Task 4

1. Retrieve the customer(s) with the highest account balance.

Query:

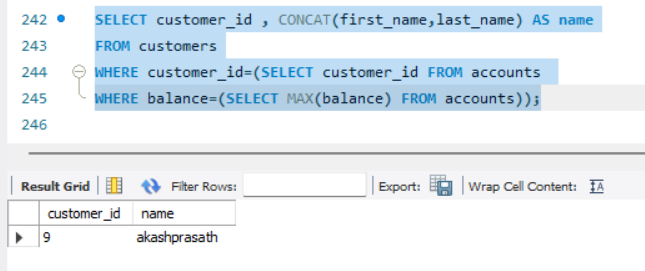
SELECT customer\_id , CONCAT(first\_name,last\_name) AS name

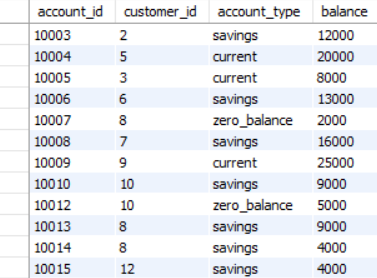
FROM customers

WHERE customer\_id=(SELECT customer\_id FROM accounts

WHERE balance=(SELECT MAX(balance) FROM accounts));

Output:





1. Calculate the average account balance for customers who have more than one account.

Query:

SELECT AVG(balance) FROM accounts

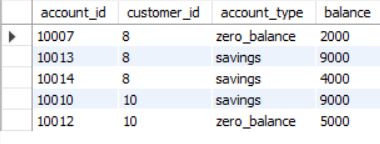
WHERE customer\_id IN

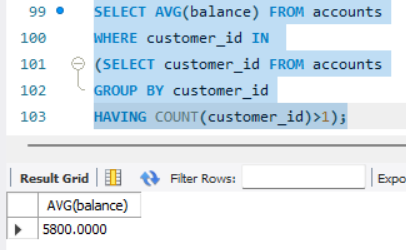
(SELECT customer\_id FROM accounts

GROUP BY customer\_id

HAVING COUNT(customer\_id)>1);

Output:





1. Retrieve accounts with transactions whose amounts exceed the average transaction amount.

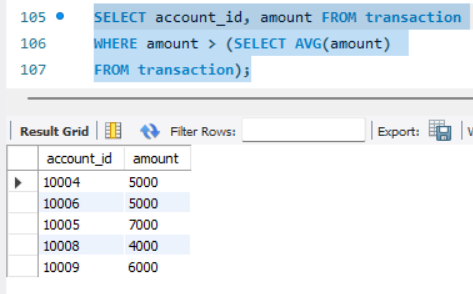
Query:

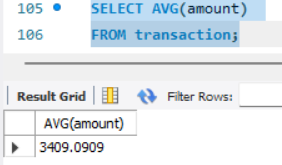
SELECT account\_id FROM transaction

WHERE amount > (SELECT AVG(amount)

FROM transaction);

Output:





1. Identify customers who have no recorded transactions.

Query:

SELECT customer\_id,CONCAT(first\_name,last\_name) AS name

FROM customers

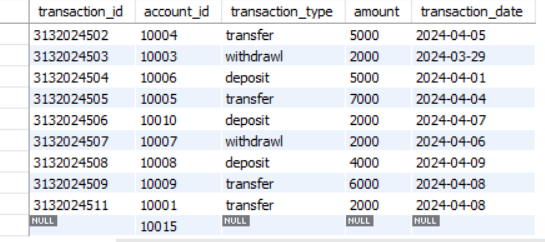
WHERE customer\_id = (SELECT customer\_id

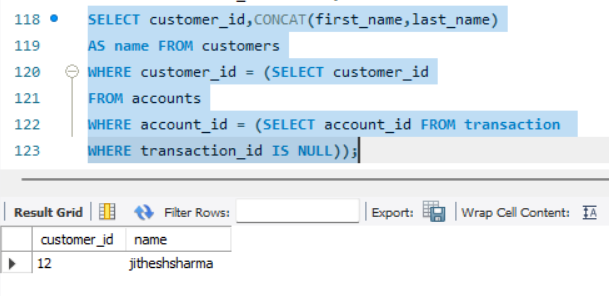
FROM accounts

WHERE account\_id = (SELECT account\_id FROM transaction

WHERE transaction\_id IS NULL));

Output:





1. Calculate the total balance of accounts with no recorded transactions.

Query:

SELECT balance

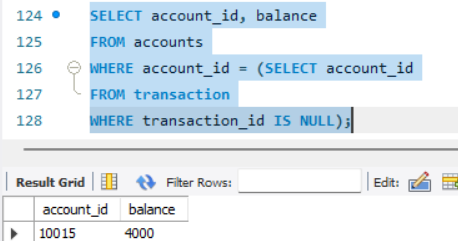
FROM accounts

WHERE account\_id = (SELECT account\_id

FROM transaction

WHERE transaction\_id IS NULL);

Output:



1. Retrieve transactions for accounts with the lowest balance.

Query:

SELECT transaction\_id,transaction\_type,amount,

transaction\_date

FROM transaction

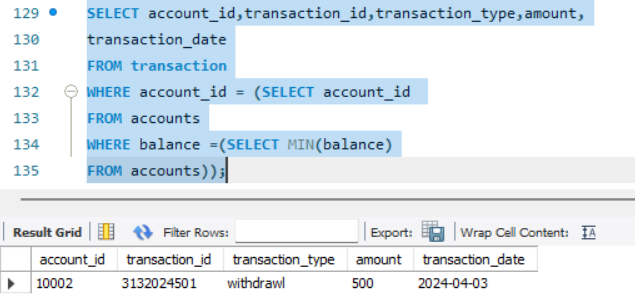
WHERE account\_id = (SELECT account\_id

FROM accounts

WHERE balance =(SELECT MIN(balance)

FROM accounts));

Output:



1. Identify customers who have accounts of multiple types.

Query:

SELECT customer\_id,CONCAT(first\_name,last\_name)

AS PeopleWithMultipleTypeAccounts

FROM customers

WHERE customer\_id IN (SELECT DISTINCT(customer\_id)

FROM accounts

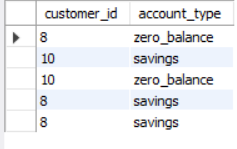
WHERE customer\_id IN (SELECT customer\_id

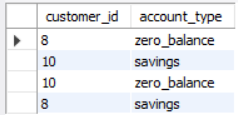
FROM accounts

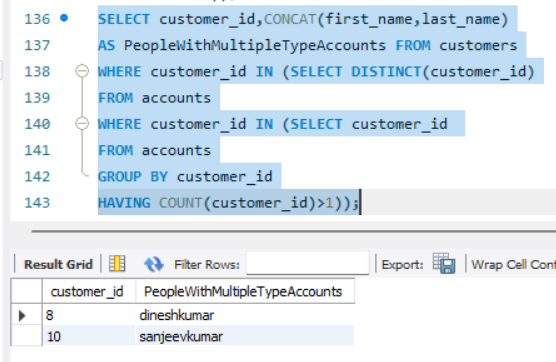
GROUP BY customer\_id

HAVING COUNT(customer\_id)>1));

Output:







1. Calculate the percentage of each account type out of the total number of accounts.

Query:

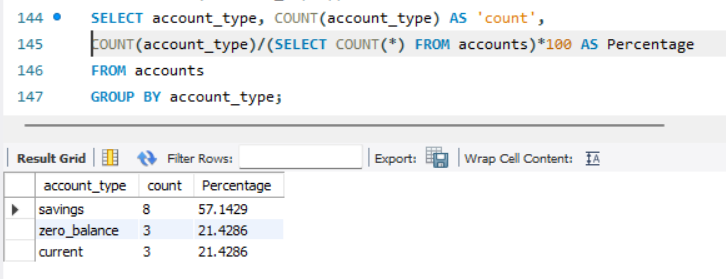
SELECT account\_type, COUNT(account\_type) AS 'count',

COUNT(account\_type)/(SELECT COUNT(\*) FROM accounts)\*100 AS Percentage

FROM accounts

GROUP BY account\_type;

Output:



1. Retrieve all transactions for a customer with a given customer\_id.

Query:

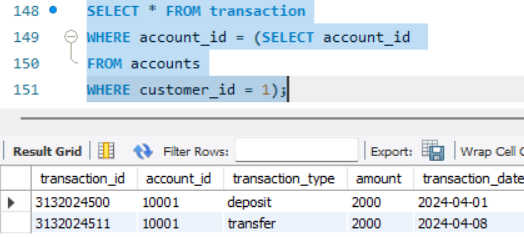
SELECT \* FROM transaction

WHERE account\_id = (SELECT account\_id

FROM accounts

WHERE customer\_id = 1);

Output:



1. Calculate the total balance for each account type, including a subquery within the SELECT clause.

Query:

SELECT account\_type,SUM(balance) AS total\_balance,

(SELECT COUNT(\*) FROM accounts A

WHERE A.account\_type=accounts.account\_type)

AS account\_type\_count

FROM accounts

GROUP BY account\_type;

Output:

