





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'

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

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

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

FROM customer\_master cm INNER JOIN account\_master am

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

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

ORDER BY am.customer\_number, account\_number

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

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

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

FROM customer\_master cm INNER JOIN account\_master am

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

WHERE account\_status='Terminated'

ORDER BY am.customer\_number, account\_number

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

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

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

FROM account\_master am INNER JOIN transaction\_details td

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

WHERE customer\_number like '%001'

GROUP BY transaction\_type

ORDER BY transaction\_type

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)

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

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

FROM account\_master am INNER JOIN transaction\_details td

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

WHERE transaction\_type='deposit'

GROUP BY account\_number

ORDER BY account\_number

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

select

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

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

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

select firstname

FROM customer\_master cm INNER JOIN account\_master am

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

group by firstname

having count(account\_number)>1

order by firstname;

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

Display the records sorted in order based on customer number.

SELECT ld.customer\_number, firstname, lastname

FROM customer\_master cm INNER JOIN loan\_details ld

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

GROUP BY customer\_number

HAVING count(branch\_id)>1

ORDER BY customer\_number

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

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

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

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

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

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

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

SELECT count(ld.customer\_number) Count

FROM customer\_master cm INNER JOIN loan\_details ld

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

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

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

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

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

SELECT td.account\_number

FROM account\_master am INNER JOIN transaction\_details td

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

group by td.account\_number

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

(SELECT count(td.transaction\_number)

FROM account\_master am INNER JOIN transaction\_details td

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

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

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

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

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

select branch\_name,branch\_city

FROM branch\_master INNER JOIN account\_master

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

group by branch\_name

having count(customer\_number)>=ALL

(select count(customer\_number)

FROM branch\_master INNER JOIN account\_master

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

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

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

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

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

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

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

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

FROM account\_master am INNER JOIN transaction\_details td

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

GROUP BY td.account\_number

HAVING Withdrawal > Deposit

ORDER BY am.account\_number

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

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

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

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

THEN transaction\_amount END)) -

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

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

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

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

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

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

FROM customer\_master cm inner JOIN account\_master am

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

INNER JOIN transaction\_details td

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

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

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

SELECT firstname

FROM customer\_master INNER JOIN account\_master

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

GROUP BY firstname

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

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

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

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

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

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

SELECT ld.customer\_number, firstname, lastname

FROM customer\_master cm INNER JOIN loan\_details ld

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

group by customer\_number

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

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

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

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

FROM customer\_master cm INNER JOIN loan\_details ld

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

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

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

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

SELECT account\_number, firstname, lastname

FROM customer\_master cm INNER JOIN account\_master am

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

WHERE account\_status='Active'

ORDER BY account\_number

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

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

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

FROM customer\_master order by customer\_number

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

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

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

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

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

SELECT firstname, customer\_city,account\_number

FROM customer\_master cm INNER JOIN account\_master am

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

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