,'01:00:00') Delayed_Arrival_Time FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
and ati.flight_departure_date='2013-05-06'
ORDER BY profile_id;
```

| profile_id | first_name | last_name | flight_id | flight_departure_date | departure_time | arrival_time | Delayed_Departure_Time | Delayed_Arrival_Time |
|------------|------------|-----------|-----------|-----------------------|----------------|--------------|------------------------|----------------------|
| 1          | John       | Stuart    | 3178      | 2013-05-06            | 08:00:00       | 09:00:00     | 09:00:00               | 10:00:00             |

**15. Write a query to display the number of tickets as "No\_of\_Tickets" booked by Kochi Customers. The Query should display the Profile\_Id, First\_Name, Base\_Location and number of tickets booked. Hint: Use String functions to get the base location of customer from their Address and give alias name as "Base\_Location" Display the records sorted in ascending order based on customer first name.**

```
SELECT ap.profile_id,ap.first_name,
substring_index(substring_index(ap.address,'-',2),'-',1) Base_Location,
count(ati.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ati.profile_id=ap.profile_id
WHERE ap.address LIKE '%Kochi%'
ORDER BY ap.first_name;
```

| profile_id | first_name | Base_Location | No_of_Tickets |
|------------|------------|---------------|---------------|
| 3          | Raj        | Kochi         | 3             |

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

```
SELECT af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) No_of_services
FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
WHERE month(flight_departure_date)='05'
GROUP BY af.flight_id,af.from_location,af.to_location
ORDER BY af.flight_id;
```

| flight_id | from_location | to_location | No_of_services |
|-----------|---------------|-------------|----------------|
| 3175      | chennai       | hyderabad   | 1              |
| 3178      | chennai       | hyderabad   | 1              |

17. Write a query to display profile id, last name, mobile number and email id of passengers whose base location is chennai. Display the records sorted in ascending order based on profile id.

```
SELECT profile_id, last_name, mobile_number, email_id
FROM air_passenger_profile
WHERE address LIKE '%Chennai%'
ORDER BY profile_id;
```

| profile_id | last_name | mobile_number | email_id       |
|------------|-----------|---------------|----------------|
| 5          | Stark     | 9832015785    | tony@gmail.com |

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

```
SELECT count(flight_id) FLIGHT_COUNT FROM air_flight
WHERE from_location='CHENNAI'
AND departure_time BETWEEN '06:00:00' AND '18:00:00';
```

| FLIGHT_COUNT |
|--------------|
| 4            |

**19. Write a query to display unique profile id,first name , email id and contact number of passenger(s) who travelled on flight with id 3178. Display the records sorted in ascending order based on first name.**

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.email_id,ap.mobile_number FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
WHERE ati.flight_id='3178'
ORDER BY ap.first_name;
```

| profile_id | first_name | email_id         | mobile_number |
|------------|------------|------------------|---------------|
| 1          | John       | john@gmail.com   | 9865263251    |
| 2          | Robert     | robert@yahoo.com | 9733015875    |

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

```
SELECT flight_id,flight_departure_date,
case when price<3000 then 'AIR PASSENGER'
 when price>=3000 and price<4000 then 'AIR BUS'
 when price>=4000 then 'EXECUTIVE PASSENGER'
end FLIGHT_TYPE FROM air_flight_details
ORDER BY flight_id,flight_departure_date;
```

| flight_id | flight_departure_date | FLIGHT_TYPE         |
|-----------|-----------------------|---------------------|
| 3170      | 2013-02-14            | AIR PASSENGER       |
| 3171      | 2013-03-15            | EXECUTIVE PASSENGER |
| 3172      | 2013-02-05            | AIR BUS             |
| 3173      | 2013-04-07            | AIR PASSENGER       |
| 3174      | 2013-04-05            | AIR BUS             |
| 3175      | 2013-05-25            | AIR BUS             |
| 3176      | 2013-03-14            | EXECUTIVE PASSENGER |
| 3177      | 2013-06-15            | AIR PASSENGER       |
| 3178      | 2013-05-06            | AIR BUS             |
| 3179      | 2013-04-03            | EXECUTIVE PASSENGER |
| 3180      | 2013-04-02            | AIR BUS             |

**21. Write a query to display the credit card type and no of credit cards used on the same type. Display the records sorted in ascending order based on credit card type.** Hint: Use CARD\_COUNT AS Alias name for no of cards.

```
SELECT card_type, count(card_type) Card_Count FROM air_credit_card_details
GROUP BY card_type ORDER BY card_type;
```

| card_type | Card_Count |
|-----------|------------|
| credit    | 4          |
| debit     | 5          |

**22. Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com. The Serial No will be the last three digits of profile ID.** Hint: Use SERIAL\_NO as Alias name for serial number. Display the records sorted in ascending order based on name.

```
SELECT substring(profile_id,-3) SERIAL_NO,first_name,mobile_number,email_id FROM
air_passenger_profile
WHERE email_id LIKE '%@gmail.com'
ORDER BY first_name;
```

| SERIAL_NO | first_name | mobile_number | email_id       |
|-----------|------------|---------------|----------------|
|           | John       | 9865263251    | john@gmail.com |
|           | Tony       | 9832015785    | tony@gmail.com |

**23. Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight\_id, from\_location, to\_location, least number of Services as “No\_of\_Services”** Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight if there are multiple flights, display them sorted in ascending order based on flight id.

```
SELECT afd.flight_id,af.from_location,af.to_location,count(afd.flight_id) No_of_Services
FROM air_flight_details afd JOIN air_flight af ON af.flight_id=afd.flight_id
WHERE monthname(afd.flight_departure_date)='May'
GROUP BY afd.flight_departure_date HAVING count(afd.flight_id) <=
ALL(SELECT count(flight_id) FROM air_flight_details
WHERE monthname(flight_departure_date)='May'
GROUP BY flight_departure_date)
```

ORDER BY flight\_id;

| flight_id | from_location | to_location | No_of_Services |
|-----------|---------------|-------------|----------------|
| 3175      | chennai       | hyderabad   | 1              |
| 3178      | chennai       | hyderabad   | 1              |

24. Write a query to display the flights available in Morning, AfterNoon, Evening& Night. The Query should display the Flight\_Id, From\_Location, To\_Location , Departure\_Time, time of service as "Time\_of\_Service". Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning, 12:00:01 to 18:00:00 Hrs -AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - NightDisplay the records sorted in ascending order based on flight id.

```
SELECT flight_id,from_location,to_location,Departure_Time,
CASE
WHEN departure_time BETWEEN ('05:00:01') AND ('12:00:00')
THEN 'Morning'
WHEN departure_time BETWEEN ('12:00:01') AND ('18:00:00')
THEN 'AfterNoon'
WHEN departure_time BETWEEN ('18:00:01') AND ('24:00:00')
THEN 'Evening'
WHEN departure_time='00:00:00'
THEN 'Evening'
WHEN departure_time BETWEEN ('00:00:01') AND ('05:00:00')
THEN 'Night'
END Time_of_Service
FROM air_flight
order by flight_id;
```

| flight_id | from_location | to_location | Departure_Time | Time_of_Service |
|-----------|---------------|-------------|----------------|-----------------|
| 3170      | delhi         | kolkata     | 21:00:00       | Evening         |
| 3171      | chennai       | delhi       | 08:00:00       | Morning         |
| 3172      | kolkata       | chennai     | 11:30:00       | Morning         |
| 3173      | hyderabad     | chennai     | 06:30:00       | Morning         |
| 3174      | kolkata       | delhi       | 00:00:00       | Evening         |
| 3175      | chennai       | hyderabad   | 15:00:00       | AfterNoon       |
| 3176      | kochi         | chennai     | 18:00:00       | AfterNoon       |
| 3177      | delhi         | kochi       | 19:00:00       | Evening         |
| 3178      | chennai       | hyderabad   | 08:00:00       | Morning         |
| 3179      | chennai       | kolkata     | 14:00:00       | AfterNoon       |
| 3180      | kolkata       | kochi       | 05:00:00       | Night           |

25. Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No\_of\_Flights”. Hint: Get the distinct from location and to location. Display the records sorted in ascending order based on from location.

```
SELECT from_location, count(flight_id) No_of_Flights FROM
air_flight GROUP BY from_location
ORDER BY from_location;
```

| from_location | No_of_Flights |
|---------------|---------------|
| chennai       | 4             |
| delhi         | 2             |
| hyderabad     | 1             |
| kochi         | 1             |
| kolkata       | 3             |

26. Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight\_id, from\_location, To\_location, flight\_departure\_date and the number of passengers as “No\_of\_Passengers”. Hint: The Number of passengers inclusive of all the tickets booked with single profile id. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT af.flight_id, af.from_location, af.to_location, ati.flight_departure_date,
count(ati.ticket_id) No_of_Passengers FROM
air_flight af JOIN air_ticket_info ati ON af.flight_id=ati.flight_id
GROUP BY af.flight_id, af.from_location, af.to_location, ati.flight_departure_date
```

ORDER BY af.flight\_id,ati.flight\_departure\_date;

| flight_id | from_location | to_location | flight_departure_date | No_of_Passengers |
|-----------|---------------|-------------|-----------------------|------------------|
| 3171      | chennai       | delhi       | 2013-03-15            | 2                |
| 3172      | kolkata       | chennai     | 2013-02-06            | 1                |
| 3175      | chennai       | hyderabad   | 2013-05-25            | 1                |
| 3176      | kochi         | chennai     | 2013-03-14            | 1                |
| 3177      | delhi         | kochi       | 2013-06-15            | 2                |
| 3178      | chennai       | hyderabad   | 2013-05-06            | 1                |
| 3178      | chennai       | hyderabad   | 2013-06-05            | 1                |
| 3179      | chennai       | kolkata     | 2013-04-03            | 1                |
| 3180      | kolkata       | kochi       | 2013-04-02            | 1                |

27. Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight\_Id, From\_Location, To\_Location, Total\_Seats, seats booked as "No\_of\_Seats\_Booked". Display the records sorted in ascending order based on flight id and then by No\_of\_Seats\_Booked.

```
SELECT af.flight_id,af.from_location,af.to_location,af.total_seats,
(af.total_seats-afd.available_seats) No_of_Seats_Booked FROM
air_flight_details afd JOIN air_flight af ON afd.flight_id=af.flight_id
WHERE (af.total_seats-afd.available_seats)>(af.total_seats*0.1)
ORDER BY flight_id,No_of_Seats_Booked;
```

| flight_id | from_location | to_location | total_seats | No_of_Seats_Booked |
|-----------|---------------|-------------|-------------|--------------------|
| 3170      | delhi         | kolkata     | 100         | 90                 |
| 3171      | chennai       | delhi       | 100         | 100                |
| 3172      | kolkata       | chennai     | 100         | 68                 |
| 3173      | hyderabad     | chennai     | 100         | 88                 |
| 3174      | kolkata       | delhi       | 100         | 97                 |
| 3175      | chennai       | hyderabad   | 200         | 190                |
| 3176      | kochi         | chennai     | 100         | 98                 |
| 3177      | delhi         | kochi       | 200         | 200                |
| 3178      | chennai       | hyderabad   | 200         | 195                |
| 3179      | chennai       | kolkata     | 100         | 85                 |
| 3180      | kolkata       | kochi       | 200         | 186                |

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

```
SELECT af.flight_id,afd.flight_Departure_Date,af.From_Location,af.To_Location,af.duration
```

```

FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
WHERE af.duration<'01:10:00';

```

| flight_id | flight_Departure_Date | From_Location | To_Location | duration |
|-----------|-----------------------|---------------|-------------|----------|
| 3173      | 2013-04-07            | hyderabad     | chennai     | 00:45:00 |
| 3175      | 2013-05-25            | chennai       | hyderabad   | 01:00:00 |
| 3176      | 2013-03-14            | kochi         | chennai     | 01:05:00 |
| 3178      | 2013-05-06            | chennai       | hyderabad   | 01:00:00 |
| 3179      | 2013-04-03            | chennai       | kolkata     | 01:00:00 |
| 3180      | 2013-04-02            | kolkata       | kochi       | 00:45:00 |

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

```

SELECT afd.flight_id,af.from_location,af.to_location,
count(afd.flight_departure_date) No_of_Service, avg(price) Average_Price
FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id,af.from_location,af.to_location
HAVING avg(price)>(SELECT avg(price) FROM air_flight_details)
ORDER BY average_price;

```

| flight_id | from_location | to_location | No_of_Service | Average_Price |
|-----------|---------------|-------------|---------------|---------------|
| 3175      | chennai       | hyderabad   | 1             | 3500.000000   |
| 3174      | kolkata       | delhi       | 1             | 3800.000000   |
| 3179      | chennai       | kolkata     | 1             | 4000.000000   |
| 3171      | chennai       | delhi       | 1             | 5000.000000   |
| 3176      | kochi         | chennai     | 1             | 8000.000000   |

# MOVIE

```
CREATE DATABASE video;USE video;
```

```
Create table CUSTOMER_MASTER
```

```
(CUSTOMER_ID Varchar(10),CUSTOMER_NAME Varchar(30) NOT NULL,CONTACT_NO
BIGINT(10),CONTACT_ADD Varchar(20),DATE_OF_REGISTRATION Date NOT NULL,AGE
Varchar(15)NOT NULL,Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID));
```

```
Create table LIBRARY_CARD_MASTER
```

```
(CARD_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT
BIGINT(50),NUMBER_OF_YEARS bigint(10) NOT NULL,Constraint MT_cts2 PRIMARY
KEY(CARD_ID));
```

```
Create table MOVIES_MASTER
```

```
(MOVIE_ID Varchar(10), MOVIE_NAME Varchar(50) NOT NULL,RELEASE_DATE Varchar(30)
NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION VARCHAR(10) NOT NULL,
 MOVIE_TYPE Varchar(3),MOVIE_CATEGORY VARCHAR(20) NOT NULL,DIRECTOR
VARCHAR(20) NOT NULL,
```

```
LEAD_ROLE_1 Varchar(3) NOT NULL,LEAD_ROLE_2 VARCHAR(4) NOT NULL,RENT_COST
BIGINT(10),Constraint MT_cts4 PRIMARY KEY(MOVIE_ID));
```

```
Create table CUSTOMER_CARD_DETAILS
```

```
(CUSTOMER_ID Varchar(10),CARD_ID VARCHAR(10),ISSUE_DATE DATE NOT NULL,Constraint
MT_cts3 PRIMARY KEY(CUSTOMER_ID),Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID)
References CUSTOMER_MASTER(CUSTOMER_ID),Constraint MT_CTS42 FOREIGN KEY(CARD_ID)
References LIBRARY_CARD_MASTER(CARD_ID));
```

```
Create table CUSTOMER_ISSUE_DETAILS
```

```
(ISSUE_ID Varchar(10) NOT NULL,CUSTOMER_ID Varchar(10) NOT NULL,MOVIE_ID
VARCHAR(10),ISSUE_DATE Date NOT NULL,RETURN_DATE Date NOT NULL,
ACTUAL_DATE_RETURN Date NOT NULL,Constraint MT_cts5 PRIMARY
KEY(ISSUE_ID),Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References
CUSTOMER_MASTER(CUSTOMER_ID), Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID)
References MOVIES_MASTER(MOVIE_ID));
```

```
Insert into CUSTOMER_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-12', '21');
Insert into CUSTOMER_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS007', 'NAME7', 3210987654,'ADD7', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS008', 'NAME8', 2109876543,'ADD8', '2013-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS009', 'NAME9', NULL,'ADD9', '2013-02-12', '21');
Insert into CUSTOMER_MASTER Values('CUS010', 'NAM10', 9934567890,'ADD10', '2013-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS011', 'NAM11', 9875678910,'ADD11', '2013-02-12',
'21');
```

```
Insert into LIBRARY_CARD_MASTER Values('CR001', 'Silver', 200, 5);
Insert into LIBRARY_CARD_MASTER Values('CR002', 'Gold', 400, 9);
Insert into LIBRARY_CARD_MASTER Values('CR003', 'Platinum', 600, 8);
Insert into LIBRARY_CARD_MASTER Values('CR004', 'VISA', 800, 7);
Insert into LIBRARY_CARD_MASTER Values('CR005', 'CREDIT', 1200, 6);
```

```
Insert into MOVIES_MASTER Values('MV001', 'DIEHARD', '2012-05-13','ENGLISH', 4 , '2HRS',
'U/A','ACTION','DIR1','L1','L2',100);
```

Insert into MOVIES\_MASTER Values('MV002', 'THE MATRIX', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR2','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV003', 'INCEPTION', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR3','L15','L2',100);

Insert into MOVIES\_MASTER Values('MV004', 'DARK KNIGHT', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR4','L15','L2',100);

Insert into MOVIES\_MASTER Values('MV005', 'OFFICE S', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR5','L12','L24',100);

Insert into MOVIES\_MASTER Values('MV006', 'SHAWN OF DEAD', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR6','L1','L25',100);

Insert into MOVIES\_MASTER Values('MV007', 'YOUNG FRANKEN', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR7','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV008', 'CAS', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR8','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV009', 'GWW', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR9','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV010', 'TITANIC', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR10','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV011', 'THE NOTE BOOK', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR11','L1','L2',100);

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS001', 'CR001', '2012-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS002', 'CR002', '2012-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS003', 'CR002', '2013-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS004', 'CR003', '2013-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS005', 'CR003', '2012-05-13');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS001', 'CUS001', 'MV001', '2012-05-13', '2012-05-13','2012-05-13');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS002', 'CUS001', 'MV001', '2012-05-01', '2012-05-16','2012-05-16');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS003', 'CUS002', 'MV004', '2012-05-02', '2012-05-06','2012-05-16');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS004', 'CUS002', 'MV004', '2012-04-03', '2012-04-16','2012-04-20');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS005', 'CUS002', 'MV009', '2012-04-04', '2012-04-16','2012-04-20');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS006', 'CUS003', 'MV002', '2012-03-30', '2012-04-15','2012-04-20');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS007', 'CUS003', 'MV003', '2012-04-20', '2012-05-05','2012-05-05');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS008', 'CUS003', 'MV005', '2012-04-21', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS009', 'CUS003', 'MV001', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS010', 'CUS003', 'MV009', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS011', 'CUS003', 'MV010', '2012-04-23', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS012', 'CUS003', 'MV010', '2012-04-24', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS013', 'CUS003', 'MV008', '2012-04-25', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS014', 'CUS004', 'MV007', '2012-04-26', '2012-05-07','2012-05-25');

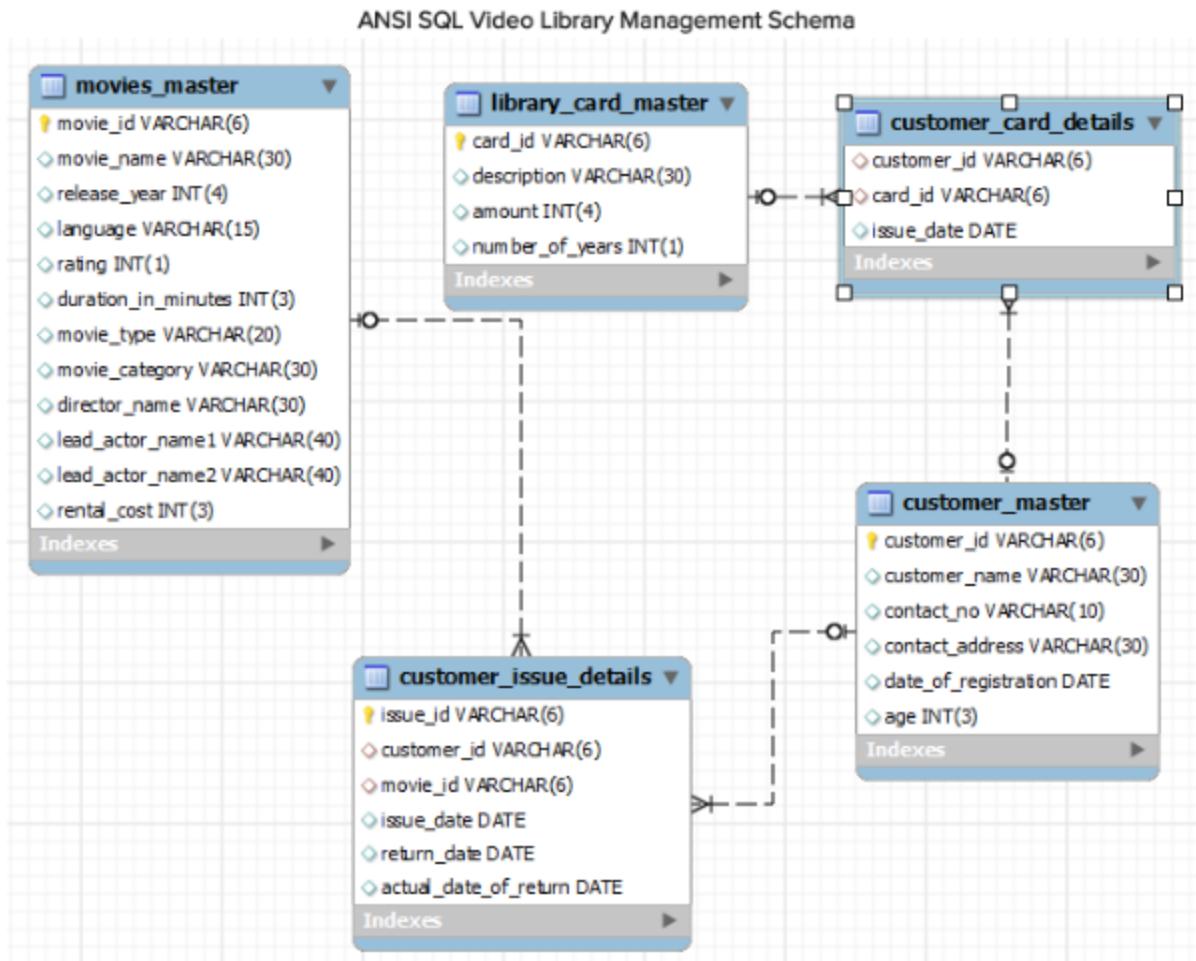
Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS015', 'CUS004', 'MV006', '2012-04-27', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS016', 'CUS004', 'MV006', '2012-04-28', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS017', 'CUS004', 'MV001', '2012-04-29', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS018', 'CUS010', 'MV008', '2012-04-24', '2012-05-07','2012-05-25');

```
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS019', 'CUS011', 'MV009', '2012-04-27', '2012-05-07','2012-05-25');
```



MOVIE MASTER

| LEAD_ROLE_2 | RENT_COST |
|-------------|-----------|
| L2          | 100       |
| L24         | 100       |
| L25         | 100       |
| L2          | 100       |
| NULL        | NULL      |

### CUSTOMER MASTER

| CUSTOMER_ID | CUSTOMER_NAME | CONTACT_NO | CONTACT_ADD | DATE_OF_REGISTRATION | AGE  |
|-------------|---------------|------------|-------------|----------------------|------|
| CUS001      | AMIT          | 9876543210 | ADD1        | 2012-02-12           | 21   |
| CUS002      | ABDHUL        | 8765432109 | ADD2        | 2012-02-12           | 21   |
| CUS003      | GAYAN         | 7654321098 | ADD3        | 2012-02-12           | 21   |
| CUS004      | RADHA         | 6543210987 | ADD4        | 2012-02-12           | 21   |
| CUS005      | GURU          | NULL       | ADD5        | 2012-02-12           | 21   |
| CUS006      | MOHAN         | 4321098765 | ADD6        | 2012-02-12           | 21   |
| CUS007      | NAME7         | 3210987654 | ADD7        | 2012-02-12           | 21   |
| CUS008      | NAME8         | 2109876543 | ADD8        | 2013-02-12           | 21   |
| CUS009      | NAME9         | NULL       | ADD9        | 2013-02-12           | 21   |
| CUS010      | NAM10         | 9934567890 | ADD10       | 2013-02-12           | 21   |
| CUS011      | NAM11         | 9875678910 | ADD11       | 2013-02-12           | 21   |
| NULL        | NULL          | NULL       | NULL        | NULL                 | NULL |

### LIBRARY CARD MASTER

| CARD_ID | DESCRIPTION | AMOUNT | NUMBER_OF_YEARS |
|---------|-------------|--------|-----------------|
| CR001   | Silver      | 200    | 5               |
| CR002   | Gold        | 400    | 9               |
| CR003   | Platinum    | 600    | 8               |
| CR004   | VISA        | 800    | 7               |
| CR005   | CREDIT      | 1200   | 6               |
| NULL    | NULL        | NULL   | NULL            |

### CUSTOMER CARD DETAILS

| CUSTOMER_ID | CARD_ID | ISSUE_DATE |
|-------------|---------|------------|
| CUS001      | CR001   | 2012-05-13 |
| CUS002      | CR002   | 2012-05-13 |
| CUS003      | CR002   | 2013-05-13 |
| CUS004      | CR003   | 2013-05-13 |
| CUS005      | CR003   | 2012-05-13 |
| NULL        | NULL    | NULL       |

#### CUSTOMER ISSUE DETAILS

| ISSUE_ID | CUSTOMER_ID | MOVIE_ID | ISSUE_DATE | RETURN_DATE | ACTUAL_DATE_RETURN |
|----------|-------------|----------|------------|-------------|--------------------|
| IS001    | CUS001      | MV001    | 2012-05-13 | 2012-05-13  | 2012-05-13         |
| IS002    | CUS001      | MV001    | 2012-05-01 | 2012-05-16  | 2012-05-16         |
| IS003    | CUS002      | MV004    | 2012-05-02 | 2012-05-06  | 2012-05-16         |
| IS004    | CUS002      | MV004    | 2012-04-03 | 2012-04-16  | 2012-04-20         |
| IS005    | CUS002      | MV009    | 2012-04-04 | 2012-04-16  | 2012-04-20         |
| IS006    | CUS003      | MV002    | 2012-03-30 | 2012-04-15  | 2012-04-20         |
| IS007    | CUS003      | MV003    | 2012-04-20 | 2012-05-05  | 2012-05-05         |
| IS008    | CUS003      | MV005    | 2012-04-21 | 2012-05-07  | 2012-05-25         |
| IS009    | CUS003      | MV001    | 2012-04-22 | 2012-05-07  | 2012-05-25         |
| IS010    | CUS003      | MV009    | 2012-04-22 | 2012-05-07  | 2012-05-25         |
| IS011    | CUS003      | MV010    | 2012-04-23 | 2012-05-07  | 2012-05-25         |
| IS012    | CUS003      | MV010    | 2012-04-24 | 2012-05-07  | 2012-05-25         |
| IS013    | CUS003      | MV008    | 2012-04-25 | 2012-05-07  | 2012-05-25         |
| IS014    | CUS004      | MV007    | 2012-04-26 | 2012-05-07  | 2012-05-25         |
| IS015    | CUS004      | MV006    | 2012-04-27 | 2012-05-07  | 2012-05-25         |
| IS016    | CUS004      | MV006    | 2012-04-28 | 2012-05-07  | 2012-05-25         |
| IS017    | CUS004      | MV001    | 2012-04-29 | 2012-05-07  | 2012-05-25         |
| IS018    | CUS010      | MV008    | 2012-04-24 | 2012-05-07  | 2012-05-25         |
| IS019    | CUS011      | MV009    | 2012-04-27 | 2012-05-07  | 2012-05-25         |
| NULL     | NULL        | NULL     | NULL       | NULL        | NULL               |

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

```
SELECT m.MOVIE_NAME, count(ISSUE_ID) ISSUE_COUNT FROM
movies_master m LEFT JOIN customer_issue_details c ON m.MOVIE_ID=c.MOVIE_ID
GROUP BY m.movie_name
```

ORDER BY ISSUE\_COUNT DESC,MOVIE\_NAME;

| MOVIE_NAME    | ISSUE_COUNT |
|---------------|-------------|
| DIEHARD       | 4           |
| GWW           | 3           |
| CAS           | 2           |
| DARK KNIGHT   | 2           |
| SHAWN OF DEAD | 2           |
| TITANIC       | 2           |
| INCEPTION     | 1           |
| OFFICE S      | 1           |
| THE MATRIX    | 1           |
| YOUNG FRANKEN | 1           |
| THE NOTE BOOK | 0           |

2. Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT\_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

```
SELECT CUSTOMER_ID,CUSTOMER_NAME,AGE,ifnull(
concat('+91-',substring(contact_no,1,3),'-',
substring(contact_no,4,3),'-',substring(contact_no,7)),'N/A') CONTACT_ISD
FROM customer_master WHERE age>25 and year(date_of_registration)='2012'
ORDER BY age,CUSTOMER_NAME;
```

| CUSTOMER_ID | CUSTOMER_NAME | AGE | CONTACT_ISD |
|-------------|---------------|-----|-------------|
|             |               |     |             |

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO\_OF\_MOVIES as alias name for number of movies.

```
SELECT MOVIE_CATEGORY,count(MOVIE_ID) NO_OF_MOVIES FROM
movies_master GROUP BY MOVIE_CATEGORY
ORDER BY NO_OF_MOVIES DESC,MOVIE_CATEGORY;
```

| MOVIE_CATEGORY | NO_OF_MOVIES |
|----------------|--------------|
| ACTION         | 4            |
| ROMANCE        | 4            |
| COMEDY         | 3            |

4. Write a query to display the number of customers having card with description "Gold card".

<br/> Hint: Use CUSTOMER\_COUNT as alias name for number of customers

```
SELECT count(c.customer_id) CUSTOMER_COUNT FROM
library_card_master l JOIN customer_card_details c ON l.CARD_ID=c.CARD_ID
WHERE description='Gold';
```

| CUSTOMER_COUNT |
|----------------|
| 2              |

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order. Use REGISTERED\_YEAR as alias name for year of registration.

```
SELECT c.customer_id,c.customer_name,
year(c.DATE_OF_REGISTRATION) REGISTERED_YEAR,cd.card_id,cd.issue_date FROM
customer_master c JOIN customer_card_details cd ON c.customer_id=cd.customer_id
ORDER BY CUSTOMER_NAME DESC;
```

| customer_id | customer_name | REGISTERED_YEAR | card_id | issue_date |
|-------------|---------------|-----------------|---------|------------|
| CUS004      | RADHA         | 2012            | CR003   | 2013-05-13 |
| CUS005      | GURU          | 2012            | CR003   | 2012-05-13 |
| CUS003      | GAYAN         | 2012            | CR002   | 2013-05-13 |
| CUS001      | AMIT          | 2012            | CR001   | 2012-05-13 |
| CUS002      | ABDHUL        | 2012            | CR002   | 2012-05-13 |

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order by customer name.

```
SELECT ci.issue_id,ci.CUSTOMER_ID,c.CUSTOMER_NAME FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
WHERE customer_name LIKE 'R%' and ci.actual_date_return>ci.return_date
```

ORDER BY customer\_name;

| issue_id | CUSTOMER_ID | CUSTOMER_NAME |
|----------|-------------|---------------|
| IS014    | CUS004      | RADHA         |
| IS015    | CUS004      | RADHA         |
| IS016    | CUS004      | RADHA         |
| IS017    | CUS004      | RADHA         |

**7.** Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT\_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10. Hint: Use AMOUNT\_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name.

```
SELECT c.CUSTOMER_ID,c.CUSTOMER_NAME,l.card_id,l.DESCRIPTION,
concat('$',round(amount/52.42)) AMOUNT_DOLLAR FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
JOIN customer_card_details cc ON cc.customer_id=c.customer_id
JOIN library_card_master l ON cc.card_id=l.card_id
WHERE c.DATE_OF_REGISTRATION=ci.issue_date
ORDER BY customer_name;
```

| CUSTOMER_ID | CUSTOMER_NAME | card_id | DESCRIPTION | AMOUNT_DOLLAR |
|-------------|---------------|---------|-------------|---------------|
|             |               |         |             |               |

**8.** Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'. Display customer name in upper case. Hint: Use CUSTOMER\_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.

```
SELECT CUSTOMER_ID,upper(CUSTOMER_NAME) CUSTOMER_NAME,contact_no,contact_add
FROM
customer_master WHERE contact_add LIKE '%Nagar' and
customer_id NOT IN (SELECT customer_id FROM customer_card_details)
and customer_id IN (SELECT customer_id FROM customer_issue_details)
```

ORDER BY CUSTOMER\_NAME;

| CUSTOMER_ID | CUSTOMER_NAME | contact_no | contact_add |
|-------------|---------------|------------|-------------|
|-------------|---------------|------------|-------------|

**9. Write a query to display the movie id, movie name, release year, director name of movies acted by the lead actor 1 who acted maximum number of movies. Display the records sorted in ascending order based on movie name.**

```
SELECT movie_id, movie_name, release_date, director FROM movies_master
WHERE lead_role_1 IN (SELECT lead_role_1 FROM
(SELECT lead_role_1, count(movie_id) ct FROM movies_master
GROUP BY lead_role_1) t WHERE t.ct >= ALL (SELECT count(movie_id)
FROM movies_master GROUP BY lead_role_1)) ORDER BY movie_name;
```

| movie_id | movie_name     | release_date | director |
|----------|----------------|--------------|----------|
| MV008    | CAS            | 2012-05-13   | DIR8     |
| MV001    | DIEHARD        | 2012-05-13   | DIR1     |
| MV009    | GWW            | 2012-05-13   | DIR9     |
| MV006    | SHAWN OF DEAD  | 2012-05-13   | DIR6     |
| MV002    | THE MATRIX     | 2012-05-13   | DIR2     |
| MV011    | THE NOTE BOOK  | 2012-05-13   | DIR11    |
| MV010    | TITANIC        | 2012-05-13   | DIR10    |
| MV007    | YOUNG FRANK... | 2012-05-13   | DIR7     |

**10. Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0. <br> Hint: Use MOVIE\_COUNT as alias name for number of movies issued.**

```
SELECT c.customer_name, count(ci.movie_id) MOVIE_COUNT FROM
customer_master c LEFT JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
GROUP BY c.customer_id ORDER BY c.customer_name;
```

| customer_name | MOVIE_COUNT |
|---------------|-------------|
| ABDHUL        | 3           |
| AMIT          | 2           |
| GAYAN         | 8           |
| GURU          | 0           |
| MOHAN         | 0           |
| NAM10         | 1           |
| NAM11         | 1           |
| NAME7         | 0           |
| NAME8         | 0           |
| NAME9         | 0           |
| RADHA         | 4           |

**11. Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number. Serial number can be generated from the issue id , that is last two characters of issue id is the serial number. For Example Assume the issue id is I00005 then the serial number is 05 Hint: Alias name for serial number is 'SERIAL\_NO'**

```
SELECT substring(ci.issue_id,-2) SERIAL_NO,ci.issue_id,c.customer_id,c.customer_name,
m.movie_id,m.movie_name FROM customer_master c JOIN customer_issue_details ci
ON c.customer_id=ci.customer_id JOIN movies_master m ON m.movie_id=ci.movie_id
ORDER BY SERIAL_NO;
```

| SERIAL_NO | issue_id | customer_id | customer_name | movie_id | movie_name    |
|-----------|----------|-------------|---------------|----------|---------------|
| 01        | IS001    | CUS001      | AMIT          | MV001    | DIEHARD       |
| 02        | IS002    | CUS001      | AMIT          | MV001    | DIEHARD       |
| 03        | IS003    | CUS002      | ABDHUL        | MV004    | DARK KNIGHT   |
| 04        | IS004    | CUS002      | ABDHUL        | MV004    | DARK KNIGHT   |
| 05        | IS005    | CUS002      | ABDHUL        | MV009    | GWW           |
| 06        | IS006    | CUS003      | GAYAN         | MV002    | THE MATRIX    |
| 07        | IS007    | CUS003      | GAYAN         | MV003    | INCEPTION     |
| 08        | IS008    | CUS003      | GAYAN         | MV005    | OFFICE S      |
| 09        | IS009    | CUS003      | GAYAN         | MV001    | DIEHARD       |
| 10        | IS010    | CUS003      | GAYAN         | MV009    | GWW           |
| 11        | IS011    | CUS003      | GAYAN         | MV010    | TITANIC       |
| 12        | IS012    | CUS003      | GAYAN         | MV010    | TITANIC       |
| 13        | IS013    | CUS003      | GAYAN         | MV008    | CAS           |
| 14        | IS014    | CUS004      | RADHA         | MV007    | YOUNG FRAN... |
| 15        | IS015    | CUS004      | RADHA         | MV006    | SHAWN OF D... |
| 16        | IS016    | CUS004      | RADHA         | MV006    | SHAWN OF D... |
| 17        | IS017    | CUS004      | RADHA         | MV001    | DIEHARD       |
| 18        | IS018    | CUS010      | NAM10         | MV008    | CAS           |
| 19        | IS019    | CUS011      | NAM11         | MV009    | GWW           |

**12. Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.**

```
SELECT ci.issue_id,ci.issue_date,c.customer_id,c.customer_name,c.contact_no FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
and year(ci.issue_date)='2013' ORDER BY ci.issue_date DESC;
```

| issue_id | issue_date | customer_id | customer_name | contact_no |
|----------|------------|-------------|---------------|------------|
|----------|------------|-------------|---------------|------------|

**13. Write a query to display movie id , movie name and actor names of movies which are not issued to any customers. <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.**

```
SELECT movie_id,movie_name,concat(lead_role_1,' & ',lead_role_2) ACTOR FROM movies_master
```

```
WHERE movie_id NOT IN (SELECT movie_id FROM customer_issue_details) ORDER BY movie_name;
```

| movie_id | movie_name    | ACTOR   |
|----------|---------------|---------|
| MV011    | THE NOTE BOOK | L1 & L2 |

**14. Write a query to display the director's name, movie name and lead\_actor\_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR\_NAME as alias name for director name column Display the records sorted in ascending order based on director\_name and then by movie\_name in descending order.**

```
SELECT upper(director) DIRECTOR_NAME,movie_name,lead_role_1 FROM movies_master
GROUP BY director HAVING count(movie_id)>1 ORDER BY director,movie_name DESC;
```

| DIRECTOR_NAME | movie_name | lead_role_1 |
|---------------|------------|-------------|
|---------------|------------|-------------|

**15. Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <br> Hint:Use NO\_OF\_CUSTOMERS as alias name for number of customers.**

```
SELECT count(customer_id) NO_OF_CUSTOMER FROM customer_master
WHERE contact_no is not null and year(date_of_registration)='2012';
```

| NO_OF_CUSTOMER |
|----------------|
| 6              |

**16. Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT\_DETAILS as alias name for customer contact.**

```

SELECT c.customer_name,ifnull(c.contact_no,c.contact_addr)
CONTACT_DETAILS,l.card_id,l.description FROM
customer_master c LEFT JOIN customer_card_details cc ON c.customer_id=cc.customer_id
LEFT JOIN library_card_master l ON l.card_id=cc.card_id
ORDER BY customer_name;

```

| customer_name | CONTACT_DETAILS | card_id | description |
|---------------|-----------------|---------|-------------|
| ABDHUL        | 8765432109      | CR002   | Gold        |
| AMIT          | 9876543210      | CR001   | Silver      |
| GAYAN         | 7654321098      | CR002   | Gold        |
| GURU          | ADD5            | CR003   | Platinum    |
| MOHAN         | 4321098765      | NULL    | NULL        |
| NAM10         | 9934567890      | NULL    | NULL        |
| NAM11         | 9875678910      | NULL    | NULL        |
| NAME7         | 3210987654      | NULL    | NULL        |
| NAME8         | 2109876543      | NULL    | NULL        |
| NAME9         | ADD9            | NULL    | NULL        |
| RADHA         | 6543210987      | CR003   | Platinum    |

**17.** Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO\_OF\_TIMES as alias name for number of times

```

SELECT ci.customer_id,c.customer_name,count(ci.movie_id) NO_OF_TIMES FROM
customer_issue_details ci JOIN customer_master c ON c.customer_id=ci.customer_id
GROUP BY ci.customer_id,ci.movie_id HAVING count(movie_id)>1
ORDER BY customer_name DESC;

```

| customer_id | customer_name | NO_OF_TIMES |
|-------------|---------------|-------------|
| CUS004      | RADHA         | 2           |
| CUS003      | GAYAN         | 2           |
| CUS001      | AMIT          | 2           |
| CUS002      | ABDHUL        | 2           |

**18.**Write a query to display customer id, customer name,contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-

**456-2345" format.&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.**

```
SELECT c.customer_id,c.customer_name,concat('+91-',substring(c.contact_no,1,3),'-',
substring(c.contact_no,4,3),'-',substring(c.contact_no,7)) CONTACT_ISD
,m.movie_category,count(cc.movie_id) NO_OF_MOVIES FROM customer_master c JOIN
customer_issue_details cc
ON c.customer_id=cc.customer_id JOIN movies_master m ON m.movie_id=cc.movie_id
GROUP BY c.customer_id,m.movie_category HAVING count(cc.movie_id)>1
ORDER BY customer_name,movie_category;
```

| customer_id | customer_name | CONTACT_ISD      | movie_category | NO_OF_MOVIES |
|-------------|---------------|------------------|----------------|--------------|
| CUS002      | ABDHUL        | +91-876-543-2109 | ACTION         | 2            |
| CUS001      | AMIT          | +91-987-654-3210 | ACTION         | 2            |
| CUS003      | GAYAN         | +91-765-432-1098 | ACTION         | 3            |
| CUS003      | GAYAN         | +91-765-432-1098 | ROMANCE        | 4            |
| CUS004      | RADHA         | +91-654-321-0987 | COMEDY         | 3            |

**19. Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies. For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.**

```
SELECT cid.customer_id , customer_name FROM customer_master cm JOIN
customer_issue_details cid ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)>=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id)
UNION
```

```

SELECT cid.customer_id , customer_name FROM
customer_master cm JOIN customer_issue_details cid
ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)<=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id) ORDER BY customer_name;

```

| customer_id | customer_name |
|-------------|---------------|
| CUS003      | GAYAN         |
| CUS010      | NAM10         |
| CUS011      | NAM11         |

**20.**Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once. Hint: Use NO\_OF\_TIMES as alias name Display the records in ascending order based on customer name.

```

SELECT c.customer_id,c.customer_name,count(m.movie_id) NO_OF_TIMES FROM
customer_master c JOIN customer_issue_details cc ON c.customer_id=cc.customer_id
JOIN movies_master m ON m.movie_id=cc.movie_id
WHERE m.movie_category='Comedy'
GROUP BY c.customer_id HAVING count(m.movie_id)>1
ORDER BY customer_name;

```

| customer_id | customer_name | NO_OF_TIMES |
|-------------|---------------|-------------|
| CUS004      | RADHA         | 3           |

**21.**Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL\_COST. Display the records sorted in ascending order based on customer id

```

SELECT cid.customer_id, sum(m.rent_cost) TOTAL_COST FROM customer_issue_details cid
JOIN movies_master mm ON cid.movie_id=mm.movie_id GROUP BY cid.customer_id order by
customer_id;

```

| customer_id | TOTAL_COST |
|-------------|------------|
| CUS001      | 200        |
| CUS002      | 300        |
| CUS003      | 800        |
| CUS004      | 400        |
| CUS010      | 100        |
| CUS011      | 100        |

# LOAN

```
create database loan;
```

```
use loan;
```

```
CREATE TABLE loan_card_master
```

```
(
 loan_id varchar(6) PRIMARY KEY,
 loan_type varchar(15),
 duration_in_years int(2)
);
```

```
CREATE TABLE employee_master
```

```
(
 employee_id varchar(6) PRIMARY KEY,
 employee_name varchar(20),
 designation varchar(25),
 department varchar(25),
 gender char(1),
 date_of_birth date,
 date_of_joining date
);
```

```
CREATE TABLE item_master
```

```
(
 item_id varchar(6) PRIMARY KEY,
 item_description varchar(25),
```

```

issue_status char(1),
item_make varchar(25),
item_category varchar(20),
item_valuation int(6)

);

CREATE TABLE employee_card_details
(
 employee_id varchar(6) REFERENCES employee_master,
 loan_id varchar(6) REFERENCES loan_card_master,
 card_issue_date date
);

CREATE TABLE employee_issue_details
(
 issue_id varchar(6) PRIMARY KEY,
 employee_id varchar(6) REFERENCES employee_master,
 item_id varchar(6) REFERENCES item_master,
 issue_date date,
 return_date date
);


```

insert into loan\_card\_master values('L00001','Furniture',5);

insert into loan\_card\_master values('L00002','Stationary',0);

insert into loan\_card\_master values('L00003','Crockery',1);

```
insert into employee_issue_details values('ISS001','E00001','I00001','2012-02-03','2014-02-03');
insert into employee_issue_details values('ISS002','E00001','I00004','2012-02-03','2020-02-03');
insert into employee_issue_details values('ISS003','E00002','I00005','2013-01-03','2015-01-03');
insert into employee_issue_details values('ISS004','E00003','I00007','2010-07-04','2012-07-04');
insert into employee_issue_details values('ISS005','E00003','I00008','2010-07-04','2012-08-05');
insert into employee_issue_details values('ISS006','E00003','I00010','2012-03-14','2012-06-15');
insert into employee_issue_details values('ISS007','E00004','I00012','2013-04-14','2016-04-14');
insert into employee_issue_details values('ISS008','E00006','I00018','2012-08-18','2019-04-17');
insert into employee_issue_details values('ISS009','E00004','I00018','2013-04-18','2013-05-18');
```

```
insert into employee_master values('E00001','Ram','Manager','Finance','M','1973-12-01','2000-01-01');
insert into employee_master values('E00002','Abhay','Assistant Manager','Finance','M','1976-01-01','2006-12-01');
insert into employee_master values('E00003','Anita','Senior Executive','Marketing','F','1977-05-12','2007-03-21');
insert into employee_master values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');
insert into employee_master values('E00005','Radhica','Manager','HR','F','1976-07-22','2004-01-23');
insert into employee_master values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');
```

```
insert into employee_card_details values('E00001','L00001','2000-01-01');
insert into employee_card_details values('E00001','L00002','2000-01-01');
insert into employee_card_details values('E00001','L00003','2002-12-14');
insert into employee_card_details values('E00002','L00001','2007-02-01');
```

```
insert into employee_card_details values('E00002','L00002','2007-03-11');
insert into employee_card_details values('E00003','L00001','2007-04-15');
insert into employee_card_details values('E00003','L00002','2007-04-15');
insert into employee_card_details values('E00003','L00003','2007-04-15');
```

```
INSERT INTO item_master VALUES ('I00001','Tea Table','Y','Wooden','Furniture',5000);
INSERT INTO item_master VALUES ('I00002','Dinning Table','N','Wooden','Furniture',15000);
INSERT INTO item_master VALUES ('I00003','Tea Table','N','Steel','Furniture',6000);
INSERT INTO item_master VALUES ('I00004','Side Table','Y','Wooden','Furniture',2000);
INSERT INTO item_master VALUES ('I00005','Side Table','Y','Steel','Furniture',1500);
INSERT INTO item_master VALUES ('I00006','Tea Table','N','Steel','Furniture',7000);
INSERT INTO item_master VALUES ('I00007','Dinning Chair','Y','Wooden','Furniture',1500);
INSERT INTO item_master VALUES ('I00008','Tea Table','Y','Wooden','Furniture',4000);
INSERT INTO item_master VALUES ('I00009','Sofa','N','Wooden','Furniture',18000);
INSERT INTO item_master VALUES ('I00010','Cupboard','Y','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00011','Cupboard','N','Steel','Furniture',14000);
INSERT INTO item_master VALUES ('I00012','Double Bed','Y','Wooden','Furniture',21000);
INSERT INTO item_master VALUES ('I00013','Double Bed','Y','Wooden','Furniture',20000);
INSERT INTO item_master VALUES ('I00014','Single Bed','Y','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00015','Single Bed','N','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00016','Tea Set','Y','Glass','Crockery',3000);
INSERT INTO item_master VALUES ('I00017','Tea Set','Y','Bonechina','Crockery',4000);
INSERT INTO item_master VALUES ('I00018','Dinning Set','Y','Glass','Crockery',4500);
INSERT INTO item_master VALUES ('I00019','Dinning Set','N','Bonechina','Crockery',5000);
INSERT INTO item_master VALUES ('I00020','Pencil','Y','Wooden','Stationary',5);
```

```
INSERT INTO item_master VALUES ('I00021','Pen','Y','Plastic','Stationary',100);
INSERT INTO item_master VALUES ('I00022','Pen','N','Plastic','Stationary',200);
```

### LOAN CARD MASTER

| loan_id | loan_type  | duration_in_years |
|---------|------------|-------------------|
| L00001  | Furniture  | 5                 |
| L00002  | Stationary | 0                 |
| L00003  | Crockery   | 1                 |
| NULL    | NULL       | NULL              |

### EMPLOYEE CARD DETAILS

| employee_id | loan_id | card_issue_date |
|-------------|---------|-----------------|
| E00001      | L00001  | 2000-01-01      |
| E00001      | L00002  | 2000-01-01      |
| E00001      | L00003  | 2002-12-14      |
| E00002      | L00001  | 2007-02-01      |
| E00002      | L00002  | 2007-03-11      |
| E00003      | L00001  | 2007-04-15      |
| E00003      | L00002  | 2007-04-15      |
| E00003      | L00003  | 2007-04-15      |

## **EMPLOYEE ISSUE DETAILS**

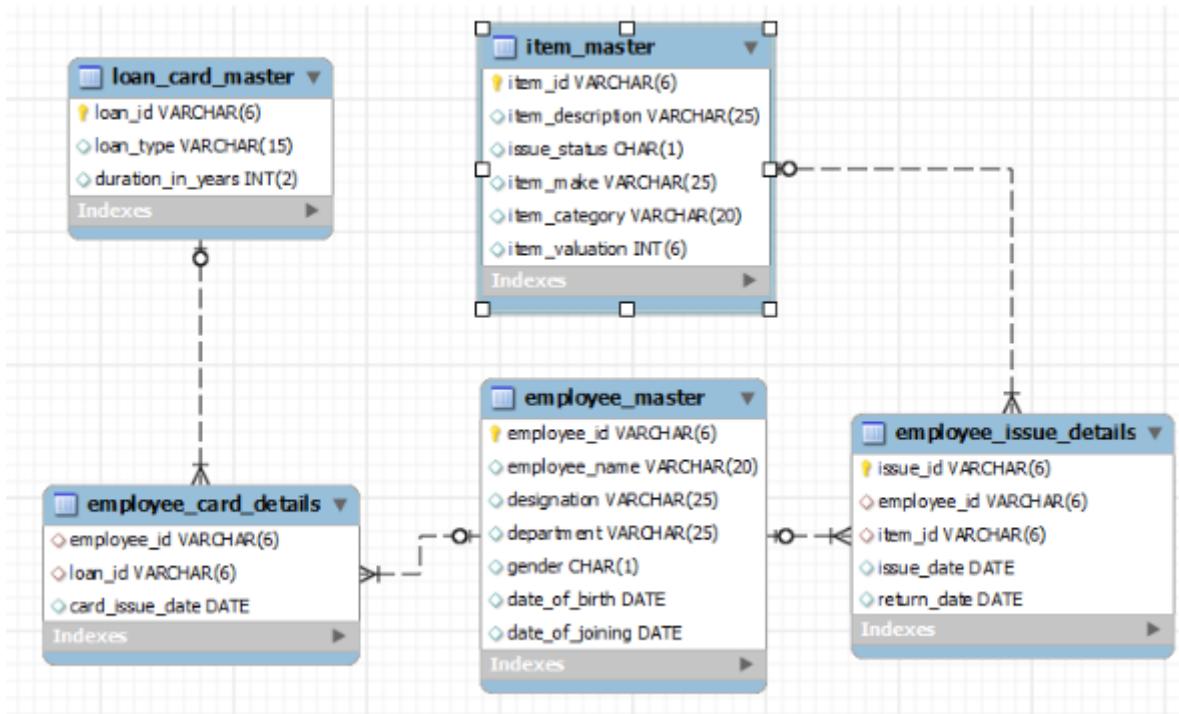
| issue_id | employee_id | item_id | issue_date | return_date |
|----------|-------------|---------|------------|-------------|
| ISS001   | E00001      | I00001  | 2012-02-03 | 2014-02-03  |
| ISS002   | E00001      | I00004  | 2012-02-03 | 2020-02-03  |
| ISS003   | E00002      | I00005  | 2013-01-03 | 2015-01-03  |
| ISS004   | E00003      | I00007  | 2010-07-04 | 2012-07-04  |
| ISS005   | E00003      | I00008  | 2010-07-04 | 2012-08-05  |
| ISS006   | E00003      | I00010  | 2012-03-14 | 2012-06-15  |
| ISS007   | E00004      | I00012  | 2013-04-14 | 2016-04-14  |
| ISS008   | E00006      | I00018  | 2012-08-18 | 2019-04-17  |
| ISS009   | E00004      | I00018  | 2013-04-18 | 2013-05-18  |
| NUL      | NUL         | NUL     | NUL        | NUL         |

## **EMPLOYEE MASTER**

| employee_id | employee_name | designation       | department | gender | date_of_birth | date_of_joining |
|-------------|---------------|-------------------|------------|--------|---------------|-----------------|
| E00001      | Ram           | Manager           | Finance    | M      | 1973-12-01    | 2000-01-01      |
| E00002      | Abhay         | Assistant Manager | Finance    | M      | 1976-01-01    | 2006-12-01      |
| E00003      | Anita         | Senior Executive  | Marketing  | F      | 1977-05-12    | 2007-03-21      |
| E00004      | Zuben         | Manager           | Marketing  | M      | 1974-10-12    | 2003-07-23      |
| E00005      | Radhica       | Manager           | HR         | F      | 1976-07-22    | 2004-01-23      |
| E00006      | John          | Executive         | HR         | M      | 1983-11-08    | 2010-05-17      |
| NUL         | NUL           | NUL               | NUL        | NUL    | NUL           | NUL             |

## ITEM MASTER

| item_id | item_description | issue_status | item_make | item_category | item_valuation |
|---------|------------------|--------------|-----------|---------------|----------------|
| I00001  | Tea Table        | Y            | Wooden    | Furniture     | 5000           |
| I00002  | Dinning Table    | N            | Wooden    | Furniture     | 15000          |
| I00003  | Tea Table        | N            | Steel     | Furniture     | 6000           |
| I00004  | Side Table       | Y            | Wooden    | Furniture     | 2000           |
| I00005  | Side Table       | Y            | Steel     | Furniture     | 1500           |
| I00006  | Tea Table        | N            | Steel     | Furniture     | 7000           |
| I00007  | Dinning Chair    | Y            | Wooden    | Furniture     | 1500           |
| I00008  | Tea Table        | Y            | Wooden    | Furniture     | 4000           |
| I00009  | Sofa             | N            | Wooden    | Furniture     | 18000          |
| I00010  | Cupboard         | Y            | Steel     | Furniture     | 10000          |
| I00011  | Cupboard         | N            | Steel     | Furniture     | 14000          |
| I00012  | Double Bed       | Y            | Wooden    | Furniture     | 21000          |
| I00013  | Double Bed       | Y            | Wooden    | Furniture     | 20000          |
| I00014  | Single Bed       | Y            | Steel     | Furniture     | 10000          |
| I00015  | Single Bed       | N            | Steel     | Furniture     | 10000          |
| I00016  | Tea Set          | Y            | Glass     | Crockery      | 3000           |
| I00017  | Tea Set          | Y            | Bonechina | Crockery      | 4000           |
| I00018  | Dinning Set      | Y            | Glass     | Crockery      | 4500           |
| I00019  | Dinning Set      | N            | Bonechina | Crockery      | 5000           |
| I00020  | Pencil           | Y            | Wooden    | Stationary    | 5              |
| I00021  | Pen              | Y            | Plastic   | Stationary    | 100            |
| I00022  | Pen              | N            | Plastic   | Stationary    | 200            |
| NUL     | NUL              | NUL          | NUL       | NUL           | NUL            |



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

```
SELECT item_category, count(item_id) Count_category FROM
item_master GROUP BY item_category ORDER BY Count_category DESC;
```

| item_category | Count_category |
|---------------|----------------|
| Furniture     | 15             |
| Crockery      | 4              |
| Stationary    | 3              |

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

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

| No_of_Employees |
|-----------------|
| 2               |

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

```
SELECT employee_id, employee_name, designation, department FROM employee_master
```

WHERE employee\_id NOT IN (SELECT employee\_id FROM employee\_issue\_details)  
 ORDER BY employee\_id;

| employee_id | employee_name | designation | department |
|-------------|---------------|-------------|------------|
| E00005      | Radhica       | Manager     | HR         |
| NULL        | NULL          | NULL        | NULL       |

4. Write a query to display the employee id, employee name who was issued an item of highest valuation. In case of multiple records, display the records sorted in ascending order based on employee id.[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

```
SELECT employee_id,employee_name FROM employee_master
WHERE employee_id IN(SELECT employee_id FROM employee_issue_details
WHERE item_id IN (SELECT item_id FROM item_master
WHERE item_valuation=(SELECT max(item_valuation) FROM
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)));
```

| employee_id | employee_name |
|-------------|---------------|
| E00004      | Zuben         |
| NULL        | NULL          |

5. Write a query to display issue\_id, employee\_id, employee\_name. Display the records sorted in ascending order based on issue id.

```
SELECT eid.issue_id, eid.employee_id, em.employee_name
FROM employee_master em JOIN employee_issue_details eid
ON em.employee_id=eid.employee_id ORDER BY eid.issue_id;
```

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

**6. Write a query to display employee id, employee name who don't have loan cards. Display the records sorted in ascending order based on employee id.**

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

| employee_id | employee_name |
|-------------|---------------|
| E00004      | Zuben         |
| E00005      | Radhica       |
| E00006      | John          |
| NULL        | NULL          |

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

```
SELECT count(loan_id) No_of_Cards FROM
employee_card_details WHERE employee_id IN
(SELECT employee_id FROM employee_master WHERE employee_name='Ram');
```

**(or)**

```
SELECT count(loan_id) No_of_Cards FROM
employee_card_details c JOIN employee_master e
ON c.employee_id = e.employee_id
WHERE e.employee_name= 'Ram';
```

| No_of_Cards |
|-------------|
| 3           |

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

```
SELECT count(e.employee_id) Count_Stationary
FROM employee_card_details e JOIN loan_card_master l
ON e.loan_id=l.loan_id WHERE l.loan_type='Stationary';
```

| Count_Stationary |
|------------------|
| 3                |

**9. Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then**

```
SELECT e.employee_id,employee_name,count(e.item_id) Count FROM
```

```
employee_issue_details e JOIN employee_master em ON e.employee_id=em.employee_id
GROUP BY e.employee_id ORDER BY count DESC,e.employee_id;
```

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

**10.** Write a query to display the employee id, employee name who was issued an item of minimum valuation. In case of multiple records, display them sorted in ascending order based on employee id. [Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

```
SELECT employee_id,employee_name FROM employee_master
WHERE employee_id IN(SELECT employee_id FROM employee_issue_details
WHERE item_id IN (SELECT item_id FROM item_master
WHERE item_valuation=(SELECT min(item_valuation) FROM
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)))
ORDER BY employee_id;
```

| employee_id | employee_name |
|-------------|---------------|
| E00002      | Abhay         |
| E00003      | Anita         |
| NULL        | NULL          |

**11.** Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL\_VALUATION. Display the records sorted in ascending order based on employee id. Consider only employees who have been issued atleast 1 item.

```
SELECT e.employee_id,em.employee_name,sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id ORDER BY employee_id;
```

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00001      | Ram           | 7000            |
| E00002      | Abhay         | 1500            |
| E00003      | Anita         | 15500           |
| E00004      | Zuben         | 25500           |
| E00006      | John          | 4500            |

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

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE datediff(ei.return_date,ei.issue_date)>365
ORDER BY employee_id;
```

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

**13.** Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT\_ITEMS. Display the records sorted in ascending order on employee id.

```
SELECT e.employee_id,e.employee_name,count(ei.item_id) COUNT_ITEMS FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON ei.item_id=i.item_id
WHERE i.item_category='Furniture'
GROUP BY ei.employee_id HAVING count(ei.item_id)>1;
```

| employee_id | employee_name | COUNT_ITEMS |
|-------------|---------------|-------------|
| E00001      | Ram           | 2           |
| E00003      | Anita         | 3           |

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

```
SELECT gender,count(employee_id) FROM employee_master
```

```
GROUP BY gender ORDER BY gender;
```

| gender | count(employee_id) |
|--------|--------------------|
| F      | 2                  |
| M      | 4                  |

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

```
SELECT employee_id,employee_name FROM employee_master
```

```
WHERE year(date_of_joining)>'2005'
```

```
ORDER BY employee_id;
```

| employee_id | employee_name |
|-------------|---------------|
| E00002      | Abhay         |
| E00003      | Anita         |
| E00006      | John          |
| NULL        | NULL          |

**16. Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No\_of\_Furnitures. Display the records sorted in ascending order based on issue\_status.**

```
SELECT issue_status,count(item_id) No_of_Furniture FROM
```

```
item_master WHERE item_category='Furniture'
```

```
GROUP BY issue_status ORDER BY issue_status;
```

| issue_status | No_of_Furniture |
|--------------|-----------------|
| N            | 6               |
| Y            | 9               |

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

```
SELECT item_category,item_make,item_description,count(item_id) No_of_items FROM
```

```
item_master GROUP BY item_category,item_make,item_description
```

```
ORDER BY item_category,item_make,item_description;
```

| item_category | item_make | item_description | No_of_items |
|---------------|-----------|------------------|-------------|
| Crockery      | Bonechina | Dinning Set      | 1           |
| Crockery      | Bonechina | Tea Set          | 1           |
| Crockery      | Glass     | Dinning Set      | 1           |
| Crockery      | Glass     | Tea Set          | 1           |
| Furniture     | Steel     | Cupboard         | 2           |
| Furniture     | Steel     | Side Table       | 1           |
| Furniture     | Steel     | Single Bed       | 2           |
| Furniture     | Steel     | Tea Table        | 2           |
| Furniture     | Wooden    | Dinning Chair    | 1           |
| Furniture     | Wooden    | Dinning Table    | 1           |
| Furniture     | Wooden    | Double Bed       | 2           |
| Furniture     | Wooden    | Side Table       | 1           |
| Furniture     | Wooden    | Sofa             | 1           |
| Furniture     | Wooden    | Tea Table        | 2           |
| Stationary    | Plastic   | Pen              | 2           |
| Stationary    | Wooden    | Pencil           | 1           |

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

```
SELECT e.employee_id,employee_name,i.item_id,i.item_description FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
WHERE month(ei.issue_date)='01' and year(ei.issue_date)='2013'
ORDER BY employee_id,item_id;
```

| employee_id | employee_name | item_id | item_description |
|-------------|---------------|---------|------------------|
| E00002      | Abhay         | I00005  | Side Table       |

**19. Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.Give the alias name for category count as COUNT\_CATEGORY.Display the records sorted in ascending order based on employee id.**

```
SELECT ei.employee_id,e.employee_name,count(DISTINCT i.item_category) COUNT_CATEGORY FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
```

```

GROUP BY ei.employee_id
HAVING COUNT_CATEGORY >= 2
ORDER BY employee_id;

```

| employee_id | employee_name | COUNT_CATEGORY |
|-------------|---------------|----------------|
| E00004      | Zuben         | 2              |

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

```

SELECT item_id, item_description FROM item_master
WHERE item_id NOT IN (SELECT item_id from employee_issue_details)
ORDER BY item_id;

```

| item_id | item_description |
|---------|------------------|
| I00002  | Dinning Table    |
| I00003  | Tea Table        |
| I00006  | Tea Table        |
| I00009  | Sofa             |
| I00011  | Cupboard         |
| I00013  | Double Bed       |
| I00014  | Single Bed       |
| I00015  | Single Bed       |
| I00016  | Tea Set          |
| I00017  | Tea Set          |
| I00019  | Dinning Set      |
| I00020  | Pencil           |
| I00021  | Pen              |
| I00022  | Pen              |
| NULL    | NULL             |

**21. Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed. ]**

```

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id = i.item_id
JOIN employee_master em ON em.employee_id = e.employee_id

```

```

GROUP BY e.employee_id HAVING sum(i.item_valuation)<=ALL(
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id);

```

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00002      | Abhay         | 1500            |

22. Write a query to display the employee id, employee name, card issue date and card valid date. Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD\_VALID\_DATE. [Hint: Validity in years for the loan card is given in loan\_card\_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'. ]

```

SELECT e.employee_id,e.employee_name,card_issue_date,
case
when l.duration_in_years>0 then date_add(ec.card_issue_date,interval l.duration_in_years year)
when l.duration_in_years=0 then 'No Validity Date' end CARD_VALID_DATE
FROM
employee_master e JOIN employee_card_details ec ON e.employee_id=ec.employee_id
JOIN loan_card_master l ON l.loan_id=ec.loan_id
ORDER BY employee_name,CARD_VALID_DATE;

```

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

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

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
```

```

employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE e.employee_id NOT IN (SELECT employee_id FROM employee_issue_details
WHERE year(issue_date)='2013')
ORDER BY employee_id;

```

| employee_id | employee_name |
|-------------|---------------|
| E00001      | Ram           |
| E00003      | Anita         |
| E00006      | John          |

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

```

SELECT issue_id, eid.employee_id, employee_name, im.item_id, item_description, issue_date
FROM employee_issue_details eid JOIN employee_master em ON eid.employee_id=em.employee_id
JOIN item_master im ON eid.item_id=im.item_id
ORDER BY issue_date DESC, issue_id;

```

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

**25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]**

```

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id HAVING sum(i.item_valuation)>=ALL(

```

```
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id);
```

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00004      | Zuben         | 25500           |

## Airline Flight Management :

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

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

15 rows

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

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

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

2. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines.

The Query should display profile\_id, customer's first\_name, Address and Number of tickets booked as "No\_of\_Tickets"

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

1 row

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

```

count(ati.ticket_id) <=all

(select count(ati.ticket_id) from air_passenger_profile apf
join air_ticket_info ati on apf.profile_id=ati.profile_id group by apf.profile_id) order by
first_name;

```

| PROFILE_ID | FIRST_NAME | ADDRESS                        | NO_OF_TICKETS |
|------------|------------|--------------------------------|---------------|
| PFL008     | GANESH     | 45 3RD<br>ST, HYDERABAD-<br>24 | 1             |

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

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

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

9 rows

```

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

```

| FROM_LOCATION | TO_LOCATION | MONTH_NAME | NO_OF_SERVICES |
|---------------|-------------|------------|----------------|
| BENGALURU     | CHENNAI     | MAY        | 7              |

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

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

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

1 row

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

```

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

```

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

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

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

3 rows

```

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

```

| PROFILE_ID | FIRST_NAME | LAST_NAME | FLIGHT_ID | FLIGHT_DEPARTURE_DATE | NO_OF_TICKETS |
|------------|------------|-----------|-----------|-----------------------|---------------|
| PFL001     | LATHA      | SANKAR    | 1265      | 2013-04-29            | 1             |

|        |        |        |      |            |   |
|--------|--------|--------|------|------------|---|
| PFL004 | AARTHI | RAMESH | 1265 | 2013-05-29 | 1 |
| PFL005 | SIVA   | KUMAR  | 916  | 2013-05-06 | 2 |

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

3 rows

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

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

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

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

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

11 rows

```
select af.flight_id,af.from_location,af.to_location,avg(afd.price)
```

```

from air_flight af join air_flight_details afd
on af.flight_id=afd.flight_id group by af.flight_id,af.from_location,af.to_location
order by af.flight_id,af.from_location,af.to_location;

```

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

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

Give an alias to the name as customer\_name.

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

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

3 rows

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

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

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

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

2 rows

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

|            |
|------------|
| PROFILE_ID |
| PFL002     |
| PFL007     |

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

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

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

7 rows

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

| IGHT_ID | FROM_LOCATION | TO_LOCATION | NO_OF_TICKETS |
|---------|---------------|-------------|---------------|
| 1011    | HYDERABAD     | CHENNAI     | 4             |
| 1262    | HYDERABAD     | CHENNAI     | 1             |
| 1265    | CHENNAI       | HYDERABAD   | 2             |
| 3004    | BENGALURU     | CHENNAI     | 3             |
| 3148    | CHENNAI       | BENGALURU   | 7             |

|      |         |           |   |
|------|---------|-----------|---|
| 3244 | KOCHI   | CHENNAI   | 7 |
| 916  | CHENNAI | HYDERABAD | 2 |

11. Write a query to display the no of services offered by each flight and the total price of the services.

The Query should display flight\_id, number of services as “No\_of\_Services” and the cost as “Total\_Price” in the same order.

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

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

11 rows

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

| FLIGHT_ID | NO_OF_SERVICES | TOTAL_PRICE |
|-----------|----------------|-------------|
| 916       | 4              | 14798.00    |
| 1265      | 4              | 13997.00    |
| 3307      | 4              | 13236.00    |
| 3013      | 4              | 13031.00    |

|      |   |          |
|------|---|----------|
| 289  | 4 | 13031.00 |
| 1011 | 3 | 12325.00 |
| 3004 | 3 | 9959.00  |
| 3241 | 3 | 9911.00  |
| 3148 | 3 | 8877.00  |
| 1262 | 2 | 6889.00  |
| 3244 | 2 | 6743.00  |

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

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

9 rows

```

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

```

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

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

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

1 row

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

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

|            |
|------------|
| PROFILE_ID |
|------------|

PFL008

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

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

4 rows

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

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

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

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

1 row

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

| PROFILE_ID | FIRST_NAME | LAST_NAME | FLIGHT_ID | FLIGHT_DEPARTURE_DATE | DEPARTURE_TIME | ARRIVAL_TIME |
|------------|------------|-----------|-----------|-----------------------|----------------|--------------|
| PFL005     | SIVA       | KUMAR     | 916       | 2013-05-06            | 19:55:00       | 21:00:00     |

| DELAYED_DEPARTURE_TIME | DELAYED_ARRIVAL_TIME |
|------------------------|----------------------|
| 20:55:00               | 22:00:00             |

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

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

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

**2 rows**

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

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

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

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

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

**11 rows**

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

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

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

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

2 rows

```
select profile_id, last_name, mobile_number, email_id from air_passenger_profile where
```

```
substring_index(substring_index(address,'',-1),'-',1)='chennai'
order by profile_id;
```

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

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

1 row

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

| FLIGHT_COUNT |
|--------------|
| 3            |

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

2 rows

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

join air_ticket_info ati on app.profile_id=ati.profile_id

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

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

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

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

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

11 rows

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

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | DEPARTURE_TIME | TIME_OF_SERVICE |
|-----------|---------------|-------------|----------------|-----------------|
| 1011      | HYDERABAD     | CHENNAI     | 12:30:00       | AFTERNOON       |

|      |           |           |          |         |
|------|-----------|-----------|----------|---------|
| 1262 | HYDERABAD | CHENNAI   | 06:00:00 | MORNING |
| 1265 | CHENNAI   | HYDERABAD | 21:25:00 | EVENING |
| 289  | CHENNAI   | KOCHI     | 08:40:00 | MORNING |
| 3004 | BENGALURU | CHENNAI   | 09:05:00 | MORNING |
| 3013 | CHENNAI   | BENGALURU | 07:40:00 | MORNING |
| 3148 | CHENNAI   | BENGALURU | 20:15:00 | EVENING |
| 3241 | CHENNAI   | KOCHI     | 10:40:00 | MORNING |
| 3244 | KOCHI     | CHENNAI   | 21:10:00 | EVENING |
| 3307 | BENGALURU | CHENNAI   | 18:45:00 | EVENING |
| 916  | CHENNAI   | HYDERABAD | 19:55:00 | EVENING |

21. Please follow instructions given below.

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

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

36 rows

```
select flight_id, flight_departure_date,
 case when price < 3000 then 'AIR PASSENGER'
```

```

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

```

| FLIGHT_ID | FLIGHT_DEPARTURE_DATE | FLIGHT_TYPE         |
|-----------|-----------------------|---------------------|
| 1011      | 2013-04-30            | EXECUTIVE PASSENGER |
| 1011      | 2013-05-09            | EXECUTIVE PASSENGER |
| 1011      | 2013-05-21            | AIR BUS             |
| 1262      | 2013-05-20            | AIR BUS             |
| 1262      | 2013-05-29            | AIR BUS             |
| 1265      | 2013-04-29            | EXECUTIVE PASSENGER |
| 1265      | 2013-05-14            | AIR BUS             |
| 1265      | 2013-05-18            | EXECUTIVE PASSENGER |
| 1265      | 2013-05-29            | AIR PASSENGER       |
| 289       | 2013-05-06            | AIR BUS             |
| 289       | 2013-05-08            | AIR BUS             |

|      |            |                     |
|------|------------|---------------------|
| 289  | 2013-05-20 | AIR BUS             |
| 289  | 2013-05-31 | AIR PASSENGER       |
| 3004 | 2013-05-02 | AIR BUS             |
| 3004 | 2013-05-19 | AIR BUS             |
| 3004 | 2013-05-24 | AIR BUS             |
| 3013 | 2013-05-04 | AIR BUS             |
| 3013 | 2013-05-06 | AIR BUS             |
| 3013 | 2013-05-22 | AIR BUS             |
| 3013 | 2013-05-30 | AIR PASSENGER       |
| 3148 | 2013-05-16 | AIR BUS             |
| 3148 | 2013-05-21 | AIR BUS             |
| 3148 | 2013-06-01 | AIR PASSENGER       |
| 3241 | 2013-05-01 | EXECUTIVE PASSENGER |
| 3241 | 2013-05-13 | AIR BUS             |

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

22. Please follow instructions given below.

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

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

3 rows

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

| CARD_TYPE | CARD_COUNT |
|-----------|------------|
| GOLD      | 3          |
| INSTANT   | 2          |
| PLATINUM  | 3          |

23. Please follow instructions given below.

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

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

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

Display the records sorted in ascending order based on name.

6 rows

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

| SERIAL_NO | FIRST_NAME | MOBILE_NUMBER | EMAIL_ID         |
|-----------|------------|---------------|------------------|
| 004       | AARTHI     | 9595652530    | AARTHI@GMAIL.COM |

|     |          |            |                    |
|-----|----------|------------|--------------------|
| 008 | GANESH   | 9375237890 | GANESH@GMAIL.COM   |
| 007 | GAYATHRI | 8073245678 | GAYATHRI@GMAIL.COM |
| 001 | LATHA    | 9876543210 | LATHA@GMAIL.COM    |
| 006 | RAMESH   | 9432198760 | RAMESH@GMAIL.COM   |
| 005 | SIVA     | 9884416986 | SIVA@GMAIL.COM     |

24. Please follow instructions given below.

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

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

4 rows

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

| LIGHT_ID | FROM_LOCATION | TO_LOCATION | NO_OF_SERVICES |
|----------|---------------|-------------|----------------|
|          |               |             |                |

|      |           |           |   |
|------|-----------|-----------|---|
| 1011 | HYDERABAD | CHENNAI   | 2 |
| 1262 | HYDERABAD | CHENNAI   | 2 |
| 3148 | CHENNAI   | BENGALURU | 2 |
| 3244 | KOCHI     | CHENNAI   | 2 |

25. Please follow instructions given below.

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

Hint: Get the distinct from location and to location.

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

4 rows

```
select distinct from_location, count(to_location) as No_of_Flights from
air_flight
group by from_location order by from_location;
```

| FROM_LOCATION | NO_OF_FLIGHTS |
|---------------|---------------|
| BENGALURU     | 2             |
| CHENNAI       | 6             |
| HYDERABAD     | 2             |
| KOCHI         | 1             |

**26.**Please follow instructions given below.

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

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

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

9 rows

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

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

**27.**Please follow instructions given below.

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

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

1 row

```
select af.flight_id,af.from_location,af.to_location,af.total_seats,(af.total_seats-afd.available_seats)
```

as No\_of\_Seats\_Booked from air\_flight af join air\_flight\_details afd on af.flight\_id=afd.flight\_id where (af.total\_seats-afd.available\_seats)>(af.total\_seats\*0.1) group by flight\_id order by flight\_id, No\_of\_Seats\_Booked;

| FLIGHT_ID | FROM_LOCATION | TO_LOCATION | TOTAL_SEATS | NO_OF_SEATS_BOOKED |
|-----------|---------------|-------------|-------------|--------------------|
| 3244      | KOCHI         | CHENNAI     | 50          | 7                  |

**28. Please follow instructions given below.**

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

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

**14 rows**

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

| FLIGHT_ID | FLIGHT_DEPARTURE_DATE | FROM_LOCATION | TO_LOCATION | DURATION |
|-----------|-----------------------|---------------|-------------|----------|
| 3013      | 2013-05-04            | CHENNAI       | BENGALURU   | 01:05:00 |
| 3013      | 2013-05-06            | CHENNAI       | BENGALURU   | 01:05:00 |

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

29. Please follow instructions given below.

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

4 rows

```

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

```

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

| QuestionText                                                                                                                                                                                                                  | QuestionType | Choice1                                                                                                                           | Choice2                                                                                      | Choice3                                                | Choice4                               | Choice5                    | Grade1 | Grade2 | Grade3 | Grade4 | Grade5 | AnswerDescrip | QuestionMedia | AnswerMedia | Author | Reviewer | Is Numeric |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|--------------------------------------------------------|---------------------------------------|----------------------------|--------|--------|--------|--------|--------|---------------|---------------|-------------|--------|----------|------------|
| Please read the question carefully and choose the most appropriate option. Which of the given options are TRUE regarding 'Constraints'?                                                                                       | MCQ          | A PRIMARY KEY constraint does not automatically have a UNIQUE constraint defined on it.                                           | The UNIQUE constraint uniquely identifies each record in a database table.                   | None of the listed options                             | All listed options                    |                            | 0      | 1      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. The main reason that constraints are added to a table is:                                                                                          | MCQ          | None of the listed options                                                                                                        | Constraints gives programmers job security                                                   | Constraints ensure data integrity                      | Constraints add a level of complexity |                            | 0      | 0      | 1      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. Which of the given options are TRUE regarding 'Constraints'?                                                                                       | MCQ          | None of the listed options                                                                                                        | The NOT NULL constraint enforces a column to NOT accept NULL values.                         | All listed options                                     |                                       |                            | 0      | 0      | 1      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: If you want to select rows that satisfy at least one of the given conditions, you can use the logical operator, AND.               | MCQ          | Only statement 2                                                                                                                  | Both statement 1 and statement 2                                                             | None of the listed options                             | Only statement 1                      |                            | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. A column defined as NOT NULL can have a DEFAULT value of NULL. True or False?                                                                      | MCQ          | true                                                                                                                              | false                                                                                        |                                                        |                                       |                            | 0      | 1      |        |        |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. You can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table. State whether the above statement is TRUE or FALSE. | MCQ          | true                                                                                                                              | false                                                                                        |                                                        |                                       |                            | 1      | 0      |        |        |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. What is a primary key?                                                                                                                             | MCQ          | The primary key is a column or combination of columns whose values uniquely identify each row in the table.                       | The primary key column is a column or combination of columns whose values can be non-unique. | The primary key is a column that can have NULL values. |                                       |                            | 1      | 0      | 0      |        |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. A table must have at least one not null constraint and one unique constraint. True or False?                                                       | MCQ          | true                                                                                                                              | false                                                                                        |                                                        |                                       |                            | 0      | 1      |        |        |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. Statement 1: Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.     | MCQ          | Both statement 1 and statement 2                                                                                                  | None of the listed options                                                                   | Only statement 1                                       | Only statement 2                      |                            | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Statement 2: Arithmetic operators manipulate numeric operands.                                                                                                                                                                | MCQ          | Which of the above statements are TRUE?                                                                                           | Both statement 1 and statement 2                                                             | None of the listed options                             | Only statement 1                      | Only statement 2           | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option.                                                                                                                                                    | MCQ          | Statement 1: UNION returns all distinct rows selected by either query.                                                            | Both statement 1 and statement 2                                                             | None of the listed options                             | Only statement 1                      |                            | 0      | 1      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Statement 2: 'INTERSECT' returns all distinct rows selected by both queries.                                                                                                                                                  | MCQ          | Which of the above statements is TRUE?                                                                                            | Only statement 2                                                                             | Both statement 1 and statement 2                       | None of the listed options            | Only statement 1           | 0      | 1      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. To automatically delete rows in a child table when a parent record is deleted use:                                                                 | MCQ          | ON DELETE CASCADE                                                                                                                 | ON DELETE SET NULL                                                                           | ON DELETE ORPHAN                                       | None of the listed options            |                            | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. Statement 1: Each table can have only ONE primary key per table                                                                                    | MCQ          | Both statement 1 and statement 2                                                                                                  | Only statement 1                                                                             | Only statement 2                                       | None of the listed options            |                            | 0      | 1      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Statement 2: A primary key column can contain NULL values                                                                                                                                                                     | MCQ          | Which of the above statements are TRUE?                                                                                           | Only statement 2                                                                             | Only statement 1                                       | Only statement 2                      | None of the listed options | 0      | 1      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option.                                                                                                                                                    | MCQ          | Statement 1: A FOREIGN KEY in one table points to a PRIMARY KEY in another table.                                                 | Both statement 1 and statement 2                                                             | None of the listed options                             | Only statement 1                      | Only statement 2           | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Statement 2: If you define a CHECK constraint on a single column it allows only certain values for this column.                                                                                                               | MCQ          | Which of the given options are TRUE?                                                                                              | Both statement 1 and statement 2                                                             | None of the listed options                             | Only statement 1                      | Only statement 2           | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. Which operator returns all distinct rows selected by the first query and not the second?                                                           | MCQ          | MINUS                                                                                                                             | UNION                                                                                        | INTERSECT                                              | UNION ALL                             |                            | 1      | 0      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. A table can have more than one UNIQUE key constraint. True or False?                                                                               | MCQ          | false                                                                                                                             | true                                                                                         |                                                        |                                       |                            | 0      | 1      |        |        |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. Primary Key does allow the Null Values, where as in Unique key doesn't accept the Null values. State whether the statement is true or false        | MCQ          | true                                                                                                                              | false                                                                                        |                                                        |                                       |                            | 0      | 1      |        |        |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option.                                                                                                                                                    | MCQ          | Statement 1: AND Returns TRUE if both component conditions are TRUE. Returns FALSE if either is FALSE; otherwise returns UNKNOWN. | None of the listed options                                                                   | Only statement 1                                       | Both statement 1 and statement 2      | Only statement 2           | 0      | 1      | 0      | 0      |        | TEXT          | TEXT          |             |        |          |            |
| Statement 2: 'EXISTS' returns FALSE if a sub-query returns at least one row.                                                                                                                                                  | MCQ          | Which of the above statements are TRUE?                                                                                           | WHERE                                                                                        | RESTRICT                                               | GROUP BY                              | HAVING                     | 0      | 0      | 0      | 1      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. Which clause should you use to exclude group results?                                                                                              | MCQ          | None                                                                                                                              | Materialized View                                                                            | Normalization                                          | data Integrity                        |                            | 0      | 0      | 0      | 1      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. ON UPDATE CASCADE ensures which of the following?                                                                                                  | MCQ          | create a new table in the database                                                                                                | modify an existing table in a database                                                       | delete a table from the database                       |                                       |                            | 0      | 0      | 0      | 1      |        | TEXT          | TEXT          |             |        |          |            |
| Please read the question carefully and choose the most appropriate option. The SQL DROP TABLE clause is used to...                                                                                                            | MCQ          |                                                                                                                                   |                                                                                              |                                                        |                                       |                            | 0      | 0      | 1      |        |        | TEXT          | TEXT          |             |        |          |            |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |                                                                                           |                                                                                                                 |                                                                                                                           |                                                                                                                 |  |   |   |   |   |      |      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|--|---|---|---|---|------|------|
| Please read the question carefully and choose the most appropriate option. Which one will delete the table data as well as table structure?                                                                                                                                                                                                                                                                                                                                                                      | MCQ | DISTINCT                                                                                  | TRUNCATE                                                                                                        | REMOVE                                                                                                                    | DROP                                                                                                            |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. To remove duplicate rows from the result set of a SELECT use the following keyword:                                                                                                                                                                                                                                                                                                                                                   | MCQ | NO DUPLICATE                                                                              | DISTINCT                                                                                                        | None of the listed options                                                                                                | UNIQUE                                                                                                          |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which of the given options are TRUE about 'varchar' datatype?                                                                                                                                                                                                                                                                                                                                                                         | MCQ | All listed options                                                                        | Holds a variable length string (can contain letters, numbers, and special characters).                          | None of the listed options                                                                                                | Its maximum size is specified in parenthesis.                                                                   |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which of the following SQL statements is correct?                                                                                                                                                                                                                                                                                                                                                                                     | MCQ | SELECT CustomerName, COUNT(CustomerName) FROM Orders ORDER BY CustomerName                | SELECT CustomerName, COUNT(CustomerName) FROM Orders GROUP BY CustomerName                                      | SELECT CustomerName, COUNT(CustomerName) FROM Orders                                                                      |                                                                                                                 |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which of the following is not a valid aggregate function?                                                                                                                                                                                                                                                                                                                                                                             | MCQ | COUNT                                                                                     | COMPUTE                                                                                                         | MAX                                                                                                                       | MIN                                                                                                             |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Can you use HAVING clause and WHERE clause SQL clauses in one SQL statement?                                                                                                                                                                                                                                                                                                                                                          | MCQ | true                                                                                      | false                                                                                                           |                                                                                                                           |                                                                                                                 |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Examine the structure of the EMPLOYEES table: EMPLOYEE_ID NUMBER(4) Primary Key FIRST_NAME VARCHAR2(25) LAST_NAME VARCHAR2(25) HIRE_DATE DATE Which UPDATE statement is valid?                                                                                                                                                                                                                                                        | MCQ | UPDATE employees SET first_name = 'John' AND last_name = 'Smith' WHERE employee_id = 180; | UPDATE employees SET first_name = 'John' SET last_name = 'Smith' WHERE employee_id = 180;                       | UPDATE employees SET first_name = 'John', last_name = 'Smith' WHERE employee_id = 180;                                    | UPDATE employees SET first_name = 'John', last_name = 'Smith' WHERE employee_id = 180;                          |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The CUSTOMERS table has these columns: CUSTOMER_ID NUMBER(4) NOT NULL CUSTOMER_NAME VARCHAR2(100) NOT NULL STREET_ADDRESS VARCHAR2(150) CITY_ADDRESS VARCHAR2(50) STATE_ADDRESS VARCHAR2(50) PROVINCE_ADDRESS VARCHAR2(50) COUNTRY_ADDRESS VARCHAR2(50) POST_CODE VARCHAR2(10) CUSTOMER_PHONE VARCHAR2(20) A sale is being advertised to the customers in France. Which WHERE clause identifies customers that are located in France? | MCQ | None                                                                                      | WHERE lower(country_address) = "france"                                                                         | WHERE lower(country_address) IS "france"                                                                                  | WHERE lower(country_address) = "france"                                                                         |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. What SQL clause is used to restrict the rows returned by a query?                                                                                                                                                                                                                                                                                                                                                                     | MCQ | HAVING                                                                                    | AND                                                                                                             | FROM                                                                                                                      | WHERE                                                                                                           |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which SQL statement is used to insert a new data in a database?                                                                                                                                                                                                                                                                                                                                                                       | MCQ | INSERT NEW                                                                                | ADD                                                                                                             | UPDATE                                                                                                                    | INSERT INTO                                                                                                     |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. What does the ALTER TABLE clause do?                                                                                                                                                                                                                                                                                                                                                                                                  | MCQ | The SQL ALTER TABLE clause deletes data from a database table.                            | The SQL ALTER TABLE clause is used to insert data into a database table.                                        | The SQL ALTER TABLE clause modifies a table definition by altering, adding, or deleting table columns and/or constraints. | The SQL ALTER TABLE clause is used to create a database table                                                   |  | 0 | 0 | 1 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. What is the standard way to separate each SQL statement in database systems that allow more than one SQL statement to be executed in the same call to the server.                                                                                                                                                                                                                                                                     | MCQ | Comma                                                                                     | Colon                                                                                                           | All listed options                                                                                                        | Semicolon                                                                                                       |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which of the given options are TRUE about TCL?                                                                                                                                                                                                                                                                                                                                                                                        | MCQ | All listed options                                                                        | TCL consists of commands: COMMIT and ROLLBACK                                                                   | None of the listed options                                                                                                | TCL contains the commands which are required for Transaction Management.                                        |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The result of a SELECT statement can contain duplicate rows.                                                                                                                                                                                                                                                                                                                                                                          | MCQ | true                                                                                      | false                                                                                                           |                                                                                                                           |                                                                                                                 |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Statement1: Data types specify what the type of data can be for that particular column Statement2: Varchar is a datatype in SQL                                                                                                                                                                                                                                                                                                       | MCQ | Only statement 1                                                                          | Only statement 2                                                                                                | Both statement 1 and statement 2                                                                                          | None of the listed options                                                                                      |  | 0 | 0 | 1 | 0 | TEXT | TEXT |
| Which of the above statements is TRUE?                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | MCQ | The AS clause is used with the JOIN clause only.                                          | The AS clause is used to change the name of a column in the result set or to assign a name to a derived column. | The AS clause defines a search condition.                                                                                 | The AS clause is used to change the name of a column in the result set or to assign a name to a derived column. |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. What is the purpose of the SQL AS clause?                                                                                                                                                                                                                                                                                                                                                                                             | MCQ | limits the rows & columns returned                                                        | limits the columns data that are returned.                                                                      | limits the row data that are returned.                                                                                    | The AS clause is used to change the name of a column in the result set or to assign a name to a derived column. |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The SQL WHERE clause:                                                                                                                                                                                                                                                                                                                                                                                                                 | MCQ | NONE                                                                                      | LIKE %opt%                                                                                                      | LIKE ^opt\$                                                                                                               | LIKE opt\$                                                                                                      |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. In a LIKE clause, you could ask for any value ending in "opt" by writing                                                                                                                                                                                                                                                                                                                                                              | MCQ | Defines indexes (keys)                                                                    | Specifies links between tables, and imposes constraints between tables                                          | All listed options                                                                                                        | allows database tables to be created or deleted                                                                 |  | 0 | 0 | 1 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. DDL part of SQL does which of the following?                                                                                                                                                                                                                                                                                                                                                                                          | MCQ |                                                                                           |                                                                                                                 |                                                                                                                           |                                                                                                                 |  | 0 | 0 | 0 | 0 | TEXT | TEXT |

|                                                                                                                                                                                                                                     |     |                            |                                                                                                                                                 |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  |     |     |     |   |  |      |      |  |  |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------|-------------------------------------------------------------------|--|-----|-----|-----|---|--|------|------|--|--|--|
|                                                                                                                                                                                                                                     |     |                            | You can use aggregate functions only in the column list of the SELECT clause and in the WHERE clause of a SELECT statement.                     | You can pass column names, expressions, constants, or parameters to an aggregate function. | You can mix single row columns with aggregate functions in the column list of a SELECT statement by grouping on the single row columns. | You can use aggregate functions in any clause of a SELECT statement. |                                                                   |  |     |     |     |   |  |      |      |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which two are true about aggregate functions? (Choose two)                                                                                               | MCA |                            |                                                                                                                                                 |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  | 0   | 0.5 | 0.5 | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. A SELECT command without a WHERE clause returns?                                                                                                         | MCQ |                            | SELECT is invalid without a WHERE clause                                                                                                        | Nothing                                                                                    | All the records from a table, or information about all the records                                                                      | Both statement 1 and statement 2                                     | All the records from a table that match the previous WHERE clause |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Statement 1: GRANT, DENY and REVOKE are DCL commands. Statement 2: CREATE, ALTER, DROP, TRUNCATE are DDL commands.                                       | MCQ | None of the listed options | Only statement 1                                                                                                                                | Only statement 2                                                                           | LIKE ..... (that's six dots)                                                                                                            | LIKE _____ (that's six underscores)                                  | LIKE ^_____ (that's six underscore characters)                    |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. In a LIKE clause, you can ask for any 6 letter value by writing?                                                                                         | MCQ | DATABASE student           | CREATE ?!                                                                                                                                       | DATABASE /student                                                                          | CREATE DATABASE student                                                                                                                 |                                                                      |                                                                   |  | 0   | 0   | 0   | 1 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which of the following commands should be used to create a database named "student"?                                                                     | MCQ | Add                        | Insert                                                                                                                                          | Update                                                                                     | Alter                                                                                                                                   |                                                                      |                                                                   |  | 0   | 1   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which statement is used to query the database and retrieve selected data that match the criteria that you specify?                                       | MCQ | UPDATE                     | RETRIEVE                                                                                                                                        | SELECT                                                                                     | INSERT                                                                                                                                  |                                                                      |                                                                   |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Statement 1: The DELETE statement is used to delete columns in a table. Statement 2: The UPDATE statement is used to update existing records in a table. | MCQ | Only statement 2           | Both statement 1 and statement 2                                                                                                                | None of the listed options                                                                 | Only statement 1                                                                                                                        |                                                                      |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Which of the above statements are TRUE?                                                                                                                                                                                             | MCQ | false                      | true                                                                                                                                            |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  | 0   | 1   |     |   |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Statement 1: DCL contains the commands which protect data from unauthorized access.                                                                      | MCQ | Only statement 1           | None of the listed options                                                                                                                      | Both statement 1 and statement 2                                                           | Only statement 2                                                                                                                        |                                                                      |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Statement 2: DCL consists of 2 commands: COMMIT and ROLLBACK                                                                                                                                                                        | MCQ | Only statement 1           | None of the listed options                                                                                                                      | Both statement 1 and statement 2                                                           | Only statement 2                                                                                                                        |                                                                      |                                                                   |  | 0   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Which of the above statements are TRUE?                                                                                                                                                                                             | MCQ |                            |                                                                                                                                                 |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which two statements are true regarding the ORDER BY clause? (Choose two)                                                                                | MCA |                            | The sort is in ascending order by default.                                                                                                      | The ORDER BY clause comes last in the SELECT statement.                                    | The ORDER BY clause is executed on the client side                                                                                      | The sort is in descending order by default                           |                                                                   |  | 0.5 | 0.5 | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. SQL can be used to:                                                                                                                                      | MCQ |                            | All the listed operation can be done by SQL.                                                                                                    | query database date only.                                                                  | create database structures only.                                                                                                        | Modify the database                                                  |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which of the given options return rows when there is at least one match in both tables?                                                                  | MCQ | GROUP BY                   | WHERE                                                                                                                                           | JOIN                                                                                       | ORDER BY                                                                                                                                |                                                                      |                                                                   |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which type of join combines the results of both left and right outer joins?                                                                              | MCQ | Cross Join                 | Full Outer Join                                                                                                                                 | Inner join                                                                                 | All listed options                                                                                                                      |                                                                      |                                                                   |  | 0   | 1   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which syntax would be used to retrieve all rows in both the EMPLOYEES and DEPARTMENTS tables, even when there is no match?                               | MCQ | Self join                  | Natural join                                                                                                                                    | Outer join                                                                                 | Inner join                                                                                                                              |                                                                      |                                                                   |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. The INNER JOIN clause...                                                                                                                                 | MCQ |                            | returns only the rows from the first table, which have non-matching values with the second table in the field on which the 2 tables are joined. | returns all rows that have matching values in the field on which the 2 tables are joined.  |                                                                                                                                         |                                                                      |                                                                   |  | 0   | 0   | 1   |   |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. What is true about joining tables through an equijoin?                                                                                                   | MCQ |                            | You can join n tables (all having single column primary keys) in a SQL statement by specifying a minimum of n-1 join conditions.                | You can join a maximum of two tables through an equijoin.                                  | All listed options                                                                                                                      | You can join a maximum of two tables through an equijoin.            |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Statement 1: In case of Natural Joins, common columns are columns that have the same number of rows in both tables.                                      | MCQ | Only statement 1           | Both statement 1 and statement 2                                                                                                                | Only statement 2                                                                           | None of the listed options                                                                                                              |                                                                      |                                                                   |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Statement 2: JOIN ON syntax is much more readable and maintainable than the natural join syntax.                                                                                                                                    | MCQ | foreign keys.              | composite keys.                                                                                                                                 | determinants.                                                                              | candidate keys                                                                                                                          |                                                                      |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Which of the above statements are TRUE?                                                                                                                                                                                             | MCQ | Only statement 2           | Only statement 1                                                                                                                                | None of the listed options                                                                 | Both statement 1 and statement 2                                                                                                        |                                                                      |                                                                   |  | 0   | 0   | 0   | 1 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. In the relational model, relationships between relations or tables are created by using:                                                                 | MCQ |                            |                                                                                                                                                 |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  | 1   | 0   | 0   | 0 |  | TEXT | TEXT |  |  |  |
| Please read the question carefully and choose the most appropriate option. Statement 1: CROSS JOIN returns the Cartesian product of the sets of rows from the joined tables.                                                        | MCQ |                            |                                                                                                                                                 |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  | 0   | 0   | 1   | 0 |  | TEXT | TEXT |  |  |  |
| Statement 2: You can have multiple conditions for the ON clause just like you can in a WHERE clause.                                                                                                                                | MCQ | Only statement 2           | Only statement 1                                                                                                                                | None of the listed options                                                                 | Both statement 1 and statement 2                                                                                                        |                                                                      |                                                                   |  | 0   | 0   | 0   | 1 |  | TEXT | TEXT |  |  |  |
| Which of the above statements is TRUE?                                                                                                                                                                                              | MCQ |                            |                                                                                                                                                 |                                                                                            |                                                                                                                                         |                                                                      |                                                                   |  | 0   | 0   | 0   | 1 |  | TEXT | TEXT |  |  |  |

|                                                                                                                                                                                                                                          |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|--------------------------------------------------------------------------------------------|--------------------------------------------------|----------------------------------------------------------------------------|------------------------|--|--|---|---|---|---|------|------|
|                                                                                                                                                                                                                                          |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: The FULL OUTER JOIN will return all rows, as long as there's matching data in one of the tables.                                              |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| Statement 2: FULL OUTER JOIN includes all the rows from both the participating tables and does not select either the LEFT or RIGHT table from the JOIN key word.                                                                         | MCQ | Only statement 1                                                                           | Both statement 1 and statement 2                 | None of the listed options                                                 | Only statement 2       |  |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Which of the above statements are TRUE?                                                                                                                                                                                                  |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| Please read the question carefully and choose the most appropriate option. If table A have 10 rows and table B have 5 rows, how many rows will be returned if you perform a cartesian join on those two tables?                          | MCQ | 5                                                                                          | 15                                               | 10                                                                         | 50                     |  |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. A table may be joined to itself.                                                                                                                              | MCQ | false                                                                                      | true                                             |                                                                            |                        |  |  | 0 | 1 |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which of the given options return all rows from the left table, even if there are no matches in the right table?                                              | MCQ | RIGHT JOIN                                                                                 | JOIN                                             | CROSS JOIN                                                                 | LEFT JOIN              |  |  | 0 | 0 | 0 | 1 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. A Self Join is a type of sql join which is used to join a table to itself, particularly when the table has a FOREIGN KEY that references its own PRIMARY KEY. |     |                                                                                            |                                                  |                                                                            |                        |  |  | 0 | 1 |   |   | TEXT | TEXT |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | false                                                                                      | true                                             |                                                                            |                        |  |  |   |   |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The join is the ANSI-standard syntax used to generate a Cartesian product.                                                                                    | MCQ | ALL                                                                                        | FULL                                             | CROSS                                                                      | NATURAL                |  |  | 0 | 0 | 1 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which type of join does not require each record in the two joined tables to have a matching record?                                                           | MCQ | Outer Join                                                                                 | Equi Join                                        | Inner join                                                                 | Self join              |  |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. We refer to a join as a self-join when?                                                                                                                       | MCQ | we are using left and right join together                                                  | we are joining more than 2 tables                | we are joining table to itself                                             |                        |  |  | 0 | 0 | 1 |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. SQL joins are used to query data from two or more tables, based on _____.                                                                                     | MCQ | None of the listed options                                                                 | a relationship between certain columns in tables | a relationship between certain rows in tables.                             | All listed options     |  |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups.                                                     |     |                                                                                            |                                                  |                                                                            |                        |  |  | 1 | 0 |   |   | TEXT | TEXT |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | true                                                                                       | false                                            |                                                                            |                        |  |  |   |   |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: SQL aggregate functions return a single value, calculated from values in a column.                                                            |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| Statement 2: AVG() returns the average value                                                                                                                                                                                             | MCQ | None of the listed options                                                                 | Both statement 1 and statement 2                 | Only statement 2                                                           | Only statement 1       |  |  | 0 | 1 | 0 | 0 | TEXT | TEXT |
| Which of the above statements is TRUE?                                                                                                                                                                                                   |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Which of the given options is TRUE about LIKE clause?                                                                                                         | MCQ | The percent sign represents zero, one, or multiple characters, when used with LIKE clause. | Both the statements given                        | The underscore represents a single number or character.                    |                        |  |  | 0 | 1 | 0 |   | TEXT | TEXT |
| Please read the question carefully and choose the most inappropriate option. LIKE clause is used to compare a value to similar values using logical operators. State whether the above statement is TRUE or FALSE.                       | MCQ | false                                                                                      | true                                             |                                                                            |                        |  |  | 1 | 0 |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.                                                          |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | true                                                                                       | false                                            |                                                                            |                        |  |  | 1 | 0 |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The ROUND() function is used to round a numeric field to the nearest hundred.                                                                                 |     |                                                                                            |                                                  |                                                                            |                        |  |  | 1 | 0 |   |   | TEXT | TEXT |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | false                                                                                      | true                                             |                                                                            |                        |  |  |   |   |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The HAVING clause places conditions on the selected columns, whereas the WHERE clause places conditions on groups created by the GROUP BY clause.             |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | false                                                                                      | true                                             |                                                                            |                        |  |  | 1 | 0 |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The SUM function allows for summing the total for a numeric column.                                                                                           | MCQ | SUM function allows for summing the total for a numeric column.                            | None of the listed options                       | COUNT function is used to count the number of columns in a database table. | All listed options     |  |  | 1 | 0 | 0 | 0 | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Statement 1: Numeric functions accept numeric input and return string values.                                                                                 |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| Statement 2: Single-row functions return a single result row for every row of a queried table or view.                                                                                                                                   | MCQ | Only statement 1                                                                           | None of the listed options                       | Only statement 2                                                           | All the listed options |  |  | 0 | 0 | 1 | 0 | TEXT | TEXT |
| Which of the above statements are TRUE?                                                                                                                                                                                                  |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. The percent sign and the underscore cannot be used in combinations, when using LIKE clause.                                                                   |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | false                                                                                      | true                                             |                                                                            |                        |  |  | 1 | 0 |   |   | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. You cannot add a subquery to a SELECT clause as a column expression in the SELECT list.                                                                       |     |                                                                                            |                                                  |                                                                            |                        |  |  |   |   |   |   |      |      |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                      | MCQ | true                                                                                       | false                                            |                                                                            |                        |  |  | 0 | 1 |   |   | TEXT | TEXT |

|                                                                                                                                                                                                                                                                                                                                                                                                                  |     |                                                                                                                                                                                              |                                                                |                                                                            |                                                                                 |                    |   |   |   |   |      |      |  |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------|--------------------|---|---|---|---|------|------|--|--|
| Please read the question carefully and choose the most appropriate option. Which of the following is valid SQL for an Index?                                                                                                                                                                                                                                                                                     | MCQ | CREATE INDEX ID;                                                                                                                                                                             | REMOVE INDEX ID;                                               | CHANGE INDEX ID;                                                           | ADD INDEX ID;                                                                   |                    | 1 | 0 | 0 | 0 | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: A subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.<br><br>Statement 2: A subquery can be nested inside the WHERE or HAVING clause of an outer SELECT, INSERT, UPDATE or DELETE statement, or inside another subquery. | MCQ | Both statement 1 and statement 2                                                                                                                                                             | Only statement 2                                               | None of the listed options                                                 | Only statement 1                                                                |                    | 1 | 0 | 0 | 0 | TEXT | TEXT |  |  |
| Which of the above statements are TRUE?<br>Please read the question carefully and choose the most appropriate option. View can be removed using which command?                                                                                                                                                                                                                                                   | MCQ | DELETE VIEW                                                                                                                                                                                  | REMOVE VIEW                                                    | All listed options                                                         | DROP VIEW                                                                       |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: A view can be accessed with the use of SQL SELECT statement like a table.<br><br>Statement 2: A view can be made up by selecting data from more than one tables.                                                                                                                                                      | MCQ | Only statement 2                                                                                                                                                                             | Both statement 1 and statement 2                               | None of the listed options                                                 | Only statement 1                                                                |                    | 0 | 1 | 0 | 0 | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option. An index helps speed up SELECT queries and WHERE clauses, but it slows down data input, with UPDATE and INSERT statements.                                                                                                                                                                                                            | MCQ | true                                                                                                                                                                                         | false                                                          |                                                                            |                                                                                 |                    | 1 | 0 |   |   | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: Clustered index physically rearranges the data that users inserts in your tables.<br><br>Statement 2: There can be 2000 non-clustered index per table.                                                                                                                                                                | MCQ | Both statement 1 and statement 2                                                                                                                                                             | None of the listed options                                     | Only statement 2                                                           | Only statement 1                                                                |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Which of the above statement are TRUE?                                                                                                                                                                                                                                                                                                                                                                           | MCQ | An inline view exists only inside the FROM clause as a run-time result set.                                                                                                                  | All listed options                                             | A subquery exists only inside of the FROM clause as a run-time result set. | An inline view exists only inside of the WHERE clause as a run-time result set. |                    | 1 | 0 | 0 | 0 | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option. Which of the given options are TRUE?                                                                                                                                                                                                                                                                                                  | MCQ | false                                                                                                                                                                                        | true                                                           |                                                                            |                                                                                 |                    | 0 | 1 | 0 | 0 | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: If a subquery is not dependent on the outer query it is called a non-correlated subquery.<br><br>Statement 2: Subqueries cannot be used with the comparison operators.                                                                                                                                                | MCQ | None of the listed options                                                                                                                                                                   | Only statement 1                                               | Only statement 2                                                           | Both statement 1 and statement 2                                                |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option. A query is called correlated subquery when both the inner query and the outer query are interdependent.                                                                                                                                                                                                                               | MCQ |                                                                                                                                                                                              |                                                                |                                                                            |                                                                                 |                    |   |   |   |   |      |      |  |  |
| State whether the above statement is TRUE or FALSE.                                                                                                                                                                                                                                                                                                                                                              | MCQ | false                                                                                                                                                                                        | true                                                           |                                                                            |                                                                                 |                    | 0 | 1 |   |   | TEXT | TEXT |  |  |
| Please read the question carefully and choose the most appropriate option.<br>Statement 1: The SQL subquery is a SELECT query that is embedded in the main SELECT statement.<br><br>Statement 2: A subquery cannot return more than one rows                                                                                                                                                                     | MCQ | Both statement 1 and statement 2                                                                                                                                                             | None of the listed options                                     | Only statement 2                                                           | Only statement 1                                                                |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Which of the above statements is TRUE?                                                                                                                                                                                                                                                                                                                                                                           | MCQ | Carefully read the question and answer accordingly. The following SW process model can be represented schematically as a series of major technical activities and their associated artefacts | All of the listed options                                      | Incremental model                                                          | Concurrent model                                                                | Component assembly | 0 | 0 | 1 | 0 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. If Quality Control and Quality Assurance are compared.                                                                                                                                                                                                                                                                                                       | MCQ | Both are literally the same                                                                                                                                                                  | QA is done by the client and QC is done by the software vendor | QC is a higher activity in the management Hierarchy                        | QA is a higher activity in the management Hierarchy                             |                    | 0 | 0 | 1 | 0 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. To produce a good quality product, process should be                                                                                                                                                                                                                                                                                                         | MCQ | None                                                                                                                                                                                         | Rigorous                                                       | Complex                                                                    | Efficient                                                                       |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Software processes can be constructed out of pre-existing software patterns to meet the needs of a software project. State True or False                                                                                                                                                                                                                     | MCQ | false                                                                                                                                                                                        | true                                                           |                                                                            |                                                                                 |                    | 0 | 1 |   |   | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Who is essentially responsible for the quality of a product?                                                                                                                                                                                                                                                                                                 | MCQ | Development Manager                                                                                                                                                                          | Customer                                                       | QA Manager                                                                 |                                                                                 |                    | 1 | 0 | 0 |   | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Data structure suitable for the application is discussed in ?                                                                                                                                                                                                                                                                                                | MCQ | procedural design                                                                                                                                                                            | architectural design                                           | interface design                                                           | data design                                                                     |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Using software process improvement model will help a company                                                                                                                                                                                                                                                                                                 | MCQ | To decrease the defect rate                                                                                                                                                                  | All of the listed options                                      | To increase profitability                                                  | To decrease development time                                                    |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. The object relationship pair of data model is represented graphically by using                                                                                                                                                                                                                                                                               | MCQ | All of the listed options                                                                                                                                                                    | Flow chart                                                     | Data flow diagram                                                          | Entity relationship diagram                                                     |                    | 0 | 0 | 0 | 1 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Which one is the most important feature of spiral model                                                                                                                                                                                                                                                                                                      | MCQ | Efficiency management                                                                                                                                                                        | Performance Management                                         | Risk Management                                                            | Quality management                                                              |                    | 0 | 0 | 1 | 0 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Process models are described as agile because they                                                                                                                                                                                                                                                                                                           | MCQ | do not waste development time on planning activities                                                                                                                                         | emphasize moneuvability and adaptability                       | eliminate the need for cumbersome documentation                            | make extensive use of prototype creation                                        |                    | 0 | 1 | 0 | 0 | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. People who perform software quality assurance must look at the software from the customer's perspective.                                                                                                                                                                                                                                                     | MCQ | false                                                                                                                                                                                        | true                                                           |                                                                            |                                                                                 |                    | 0 | 1 |   |   | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. In software quality assurance work there is no difference between software verification and software validation. State True/False                                                                                                                                                                                                                            | MCQ | false                                                                                                                                                                                        | true                                                           |                                                                            |                                                                                 |                    | 1 | 0 |   |   | TEXT | TEXT |  |  |
| Carefully read the question and answer accordingly. Software product can be manufactured using the same technologies used for other engineering artifacts. State True or False                                                                                                                                                                                                                                   | MCQ | true                                                                                                                                                                                         | false                                                          |                                                                            |                                                                                 |                    | 0 | 1 |   |   | TEXT | TEXT |  |  |

|                                                                                                                                                                                                        |     |                                                                |                                                        |                                                          |                                                    |                 |  |   |   |   |   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----------------------------------------------------------------|--------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------|-----------------|--|---|---|---|---|
| Carefully read the question and answer accordingly. A key concept of quality control is that all work products                                                                                         | MCQ | have measurable specifications for process outputs             | are thoroughly tested before delivery to the customer  | are delivered on time and under budget                   | have complete documentation                        |                 |  |   |   |   |   |
| Carefully read the question and answer accordingly. What are the qualities of a good SW?                                                                                                               | MCQ | Portability                                                    | Reusability                                            | All of the listed options                                | Inter Operability                                  |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. Software Safety is a quality assurance activity that focuses on hazards that                                                                       | MCQ | may result from user input errors                              | prevent profitable marketing of the final product      | may cause an entire system to fail                       | affect the reliability of a software component     |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. The goal of quality assurance is to provide management with the data needed to determine which software engineers are producing the most defects.  | MCQ | false                                                          | true                                                   |                                                          |                                                    |                 |  | 1 | 0 |   |   |
| Carefully read the question and answer accordingly. Variation control in the context of software engineering involves controlling variation in the                                                     | MCQ | process applied                                                | All of the listed options                              | resources expended                                       | product quality attributes                         |                 |  | 0 | 1 | 0 | 0 |
| Carefully read the question and answer accordingly. What is used to measure the characteristics of the documentation and code                                                                          | MCQ | Process metrics                                                | Software Quality metrics                               | Product metrics                                          | None of the listed options                         |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. The purpose of software reviews is to uncover errors in work products so they can be removed before moving on to the next phase of development.    | MCQ | false                                                          | true                                                   |                                                          |                                                    |                 |  | 0 | 1 |   |   |
| Carefully read the question and answer accordingly. In object oriented design of software , objects have                                                                                               | MCQ | attributes, name and operations                                | None of the listed options                             | operations and names only                                | attributes and names only                          |                 |  | 1 | 0 | 0 | 0 |
| Carefully read the question and answer accordingly. Which of these criteria are useful in assessing the effectiveness of a particular design notation                                                  | MCQ | modularity                                                     | size                                                   | maintainability                                          | simplicity                                         | maintainability |  | 0 | 0 | 0 | 0 |
| Carefully read the question and answer accordingly. Which of the following is the correct definition for DFD                                                                                           | MCQ | The primary output of the system design phase                  | The modern version of flowchart                        | Mainly used at systems specification stages              | All of the listed options                          |                 |  | 1 | 0 | 0 | 0 |
| Carefully read the question and answer accordingly. Which of the following comments about object oriented design of software, is not true                                                              | MCQ | Classes are defined based on the attributes of objects         | An object can belong to two classes                    | classes are always different                             | Objects inherit the properties of class            |                 |  | 0 | 1 | 0 | 0 |
| Carefully read the question and answer accordingly. In system design, we do following                                                                                                                  | MCQ | Hardware design after software                                 | Software design after hardware                         | Parallel hardware and software design                    | No hardware design needed                          |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. Which of these is a graphical notation for depicting procedural detail                                                                             | MCQ | flowchart                                                      | ER diagram                                             | decision table                                           | process diagram                                    |                 |  | 1 | 0 | 0 | 0 |
| Carefully read the question and answer accordingly. The term module in the design phase refers to                                                                                                      | MCQ | Sub programs                                                   | Procedures                                             | All of the listed options                                | Functions                                          |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. Informational cohesion is a realization of                                                                                                         | MCQ | structured programming                                         | Modularity                                             | Concurrency                                              | Data abstraction                                   |                 |  | 0 | 0 | 0 | 1 |
| Carefully read the question and answer accordingly. The work products produced during requirement elicitation will vary                                                                                | MCQ | size of the product being built                                | size of the budget                                     | stakeholders needs                                       | software process being used                        |                 |  | 1 | 0 | 0 | 0 |
| Carefully read the question and answer accordingly. Object-oriented analysis techniques can be used to identify and refine user tasks                                                                  | MCQ | true                                                           | false                                                  |                                                          |                                                    |                 |  | 0 | 1 |   |   |
| Carefully read the question and answer accordingly. Change cannot be easily accommodated in most software systems, unless the system was designed with change in mind. State True/False                | MCQ | false                                                          | true                                                   |                                                          |                                                    |                 |  | 0 | 1 |   |   |
| Carefully read the question and answer accordingly. Software Engineering is the systematic approach to the development, operation, maintenance and retirement of software. This definition is given by | MCQ | Bauer                                                          | Boehm                                                  | Charles Babbage                                          | IEEE                                               |                 |  | 0 | 0 | 0 | 1 |
| Carefully read the question and answer accordingly. Which of the items listed below is not one of the software engineering layers                                                                      | MCQ | Tools                                                          | Manufacturing                                          | Methods                                                  | Process                                            |                 |  | 0 | 1 | 0 | 0 |
| Carefully read the question and answer accordingly. Software engineering umbrella activities are only applied during the initial phases of software development projects. State True or False          | MCQ | true                                                           | false                                                  |                                                          |                                                    |                 |  | 0 | 1 |   |   |
| Carefully read the question and answer accordingly. Which phase is not available in s/w life cycle                                                                                                     | MCQ | Coding                                                         | Specifications                                         | Design                                                   | Installation & Maintenance                         |                 |  | 0 | 0 | 0 | 1 |
| Carefully read the question and answer accordingly. The Prototype is a                                                                                                                                 | MCQ | Mini model of existing system                                  | Working model of existing system                       | Mini model of processed system                           | None of the listed options                         |                 |  | 0 | 1 | 0 | 0 |
| Carefully read the question and answer accordingly. Which of the following is not Risk characteristic                                                                                                  | MCQ | Neither intrinsically good nor bad                             | Something to fear but not something to manage          | Probability of loss                                      | Inherent in every project                          |                 |  | 0 | 1 | 0 | 0 |
| Carefully read the question and answer accordingly. Management of software development is dependent upon                                                                                               | MCQ | People                                                         | Process                                                | All of the listed options                                | Product                                            |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. The best way to conduct a requirements validation review is to                                                                                     | MCQ | send them to the design team and see if they have any concerns | examine the system model for errors                    | use a checklist of questions to examine each requirement | have the customer look over the requirements       |                 |  | 0 | 0 | 1 | 0 |
| Carefully read the question and answer accordingly. Software engineering aims at developing                                                                                                            | MCQ | Reliable Software                                              | None of the listed options                             | Cost Effective Software                                  | Reliable and cost effective Software               |                 |  | 0 | 0 | 0 | 1 |
| Carefully read the question and answer accordingly. Major component of Risk Analysis are                                                                                                               | MCQ | The potential loss impact associated with the event            | The probability that the negative event will occur and | The potential loss impact associated with the event      | The probability that the negative event will occur |                 |  | 0 | 1 | 0 | 0 |
| Carefully read the question and answer accordingly. Milestones are used to                                                                                                                             | MCQ | Know the status of the project                                 | Know the cost of the project                           | Know the user expectations                               | None of the listed options                         |                 |  | 1 | 0 | 0 | 0 |
| Carefully read the question and answer accordingly. The review is one of the methods of quality assurance. The methods are                                                                             | MCQ | Walkthrough                                                    | Inspection                                             | Testing                                                  | All of the listed options                          |                 |  | 0 | 0 | 0 | 1 |
| Carefully read the question and answer accordingly. Software engineering approach is used to achieve                                                                                                   | MCQ | Better performance of hw...                                    | Reusable software                                      | Error free softw...                                      | Quality software product                           |                 |  | 0 | 0 | 0 | 1 |

|                                                                                                                                                                                                                     |     |                                                                           |                                                                                                                                  |                                                                                                                    |                                                                            |                                    |  |       |       |   |       |  |      |      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|------------------------------------|--|-------|-------|---|-------|--|------|------|
|                                                                                                                                                                                                                     |     |                                                                           |                                                                                                                                  |                                                                                                                    |                                                                            |                                    |  |       |       |   |       |  |      |      |
| Carefully read the question and answer accordingly. Which is not the responsibility of customer/ user of the software                                                                                               | MCQ | Prepare resource plan                                                     | Plan how and by whom each acceptance activity will be performed                                                                  | Plan resources for providing information on which to base acceptance decisions                                     | Prepare the acceptance plan                                                |                                    |  | 1     | 0     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. A stakeholder is anyone who will purchase the completed software system under development. State True/False                                                     | MCQ | false                                                                     | true                                                                                                                             |                                                                                                                    |                                                                            |                                    |  | 1     | 0     |   |       |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. Project risk factor is considered in                                                                                                                            | MCQ | Water fall                                                                | Spiral                                                                                                                           | All of the listed options                                                                                          | Prototypy                                                                  |                                    |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. The prototyping model of software development is                                                                                                                | MCQ | The best approach to use for projects with large development teams        | A risky model that rarely produces a meaningful product                                                                          | A useful approach when a customer cannot define requirements clearly                                               | A reasonable approach when requirements are well defined                   |                                    |  | 0     | 0     | 1 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. Control flow diagrams are                                                                                                                                       | MCQ | required for all systems.                                                 | used in place of data flow diagrams.                                                                                             | Relationships between data objects                                                                                 | needed to model event driven systems.                                      | useful for mod                     |  | 0     | 0     | 1 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. A data model consists of the following information                                                                                                              | MCQ | All of the listed options                                                 | The attributes that describe data object                                                                                         | Relationships that connect data objects to one another                                                             | Data Object                                                                |                                    |  | 1     | 0     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. If following is not a UML diagram used creating a system analysis model                                                                                         | MCQ | Activity diagram                                                          | Class diagram                                                                                                                    | State diagram                                                                                                      | Dataflow diagram                                                           |                                    |  | 1     | 0     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. If requirements are frequently changing, which model is best suited                                                                                             | MCQ | RAD                                                                       | Prototype                                                                                                                        | Water fall                                                                                                         | Spiral                                                                     |                                    |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. A data flow diagram must be accompanied by descriptive text in order to describe the functional requirements for a software product. State True/False           | MCQ | true                                                                      | false                                                                                                                            |                                                                                                                    |                                                                            |                                    |  | 1     | 0     |   |       |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. The incremental model of software development is                                                                                                                | MCQ | The best approach to use for projects with large development teams.       | A good approach when a working core product is required quickly                                                                  | A reasonable approach when requirements are well defined                                                           | A revolutionary model that is not used for commercial products             |                                    |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. Which is not a software life cycle model                                                                                                                        | MCQ | Spiral                                                                    | Capability Maturity Model                                                                                                        | Water fall                                                                                                         | Prototype                                                                  |                                    |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. If requirements are understandable, easy, defined, which model is best suited                                                                                   | MCQ | Prototype                                                                 | None                                                                                                                             | Water fall                                                                                                         | Spiral                                                                     |                                    |  | 0     | 0     | 1 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. The entity relationship diagram                                                                                                                                 | MCQ | depicts functions that transform the data flow                            | depicts relationships between data objects                                                                                       | indicates how data are transformed by the system                                                                   | indicates system reactions to external events                              |                                    |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. What exactly Baseline means                                                                                                                                     | MCQ | A quantitative measure of the current level of performance                | A single software product that may or may not support a business function                                                        | A test or analysis conducted after an application is moved into production                                         | None of the listed options                                                 |                                    |  | 1     | 0     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. Which of these are valid software configuration items?                                                                                                          | MCQ | test data                                                                 | software tools                                                                                                                   | executable programs                                                                                                | All of the listed options                                                  | documentation                      |  | 0     | 0     | 0 | 1     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. The primary purpose of configuration status reporting is to                                                                                                     | MCQ | make sure that change information is communicated to all affected parties | evaluate the performance of software developers and organizations                                                                | None of the listed options                                                                                         | allow revision of project schedules and cost estimates by project managers |                                    |  | 1     | 0     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. When software configuration management is a formal activity, the software configuration audit is conducted by the                                               | MCQ | senior managers                                                           | development team                                                                                                                 | quality assurance group                                                                                            | testing specialists                                                        |                                    |  | 0     | 0     | 1 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. A new _____ is defined when major changes have been made to one or more configuration objects.                                                                  | MCQ | item                                                                      | version                                                                                                                          | entity                                                                                                             | variant                                                                    |                                    |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. What is configuration management in software engineering                                                                                                        | MCQ | management of the configurable components in a system                     | in object-oriented programming, the management of objects that control the configuration of some other function(s) in the system | the identification of the configuration of a system at discrete points in time due to changes to the configuration | overall management of the design of the system                             |                                    |  | 0     | 0     | 1 | 0     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. In requirements validation the requirements model is reviewed to ensure its technical feasibility. State True/False                                             | MCQ | false                                                                     | true                                                                                                                             |                                                                                                                    |                                                                            |                                    |  | 1     | 0     |   |       |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. A configuration object is a _____ created by a software engineer during some phase of the software development process.                                         | MCQ | All of the listed options                                                 | program data structure                                                                                                           | unit of information                                                                                                | a software component                                                       |                                    |  | 0     | 0     | 0 | 1     |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. The ability to track relationships and changes to configuration objects is one of the most important features of the SCM repository.                            | MCQ | true                                                                      | false                                                                                                                            |                                                                                                                    |                                                                            |                                    |  | 1     | 0     |   |       |  | TEXT | TEXT |
| Carefully read the question and answer accordingly. Which of the following tasks is not part of software configuration management?                                                                                  | MCQ | change control                                                            | version control                                                                                                                  | statistical quality control                                                                                        | reporting                                                                  |                                    |  | 0     | 0     | 1 | 0     |  | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. An input field takes the birth year of the user ranging from 1960 to 1995. The boundary values for testing this field are?               | MCQ | 0,1960,1995                                                               | 1959, 1960, 1961, 1994, 1995, 1996                                                                                               | 1960, 1995, 1996                                                                                                   | 0, 1959, 1960, 1961, 1994, 1995, 1996                                      | 1959, 1960, 1961, 1994, 1995, 1996 |  | 0     | 1     | 0 | 0     |  | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Test scenarios have to be written with the consideration of ?                                                                            | MCA | Business rules                                                            | Functional standards                                                                                                             | None of the listed options                                                                                         | Non functional standards                                                   |                                    |  | 0.333 | 0.333 | 0 | 0.333 |  | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. State whether True or False. Testers should be involved in reviewing documents as soon as drafts are available in the development cycle. | MCQ | true                                                                      | false                                                                                                                            |                                                                                                                    |                                                                            |                                    |  | 1     | 0     |   |       |  | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Alternate flows can be tested by themselves (State True or false)                                                                        | MCQ | true                                                                      | false                                                                                                                            |                                                                                                                    |                                                                            |                                    |  | 0     | 1     |   |       |  | TEXT | TEXT |

|                                                                                                                                                                                   |     |                                                                                                                   |                                                                                                                                                            |                                                                                                                                                        |                                                                                                                                                        |                   |      |       |       |      |       |      |      |      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|------|-------|-------|------|-------|------|------|------|
| Please read the question carefully and choose the most appropriate option. We derive _____ by using the test design techniques                                                    | MCQ | All the listed options                                                                                            | Test Scenario                                                                                                                                              | Test case                                                                                                                                              | None of the listed options                                                                                                                             | Test condition    | 0    | 0     | 1     | 0    | 0     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. State whether True or False. A use case can result into more than one scenario.                        | MCQ | true                                                                                                              | false                                                                                                                                                      |                                                                                                                                                        |                                                                                                                                                        |                   | 1    | 0     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Test data preparation data is done during _____?                                                       | MCQ | Test condition defining process                                                                                   | Test Development process                                                                                                                                   | Test Execution process                                                                                                                                 | Test Scenario identification process                                                                                                                   |                   | 0    | 1     | 0     | 0    |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. For a given set of boundaries, how many boundary values are possible?                                  | MCQ | 2                                                                                                                 | None of the listed options                                                                                                                                 | 4                                                                                                                                                      | 8                                                                                                                                                      | 6                 |      | 0     | 0     | 0    | 0     | 1    | TEXT | TEXT |
| Please read the question carefully and choose the most appropriate option. Test Scenarios have case specific data assigned to them (State True or False)                          | MCQ | true                                                                                                              | false                                                                                                                                                      |                                                                                                                                                        |                                                                                                                                                        |                   | 0    | 1     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Which of the following statements is/are true?                                                         | MCA | Test scenario involves the expected results.                                                                      | Test case includes the steps to execute.                                                                                                                   | Test scenario defines the setup to perform the tests                                                                                                   | Test cases are developed from Test conditions.                                                                                                         | Test case incl    | 0    | 0.5   | 0     | 0    | 0.5   | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Test conditions can be valid or invalid (State True or False)                                          | MCQ | false                                                                                                             | true                                                                                                                                                       |                                                                                                                                                        |                                                                                                                                                        |                   | 0    | 1     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Which of the below is not an activity involved in Test execution process?                              | MCQ | Test data setup                                                                                                   | Retesting of defects                                                                                                                                       | Build verification process                                                                                                                             | Test case execution                                                                                                                                    | Defect Trackin    | 1    | 0     | 0     | 0    | 0     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. The conditions that need to be verified by the tester after an activity is performed are called _____? | MCQ | Exceptions                                                                                                        | Post condition                                                                                                                                             | Pre condition                                                                                                                                          | Triggers                                                                                                                                               |                   | 0    | 1     | 0     | 0    |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. State whether True or False. Triage meeting is done before fixing the defect.                          | MCQ | false                                                                                                             | true                                                                                                                                                       |                                                                                                                                                        |                                                                                                                                                        |                   | 0    | 1     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Which is the correct order to be followed for a Build Verification Process?                            | MCQ | A. Build the compiled code into software B. Add the release notes C. Perform Smoke/ Sanity Test D. Test Execution |                                                                                                                                                            | A. Review the code B. Build the compiled code into software C. Perform Smoke/ Sanity Test D. Test Execution                                            |                                                                                                                                                        |                   | 0    | 0     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. A defect is found after retest. What are all the possible stages this defect may undergo?              | MCQ | Reopen, Fixed, Closed                                                                                             | Open, Fixed, Reopen, Closed                                                                                                                                | Reopen, Fixed                                                                                                                                          | Deferred, Open, Fixed, Reopen, Closed                                                                                                                  |                   | 1    | 0     | 0     | 0    | 0     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Which is not a major task of test implementation and execution?                                        | MCQ | Verifying that the test environment has been set up correctly                                                     | Develop and prioritizing test cases, creating test data, writing test procedures and optionally, preparing test harness and writing automated test scripts | Verifying that the test environment has been set up correctly and optionally, preparing test logs against the exit criteria specified in test planning | Checking test logs against the exit criteria specified in test planning                                                                                | Logging the ou    | 0    | 0     | 0     | 0    | 1     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. State whether True or False. Release notes are prepared by developer/ development team.                | MCQ | true                                                                                                              | false                                                                                                                                                      |                                                                                                                                                        |                                                                                                                                                        |                   | 0    | 1     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Which is not a major task of test implementation and execution?                                        | MCQ |                                                                                                                   | Develop and prioritizing test cases, creating test data, writing test procedures and optionally, preparing test harness and writing automated test scripts | Logging the outcome of test execution and recording the identities and versions of the software under test, test tools and testware                    | Verifying that the test environment has been set up correctly and optionally, preparing test logs against the exit criteria specified in test planning | Checking test     | 0    | 0     | 0     | 0    | 1     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. What are the subsequent states that a new defect can undergo?                                          | MCA | Closed                                                                                                            | Rejected                                                                                                                                                   | Deferred                                                                                                                                               | Fixed                                                                                                                                                  | Open              | 0    | 0.333 | 0.333 | 0    | 0.333 | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. State whether True or False. Developer has to ensure that the pre requisite of each test case are met. | MCQ | True                                                                                                              | false                                                                                                                                                      |                                                                                                                                                        |                                                                                                                                                        |                   | 0    | 1     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. What are the action items if an application does not behave as expected?                               | MCA | Update status of the defect                                                                                       | Execute next step of same test case                                                                                                                        | Log defect                                                                                                                                             | Retest                                                                                                                                                 |                   | 0.5  | 0     | 0.5   | 0    |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Software testing ensures which of the below?                                                           | MCQ | Use of proper test approach                                                                                       | Usage of design architecture                                                                                                                               | Proper causal analysis                                                                                                                                 | Requirement satisfaction and usage of best design architecture                                                                                         | None of the lis   | 0    | 0     | 0     | 1    | 0     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Match the following. A) Self review B) Formal review C) Informal review                                | MCQ | A - 3, B - 1, C - 2                                                                                               | A - 2, B - 1, C - 3                                                                                                                                        | A - 2, B - 2, C - 1                                                                                                                                    | A - 3, B - 2, C - 1                                                                                                                                    | A - 1, B - 2, C   | 0    | 0     | 0     | 1    | 0     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Review of Test case Artifact is done with the help of?                                                 | MCQ | Reviewer                                                                                                          | Self review                                                                                                                                                | Author                                                                                                                                                 | Peer review                                                                                                                                            | Checklist         | 0    | 0     | 0     | 0    | 1     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. In causal analysis which attributes among below assist in analyzing the effect?                        | MCA | Failures                                                                                                          | Cause                                                                                                                                                      | Requirement gathering                                                                                                                                  | Reason                                                                                                                                                 | Test Approach     | 0    | 0.5   | 0     | 0.5  | 0     | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Selenium tools helps to develop Automated test scripts                                                 | MCQ | true                                                                                                              | false                                                                                                                                                      |                                                                                                                                                        |                                                                                                                                                        |                   | 1    | 0     |       |      |       | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Which of the statements is applicable to software testing?                                             | MCA | Helps to provide a reliable system                                                                                | Helps prevent the defects                                                                                                                                  | None of the listed options                                                                                                                             | Helps to identify completeness of the software                                                                                                         | Helps in identifi | 0.25 | 0.25  | 0     | 0.25 | 0.25  | TEXT | TEXT |      |
| Please read the question carefully and choose the most appropriate option. Test environment check up is part of _____                                                             | MCQ | Test Execution                                                                                                    | None of the listed options                                                                                                                                 | Test Scenario                                                                                                                                          | Test Development                                                                                                                                       | Test Design       | 1    | 0     | 0     | 0    | 0     | TEXT | TEXT |      |

|                                                                                                                                                                   |     |                                                                                           |                            |                                                                                                                               |                                                                                                                               |               |      |      |   |      |      |      |      |  |  |  |  |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-------------------------------------------------------------------------------------------|----------------------------|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|---------------|------|------|---|------|------|------|------|--|--|--|--|
| Please read the question carefully and choose the most appropriate option. What are the possible causes for ending up into 0.1 % defective application?           | MCA | Developers tend to neglect test approach to the developed product.                        | Defective code             | Less knowledge on development language                                                                                        | Lack of domain knowledge                                                                                                      | Misunderstood | 0.25 | 0.25 | 0 | 0.25 | 0.25 | TEXT | TEXT |  |  |  |  |
| Please read the question carefully and choose the most appropriate option. Which of the following map the corresponding phases from SDLC with STLC.               | MCQ | Requirement Analysis - Test Planning Design and Code - Test Design Testing - Unit Testing | None of the listed options | Requirement Analysis - Test Planning Design and Code - Test Design Testing - Component Integration testing and System testing | Requirement Analysis - Test Planning Design and Code - Test Design Testing - Component Integration testing and System testing | Requirement / | 0    | 0    | 0 | 1    | 0    | TEXT | TEXT |  |  |  |  |
| Please read the question carefully and choose the most appropriate option. State whether true or false. QC is used for logging the outcome of the test execution. | MCQ | false                                                                                     | true                       |                                                                                                                               |                                                                                                                               |               | 0    | 1    |   |      |      | TEXT | TEXT |  |  |  |  |

# BANK

```
create database bank;
```

```
use bank;
```

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

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

```
CREATE TABLE account_master
(account_number VARCHAR(255),
customer_number VARCHAR(255),
branch_id VARCHAR(255),
opening_balance INT(20),
account_opening_date DATE,
account_type VARCHAR(10),
account_status VARCHAR(10),
PRIMARY KEY (account_number),
FOREIGN KEY (customer_number) references customer_master(customer_number),
FOREIGN KEY (branch_id) references branch_master(branch_id));
```

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

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

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

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

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

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

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

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

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

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

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

insert into customer_master values('C00010','SHANKAR',NULL, 'NAIR', 'CHENNAI', '8765489076',
 'SERVICE', '1976-07-12');
```

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

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

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

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

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

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

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

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

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

```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```

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

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

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

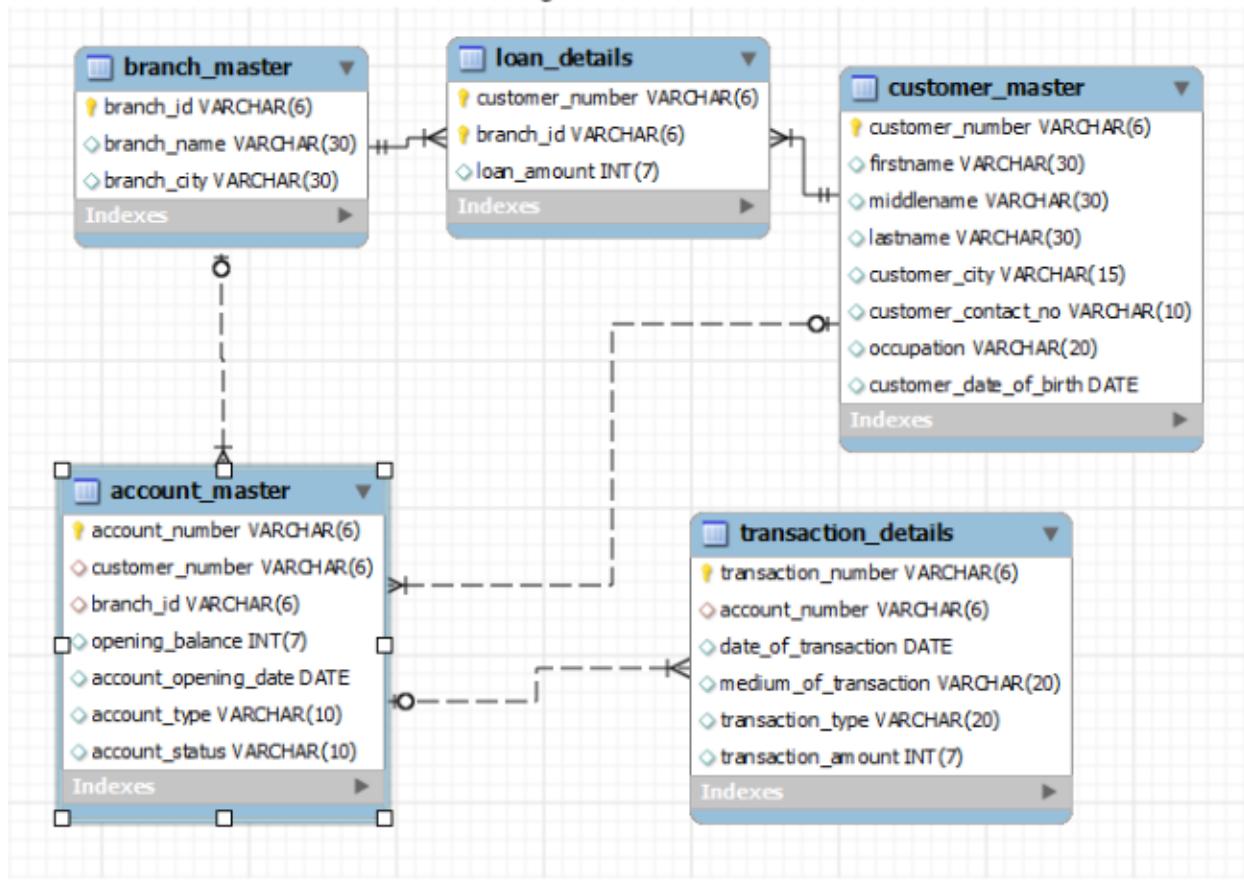
insert into transaction_details values('T00011', 'A00002', '2013-03-25', 'CASH'
 , 'WITHDRAWAL' ,7000);

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

insert into Loan_details values('C00001', 'B00001', 100000);
insert into Loan_details values('C00002', 'B00002', 200000);
insert into Loan_details values('C00009', 'B00008', 400000);
insert into Loan_details values('C00010', 'B00009', 500000);
insert into Loan_details values('C00001', 'B00003', 600000);
insert into Loan_details values('C00002', 'B00001', 600000);
```

---

ANSI SQL Bank Management Schema



## CUSTOMER MASTER

## ACCOUNT MASTER

| account_number | customer_number | branch_id | opening_balance | account_opening_date | account_type | account_status |
|----------------|-----------------|-----------|-----------------|----------------------|--------------|----------------|
| A00001         | C00001          | B00001    | 1000            | 2012-12-15           | SAVING       | ACTIVE         |
| A00002         | C00002          | B00001    | 1000            | 2012-06-12           | SAVING       | ACTIVE         |
| A00003         | C00003          | B00002    | 1000            | 2012-05-17           | SAVING       | ACTIVE         |
| A00004         | C00002          | B00005    | 1000            | 2013-01-27           | SAVING       | ACTIVE         |
| A00005         | C00006          | B00006    | 1000            | 2012-12-17           | SAVING       | ACTIVE         |
| A00006         | C00007          | B00007    | 1000            | 2010-08-12           | SAVING       | SUSPENDED      |
| A00007         | C00007          | B00001    | 1000            | 2012-10-02           | SAVING       | ACTIVE         |
| A00008         | C00001          | B00003    | 1000            | 2009-11-09           | SAVING       | TERMINATED     |
| A00009         | C00003          | B00007    | 1000            | 2008-11-30           | SAVING       | TERMINATED     |
| A00010         | C00004          | B00002    | 1000            | 2013-03-01           | SAVING       | ACTIVE         |
| NULL           | NULL            | NULL      | NULL            | NULL                 | NULL         | NULL           |

## BRANCH MASTER

| branch_id | branch_name           | branch_city |
|-----------|-----------------------|-------------|
| B00001    | ASAF ALI ROAD         | DELHI       |
| B00002    | NEW DELHI MAIN BRANCH | DELHI       |
| B00003    | DELHI CANTT           | DELHI       |
| B00004    | JASOLA                | DELHI       |
| B00005    | MAHIM                 | MUMBAI      |
| B00006    | VILE PARLE            | MUMBAI      |
| B00007    | MANDVI                | MUMBAI      |
| B00008    | JADAVPUR              | KOLKATA     |
| B00009    | KODAMBAKKAM           | CHENNAI     |
| NULL      | NULL                  | NULL        |

## LOAN DETAILS

| customer_number | branch_id | loan_amount |
|-----------------|-----------|-------------|
| C00001          | B00001    | 100000      |
| C00002          | B00002    | 200000      |
| C00009          | B00008    | 400000      |
| C00010          | B00009    | 500000      |
| C00001          | B00003    | 600000      |
| C00002          | B00001    | 600000      |

## TRANSACTION DETAILS

| transaction_number | account_number | date_of_transaction | medium_of_transaction | transaction_type | transaction_amount |
|--------------------|----------------|---------------------|-----------------------|------------------|--------------------|
| T00001             | A00001         | 2013-01-01          | CHEQUE                | DEPOSIT          | 2000               |
| T00002             | A00001         | 2013-02-01          | CASH                  | WITHDRAWAL       | 1000               |
| T00003             | A00002         | 2013-01-01          | CASH                  | DEPOSIT          | 2000               |
| T00004             | A00002         | 2013-02-01          | CASH                  | DEPOSIT          | 3000               |
| T00005             | A00007         | 2013-01-11          | CASH                  | DEPOSIT          | 7000               |
| T00006             | A00007         | 2013-01-13          | CASH                  | DEPOSIT          | 9000               |
| T00007             | A00001         | 2013-03-13          | CASH                  | DEPOSIT          | 4000               |
| T00008             | A00001         | 2013-03-14          | CHEQUE                | DEPOSIT          | 3000               |
| T00009             | A00001         | 2013-03-21          | CASH                  | WITHDRAWAL       | 9000               |
| T00010             | A00001         | 2013-03-22          | CASH                  | WITHDRAWAL       | 2000               |
| T00011             | A00002         | 2013-03-25          | CASH                  | WITHDRAWAL       | 7000               |
| T00012             | A00007         | 2013-03-26          | CASH                  | WITHDRAWAL       | 2000               |
| NULL               | NULL           | NULL                | NULL                  | NULL             | NULL               |

## QUERIES

1. Write a query to display account number, customer's number, customer's firstname, lastname, account opening date. Display the records sorted in ascending order based on account number.

```
SELECT a.account_number,c.customer_number,c.firstname,c.lastname,a.account_number
FROM customer_master c JOIN account_master a ON
c.customer_number=a.customer_number
ORDER BY a.account_number;
```

| account_number | customer_number | firstname | lastname | account_opening_date |
|----------------|-----------------|-----------|----------|----------------------|
| A00001         | C00001          | RAMESH    | SHARMA   | 2012-12-15           |
| A00002         | C00002          | AVINASH   | MINHA    | 2012-06-12           |
| A00003         | C00003          | RAHUL     | RASTOGI  | 2012-05-17           |
| A00004         | C00002          | AVINASH   | MINHA    | 2013-01-27           |
| A00005         | C00006          | CHITRESH  | BARWE    | 2012-12-17           |
| A00006         | C00007          | AMIT      | BORKAR   | 2010-08-12           |
| A00007         | C00007          | AMIT      | BORKAR   | 2012-10-02           |
| A00008         | C00001          | RAMESH    | SHARMA   | 2009-11-09           |
| A00009         | C00003          | RAHUL     | RASTOGI  | 2008-11-30           |
| A00010         | C00004          | PARUL     | GANDHI   | 2013-03-01           |

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

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

| cust_count |
|------------|
| 4          |

3. Write a query to display the customer number, customer firstname, account number for the customer's whose accounts were created after 15th of any month. Display the records sorted in ascending order based on customer number and then by account number.

```
SELECT c.customer_number,c.firstname,a.account_number FROM account_master a join
customer_master c ON c.customer_number=a.customer_number WHERE
day(a.account_opening_date)>'15' ORDER BY c.customer_number,a.account_number;
```

| customer_number | firstname | account_number |
|-----------------|-----------|----------------|
| C00002          | AVINASH   | A00004         |
| C00003          | RAHUL     | A00003         |
| C00003          | RAHUL     | A00009         |
| C00006          | CHITRESH  | A00005         |

**4. Write a query to display customer number, customer's first name, account number where the account status is terminated. Display the records sorted in ascending order based on customer number and then by account number.**

```
SELECT c.customer_number,c.firstname,a.account_number
FROM account_master a JOIN customer_master c
ON c.customer_number=a.customer_number
WHERE a.account_status='Terminated'
ORDER BY c.customer_number,a.account_number;
```

| customer_number | firstname | account_number |
|-----------------|-----------|----------------|
| C00001          | RAMESH    | A00008         |
| C00003          | RAHUL     | A00009         |

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

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

| transaction_type | Trans_count |
|------------------|-------------|
| DEPOSIT          | 3           |
| WITHDRAWAL       | 3           |

**6. Write a query to display the number of customers who have registration but no account in the bank. Give the alias name as Count\_Customer for number of customers.**

```
SELECT count(customer_number) Count_Customer FROM customer_master
```

```
WHERE customer_number NOT IN (SELECT customer_number FROM account_master);
```

|                |
|----------------|
| Count_customer |
| 4              |

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

```
SELECT a.account_number,a.opening_balance+sum(t.transaction_amount)
```

```
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
```

```
WHERE t.transaction_type='Deposit' GROUP BY t.account_number;
```

| account_number | Deposit_Amount |
|----------------|----------------|
| A00001         | 10000          |
| A00002         | 6000           |
| A00007         | 17000          |

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

```
SELECT branch.branch_city, count(account.account_number) No_of_Accounts
```

```
FROM branch_master LEFT JOIN account_master
```

```
ON account.branch_id=branch.branch_id
```

```
GROUP BY branch.branch_city ORDER BY branch_city;
```

| branch_city | No_of_accounts |
|-------------|----------------|
| CHENNAI     | 0              |
| DELHI       | 6              |
| KOLKATA     | 0              |
| MUMBAI      | 4              |

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

```
SELECT c.firstname FROM
customer_master c JOIN account_master a ON a.customer_number=c.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1;
```

firstname

AMIT

AVINASH

RAHUL

RAMESH

**10. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.**

```
SELECT c.customer_number,c.firstname,c.lastname FROM
customer_master c JOIN loan_details l ON c.customer_number=l.customer_number
GROUP BY l.customer_number HAVING count(l.branch_id)>1
ORDER BY c.customer_number;
```

| customer_number | firstname | lastname |
|-----------------|-----------|----------|
| C00001          | RAMESH    | SHARMA   |
| C00002          | AVINASH   | MINHA    |

**11. Write a query to display the customer's number, customer's firstname, customer's city and branch city where the city of the customer and city of the branch is different. Display the records sorted in ascending order based on customer number.**

```
SELECT c.customer_number,c.firstname,c.customer_city,b.branch_city FROM
Customer_master c JOIN Account_master a ON c.customer_number=a.customer_number
JOIN Branch_master b ON b.branch_id=a.branch_id
WHERE b.branch_city<>c.customer_city
ORDER BY c.customer_number;
```

| customer_number | firstname | customer_city | branch_city |
|-----------------|-----------|---------------|-------------|
| C00002          | AVINASH   | DELHI         | MUMBAI      |
| C00003          | RAHUL     | DELHI         | MUMBAI      |
| C00007          | AMIT      | MUMBAI        | DELHI       |

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

```
SELECT count(c.customer_number)Count FROM customer_master c JOIN loan_details l
ON c.customer_number=l.customer_number
WHERE c.customer_number NOT IN (SELECT customer_number FROM account_master);
```

|       |
|-------|
| Count |
| 2     |

13. Write a query to display the account number who has done the highest transaction. For example the account A00023 has done 5 transactions i.e. suppose 3 withdrawal and 2 deposits. Whereas the account A00024 has done 3 transactions i.e. suppose 2 withdrawals and 1 deposit. So account number of A00023 should be displayed. In case of multiple records, display the records sorted in ascending order based on account number.

```
SELECT account_number FROM transaction_details
GROUP BY account_number
HAVING count(transaction_number)>=ALL
(SELECT count(transaction_number) FROM transaction_details
GROUP BY account_number) ORDER BY account_number;
```

|                |
|----------------|
| account_number |
| A00001         |

14. Write a query to show the branch name,branch city where we have the maximum customers. For example the branch B00019 has 3 customers, B00020 has 7 and B00021 has 10. So branch id B00021 is having maximum customers. If B00021 is Koramangla branch Bangalore, Koramangla branch should be displayed along with city name Bangalore. In case of multiple records, display the records sorted in ascending order based on branch name.

```
SELECT b.branch_name,b.branch_city FROM
Branch_master b JOIN account a ON a.branch_id=b.branch_id
GROUP BY a.branch_id HAVING count(a.customer_number)>=ALL
(SELECT count(customer_number) FROM
Account_master GROUP BY branch_id)
ORDER BY b.branch_name;
```

| branch_name   | branch_city |
|---------------|-------------|
| ASAF ALI ROAD | DELHI       |

15. Write a query to display all those account number, deposit, withdrawal where withdrawal is more than deposit amount. Hint: Deposit should include opening balance as well. For example A00011 account opened with Opening Balance 1000 and A00011 deposited 2000 rupees on 2012-12-01 and 3000 rupees on 2012-12-02. The same account i.e A00011 withdrawn 3000 rupees on 2013-01-01 and 7000 rupees on 2013-01-03. So the total deposited amount is 6000 and total withdrawal amount is 10000. So withdrawal amount is more than deposited amount for account number A00011. Display the records sorted in ascending order based on account number.

```

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

```

**(or)**

```

SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d Deposit,ifnull(t1.w,0) Withdrawal FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account_master a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.w,0)>t2.d
ORDER BY account_number;

```

| account_number | Deposit | Withdrawal |
|----------------|---------|------------|
| A00001         | 10000   | 12000      |
| A00002         | 6000    | 7000       |

**16. Write a query to show the balance amount for account number that ends with 001. Note: Balance amount includes account opening balance also. Give alias name as Balance\_Amount. For example A00015 is having an opening balance of 1000. A00015 has deposited 2000 on 2012-06-12 and deposited 3000 on 2012-07-13. The same account has drawn money of 500 on 2012-08-12 , 500 on 2012-09-15, 1000 on 2012-12-17. So balance amount is 4000 i.e (1000 (opening balance)+2000+3000 ) – (500+500+1000).**

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

**(or)**

```
SELECT ifnull(t1.account_number,t2.account_number) account_number,
t2.d-ifnull(t1.w,0) Balance_Amount FROM
(SELECT account_number,transaction_type,sum(transaction_amount) w from transaction_details
WHERE transaction_type='Withdrawal' GROUP BY account_number) t1
RIGHT JOIN
(SELECT a.account_number,a.opening_balance+sum(t.transaction_amount) d
FROM account a JOIN transaction_details t ON a.account_number=t.account_number
WHERE t.transaction_type='Deposit' GROUP BY t.account_number) t2
ON t1.account_number=t2.account_number
WHERE ifnull(t1.account_number,t2.account_number) LIKE '%001'
ORDER BY account_number;
```

| account_number | Balance_Amount |
|----------------|----------------|
| A00001         | -2000          |

**17. Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as**

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

```
SELECT c.customer_number,c.firstname,t.account_number, count(t.account_number) Count_Trans
FROM transaction_details t JOIN account_master a ON a.account_number=t.account_number
JOIN customer c ON c.customer_number=a.customer_number
GROUP BY t.account_number ORDER BY c.customer_number, a.account_number;
```

| customer_number | firstname | account_number | Count_Trans |
|-----------------|-----------|----------------|-------------|
| C00001          | RAMESH    | A00001         | 6           |
| C00002          | AVINASH   | A00002         | 3           |
| C00007          | AMIT      | A00007         | 3           |

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

```
SELECT c.firstname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number
GROUP BY a.customer_number HAVING count(a.account_number)>1
ORDER BY c.firstname;
```

| firstname |
|-----------|
| AMIT      |
| AVINASH   |
| RAHUL     |
| RAMESH    |

**19.** Write a query to display the customer number, firstname, lastname for those client where total loan amount taken is maximum and at least taken from 2 branches. For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan amount for customer C00012 is 600000. C00013 took a loan of 100000 from bank branch B00009 and 200000 from bank branch B00011. So total loan taken is 300000. So loan taken by C00012 is more then C00013.

```
SELECT Id.customer_number,firstname,lastname
FROM customer_master cm JOIN loan_details Id
ON cm.customer_number=Id.customer_number
GROUP BY customer_number
HAVING count(branch_id)>=2 AND sum(loan_amount)>=
```

```
ALL(SELECT sum(loan_amount) FROM loan GROUP BY customer_number);
```

| customer_number | firstname | lastname |
|-----------------|-----------|----------|
| C00002          | AVINASH   | MINHA    |

20. Write a query to display the customer's number, customer's firstname, branch id and loan amount for people who have taken loans. Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.

```
SELECT c.customer_number,c.firstname,l.branch_id,l.loan_amount FROM
Customer_master c JOIN loan_details l ON c.customer_number=l.customer_number
ORDER BY c.customer_number,l.branch_id,l.loan_amount;
```

| customer_number | firstname | branch_id | loan_amount |
|-----------------|-----------|-----------|-------------|
| C00001          | RAMESH    | B00001    | 100000      |
| C00001          | RAMESH    | B00003    | 600000      |
| C00002          | AVINASH   | B00001    | 600000      |
| C00002          | AVINASH   | B00002    | 200000      |
| C00009          | ABHISHEK  | B00008    | 400000      |
| C00010          | SHANKAR   | B00009    | 500000      |

21. Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count\_Branch. Display the records sorted in ascending order based on city name.

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

| branch_city | Count_Branch |
|-------------|--------------|
| CHENNAI     | 1            |
| DELHI       | 4            |
| KOLKATA     | 1            |
| MUMBAI      | 3            |

22. Write a query to display account id, customer's firstname, customer's lastname for the customer's whose account is Active. Display the records sorted in ascending order based on account id /account number.

```
SELECT a.account_number,c.firstname,c.lastname FROM
Customer_master c JOIN account_master a ON c.customer_number=a.customer_number and
a.account_status='Active'
```

ORDER BY a.account\_number;

| account_number | firstname | lastname |
|----------------|-----------|----------|
| A00001         | RAMESH    | SHARMA   |
| A00002         | AVINASH   | MINHA    |
| A00003         | RAHUL     | RASTOGI  |
| A00004         | AVINASH   | MINHA    |
| A00005         | CHITRESH  | BARWE    |
| A00007         | AMIT      | BORKAR   |
| A00010         | PARUL     | GANDHI   |

23. Write a query to display customer's number, first name and middle name. For the customers who don't have middle name, display their last name as middle name. Give the alias name as Middle\_Name. Display the records sorted in ascending order based on customer number.

SELECT customer\_number,firstname,ifnull(middlename,lastname) Middle\_name FROM

Customer\_master ORDER BY customer\_number;

| customer_number | firstname | Middle_name |
|-----------------|-----------|-------------|
| C00001          | RAMESH    | CHANDRA     |
| C00002          | AVINASH   | SUNDER      |
| C00003          | RAHUL     | NULL        |
| C00004          | PARUL     | NULL        |
| C00005          | NAVEEN    | CHANDRA     |
| C00006          | CHITRESH  | NULL        |
| C00007          | AMIT      | KUMAR       |
| C00008          | NISHA     | DAMLE       |
| C00009          | ABHISHEK  | DUTTA       |
| C00010          | SHANKAR   | NAIR        |

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

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

Customer\_master ORDER BY year(customer\_date\_of\_birth),customer\_number;

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

25. Write a query to display the customers firstname, city and account number whose occupation are not into Business, Service or Student. Display the records sorted in ascending order based on customer first name and then by account number.

```
SELECT c.firstname,c.customer_city,a.account_number FROM
Customer_master c JOIN account_master a ON a.customer_number=c.customer_number
WHERE c.occupation NOT IN ('Service','Student','Business')
ORDER BY c.firstname,a.account_number;
```

| firstname | customer_city | account_number |
|-----------|---------------|----------------|
| PARUL     | DELHI         | A00010         |

# AIRLINES

```
create database flight;
```

```
use flight;
```

```
CREATE TABLE air_credit_card_details
(
profile_id VARCHAR(10) NOT NULL,
```

```
card_number BIGINT,
card_type VARCHAR(45),
expiration_month INT,
expiration_year INT
);
```

```
CREATE TABLE air_passenger_profile
(
profile_id VARCHAR(10) NOT NULL ,
password VARCHAR(45) NULL ,
first_name VARCHAR(45) NULL ,
last_name VARCHAR(45) NULL ,
address VARCHAR(45) NULL ,
mobile_number BIGINT NULL ,
email_id VARCHAR(45) NULL
);
```

```
CREATE TABLE air_ticket_info
(
ticket_id VARCHAR(45) NOT NULL ,
profile_id VARCHAR(10) NULL ,
flight_id VARCHAR(45) NULL ,
flight_departure_date DATE NULL ,
status VARCHAR(45) NULL
);
```

```
CREATE TABLE air_flight_details
(
flight_id VARCHAR(45) NOT NULL ,
```

```
flight_departure_date DATE NULL ,
price DECIMAL(10,2) NULL ,
available_seats INT NULL
);
```

```
CREATE TABLE air_flight
(
flight_id VARCHAR(45) NOT NULL ,
airline_id VARCHAR(45) NULL ,
airline_name VARCHAR(45) NULL ,
from_location VARCHAR(45) NULL ,
to_location VARCHAR(45) NULL ,
departure_time TIME NULL ,
arrival_time TIME NULL ,
duration TIME NULL ,
total_seats INT NULL
);
```

```
INSERT INTO air_credit_card_details VALUES
(1, 622098761234, 'debit', 5, 2013),
(2, 652362563625, 'credit', 1, 2013),
(1, 765432345678, 'credit', 2, 2013),
(3, 654378561234, 'debit', 6, 2013),
(4, 625417895623, 'debit', 2, 2013),
(5, 865478956325, 'debit', 3, 2013),
(6, 789563521457, 'credit', 4, 2013),
(2, 543267895432, 'credit', 8, 2013),
(1, 256369856321, 'debit', 1, 2013);
```

```
INSERT INTO air_flight VALUES
```

```
(3173, 'MH370', 'abc', 'hyderabad', 'chennai', '06:30:00', '07:15:00',
 '0:45:00', 100),

(3178, 'MH17', 'def', 'chennai', 'hyderabad', '08:00:00', '09:00:00',
 '1:00:00', 200),

(3172, 'AR342', 'fgh', 'kolkata', 'chennai', '11:30:00', '13:00:00',
 '1:30:00', 100),

(3071, 'JT564', 'jkl', 'chennai', 'delhi', '08:00:00', '10:00:00',
 '2:00:00', 100),

(3170, 'DT345', 'xyz', 'delhi', 'kolkata', '21:00:00', '22:30:00',
 '100),

(3175, 'MJ654', 'abc', 'chennai', 'hyderabad', '15:00:00', '16:00:00',
 '1:00:00', 200),

(3176, 'MH370', 'def', 'kochi', 'chennai', '18:00:00', '19:05:00',
 '100),

(3177, 'MH45', 'fgh', 'delhi', 'kochi', '19:00:00', '21:00:00',
 '2:00:00', 200),

(3174, 'MH321', 'xyz', 'kolkata', 'delhi', '0:00:00', '2:00:00',
 '100),

(3179, 'JT435', 'abc', 'chennai', 'kolkata', '14:00:00', '15:00:00',
 '100),

(3180, 'JT456', 'ijk', 'kolkata', 'kochi', '5:00:00', '5:45:00',
 '0:45:00', 200);
```

```
INSERT INTO air_flight_details VALUES
```

```
(3170, '2013-02-14', 1000, 10),
(3171, '2013-03-15', 5000, 0),
(3172, '2013-02-05', 3000, 32),
(3173, '2013-04-07', 2000, 12),
(3174, '2013-04-05', 3800, 3),
(3175, '2013-05-25', 3500, 10),
(3176, '2013-03-14', 8000, 2),
```

```
(3177, '2013-06-15', 1500, 0),
(3178, '2013-05-06', 3000, 5),
(3179, '2013-04-03', 4000, 15),
(3180, '2013-04-02', 3000, 14);
```

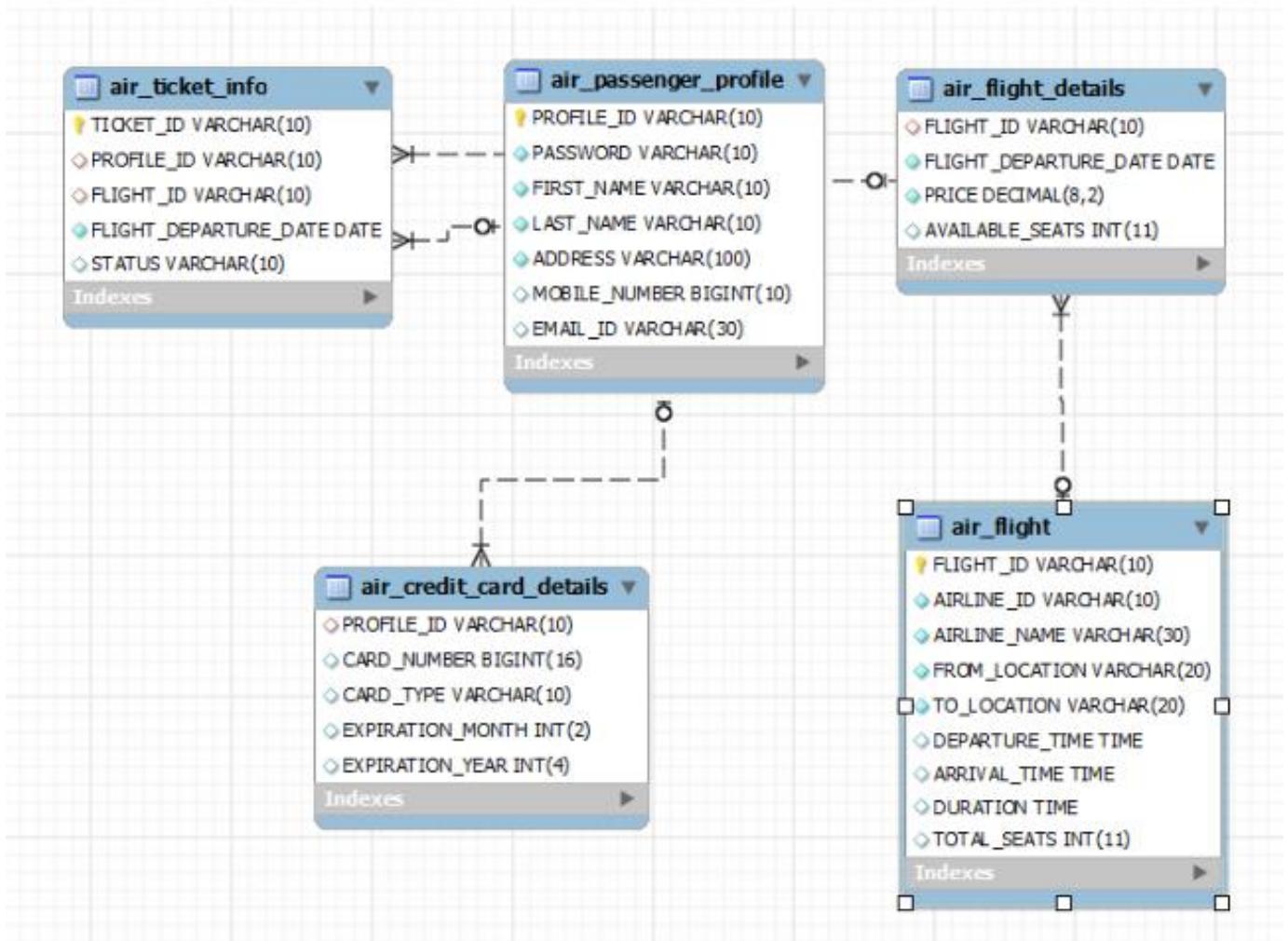
INSERT INTO air\_ticket\_info VALUES

```
(1, 1, 3178, '2013-05-06', 'delayed'),
(2, 5, 3179, '2013-04-03', 'on time'),
(2, 4, 3180, '2013-04-02', 'on time'),
(1, 2, 3177, '2013-06-15', 'on time'),
(1, 3, 3176, '2013-03-14', 'on time'),
(3, 1, 3171, '2013-03-15', 'on time'),
(4, 4, 3172, '2013-02-06', 'delayed'),
(5, 2, 3178, '2013-06-05', 'on time'),
(4, 3, 3171, '2013-03-15', 'on time'),
(5, 1, 3175, '2013-05-25', 'on time'),
(6, 3, 3177, '2013-06-15', 'on time');
```

INSERT INTO air\_passenger\_profile VALUES

```
(1, 'godbless', 'John', 'Stuart', 'Street 21, Near Bus Stop-Hyderabad-432126',
9865263251, 'john@gmail.com'),
(2, 'heyyaa', 'Robert', 'Clive', 'Sector 3, Technopolis-Kolkata-700102',
9733015875, 'robert@yahoo.com'),
(3, 'hello123', 'Raj', 'Sharma', 'House No. 3, Anna Nagar-Kochi-452314',
9775470232, 'raj3452@hotmail.com'),
(4, 'yesboss', 'Sanjay', 'Mittal', '21 Cauunaught Place-Delhi-144985',
9856856321, 'sanjay@yahoo.com'),
```

```
(5, 'imhere', 'Tony', 'Stark', '51A, Greams Lane-Chennai-144587',
9832015785, 'tony@gmail.com');
```



### AIR TICKET INFO

| ticket_id | profile_id | flight_id | flight_departure_date | status  |
|-----------|------------|-----------|-----------------------|---------|
| 1         | 1          | 3178      | 2013-05-06            | delayed |
| 2         | 5          | 3179      | 2013-04-03            | on time |
| 2         | 4          | 3180      | 2013-04-02            | on time |
| 1         | 2          | 3177      | 2013-06-15            | on time |
| 1         | 3          | 3176      | 2013-03-14            | on time |
| 3         | 1          | 3171      | 2013-03-15            | on time |
| 4         | 4          | 3172      | 2013-02-06            | delayed |
| 5         | 2          | 3178      | 2013-06-05            | on time |
| 4         | 3          | 3171      | 2013-03-15            | on time |
| 5         | 1          | 3175      | 2013-05-25            | on time |
| 6         | 3          | 3177      | 2013-06-15            | on time |

### AIR PASSENGER DETAILS

| profile_id | password | first_name | last_name | address                                   | mobile_number | email_id           |
|------------|----------|------------|-----------|-------------------------------------------|---------------|--------------------|
| 1          | godbless | John       | Stuart    | Street 21, Near Bus Stop-Hyderabad-432126 | 9865263251    | john@gmail.com     |
| 2          | heyyaa   | Robert     | Clive     | Sector 3, Technopolis-Kolkata-700102      | 9733015875    | robert@yahoo.com   |
| 3          | hello123 | Raj        | Sharma    | House No. 3, Anna Nagar-Kochi-452314      | 9775470232    | raj3452@hotmail... |
| 4          | yesboss  | Sanjay     | Mittal    | 21 Cauunaught Place-Delhi-144985          | 9856856321    | sanjay@yahoo.c...  |
| 5          | imhere   | Tony       | Stark     | 51A, Greams Lane-Chennai-144587           | 9832015785    | tony@gmail.com     |

### AIR FLIGHT DETAILS

| flight_id | flight_departure_date | price   | available_seats |
|-----------|-----------------------|---------|-----------------|
| 3170      | 2013-02-14            | 1000.00 | 10              |
| 3171      | 2013-03-15            | 5000.00 | 0               |
| 3172      | 2013-02-05            | 3000.00 | 32              |
| 3173      | 2013-04-07            | 2000.00 | 12              |
| 3174      | 2013-04-05            | 3800.00 | 3               |
| 3175      | 2013-05-25            | 3500.00 | 10              |
| 3176      | 2013-03-14            | 8000.00 | 2               |
| 3177      | 2013-06-15            | 1500.00 | 0               |
| 3178      | 2013-05-06            | 3000.00 | 5               |
| 3179      | 2013-04-03            | 4000.00 | 15              |
| 3180      | 2013-04-02            | 3000.00 | 14              |

## AIR CREDIT CARD DETAILS

| profile_id | card_number  | card_type | expiration_month | expiration_year |
|------------|--------------|-----------|------------------|-----------------|
| 1          | 622098761234 | debit     | 5                | 2013            |
| 2          | 652362563625 | credit    | 1                | 2013            |
| 1          | 765432345678 | credit    | 2                | 2013            |
| 3          | 654378561234 | debit     | 6                | 2013            |
| 4          | 625417895623 | debit     | 2                | 2013            |
| 5          | 865478956325 | debit     | 3                | 2013            |
| 6          | 789563521457 | credit    | 4                | 2013            |
| 2          | 543267895432 | credit    | 8                | 2013            |
| 1          | 256369856321 | debit     | 1                | 2013            |

## AIR FLIGHT

| flight_id | airline_id | airline_name | from_location | to_location | departure_time | arrival_time | duration | total_seats |
|-----------|------------|--------------|---------------|-------------|----------------|--------------|----------|-------------|
| 3170      | DT345      | xyz          | delhi         | kolkata     | 21:00:00       | 22:30:00     | 01:30:00 | 100         |
| 3171      | JT564      | JKL          | chennai       | delhi       | 08:00:00       | 10:00:00     | 02:00:00 | 100         |
| 3172      | AR342      | fgh          | kolkata       | chennai     | 11:30:00       | 13:00:00     | 01:30:00 | 100         |
| 3173      | MH370      | abc          | hyderabad     | chennai     | 06:30:00       | 07:15:00     | 00:45:00 | 100         |
| 3174      | MH321      | xyz          | kolkata       | delhi       | 00:00:00       | 02:00:00     | 02:00:00 | 100         |
| 3175      | MJ654      | abc          | chennai       | hyderabad   | 15:00:00       | 16:00:00     | 01:00:00 | 200         |
| 3176      | MH370      | def          | kochi         | chennai     | 18:00:00       | 19:05:00     | 01:05:00 | 100         |
| 3177      | MH45       | fgh          | delhi         | kochi       | 19:00:00       | 21:00:00     | 02:00:00 | 200         |
| 3178      | MH17       | def          | chennai       | hyderabad   | 08:00:00       | 09:00:00     | 01:00:00 | 200         |
| 3179      | JT435      | abc          | chennai       | kolkata     | 14:00:00       | 15:00:00     | 01:00:00 | 100         |
| 3180      | JT456      | ijk          | kolkata       | kochi       | 05:00:00       | 05:45:00     | 00:45:00 | 200         |

## QUERIES

1. Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight\_Id,From\_Location,To\_Location,Month Name as "Month\_Name" and average price as "Average\_Price". Display the records sorted in ascending order based on flight id and then by Month Name.

```
SELECT f.flight_id,f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
AVG(price) Average_Price FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id WHERE f.airline_name = 'abc'
```

GROUP BY f.flight\_id,f.from\_location,f.to\_location,Month\_Name

ORDER BY f.flight\_id, Month\_Name;

| flight_id | from_location | to_location | Month_Name | Average_Price |
|-----------|---------------|-------------|------------|---------------|
| 3173      | hyderabad     | chennai     | April      | 2000.000000   |
| 3175      | chennai       | hyderabad   | May        | 3500.000000   |
| 3179      | chennai       | kolkata     | April      | 4000.000000   |

2. Write a query to display the number of flight services between locations in a month. The Query should display From\_Location, To\_Location, Month as "Month\_Name" and number of flight services as "No\_of\_Services". Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight. The records should be displayed in ascending order based on From\_Location and then by To\_Location and then by month name.

```
SELECT f.from_location,f.to_location,
monthname(af.flight_departure_date) Month_Name,
count(af.flight_departure_date) No_of_Services
FROM air_flight f JOIN air_flight_details af
ON f.flight_id = af.flight_id
GROUP BY f.from_location,f.to_location,Month_Name
ORDER BY f.from_location,f.to_Location,Month_Name;
```

| from_location | to_location | Month_Name | No_of_Services |
|---------------|-------------|------------|----------------|
| chennai       | delhi       | March      | 1              |
| chennai       | hyderabad   | May        | 2              |
| chennai       | kolkata     | April      | 1              |
| delhi         | kochi       | June       | 1              |
| delhi         | kolkata     | February   | 1              |
| hyderabad     | chennai     | April      | 1              |
| kochi         | chennai     | March      | 1              |
| kolkata       | chennai     | February   | 1              |
| kolkata       | delhi       | April      | 1              |
| kolkata       | kochi       | April      | 1              |

3. Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile\_id, customer's first\_name, Address and Number of tickets

**booked as “No\_of\_Tickets”Display the records sorted in ascending order based on customer's first name.**

```
SELECT ap.profile_id,ap.first_name,ap.address,count(ati.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id and af.airline_name='abc'
GROUP BY ap.profile_id,ap.first_name,ap.address HAVING count(ati.ticket_id)<=ALL
(SELECT count(ticket_id)
FROM air_ticket_info GROUP BY profile_id)
ORDER BY ap.first_name;
```

| profile_id | first_name | address                                   | No_of_Tickets |
|------------|------------|-------------------------------------------|---------------|
| 1          | John       | Street 21, Near Bus Stop-Hyderabad-432126 | 1             |
| 5          | Tony       | 51A, Greams Lane-Chennai-144587           | 1             |

**4. Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile\_id,first\_name,last\_name, Flight\_Id , Departure\_Date and number of tickets booked as “No\_of\_Tickets”.Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.**

```
SELECT ap.profile_id,ap.first_name,ap.last_name,af.flight_id,ati.flight_departure_date,
count(ati.profile_id) No_of_Tickets FROM
air_ticket_info ati JOIN air_passenger_profile ap ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.flight_id,ati.profile_id
ORDER BY ap.profile_id,af.flight_id,ati.flight_departure_date;
```

| profile_id | first_name | last_name | flight_id | flight_departure_date | No_of_Tickets |
|------------|------------|-----------|-----------|-----------------------|---------------|
| 1          | John       | Stuart    | 3175      | 2013-05-25            | 1             |
| 1          | John       | Stuart    | 3178      | 2013-05-06            | 1             |
| 2          | Robert     | Clive     | 3178      | 2013-06-05            | 1             |

**5. Write a query to display flight id,from location, to location and ticket price of flights whose departure is in the month of april.Display the records sorted in ascending order based on flight id and then by from location.**

```
SELECT af.flight_id,af.from_location,af.to_location,afd.price FROM
```

```

air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
and month(afd.flight_departure_date)='04'
ORDER BY af.flight_id,af.from_location;

```

| flight_id | from_location | to_location | price   |
|-----------|---------------|-------------|---------|
| 3173      | hyderabad     | chennai     | 2000.00 |
| 3174      | kolkata       | delhi       | 3800.00 |
| 3179      | chennai       | kolkata     | 4000.00 |
| 3180      | kolkata       | kochi       | 3000.00 |

6. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight\_id, from\_location, to\_location and Average price as “Price”. Display the records sorted in ascending order based on flight id and then by from\_location and then by to\_location.

```

SELECT af.flight_id,af.from_location,af.to_location,avg(afd.price) Average_Price FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id
ORDER BY af.flight_id,af.from_location,af.to_location;

```

| flight_id | from_location | to_location | Average_Price |
|-----------|---------------|-------------|---------------|
| 3170      | delhi         | kolkata     | 1000.000000   |
| 3171      | chennai       | delhi       | 5000.000000   |
| 3172      | kolkata       | chennai     | 3000.000000   |
| 3173      | hyderabad     | chennai     | 2000.000000   |
| 3174      | kolkata       | delhi       | 3800.000000   |
| 3175      | chennai       | hyderabad   | 3500.000000   |
| 3176      | kochi         | chennai     | 8000.000000   |
| 3177      | delhi         | kochi       | 1500.000000   |
| 3178      | chennai       | hyderabad   | 3000.000000   |
| 3179      | chennai       | kolkata     | 4000.000000   |
| 3180      | kolkata       | kochi       | 3000.000000   |

7. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile\_id, customer\_name (combine first\_name & last\_name with comma in

**b/w), address of the customer. Give an alias to the name as customer\_name.Hint: Query should fetch unique customers irrespective of multiple tickets booked.Display the records sorted in ascending order based on profile id.**

```
SELECT ap.profile_id,concat(ap.first_name,',',ap.last_name) customer_name,ap.address FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
GROUP BY ati.profile_id
ORDER BY ap.profile_id;
```

| profile_id | Customer_name | address                                   |
|------------|---------------|-------------------------------------------|
| 1          | John,Stuart   | Street 21, Near Bus Stop-Hyderabad-432126 |
| 2          | Robert,Clive  | Sector 3, Technopolis-Kolkata-700102      |

**8. Write a query to display profile id of the passenger(s) who has/have booked maximum number of tickets.In case of multiple records, display the records sorted in ascending order based on profile id.**

```
SELECT profile_id FROM air_ticket_info
group by profile_id
having count(ticket_id)>=all(select count(ticket_id)
from air_ticket_info
group by profile_id) order by profile_id;
```

| profile_id |
|------------|
| 1          |
| 3          |

**9. Write a query to display the total number of tickets as “No\_of\_Tickets” booked in each flight in ABC Airlines. The Query should display the flight\_id, from\_location, to\_location and the number of tickets. Display only the flights in which atleast 1 ticket is booked.Display the records sorted in ascending order based on flight id.**

```
SELECT f.flight_id,f.from_location,f.to_location,COUNT(t.ticket_id) AS No_of_Tickets
FROM air_ticket_info t JOIN air_flight f
ON f.flight_id = t.flight_id where AIRLINE_NAME = 'abc' GROUP by
f.flight_id,f.from_location,f.to_location
```

having count(t.ticket\_id)>=1

ORDER by f.flight\_id;

| flight_id | from_location | to_location | No_of_Tickets |
|-----------|---------------|-------------|---------------|
| 3175      | chennai       | hyderabad   | 1             |
| 3179      | chennai       | kolkata     | 1             |

**10.** Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight\_id, number of services as “No\_of\_Services” and the cost as “Total\_Price” in the same order. Order the result by Total Price in descending order and then by flight\_id in descending order. Hint: The number of services can be calculated from the number of scheduled departure dates of the flight

```
SELECT flight_id,count(flight_departure_date) No_of_services,sum(price) Total_Price FROM
air_flight_details GROUP BY flight_id

ORDER BY Total_price DESC,flight_id DESC;
```

| flight_id | No_of_services | Total_Price |
|-----------|----------------|-------------|
| 3176      | 1              | 8000.00     |
| 3171      | 1              | 5000.00     |
| 3179      | 1              | 4000.00     |
| 3174      | 1              | 3800.00     |
| 3175      | 1              | 3500.00     |
| 3180      | 1              | 3000.00     |
| 3178      | 1              | 3000.00     |
| 3172      | 1              | 3000.00     |
| 3173      | 1              | 2000.00     |
| 3177      | 1              | 1500.00     |
| 3170      | 1              | 1000.00     |

**11.** Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight\_id, flight\_departure\_date and the number of passengers as “No\_of\_Passengers” in the same order. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT flight_id,flight_departure_date,count(ticket_id) No_of_passengerS FROM
air_ticket_info GROUP BY flight_id,flight_departure_date

ORDER BY flight_id,flight_departure_date;
```

| flight_id | flight_departure_date | No_of_passengers |
|-----------|-----------------------|------------------|
| 3171      | 2013-03-15            | 2                |
| 3172      | 2013-02-06            | 1                |
| 3175      | 2013-05-25            | 1                |
| 3176      | 2013-03-14            | 1                |
| 3177      | 2013-06-15            | 2                |
| 3178      | 2013-05-06            | 1                |
| 3178      | 2013-06-05            | 1                |
| 3179      | 2013-04-03            | 1                |
| 3180      | 2013-04-02            | 1                |

**12. Write a query to display profile id of passenger(s) who booked minimum number of tickets. In case of multiple records, display the records sorted in ascending order based on profile id.**

```
SELECT profile_id FROM air_ticket_info
GROUP BY profile_id HAVING count(ticket_id)<=ALL
(SELECT count(ticket_id) FROM air_ticket_info GROUP BY profile_id)
ORDER BY profile_id;
```

| profile_id |
|------------|
| 5          |

**13. Write a query to display unique passenger profile id, first name, mobile number and email address of passengers who booked ticket to travel from HYDERABAD to CHENNAI. Display the records sorted in ascending order based on profile id.**

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.mobile_number,ap.email_id FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON ati.flight_id=af.flight_id
WHERE af.from_location='Hyderabad' and af.to_location='Chennai'
ORDER BY profile_id;
```

| profile_id | first_name | mobile_number | email_id |
|------------|------------|---------------|----------|
|            |            |               |          |

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

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.last_name,ati.flight_id,ati.flight_departure_date,
af.departure_time,af.arrival_time,
addtime(af.departure_time,'01:00:00') Delayed_Departure_Time,
addtime(af.arrival_time,'01:00:00') Delayed_Arrival_Time FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
JOIN air_flight af ON af.flight_id=ati.flight_id
WHERE af.from_location='Chennai' and af.to_location='Hyderabad'
and ati.flight_departure_date='2013-05-06'
ORDER BY profile_id;
```

| profile_id | first_name | last_name | flight_id | flight_departure_date | departure_time | arrival_time | Delayed_Departure_Time | Delayed_Arrival_Time |
|------------|------------|-----------|-----------|-----------------------|----------------|--------------|------------------------|----------------------|
| 1          | John       | Stuart    | 3178      | 2013-05-06            | 08:00:00       | 09:00:00     | 09:00:00               | 10:00:00             |

**15. Write a query to display the number of tickets as "No\_of\_Tickets" booked by Kochi Customers. The Query should display the Profile\_Id, First\_Name, Base\_Location and number of tickets booked. Hint: Use String functions to get the base location of customer from their Address and give alias name as "Base\_Location" Display the records sorted in ascending order based on customer first name.**

```
SELECT ap.profile_id,ap.first_name,
substring_index(substring_index(ap.address,'-',2),'-',1) Base_Location,
count(ati.ticket_id) No_of_Tickets FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ati.profile_id=ap.profile_id
WHERE ap.address LIKE '%Kochi%'
ORDER BY ap.first_name;
```

| profile_id | first_name | Base_Location | No_of_Tickets |
|------------|------------|---------------|---------------|
| 3          | Raj        | Kochi         | 3             |

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

```
SELECT af.flight_id,af.from_location,af.to_location,count(afd.flight_departure_date) No_of_services
FROM
air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
WHERE month(flight_departure_date)='05'
GROUP BY af.flight_id,af.from_location,af.to_location
ORDER BY af.flight_id;
```

| flight_id | from_location | to_location | No_of_services |
|-----------|---------------|-------------|----------------|
| 3175      | chennai       | hyderabad   | 1              |
| 3178      | chennai       | hyderabad   | 1              |

17. Write a query to display profile id, last name, mobile number and email id of passengers whose base location is chennai. Display the records sorted in ascending order based on profile id.

```
SELECT profile_id, last_name, mobile_number, email_id
FROM air_passenger_profile
WHERE address LIKE '%Chennai%'
ORDER BY profile_id;
```

| profile_id | last_name | mobile_number | email_id       |
|------------|-----------|---------------|----------------|
| 5          | Stark     | 9832015785    | tony@gmail.com |

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

```
SELECT count(flight_id) FLIGHT_COUNT FROM air_flight
WHERE from_location='CHENNAI'
AND departure_time BETWEEN '06:00:00' AND '18:00:00';
```

| FLIGHT_COUNT |
|--------------|
| 4            |

**19. Write a query to display unique profile id,first name , email id and contact number of passenger(s) who travelled on flight with id 3178. Display the records sorted in ascending order based on first name.**

```
SELECT DISTINCT ap.profile_id,ap.first_name,ap.email_id,ap.mobile_number FROM
air_passenger_profile ap JOIN air_ticket_info ati ON ap.profile_id=ati.profile_id
WHERE ati.flight_id='3178'
ORDER BY ap.first_name;
```

| profile_id | first_name | email_id         | mobile_number |
|------------|------------|------------------|---------------|
| 1          | John       | john@gmail.com   | 9865263251    |
| 2          | Robert     | robert@yahoo.com | 9733015875    |

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

```
SELECT flight_id,flight_departure_date,
case when price<3000 then 'AIR PASSENGER'
 when price>=3000 and price<4000 then 'AIR BUS'
 when price>=4000 then 'EXECUTIVE PASSENGER'
end FLIGHT_TYPE FROM air_flight_details
ORDER BY flight_id,flight_departure_date;
```

| flight_id | flight_departure_date | FLIGHT_TYPE         |
|-----------|-----------------------|---------------------|
| 3170      | 2013-02-14            | AIR PASSENGER       |
| 3171      | 2013-03-15            | EXECUTIVE PASSENGER |
| 3172      | 2013-02-05            | AIR BUS             |
| 3173      | 2013-04-07            | AIR PASSENGER       |
| 3174      | 2013-04-05            | AIR BUS             |
| 3175      | 2013-05-25            | AIR BUS             |
| 3176      | 2013-03-14            | EXECUTIVE PASSENGER |
| 3177      | 2013-06-15            | AIR PASSENGER       |
| 3178      | 2013-05-06            | AIR BUS             |
| 3179      | 2013-04-03            | EXECUTIVE PASSENGER |
| 3180      | 2013-04-02            | AIR BUS             |

**21. Write a query to display the credit card type and no of credit cards used on the same type. Display the records sorted in ascending order based on credit card type.** Hint: Use CARD\_COUNT AS Alias name for no of cards.

```
SELECT card_type, count(card_type) Card_Count FROM air_credit_card_details
GROUP BY card_type ORDER BY card_type;
```

| card_type | Card_Count |
|-----------|------------|
| credit    | 4          |
| debit     | 5          |

**22. Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com. The Serial No will be the last three digits of profile ID.** Hint: Use SERIAL\_NO as Alias name for serial number. Display the records sorted in ascending order based on name.

```
SELECT substring(profile_id,-3) SERIAL_NO,first_name,mobile_number,email_id FROM
air_passenger_profile
WHERE email_id LIKE '%@gmail.com'
ORDER BY first_name;
```

| SERIAL_NO | first_name | mobile_number | email_id       |
|-----------|------------|---------------|----------------|
|           | John       | 9865263251    | john@gmail.com |
|           | Tony       | 9832015785    | tony@gmail.com |

**23. Write a query to display the flight(s) which has least number of services in the month of May. The Query should fetch flight\_id, from\_location, to\_location, least number of Services as “No\_of\_Services”** Hint: Number of services offered can be calculated from the number of scheduled departure dates of a flight if there are multiple flights, display them sorted in ascending order based on flight id.

```
SELECT afd.flight_id,af.from_location,af.to_location,count(afd.flight_id) No_of_Services
FROM air_flight_details afd JOIN air_flight af ON af.flight_id=afd.flight_id
WHERE monthname(afd.flight_departure_date)='May'
GROUP BY afd.flight_departure_date HAVING count(afd.flight_id) <=
ALL(SELECT count(flight_id) FROM air_flight_details
WHERE monthname(flight_departure_date)='May'
GROUP BY flight_departure_date)
```

ORDER BY flight\_id;

| flight_id | from_location | to_location | No_of_Services |
|-----------|---------------|-------------|----------------|
| 3175      | chennai       | hyderabad   | 1              |
| 3178      | chennai       | hyderabad   | 1              |

24. Write a query to display the flights available in Morning, AfterNoon, Evening& Night. The Query should display the Flight\_Id, From\_Location, To\_Location , Departure\_Time, time of service as "Time\_of\_Service". Time of Service should be calculated as: From 05:00:01 Hrs to 12:00:00 Hrs - Morning, 12:00:01 to 18:00:00 Hrs -AfterNoon, 18:00:01 to 24:00:00 - Evening and 00:00:01 to 05:00:00 - NightDisplay the records sorted in ascending order based on flight id.

```
SELECT flight_id,from_location,to_location,Departure_Time,
CASE
WHEN departure_time BETWEEN ('05:00:01') AND ('12:00:00')
THEN 'Morning'
WHEN departure_time BETWEEN ('12:00:01') AND ('18:00:00')
THEN 'AfterNoon'
WHEN departure_time BETWEEN ('18:00:01') AND ('24:00:00')
THEN 'Evening'
WHEN departure_time='00:00:00'
THEN 'Evening'
WHEN departure_time BETWEEN ('00:00:01') AND ('05:00:00')
THEN 'Night'
END Time_of_Service
FROM air_flight
order by flight_id;
```

| flight_id | from_location | to_location | Departure_Time | Time_of_Service |
|-----------|---------------|-------------|----------------|-----------------|
| 3170      | delhi         | kolkata     | 21:00:00       | Evening         |
| 3171      | chennai       | delhi       | 08:00:00       | Morning         |
| 3172      | kolkata       | chennai     | 11:30:00       | Morning         |
| 3173      | hyderabad     | chennai     | 06:30:00       | Morning         |
| 3174      | kolkata       | delhi       | 00:00:00       | Evening         |
| 3175      | chennai       | hyderabad   | 15:00:00       | AfterNoon       |
| 3176      | kochi         | chennai     | 18:00:00       | AfterNoon       |
| 3177      | delhi         | kochi       | 19:00:00       | Evening         |
| 3178      | chennai       | hyderabad   | 08:00:00       | Morning         |
| 3179      | chennai       | kolkata     | 14:00:00       | AfterNoon       |
| 3180      | kolkata       | kochi       | 05:00:00       | Night           |

25. Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No\_of\_Flights”. Hint: Get the distinct from location and to location. Display the records sorted in ascending order based on from location.

```
SELECT from_location, count(flight_id) No_of_Flights FROM
air_flight GROUP BY from_location
ORDER BY from_location;
```

| from_location | No_of_Flights |
|---------------|---------------|
| chennai       | 4             |
| delhi         | 2             |
| hyderabad     | 1             |
| kochi         | 1             |
| kolkata       | 3             |

26. Write a query to display the number of passengers traveled in each flight in each scheduled date. The Query should display flight\_id, from\_location, To\_location, flight\_departure\_date and the number of passengers as “No\_of\_Passengers”. Hint: The Number of passengers inclusive of all the tickets booked with single profile id. Display the records sorted in ascending order based on flight id and then by flight departure date.

```
SELECT af.flight_id, af.from_location, af.to_location, ati.flight_departure_date,
count(ati.ticket_id) No_of_Passengers FROM
air_flight af JOIN air_ticket_info ati ON af.flight_id=ati.flight_id
GROUP BY af.flight_id, af.from_location, af.to_location, ati.flight_departure_date
```

ORDER BY af.flight\_id,ati.flight\_departure\_date;

| flight_id | from_location | to_location | flight_departure_date | No_of_Passengers |
|-----------|---------------|-------------|-----------------------|------------------|
| 3171      | chennai       | delhi       | 2013-03-15            | 2                |
| 3172      | kolkata       | chennai     | 2013-02-06            | 1                |
| 3175      | chennai       | hyderabad   | 2013-05-25            | 1                |
| 3176      | kochi         | chennai     | 2013-03-14            | 1                |
| 3177      | delhi         | kochi       | 2013-06-15            | 2                |
| 3178      | chennai       | hyderabad   | 2013-05-06            | 1                |
| 3178      | chennai       | hyderabad   | 2013-06-05            | 1                |
| 3179      | chennai       | kolkata     | 2013-04-03            | 1                |
| 3180      | kolkata       | kochi       | 2013-04-02            | 1                |

27. Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight\_Id, From\_Location, To\_Location, Total\_Seats, seats booked as "No\_of\_Seats\_Booked". Display the records sorted in ascending order based on flight id and then by No\_of\_Seats\_Booked.

```
SELECT af.flight_id,af.from_location,af.to_location,af.total_seats,
(af.total_seats-afd.available_seats) No_of_Seats_Booked FROM
air_flight_details afd JOIN air_flight af ON afd.flight_id=af.flight_id
WHERE (af.total_seats-afd.available_seats)>(af.total_seats*0.1)
ORDER BY flight_id,No_of_Seats_Booked;
```

| flight_id | from_location | to_location | total_seats | No_of_Seats_Booked |
|-----------|---------------|-------------|-------------|--------------------|
| 3170      | delhi         | kolkata     | 100         | 90                 |
| 3171      | chennai       | delhi       | 100         | 100                |
| 3172      | kolkata       | chennai     | 100         | 68                 |
| 3173      | hyderabad     | chennai     | 100         | 88                 |
| 3174      | kolkata       | delhi       | 100         | 97                 |
| 3175      | chennai       | hyderabad   | 200         | 190                |
| 3176      | kochi         | chennai     | 100         | 98                 |
| 3177      | delhi         | kochi       | 200         | 200                |
| 3178      | chennai       | hyderabad   | 200         | 195                |
| 3179      | chennai       | kolkata     | 100         | 85                 |
| 3180      | kolkata       | kochi       | 200         | 186                |

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

```
SELECT af.flight_id,afd.flight_Departure_Date,af.From_Location,af.To_Location,af.duration
```

FROM air\_flight af JOIN air\_flight\_details afd ON af.flight\_id=afd.flight\_id  
WHERE af.duration<'01:10:00';

| flight_id | flight_Departure_Date | From_Location | To_Location | duration |
|-----------|-----------------------|---------------|-------------|----------|
| 3173      | 2013-04-07            | hyderabad     | chennai     | 00:45:00 |
| 3175      | 2013-05-25            | chennai       | hyderabad   | 01:00:00 |
| 3176      | 2013-03-14            | kochi         | chennai     | 01:05:00 |
| 3178      | 2013-05-06            | chennai       | hyderabad   | 01:00:00 |
| 3179      | 2013-04-03            | chennai       | kolkata     | 01:00:00 |
| 3180      | 2013-04-02            | kolkata       | kochi       | 00:45:00 |

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

```
SELECT afd.flight_id,af.from_location,af.to_location,
count(afd.flight_departure_date) No_of_Service, avg(price) Average_Price
FROM air_flight af JOIN air_flight_details afd ON af.flight_id=afd.flight_id
GROUP BY af.flight_id,af.from_location,af.to_location
HAVING avg(price)>(SELECT avg(price) FROM air_flight_details)
ORDER BY average_price;
```

| flight_id | from_location | to_location | No_of_Service | Average_Price |
|-----------|---------------|-------------|---------------|---------------|
| 3175      | chennai       | hyderabad   | 1             | 3500.000000   |
| 3174      | kolkata       | delhi       | 1             | 3800.000000   |
| 3179      | chennai       | kolkata     | 1             | 4000.000000   |
| 3171      | chennai       | delhi       | 1             | 5000.000000   |
| 3176      | kochi         | chennai     | 1             | 8000.000000   |

# MOVIE

```
CREATE DATABASE video;USE video;
```

```
Create table CUSTOMER_MASTER
```

```
(CUSTOMER_ID Varchar(10),CUSTOMER_NAME Varchar(30) NOT NULL,CONTACT_NO
BIGINT(10),CONTACT_ADD Varchar(20),DATE_OF_REGISTRATION Date NOT NULL,AGE
Varchar(15)NOT NULL,Constraint MT_cts1 PRIMARY KEY(CUSTOMER_ID));
```

```
Create table LIBRARY_CARD_MASTER
```

```
(CARD_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT
BIGINT(50),NUMBER_OF_YEARS bigint(10) NOT NULL,Constraint MT_cts2 PRIMARY
KEY(CARD_ID));
```

```
Create table MOVIES_MASTER
```

```
(MOVIE_ID Varchar(10), MOVIE_NAME Varchar(50) NOT NULL,RELEASE_DATE Varchar(30)
NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION VARCHAR(10) NOT NULL,
 MOVIE_TYPE Varchar(3),MOVIE_CATEGORY VARCHAR(20) NOT NULL,DIRECTOR
VARCHAR(20) NOT NULL,
```

```
LEAD_ROLE_1 Varchar(3) NOT NULL,LEAD_ROLE_2 VARCHAR(4) NOT NULL,RENT_COST
BIGINT(10),Constraint MT_cts4 PRIMARY KEY(MOVIE_ID));
```

```
Create table CUSTOMER_CARD_DETAILS
```

```
(CUSTOMER_ID Varchar(10),CARD_ID VARCHAR(10),ISSUE_DATE DATE NOT NULL,Constraint
MT_cts3 PRIMARY KEY(CUSTOMER_ID),Constraint MT_CTS41 FOREIGN KEY(CUSTOMER_ID)
References CUSTOMER_MASTER(CUSTOMER_ID),Constraint MT_CTS42 FOREIGN KEY(CARD_ID)
References LIBRARY_CARD_MASTER(CARD_ID));
```

```
Create table CUSTOMER_ISSUE_DETAILS
```

```
(ISSUE_ID Varchar(10) NOT NULL,CUSTOMER_ID Varchar(10) NOT NULL,MOVIE_ID
VARCHAR(10),ISSUE_DATE Date NOT NULL,RETURN_DATE Date NOT NULL,
ACTUAL_DATE_RETURN Date NOT NULL,Constraint MT_cts5 PRIMARY
KEY(ISSUE_ID),Constraint MT_Mem FOREIGN KEY(CUSTOMER_ID) References
CUSTOMER_MASTER(CUSTOMER_ID), Constraint MT_Mem1 FOREIGN KEY(MOVIE_ID)
References MOVIES_MASTER(MOVIE_ID));
```

```
Insert into CUSTOMER_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-12', '21');
Insert into CUSTOMER_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS007', 'NAME7', 3210987654,'ADD7', '2012-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS008', 'NAME8', 2109876543,'ADD8', '2013-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS009', 'NAME9', NULL,'ADD9', '2013-02-12', '21');
Insert into CUSTOMER_MASTER Values('CUS010', 'NAM10', 9934567890,'ADD10', '2013-02-12',
'21');
Insert into CUSTOMER_MASTER Values('CUS011', 'NAM11', 9875678910,'ADD11', '2013-02-12',
'21');
```

```
Insert into LIBRARY_CARD_MASTER Values('CR001', 'Silver', 200, 5);
Insert into LIBRARY_CARD_MASTER Values('CR002', 'Gold', 400, 9);
Insert into LIBRARY_CARD_MASTER Values('CR003', 'Platinum', 600, 8);
Insert into LIBRARY_CARD_MASTER Values('CR004', 'VISA', 800, 7);
Insert into LIBRARY_CARD_MASTER Values('CR005', 'CREDIT', 1200, 6);
```

```
Insert into MOVIES_MASTER Values('MV001', 'DIEHARD', '2012-05-13','ENGLISH', 4 , '2HRS',
'U/A','ACTION','DIR1','L1','L2',100);
```

Insert into MOVIES\_MASTER Values('MV002', 'THE MATRIX', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR2','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV003', 'INCEPTION', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','ACTION','DIR3','L15','L2',100);

Insert into MOVIES\_MASTER Values('MV004', 'DARK KNIGHT', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ACTION','DIR4','L15','L2',100);

Insert into MOVIES\_MASTER Values('MV005', 'OFFICE S', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR5','L12','L24',100);

Insert into MOVIES\_MASTER Values('MV006', 'SHAWN OF DEAD', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR6','L1','L25',100);

Insert into MOVIES\_MASTER Values('MV007', 'YOUNG FRANKEN', '2012-05-13','ENGLISH', 4 , '2HRS', 'U/A','COMEDY','DIR7','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV008', 'CAS', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR8','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV009', 'GWW', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR9','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV010', 'TITANIC', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR10','L1','L2',100);

Insert into MOVIES\_MASTER Values('MV011', 'THE NOTE BOOK', '2012-05-13','ENGLISH', 4 , '2HRS', 'A','ROMANCE','DIR11','L1','L2',100);

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS001', 'CR001', '2012-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS002', 'CR002', '2012-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS003', 'CR002', '2013-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS004', 'CR003', '2013-05-13');

Insert into CUSTOMER\_CARD\_DETAILS Values('CUS005', 'CR003', '2012-05-13');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS001', 'CUS001', 'MV001', '2012-05-13', '2012-05-13','2012-05-13');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS002', 'CUS001', 'MV001', '2012-05-01', '2012-05-16','2012-05-16');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS003', 'CUS002', 'MV004', '2012-05-02', '2012-05-06','2012-05-16');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS004', 'CUS002', 'MV004', '2012-04-03', '2012-04-16','2012-04-20');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS005', 'CUS002', 'MV009', '2012-04-04', '2012-04-16','2012-04-20');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS006', 'CUS003', 'MV002', '2012-03-30', '2012-04-15','2012-04-20');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS007', 'CUS003', 'MV003', '2012-04-20', '2012-05-05','2012-05-05');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS008', 'CUS003', 'MV005', '2012-04-21', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS009', 'CUS003', 'MV001', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS010', 'CUS003', 'MV009', '2012-04-22', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS011', 'CUS003', 'MV010', '2012-04-23', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS012', 'CUS003', 'MV010', '2012-04-24', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS013', 'CUS003', 'MV008', '2012-04-25', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS014', 'CUS004', 'MV007', '2012-04-26', '2012-05-07','2012-05-25');

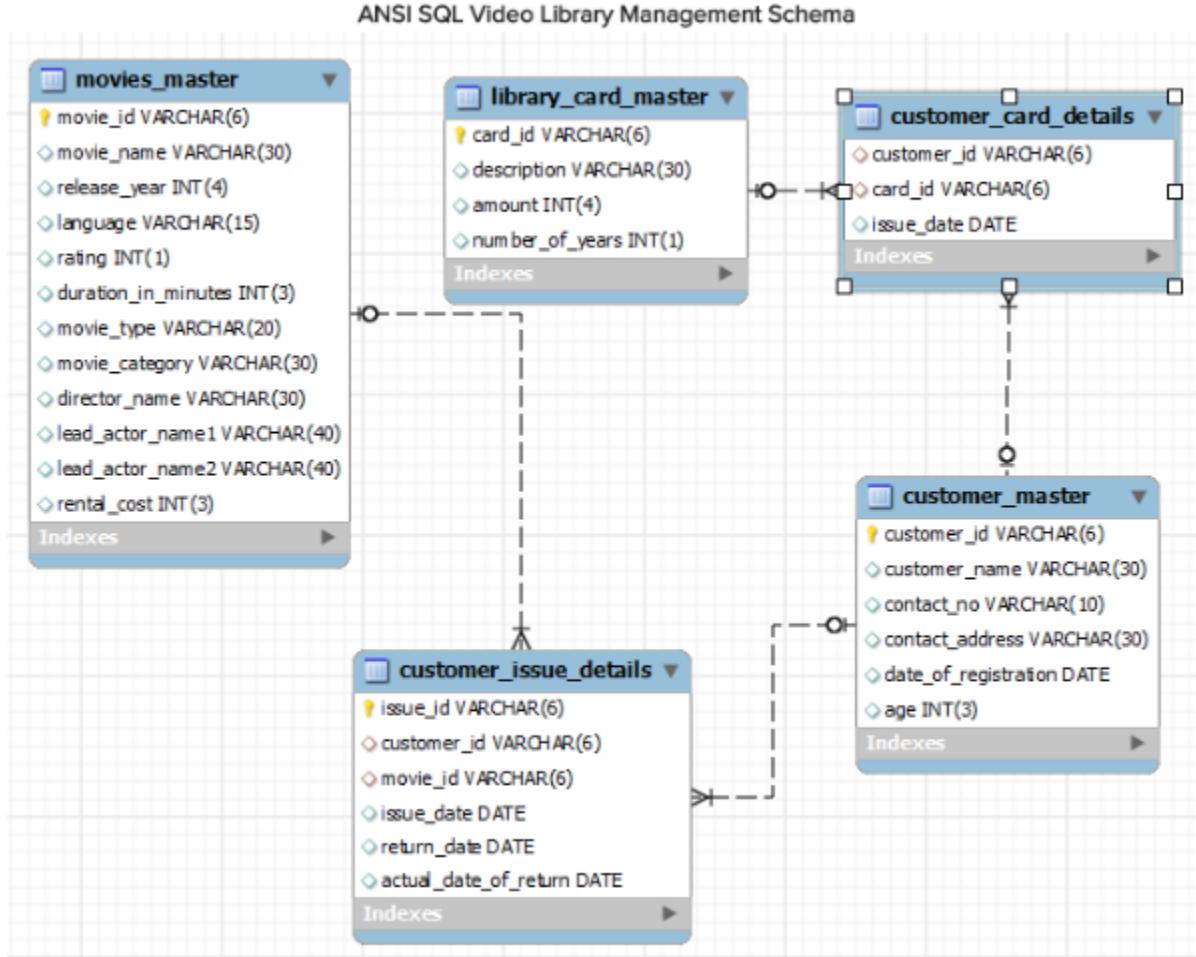
Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS015', 'CUS004', 'MV006', '2012-04-27', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS016', 'CUS004', 'MV006', '2012-04-28', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS017', 'CUS004', 'MV001', '2012-04-29', '2012-05-07','2012-05-25');

Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS018', 'CUS010', 'MV008', '2012-04-24', '2012-05-07','2012-05-25');

```
Insert into CUSTOMER_ISSUE_DETAILS Values ('IS019', 'CUS011', 'MV009', '2012-04-27', '2012-05-07','2012-05-25');
```



MOVIE MASTER

| LEAD_ROLE_2 | RENT_COST |
|-------------|-----------|
| L2          | 100       |
| L24         | 100       |
| L25         | 100       |
| L2          | 100       |
| NULL        | NULL      |

#### CUSTOMER MASTER

| CUSTOMER_ID | CUSTOMER_NAME | CONTACT_NO | CONTACT_ADD | DATE_OF_REGISTRATION | AGE  |
|-------------|---------------|------------|-------------|----------------------|------|
| CUS001      | AMIT          | 9876543210 | ADD1        | 2012-02-12           | 21   |
| CUS002      | ABDHUL        | 8765432109 | ADD2        | 2012-02-12           | 21   |
| CUS003      | GAYAN         | 7654321098 | ADD3        | 2012-02-12           | 21   |
| CUS004      | RADHA         | 6543210987 | ADD4        | 2012-02-12           | 21   |
| CUS005      | GURU          | NULL       | ADD5        | 2012-02-12           | 21   |
| CUS006      | MOHAN         | 4321098765 | ADD6        | 2012-02-12           | 21   |
| CUS007      | NAME7         | 3210987654 | ADD7        | 2012-02-12           | 21   |
| CUS008      | NAME8         | 2109876543 | ADD8        | 2013-02-12           | 21   |
| CUS009      | NAME9         | NULL       | ADD9        | 2013-02-12           | 21   |
| CUS010      | NAM10         | 9934567890 | ADD10       | 2013-02-12           | 21   |
| CUS011      | NAM11         | 9875678910 | ADD11       | 2013-02-12           | 21   |
| NULL        | NULL          | NULL       | NULL        | NULL                 | NULL |

#### LIBRARY CARD MASTER

| CARD_ID | DESCRIPTION | AMOUNT | NUMBER_OF_YEARS |
|---------|-------------|--------|-----------------|
| CR001   | Silver      | 200    | 5               |
| CR002   | Gold        | 400    | 9               |
| CR003   | Platinum    | 600    | 8               |
| CR004   | VISA        | 800    | 7               |
| CR005   | CREDIT      | 1200   | 6               |
| NULL    | NULL        | NULL   | NULL            |

#### CUSTOMER CARD DETAILS

| CUSTOMER_ID | CARD_ID | ISSUE_DATE |
|-------------|---------|------------|
| CUS001      | CR001   | 2012-05-13 |
| CUS002      | CR002   | 2012-05-13 |
| CUS003      | CR002   | 2013-05-13 |
| CUS004      | CR003   | 2013-05-13 |
| CUS005      | CR003   | 2012-05-13 |
| NULL        | NULL    | NULL       |

#### CUSTOMER ISSUE DETAILS

| ISSUE_ID | CUSTOMER_ID | MOVIE_ID | ISSUE_DATE | RETURN_DATE | ACTUAL_DATE_RETURN |
|----------|-------------|----------|------------|-------------|--------------------|
| IS001    | CUS001      | MV001    | 2012-05-13 | 2012-05-13  | 2012-05-13         |
| IS002    | CUS001      | MV001    | 2012-05-01 | 2012-05-16  | 2012-05-16         |
| IS003    | CUS002      | MV004    | 2012-05-02 | 2012-05-06  | 2012-05-16         |
| IS004    | CUS002      | MV004    | 2012-04-03 | 2012-04-16  | 2012-04-20         |
| IS005    | CUS002      | MV009    | 2012-04-04 | 2012-04-16  | 2012-04-20         |
| IS006    | CUS003      | MV002    | 2012-03-30 | 2012-04-15  | 2012-04-20         |
| IS007    | CUS003      | MV003    | 2012-04-20 | 2012-05-05  | 2012-05-05         |
| IS008    | CUS003      | MV005    | 2012-04-21 | 2012-05-07  | 2012-05-25         |
| IS009    | CUS003      | MV001    | 2012-04-22 | 2012-05-07  | 2012-05-25         |
| IS010    | CUS003      | MV009    | 2012-04-22 | 2012-05-07  | 2012-05-25         |
| IS011    | CUS003      | MV010    | 2012-04-23 | 2012-05-07  | 2012-05-25         |
| IS012    | CUS003      | MV010    | 2012-04-24 | 2012-05-07  | 2012-05-25         |
| IS013    | CUS003      | MV008    | 2012-04-25 | 2012-05-07  | 2012-05-25         |
| IS014    | CUS004      | MV007    | 2012-04-26 | 2012-05-07  | 2012-05-25         |
| IS015    | CUS004      | MV006    | 2012-04-27 | 2012-05-07  | 2012-05-25         |
| IS016    | CUS004      | MV006    | 2012-04-28 | 2012-05-07  | 2012-05-25         |
| IS017    | CUS004      | MV001    | 2012-04-29 | 2012-05-07  | 2012-05-25         |
| IS018    | CUS010      | MV008    | 2012-04-24 | 2012-05-07  | 2012-05-25         |
| IS019    | CUS011      | MV009    | 2012-04-27 | 2012-05-07  | 2012-05-25         |
| NULL     | NULL        | NULL     | NULL       | NULL        | NULL               |

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

```
SELECT m.MOVIE_NAME, count(ISSUE_ID) ISSUE_COUNT FROM
movies_master m LEFT JOIN customer_issue_details c ON m.MOVIE_ID=c.MOVIE_ID
GROUP BY m.movie_name
```

ORDER BY ISSUE\_COUNT DESC,MOVIE\_NAME;

| MOVIE_NAME    | ISSUE_COUNT |
|---------------|-------------|
| DIEHARD       | 4           |
| GWW           | 3           |
| CAS           | 2           |
| DARK KNIGHT   | 2           |
| SHAWN OF DEAD | 2           |
| TITANIC       | 2           |
| INCEPTION     | 1           |
| OFFICE S      | 1           |
| THE MATRIX    | 1           |
| YOUNG FRANKEN | 1           |
| THE NOTE BOOK | 0           |

2. Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT\_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

```
SELECT CUSTOMER_ID,CUSTOMER_NAME,AGE,ifnull(
concat('+91-',substring(contact_no,1,3),'-',
substring(contact_no,4,3),'-',substring(contact_no,7)),'N/A') CONTACT_ISD
FROM customer_master WHERE age>25 and year(date_of_registration)='2012'
ORDER BY age,CUSTOMER_NAME;
```

| CUSTOMER_ID | CUSTOMER_NAME | AGE | CONTACT_ISD |
|-------------|---------------|-----|-------------|
|             |               |     |             |

3. Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order. Hint: Use NO\_OF\_MOVIES as alias name for number of movies.

```
SELECT MOVIE_CATEGORY,count(MOVIE_ID) NO_OF_MOVIES FROM
movies_master GROUP BY MOVIE_CATEGORY
ORDER BY NO_OF_MOVIES DESC,MOVIE_CATEGORY;
```

| MOVIE_CATEGORY | NO_OF_MOVIES |
|----------------|--------------|
| ACTION         | 4            |
| ROMANCE        | 4            |
| COMEDY         | 3            |

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

```
SELECT count(c.customer_id) CUSTOMER_COUNT FROM
library_card_master l JOIN customer_card_details c ON l.CARD_ID=c.CARD_ID
WHERE description='Gold';
```

| CUSTOMER_COUNT |
|----------------|
| 2              |

5. Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order. Use REGISTERED\_YEAR as alias name for year of registration.

```
SELECT c.customer_id,c.customer_name,
year(c.DATE_OF_REGISTRATION) REGISTERED_YEAR,cd.card_id,cd.issue_date FROM
customer_master c JOIN customer_card_details cd ON c.customer_id=cd.customer_id
ORDER BY CUSTOMER_NAME DESC;
```

| customer_id | customer_name | REGISTERED_YEAR | card_id | issue_date |
|-------------|---------------|-----------------|---------|------------|
| CUS004      | RADHA         | 2012            | CR003   | 2013-05-13 |
| CUS005      | GURU          | 2012            | CR003   | 2012-05-13 |
| CUS003      | GAYAN         | 2012            | CR002   | 2013-05-13 |
| CUS001      | AMIT          | 2012            | CR001   | 2012-05-13 |
| CUS002      | ABDHUL        | 2012            | CR002   | 2012-05-13 |

6. Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order by customer name.

```
SELECT ci.issue_id,ci.CUSTOMER_ID,c.CUSTOMER_NAME FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
WHERE customer_name LIKE 'R%' and ci.actual_date_return>ci.return_date
```

ORDER BY customer\_name;

| issue_id | CUSTOMER_ID | CUSTOMER_NAME |
|----------|-------------|---------------|
| IS014    | CUS004      | RADHA         |
| IS015    | CUS004      | RADHA         |
| IS016    | CUS004      | RADHA         |
| IS017    | CUS004      | RADHA         |

**7. Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered. For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details. AMOUNT\_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount. Example Assume 500 is the amount then dollar value will be \$10. Hint: Use AMOUNT\_DOLLAR as alias name for amount in dollar. Display the records in ascending order based on customer name.**

```
SELECT c.CUSTOMER_ID,c.CUSTOMER_NAME,l.card_id,l.DESCRIPTION,
concat('$',round(amount/52.42)) AMOUNT_DOLLAR FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
JOIN customer_card_details cc ON cc.customer_id=c.customer_id
JOIN library_card_master l ON cc.card_id=l.card_id
WHERE c.DATE_OF_REGISTRATION=ci.issue_date
ORDER BY customer_name;
```

| CUSTOMER_ID | CUSTOMER_NAME | card_id | DESCRIPTION | AMOUNT_DOLLAR |
|-------------|---------------|---------|-------------|---------------|
|             |               |         |             |               |

**8. Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'. Display customer name in upper case. Hint: Use CUSTOMER\_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.**

```
SELECT CUSTOMER_ID,upper(CUSTOMER_NAME) CUSTOMER_NAME,contact_no,contact_add
FROM
customer_master WHERE contact_add LIKE '%Nagar' and
customer_id NOT IN (SELECT customer_id FROM customer_card_details)
and customer_id IN (SELECT customer_id FROM customer_issue_details)
```

ORDER BY CUSTOMER\_NAME;

| CUSTOMER_ID | CUSTOMER_NAME | contact_no | contact_add |
|-------------|---------------|------------|-------------|
|-------------|---------------|------------|-------------|

**9. Write a query to display the movie id, movie name, release year, director name of movies acted by the lead actor 1 who acted maximum number of movies. Display the records sorted in ascending order based on movie name.**

```
SELECT movie_id, movie_name, release_date, director FROM movies_master
WHERE lead_role_1 IN (SELECT lead_role_1 FROM
(SELECT lead_role_1, count(movie_id) ct FROM movies_master
GROUP BY lead_role_1) t WHERE t.ct >= ALL (SELECT count(movie_id)
FROM movies_master GROUP BY lead_role_1)) ORDER BY movie_name;
```

| movie_id | movie_name     | release_date | director |
|----------|----------------|--------------|----------|
| MV008    | CAS            | 2012-05-13   | DIR8     |
| MV001    | DIEHARD        | 2012-05-13   | DIR1     |
| MV009    | GWW            | 2012-05-13   | DIR9     |
| MV006    | SHAWN OF DEAD  | 2012-05-13   | DIR6     |
| MV002    | THE MATRIX     | 2012-05-13   | DIR2     |
| MV011    | THE NOTE BOOK  | 2012-05-13   | DIR11    |
| MV010    | TITANIC        | 2012-05-13   | DIR10    |
| MV007    | YOUNG FRANK... | 2012-05-13   | DIR7     |

**10. Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0. <br> Hint: Use MOVIE\_COUNT as alias name for number of movies issued.**

```
SELECT c.customer_name, count(ci.movie_id) MOVIE_COUNT FROM
customer_master c LEFT JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
GROUP BY c.customer_id ORDER BY c.customer_name;
```

| customer_name | MOVIE_COUNT |
|---------------|-------------|
| ABDHUL        | 3           |
| AMIT          | 2           |
| GAYAN         | 8           |
| GURU          | 0           |
| MOHAN         | 0           |
| NAM10         | 1           |
| NAM11         | 1           |
| NAME7         | 0           |
| NAME8         | 0           |
| NAME9         | 0           |
| RADHA         | 4           |

**11. Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number. Serial number can be generated from the issue id , that is last two characters of issue id is the serial number. For Example Assume the issue id is I00005 then the serial number is 05 Hint: Alias name for serial number is 'SERIAL\_NO'**

```
SELECT substring(ci.issue_id,-2) SERIAL_NO,ci.issue_id,c.customer_id,c.customer_name,
m.movie_id,m.movie_name FROM customer_master c JOIN customer_issue_details ci
ON c.customer_id=ci.customer_id JOIN movies_master m ON m.movie_id=ci.movie_id
ORDER BY SERIAL_NO;
```

| SERIAL_NO | issue_id | customer_id | customer_name | movie_id | movie_name    |
|-----------|----------|-------------|---------------|----------|---------------|
| 01        | IS001    | CUS001      | AMIT          | MV001    | DIEHARD       |
| 02        | IS002    | CUS001      | AMIT          | MV001    | DIEHARD       |
| 03        | IS003    | CUS002      | ABDHUL        | MV004    | DARK KNIGHT   |
| 04        | IS004    | CUS002      | ABDHUL        | MV004    | DARK KNIGHT   |
| 05        | IS005    | CUS002      | ABDHUL        | MV009    | GWW           |
| 06        | IS006    | CUS003      | GAYAN         | MV002    | THE MATRIX    |
| 07        | IS007    | CUS003      | GAYAN         | MV003    | INCEPTION     |
| 08        | IS008    | CUS003      | GAYAN         | MV005    | OFFICE S      |
| 09        | IS009    | CUS003      | GAYAN         | MV001    | DIEHARD       |
| 10        | IS010    | CUS003      | GAYAN         | MV009    | GWW           |
| 11        | IS011    | CUS003      | GAYAN         | MV010    | TITANIC       |
| 12        | IS012    | CUS003      | GAYAN         | MV010    | TITANIC       |
| 13        | IS013    | CUS003      | GAYAN         | MV008    | CAS           |
| 14        | IS014    | CUS004      | RADHA         | MV007    | YOUNG FRAN... |
| 15        | IS015    | CUS004      | RADHA         | MV006    | SHAWN OF D... |
| 16        | IS016    | CUS004      | RADHA         | MV006    | SHAWN OF D... |
| 17        | IS017    | CUS004      | RADHA         | MV001    | DIEHARD       |
| 18        | IS018    | CUS010      | NAM10         | MV008    | CAS           |
| 19        | IS019    | CUS011      | NAM11         | MV009    | GWW           |

**12. Write a query to display the issue id, issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.**

```
SELECT ci.issue_id,ci.issue_date,c.customer_id,c.customer_name,c.contact_no FROM
customer_master c JOIN customer_issue_details ci ON c.customer_id=ci.customer_id
and year(ci.issue_date)='2013' ORDER BY ci.issue_date DESC;
```

| issue_id | issue_date | customer_id | customer_name | contact_no |
|----------|------------|-------------|---------------|------------|
|----------|------------|-------------|---------------|------------|

**13. Write a query to display movie id , movie name and actor names of movies which are not issued to any customers. <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.**

```
SELECT movie_id,movie_name,concat(lead_role_1,' & ',lead_role_2) ACTOR FROM movies_master
```

```
WHERE movie_id NOT IN (SELECT movie_id FROM customer_issue_details) ORDER BY movie_name;
```

| movie_id | movie_name    | ACTOR   |
|----------|---------------|---------|
| MV011    | THE NOTE BOOK | L1 & L2 |

**14. Write a query to display the director's name, movie name and lead\_actor\_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR\_NAME as alias name for director name column Display the records sorted in ascending order based on director\_name and then by movie\_name in descending order.**

```
SELECT upper(director) DIRECTOR_NAME,movie_name,lead_role_1 FROM movies_master
GROUP BY director HAVING count(movie_id)>1 ORDER BY director,movie_name DESC;
```

| DIRECTOR_NAME | movie_name | lead_role_1 |
|---------------|------------|-------------|
|               |            |             |

**15. Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <br> Hint:Use NO\_OF\_CUSTOMERS as alias name for number of customers.**

```
SELECT count(customer_id) NO_OF_CUSTOMER FROM customer_master
WHERE contact_no is not null and year(date_of_registration)='2012';
```

| NO_OF_CUSTOMER |
|----------------|
| 6              |

**16. Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT\_DETAILS as alias name for customer contact.**

```

SELECT c.customer_name,ifnull(c.contact_no,c.contact_add)
CONTACT_DETAILS,l.card_id,l.description FROM
customer_master c LEFT JOIN customer_card_details cc ON c.customer_id=cc.customer_id
LEFT JOIN library_card_master l ON l.card_id=cc.card_id
ORDER BY customer_name;

```

| customer_name | CONTACT_DETAILS | card_id | description |
|---------------|-----------------|---------|-------------|
| ABDHUL        | 8765432109      | CR002   | Gold        |
| AMIT          | 9876543210      | CR001   | Silver      |
| GAYAN         | 7654321098      | CR002   | Gold        |
| GURU          | ADD5            | CR003   | Platinum    |
| MOHAN         | 4321098765      | NULL    | NULL        |
| NAM10         | 9934567890      | NULL    | NULL        |
| NAM11         | 9875678910      | NULL    | NULL        |
| NAME7         | 3210987654      | NULL    | NULL        |
| NAME8         | 2109876543      | NULL    | NULL        |
| NAME9         | ADD9            | NULL    | NULL        |
| RADHA         | 6543210987      | CR003   | Platinum    |

**17. Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO\_OF\_TIMES as alias name for number of times**

```

SELECT ci.customer_id,c.customer_name,count(ci.movie_id) NO_OF_TIMES FROM
customer_issue_details ci JOIN customer_master c ON c.customer_id=ci.customer_id
GROUP BY ci.customer_id,ci.movie_id HAVING count(movie_id)>1
ORDER BY customer_name DESC;

```

| customer_id | customer_name | NO_OF_TIMES |
|-------------|---------------|-------------|
| CUS004      | RADHA         | 2           |
| CUS003      | GAYAN         | 2           |
| CUS001      | AMIT          | 2           |
| CUS002      | ABDHUL        | 2           |

**18. Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-**

**456-2345" format.&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.**

```
SELECT c.customer_id,c.customer_name,concat('+91-',substring(c.contact_no,1,3),'-',
substring(c.contact_no,4,3),'-',substring(c.contact_no,7)) CONTACT_ISD
,m.movie_category,count(cc.movie_id) NO_OF_MOVIES FROM customer_master c JOIN
customer_issue_details cc
ON c.customer_id=cc.customer_id JOIN movies_master m ON m.movie_id=cc.movie_id
GROUP BY c.customer_id,m.movie_category HAVING count(cc.movie_id)>1
ORDER BY customer_name,movie_category;
```

| customer_id | customer_name | CONTACT_ISD      | movie_category | NO_OF_MOVIES |
|-------------|---------------|------------------|----------------|--------------|
| CUS002      | ABDHUL        | +91-876-543-2109 | ACTION         | 2            |
| CUS001      | AMIT          | +91-987-654-3210 | ACTION         | 2            |
| CUS003      | GAYAN         | +91-765-432-1098 | ACTION         | 3            |
| CUS003      | GAYAN         | +91-765-432-1098 | ROMANCE        | 4            |
| CUS004      | RADHA         | +91-654-321-0987 | COMEDY         | 3            |

**19. Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies. For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.**

```
SELECT cid.customer_id , customer_name FROM customer_master cm JOIN
customer_issue_details cid ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)>=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id)
UNION
```

```

SELECT cid.customer_id , customer_name FROM
customer_master cm JOIN customer_issue_details cid
ON cm.customer_id=cid.customer_id
GROUP BY customer_id , customer_name
HAVING count(movie_id)<=ALL(SELECT count(movie_id)
FROM customer_issue_details
GROUP BY customer_id) ORDER BY customer_name;

```

| customer_id | customer_name |
|-------------|---------------|
| CUS003      | GAYAN         |
| CUS010      | NAM10         |
| CUS011      | NAM11         |

**20.**Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once. Hint: Use NO\_OF\_TIMES as alias name Display the records in ascending order based on customer name.

```

SELECT c.customer_id,c.customer_name,count(m.movie_id) NO_OF_TIMES FROM
customer_master c JOIN customer_issue_details cc ON c.customer_id=cc.customer_id
JOIN movies_master m ON m.movie_id=cc.movie_id
WHERE m.movie_category='Comedy'
GROUP BY c.customer_id HAVING count(m.movie_id)>1
ORDER BY customer_name;

```

| customer_id | customer_name | NO_OF_TIMES |
|-------------|---------------|-------------|
| CUS004      | RADHA         | 3           |

**21.**Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL\_COST. Display the records sorted in ascending order based on customer id

```

SELECT cid.customer_id, sum(m.rent_cost) TOTAL_COST FROM customer_issue_details cid
JOIN movies_master mm ON cid.movie_id=mm.movie_id GROUP BY cid.customer_id order by
customer_id;

```

| customer_id | TOTAL_COST |
|-------------|------------|
| CUS001      | 200        |
| CUS002      | 300        |
| CUS003      | 800        |
| CUS004      | 400        |
| CUS010      | 100        |
| CUS011      | 100        |

# LOAN

```
create database loan;
```

```
use loan;
```

```
CREATE TABLE loan_card_master
```

```
(
 loan_id varchar(6) PRIMARY KEY,
 loan_type varchar(15),
 duration_in_years int(2)
);
```

```
CREATE TABLE employee_master
```

```
(
 employee_id varchar(6) PRIMARY KEY,
 employee_name varchar(20),
 designation varchar(25),
 department varchar(25),
 gender char(1),
 date_of_birth date,
 date_of_joining date
);
```

```
CREATE TABLE item_master
```

```
(
 item_id varchar(6) PRIMARY KEY,
 item_description varchar(25),
```

```

issue_status char(1),
item_make varchar(25),
item_category varchar(20),
item_valuation int(6)

);

CREATE TABLE employee_card_details
(
 employee_id varchar(6) REFERENCES employee_master,
 loan_id varchar(6) REFERENCES loan_card_master,
 card_issue_date date
);

CREATE TABLE employee_issue_details
(
 issue_id varchar(6) PRIMARY KEY,
 employee_id varchar(6) REFERENCES employee_master,
 item_id varchar(6) REFERENCES item_master,
 issue_date date,
 return_date date
);

);

insert into loan_card_master values('L00001','Furniture',5);
insert into loan_card_master values('L00002','Stationary',0);
insert into loan_card_master values('L00003','Crockery',1);

```

```
insert into employee_issue_details values('ISS001','E00001','I00001','2012-02-03','2014-02-03');
insert into employee_issue_details values('ISS002','E00001','I00004','2012-02-03','2020-02-03');
insert into employee_issue_details values('ISS003','E00002','I00005','2013-01-03','2015-01-03');
insert into employee_issue_details values('ISS004','E00003','I00007','2010-07-04','2012-07-04');
insert into employee_issue_details values('ISS005','E00003','I00008','2010-07-04','2012-08-05');
insert into employee_issue_details values('ISS006','E00003','I00010','2012-03-14','2012-06-15');
insert into employee_issue_details values('ISS007','E00004','I00012','2013-04-14','2016-04-14');
insert into employee_issue_details values('ISS008','E00006','I00018','2012-08-18','2019-04-17');
insert into employee_issue_details values('ISS009','E00004','I00018','2013-04-18','2013-05-18');
```

```
insert into employee_master values('E00001','Ram','Manager','Finance','M','1973-12-01','2000-01-01');
insert into employee_master values('E00002','Abhay','Assistant Manager','Finance','M','1976-01-01','2006-12-01');
insert into employee_master values('E00003','Anita','Senior Executive','Marketing','F','1977-05-12','2007-03-21');
insert into employee_master values('E00004','Zuben','Manager','Marketing','M','1974-10-12','2003-07-23');
insert into employee_master values('E00005','Radhica','Manager','HR','F','1976-07-22','2004-01-23');
insert into employee_master values('E00006','John','Executive','HR','M','1983-11-08','2010-05-17');
```

```
insert into employee_card_details values('E00001','L00001','2000-01-01');
insert into employee_card_details values('E00001','L00002','2000-01-01');
insert into employee_card_details values('E00001','L00003','2002-12-14');
insert into employee_card_details values('E00002','L00001','2007-02-01');
```

```
insert into employee_card_details values('E00002','L00002','2007-03-11');
insert into employee_card_details values('E00003','L00001','2007-04-15');
insert into employee_card_details values('E00003','L00002','2007-04-15');
insert into employee_card_details values('E00003','L00003','2007-04-15');
```

```
INSERT INTO item_master VALUES ('I00001','Tea Table','Y','Wooden','Furniture',5000);
INSERT INTO item_master VALUES ('I00002','Dinning Table','N','Wooden','Furniture',15000);
INSERT INTO item_master VALUES ('I00003','Tea Table','N','Steel','Furniture',6000);
INSERT INTO item_master VALUES ('I00004','Side Table','Y','Wooden','Furniture',2000);
INSERT INTO item_master VALUES ('I00005','Side Table','Y','Steel','Furniture',1500);
INSERT INTO item_master VALUES ('I00006','Tea Table','N','Steel','Furniture',7000);
INSERT INTO item_master VALUES ('I00007','Dinning Chair','Y','Wooden','Furniture',1500);
INSERT INTO item_master VALUES ('I00008','Tea Table','Y','Wooden','Furniture',4000);
INSERT INTO item_master VALUES ('I00009','Sofa','N','Wooden','Furniture',18000);
INSERT INTO item_master VALUES ('I00010','Cupboard','Y','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00011','Cupboard','N','Steel','Furniture',14000);
INSERT INTO item_master VALUES ('I00012','Double Bed','Y','Wooden','Furniture',21000);
INSERT INTO item_master VALUES ('I00013','Double Bed','Y','Wooden','Furniture',20000);
INSERT INTO item_master VALUES ('I00014','Single Bed','Y','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00015','Single Bed','N','Steel','Furniture',10000);
INSERT INTO item_master VALUES ('I00016','Tea Set','Y','Glass','Crockery',3000);
INSERT INTO item_master VALUES ('I00017','Tea Set','Y','Bonechina','Crockery',4000);
INSERT INTO item_master VALUES ('I00018','Dinning Set','Y','Glass','Crockery',4500);
INSERT INTO item_master VALUES ('I00019','Dinning Set','N','Bonechina','Crockery',5000);
INSERT INTO item_master VALUES ('I00020','Pencil','Y','Wooden','Stationary',5);
```

```
INSERT INTO item_master VALUES ('I00021','Pen','Y','Plastic','Stationary',100);
INSERT INTO item_master VALUES ('I00022','Pen','N','Plastic','Stationary',200);
```

## LOAN CARD MASTER

| loan_id | loan_type  | duration_in_years |
|---------|------------|-------------------|
| L00001  | Furniture  | 5                 |
| L00002  | Stationary | 0                 |
| L00003  | Crockery   | 1                 |
| NULL    | NULL       | NULL              |

## EMPLOYEE CARD DETAILS

| employee_id | loan_id | card_issue_date |
|-------------|---------|-----------------|
| E00001      | L00001  | 2000-01-01      |
| E00001      | L00002  | 2000-01-01      |
| E00001      | L00003  | 2002-12-14      |
| E00002      | L00001  | 2007-02-01      |
| E00002      | L00002  | 2007-03-11      |
| E00003      | L00001  | 2007-04-15      |
| E00003      | L00002  | 2007-04-15      |
| E00003      | L00003  | 2007-04-15      |

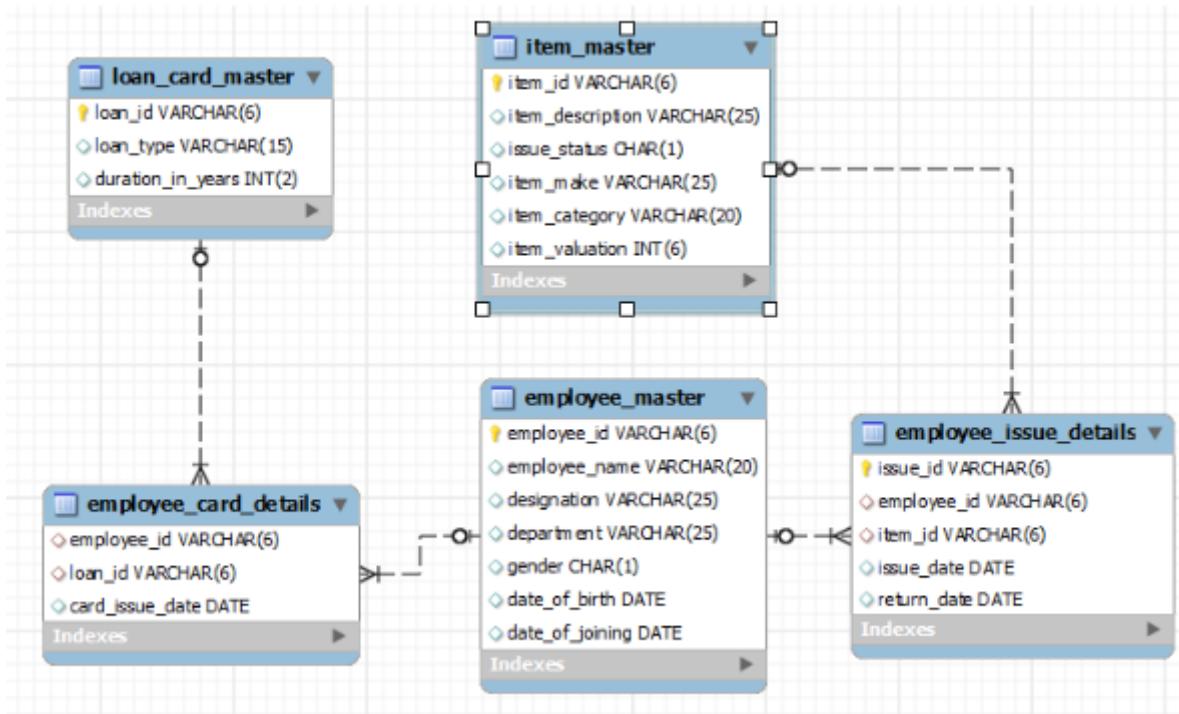
## **EMPLOYEE ISSUE DETAILS**

| issue_id | employee_id | item_id | issue_date | return_date |
|----------|-------------|---------|------------|-------------|
| ISS001   | E00001      | I00001  | 2012-02-03 | 2014-02-03  |
| ISS002   | E00001      | I00004  | 2012-02-03 | 2020-02-03  |
| ISS003   | E00002      | I00005  | 2013-01-03 | 2015-01-03  |
| ISS004   | E00003      | I00007  | 2010-07-04 | 2012-07-04  |
| ISS005   | E00003      | I00008  | 2010-07-04 | 2012-08-05  |
| ISS006   | E00003      | I00010  | 2012-03-14 | 2012-06-15  |
| ISS007   | E00004      | I00012  | 2013-04-14 | 2016-04-14  |
| ISS008   | E00006      | I00018  | 2012-08-18 | 2019-04-17  |
| ISS009   | E00004      | I00018  | 2013-04-18 | 2013-05-18  |
| NUL      | NUL         | NUL     | NUL        | NUL         |

## **EMPLOYEE MASTER**

## ITEM MASTER

| item_id | item_description | issue_status | item_make | item_category | item_valuation |
|---------|------------------|--------------|-----------|---------------|----------------|
| I00001  | Tea Table        | Y            | Wooden    | Furniture     | 5000           |
| I00002  | Dinning Table    | N            | Wooden    | Furniture     | 15000          |
| I00003  | Tea Table        | N            | Steel     | Furniture     | 6000           |
| I00004  | Side Table       | Y            | Wooden    | Furniture     | 2000           |
| I00005  | Side Table       | Y            | Steel     | Furniture     | 1500           |
| I00006  | Tea Table        | N            | Steel     | Furniture     | 7000           |
| I00007  | Dinning Chair    | Y            | Wooden    | Furniture     | 1500           |
| I00008  | Tea Table        | Y            | Wooden    | Furniture     | 4000           |
| I00009  | Sofa             | N            | Wooden    | Furniture     | 18000          |
| I00010  | Cupboard         | Y            | Steel     | Furniture     | 10000          |
| I00011  | Cupboard         | N            | Steel     | Furniture     | 14000          |
| I00012  | Double Bed       | Y            | Wooden    | Furniture     | 21000          |
| I00013  | Double Bed       | Y            | Wooden    | Furniture     | 20000          |
| I00014  | Single Bed       | Y            | Steel     | Furniture     | 10000          |
| I00015  | Single Bed       | N            | Steel     | Furniture     | 10000          |
| I00016  | Tea Set          | Y            | Glass     | Crockery      | 3000           |
| I00017  | Tea Set          | Y            | Bonechina | Crockery      | 4000           |
| I00018  | Dinning Set      | Y            | Glass     | Crockery      | 4500           |
| I00019  | Dinning Set      | N            | Bonechina | Crockery      | 5000           |
| I00020  | Pencil           | Y            | Wooden    | Stationary    | 5              |
| I00021  | Pen              | Y            | Plastic   | Stationary    | 100            |
| I00022  | Pen              | N            | Plastic   | Stationary    | 200            |
| NUL     | NUL              | NUL          | NUL       | NUL           | NUL            |



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

```
SELECT item_category, count(item_id) Count_category FROM
item_master GROUP BY item_category ORDER BY Count_category DESC;
```

| item_category | Count_category |
|---------------|----------------|
| Furniture     | 15             |
| Crockery      | 4              |
| Stationary    | 3              |

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

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

| No_of_Employees |
|-----------------|
| 2               |

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

```
SELECT employee_id, employee_name, designation, department FROM employee_master
```

```

WHERE employee_id NOT IN (SELECT employee_id FROM employee_issue_details)
ORDER BY employee_id;

```

| employee_id | employee_name | designation | department |
|-------------|---------------|-------------|------------|
| E00005      | Radhica       | Manager     | HR         |
| NULL        | NULL          | NULL        | NULL       |

4. Write a query to display the employee id, employee name who was issued an item of highest valuation. In case of multiple records, display the records sorted in ascending order based on employee id.[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

```

SELECT employee_id,employee_name FROM employee_master
WHERE employee_id IN(SELECT employee_id FROM employee_issue_details
WHERE item_id IN (SELECT item_id FROM item_master
WHERE item_valuation=(SELECT max(item_valuation) FROM
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)));

```

| employee_id | employee_name |
|-------------|---------------|
| E00004      | Zuben         |
| NULL        | NULL          |

5. Write a query to display issue\_id, employee\_id, employee\_name. Display the records sorted in ascending order based on issue id.

```

SELECT eid.issue_id, eid.employee_id, em.employee_name
FROM employee_master em JOIN employee_issue_details eid
ON em.employee_id=eid.employee_id ORDER BY eid.issue_id;

```

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

**6. Write a query to display employee id, employee name who don't have loan cards. Display the records sorted in ascending order based on employee id.**

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

| employee_id | employee_name |
|-------------|---------------|
| E00004      | Zuben         |
| E00005      | Radhica       |
| E00006      | John          |
| NULL        | NULL          |

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

```
SELECT count(loan_id) No_of_Cards FROM
employee_card_details WHERE employee_id IN
(SELECT employee_id FROM employee_master WHERE employee_name='Ram');
```

**(or)**

```
SELECT count(loan_id) No_of_Cards FROM
employee_card_details c JOIN employee_master e
ON c.employee_id = e.employee_id
WHERE e.employee_name= 'Ram';
```

| No_of_Cards |
|-------------|
| 3           |

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

```
SELECT count(e.employee_id) Count_Stationary
FROM employee_card_details e JOIN loan_card_master l
ON e.loan_id=l.loan_id WHERE l.loan_type='Stationary';
```

| Count_Stationary |
|------------------|
| 3                |

**9. Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then**

```
SELECT e.employee_id,employee_name,count(e.item_id) Count FROM
```

```

employee_issue_details e JOIN employee_master em ON e.employee_id=em.employee_id
GROUP BY e.employee_id ORDER BY count DESC,e.employee_id;

```

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

**10.** Write a query to display the employee id, employee name who was issued an item of minimum valuation. In case of multiple records, display them sorted in ascending order based on employee id. [Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

```

SELECT employee_id,employee_name FROM employee_master
WHERE employee_id IN(SELECT employee_id FROM employee_issue_details
WHERE item_id IN (SELECT item_id FROM item_master
WHERE item_valuation=(SELECT min(item_valuation) FROM
item_master i JOIN employee_issue_details e ON i.item_id=e.item_id)))
ORDER BY employee_id;

```

| employee_id | employee_name |
|-------------|---------------|
| E00002      | Abhay         |
| E00003      | Anita         |
| NULL        | NULL          |

**11.** Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL\_VALUATION. Display the records sorted in ascending order based on employee id. Consider only employees who have been issued atleast 1 item.

```

SELECT e.employee_id,em.employee_name,sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id ORDER BY employee_id;

```

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00001      | Ram           | 7000            |
| E00002      | Abhay         | 1500            |
| E00003      | Anita         | 15500           |
| E00004      | Zuben         | 25500           |
| E00006      | John          | 4500            |

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

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE datediff(ei.return_date,ei.issue_date)>365
ORDER BY employee_id;
```

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

**13.** Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT\_ITEMS. Display the records sorted in ascending order on employee id.

```
SELECT e.employee_id,e.employee_name,count(ei.item_id) COUNT_ITEMS FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON ei.item_id=i.item_id
WHERE i.item_category='Furniture'
GROUP BY ei.employee_id HAVING count(ei.item_id)>1;
```

| employee_id | employee_name | COUNT_ITEMS |
|-------------|---------------|-------------|
| E00001      | Ram           | 2           |
| E00003      | Anita         | 3           |

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

```
SELECT gender,count(employee_id) FROM employee_master
```

```
GROUP BY gender ORDER BY gender;
```

| gender | count(employee_id) |
|--------|--------------------|
| F      | 2                  |
| M      | 4                  |

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

```
SELECT employee_id,employee_name FROM employee_master
```

```
WHERE year(date_of_joining)>'2005'
```

```
ORDER BY employee_id;
```

| employee_id | employee_name |
|-------------|---------------|
| E00002      | Abhay         |
| E00003      | Anita         |
| E00006      | John          |
| NULL        | NULL          |

**16. Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No\_of\_Furnitures. Display the records sorted in ascending order based on issue\_status.**

```
SELECT issue_status,count(item_id) No_of_Furniture FROM
```

```
item_master WHERE item_category='Furniture'
```

```
GROUP BY issue_status ORDER BY issue_status;
```

| issue_status | No_of_Furniture |
|--------------|-----------------|
| N            | 6               |
| Y            | 9               |

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

```
SELECT item_category,item_make,item_description,count(item_id) No_of_items FROM
```

```
item_master GROUP BY item_category,item_make,item_description
```

```
ORDER BY item_category,item_make,item_description;
```

| item_category | item_make | item_description | No_of_items |
|---------------|-----------|------------------|-------------|
| Crockery      | Bonechina | Dinning Set      | 1           |
| Crockery      | Bonechina | Tea Set          | 1           |
| Crockery      | Glass     | Dinning Set      | 1           |
| Crockery      | Glass     | Tea Set          | 1           |
| Furniture     | Steel     | Cupboard         | 2           |
| Furniture     | Steel     | Side Table       | 1           |
| Furniture     | Steel     | Single Bed       | 2           |
| Furniture     | Steel     | Tea Table        | 2           |
| Furniture     | Wooden    | Dinning Chair    | 1           |
| Furniture     | Wooden    | Dinning Table    | 1           |
| Furniture     | Wooden    | Double Bed       | 2           |
| Furniture     | Wooden    | Side Table       | 1           |
| Furniture     | Wooden    | Sofa             | 1           |
| Furniture     | Wooden    | Tea Table        | 2           |
| Stationary    | Plastic   | Pen              | 2           |
| Stationary    | Wooden    | Pencil           | 1           |

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

```
SELECT e.employee_id,employee_name,i.item_id,i.item_description FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
WHERE month(ei.issue_date)='01' and year(ei.issue_date)='2013'
ORDER BY employee_id,item_id;
```

| employee_id | employee_name | item_id | item_description |
|-------------|---------------|---------|------------------|
| E00002      | Abhay         | I00005  | Side Table       |

**19. Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.Give the alias name for category count as COUNT\_CATEGORY.Display the records sorted in ascending order based on employee id.**

```
SELECT ei.employee_id,e.employee_name,count(DISTINCT i.item_category) COUNT_CATEGORY FROM
employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
JOIN item_master i ON i.item_id=ei.item_id
```

```

GROUP BY ei.employee_id
HAVING COUNT_CATEGORY >= 2
ORDER BY employee_id;

```

| employee_id | employee_name | COUNT_CATEGORY |
|-------------|---------------|----------------|
| E00004      | Zuben         | 2              |

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

```

SELECT item_id, item_description FROM item_master
WHERE item_id NOT IN (SELECT item_id from employee_issue_details)
ORDER BY item_id;

```

| item_id | item_description |
|---------|------------------|
| I00002  | Dinning Table    |
| I00003  | Tea Table        |
| I00006  | Tea Table        |
| I00009  | Sofa             |
| I00011  | Cupboard         |
| I00013  | Double Bed       |
| I00014  | Single Bed       |
| I00015  | Single Bed       |
| I00016  | Tea Set          |
| I00017  | Tea Set          |
| I00019  | Dinning Set      |
| I00020  | Pencil           |
| I00021  | Pen              |
| I00022  | Pen              |
| NULL    | NULL             |

**21. Write a query to display the employee id, employee name and total valuation for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed. ]**

```

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id = i.item_id
JOIN employee_master em ON em.employee_id = e.employee_id

```

```

GROUP BY e.employee_id HAVING sum(i.item_valuation)<=ALL(
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id);

```

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00002      | Abhay         | 1500            |

**22. Write a query to display the employee id, employee name, card issue date and card valid date. Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD\_VALID\_DATE.[Hint: Validity in years for the loan card is given in loan\_card\_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'. ]**

```

SELECT e.employee_id,e.employee_name,card_issue_date,
case
when l.duration_in_years>0 then date_add(ec.card_issue_date,interval l.duration_in_years year)
when l.duration_in_years=0 then 'No Validity Date' end CARD_VALID_DATE
FROM
employee_master e JOIN employee_card_details ec ON e.employee_id=ec.employee_id
JOIN loan_card_master l ON l.loan_id=ec.loan_id
ORDER BY employee_name,CARD_VALID_DATE;

```

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

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

```
SELECT DISTINCT e.employee_id,e.employee_name FROM
```

```

employee_master e JOIN employee_issue_details ei ON e.employee_id=ei.employee_id
WHERE e.employee_id NOT IN (SELECT employee_id FROM employee_issue_details
WHERE year(issue_date)='2013')
ORDER BY employee_id;

```

| employee_id | employee_name |
|-------------|---------------|
| E00001      | Ram           |
| E00003      | Anita         |
| E00006      | John          |

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

```

SELECT issue_id, eid.employee_id, employee_name, im.item_id, item_description, issue_date
FROM employee_issue_details eid JOIN employee_master em ON eid.employee_id=em.employee_id
JOIN item_master im ON eid.item_id=im.item_id
ORDER BY issue_date DESC, issue_id;

```

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

**25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]**

```

SELECT e.employee_id, em.employee_name, sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id HAVING sum(i.item_valuation)>=ALL(

```

```
SELECT sum(i.item_valuation) TOTAL_VALUATION FROM
item_master i JOIN employee_issue_details e ON e.item_id=i.item_id
JOIN employee_master em ON em.employee_id=e.employee_id
GROUP BY e.employee_id);
```

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00004      | Zuben         | 25500           |

### 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;
```

| MOVIE_NAME         | ISSUE_COUNT |
|--------------------|-------------|
| DIE HARD           | 4           |
| GONE WITH THE WIND | 3           |
| CASABLANCA         | 2           |
| SHAUN OF THE DEAD  | 2           |
| THE DARK KNIGHT    | 2           |
| TITANIC            | 2           |

|                    |   |
|--------------------|---|
| INCEPTION          | 1 |
| OFFICE SPACE       | 1 |
| THE MATRIX         | 1 |
| YOUNG FRANKENSTEIN | 1 |
| THE NOTEBOOK       | 0 |

2. Please follow instructions given below.

Write a query to display id, name, age, contact no of customers whose age is greater than 25 and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT\_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

4 rows

```
select customer_id,customer_name,age,
ifnull(concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',
'substring(contact_no,7,4)),'N/A')

as CONTACT_ISD from customer_master where age>25 and year(date_of_registration)=2012

order by age,customer_name;
```

| CUSTOMER_ID | CUSTOMER_NAME | AGE | CONTACT_ISD      |
|-------------|---------------|-----|------------------|
| C00007      | GEETHA REDDY  | 30  | +91-897-616-7890 |
| C00005      | SHIV PRASAD   | 30  | N/A              |

|        |             |    |                  |
|--------|-------------|----|------------------|
| C00002 | AGNESH      | 35 | +91-892-315-6781 |
| C00004 | RAJIB MITRA | 45 | +91-983-035-6781 |

3. Please follow instructions given below.

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

Hint: Use NO\_OF\_MOVIES as alias name for number of movies.

3 rows

Ans:

```
select movie_category,count(movie_id) as NO_OF_MOVIES from movies_master group by movie_category
order by NO_OF_MOVIES desc,movie_category asc;
```

| MOVIE_CATEGORY | NO_OF_MOVIES |
|----------------|--------------|
| ACTION         | 4            |
| ROMANCE        | 4            |
| COMEDY         | 3            |

4. Please follow instructions given below.

Write a query to display the number of customers having card with description “Gold card”. <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';
```

|                |
|----------------|
| CUSTOMER_COUNT |
| 2              |

4. Please follow instructions given below.

Write a query to display the customer id, customer name, year of registration, library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

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

5 rows

```
select cm.customer_id,cm.customer_name,year(cm.date_of_registration) as
REGISTERED_YEAR,ccd.card_id,ccd.issue_date
from customer_master cm join customer_card_details ccd on cm.customer_id=ccd.customer_id
order by cm.customer_name desc;
```

| CUSTOMER_ID | CUSTOMER_NAME  | REGISTERED_YEAR | CARD_ID | ISSUE_DATE |
|-------------|----------------|-----------------|---------|------------|
| C00003      | T RAMACHANDRAN | 2012            | CRD002  | 2012-11-02 |
| C00005      | SHIV PRASAD    | 2012            | CRD003  | 2012-12-26 |
| C00004      | RAJIB MITRA    | 2012            | CRD003  | 2012-11-21 |
| C00001      | NITIN          | 2012            | CRD001  | 2012-10-15 |
| C00002      | AGNESH         | 2012            | CRD002  | 2012-12-01 |

5. Please follow instructions given below.

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

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

2 rows

```
select cid.issue_id,cid.customer_id,cm.customer_name from customer_issue_details cid join
customer_master cm on cid.customer_id=cm.customer_id where cm.customer_name like 'R%'
and cid.actual_date_return>cid.return_date order by cm.customer_name;
```

| ISSUE_ID | CUSTOMER_ID | CUSTOMER_NAME |
|----------|-------------|---------------|
| I00008   | C00010      | RAGHAV SINGH  |
| I00007   | C00004      | RAJIB MITRA   |

6. Please follow instructions given below.

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT\_DOLLAR = amount/52.42 and round it to zero decimal places and display as \$Amount.  
Example Assume 500 is the amount then dollar value will be \$10.

Hint: Use AMOUNT\_DOLLAR as alias name for amount in dollar.

Display the records in ascending order based on customer name.

```
SELECT ccd.customer_id, customer_name, ccd.card_id, description,concat('$',round(amount/52.42,0))
AMOUNT_DOLLAR FROM customer_master cm INNER JOIN customer_card_details ccd ON
cm.customer_id=ccd.customer_id INNER JOIN library_card_master lcm ON ccd.card_id=lcm.card_id
```

```
INNER JOIN customer_issue_details cid ON cid.customer_id = cm.customer_id WHERE
cm.date_of_registration=cid.issue_date order by customer_name;
```

| CUSTOMER_ID | CUSTOMER_NAME  | CARD_ID | DESCRIPTION   | AMOUNT_DOLLAR |
|-------------|----------------|---------|---------------|---------------|
| C00001      | NITIN          | CRD001  | SILVER CARD   | \$19          |
| C00004      | RAJIB MITRA    | CRD003  | PLATINUM CARD | \$57          |
| C00003      | T RAMACHANDRAN | CRD002  | GOLD CARD     | \$38          |

7. Please follow instructions given below.

Write a query to display the customer id, customer name, contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use CUSTOMER\_NAME as alias name for customer name.  
Display the details sorted in ascending order based on customer name.

```
SELECT customer_id , upper(customer_name) CUSTOMER_NAME,contact_no,contact_address FROM
customer_master WHERE customer_id NOT IN (select customer_id from customer_card_details) AND
customer_id IN (SELECT customer_id from customer_issue_details) and contact_address like
'%Nagar' order by customer_name ;
```

| CUSTOMER_ID | CUSTOMER_NAME | CONTACT_NO | CONTACT_ADDRESS         |
|-------------|---------------|------------|-------------------------|
| C00010      | RAGHAV SINGH  | 9675167890 | A/6 NEHRU JAWAHAR NAGAR |

8. Please follow instructions given below.

Write a query to display the movie id, movie name, release year, director name of movies acted by the lead actor who acted maximum number of movies. Display the records sorted in ascending order based on movie name.

```
select movie_id, movie_name , release_date ,director from movies_master where
lead_actor_name1 in(select lead_actor_name1 from(select
lead_actor_name1, count(movie_id) ct from movies_master group by lead_actor_name1)t where
t.ct>=all(select count(movie_id) from movies_master
group by lead_actor_name1))order by movie_name;
```

| MOVIE_ID | MOVIE_NAME | RELEASE_YEAR | DIRECTOR_NAME     |
|----------|------------|--------------|-------------------|
| M00004   | INCEPTION  | 2010         | CHRISTOPHER NOLAN |
| M00011   | TITANIC    | 1997         | JAMES CAMERON     |

9. Please follow instructions given below.

<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;
```

| CUSTOMER_NAME | MOVIE_COUNT |
|---------------|-------------|
|               |             |

|                   |   |
|-------------------|---|
| AGNESH            | 3 |
| AJAY GHOSH        | 0 |
| GEETHA REDDY      | 0 |
| NITIN             | 2 |
| RAGHAV SINGH      | 1 |
| RAJ SEKHANRAN     | 1 |
| RAJAN PILLAI      | 0 |
| RAJIB MITRA       | 4 |
| RIA NATRAJAN      | 0 |
| SHIV PRASAD       | 0 |
| T<br>RAMACHANDRAN | 8 |

10. Please follow instructions given below.

Write a query to display serial number, issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

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

For Example Assume the issue id is I00005 then the serial number is 05

Hint: Alias name for serial number is 'SERIAL\_NO'

**19 rows**

```
select substring(cid.issue_id,5,2) as
SERIAL_NO,cid.issue_id,cid.customer_id,cm.customer_name,mm.movie_id,mm.movie_name

from customer_issue_details cid join customer_master cm on cm.customer_id=cid.customer_id

join movies_master mm on cid.movie_id=mm.movie_id group by
SERIAL_NO,cid.customer_id,mm.movie_id

order by SERIAL_NO;
```

| SERIAL_NO | ISSUE_ID | CUSTOMER_ID | CUSTOMER_NAME  | MOVIE_ID | MOVIE_NAME         |
|-----------|----------|-------------|----------------|----------|--------------------|
| 01        | I00001   | C00001      | NITIN          | M00001   | DIE HARD           |
| 02        | I00002   | C00002      | AGNESH         | M00002   | THE DARK KNIGHT    |
| 03        | I00003   | C00002      | AGNESH         | M00002   | THE DARK KNIGHT    |
| 04        | I00004   | C00003      | T RAMACHANDRAN | M00003   | THE MATRIX         |
| 05        | I00005   | C00003      | T RAMACHANDRAN | M00004   | INCEPTION          |
| 06        | I00006   | C00003      | T RAMACHANDRAN | M00005   | OFFICE SPACE       |
| 07        | I00007   | C00004      | RAJIB MITRA    | M00006   | YOUNG FRANKENSTEIN |
| 08        | I00008   | C00010      | RAGHAV SINGH   | M00008   | CASABLANCA         |

|    |        |        |                   |        |                    |
|----|--------|--------|-------------------|--------|--------------------|
| 09 | I00009 | C00011 | RAJ SEKHANRAN     | M00010 | GONE WITH THE WIND |
| 10 | I00010 | C00004 | RAJIB MITRA       | M00007 | SHAUN OF THE DEAD  |
| 11 | I00011 | C00004 | RAJIB MITRA       | M00007 | SHAUN OF THE DEAD  |
| 12 | I00012 | C00001 | NITIN             | M00001 | DIE HARD           |
| 13 | I00013 | C00003 | T<br>RAMACHANDRAN | M00001 | DIE HARD           |
| 14 | I00014 | C00003 | T<br>RAMACHANDRAN | M00010 | GONE WITH THE WIND |
| 15 | I00015 | C00003 | T<br>RAMACHANDRAN | M00011 | TITANIC            |
| 16 | I00016 | C00003 | T<br>RAMACHANDRAN | M00011 | TITANIC            |
| 17 | I00017 | C00003 | T<br>RAMACHANDRAN | M00008 | CASABLANCA         |
| 18 | I00018 | C00002 | AGNESH            | M00010 | GONE WITH THE WIND |
| 19 | I00019 | C00004 | RAJIB MITRA       | M00001 | DIE HARD           |
|    |        |        |                   |        |                    |

11. Please follow instructions given below.

Write a query to display the issue id,issue date, customer id, customer name and contact number for videos that are issued in the year 2013. Display the records in descending order based on issue date of the video.

7 rows

```
select cid.issue_id,cid.issue_date,cid.customer_id,cm.customer_name,cm.contact_no
from customer_issue_details cid join customer_master cm on cid.customer_id=cm.customer_id
where year(issue_date)=2013 group by issue_id,issue_date,customer_id order by
issue_date desc;
```

| ISSUE_ID | ISSUE_DATE | CUSTOMER_ID | CUSTOMER_NAME  | CONTACT_NO |
|----------|------------|-------------|----------------|------------|
| I00012   | 2013-11-28 | C00001      | NITIN          | 9830354218 |
| I00017   | 2013-04-15 | C00003      | T RAMACHANDRAN | 9831289761 |
| I00009   | 2013-03-16 | C00011      | RAJ SEKHANRAN  | 8423178906 |
| I00016   | 2013-03-05 | C00003      | T RAMACHANDRAN | 9831289761 |
| I00008   | 2013-03-02 | C00010      | RAGHAV SINGH   | 9675167890 |
| I00015   | 2013-02-03 | C00003      | T RAMACHANDRAN | 9831289761 |
| I00014   | 2013-01-02 | C00003      | T RAMACHANDRAN | 9831289761 |

12. Please follow instructions given below.

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <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;
```

| MOVIE_ID | MOVIE_NAME   | ACTORS                        |
|----------|--------------|-------------------------------|
| M00009   | THE NOTEBOOK | RYAN GOSLING & RACHEL MCADAMS |

13. Please follow instructions given below.

Write a query to display the director's name, movie name and lead\_actor\_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR\_NAME as alias name for director name column Display the records sorted in ascending order based on director\_name and then by movie\_name in descending order.

2 rows

```
SELECT upper(director_name) DIRECTOR_NAME,movie_name,lead_actor_name1 FROM
movies_master WHERE director_name in (SELECT director_name FROM movies_master GROUP BY
director_name HAVING count(movie_id)>1) order by director_name, movie_name desc;
```

| DIRECTOR_NAME     | MOVIE_NAME      | LEAD_ACTOR_NAME1 |
|-------------------|-----------------|------------------|
| CHRISTOPHER NOLAN | THE DARK KNIGHT | CHRISTIAN BALE   |

|                   |           |                   |
|-------------------|-----------|-------------------|
| CHRISTOPHER NOLAN | INCEPTION | LEONARDO DICAPRIO |
|-------------------|-----------|-------------------|

14. Please follow instructions given below.

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <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'
```

| NO_OF_CUSTOMERS |
|-----------------|
| 6               |

15. Please follow instructions given below.

Write a query to display the customer's name, contact number, library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT\_DETAILS as alias name for customer contact.

11 rows

```
select cm.customer_name,ifnull(cm.contact_no,cm.contact_add) as
CONTACT_DETAILS,lcd.card_id,lcd.description from customer_master cm
left join customer_card_details ccd on cm.customer_id=ccd.customer_id
left join library_card_master lcd on ccd.card_id=lcd.card_id group by
customer_name,description,CONTACT_DETAILS
order by customer_name;
```

| CUSTOMER_NAME     | CONTACT_DETAILS                | CARD_ID | DESCRIPTION      |
|-------------------|--------------------------------|---------|------------------|
| AGNESH            | 8923156781                     | CRD002  | GOLD CARD        |
| AJAY GHOSH        | 8763478901                     | NULL    | NULL             |
| GEETHA REDDY      | 8976167890                     | NULL    | NULL             |
| NITIN             | 9830354218                     | CRD001  | SILVER CARD      |
| RAGHAV SINGH      | 9675167890                     | NULL    | NULL             |
| RAJ SEKHANRAN     | 8423178906                     | NULL    | NULL             |
| RAJAN PILLAI      | A 1/66<br>KODAMBAKKAM          | NULL    | NULL             |
| RAJIB MITRA       | 9830356781                     | CRD003  | PLATINUM<br>CARD |
| RIA NATRAJAN      | 9856723190                     | NULL    | NULL             |
| SHIV PRASAD       | 2/2 PHASE II,<br>JAWAHAR NAGAR | CRD003  | PLATINUM<br>CARD |
| T<br>RAMACHANDRAN | 9831289761                     | CRD002  | GOLD CARD        |

16. Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by

customer name in 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;
```

| CUSTOMER_ID | CUSTOMER_NAME  | NO_OF_TIMES |
|-------------|----------------|-------------|
| C00003      | T RAMACHANDRAN | 2           |
| C00004      | RAJIB MITRA    | 2           |
| C00001      | NITIN          | 2           |
| C00002      | AGNESH         | 2           |

17. Please follow instructions given below.

Write a query to display customer id, customer name, contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.&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;

```

| CUSTOMER_ID | CUSTOMER_NAME  | CONTACT_ISD      | MOVIE_CATEGORY | NO_OF_MOVIES |
|-------------|----------------|------------------|----------------|--------------|
| C00002      | AGNESH         | +91-892-315-6781 | ACTION         | 2            |
| C00001      | NITIN          | +91-983-035-4218 | ACTION         | 2            |
| C00004      | RAJIB MITRA    | +91-983-035-6781 | COMEDY         | 3            |
| C00003      | T RAMACHANDRAN | +91-983-128-9761 | ACTION         | 3            |
| C00003      | T RAMACHANDRAN | +91-983-128-9761 | ROMANCE        | 4            |

18. Please follow instructions given below.

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

3 rows

```
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) >= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by cm.customer_name)
union all
(select cm.customer_id,cm.customer_name from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id
having count(cid.issue_id) <= all (select count(cid.issue_id) from customer_master cm
join customer_issue_details cid
on cm.customer_id=cid.customer_id group by cm.customer_id) order by cm.customer_name);
```

| CUSTOMER_ID | CUSTOMER_NAME  |
|-------------|----------------|
| C00003      | T RAMACHANDRAN |
| C00010      | RAGHAV SINGH   |
| C00011      | RAJ SEKHANRAN  |

19. Please follow instructions given below.

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO\_OF\_TIMES as alias name

Display the records in ascending order based on customer name.

1 row

```
select cm.customer_id,cm.customer_name,count(mm.movie_id) as NO_OF_TIMES from
customer_master cm

join customer_issue_details cid on cm.customer_id=cid.customer_id join

movies_master mm on cid.movie_id=mm.movie_id where mm.movie_category='comedy' group by
customer_id

order by customer_name>1;
```

| CUSTOMER_ID | CUSTOMER_NAME | NO_OF_TIMES |
|-------------|---------------|-------------|
| C00004      | RAJIB MITRA   | 3           |

20. Please follow instructions given below.

Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL\_COST. Display the records sorted in ascending order based on customer id

6 rows

```
select cid.customer_id,sum(mm.rental_cost) as TOTAL_COST from customer_issue_details cid

join movies_master mm

on cid.movie_id=mm.movie_id group by customer_id order by customer_id;
```

|   | customer_id | TOTAL_COST |
|---|-------------|------------|
| ▶ | C00001      | 200        |
|   | C00002      | 300        |
|   | C00003      | 1700       |
|   | C00004      | 400        |
|   | C00010      | 1000       |
|   | C00011      | 100        |

## Item Loan Database Queries

1. Please follow instructions given below.

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

3 rows

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

| Filter: |               |                |
|---------|---------------|----------------|
|         | item_category | Count_category |
| ▶       | furniture     | 15             |
|         | Crockery      | 4              |
|         | Stationary    | 3              |

2. Please follow instructions given below.

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

1 row

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

| Filter: |                 |
|---------|-----------------|
|         | No_of_Employees |
| ▶       | 2               |

3. Please follow instructions given below.

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

1 row

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

|   | employee_id | employee_name | designation | department |
|---|-------------|---------------|-------------|------------|
| ▶ | E00005      | Radica        | Manager     | HR         |
| * | NULL        | NULL          | NULL        | NULL       |

4. Please follow instructions given below.

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

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

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

1 row

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

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00004      | Zuben         |

5. Please follow instructions given below.

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

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

9 rows

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

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

order by eid.issue\_id;

6. Please follow instructions given below.

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

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

3 rows

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

```
order by employee_id;
```

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00004      | Zuben         |
|   | E00005      | Radica        |
|   | E00006      | John          |
| * | NULL        | NULL          |

7. Please follow instructions given below.

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

1 row

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

|   | No_of_Cards |
|---|-------------|
| ▶ | 3           |

8. Please follow instructions given below.

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

1 row

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

|   | Count_Stationary |
|---|------------------|
| ▶ | 3                |

9. Please follow instructions given below.

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

5 rows

```
select em.employee_id,em.employee_name,count(eid.item_id) as Count from employee_master em
join
employee_issue_details eid on em.employee_id=eid.employee_id group by em.employee_id having
count(eid.item_id)>=1 order by Count desc,employee_id asc;
```

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

10. Please follow instructions given below.

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

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

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

2 rows

```
select em.employee_id,em.employee_name from employee_master em join employee_issue_details eid
on em.employee_id=eid.employee_id join item_master im on eid.item_id=im.item_id
and im.item_valuation<=all (select im.item_valuation from employee_master em join
employee_issue_details eid
```

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

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00002      | Abhay         |
|   | E00003      | Anita         |

11. Please follow instructions given below.

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

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

Consider only employees who have been issued atleast 1 item.

5 rows

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

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

12. Please follow instructions given below.

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

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

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

5 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid
on em.employee_id=eid.employee_id where datediff(return_date,issue_date)>365 order by
employee_id;
```

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

13. Please follow instructions given below.

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

Display the records sorted in ascending order on employee id.

2 rows

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

```
join employee_issue_details eid on em.employee_id=eid.employee_id join item_master im
on eid.item_id=im.item_id where item_category='furniture' group by employee_id having
count(COUNT_ITEMS)>1 order by employee_id;
```

|   | employee_id | employee_name | COUNT_ITEMS |
|---|-------------|---------------|-------------|
| ▶ | E00001      | Ram           | 2           |
|   | E00003      | Anita         | 3           |

14. Please follow instructions given below.

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

2 rows

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

|   | gender | No_of_Employees |
|---|--------|-----------------|
| ▶ | F      | 2               |
|   | M      | 4               |

gender order by gender;

15. Please follow instructions given below.

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

3 rows

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

|  | employee_id | employee_name |
|--|-------------|---------------|
|  | E00002      | Abhay         |
|  | E00003      | Anita         |
|  | E00006      | John          |

order by employee\_id;

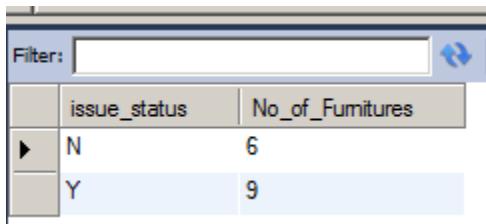
16. Please follow instructions given below.

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

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

2 rows

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



|   | issue_status | No_of_Furnitures |
|---|--------------|------------------|
| ▶ | N            | 6                |
|   | Y            | 9                |

17. Please follow instructions given below.

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

16 rows

```
select item_category,item_make,item_description,count(item_id) as No_of_Items from
item_master im group by item_category,item_make,item_description order by
item_category,item_make,item_description;
```

|   | item_category | item_make | item_description | No_of_Items |
|---|---------------|-----------|------------------|-------------|
| ▶ | Crockery      | Bonechina | Dining Set       | 1           |
|   | Crockery      | Bonechina | Tea Set          | 1           |
|   | Crockery      | Glass     | Dining Set       | 1           |
|   | Crockery      | Glass     | Tea Set          | 1           |
|   | furniture     | Steel     | Cupboard         | 2           |
|   | furniture     | Steel     | Side Table       | 1           |
|   | furniture     | Steel     | Single Bed       | 2           |
|   | furniture     | Steel     | Tea Table        | 2           |
|   | furniture     | Wooden    | Dining Chair     | 1           |
|   | furniture     | Wooden    | Dining Table     | 1           |
|   | furniture     | Wooden    | Double Bed       | 2           |
|   | furniture     | Wooden    | Side Table       | 1           |
|   | furniture     | Wooden    | Sofa             | 1           |
|   | furniture     | Wooden    | Tea Table        | 2           |
|   | Stationary    | Plastic   | Pen              | 2           |
|   | Stationary    | Wooden    | Pencil           | 1           |

18. Please follow instructions given below.

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

1 row

```
select em.employee_id,em.employee_name,im.item_id,im.item_description from employee_master em
join
employee_issue_details eid on em.employee_id=eid.employee_id join item_master im on
eid.item_id=im.item_id where year(eid.issue_date)=2013 and month(eid.issue_date)=01 order by
em.employee_id,im.item_id;
```

|   | employee_id | employee_name | item_id | item_description |
|---|-------------|---------------|---------|------------------|
| ▶ | E00002      | Abhay         | I00005  | Side Table       |

19. Please follow instructions given below.

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

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

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

1 row

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

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

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

order by em.employee_id;
```

|   | employee_id | employee_name | COUNT_CATEGORY |
|---|-------------|---------------|----------------|
| ▶ | E00004      | Zuben         | 2              |

20. Please follow instructions given below.

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

14 rows

```
select item_id,item_description from item_master where item_id not in (select item_id
from employee_issue_details) order by item_id;
```

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

21. Please follow instructions given below.

Write a query to display the employee id, employee name and total valuation for the employees who 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;
```

|   | employee_id | employee_name | TOTAL_VALUATION |
|---|-------------|---------------|-----------------|
| ▶ | E00002      | Abhay         | 1500.00         |

22. Please follow instructions given below.

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

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

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

```

SELECT ecd.employee_id,employee_name,
card_issue_date, if(lcd.duration_in_years=0, 'NO-VALIDITY DATE', date_add(ec.card_issue_date, interval
duration_in_years year)) as CARD_VALIDITY_DATE

FROM employee_master em INNER JOIN
employee_card_details ecd
ON em.employee_id=ecd.employee_id
INNER JOIN loan_card_master lcd
ON ecd.loan_id=lcd.loan_id
order by employee_name, CARD_VALID_DATE;

```

Filter:  Export: Autosize:

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

23. Please follow instructions given below.

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

3 rows

```
select distinct em.employee_id,em.employee_name from employee_master em join
employee_issue_details eid on
em.employee_id=eid.employee_id where em.employee_id not in
(select employee_id from employee_issue_details where year(issue_date)=2013)
order by employee_id;
```

Filter:  Export:

|   | employee_id | employee_name |
|---|-------------|---------------|
| ▶ | E00001      | Ram           |
|   | E00003      | Anita         |
|   | E00006      | John          |

24. Please follow instructions given below.

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

9 rows

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

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

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

<br>[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issued item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name 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);;
```

Filter:  Export: Autosize:

| employee_id | employee_name | TOTAL_VALUATION |
|-------------|---------------|-----------------|
| E00004      | Zuben         | 25500.00        |

```
SELECT cus.profile_id, cus.first_name, cus.address, mintkt.No_of_Tickets
FROM air_passenger_profile cus, (SELECT MIN(s.Tot_No_of_Tickets) AS No_of_Tickets
FROM (SELECT profile_id, COUNT(ticket_id) AS Tot_No_of_Tickets FROM air_ticket_info
GROUP BY profile_id) s) mintkt,
(SELECT profile_id, COUNT(ticket_id) AS Tot_No_of_Tickets FROM air_ticket_info GROUP BY
profile_id) totkt
WHERE mintkt.No_of_Tickets = totkt.Tot_No_of_Tickets
AND cus.profile_id = totkt.profile_id
order by cus.first_name;
```