Bank Management Schema

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

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

2.Please follow instructions given below.
Write a query to display the number of customer’s from Delhi.
Give the count an alias name of Cust_Count.

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

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

```
ANS:   SELECT  am.customer_number, firstname, account_number
FROM customer_master cm INNER JOIN account_master am
```

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

****

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

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

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

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

6. Please follow instructions given below.
Write a query to display the number of customers who have registration but no account in
the bank.Give the alias name as Count_Customer for number of customers.

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

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

```
ANS:   SELECT td.account_number, opening_balance+sum(transaction_amount) Deposit_Amount
FROM account_masteram INNER JOIN transaction_details td
ON am.account_number=td.account_number
WHERE transaction_type='deposit'
```

```
GROUP BY  account_number
ORDER BY account_number
```

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

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

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

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

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

```
ANS:   SELECT ld.customer_number, firstname, lastname
FROM customer_master cm INNER JOIN loan_detailsld
ON cm.customer_number=ld.customer_number
GROUP BY customer_number
HAVING count(branch_id)>1
ORDER BY customer_number
```

11.Please follow instructions given below.Write a query to display the customer’s number, customer’s firstname, customer’s city and branch city where the city of the customer and city of the branch is different.Display the records sorted in ascending order based on customer number.

```
ANS:   selectcustomer_master.customer_number, firstname, customer_city,
branch_city
fromaccount_master inner join customer_master on
account_master.customer_number = customer_master.customer_number
inner join branch_master on account_master.branch_id = branch_master.branch_id
wherecustomer_city != branch_city  order by customer_master.customer_number;
```

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

ANS: SELECT count(ld.customer_number) Count
FROM customer_master cm INNER JOIN loan_detailsld
ON cm.customer_number=ld.customer_number
WHERE cm.customer_number NOT IN (SELECTcustomer_number FROM account_master)

****

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

ANS: SELECT td.account_number
FROM account_masteram INNER JOIN transaction_details td
ON am.account_number=td.account_number

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

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

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

****

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

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

****

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

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

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

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

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

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

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

For example the customer C00012 took a loan of 100000 from bank branch with id B00009 and C00012 Took a loan of 500000 from bank branch with id B00010. So total loan

amount for
customer C00012 is 600000. C00013 took a loan of 100000 from bank branch
B00009 and 200000
from bank branch B00011. So total loan taken is 300000. So loan taken by
C00012 is more
then C00013.

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

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

```
ANS:  SELECT ld.customer_number, firstname,branch_id, loan_amount
FROM customer_master  cm INNER JOIN loan_detailsld
ON cm.customer_number=ld.customer_number order by cm.customer_number,
branch_id,loan_amount
```

21. Please follow instructions given below.
Write a query to display city name and count of branches in that city. Give the
count of

branches an alias name of Count_Branch.Display the records sorted in ascending order based
on city name.

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

22.Please follow instructions given below.
Write a query to display account id, customer’s firstname, customer’slastname for the
customer’s whose account is Active.Display the records sorted in ascending order based on
account id /account number.

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

****

23.Please follow instructions given below. Write a query to display customer’s number,
first name and middle name. For the customers who don’t have middle name, display their
last name as middle name. Give the alias name as Middle_Name.

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

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

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

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

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

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

ITEM LOAN

1.Please follow instructions given below.
<br>Write a query to display category and number of items in that category. Give the count an alias name of Count_category. Display the details on the sorted order of count in descending order.

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

2. Please follow instructions given below.
<br>Write a query to display the number of employees in HR department. Give the alias name as No_of_Employees.

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

3.Please follow instructions given below.

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

```
ANS:  SELECT employee_id, employee_name, designation, department
FROM employee_master WHERE employee_id
NOT IN ( SELECTemployee_id FROM employee_issue_details)
order by employee_id;
```

4. Please follow instructions given below.
<br><br>Write a query to display the employee id, employee name who was issued an item of highest valuation.<br> In case of multiple records, display the records sorted in ascending order based on employee id.[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

```
ANS:  SELECT eid.employee_id, employee_name
FROM employee_masterem INNER JOIN employee_issue_detailseid
ON em.employee_id=eid.employee_id
INNER JOIN item_masterim
ON eid.item_id=im.item_id
WHERE item_valuation=(SELECT max(item_valuation)
FROM employee_issue_detailseid INNER JOIN item_masterim
ON eid.item_id=im.item_id) order by eid.employee_id;
```

5. Please follow instructions given below.
<br>Write a query to display issue_id, employee_id, employee_name.
Display the records sorted in ascending order based on issue id.

```
ANS:  SELECT issue_id, eid.employee_id, employee_name
FROM employee_masterem INNER JOIN  employee_issue_detailseid
```

```
ON em.employee_id=eid.employee_id order by issue_id;
```

6. Please follow instructions given below.
<br>Write a query to display employee id, employee name who don’t have loan cards.
Display the records sorted in ascending order based on employee id.

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

7.Please follow instructions given below.
<br>Write a query to count the number of cards issued to an employee “Ram”.
Give the count an alias name as No_of_Cards.

```
ANS:  SELECT count(loan_id)  AS No_of_Cards
FROM  employee_card_details c
JOIN  employee_master e
ON    c.employee_id = e.employee_id
WHERE e.employee_name= 'Ram'
GROUP BY c.employee_id
```

8.Please follow instructions given below.
<br>Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count_stationary.

```
ANS:  SELECT count(employee_id)  Count_stationary
FROM  employee_card_detailsecd INNER JOIN  loan_card_masterlcd
ON ecd.loan_id=lcd.loan_id
WHERE  loan_type='stationary'
```

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

```
ANS:  SELECT eid.employee_id, employee_name, count(item_id) Count
FROM  employee_masterem INNER JOIN employee_issue_detailseid
ON em.employee_id=eid.employee_id
GROUP BY employee_id order by count DESC, eid.employee_id;
```

10. Please follow instructions given below.
<br>Write a query to display the employee id, employee name who was issued an item of minimum valuation.
<br>In case of multiple records, display them sorted in ascending order based on employee id.
[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

```
ANS:  SELECT eid.employee_id, employee_name
FROM employee_masterem INNER JOIN employee_issue_detailseid
ON em.employee_id=eid.employee_id
INNER JOIN item_masterim
ON eid.item_id=im.item_id
WHERE item_valuation=(SELECT min(item_valuation)
FROM employee_issue_detailseid INNER JOIN item_masterim
ON eid.item_id=im.item_id)
order by eid.employee_id;
```

11. Please follow instructions given below.
<br>Write a query to display the employee id, employee name and total

valuation of the product issued to each employee. Give the alias name as TOTAL_VALUATION.

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

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

```
ANS:  SELECT em.employee_id, employee_name, sum(item_valuation)
TOTAL_VALUATION
FROM employee_masteremINNER  JOINemployee_issue_detailseid
ON em.employee_id=eid.employee_id
INNER JOIN item_masterim
ON eid.item_id=im.item_id
GROUP BY eid.employee_id
ORDER BY eid.employee_id;
```

12. Please follow instructions given below.

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

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

```
ANS:  SELECT DISTINCT eid.employee_id,employee_name
FROM  employee_issue_detailseid JOIN employee_masterem
ON eid.employee_id=em.employee_id
WHERE datediff(return_date,issue_date) >  365
order by eid.employee_id;
```

13. Please follow instructions given below.

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

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

```
ANS:  SELECT eid.employee_id, employee_name, count(eid.item_id)
COUNT_ITEMS
FROM employee_issue_detailseid JOIN item_masterim
ON eid.item_id=im.item_id
JOIN employee_masterem
ON eid.employee_id=em.employee_id
WHERE item_category='furniture'
GROUP BY employee_id
HAVING COUNT_ITEMS>1
order by eid.employee_id;
```

14. Please follow instructions given below.
<br>Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No_of_Employees. Display the records sorted in ascending order based on gender.

```
ANS:  SELECT gender , count(gender) No_of_Employees
FROM employee_master
GROUP BY gender ORDER BY gender
```

15. Please follow instructions given below.
<br>Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

```
ANS:  SELECT employee_id, employee_name
FROM employee_master
WHERE extract(year from date_of_joining)>2005
order by employee_id;
```

16. Please follow instructions given below.
<br>Write a query to get the number of items of the furniture category issued

and not issued. The query should display issue status and the number of furniture as No_of_Furnitures.
Display the records sorted in ascending order based on issue_status.

```
ANS:  SELECT  issue_status, count(item_id) No_of_Furnitures
FROM item_master
WHERE  item_category='Furniture'
GROUP BY issue_status
ORDER BY issue_status
```

17. Please follow instructions given below.
<br>Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No_of_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.

```
ANS:  SELECT item_category, item_make,item_description, count(item_description) No_of_Items FROM item_master
GROUP BY item_category, item_make, item_description ORDER BY item_category, item_make,item_description;
```

18. Please follow instructions given below.
<br>Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

```
ANS:  SELECT  eid.employee_id, employee_name, eid.item_id, item_description
FROM employee_issue_detailseid JOIN employee_masterem
ON eid.employee_id=em.employee_id
JOIN item_masterim
ON eid.item_id=im.item_id
```

```
WHERE extract(month from issue_date)=1 and extract(year from issue_date)=2013
ORDER BY eid.employee_id, eid.item_id;
```

19. Please follow instructions given below.
<br>Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.Give the alias name for category count as COUNT_CATEGORY.Display the records sorted in ascending order based on employee id.

```
ANS:  SELECT em.employee_id,employee_name, count(distinct item_category)
COUNT_CATEGORY
 FROM employee_issue_detailseid JOIN item_masterim
ON eid.item_id=im.item_id
JOIN employee_masterem
ON eid.employee_id=em.employee_id
GROUP BY employee_id
HAVING COUNT_CATEGORY>=2
ORDER BY employee_id;
```

20. Please follow instructions given below.
<br>Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.

```
ANS:  SELECT item_id, item_description
FROM item_master
WHERE item_id
NOT IN ( SELECTitem_id from employee_issue_details)
ORDER BY item_id;
```

21. Please follow instructions given below.

<br>Write a query to display the employee id, employee name and&nbsp;&nbsp; totalvaluation&nbsp;for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL_VALUATION.
[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed.]

```
ANS: select employee_issue_details.employee_id,employee_master.employee_name,sum(item_master.item_valuation)as TOTAL_VALUATION from
employee_issue_details inner join item_master
onitem_master.item_id = employee_issue_details.item_id
inner join employee_master
onemployee_master.employee_id=employee_issue_details.employee_id
group by employee_issue_details.employee_id
order by TOTAL_VALUATION asc limit 1;
```

22. Please follow instructions given below.
Write a query to display the employee id, employee name, card issue date and card valid date.Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD_VALID_DATE.
[Hint: Validity in years for the loan card is given in loan_card_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'.]

```
ANS:  SELECT ecd.employee_id,employee_name,
card_issue_date, CASE duration_in_years
WHEN 0 THEN 'No Validity Date'
ELSE DATE_ADD(card_issue_date, INTERVAL duration_in_years YEAR)
END CARD_VALID_DATE
FROM employee_masterem INNER JOIN
```

```
employee_card_detailsecd
ON em.employee_id=ecd.employee_id
INNER JOIN loan_card_masterlcd
ON ecd.loan_id=lcd.loan_id
order by employee_name,  CARD_VALID_DATE;
```

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

```
ANS:  SELECT DISTINCT a.employee_id,b.employee_name FROM employee_issu
e_details a JOIN
employee_master b ON a.employee_id=b.employee_id WHERE a.employee_id
NOT IN(SELECT employee_id
FROM employee_issue_details WHERE (EXTRACT(YEAR FROM issue_date)=2013
))
ORDER BY a.employee_id;
```

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

```
ANS:  SELECT issue_id, eid.employee_id, employee_name, im.item_id,
item_description,issue_date
FROM employee_issue_detailseid INNER JOIN employee_masterem
ON eid.employee_id=em.employee_id
INNER JOIN item_masterim
ON eid.item_id=im.item_id
ORDER BY issue_datedesc, issue_idasc;
```

25. Please follow instructions given below.
<br>Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product.&nbsp; Give the alias name for total valuation as TOTAL_VALUATION.&nbsp;
<br>[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display.]

ANS:

```
select employee_issue_details.employee_id,employee_master.employee_name,sum(item_master.item_valuation)as TOTAL_VALUATION from
employee_issue_details inner join item_master
onitem_master.item_id = employee_issue_details.item_id
inner join employee_master
onemployee_master.employee_id=employee_issue_details.employee_id
group by employee_issue_details.employee_id
order by TOTAL_VALUATION desc limit 1;
```

VIDEO LIBRARY

1.Please follow instructions given below.
Write a query to display movie names and number of times that movie is issued to customers. Incase movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order).
The Alias name for the number of movies issued is ISSUE_COUNT.
ANS: SELECT movie_name, count(cid.movie_id) ISSUE_COUNT FROM movies_master mm LEFT JOIN customer_issue_detailscid ON mm.movie_id=cid.movie_id GROUP BY movie_name ORDER BY Issue_count DESC, movie_name;

2.Please follow instructions given below.
Write a query to display id,name,age,contact no of customers whose age is greater than 25 and and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.
ANS: select customer_id,customer_name, age, coalesce(concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',substring(contact_no,7)),'N/A') CONTACT_ISD from customer_master where age>25 and substring(date_of_registration,1,4)=2012 order by age asc, customer_name;

3. Please follow instructions given below.
<br>Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.
<br/>Hint: Use NO_OF_MOVIES as alias name for number of movies.
ANS: SELECT movie_category, count(movie_id) NO_OF_MOVIES FROM movies_master GROUP BY movie_category order by no_of_moviesdesc, movie_categoryasc;

4.Please follow instructions given below.
<br>Write a query to display the number of customers having card with description “Gold card”. <br/>Hint: Use CUSTOMER_COUNT as alias name for number of customers
ANS: SELECT count(customer_id) CUSTOMER_COUNT FROM library_card_master lcm INNER JOIN customer_card_detailsccd ON lcm.card_id=ccd.card_id WHERE description='Gold Card';

5.Please follow instructions given below.<br>
Write a query to display the customer id, customer name, year of registration,library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.
<br> Use REGISTERED_YEAR as alias name for year of registration.
ANS: SELECT c.customer_id, c.customer_name, extract(year from c.date_of_registration) REGISTERED_YEAR,cd.card_id,cd.issue_date FROM customer_master c join customer_card_details cd on c.customer_id=cd.customer_id ORDER BY c.customer_name DESC;

6. Please follow instructions given below.<br>
Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.
<br>
Display the records sorted in ascending order based on customer name.
ANS: SELECT issue_id ,cid.customer_id, customer_name FROM customer_issue_detailscid INNER JOIN customer_master cm ON cid.customer_id=cm.customer_id WHERE actual_date_of_return>return_date and customer_name like 'R%' order by customer_name;

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

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

8. Please follow instructions given below.
Write a query to display the customer id, customer name,contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.
Display customer name in upper case. Hint: Use CUSTOMER_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.
ANS: SELECT customer_id , upper(customer_name) CUSTOMER_NAME,contact_no,contact_address FROM customer_master WHERE customer_id NOT IN (select customer_id from customer_card_details) AND customer_id IN (SELECT

customer_id from customer_issue_details) and contact_address like '%Nagar' order by customer_name ;

9. Please follow instructions given below.
Write a query to display the movie id, movie name,releaseyear,director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.
ANS: select movie_id,movie_name,release_year,director_name from movies_master where lead_actor_name1 in(select lead_actor_name1 from (select lead_actor_name1,count(movie_id)ct from movies_master group by lead_actor_name1)t where t.ct>=all(select count(movie_id) from movies_master group by lead_actor_name1)) order by movie_name;

10. Please follow instructions given below.
<br>
Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0. <br>Hint: Use MOVIE_COUNT as alias name for number of movies issued.
ANS: SELECT customer_name, count(movie_id) MOVIE_COUNT FROM customer_master cm LEFT OUTER JOIN customer_issue_detailscid ON cm.customer_id=cid.customer_id GROUP BY customer_name ORDER BY customer_name;
11. Please follow instructions given below.
<br>
Write a query to display serial number,issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.
<br/>Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.

<br/>For Example Assume the issue id is I00005 then the serial number is 05 <br/>Hint: Alias name for serial number is 'SERIAL_NO'<br/><br/><br>

ANS: SELECT substring(issue_id,5,2) SERIAL_NO,issue_id, cid.customer_id,customer_name, mm.movie_id, movie_name FROM customer_issue_detailscid INNER JOIN customer_master cm ON cid.customer_id=cm.customer_id INNER JOIN movies_master mm ON cid.movie_id=mm.movie_id order by SERIAL_NO ASC;

12. Please follow instructions given below.

<br>

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

ANS: SELECT issue_id, issue_date, cid.customer_id, customer_name,contact_no FROM customer_issue_detailscid INNER JOIN customer_master cm ON cid.customer_id=cm.customer_id where extract(year from issue_date)=2013 order by issue_date DESC

13. Please follow instructions given below.

<br>

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <br> Actors Name to be displayed in the below format.LEAD_ACTOR_ONE space ambersant space LEAD_ACTOR_TWO .

Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom& Maria"Hint:Use ACTORS as alias name for actors name. <br> Display the records in ascending order based on movie name.

ANS: SELECT movie_id, movie_name,concat(lead_actor_name1,' & ',lead_actor_name2) ACTORS FROM movies_master WHERE movie_id NOT IN (SELECT movie_id from customer_issue_details) order by movie_nameasc;

14.Please follow instructions given below.
Write a query to display the director's name, movie name and lead_actor_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR_NAME as alias name for director name column Display the records sorted in ascending order based on director_name and then by movie_name in descending order.
ANS: SELECT upper(director_name) DIRECTOR_NAME,movie_name,lead_actor_name1 FROM movies_master WHERE director_name in (SELECT director_name FROM movies_master GROUP BY director_name HAVING count(movie_id)>1) order by director_name, movie_namedesc;

15.Please follow instructions given below.
<br>
Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <br>Hint:Use NO_OF_CUSTOMERS as alias name for number of customers.
ANS: SELECT count(customer_id) NO_OF_CUSTOMERS FROM customer_master WHERE extract(year from date_of_registration)=2012 and contact_no is not null

16.Please follow instructions given below.
<br>
Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT_DETAILS as alias name for customer contact.
ANS: SELECT c.customer_name,coalesce(c.contact_no,c.contact_address) CONTACT_DETAILS,cd.card_id,cd.description FROM customer_master c left join

customer_card_detailsccd on ccd.customer_id=c.customer_id left join library_card_master cd on cd.card_id=ccd.card_id order by customer_name;

17.Please follow instructions given below.

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decendingorder For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john.

Hint: Use NO_OF_TIMES as alias name for number of times

ANS: SELECT cid.customer_id, customer_name,count(movie_id) NO_OF_TIMES FROM customer_master cm INNER JOIN customer_issue_detailscid ON cm.customer_id=cid.customer_id group by customer_id, customer_name,movie_id having count(movie_id)>1 order by customer_namedesc;

18. Please follow instructions given below.

Write a query to display customer id, customer name,contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category.

Example: Display contact number as "+91-876-456-2345" format.&nbsp;

<br>Hint:Use NO_OF_MOVIES as alias name for number of movies column.

<br>Hint:Use CONTACT_ISD as alias name for contact number.

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

ANS: SELECT cid.customer_id,customer_name,concat('+91-',substring(contact_no,1,3),'-',substring(contact_no,4,3),'-',substring(contact_no,7)) CONTACT_ISD, movie_category,count(movie_category) NO_OF_MOVIES FROM customer_issue_detailscid INNER JOIN movies_master mm ON cid.movie_id=mm.movie_id INNER JOIN customer_master cm ON cm.customer_id=cid.customer_id group by customer_id,customer_name,CONTACT_ISD,movie_category having

```
count(movie_category)>1 order by customer_name, movie_category;
```

19. Please follow instructions given below.

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

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

```
ANS:  (selectcid.customer_id , customer_name FROM  customer_master cm
INNER JOIN customer_issue_detailscid
ON cm.customer_id=cid.customer_id
group by customer_id , customer_name
having count(movie_id)>=ALL(select  count(movie_id)
FROM customer_issue_details
group by customer_id) order by customer_name)
UNION ALL
(selectcid.customer_id , customer_name FROM  customer_master cm INNER
JOIN customer_issue_detailscid
ON cm.customer_id=cid.customer_id
group by customer_id , customer_name
having count(movie_id)<=ALL(select  count(movie_id)
FROM customer_issue_details
```

group by customer_id) order by customer_name) ;

20. Please follow instructions given below.
Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.
Hint: Use NO_OF_TIMES as alias name
Display the records in ascending order based on customer name.
ANS: SELECT cid.customer_id,customer_name,count(cid.movie_id) NO_OF_TIMES FROM customer_master cm INNER JOIN customer_issue_detailscid ON cm.customer_id=cid.customer_id INNER JOIN movies_master mm ON cid.movie_id=mm.movie_id WHERE movie_category='Comedy' GROUP BY customer_name HAVING count(cid.customer_id)>1 order by customer_name;
21. Please follow instructions given below.
Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL_COST.
Display the records sorted in ascending order based on customer id
ANS: SELECT cid.customer_id, sum(rental_cost) TOTAL_COST FROM customer_issue_detailscid INNER JOIN movies_master mm ON cid.movie_id=mm.movie_id GROUP BY customer_id order by customer_id;

FLIGHT MANAGEMENT

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

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

2. Display the name of the youngest passenger.

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

****

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

```
select passname,passdob,
round(datediff(current_date,passdob)/365) as age from passenger;
```

4. Display the number of flights leaving Kolkata.

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

****

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

```
select a.flightsource from flight a
group by a.flightsource having count(a.flightsource)=
(select count(b.flightdest) from flight b
where a.flightsource=b.flightdest);
```

6.
Display the name of the city which has flight source but no destination.

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

****

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

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

8. Display the number of passenger in each flight.
Use column alias “PassCount”.

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

9. Display the name and date of birth of passengers who are senior citizen

(age>=60).

```
select passname,passdob from passenger
where round((datediff(current_date,passdob)/365))>=60;
```

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

```
select bookingid from booking_details group by bookingid having count(passid)
=
(select count(passid) from booking_details group by bookingid
order by count(passid) desc limit 1);
```

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

```
select a.bookingid,sum(b.ticketcost) as Total_Fare from booking_details a,flight
b,
booking c
where c.flightid=b.flightid and a.bookingid=c.bookingid group by a.bookingid;
```

12. Display the booking id (ticket) and the total cost for the booking.
Use column alias “Total Fare”.
Consider giving a rebate of 50% to senior citizen (age>=60).

```
select a.bookingid,sum(
case
when datediff(current_date,d.passdob)/365>=60 then b.ticketcost*0.5
else b.ticketcost end) as Total_fare
 from booking_details a,flight b,booking c,passenger d
where a.bookingid=c.bookingid and b.flightid=c.flightid and a.passid=d.passid
group by a.bookingid;
```

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

```
select flightdest from flight
group by flightdest
having count(flightid)=(select count(flightid)
from flight group by flightdest
order by count(flightid) desc limit 1);
```

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

```
select passname from passenger
where passid in(select passid from booking_details
group by passid having count(bookingid)>1);
```

15. Display flightid with no of booking.

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

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

1.

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

17. Display flights having the same source and destination.

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

****

18. Display the record in the following format.
 Column alias “Booking Summary”
#Hints:“ Ticket No:1 Flight id: 1 Total Passengers :3 Total Fare:6000”
#“Ticket No:2 Flight id: 3 Total Passengers :1 Total Fare :2500”

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

****

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

```
select flightid,flightdate,flightdate + interval 4 hour as flight_new_date from flight
where flightid=2;
```

20. Display passenger name , date of birth sorted by the month of birth .
(Jan ? Dec).

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

Airline FilghtManagement Latest

1.Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id,From_location,To_Location,Mont

h Name as “Month_Name” and average price as “Average_Price”
Display the records sorted in ascending order based on flight id and then by Month Name.

```
SELECT  f.flight_id,
 f.from_location,
 f.to_location,monthname(fd.flight_departure_date),
 avg(fd.price) as Average_Price
FROM  air_flight_detailsfd, air_flight f
WHERE  f.flight_id=fd.flight_id  and f.airline_name="ABC "
GROUP BY f.flight_id, fd.flight_departure_date,f.from_location, f.to_location
order by f.flight_id, fp.Month_Name;
```

2.Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile_id, customer’s first_name, Address and Number of tickets booked as “No_of_Tickets”
Display the records sorted in ascending order based on customer's first name.

```
SELECT p.profile_id, p.first_name, p.address, count(t.ticket_id) as  No_of_Tickets
FROM air_passenger_profile p, air_flight f, air_ticket_info t
WHERE p.profile_id=t.profile_id
 AND t.flight_id=f.flight_id
 AND f.airline_name='abc'
GROUP BY p.profile_id
HAVING count(t.ticket_id)
IN (SELECT min(c) FROM
 (SELECT count(t.ticket_id) c     FROM air_flight f, air_ticket_info t
    WHERE t.flight_id=f.flight_id  ANDf.airline_name='abc'
 GROUP BY t.profile_id )a);
```

3.Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as “Month_Name” and number of flight services as “No_of_Services”.
Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight.
The records should be displayed in ascending order based on From_Location and then by To_Location and then by month name

```
select f.from_location,f.to_location,monthname(t.flight_departure_date),
count(t.flight_departure_date) as  No_of_Services  from  air_flight f
join  air_flight_details t on t.flight_id=f.flight_id

group by f.from_location,f.to_location
order by  f.from_location,f.to_location,monthname(t.flight_departure_date)  asc;
```

4.Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The Query should display profile_id, customer’s first_name, Address and Number of tickets booked as “No_of_Tickets”
Display the records in ascending order based on customer's first name.

```
SELECT p.profile_id, p.first_name, p.address, count(t.ticket_id) as  No_of_Tickets
FROM air_passenger_profile p, air_flight f, air_ticket_info t
WHERE p.profile_id=t.profile_id
AND  t.flight_id=f.flight_id
AND  f.airline_name='abc'
GROUP BY p.profile_id
```

```
HAVING count(t.ticket_id)
IN (SELECT max(c) FROM
 (SELECT count(t.ticket_id) c     FROM air_flight f, air_ticket_info t
     WHERE t.flight_id=f.flight_id  ANDf.airline_name='abc'  GROUP BY t.profile_id
)a);
```

5.Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as “No_of_Tickets”. Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

```
select p.profile_id,p.first_name,p.last_name,t.flight_id,t.flight_departure_date,
count(t.ticket_id)  as No_of_Tickets
fromair_passenger_profile  p  join  air_ticket_info  t
onp.profile_id=t.profile_id
joinair_flight f
ont.flight_id=f.flight_id
wheref.from_location="chennai"
andf.to_location="hyderabad"
group by p.profile_id,t.flight_id,t.flight_departure_date
order by p.profile_id,t.flight_id,t.flight_departure_date;
```

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

```
SELECT fd.flight_id,af.FROM_LOCATION,af.TO_LOCATION, fd.price FROM
air_flight_detailsfd join air_flightaf on af.FLIGHT_ID=fd.FLIGHT_ID where
```

```
substring(fd.flight_departure_date,6,2)='04' order by fd.flight_id, af.FROM_LOCATION;
```

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

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

```
SELECT f.flight_id,
f.from_location,
f.to_location,
AVG(fd.Price) AS Price
FROM  air_flight f JOIN
air_flight_detailsfd
ON      f.flight_id = fd.flight_id
GROUP BY f.flight_id order by f.flight_id, f.from_location, f.to_location;
```

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

Give an alias to the name as customer_name.

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

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

```
SELECT DISTINCT c.profile_id,
CONCAT(c.first_name, ',',c.last_name) AS customer_name,
c.address
FROM  air_passenger_profile c
JOIN    air_ticket_info t
ON      c.profile_id = t.profile_id
JOIN   air_flight f
ON     f.flight_id = t.flight_id
AND  f.from_location = 'Chennai'
AND  f.to_location = 'Hyderabad'
order by c.profile_id;
```

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

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

10.Write a query to display the total number of tickets as "No_of_Tickets" booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets.
Display only the flights in which atleast 1 ticket is booked.
Display the records sorted in ascending order based on flight id.

```
SELECT f.flight_id,
f.from_location,
f.to_location,
COUNT(t.ticket_id) AS No_of_Tickets
FROM air_ticket_infot JOIN
air_flight f
ON      f.flight_id = t.flight_id
where AIRLINE_NAME = 'ABC ‘
GROUP by f.flight_id
ORDER by f.flight_id;
```

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

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

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

```
SELECT flight_id,
COUNT(flight_departure_date) AS No_of_Services,
SUM(price) AS Total_Price
FROM  air_flight_details
GROUP BY flight_id order by total_price DESC, flight_id DESC;
```

12.Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight_id, flight_departure_date and the number of passengers as “No_of_Passengers” in the same order. Display the records sorted in ascending order based on flight id and then by flight departure date.

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

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

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

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

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

```
SELECT distinct ti.PROFILE_ID,pi.first_name,pi.mobile_number, pi.email_id FROM
air_ticket_infoti join air_passenger_profile pi on pi.profile_id=ti.profile_id where
flight_id in (SELECT FLIGHT_ID FROM air_flight where FROM_LOCATION
='HYDERABAD' and to_location ='CHENNAI') order by ti.profile_id
```

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

```
SELECT DISTINCT p.profile_id,
p.first_name,
p.last_name,
t.flight_id,
t.flight_departure_date,
f.departure_time,
f.arrival_time,
ADDTIME(f.departure_time,'01:00:00') AS Delayed_Departure_Time,
ADDTIME(f.arrival_time,'01:00:00') AS Delayed_Arrival_Time
FROM air_passenger_profile p
JOIN air_ticket_info t
ON p.profile_id = t.profile_id
AND t.flight_departure_date = '2013-05-06'
JOIN air_flight f
ON t.flight_id = f.flight_id
AND f.from_location = 'Chennai'
AND f.to_location = 'Hyderabad'
order by p.profile_id;
```

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

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

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

```
SELECT cus.profile_id,
cus.First_name,
SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')-
INSTR(cus.address,',')-1)
            AS Base_Location,
s.No_of_Tickets
FROM  air_passenger_profilecus JOIN
(
SELECT profile_id,
COUNT(ticket_id) AS No_of_Tickets
FROM  air_ticket_info
GROUP BY profile_id
)s
ON s.profile_id = cus.profile_id
AND  SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')-
INSTR(cus.address,',')-1) = 'Kochi'
order by first_name;
```

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

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

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

```
selectf.flight_id,
f.from_location,
f.to_location,
count(at.flight_departure_date) as Number_of_Services
fromair_flight f join air_ticket_info  at
onat.flight_id=f.flight_id  where month(at.flight_departure_date)=05
group by f.flight_id  order by f.flight_id
```

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

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

```
SELECT PROFILE_ID,LAST_NAME,MOBILE_NUMBER,EMAIL_ID FROM air_passenger
_profile where address like '%CHENNAI%' ORDER BY PROFILE_ID;
```

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

```
SELECT count(flight_id) FLIGHT_COUNT FROM air_flight where FROM_LOCATION='
CHENNAI' and departure_time between '06:00:00' and '18:00:00';
```

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

```
SELECT distinct ti.PROFILE_ID,pi.first_name,pi.email_id,pi.mobile_number FROM
```

```
air_ticket_infoti join air_passenger_profile pi on pi.profile_id=ti.profile_id where
flight_id=3148 order by pi.first_name;
```

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

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

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

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

21.Please follow instructions given below.

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

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

22.Please follow instructions given below.

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

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

23.Please follow instructions given below.

Write a Query to display serial no, first name,mobilenumber,email id of all the passengers who holds email address from gmail.com.
The Serial No will be the last three digits of profile ID.
Hint: Use SERIAL_NO as Alias name for serial number.

Display the records sorted in ascending order based on name.

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

24.Please follow instructions given below.

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

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

```
selectf.flight_id,
f.from_location,
f.to_location,
count(at.flight_departure_date) as Number_of_Services
fromair_flight f join air_ticket_info  at
onat.flight_id=f.flight_id  where month(at.flight_departure_date)=05
group by f.flight_id
having count(at.flight_departure_date)<=all
                    (select count(at.flight_departure_date) from air_flight f
joinair_ticket_info at  on  at.flight_id=f.flight_id
where month(at.flight_departure_date)=05     group by f.flight_id);
```

25.Please follow instructions given below.

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

Hint: Get the distinct from location and to location.

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

```
SELECT f.from_location ,
COUNT(f.flight_id) AS No_of_Flights
FROM  air_flight f
JOIN
(
SELECT DISTINCT from_location AS Location
FROM  air_flight
UNION
SELECT DISTINCT to_location AS Location
FROM air_flight
)a
ON f.from_location = a.location
GROUP BY f.from_location
ORDER BY f.from_location;
```

26.Please follow instructions given below.

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

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

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

```
SELECT f.flight_id,
f.from_location,
f.to_location,
t.FLIGHT_DEPARTURE_DATE,
t.No_of_Passengers
FROM  air_flight f
JOIN
(
SELECT flight_id,
            FLIGHT_DEPARTURE_DATE,
COUNT(ticket_id) AS No_of_Passengers
FROM   air_ticket_info
GROUP BY flight_id,
                         FLIGHT_DEPARTURE_DATE
)t
ON f.flight_id = t.flight_id
order by f.flight_id, t.flight_departure_date;
```

27.Please follow instructions given below.

Write a query to display the flight details in which more than 10% of seats have been booked. The query should display Flight_Id, From_Location, To_Location,Total_Seats, seats booked as “No_of_Seats_Booked” .
Display the records sorted in ascending order based on flight id and then by No_of_Seats_Booked.

```
SELECT f.flight_id,
f.from_location,
f.to_location,
f.Total_Seats,
  (f.Total_Seats)-(fd.Available_Seats) AS No_of_Seats_Booked
FROM  air_flight f JOIN air_flight_detailsfd
ON      f.flight_id = fd.flight_id
AND   (f.Total_Seats)-(fd.Available_seats) > (f.total_seats *0.1)
order by f.flight_id, No_of_Seats_Booked;
```

28.Please follow instructions given below.

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

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

```
SELECT distinct  f.flight_id,
fd.flight_departure_date,
f.from_location,
f.to_location,
f.duration
FROM   air_flight f JOIN air_flight_detailsfd
ON     f.flight_id = fd.flight_id
WHERE  f.duration< '01:10:00'
order by f.flight_id, fd.flight_departure_date;
```

29.Please follow instructions given below.
Write a query to display the flight_id, from_location,to_location,number of services as “No_of_Services” , average ticket price as “Average_Price” whose average ticket price is greater than the total average ticket cost of all flights. Order the result by lowest average price.

```
SELECT f.flight_id,
f.from_location,
f.to_location,
a.No_of_Services,
a.Average_Price
FROM air_flight f JOIN
(
SELECT flight_id,
COUNT(flight_departure_date) AS No_of_Services,
AVG(price) AS Average_Price
FROM  air_flight_details
GROUP BY flight_id
HAVING AVG(price) > (SELECT AVG(price) FROM air_flight_details)
)a
ON f.flight_id = a.flight_id order by average_price ASC
```

AIRLINE MANAGEmeNT OLD

1.Write a query to display the average monthly ticket cost for each flight in ABC Airlines. The query should display the Flight_Id,From_location,To_Location,Month Name as “Month_Name” and average price as “Average_Price”
Display the records sorted in ascending order based on flight id and then by Month Name.

```
ANS:  SELECT f.flight_id,
f.from_location,
f.to_location,
fp.Month_Name,
fp.Average_Price
FROM  air_flight f,
(
SELECT flight_id,
MONTHNAME(flight_departure_date) AS Month_Name,
AVG(price) as Average_Price
FROM air_flight_details
GROUP BY flight_id,
MONTHNAME(flight_departure_date)
) fp
WHERE f.flight_id = fp.flight_id
andf.airline_name = 'ABC AIRLINES'
order by f.flight_id, fp.Month_Name;
```

2. Write a query to display the customer(s) who has/have booked least number

of tickets in ABC Airlines. The Query should display profile_id, customer’s first_name, Address and Number of tickets booked as “No_of_Tickets” Display the records sorted in ascending order based on customer's first name.

```
SELECT cus.profile_id,
cus.first_name,
cus.address,
mintkt.No_of_Tickets
FROM  air_passenger_profilecus,
      (SELECT MIN(s.Tot_No_of_Tickets) AS No_of_Tickets
       FROM   (SELECT profile_id,
COUNT(ticket_id) AS Tot_No_of_Tickets
               FROM   air_ticket_info
               GROUP BY profile_id
              ) s
      ) mintkt,
( SELECTprofile_id,
COUNT(ticket_id) AS Tot_No_of_Tickets
FROM  air_ticket_info
       GROUP BY profile_id
     ) tottkt
WHERE mintkt.No_of_Tickets = tottkt.Tot_No_of_Tickets
AND      cus.profile_id = tottkt.profile_id
order by cus.first_name;
```

3. Write a query to display the number of flight services between locations in a month. The Query should display From_Location, To_Location, Month as “Month_Name” and number of flight services as “No_of_Services”. Hint: The Number of Services can be calculated from the number of scheduled departure dates of a flight. The records should be displayed in ascending order based on

From_Location and then by To_Location and then by month name.

```
SELECT f.from_location,
f.to_location,
s.Month_Name,
SUM(s.No_of_Services) AS No_of_Services
FROM air_flight f JOIN
(SELECT flight_id,
MONTHNAME(flight_departure_date) AS Month_Name,
COUNT(flight_departure_date) AS No_of_Services
FROM  air_flight_details
GROUP BY flight_id,
MONTHNAME(flight_departure_date)
) s
ON s.flight_id  =f.flight_id
GROUP BY f.from_location,
f.to_Location,
s.Month_Name
order by f.from_location,
f.to_Location, s.Month_Name;
```

4. Write a query to display the customer(s) who has/have booked maximum number of tickets in ABC Airlines. The Query should display profile_id, customer’s first_name, Address and Number of tickets booked as “No_of_Tickets” Display the records in ascending order based on customer's first name.

```
SELECT cus.profile_id,
cus.first_name,
cus.address,
maxtkt.No_of_Tickets as No_of_Tickets
FROM  air_passenger_profilecus,
      (SELECT MAX(s.Tot_No_of_Tickets) AS No_of_Tickets
       FROM   (SELECT profile_id,
COUNT(ticket_id) AS Tot_No_of_Tickets
                FROM   air_ticket_info
                GROUP BY profile_id
               ) s
      ) maxtkt,
( SELECTprofile_id,
COUNT(ticket_id) AS Tot_No_of_Tickets
FROM  air_ticket_info
       GROUP BY profile_id
      ) tottkt
WHERE maxtkt.No_of_Tickets = tottkt.Tot_No_of_Tickets
AND     cus.profile_id = tottkt.profile_id
order by cus.first_name;
```

5.Write a query to display the number of tickets booked from Chennai to Hyderabad. The Query should display passenger profile_id,first_name,last_name, Flight_Id , Departure_Date and number of tickets booked as “No_of_Tickets”. Display the records sorted in ascending order based on profile id and then by flight id and then by departure date.

```
SELECT cus.profile_id,
cus.first_name,
cus.last_name,
tkt.flight_id,
tkt.flight_departure_date,
tkt.No_of_Tickets
```

```
FROM   air_passenger_profilecus JOIN
(
SELECT profile_id, FLIGHT_ID,FLIGHT_DEPARTURE_DATE,COUNT(ticket_id) AS
No_of_Tickets
FROM   air_ticket_info
WHERE  flight_id IN (SELECT flight_id
                     FROM   air_flight
WHERE  from_location = 'Chennai'
        AND    to_location = 'Hyderabad'
    )
GROUP BY profile_id,flight_id,flight_departure_date
)tkt
ON cus.profile_id = tkt.profile_id
order by cus.profile_id, tkt.flight_id, tkt.flight_departure_date;
```

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

```
SELECT fd.flight_id,af.FROM_LOCATION,af.TO_LOCATION, fd.price FROM
air_flight_detailsfd join air_flightaf on af.FLIGHT_ID=fd.FLIGHT_ID where
substring(fd.flight_departure_date,6,2)='04' order by fd.flight_id, af.FROM_LOCATION;
```

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

```
SELECT f.flight_id,
f.from_location,
f.to_location,
AVG(fd.Price) AS Price
FROM  air_flight f JOIN
air_flight_detailsfd
ON      f.flight_id = fd.flight_id
GROUP BY f.flight_id order by f.flight_id, f.from_location, f.to_location;
```

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

Hint: Query should fetch unique customers irrespective of multiple tickets booked. Display the records sorted in ascending order based on profile id.

```
SELECT DISTINCT c.profile_id,
CONCAT(c.first_name, ',',c.last_name) AS customer_name,
c.address
FROM  air_passenger_profile c
JOIN    air_ticket_info t
ON      c.profile_id = t.profile_id
JOIN   air_flight f
ON     f.flight_id = t.flight_id
AND  f.from_location = 'Chennai'
AND  f.to_location = 'Hyderabad'
order by c.profile_id;
```

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

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

10. Write a query to display the total number of tickets as “No_of_Tickets” booked in each flight in ABC Airlines. The Query should display the flight_id, from_location, to_location and the number of tickets.
Display only the flights in which atleast 1 ticket is booked.
Display the records sorted in ascending order based on flight id.

```
SELECT f.flight_id,
f.from_location,
f.to_location,
COUNT(t.ticket_id) AS No_of_Tickets
FROM air_ticket_infot JOIN
air_flight f
ON      f.flight_id = t.flight_id
where AIRLINE_NAME = 'ABC AIRLINES'
GROUP by f.flight_id
ORDER by f.flight_id;
```

11. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight_id, number of services as “No_of_Services” and the cost as “Total_Price” in the same order.Order the result by Total Price in descending order and then by flight_id in descending order.   Hint:The number of services can be calculated from the number of scheduled departure dates of the flight

```
SELECT flight_id,
COUNT(flight_departure_date) AS No_of_Services,
SUM(price) AS Total_Price
FROM  air_flight_details
GROUP BY flight_id order by total_price DESC, flight_id DESC;
```

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

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

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

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

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

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

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

```
SELECT distinct ti.PROFILE_ID,pi.first_name,pi.mobile_number, pi.email_id FROM
air_ticket_infoti join air_passenger_profile pi on pi.profile_id=ti.profile_id where
flight_id in (SELECT FLIGHT_ID FROM air_flight where FROM_LOCATION
='HYDERABAD' and to_location ='CHENNAI') order by ti.profile_id
```

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

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

```
SELECT DISTINCT p.profile_id,
p.first_name,
p.last_name,
t.flight_id,
t.flight_departure_date,
f.departure_time,
f.arrival_time,
ADDTIME(f.departure_time,'01:00:00') AS Delayed_Departure_Time,
ADDTIME(f.arrival_time,'01:00:00') AS Delayed_Arrival_Time
FROM air_passenger_profile p
JOIN air_ticket_info t
ON p.profile_id = t.profile_id
AND t.flight_departure_date = '2013-05-06'
JOIN air_flight f
ON t.flight_id = f.flight_id
AND f.from_location = 'Chennai'
AND f.to_location = 'Hyderabad'
order by p.profile_id;
```

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

Hint: Use String functions to get the base location of customer from their Address and give alias name as “Base_Location” Display the records sorted in ascending order based on customer first name.

```
SELECT cus.profile_id,
cus.First_name,
SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')-
INSTR(cus.address,',')-1)
            AS Base_Location,
s.No_of_Tickets
FROM  air_passenger_profilecus JOIN
(
SELECT profile_id,
COUNT(ticket_id) AS No_of_Tickets
FROM  air_ticket_info
GROUP BY profile_id
)s
ON s.profile_id = cus.profile_id
AND  SUBSTR(cus.address,INSTR(cus.address,',')+1,INSTR(cus.address,'-')-
INSTR(cus.address,',')-1) = 'Kochi'
order by first_name;
```

17. Write a query to display the flight_id, from_location, to_location, number of Services as “No_of_Services” offered in the month of May. Hint:The number of services can be calculated from the number of scheduled departure dates of the flight. Display the records sorted in ascending order based on flight id.

```
SELECT f.flight_id,
f.from_location,
f.to_location,
s.No_of_Services
FROM  air_flight f JOIN
(
SELECT flight_id,
COUNT(flight_departure_date) AS No_of_Services
FROM  air_flight_details
WHERE MONTH(flight_departure_date) = 5
GROUP BY flight_id
) s
ON f.flight_id = s.flight_id
order by f.flight_id;
```

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

```
SELECT PROFILE_ID,LAST_NAME,MOBILE_NUMBER,EMAIL_ID FROM air_passenger
_profile where address like '%CHENNAI%' ORDER BY PROFILE_ID;
```

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

```
SELECT count(flight_id) FLIGHT_COUNT FROM air_flight where FROM_LOCATION='
CHENNAI' and departure_time between '06:00:00' and '18:00:00';
```

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

```
SELECT distinct ti.PROFILE_ID,pi.first_name,pi.email_id,pi.mobile_number FROM
air_ticket_infoti join air_passenger_profile pi on pi.profile_id=ti.profile_id where
flight_id=3148 order by pi.first_name;
```

EMPLOYEE MANAGEMENT SYSTEM

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

```
SELECT s.emp_id, s.first_name, count(e.emp_id) as 'NO_OF_REPORTEES'
FROM emp_details e, emp_details s
WHERE s.emp_id=e.supervisor_id
GROUP BY s.emp_id
HAVING NO_OF_REPORTEES IN ( SELECT MAX(c)
    FROM ( SELECT count(*) c
     FROM emp_details e
     GROUP BY supervisor_id)a
   );
```

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

```
SELECT e.emp_id, e.first_name, e.designation_id, e.join_date, e.supervisor_id
FROM emp_details e
WHERE e.supervisor_id = (SELECT emp_id FROM emp_details WHERE
first_name='Rahul')
 AND e.emp_id NOT IN (SELECT DISTINCT emp_id FROM project_alloc);
```

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

```
SELECT e.emp_id, e.first_name, e.last_name, e.salary, e.designation_id
FROM emp_details e, (select designation_id, max(salary) max_sal FROM
emp_details GROUP BY designation_id)e1
WHERE e.designation_id = e1.designation_id AND e.salary=e1.max_sal AND
e.designation_id!='VP'
GROUP BY e.designation_id
ORDER BY e.designation_id desc;
```

4.Write to query to find for each project, find the number of employees who have either an 'OOP' or 'TEST' skill and display the project name and number of employees.
Hint: Use "NO_OF_EMPLOYEE" as alias name for displaying the number of employees.

```
SELECT p.proj_name, count(e.emp_id) as 'NO_OF_EMPLOYEE', s.skill_name
FROM project_details p, project_alloc a, skill s, emp_details e
WHERE e.emp_id=a.emp_id AND p.proj_id=a.proj_id AND e.skill_id=s.skill_id AND
s.skill_name IN ('OOP', 'TEST')
GROUP BY p.proj_name;
```

5.Write a query to find all peoples who have either a 'JAV' or 'DES' skills and either 'E1' or 'E2' proficiency levels and
display the employee ID, employee first name, designation ID, skill ID, proficiency level and years of experience of those peoples.

```
SELECT e.emp_id, e.first_name, e.designation_id, e.skill_id, s.proficiency_level,
s.year_of_experience
FROM emp_details e, skill s
WHERE e.skill_id=s.skill_id AND s.skill_name IN ('JAV', 'DES') AND s.proficiency_le
vel IN ('E1', 'E2');
```

6.Write a query to display the first name, last name, salary, bonus (calculated 3 percent bonus per salary value) and
weekly salary (including bonus) of all employees and round the weekly salary to remove the decimals. Sort the result based on the salary in descending order.
Hint: Formula for calculating bonus is salary*.03 and for calculating weekly salary is (salary/52) + ((salary*.03)/52)
Use "BONUS" as alias name for displaying bonus and "WEEKLY_SALARY" as alias name for displaying the weekly salary with bonus.

```
SELECT e.first_name, e.last_name, e.salary, 0.3*(e.salary) as 'BONUS',
round(((e.salary/52) + ((e.salary*.03)/52)),0) as 'WEEKLY_SALARY'
FROM emp_details e
ORDER BY e.salary DESC;
```

7.Write a query to display the employee id and employee name of employee who is a developer in google testing team.
Display the employee name in below format ie first character of first name dot last name.
Example Assume first name is "Ram" and last name is "Kumar" then employee name will be "R.Kumar".
Hint: Use "EMPLOYEE_NAME" as alias name for displaying employee name.
Example,

```
emp_id    EMPLOYEE_NAME
1000      R.Kumar
```

```
SELECT e.emp_id, concat(left(e.first_name,1), '.', e.last_name) as 'EMPLOYEE_NA
ME'
FROM emp_details e, dept_details d, role r, project_alloc a
```

```
WHERE e.dept_no=d.dept_no
 AND e.emp_id=a.emp_id
 AND a.role_id=r.role_id
 AND d.dept_name='Facebook Developer'
 AND r.role_name='Developer';
```

8.Write a query to find the all employees who are not currently in projects and are whose skill is 'DES' and proficiency level is 'E1' and
display the employee ID, employee first name, skill id, proficiency level and experience in years of those employees.
Hint: Employees who are not currently in projects means they either got released from the project or not at all allocated to any project.

```
SELECT e.emp_id, e.first_name, s.skill_id, s.proficiency_level, s.year_of_experience
FROM emp_details e, skill s
WHERE e.skill_id=s.skill_id
 AND s.skill_name='DES'
 AND s.proficiency_level='E1'
 AND e.emp_id NOT IN (SELECT emp_id FROM project_alloc);
```

9.Write a query to find all people who have a 'JAV' skill and 'E2' proficiency level and display the distinct employee first name,
project ID, role ID, skill ID, proficiency level and release date of those peoples who are currently in projects.
Hint: Employees who are currently in projects means they should not have got released.

```
SELECT DISTINCT e.first_name, a.proj_id, a.role_id, s.skill_id, s.proficiency_level
FROM emp_details e, skill s, project_alloc a
```

```
WHERE e.skill_id=s.skill_id
 AND e.emp_id=a.emp_id
 AND s.skill_name='JAV'
 AND s.proficiency_level='E2';
```

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

```
SELECT ds.designation_id, ds.designation_name, sum(e.salary) as
'TOTAL_SALARY'
FROM designation ds, emp_details e
WHERE ds.designation_id=e.designation_id AND e.join_date>'2004-01-01'
GROUP BY ds.designation_id
HAVING TOTAL_SALARY>50000;
```

11.Write a query to find the all employees who are currently in projects and whose designation is 'C2' and display the employee ID, employee first name, designation ID, join date and supervisor ID of the employees.
Hint: Employees who are currently in projects means they are not released from project.

```
SELECT e.emp_id, e.first_name, e.designation_id, e.join_date, e.supervisor_id
FROM emp_details e
WHERE e.designation_id='C2'
 AND e.emp_id IN (SELECT DISTINCT emp_id FROM project_alloc);
```

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

```
SELECT e.emp_id, e.first_name, e.last_name, e.designation_id, e.join_date,
e.supervisor_id
FROM emp_details e
WHERE e.supervisor_id IN (SELECT emp_id FROM emp_details WHERE
first_name='Ram')
 AND e.join_date BETWEEN '2002-01-01' and '2010-01-01'
 AND e.emp_id NOT IN (SELECT DISTINCT emp_id FROM project_alloc);
```

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

```
SELECT DISTINCT e.emp_id, e.first_name, s.first_name as 'SUPERVISOR_NAME',
ds.designation_name, p.proj_id, p.proj_name, r.role_id, r.role_name
FROM emp_details e, emp_details s, designation ds, project_details p,
project_alloc a, role r
WHERE e.designation_id=ds.designation_id
```

```
 AND e.emp_id=a.emp_id
 AND a.proj_id=p.proj_id
 AND a.role_id=r.role_id
 AND e.supervisor_id=s.emp_id
 AND s.first_name='Ram';
```

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

```
SELECT e.emp_id, e.first_name, p.proj_id, p.proj_name, count(a.role_id)
FROM emp_details e, project_details p, project_alloc a
WHERE e.emp_id=a.emp_id AND a.proj_id=p.proj_id
GROUP BY e.emp_id
HAVING count(p.proj_id) IN ( SELECT max(c)
     FROM ( SELECT e.emp_id, e.first_name, count(p.proj_id) c, p.proj_name,
count(a.role_id)
      FROM emp_details e, project_details p, project_alloc a
      WHERE e.emp_id=a.emp_id AND a.proj_id=p.proj_id
      GROUP BY e.emp_id
              )a1


   )

 AND count(a.role_id) IN ( SELECT max(c)
     FROM ( SELECT e.emp_id, e.first_name, p.proj_id, p.proj_name, count(a.role_i
d) c
      FROM emp_details e, project_details p, project_alloc a
      WHERE e.emp_id=a.emp_id AND a.proj_id=p.proj_id
      GROUP BY e.emp_id
     )a12
     );
```

15.Write a query to display the name, join date of the employees whose join date is greater than 01-JAN-2004 and sort the result based on the employee name in ascending order.
Display the name as, First Character of first name, Dot, Last Name.
Example: Assume first name is "Ram" and last name is "Kumar" then display employee name as "R.Kumar"
Hint: Use the EMPLOYEE_NAME as alias name for displaying the employees name.

```
SELECT concat(substr(e.first_name, 1,1), '.', e.last_name) as EMPLOYEE_NAME,
e.join_date
FROM emp_details e
WHERE e.join_date>'2004-01-01'
ORDER BY e.first_name ASC;
```

16.Write a query to display the distinct employee id, employee first name, designation name and supervisor ID of all employees who work for the project "Google Testing".
Hint: There can be more than one employee working in same project with multiple roles, display their details only once.

```
SELECT DISTINCT e.emp_id, e.first_name, ds.designation_name, e.supervisor_id,
p.proj_name
FROM emp_details e, designation ds, project_details p, project_alloc a
WHERE e.designation_id=ds.designation_id
 AND e.emp_id=a.emp_id
 AND a.proj_id=p.proj_id
 AND p.proj_name='GMS'
GROUP BY e.emp_id;
```

17.Write a query to find the employee Id, first name, designation name, project name and role name of the employees who are second level reportees of a given employee 'Mark'.
Example, Assume, employee 'A' reports to "Mark" and employee "B" and "C" reports to employee "A". Here, display the above details for employee "B" and "C".

```
SELECT e.emp_id, e.first_name, ds.designation_name, p.proj_name, r.role_name
FROM emp_details e, designation ds, project_details p, project_alloc a, role r
WHERE e.designation_id=ds.designation_id
 AND e.emp_id=a.emp_id
 AND p.proj_id=a.proj_id
 AND a.role_id=r.role_id
 AND e.emp_id IN ( SELECT e.emp_id
     FROM emp_details e, emp_details l1, emp_details l2
     WHERE e.supervisor_id=l1.emp_id
      AND l1.supervisor_id=l2.emp_id
      AND l2.first_name='John'
   );
```

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

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

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

```
SELECT e.emp_id, e.first_name, ds.designation_name
FROM emp_details e, designation ds
WHERE e.designation_id=ds.designation_id
 AND e.emp_id NOT IN ( SELECT emp_id
    FROM project_alloc
      )
 AND e.emp_id IN ( SELECT e.emp_id
    FROM emp_details e, emp_details l1, emp_details l2
    WHERE e.supervisor_id=l1.emp_id
     AND l1.supervisor_id=l2.emp_id
     AND l2.first_name='John'
   );
```

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

20.Write a query to display employee id,first name, last name,designation name

and skill name of all the employees irrespective of skills available. If no skills are available then display it as "N/A" for that particular employee. If an employee has 2 skills then display both the records. Sort the records based on designation name in descending order.
Hint:Use SKILL_NAME as alias name for skill name column.

```
SELECT e.emp_id, e.first_name, e.last_name, ds.designation_name, ifnull(s.skill_
name, 'N/A') as 'SKILL_NAME'
FROM (emp_details e LEFT OUTER JOIN skill s ON  e.skill_id=s.skill_id) LEFT
OUTER JOIN designation ds ON e.designation_id=ds.designation_id
ORDER BY ds.designation_name DESC;
```

21.Write a query to display the employee id, employee first name, designation name, supervisor id, supervisor first name and supervisor designation ID of all employees including employees who does not have supervisor and sort them by employee designation name in ascending order.
Hint: Use "SUPERVISOR_NAME" as alias name for displaying the supervisor first name and "SUPERVISOR_DESIGNATION" as alias name for displaying the supervisor designation ID

```
SELECT e.emp_id, e.first_name, ds.designation_name, s.emp_id, s.first_name as
'SUPERVISOR_NAME', s.designation_id as 'SUPERVISOR_DESIGNATION'
FROM (emp_details e LEFT OUTER JOIN emp_details s ON e.supervisor_id=s.emp
_id) LEFT OUTER JOIN designation ds ON e.designation_id=ds.designation_id
ORDER BY ds.designation_name ASC;
```

MOVIE SCHEMA

1.count the members who has gold cards

```
ans:select count(customer_id) from customer_card_details where card_id  in
(select card_id from library_card_master where description='gold
card');
```

2.display the name of member who issued movie and the count of the movies issued and display 0 for the member who have not issued any movie

```
ans:select customer_name,count(movie_id) count
from  customer_issue_details a,customer_master b
where  a.customer_id=b.customer_id
group by b.customer_id union
 select  customer_name,0 as count
from customer_master
where customer_id not in
(select customer_id from customer_issue_details);
```

3.display the name of the person starting with letter 'r' and category
is 'comedy'
ans:

```
   select distinct a.customer_name
from customer_master  a,customer_issue_details b,movies_master c
where  a.customer_id=b.customer_id and b.movie_id=c.movie_id and
c.movie_category='comedy'
and a.customer_name like 'r%';
```

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

```
select a.customer_id,customer_name,count(a.movie_id)*c.rent_cost  rent
from customer_issue_details a,customer_master b ,(select  movie_id,rent_cost
from movies_master) c
where  a.customer_id=b.customer_id and a.movie_id=c.movie_id
 group by
b.customer_id;
```

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

```
  select a.customer_id,customer_name,b.card_id,round(amount/54.42)  Amount
from customer_master a,customer_card_details  b,library_card_master c
where a.customer_id=b.customer_id and
b.card_id=c.card_id;
```

6.display id,name of customers who dont have library card but still
have issued the movie
ans:

```
  select distinct b.customer_id,customer_name
from  customer_issue_details a,customer_master b
where  a.customer_id=b.customer_id
and a.customer_id not in (select
customer_id from customer_card_details);
```

7.display the no.of customers with first letter 'r' and have paid fine
i.e actual return date is greater than return date

ans:

```
   select count(b.customer_id) count from (select  customer_id
from  customer_issue_details
where return_date>actual_date_return
group by  customer_id) b,customer_master c
where b.customer_id=c.customer_id and
c.customer_name like 'r%';
```

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

ans:

```
  select customer_name,a.customer_id
from customer_issue_details  a,customer_master b
where a.customer_id=b.customer_id
group by  a.customer_id
having count(a.movie_id)=(
select min(a.count) min from  (select customer_id,count(movie_id) count
from customer_issue_details
group by customer_id) a)
or count(a.movie_id)=(
select max(b.count) min  from (select customer_id,count(movie_id) count
from  customer_issue_details
group by customer_id) b);
```

9.display id,name,mobile num and description of all customers.if
mobile num is not available then display address as alias contact,for
those who does't have library cards display null as description

ans:

```
select a.customer_id,a.customer_name,coalesce (a.contact_no,a.contact_add)
contact,description
from customer_master  a,library_card_master b,customer_card_details c
where  a.customer_id=c.customer_id and c.card_id=b.card_id union
select  a.customer_id,a.customer_name,coalesce(a.contact_no,a.contact_add)
 contact,null as description
from customer_master a
where customer_id not  in (select customer_id from customer_card_details) ;
```

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

```
ans:select m.customer_id,customer_name,movie_id,count(i.movie_id) as
count_of_movie
from customer_master m join customer_issue_details i
on m.customer_id = i.customer_id
group by movie_id
having count(movie_id)>1;
```

11.display customer information those who has library cards

```
ans:
select a.customer_id,customer_name
from customer_card_details  a,customer_master b,library_card_master c
where
a.customer_id=b.customer_id and a.card_id=c.card_id;
```

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

```
ans:select distinct b.customer_id,customer_name
from  customer_issue_details a,customer_master b
where  a.customer_id=b.customer_id
and a.customer_id not in (select
customer_id from customer_card_details);
```

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

```
ans:
select substring(issue_id,4) sr_no,issue_id as  video_id,customer_id,movie_id
from customer_issue_details order by
sr_no;
```

14.display total revenue spent on videos by each customer

```
ans:select a.customer_id,count(a.movie_id)*c.rent_cost rent
from  customer_issue_details a,customer_master b ,(select
movie_id,rent_cost  from movies_master) c
where a.customer_id=b.customer_id and  a.movie_id=c.movie_id
 group by b.customer_id union
select customer_id,0  as rent
```

```
from customer_master
where customer_id not in(select
customer_id from customer_issue_details);
```

15.display customer name in perfect order
i.e 1st letter in ucase remaining lcase
ans:

```
select concat(ucase(substring(customer_name,1,1)),lcase(substring
(customer_name,2,5)) )
from customer_master;
```

16.count how many times a movie issued and arrange them in desc order and display 0 for the movie not issued
ans:

```
select * from (select a.movie_id,count(b.movie_id) count
from  movies_master a,customer_issue_details b
where a.movie_id=b.movie_id
group by a.movie_id union
select movie_id ,0 as count from  movies_master
where movie_id not in (select movie_id from
customer_issue_details)) x order by movie_id desc;
```

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

```
SELECT CUSTOMER_ID,CUSTOMER_NAME,coalesce(CAST(CONTACT_NO AS
CHAR),CAST(CONTACT_ADD AS CHAR))
from customer_master;

SELECT CUSTOMER_ID,CUSTOMER_NAME,(CASE
                    WHEN CAST(CONTACT_NO AS CHAR) IS  NULL THEN CAST(CONTACT_ADD AS CHAR)
                    ELSE CAST(CONTACT_NO AS CHAR)
                    END) AS CONTACT
                    FROM CUSTOMER_MASTER;
```

18.waq that num of customers registered in 2012 year and provided contact num
use NO_OF_CUSTOMERS as alias name.

```
ANS:SELECT COUNT(*) AS NO_OF_CUSTOMERS
FROM CUSTOMER_MASTER
WHERE DATE_OF_REGISTRATION LIKE '%2012%' AND CONTACT_NO IS NOT NULL;
```

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

```
ANS:SELECT seLect C.CUSTOMER_ID,C.CUSTOMER_NAME,year (C.DATE_OF_REGISTRATION) REGISTERED_YEAR,L.CARD_ID,D.ISSUE_DATE
FROM  CUSTOMER_MASTER C,LIBRARY_CARD_MASTER L,CUSTOMER_CARD_DETAILS D
WHERE
C.CUSTOMER_ID=D.CUSTOMER_ID AND L.CARD_ID=D.CARD_ID;
```

20. (***REPEATED)waq to display movie name and num of times movie issued to
customers..in case of no movie issued
to customers display 0.. use alias name as NO_OF_TIMES

```
SELECT M.MOVIE_NAME,COUNT(*) AS NO_OF_TIMES
FROM CUSTOMER_ISSUE_DETAILS D,MOVIES_MASTER M,CUSTOMER_MASTER C
WHERE M.MOVIE_ID=D.MOVIE_ID AND D.CUSTOMER_ID=C.CUSTOMER_ID
GROUP BY C.CUSTOMER_ID
```

```
union
SELECT MOVIE_NAME,0
FROM MOVIES_MASTER
WHERE MOVIE_NAME NOT IN (SELECT M.MOVIE_NAME
      FROM CUSTOMER_ISSUE_DETAILS D,MOVIES_MASTER M,CUSTOMER_MASTER  C
      WHERE M.MOVIE_ID=D.MOVIE_ID AND D.CUSTOMER_ID=C.CUSTOMER_ID
      GROUP BY C.CUSTOMER_ID);
```

21.waq to display customer id and customer name ,num of times movie issued to customer in comedy movie category
display only customers who has issude more than once
ANS:

```
SELECT M.CUSTOMER_NAME,M.CUSTOMER_ID,P.NO_OF_TIMES
FROM CUSTOMER_MASTER M,
(SELECT I.CUSTOMER_ID,COUNT(*) AS NO_OF_TIMES
FROM CUSTOMER_ISSUE_DETAILS I,(
                        SELECT MOVIE_ID,MOVIE_CATEGORY
                        FROM MOVIES_MASTER
                        WHERE MOVIE_CATEGORY='COMEDY')T
WHERE I.MOVIE_ID=T.MOVIE_ID
GROUP BY I.CUSTOMER_ID
HAVING COUNT(*)>1)P
WHERE M.CUSTOMER_ID=P.CUSTOMER_ID;
```

22.(***REPEATED)waq to display customerid and total rent paid by them.
use alias name as total_cost.

```
ANS:select * from movies_master;
select c.customer_id,sum(rent_cost) as total_cost
from customer_master c,customer_issue_details d,movies_master m
```

```
where c.customer_id=d.customer_id and m.movie_id=d.movie_id
group by customer_id;
```

23.waq to display customerid,cusname,contactno,num of movies issued to customer based on category and category
display the customer who has issude for more than one movie from that caregory.
display phone num as "+91-987-654-3210".

```
ANS:select m.customer_id,customer_name,
concat('+91-',substring(contact_no,1,3),'-',substring (contact_no,4,3),'-',substring(contact_no,7) )as contact_no,
count(i.movie_id) as no_of_movies,movie_category
from customer_master m join customer_issue_details i join  movies_master mv
on m.customer_id = i.customer_id and i.movie_id = mv.movie_id
group by m.customer_id
having count(*) >1;
```

MOBILE MANAGEMENT SCHEMA

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

```
select IME_NO, MODEL_NAME from mobile_master where manufacturer = 'Nokia';
```

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

```
select m.IME_NO, m.MODEL_NAME, m.MANUFACTURER, s.CAMERA_QUALITY
from mobile_master m join mobile_specification s on(m.ime_no = s.ime_no)
where camera_quality = '5MP';
```

3. "Write a Query to display the Mobile Model Name and respective number of mobiles sold on the date 23-Apr-2012 for each mobile model.
Hint: For example, if 2 ""Nokia 1100"" and 1 ""Nokia C5-03"" are sold on the date 23-Apr-2012 then display both the records. Use ""NoofMobilesSold"" as alias name for the number of mobiles field."

```
select model_name, count(model_name) NoOfMobilesSold from sales_info
where sales_date = '2012-04-23' group by model_name;
```

4. "Write a Query to display the distributor id, mobile model name, number of mobiles of the particular model name supplied to the distributors group by model name and distributor id and sort by the distributor id.
Hint: For example, if 3 ""Nokia 1100"" and 1 ""Nokia C5-03"" are sold to one distributor then display both the records. Display the distributor id, model name and number of mobiles of a particular model name. Use ""NoofMobilesSupplied

"" as alias name for the number of mobiles."

```
select d.distributor_id, m.model_name, count(model_name) NoOfMobilesSupplied from distributor d join mobile_master m on(d.distributor_id = m.distributor_id)
 group by m.model_name, m.distributor_id order by m.distributor_id;
```

5. "Write a Query to display the IME number, model name, manufacturer, price and discount of all mobiles regardless of whether the mobile is sold or not. Hint: Fetch the price, IME no and model name from mobile_master table.
Example: For the mobile model ""Samsung GalaxyTAB with IME NO ""MC1000103"" is sold and other with IME No ""MC1000110"" is not sold. Then both the mobiles details namely IME number, model name, manufacturer, price and discount needs to be displayed.

```
select DISTINCT m.ime_no, m.model_name, m.manufacturer, m.price, s.discount
 from mobile_master m join sales_info s;
```

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

```
select d.distributor_name, d.mobile, d.email from distributor d join
mobile_master m
on(d.distributor_id = m.distributor_id) where model_name = 'Nokia 1100';
```

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

```
select ime_no, model_name from mobile_master where model_name
```

```
NOT IN( select model_name from sales_info);
```

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

```
select m.ime_no, m.model_name, max(s.net_amount) Net_Amount from
mobile_master m
join sales_info s on(m.ime_no = s.ime_no);
```

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

```
select ime_no, model_name, manufacturer, price, price + (price * 10/100)
New_Price from mobile_master;
```

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

```
select model_name, manufacturer, price from mobile_master where price >=
8500 and price <= 25300 ;
```

11. Write a Query to display the Model Name, Manufacturer, Price, Warranty, Internal memory, memory card capacity, gprs support, Bluetooth, camera quality and OS for the mobile with IME NO "MC1000104

```
select m.Model_Name, m.Manufacturer, m.Price, m.Warranty_in_Years,
s.Internal_mem_in_MB,  s.Memory_Card_Capacity_GB, s.GPRS, s.Bluetooth,s.Ca
mera_Quality, s.OS from mobile_master m join mobile_specification s
on(m.ime_no = s.ime_no) where m.ime_no = 'MC1000104';
```

12. "Write a Query to display IME Number, Model Name, Manufacturer, Price ,GPRS information, Memory card capacity of mobiles which has GPRS support with memory card capacity 16GB or above.
Hint: For GPRS support use GPRS = “Yes”."

```
select m.IME_No, m.Model_Name, m.Manufacturer, m.Price, s.GPRS,
s.Memory_Card_Capacity_GB from mobile_master m join mobile_specification s
on(m.ime_no = s.ime_no) where s.GPRS = 'yes' and s.Memory_Card_Capacity_GB
 >= '16GB';
```

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

```
select c.Customer_Name, s.IME_No, s.Model_Name, s.Sales_Date, s.Net_Amount
 from customer_info c join sales_info s on( c.Customer_ID = s.Customer_ID )
group by c.Customer_Name;
```

14. "Write a Query to display the IME Number, model name, manufacturer, price and discount of all mobiles regardless of whether the mobile is sold or not. Hint: If not sold, display discount as ""Not Sold""

```
select  m.IME_No, m.Model_Name, m.Manufacturer, m.Price, IFNULL(s.Discount,
'NotSold') discount from mobile_master m left outer join sales_info s
on(m.ime_no = s.ime_no);
```

15. Write a Query to display the sales date and total net amount of all the mobiles based on the sales date that are sold between 20-APR-12 and

25-APR-12. Hint: Total net amount column should be displayed as "TotalNetAmount" (alias)

```
select Sales_Date, sum(Net_Amount) TotalNetAmount from sales_info  where
Sales_Date between '2012-04-12' and '2012-04-25' group by sales_date;
```

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

```
select m.IME_No, m.Model_Name, m.Manufacturer, m.Price, s.Battery_Life_Hrs
from mobile_master m join mobile_specification s on( m.ime_no = s.ime_no )
where s.Battery_Life_Hrs = (select max(Battery_Life_Hrs) from mobile_specification);
```

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

```
select IME_No, Model_Name, Manufacturer, Price from mobile_master
where price = (select max(price) from mobile_master);
```

18 Write a Query to display the Customer ID, Customer Name, Address, Total net amount of each customer. For example, assume customer_1 has purchased 2 mobiles such as "Nokia C5-03" and "Nokia Lumia" then sum the prices of both the mobiles and should be displayed against his customer id. Hint: Use Total_Net_Amount as alias.

```
select c.Customer_ID, c.Customer_Name, c.Address, sum(s.Net_Amount)
Total_net_amount
```

```
from customer_info c join sales_info s on(c.customer_id = s.customer_id)
```

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

```
select DISTINCT Model_Name, Manufacturer, max(Price) from mobile_master
where Manufacturer = 'Samsung';
```

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

```
select m.IME_No, m.Model_Name, m.Manufacturer, m.Distributor_ID, d.Distributor
_Name, m.Price from mobile_master m join distributor d on( m.distributor_id =
d.distributor_id)
```

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

```
select DISTINCT d.Distributor_ID, d.Distributor_Name, d.Address, d.Mobile,
d.Email
from distributor d join mobile_master m on(d.distributor_id = m.distributor_id)
where d.Distributor_ID = (select min(m.Distributor_ID) from mobile_master m
join distributor d on(m.Distributor_ID = d.distributor_id));
```

22 "Write a Query to display the Customer ID, Customer Name and Address of

the customers who have purchased the maximum amount.

```
select c.Customer_ID, c.Customer_Name, c.Address from customer_info c
join sales_info s on( c.customer_id = s.customer_id ) group by c.Customer_Name
having sum(net_amount) =
(select max(s) from (select sum(net_amount) s from sales_info group by
customer_id)t);
```

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

```
select m.model_name, m.ime_no, IF(s.discount,'Sold','NotSold') SalesStatus
from mobile_master m left outer join sales_info s on(m.ime_no = s.ime_no)
where m.model_name = 'Samsung GalaxyTAB';
```

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

```
select DISTINCT d.distributor_id, d.Distributor_Name, d.Address, d.Mobile from
distributor d
join mobile_master m on(d.distributor_id = m.distributor_id)
join mobile_specification s on(m.ime_no = s.ime_no)
where s.Network_3G = 'yes' and os like 'Android%' and camera_quality = '3.5MP';
```

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

```
select s.Model_Name, m.Manufacturer from sales_info s
join mobile_master m on (s.ime_no = m.ime_no) group by s.model_name having
count(*) =
(select max(cnt )from (select count(*) cnt  from sales_info group by
model_name)t);
```