BANK

create database bank;

use bank;

CREATE TABLE **customer\_master**(

CUSTOMER\_NUMBER VARCHAR(6),

FIRSTNAME VARCHAR(30),

middlename VARCHAR(30),

lastname VARCHAR(30),

CUSTOMER\_CITY VARCHAR(15),

CUSTOMER\_CONTACT\_NO VARCHAR(10),

occupation VARCHAR(10),

CUSTOMER\_DATE\_OF\_BIRTH DATE,

CONSTRAINT customer\_custid\_pk PRIMARY KEY (CUSTOMER\_NUMBER));

CREATE TABLE **branch\_master**(

branch\_id VARCHAR(6),

branch\_name VARCHAR(30),

branch\_city VARCHAR(30),

CONSTRAINT branch\_bid\_pk PRIMARY KEY (branch\_id));

CREATE TABLE **account\_master**

(account\_number VARCHAR(255),

customer\_number VARCHAR(255),

branch\_id VARCHAR(255),

opening\_balance INT(20),

account\_opening\_date DATE,

account\_type VARCHAR(10),

account\_status VARCHAR(10),

PRIMARY KEY (account\_number),

FOREIGN KEY (customer\_number) references customer\_master(customer\_number),

FOREIGN KEY (branch\_id) references branch\_master(branch\_id));

CREATE TABLE **transaction\_details**(

transaction\_number VARCHAR(6),

account\_number VARCHAR(6),

date\_of\_transaction DATE,

medium\_of\_transaction VARCHAR(20),

transaction\_type VARCHAR(20),

transaction\_amount INT(7),

CONSTRAINT transaction\_details\_tnumber\_pk PRIMARY KEY (transaction\_number),

CONSTRAINT transaction\_details\_acnumber\_fk FOREIGN KEY (account\_number)

REFERENCES account\_master (account\_number));

CREATE TABLE **loan\_details**

(customer\_number varchar(255),

branch\_id varchar(255),

loan\_amount bigint(20),

foreign key(customer\_number) references customer\_master(customer\_number));

insert into customer\_master values('C00001', 'RAMESH', 'CHANDRA', 'SHARMA', 'DELHI', '9543198345', 'SERVICE' ,'1976-12-06');

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

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

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

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

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

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

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

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

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

insert into branch\_master values('B00001', 'ASAF ALI ROAD','DELHI');

insert into branch\_master values('B00002','NEW DELHI MAIN BRANCH','DELHI');

insert into branch\_master values('B00003' ,'DELHI CANTT', 'DELHI');

insert into branch\_master values('B00004' ,'JASOLA', 'DELHI');

insert into branch\_master values('B00005' ,'MAHIM' ,'MUMBAI');

insert into branch\_master values('B00006' ,'VILE PARLE', 'MUMBAI');

insert into branch\_master values('B00007', 'MANDVI' ,'MUMBAI');

insert into branch\_master values('B00008' ,'JADAVPUR', 'KOLKATA');

insert into branch\_master values('B00009' ,'KODAMBAKKAM', 'CHENNAI');

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

insert into transaction\_details values('T00008', 'A00001', '2013-03-14', 'CHEQUE' ,'DEPOSIT' ,3000);

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

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

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

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

insert into Loan\_details values('C00001', 'B00001', 100000);

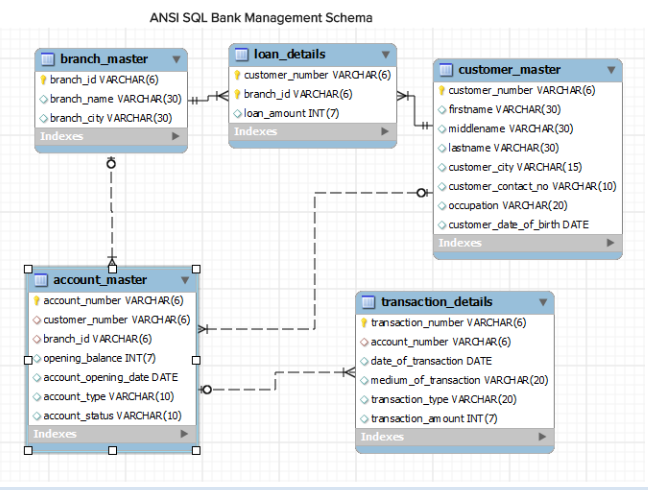
insert into Loan\_details values('C00002', 'B00002', 200000);

insert into Loan\_details values('C00009', 'B00008', 400000);

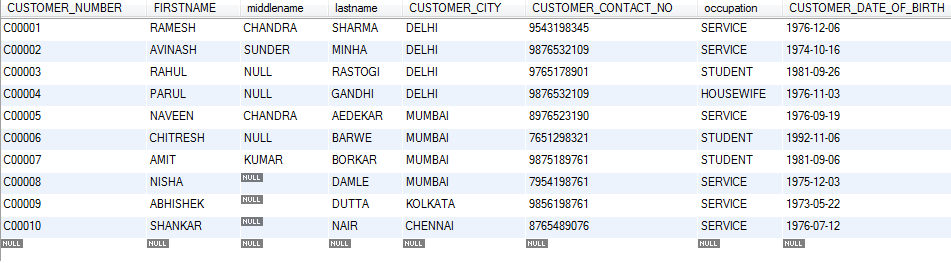
insert into Loan\_details values('C00010', 'B00009', 500000);

insert into Loan\_details values('C00001', 'B00003', 600000);

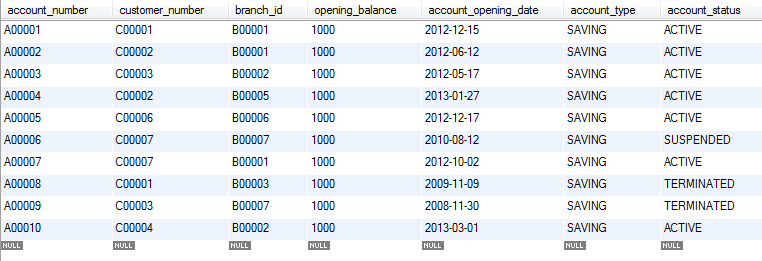
insert into Loan\_details values('C00002', 'B00001', 600000);



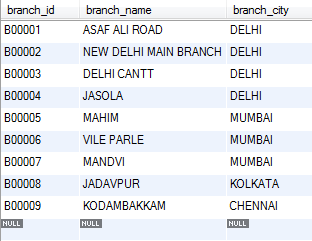
**CUSTOMER MASTER**



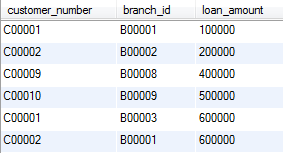
**ACCOUNT MASTER**



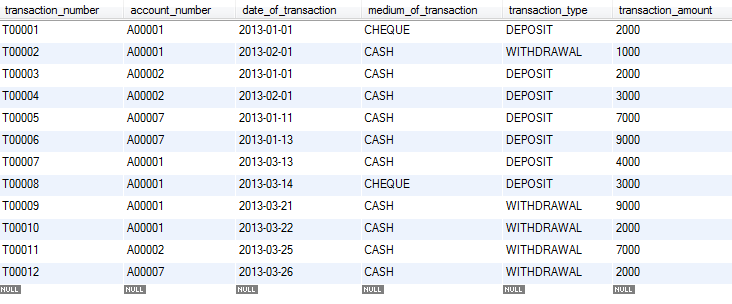
**BRANCH MASTER**



**LOAN DETAILS**



**TRANSACTION DETAILS**



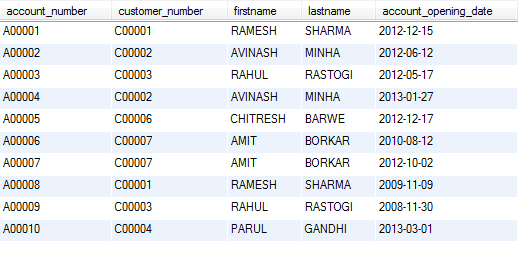
QUERIES

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

SELECT a.account\_number,c.customer\_number,c.firstname,c.lastname,a.account\_number

FROM customer\_master c JOIN account\_master a ON c.customer\_number=a.customer\_number

ORDER BY a.account\_number;



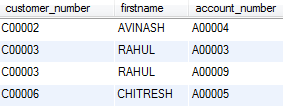
2. **Write a query to display the number of customer’s from Delhi. Give the count an alias name of Cust\_Count.**

SELECT count(customer\_number) Cust\_Count FROM customer\_master WHERE customer\_city='Delhi';



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

SELECT c.customer\_number,c.firstname,a.account\_number FROM account\_master a join customer\_master c ON c.customer\_number=a.customer\_number WHERE day(a.account\_opening\_date)>'15' ORDER BY c.customer\_number,a.account\_number;



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

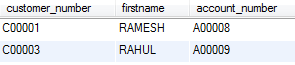
SELECT c.customer\_number,c.firstname,a.account\_number

FROMaccount\_master a JOIN customer\_master c

ON c.customer\_number=a.customer\_number

WHERE a.account\_status='Terminated'

ORDER BY c.customer\_number,a.account\_number;



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

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

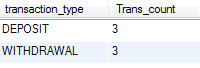
FROM account\_master am JOIN transaction\_details td

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

WHERE customer\_number like '%001'

GROUP BY transaction\_type

ORDER BY transaction\_type;



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

SELECT count(customer\_number) Count\_Customer FROM customer\_master

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

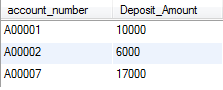


**7. Write a query to display account number and total amount deposited by each account holder (Including the opening balance). Give the total amount deposited an alias name of Deposit\_Amount. Display the records in sorted order based on account number.**

SELECT a.account\_number,a.opening\_balance+sum(t.transaction\_amount)

FROM account\_master a JOIN transaction\_details t ON a.account\_number=t.account\_number

WHERE t.transaction\_type='Deposit' GROUP BY t.account\_number;



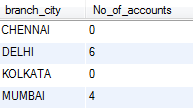
**8. Write a query to display the number of accounts opened in each city .The Query should display Branch City and number of accounts as No\_of\_Accounts.For the branch city where we don’t have any accounts opened display 0. Display the records in sorted order based on branch city.**

SELECT branch.branch\_city, count(account.account\_number) No\_of\_Accounts

FROM branch\_master LEFT JOIN account\_master

ON account.branch\_id=branch.branch\_id

GROUP BY branch.branch\_city ORDER BY branch\_city;

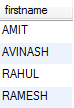


**9. Write a query to display the firstname of the customers who have more than 1 account. Display the records in sorted order based on firstname.**

SELECT c.firstname FROM

customer\_master c JOIN account\_master a ON a.customer\_number=c.customer\_number

GROUP BY a.customer\_number HAVING count(a.account\_number)>1;



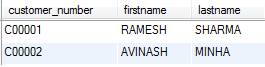
**10. Write a query to display the customer number, customer firstname, customer lastname who has taken loan from more than 1 branch. Display the records sorted in order based on customer number.**

SELECT c.customer\_number,c.firstname,c.lastname FROM

customer\_master c JOIN loan\_details l ON c.customer\_number=l.customer\_number

GROUP BY l.customer\_number HAVING count(l.branch\_id)>1

ORDER BY c.customer\_number;



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

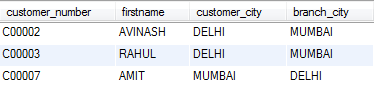
SELECT c.customer\_number,c.firstname,c.customer\_city,b.branch\_city FROM

Customer\_master c JOIN Account\_master a ON c.customer\_number=a.customer\_number

JOIN Branch\_master b ON b.branch\_id=a.branch\_id

WHERE b.branch\_city<>c.customer\_city

ORDER BY c.customer\_number;



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

SELECT count(c.customer\_number)Count FROM customer\_master c JOIN loan\_details l

ON c.customer\_number=l.customer\_number

WHERE c.customer\_number NOT IN (SELECT customer\_number FROM account\_master);



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

SELECT account\_number FROM transaction\_details

GROUP BY account\_number

HAVING count(transaction\_number)>=ALL

(SELECT count(transaction\_number) FROM transaction\_details

GROUP BY account\_number) ORDER BY account\_number;



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

SELECT b.branch\_name,b.branch\_city FROM

Branch\_master b JOIN account a ON a.branch\_id=b.branch\_id

GROUP BY a.branch\_id HAVING count(a.customer\_number)>=ALL

(SELECT count(customer\_number) FROM

Account\_master GROUP BY branch\_id)

ORDER BY b.branch\_name;



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

SELECT td.account\_number,

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

+(SELECT opening\_balance

FROM account\_master where account\_number=td.account\_number) Deposit,

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

FROM transaction\_details td

GROUP BY td.account\_number

HAVING Withdrawal > Deposit

ORDER BY td.account\_number;

**(or)**

SELECT ifnull(t1.account\_number,t2.account\_number) account\_number,

t2.d Deposit,ifnull(t1.w,0) Withdrawal FROM

(SELECT account\_number,transaction\_type,sum(transaction\_amount) w from transaction\_details

WHERE transaction\_type='Withdrawal' GROUP BY account\_number) t1

RIGHT JOIN

(SELECT a.account\_number,a.opening\_balance+sum(t.transaction\_amount) d

FROM account\_master a JOIN transaction\_details t ON a.account\_number=t.account\_number

WHERE t.transaction\_type='Deposit'GROUP BY t.account\_number) t2

ON t1.account\_number=t2.account\_number

WHERE ifnull(t1.w,0)>t2.d

ORDER BY account\_number;



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

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

THEN transaction\_amount END)) -

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

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

from account\_master where account\_number like '%001'),(SUM(CASE WHEN transaction\_type='Deposit'

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

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

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

**(or)**

SELECT ifnull(t1.account\_number,t2.account\_number) account\_number,

t2.d-ifnull(t1.w,0) Balance\_Amount FROM

(SELECT account\_number,transaction\_type,sum(transaction\_amount) w from transaction\_details

WHERE transaction\_type='Withdrawal' GROUP BY account\_number) t1

RIGHT JOIN

(SELECT a.account\_number,a.opening\_balance+sum(t.transaction\_amount) d

FROM account a JOIN transaction\_details t ON a.account\_number=t.account\_number

WHERE t.transaction\_type='Deposit'GROUP BY t.account\_number) t2

ON t1.account\_number=t2.account\_number

WHERE ifnull(t1.account\_number,t2.account\_number) LIKE '%001'

ORDER BY account\_number;



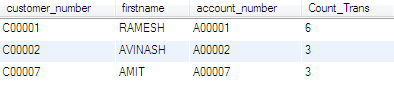
17. **Display the customer number, customer's first name, account number and number of transactions being made by the customers from each account. Give the alias name for number of transactions as Count\_Trans. Display the records sorted in ascending order based on customer number and then by account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

SELECT c.customer\_number,c.firstname,t.account\_number, count(t.account\_number) Count\_Trans

FROM transaction\_details t JOIN account\_master a ON a.account\_number=t.account\_number

JOIN customer c ON c.customer\_number=a.customer\_number

GROUP BY t.account\_number ORDER BY c.customer\_number, a.account\_number;



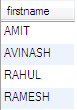
18. **Write a query to display the customer’s firstname who have multiple accounts (atleast 2 accounts). Display the records sorted in ascending order based on customer's firstname.**

SELECT c.firstname FROM

Customer\_master c JOIN account\_master a ON c.customer\_number=a.customer\_number

GROUP BY a.customer\_number HAVING count(a.account\_number)>1

ORDER BY c.firstname;



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

SELECT ld.customer\_number, firstname, lastname

FROM customer\_master cm JOIN loan\_details ld

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

GROUP BY customer\_number

HAVING count(branch\_id)>=2 AND sum(loan\_amount)>=

ALL(SELECT sum(loan\_amount) FROM loan GROUP BY customer\_number);

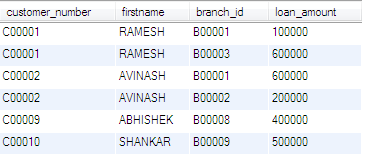


20. **Write a query to display the customer’s number, customer’s firstname, branch id and loan amount for people who have taken loans. Display the records sorted in ascending order based on customer number and then by branch id and then by loan amount.**

SELECT c.customer\_number,c.firstname,l.branch\_id,l.loan\_amount FROM

Customer\_master c JOIN loan\_details l ON c.customer\_number=l.customer\_number

ORDER BY c.customer\_number,l.branch\_id,l.loan\_amount;

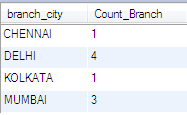


21. **Write a query to display city name and count of branches in that city. Give the count of branches an alias name of Count\_Branch. Display the records sorted in ascending order based on city name.**

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

Branch\_master GROUP BY branch\_city

ORDER BY branch\_city;

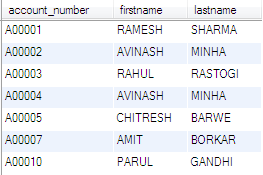


22. **Write a query to display account id, customer’s firstname, customer’s lastname for the customer’s whose account is Active. Display the records sorted in ascending order based on account id /account number.**

SELECT a.account\_number,c.firstname,c.lastname FROM

Customer\_master c JOIN account\_master a ON c.customer\_number=a.customer\_number and a.account\_status='Active'

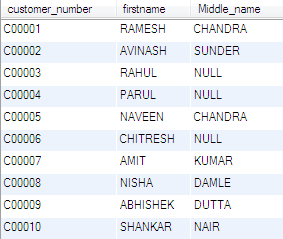
ORDER BY a.account\_number;



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

SELECT customer\_number,firstname,ifnull(middlename,lastname) Middle\_name FROM

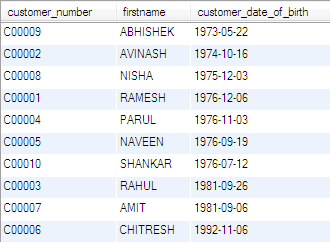
Customer\_master ORDER BY customer\_number;



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

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

Customer\_master ORDER BY year(customer\_date\_of\_birth),customer\_number;



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

SELECT c.firstname,c.customer\_city,a.account\_number FROM

Customer\_master c JOIN account\_master a ON a.customer\_number=c.customer\_number

WHERE c.occupation NOT IN ('Service','Student','Business')

ORDER BY c.firstname,a.account\_number;



AIRLINES

create database flight;

use flight;

CREATE TABLE**air\_credit\_card\_details**

(

profile\_id VARCHAR(10) NOT NULL,

card\_number BIGINT,

card\_type VARCHAR(45),

expiration\_month INT,

expiration\_year INT

);

CREATE TABLE**air\_passenger\_profile**

(

profile\_id VARCHAR(10) NOT NULL ,

password VARCHAR(45) NULL ,

first\_name VARCHAR(45) NULL ,

last\_name VARCHAR(45) NULL ,

address VARCHAR(45) NULL ,

mobile\_number BIGINT NULL ,

email\_id VARCHAR(45) NULL

);

CREATE TABLE**air\_ticket\_info**

(

ticket\_id VARCHAR(45) NOT NULL ,

profile\_id VARCHAR(10) NULL ,

flight\_id VARCHAR(45) NULL ,

flight\_departure\_date DATE NULL ,

status VARCHAR(45) NULL

);

CREATE TABLE**air\_flight\_details**

(

flight\_id VARCHAR(45) NOT NULL ,

flight\_departure\_date DATE NULL ,

price DECIMAL(10,2) NULL ,

available\_seats INT NULL

);

CREATE TABLE**air\_flight**

(

flight\_id VARCHAR(45) NOT NULL ,

airline\_id VARCHAR(45) NULL ,

airline\_name VARCHAR(45) NULL ,

from\_location VARCHAR(45) NULL ,

to\_location VARCHAR(45) NULL ,

departure\_time TIME NULL ,

arrival\_time TIME NULL ,

duration TIME NULL ,

total\_seats INT NULL

);

INSERT INTO air\_credit\_card\_details VALUES

(1, 622098761234, 'debit', 5, 2013),

(2, 652362563625, 'credit', 1, 2013),

(1, 765432345678, 'credit', 2, 2013),

(3, 654378561234, 'debit', 6, 2013),

(4, 625417895623, 'debit', 2, 2013),

(5, 865478956325, 'debit', 3, 2013),

(6, 789563521457, 'credit', 4, 2013),

(2, 543267895432, 'credit', 8, 2013),

(1, 256369856321, 'debit', 1, 2013);

INSERT INTO air\_flight VALUES

(3173, 'MH370', 'abc', 'hyderabad', 'chennai', '06:30:00', '07:15:00', '0:45:00', 100),

(3178, 'MH17', 'def', 'chennai', 'hyderabad', '08:00:00', '09:00:00', '1:00:00', 200),

(3172, 'AR342', 'fgh', 'kolkata', 'chennai', '11:30:00', '13:00:00', '1:30:00', 100),

(3071, 'JT564', 'jkl', 'chennai', 'delhi', '08:00:00', '10:00:00', '2:00:00', 100),

(3170, 'DT345', 'xyz', 'delhi', 'kolkata', '21:00:00', '22:30:00', '1:30:00', 100),

(3175, 'MJ654', 'abc', 'chennai', 'hyderabad', '15:00:00', '16:00:00', '1:00:00', 200),

(3176, 'MH370', 'def', 'kochi', 'chennai', '18:00:00', '19:05:00', '1:05:00', 100),

(3177, 'MH45', 'fgh', 'delhi', 'kochi', '19:00:00', '21:00:00', '2:00:00', 200),

(3174, 'MH321', 'xyz', 'kolkata', 'delhi', '0:00:00', '2:00:00', '2:00:00', 100),

(3179, 'JT435', 'abc', 'chennai', 'kolkata', '14:00:00', '15:00:00', '1:00:00', 100),

(3180, 'JT456', 'ijk', 'kolkata', 'kochi', '5:00:00', '5:45:00', '0:45:00', 200);

INSERT INTO air\_flight\_details VALUES

(3170, '2013-02-14', 1000, 10),

(3171, '2013-03-15', 5000, 0),

(3172, '2013-02-05', 3000, 32),

(3173, '2013-04-07', 2000, 12),

(3174, '2013-04-05', 3800, 3),

(3175, '2013-05-25', 3500, 10),

(3176, '2013-03-14', 8000, 2),

(3177, '2013-06-15', 1500, 0),

(3178, '2013-05-06', 3000, 5),

(3179, '2013-04-03', 4000, 15),

(3180, '2013-04-02', 3000, 14);

INSERT INTO air\_ticket\_info VALUES

(1, 1, 3178, '2013-05-06', 'delayed'),

(2, 5, 3179, '2013-04-03', 'on time'),

(2, 4, 3180, '2013-04-02', 'on time'),

(1, 2, 3177, '2013-06-15', 'on time'),

(1, 3, 3176, '2013-03-14', 'on time'),

(3, 1, 3171, '2013-03-15', 'on time'),

(4, 4, 3172, '2013-02-06', 'delayed'),

(5, 2, 3178, '2013-06-05', 'on time'),

(4, 3, 3171, '2013-03-15', 'on time'),

(5, 1, 3175, '2013-05-25', 'on time'),

(6, 3, 3177, '2013-06-15', 'on time');

INSERT INTO air\_passenger\_profile VALUES

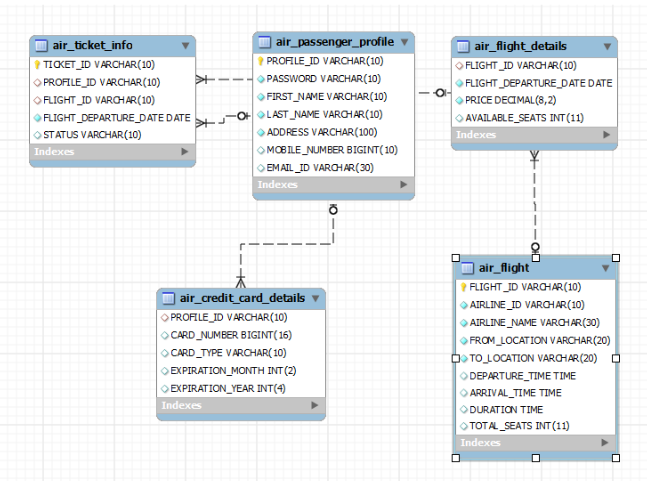
(1, 'godbless', 'John', 'Stuart', 'Street 21, Near Bus Stop-Hyderabad-432126', 9865263251, 'john@gmail.com'),

(2, 'heyyaa', 'Robert', 'Clive', 'Sector 3, Technopolis-Kolkata-700102', 9733015875, 'robert@yahoo.com'),

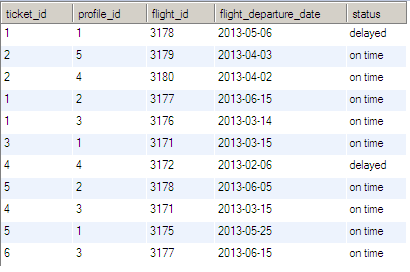
(3, 'hello123', 'Raj', 'Sharma', 'House No. 3, Anna Nagar-Kochi-452314', 9775470232, 'raj3452@hotmail.com'),

(4, 'yesboss', 'Sanjay', 'Mittal', '21 Cauunaught Place-Delhi-144985', 9856856321, 'sanjay@yahoo.com'),

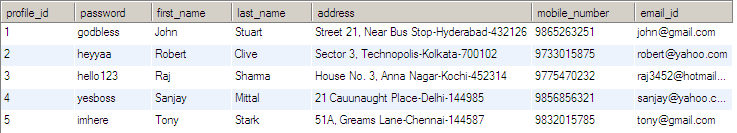
(5, 'imhere', 'Tony', 'Stark', '51A, Greams Lane-Chennai-144587', 9832015785, 'tony@gmail.com');



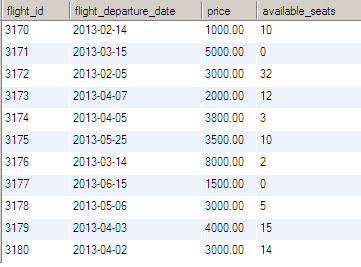
**AIR TICKET INFO**



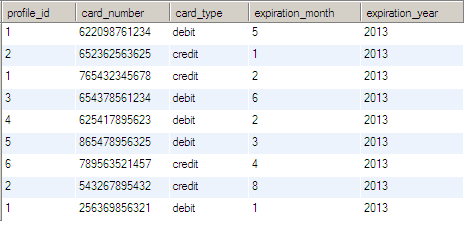
**AIR PASSENGER DETAILS**



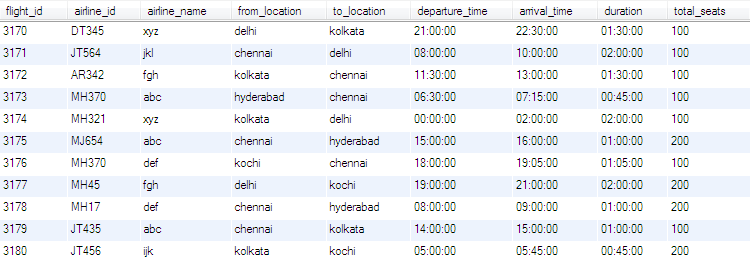
**AIR FLIGHT DETAILS**



**AIR CREDIT CARD DETAILS**



**AIR FLIGHT**



QUERIES

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

SELECT f.flight\_id,f.from\_location,f.to\_location,

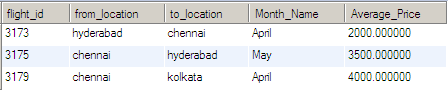
monthname(af.flight\_departure\_date) Month\_Name,

AVG(price) Average\_Price FROM air\_flight f JOIN air\_flight\_details af

ON f.flight\_id = af.flight\_id WHERE f.airline\_name = 'abc'

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

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



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

SELECT f.from\_location,f.to\_location,

monthname(af.flight\_departure\_date) Month\_Name,

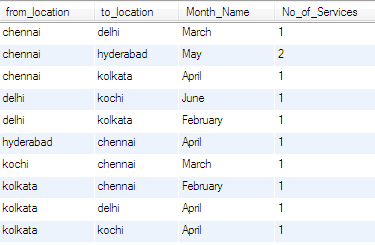
count(af.flight\_departure\_date) No\_of\_Services

FROM air\_flight f JOIN air\_flight\_details af

ON f.flight\_id = af.flight\_id

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

ORDER BY f.from\_location,f.to\_Location,Month\_Name;



3.**Write a query to display the customer(s) who has/have booked least number of tickets in ABC Airlines. The Query should display profile\_id, customer’s first\_name, Address and Number of tickets booked as “No\_of\_Tickets”Display the records sorted in ascending order based on customer's first name.**

SELECT ap.profile\_id,ap.first\_name,ap.address,count(ati.ticket\_id) No\_of\_Tickets FROM

air\_passenger\_profile ap JOIN air\_ticket\_info ati ON ap.profile\_id=ati.profile\_id

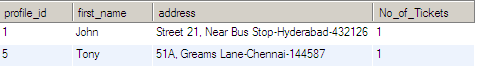
JOIN air\_flight af ON af.flight\_id=ati.flight\_id and af.airline\_name='abc'

GROUP BY ap.profile\_id,ap.first\_name,ap.address HAVING count(ati.ticket\_id)<=ALL

(SELECT count(ticket\_id)

FROM air\_ticket\_info GROUP BY profile\_id)

ORDER BY ap.first\_name;



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

SELECT ap.profile\_id,ap.first\_name,ap.last\_name,af.flight\_id,ati.flight\_departure\_date,

count(ati.profile\_id) No\_of\_Tickets FROM

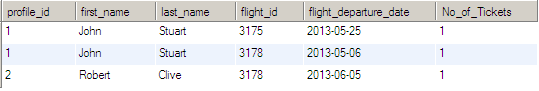
air\_ticket\_info ati JOIN air\_passenger\_profile ap ON ap.profile\_id=ati.profile\_id

JOIN air\_flight af ON af.flight\_id=ati.flight\_id

WHERE af.from\_location='Chennai' and af.to\_location='Hyderabad'

GROUP BY ati.flight\_id,ati.profile\_id

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



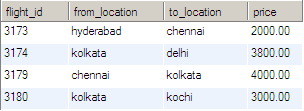
5. **Write a query to display flight id,from location, to location and ticket price of flights whose departure is in the month of april.Display the records sorted in ascending order based on flight id and then by from location.**

SELECT af.flight\_id,af.from\_location,af.to\_location,afd.price FROM

air\_flight af JOIN air\_flight\_details afd ON af.flight\_id=afd.flight\_id

and month(afd.flight\_departure\_date)='04'

ORDER BY af.flight\_id,af.from\_location;



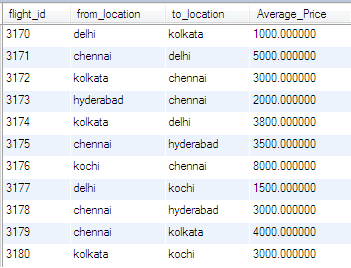
**6. Write a query to display the average cost of the tickets in each flight on all scheduled dates. The query should display flight\_id, from\_location, to\_location and Average price as “Price”. Display the records sorted in ascending order based on flight id and then by from\_location and then by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

SELECT af.flight\_id,af.from\_location,af.to\_location,avg(afd.price) Average\_Price FROM

air\_flight af JOIN air\_flight\_details afd ON af.flight\_id=afd.flight\_id

GROUP BY af.flight\_id

ORDER BY af.flight\_id,af.from\_location,af.to\_location;



**7. Write a query to display the customers who have booked tickets from Chennai to Hyderabad. The query should display profile\_id, customer\_name (combine first\_name & last\_name with comma in b/w), address of the customer. Give an alias to the name as customer\_name.Hint: Query should fetch unique customers irrespective of multiple tickets booked.Display the records sorted in ascending order based on profile id.**

SELECT ap.profile\_id,concat(ap.first\_name,',',ap.last\_name) customer\_name,ap.address FROM

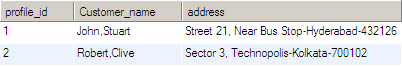
air\_passenger\_profile ap JOIN air\_ticket\_info ati ON ap.profile\_id=ati.profile\_id

JOIN air\_flight af ON af.flight\_id=ati.flight\_id

WHERE af.from\_location='Chennai' and af.to\_location='Hyderabad'

GROUP BY ati.profile\_id

ORDER BY ap.profile\_id;



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

SELECT profile\_id FROM air\_ticket\_info

group by profile\_id

having count(ticket\_id)>=all(select count(ticket\_id)

from air\_ticket\_info

group by profile\_id) order by profile\_id;



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

SELECT f.flight\_id,f.from\_location,f.to\_location,COUNT(t.ticket\_id) AS No\_of\_Tickets

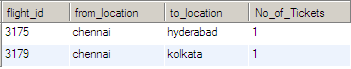
FROM air\_ticket\_info t JOIN air\_flight f

ON f.flight\_id = t.flight\_id where AIRLINE\_NAME = 'abc' GROUP by

f.flight\_id,f.from\_location,f.to\_location

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

ORDER by f.flight\_id;

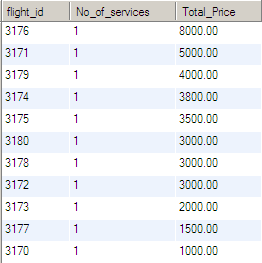


**10. Write a query to display the no of services offered by each flight and the total price of the services. The Query should display flight\_id, number of services as “No\_of\_Services” and the cost as “Total\_Price” in the same order. Order the result by Total Price in descending order and then by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

SELECT flight\_id,count(flight\_departure\_date) No\_of\_services,sum(price) Total\_Price FROM

air\_flight\_details GROUP BY flight\_id

ORDER BY Total\_price DESC,flight\_id DESC;

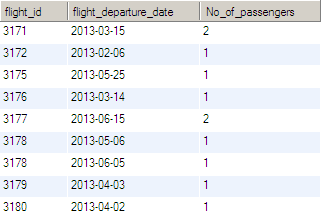


**11. Write a query to display the number of passengers who have travelled in each flight in each scheduled date. The Query should display flight\_id, flight\_departure\_date and the number of passengers as “No\_of\_Passengers” in the same order.Display the records sorted in ascending order based on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

SELECT flight\_id,flight\_departure\_date,count(ticket\_id) No\_of\_passengers FROM

air\_ticket\_info GROUP BY flight\_id,flight\_departure\_date

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



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

SELECT profile\_id FROM air\_ticket\_info

GROUP BY profile\_id HAVING count(ticket\_id)<=ALL

(SELECT count(ticket\_id) FROM air\_ticket\_info GROUP BY profile\_id)

ORDER BY profile\_id;



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

SELECT DISTINCT ap.profile\_id,ap.first\_name,ap.mobile\_number,ap.email\_id FROM

air\_passenger\_profile ap JOIN air\_ticket\_info ati ON ap.profile\_id=ati.profile\_id

JOIN air\_flight af ON ati.flight\_id=af.flight\_id

WHERE af.from\_location='Hyderabad' and af.to\_location='Chennai'

ORDER BY profile\_id;



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

SELECT DISTINCT ap.profile\_id,ap.first\_name,ap.last\_name,ati.flight\_id,ati.flight\_departure\_date,

af.departure\_time,af.arrival\_time,

addtime(af.departure\_time,'01:00:00') Delayed\_Deparuture\_Time,

addtime(af.arrival\_time,'01:00:00') Delayed\_Arrival\_Time FROM

air\_passenger\_profile ap JOIN air\_ticket\_info ati ON ap.profile\_id=ati.profile\_id

JOIN air\_flight af ON af.flight\_id=ati.flight\_id

WHERE af.from\_location='Chennai' and af.to\_location='Hyderabad'

and ati.flight\_departure\_date='2013-05-06'

ORDER BY profile\_id;



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

SELECT ap.profile\_id,ap.first\_name,

substring\_index(substring\_index(ap.address,'-',2),'-',-1) Base\_Location,

count(ati.ticket\_id) No\_of\_Tickets FROM

air\_passenger\_profile ap JOIN air\_ticket\_info ati ON ati.profile\_id=ap.profile\_id

WHERE ap.address LIKE '%Kochi%'

ORDER BY ap.first\_name;



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

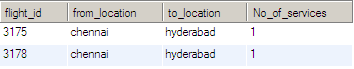
SELECT af.flight\_id,af.from\_location,af.to\_location,count(afd.flight\_departure\_date) No\_of\_services FROM

air\_flight af JOIN air\_flight\_details afd ON af.flight\_id=afd.flight\_id

WHERE month(flight\_departure\_date)='05'

GROUP BY af.flight\_id,af.from\_location,af.to\_location

ORDER BY af.flight\_id;



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

SELECT profile\_id, last\_name, mobile\_number, email\_id

FROM air\_passenger\_profile

WHERE address LIKE '%Chennai%'

ORDER BY profile\_id;



**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';



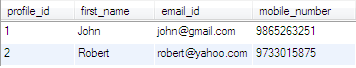
19. **Write a query to display unique profile id,first name , email id and contact number of passenger(s) who travelled on flight with id 3178. Display the records sorted in ascending order based on first name.**

SELECT DISTINCT ap.profile\_id,ap.first\_name,ap.email\_id,ap.mobile\_number FROM

air\_passenger\_profile ap JOIN air\_ticket\_info ati ON ap.profile\_id=ati.profile\_id

WHERE ati.flight\_id='3178'

ORDER BY ap.first\_name;



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

SELECT flight\_id,flight\_departure\_date,

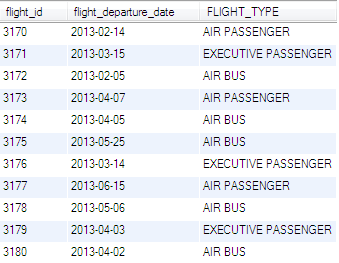
case when price<3000 then 'AIR PASSENGER'

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

when price>=4000 then 'EXECUTIVE PASSENGER'

end FLIGHT\_TYPE FROM air\_flight\_details

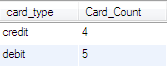
ORDER BY flight\_id,flight\_departure\_date;



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

SELECT card\_type, count(card\_type) Card\_Count FROM air\_credit\_card\_details

GROUP BY card\_type ORDER BY card\_type;



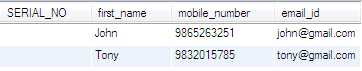
22.**Write a Query to display serial no, first name, mobile number, email id of all the passengers who holds email address from gmail.com.The Serial No will be the last three digits of profile ID.Hint: Use SERIAL\_NO as Alias name for serial number.Display the records sorted in ascending order based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

SELECT substring(profile\_id,-3) SERIAL\_NO,first\_name,mobile\_number,email\_id FROM

air\_passenger\_profile

WHERE email\_id LIKE '%@gmail.com'

ORDER BY first\_name;



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

SELECT afd.flight\_id,af.from\_location,af.to\_location,count(afd.flight\_id) No\_of\_Services

FROM air\_flight\_details afd JOIN air\_flight af ON af.flight\_id=afd.flight\_id

WHERE monthname(afd.flight\_departure\_date)='May'

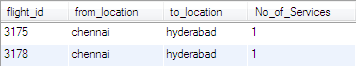
GROUP BY afd.flight\_departure\_date HAVING count(afd.flight\_id) <=

ALL(SELECT count(flight\_id) FROM air\_flight\_details

WHERE monthname(flight\_departure\_date)='May'

GROUP BY flight\_departure\_date)

ORDER BY flight\_id;



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

SELECT flight\_id,from\_location,to\_location,Departure\_Time,

CASE

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

THEN 'Morning'

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

THEN 'AfterNoon'

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

THEN 'Evening'

WHEN departure\_time='00:00:00'

THEN 'Evening'

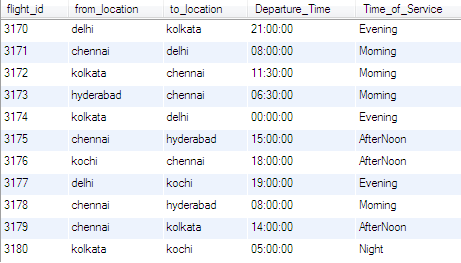
WHEN departure\_time BETWEEN ('00:00:01') AND ('05:00:00')

THEN 'Night'

END Time\_of\_Service

FROM air\_flight

order by flight\_id;

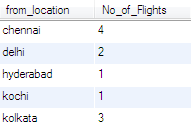


25. **Write a query to display the number of flights flying from each location. The Query should display the from location and the number of flights to other locations as “No\_of\_Flights”. Hint: Get the distinct from location and to location.Display the records sorted in ascending order based on from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

SELECT from\_location,count(flight\_id) No\_of\_Flights FROM

air\_flight GROUP BY from\_location

ORDER BY from\_location;



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

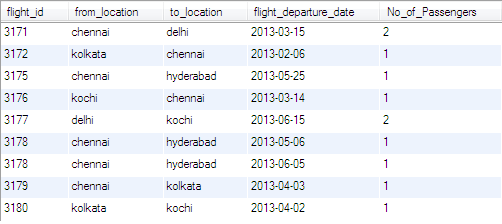
SELECT af.flight\_id,af.from\_location,af.to\_location,ati.flight\_departure\_date,

count(ati.ticket\_id) No\_of\_Passengers FROM

air\_flight af JOIN air\_ticket\_info ati ON af.flight\_id=ati.flight\_id

GROUP BY af.flight\_id,af.from\_location,af.to\_location,ati.flight\_departure\_date

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



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

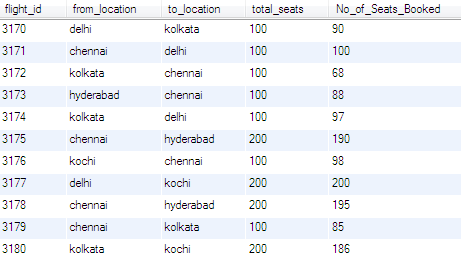
SELECT af.flight\_id,af.from\_location,af.to\_location,af.total\_seats,

(af.total\_seats-afd.available\_seats) No\_of\_Seats\_Booked FROM

air\_flight\_details afd JOIN air\_flight af ON afd.flight\_id=af.flight\_id

WHERE (af.total\_seats-afd.available\_seats)>(af.total\_seats\*0.1)

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

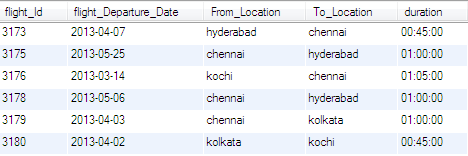


28. **Write a query to display the Flight\_Id, Flight\_Departure\_Date, From\_Location,To\_Location and Duration of all flights which has duration of travel less than 1 Hour, 10 Minutes.**

SELECT af.flight\_Id,afd.flight\_Departure\_Date,af.From\_Location,af.To\_Location,af.duration

FROM air\_flight af JOIN air\_flight\_details afd ON af.flight\_id=afd.flight\_id

WHERE af.duration<'01:10:00';



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

SELECT afd.flight\_id,af.from\_location,af.to\_location,

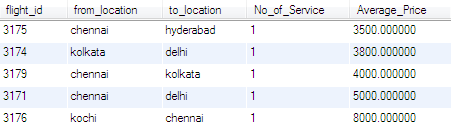
count(afd.flight\_departure\_date) No\_of\_Service, avg(price) Average\_Price

FROM air\_flight af JOIN air\_flight\_details afd ON af.flight\_id=afd.flight\_id

GROUP BY af.flight\_id,af.from\_location,af.to\_location

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

ORDER BY average\_price;



MOVIE

CREATE DATABASE video;USE video;

Create table CUSTOMER\_MASTER

(CUSTOMER\_ID Varchar(10),CUSTOMER\_NAME Varchar(30) NOT NULL,CONTACT\_NO BIGINT(10),CONTACT\_ADD Varchar(20),DATE\_OF\_REGISTRATION Date NOT NULL,AGE Varchar(15)NOT NULL,Constraint MT\_cts1 PRIMARY KEY(CUSTOMER\_ID));

Create table LIBRARY\_CARD\_MASTER

(CARD\_ID Varchar(10),DESCRIPTION Varchar(30) NOT NULL,AMOUNT BIGINT(50),NUMBER\_OF\_YEARS bigint(10) NOT NULL,Constraint MT\_cts2 PRIMARY KEY(CARD\_ID));

Create table MOVIES\_MASTER

(MOVIE\_ID Varchar(10), MOVIE\_NAME Varchar(50) NOT NULL,RELEASE\_DATE Varchar(30) NOT NULL,LANGUAGE Varchar(30),RATING int(2),DURATION VARCHAR(10) NOT NULL, MOVIE\_TYPE Varchar(3),MOVIE\_CATEGORY VARCHAR(20) NOT NULL,DIRECTOR VARCHAR(20) NOT NULL,

LEAD\_ROLE\_1 Varchar(3) NOT NULL,LEAD\_ROLE\_2 VARCHAR(4) NOT NULL,RENT\_COST BIGINT(10),Constraint MT\_cts4 PRIMARY KEY(MOVIE\_ID));

Create table CUSTOMER\_CARD\_DETAILS

(CUSTOMER\_ID Varchar(10),CARD\_ID VARCHAR(10),ISSUE\_DATE DATE NOT NULL,Constraint MT\_cts3 PRIMARY KEY(CUSTOMER\_ID),Constraint MT\_CTS41 FOREIGN KEY(CUSTOMER\_ID) References CUSTOMER\_MASTER(CUSTOMER\_ID),Constraint MT\_CTS42 FOREIGN KEY(CARD\_ID) References LIBRARY\_CARD\_MASTER(CARD\_ID));

Create table CUSTOMER\_ISSUE\_DETAILS

(ISSUE\_ID Varchar(10) NOT NULL,CUSTOMER\_ID Varchar(10) NOT NULL,MOVIE\_ID VARCHAR(10), ISSUE\_DATE Date NOT NULL,RETURN\_DATE Date NOT NULL,

ACTUAL\_DATE\_RETURN Date NOT NULL,Constraint MT\_cts5 PRIMARY KEY(ISSUE\_ID),Constraint MT\_Mem FOREIGN KEY(CUSTOMER\_ID) References CUSTOMER\_MASTER(CUSTOMER\_ID), Constraint MT\_Mem1 FOREIGN KEY(MOVIE\_ID) References MOVIES\_MASTER(MOVIE\_ID));

Insert into CUSTOMER\_MASTER Values('CUS001', 'AMIT', 9876543210,'ADD1', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS002', 'ABDHUL', 8765432109,'ADD2', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS003', 'GAYAN', 7654321098,'ADD3', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS004', 'RADHA', 6543210987,'ADD4', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS005', 'GURU', NULL,'ADD5', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS006', 'MOHAN', 4321098765,'ADD6', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS007', 'NAME7', 3210987654,'ADD7', '2012-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS008', 'NAME8', 2109876543,'ADD8', '2013-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS009', 'NAME9', NULL,'ADD9', '2013-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS010', 'NAM10', 9934567890,'ADD10', '2013-02-12', '21');

Insert into CUSTOMER\_MASTER Values('CUS011', 'NAM11', 9875678910,'ADD11', '2013-02-12', '21');

Insert into LIBRARY\_CARD\_MASTER Values('CR001', 'Silver', 200, 5);

Insert into LIBRARY\_CARD\_MASTER Values('CR002', 'Gold', 400, 9);

Insert into LIBRARY\_CARD\_MASTER Values('CR003', 'Platinum', 600, 8);

Insert into LIBRARY\_CARD\_MASTER Values('CR004', 'VISA', 800, 7);

Insert into LIBRARY\_CARD\_MASTER Values('CR005', 'CREDIT', 1200, 6);

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

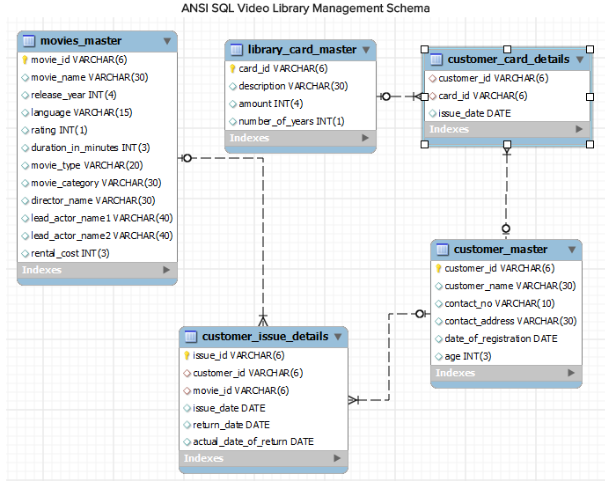
Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS015', 'CUS004', 'MV006', '2012-04-27', '2012-05-07','2012-05-25');

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

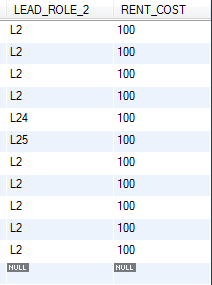
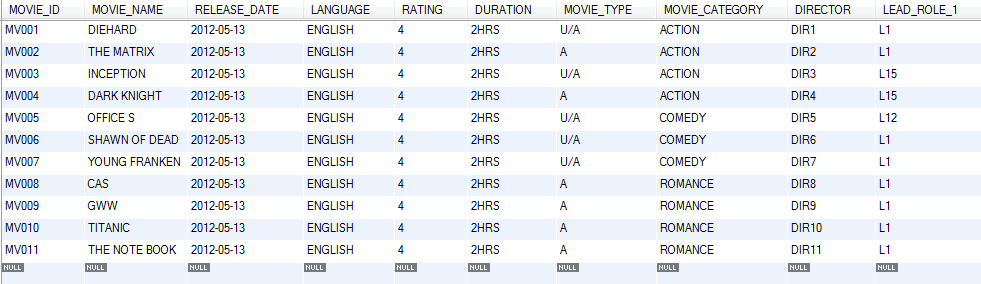
Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS017', 'CUS004', 'MV001', '2012-04-29', '2012-05-07','2012-05-25');

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

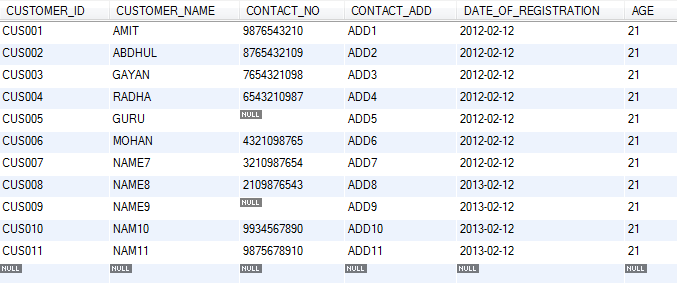
Insert into CUSTOMER\_ISSUE\_DETAILS Values ('IS019', 'CUS011', 'MV009', '2012-04-27', '2012-05-07','2012-05-25');



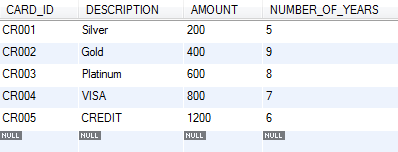
MOVIE MASTER



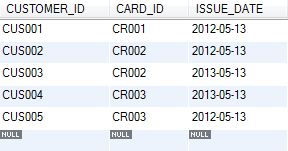
CUSTOMER MASTER



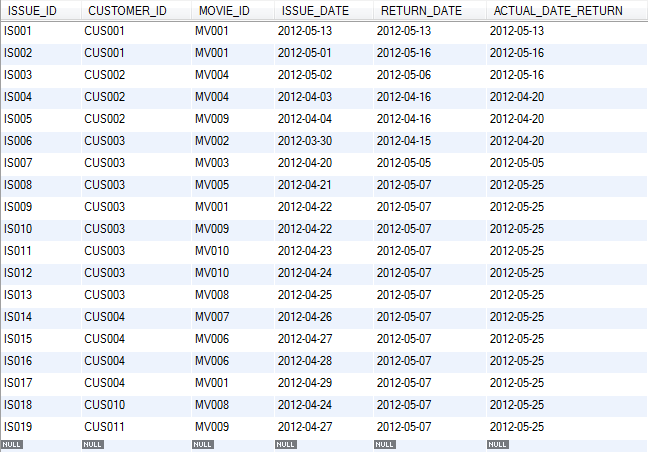
LIBRARY CARD MASTER



CUSTOMER CARD DETAILS



CUSTOMER ISSUE DETAILS



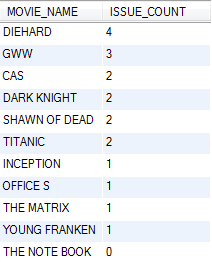
**1.Write a query to display movie names and number of times that movie is issued to customers. Incase movies are never issued to customers display number of times as 0. Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order). The Alias name for the number of movies issued is ISSUE\_COUNT.**

SELECT m.MOVIE\_NAME,count(ISSUE\_ID) ISSUE\_COUNT FROM

movies\_master m LEFT JOIN customer\_issue\_details c ON m.MOVIE\_ID=c.MOVIE\_ID

GROUP BY m.movie\_name

ORDER BY ISSUE\_COUNT DESC,MOVIE\_NAME;



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

SELECT CUSTOMER\_ID,CUSTOMER\_NAME,AGE,ifnull(

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

substring(contact\_no,4,3),'-',substring(contact\_no,7)),'N/A') CONTACT\_ISD

FROM customer\_master WHERE age>25 and year(date\_of\_registration)='2012'

ORDER BY age,CUSTOMER\_NAME;

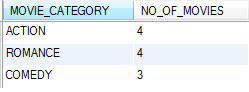


**3.Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.Hint: Use NO\_OF\_MOVIES as alias name for number of movies.**

SELECT MOVIE\_CATEGORY,count(MOVIE\_ID) NO\_OF\_MOVIES FROM

movies\_master GROUP BY MOVIE\_CATEGORY

ORDER BY NO\_OF\_MOVIES DESC,MOVIE\_CATEGORY;



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

SELECT count(c.customer\_id) CUSTOMER\_COUNT FROM

library\_card\_master l JOIN customer\_card\_details c ON l.CARD\_ID=c.CARD\_ID

WHERE description='Gold';



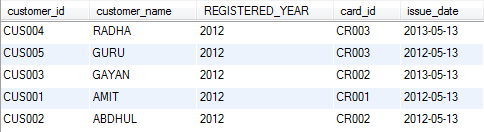
**5.Write a query to display the customer id, customer name, year of registration,library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order. Use REGISTERED\_YEAR as alias name for year of registration.**

SELECT c.customer\_id,c.customer\_name,

year(c.DATE\_OF\_REGISTRATION) REGISTERED\_YEAR,cd.card\_id,cd.issue\_date FROM

customer\_master c JOIN customer\_card\_details cd ON c.customer\_id=cd.customer\_id

ORDER BY CUSTOMER\_NAME DESC;



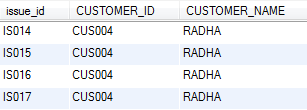
**6.Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer order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

SELECT ci.issue\_id,ci.CUSTOMER\_ID,c.CUSTOMER\_NAME FROM

customer\_master c JOIN customer\_issue\_details ci ON c.customer\_id=ci.customer\_id

WHERE customer\_name LIKE 'R%' and ci.actual\_date\_return>ci.return\_date

ORDER BY customer\_name;



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

SELECT c.CUSTOMER\_ID,c.CUSTOMER\_NAME,l.card\_id,l.DESCRIPTION,

concat('$',round(amount/52.42)) AMOUNT\_DOLLAR FROM

customer\_master c JOIN customer\_issue\_details ci ON c.customer\_id=ci.customer\_id

JOIN customer\_card\_details cc ON cc.customer\_id=c.customer\_id

JOIN library\_card\_master l ON cc.card\_id=l.card\_id

WHERE c.DATE\_OF\_REGISTRATION=ci.issue\_date

ORDER BY customer\_name;



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

SELECT CUSTOMER\_ID,upper(CUSTOMER\_NAME) CUSTOMER\_NAME,contact\_no,contact\_add FROM

customer\_master WHERE contact\_add LIKE '%Nagar' and

customer\_id NOT IN (SELECT customer\_id FROM customer\_card\_details)

and customer\_id IN (SELECT customer\_id FROM customer\_issue\_details)

ORDER BY CUSTOMER\_NAME;



**9.Write a query to display the movie id, movie name,release year,director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.**

SELECT movie\_id,movie\_name,release\_date,director FROM movies\_master

WHERE lead\_role\_1 IN(SELECT lead\_role\_1 FROM

(SELECT lead\_role\_1,count(movie\_id)ct FROM movies\_master

GROUP BY lead\_role\_1)t WHERE t.ct>=ALL(SELECT count(movie\_id)

FROM movies\_master GROUP BY lead\_role\_1)) ORDER BY movie\_name;

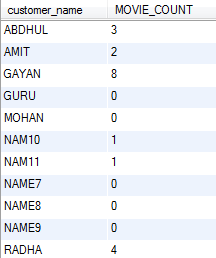


**10.Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0. <br>Hint: Use MOVIE\_COUNT as alias name for number of movies issued.**

SELECT c.customer\_name,count(ci.movie\_id) MOVIE\_COUNT FROM

customer\_master c LEFT JOIN customer\_issue\_details ci ON c.customer\_id=ci.customer\_id

GROUP BY c.customer\_id ORDER BY c.customer\_name;



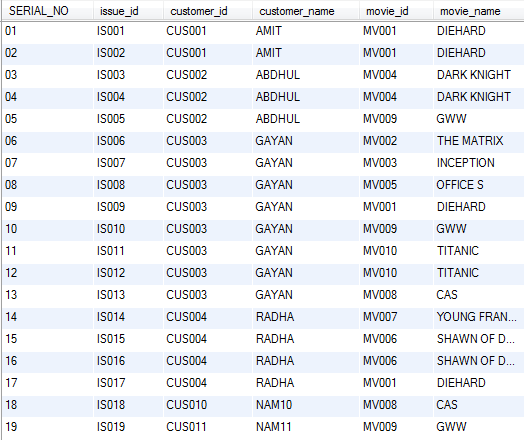
**11.Write a query to display serial number,issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.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

SELECT substring(ci.issue\_id,-2) SERIAL\_NO,ci.issue\_id,c.customer\_id,c.customer\_name,

m.movie\_id,m.movie\_name FROM customer\_master c JOIN customer\_issue\_details ci

ON c.customer\_id=ci.customer\_id JOIN movies\_master m ON m.movie\_id=ci.movie\_id

ORDER BY SERIAL\_NO;



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

SELECT ci.issue\_id,ci.issue\_date,c.customer\_id,c.customer\_name,c.contact\_no FROM

customer\_master c JOIN customer\_issue\_details ci ON c.customer\_id=ci.customer\_id

and year(ci.issue\_date)='2013' ORDER BY ci.issue\_date DESC;



**13.Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <br> Actors Name to be displayed in the below format.LEAD\_ACTOR\_ONE space ambersant space LEAD\_ACTOR\_TWO.Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria"Hint:Use ACTORS as alias name for actors name. <br> Display the records in ascending order based on movie name.**

SELECT movie\_id,movie\_name,concat(lead\_role\_1,' & ',lead\_role\_2) ACTOR FROM movies\_master

WHERE movie\_id NOT IN (SELECT movie\_id FROM customer\_issue\_details) ORDER BY movie\_name;



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

SELECT upper(director) DIRECTOR\_NAME,movie\_name,lead\_role\_1 FROM movies\_master

GROUP BY director HAVING count(movie\_id)>1 ORDER BY director,movie\_name DESC;



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

SELECT count(customer\_id) NO\_OF\_CUSTOMER FROM customer\_master

WHERE contact\_no is not null and year(date\_of\_registration)='2012';



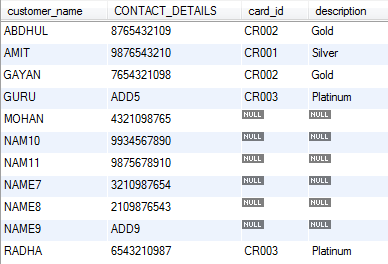
**16.Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based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

SELECT c.customer\_name,ifnull(c.contact\_no,c.contact\_add) CONTACT\_DETAILS,l.card\_id,l.description FROM

customer\_master c LEFT JOIN customer\_card\_details cc ON c.customer\_id=cc.customer\_id

LEFT JOIN library\_card\_master l ON l.card\_id=cc.card\_id

ORDER BY customer\_name;



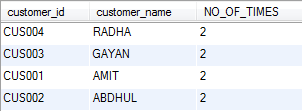
**17. Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and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

SELECT ci.customer\_id,c.customer\_name,count(ci.movie\_id) NO\_OF\_TIMES FROM

customer\_issue\_details ci JOIN customer\_master c ON c.customer\_id=ci.customer\_id

GROUP BY ci.customer\_id,ci.movie\_id HAVING count(movie\_id)>1

ORDER BY customer\_name DESC;



**18.Write a query to display customer id, customer name,contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.&nbsp; Hint:Use NO\_OF\_MOVIES as alias name for number of movies column. Hint:Use CONTACT\_ISD as alias name for contact number. Display the records sorted in ascending order based on customer name and then by movie category.**

SELECT c.customer\_id,c.customer\_name,concat('+91-',substring(c.contact\_no,1,3),'-',

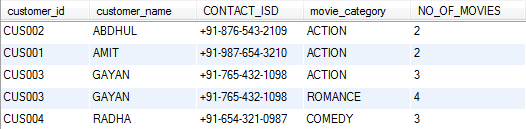
substring(c.contact\_no,4,3),'-',substring(c.contact\_no,7)) CONTACT\_ISD

,m.movie\_category,count(cc.movie\_id) NO\_OF\_MOVIES FROM customer\_master c JOIN customer\_issue\_details cc

ON c.customer\_id=cc.customer\_id JOIN movies\_master m ON m.movie\_id=cc.movie\_id

GROUP BY c.customer\_id,m.movie\_category HAVING count(cc.movie\_id)>1

ORDER BY customer\_name,movie\_category;



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

SELECT cid.customer\_id , customer\_name FROM customer\_master cm JOIN customer\_issue\_details cidON cm.customer\_id=cid.customer\_id

GROUP BY customer\_id , customer\_name

HAVING count(movie\_id)>=ALL(SELECT count(movie\_id)

FROM customer\_issue\_details

GROUP BY customer\_id)

UNION

SELECT cid.customer\_id , customer\_name FROM

customer\_master cm JOIN customer\_issue\_details cid

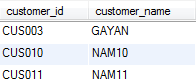
ON cm.customer\_id=cid.customer\_id

GROUP BY customer\_id , customer\_name

HAVING count(movie\_id)<=ALL(SELECT count(movie\_id)

FROM customer\_issue\_details

GROUP BY customer\_id) ORDER BY customer\_name;



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

SELECT c.customer\_id,c.customer\_name,count(m.movie\_id) NO\_OF\_TIMES FROM

customer\_master c JOIN customer\_issue\_details cc ON c.customer\_id=cc.customer\_id

JOIN movies\_master m ON m.movie\_id=cc.movie\_id

WHERE m.movie\_category='Comedy'

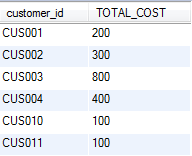
GROUP BY c.customer\_id HAVING count(m.movie\_id)>1

ORDER BY customer\_name;



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

SELECT cid.customer\_id, sum(m.rent\_cost) TOTAL\_COST FROM customer\_issue\_details cid JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id GROUP BY cid.customer\_id order by customer\_id;



LOAN

create database loan;

use loan;

CREATE TABLE loan\_card\_master

(

loan\_id varchar(6) PRIMARY KEY,

loan\_type varchar(15),

duration\_in\_years int(2)

);

CREATE TABLE employee\_master

(

employee\_id varchar(6) PRIMARY KEY,

employee\_name varchar(20),

designation varchar(25),

department varchar(25),

gender char(1),

date\_of\_birth date,

date\_of\_joining date

);

CREATE TABLE item\_master

(

item\_id varchar(6) PRIMARY KEY,

item\_description varchar(25),

issue\_status char(1),

item\_make varchar(25),

item\_category varchar(20),

item\_valuation int(6)

);

CREATE TABLE employee\_card\_details

(

employee\_id varchar(6) REFERENCES employee\_master,

loan\_id varchar(6) REFERENCES loan\_card\_master,

card\_issue\_date date

);

CREATE TABLE employee\_issue\_details

(

issue\_id varchar(6) PRIMARY KEY,

employee\_id varchar(6) REFERENCES employee\_master,

item\_id varchar(6) REFERENCES item\_master,

issue\_date date,

return\_date date

);

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

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

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

insert into employee\_issue\_details values('ISS001','E00001','I00001','2012-02-03','2014-02-03');

insert into employee\_issue\_details values('ISS002','E00001','I00004','2012-02-03','2020-02-03');

insert into employee\_issue\_details values('ISS003','E00002','I00005','2013-01-03','2015-01-03');

insert into employee\_issue\_details values('ISS004','E00003','I00007','2010-07-04','2012-07-04');

insert into employee\_issue\_details values('ISS005','E00003','I00008','2010-07-04','2012-08-05');

insert into employee\_issue\_details values('ISS006','E00003','I00010','2012-03-14','2012-06-15');

insert into employee\_issue\_details values('ISS007','E00004','I00012','2013-04-14','2016-04-14');

insert into employee\_issue\_details values('ISS008','E00006','I00018','2012-08-18','2019-04-17');

insert into employee\_issue\_details values('ISS009','E00004','I00018','2013-04-18','2013-05-18');

insert into employee\_master values('E00001','Ram','Manager','Finance','M','1973-12-01','2000-01-01');

insert into employee\_master values('E00002','Abhay','Assistant Manager','Finance','M','1976-01-01','2006-12-01');

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

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

insert into employee\_master values('E00005','Radhica','Manager','HR','F','1976-07-22','2004-01-23');

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

insert into employee\_card\_details values('E00001','L00001','2000-01-01');

insert into employee\_card\_details values('E00001','L00002','2000-01-01');

insert into employee\_card\_details values('E00001','L00003','2002-12-14');

insert into employee\_card\_details values('E00002','L00001','2007-02-01');

insert into employee\_card\_details values('E00002','L00002','2007-03-11');

insert into employee\_card\_details values('E00003','L00001','2007-04-15');

insert into employee\_card\_details values('E00003','L00002','2007-04-15');

insert into employee\_card\_details values('E00003','L00003','2007-04-15');

INSERT INTO item\_master VALUES ('I00001','Tea Table','Y','Wooden','Furniture',5000);

INSERT INTO item\_master VALUES ('I00002','Dinning Table','N','Wooden','Furniture',15000);

INSERT INTO item\_master VALUES ('I00003','Tea Table','N','Steel','Furniture',6000);

INSERT INTO item\_master VALUES ('I00004','Side Table','Y','Wooden','Furniture',2000);

INSERT INTO item\_master VALUES ('I00005','Side Table','Y','Steel','Furniture',1500);

INSERT INTO item\_master VALUES ('I00006','Tea Table','N','Steel','Furniture',7000);

INSERT INTO item\_master VALUES ('I00007','Dinning Chair','Y','Wooden','Furniture',1500);

INSERT INTO item\_master VALUES ('I00008','Tea Table','Y','Wooden','Furniture',4000);

INSERT INTO item\_master VALUES ('I00009','Sofa','N','Wooden','Furniture',18000);

INSERT INTO item\_master VALUES ('I00010','Cupboard','Y','Steel','Furniture',10000);

INSERT INTO item\_master VALUES ('I00011','Cupboard','N','Steel','Furniture',14000);

INSERT INTO item\_master VALUES ('I00012','Double Bed','Y','Wooden','Furniture',21000);

INSERT INTO item\_master VALUES ('I00013','Double Bed','Y','Wooden','Furniture',20000);

INSERT INTO item\_master VALUES ('I00014','Single Bed','Y','Steel','Furniture',10000);

INSERT INTO item\_master VALUES ('I00015','Single Bed','N','Steel','Furniture',10000);

INSERT INTO item\_master VALUES ('I00016','Tea Set','Y','Glass','Crockery',3000);

INSERT INTO item\_master VALUES ('I00017','Tea Set','Y','Bonechina','Crockery',4000);

INSERT INTO item\_master VALUES ('I00018','Dinning Set','Y','Glass','Crockery',4500);

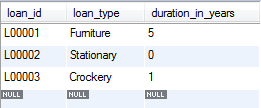
INSERT INTO item\_master VALUES ('I00019','Dinning Set','N','Bonechina','Crockery',5000);

INSERT INTO item\_master VALUES ('I00020','Pencil','Y','Wooden','Stationary',5);

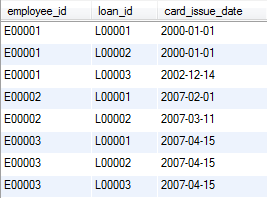
INSERT INTO item\_master VALUES ('I00021','Pen','Y','Plastic','Stationary',100);

INSERT INTO item\_master VALUES ('I00022','Pen','N','Plastic','Stationary',200);

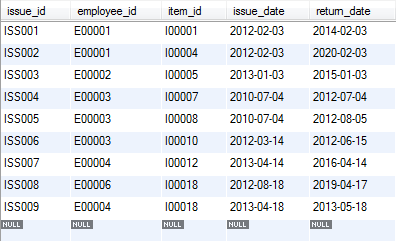
**LOAN CARD MASTER**



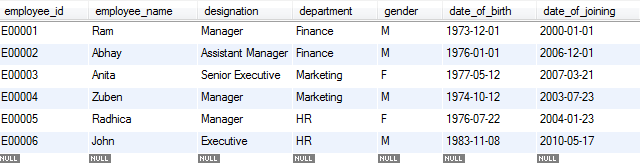
**EMPLOYEE CARD DETAILS**



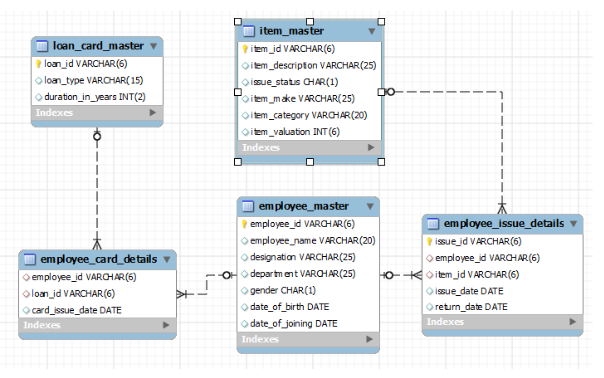
**EMPLOYEE ISSUE DETAILS**



**EMPLOYEE MASTER**



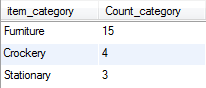
**ITEM MASTER**



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

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

item\_master GROUP BY item\_category ORDER BY Count\_category DESC;



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

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

employee\_master WHERE department='HR';

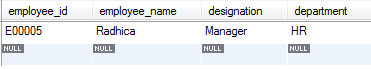


**3. Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.**

SELECT employee\_id,employee\_name,designation,department FROM employee\_master

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

ORDER BY employee\_id;



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

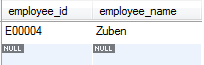
SELECT employee\_id,employee\_name FROM employee\_master

WHERE employee\_id IN(SELECT employee\_id FROM employee\_issue\_details

WHERE item\_id IN (SELECT item\_id FROM item\_master

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

item\_master i JOIN employee\_issue\_details e ON i.item\_id=e.item\_id)));

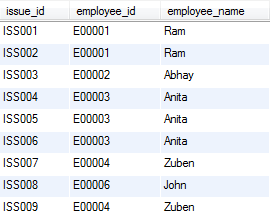


**5. Write a query to display issue\_id, employee\_id, employee\_name. Display the records sorted in ascending order based on issue id.**

SELECT eid.issue\_id, eid.employee\_id, em.employee\_name

FROM employee\_master em JOIN employee\_issue\_details eid

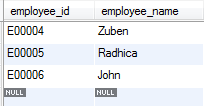
ON em.employee\_id=eid.employee\_id ORDER BY eid.issue\_id;



**6. Write a query to display employee id, employee name who don’t have loan cards.Display the records sorted in ascending order based on employee id.**

SELECT employee\_id,employee\_name FROM employee\_master

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



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

SELECT count(loan\_id) No\_of\_Cards FROM

employee\_card\_details WHERE employee\_id IN

(SELECT employee\_id FROM employee\_master WHERE employee\_name='Ram');

**(or)**

SELECT count(loan\_id) No\_of\_Cards FROM

employee\_card\_details c JOIN employee\_master e

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

WHERE e.employee\_name= 'Ram';



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

SELECT count(e.employee\_id) Count\_Stationary

FROM employee\_card\_details e JOIN loan\_card\_master l

ON e.loan\_id=l.loan\_id WHERE l.loan\_type='Stationary';

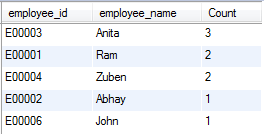


**9. Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then**

SELECT e.employee\_id,employee\_name,count(e.item\_id) Count FROM

employee\_issue\_details e JOIN employee\_master em ON e.employee\_id=em.employee\_id

GROUP BY e.employee\_id ORDER BY count DESC,e.employee\_id;



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

SELECT employee\_id,employee\_name FROM employee\_master

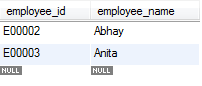
WHERE employee\_id IN(SELECT employee\_id FROM employee\_issue\_details

WHERE item\_id IN (SELECT item\_id FROM item\_master

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

item\_master i JOIN employee\_issue\_details e ON i.item\_id=e.item\_id)))

ORDER BY employee\_id;



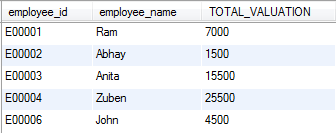
**11. Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL\_VALUATION.Display the records sorted in ascending order based on employee id.Consider only employees who have been issued atleast 1 item.**

SELECT e.employee\_id,em.employee\_name,sum(i.item\_valuation) TOTAL\_VALUATION FROM

item\_master i JOIN employee\_issue\_details e ON e.item\_id=i.item\_id

JOIN employee\_master em ON em.employee\_id=e.employee\_id

GROUP BY e.employee\_id ORDER BY employee\_id;



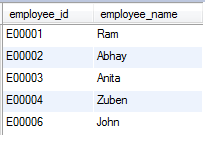
**12. Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days.Display the records sorted in ascending order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

SELECT DISTINCT e.employee\_id,e.employee\_name FROM

employee\_master e JOIN employee\_issue\_details ei ON e.employee\_id=ei.employee\_id

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

ORDER BY employee\_id;



**13. Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT\_ITEMS.Display the records sorted in ascending order on employee id.**

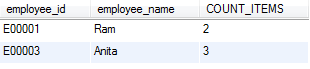
SELECT e.employee\_id,e.employee\_name,count(ei.item\_id) COUNT\_ITEMS FROM

employee\_master e JOIN employee\_issue\_details ei ON e.employee\_id=ei.employee\_id

JOIN item\_master i ON ei.item\_id=i.item\_id

WHERE i.item\_category='Furniture'

GROUP BY ei.employee\_id HAVING count(ei.item\_id)>1;



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

SELECT gender,count(employee\_id) FROM employee\_master

GROUP BY gender ORDER BY gender;



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

SELECT employee\_id,employee\_name FROM employee\_master

WHERE year(date\_of\_joining)>'2005'

ORDER BY employee\_id;

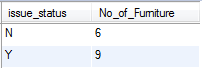


**16. Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No\_of\_Furnitures.Display the records sorted in ascending order based on issue\_status.**

SELECT issue\_status,count(item\_id) No\_of\_Furniture FROM

item\_master WHERE item\_category='Furniture'

GROUP BY issue\_status ORDER BY issue\_status;

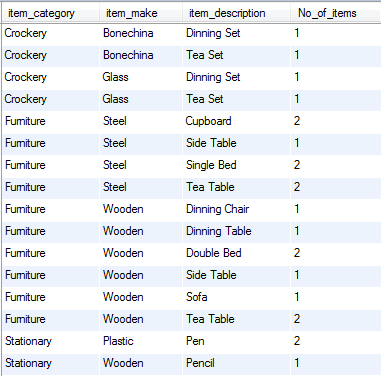


**17. Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No\_of\_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.**

SELECT item\_category,item\_make,item\_description,count(item\_id) No\_of\_items FROM

item\_master GROUP BY item\_category,item\_make,item\_description

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



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

SELECT e.employee\_id,employee\_name,i.item\_id,i.item\_description FROM

employee\_master e JOIN employee\_issue\_details ei ON e.employee\_id=ei.employee\_id

JOIN item\_master i ON i.item\_id=ei.item\_id

WHERE month(ei.issue\_date)='01' and year(ei.issue\_date)='2013'

ORDER BY employee\_id,item\_id;



**19. Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.Give the alias name for category count as COUNT\_CATEGORY.Display the records sorted in ascending order based on employee id.**

SELECT ei.employee\_id,e.employee\_name,count(DISTINCT i.item\_category) COUNT\_CATEGORY FROM

employee\_master e JOIN employee\_issue\_details ei ON e.employee\_id=ei.employee\_id

JOIN item\_master i ON i.item\_id=ei.item\_id

GROUP BY ei.employee\_id

HAVING COUNT\_CATEGORY>=2

ORDER BY employee\_id;

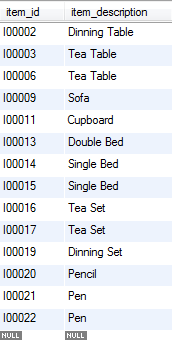


**20. Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.**

SELECT item\_id, item\_description FROM item\_master

WHERE item\_id NOT IN (SELECT item\_id from employee\_issue\_details)

ORDER BY item\_id;



**21. Write a query to display the employee id, employee name andtotal valuationfor the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed. ]**

SELECT e.employee\_id,em.employee\_name,sum(i.item\_valuation) TOTAL\_VALUATION FROM

item\_master i JOIN employee\_issue\_details e ON e.item\_id=i.item\_id

JOIN employee\_master em ON em.employee\_id=e.employee\_id

GROUP BY e.employee\_id HAVING sum(i.item\_valuation)<=ALL(

SELECT sum(i.item\_valuation) TOTAL\_VALUATION FROM

item\_master i JOIN employee\_issue\_details e ON e.item\_id=i.item\_id

JOIN employee\_master em ON em.employee\_id=e.employee\_id

GROUP BY e.employee\_id);



**22. Write a query to display the employee id, employee name, card issue date and card valid date.Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD\_VALID\_DATE.[Hint: Validity in years for the loan card is given in loan\_card\_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'. ]**

SELECT e.employee\_id,e.employee\_name,card\_issue\_date,

case

when l.duration\_in\_years>0 then date\_add(ec.card\_issue\_date,interval l.duration\_in\_years year)

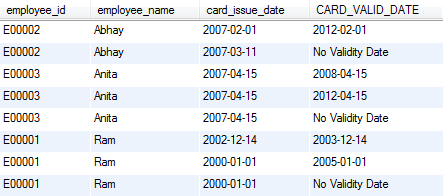
when l.duration\_in\_years=0 then 'No Validity Date' end CARD\_VALID\_DATE

FROM

employee\_master e JOIN employee\_card\_details ec ON e.employee\_id=ec.employee\_id

JOIN loan\_card\_master l ON l.loan\_id=ec.loan\_id

ORDER BY employee\_name,CARD\_VALID\_DATE;



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

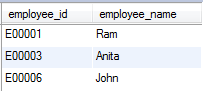
SELECT DISTINCT e.employee\_id,e.employee\_name FROM

employee\_master e JOIN employee\_issue\_details ei ON e.employee\_id=ei.employee\_id

WHERE e.employee\_id NOT IN (SELECT employee\_id FROM employee\_issue\_details

WHERE year(issue\_date)='2013')

ORDER BY employee\_id;



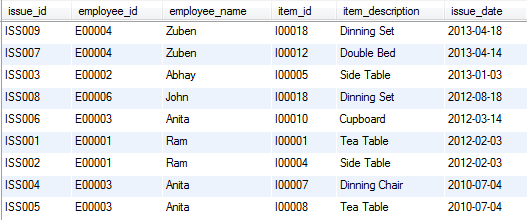
**24. Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.**

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

FROM employee\_issue\_details eid JOIN employee\_master em ON eid.employee\_id=em.employee\_id

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

ORDER BY issue\_date DESC, issue\_id;



**25. Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display. ]**

SELECT e.employee\_id,em.employee\_name,sum(i.item\_valuation) TOTAL\_VALUATION FROM

item\_master i JOIN employee\_issue\_details e ON e.item\_id=i.item\_id

JOIN employee\_master em ON em.employee\_id=e.employee\_id

GROUP BY e.employee\_id HAVING sum(i.item\_valuation)>=ALL(

SELECT sum(i.item\_valuation) TOTAL\_VALUATION FROM

item\_master i JOIN employee\_issue\_details e ON e.item\_id=i.item\_id

JOIN employee\_master em ON em.employee\_id=e.employee\_id

GROUP BY e.employee\_id);

