1. Write a SQL query to Find the average account balance for all customers.

mysql> SELECT AVG(balance) AS average\_balance

-> FROM Accounts;

+-----------------+

| average\_balance |

+-----------------+

| 5222.222222 |

+-----------------+

1 row in set (0.00 sec)

2. Write a SQL query to Retrieve the top 10 highest account balances.

mysql> SELECT \* FROM Accounts

-> ORDER BY balance DESC

-> LIMIT 10;

+------------+-------------+--------------+----------+

| account\_id | customer\_id | account\_type | balance |

+------------+-------------+--------------+----------+

| 107 | 6 | savings | 12000.00 |

| 103 | 2 | savings | 10000.00 |

| 106 | 5 | savings | 8000.00 |

| 101 | 1 | savings | 5500.00 |

| 108 | 6 | current | 5000.00 |

| 109 | 7 | current | 3000.00 |

| 102 | 1 | current | 2000.00 |

| 105 | 4 | current | 1500.00 |

| 104 | 3 | zero\_balance | 0.00 |

+------------+-------------+--------------+----------+

9 rows in set (0.01 sec)

3. Write a SQL query to Calculate Total Deposits for All Customers in specific date.  
  
mysql> UPDATE Transactions

-> SET transaction\_date = '2024-03-12 09:15:00'

-> WHERE transaction\_id IN (204, 208);

Query OK, 2 rows affected (0.02 sec)

Rows matched: 2 Changed: 2 Warnings: 0

mysql> select \* from Transactions;

+----------------+------------+------------------+---------+---------------------+

| transaction\_id | account\_id | transaction\_type | amount | transaction\_date |

+----------------+------------+------------------+---------+---------------------+

| 201 | 101 | deposit | 1000.00 | 2024-03-10 10:30:00 |

| 202 | 102 | withdrawal | 500.00 | 2024-03-11 11:00:00 |

| 203 | 103 | deposit | 2000.00 | 2024-03-12 09:15:00 |

| 204 | 104 | deposit | 1500.00 | 2024-03-12 09:15:00 |

| 205 | 105 | withdrawal | 200.00 | 2024-03-14 16:50:00 |

| 206 | 106 | deposit | 5000.00 | 2024-03-15 12:40:00 |

| 207 | 107 | withdrawal | 3000.00 | 2024-03-16 18:00:00 |

| 208 | 108 | deposit | 2500.00 | 2024-03-12 09:15:00 |

| 209 | 109 | deposit | 4000.00 | 2024-03-18 20:15:00 |

+----------------+------------+------------------+---------+---------------------+

9 rows in set (0.00 sec)

mysql> SELECT SUM(amount) AS total\_deposits

-> FROM Transactions

-> WHERE transaction\_type = 'deposit'

-> AND DATE(transaction\_date) = '2024-03-12';

+----------------+

| total\_deposits |

+----------------+

| 6000.00 |

+----------------+

1 row in set (0.00 sec)

4. Write a SQL query to Find the Oldest and Newest Customers.

mysql> SELECT

-> MIN(DOB) AS oldest\_customer,

-> MAX(DOB) AS newest\_customer

-> FROM Customers;

+-----------------+-----------------+

| oldest\_customer | newest\_customer |

+-----------------+-----------------+

| 1985-02-18 | 1996-05-05 |

+-----------------+-----------------+

1 row in set (0.00 sec)

5. Write a SQL query to Retrieve transaction details along with the account type.

mysql> SELECT t.transaction\_id, t.account\_id, a.account\_type, t.transaction\_type, t.amount, t.transaction\_date

-> FROM Transactions t

-> JOIN Accounts a ON t.account\_id = a.account\_id;

+----------------+------------+--------------+------------------+---------+---------------------+

| transaction\_id | account\_id | account\_type | transaction\_type | amount | transaction\_date |

+----------------+------------+--------------+------------------+---------+---------------------+

| 201 | 101 | savings | deposit | 1000.00 | 2024-03-10 10:30:00 |

| 202 | 102 | current | withdrawal | 500.00 | 2024-03-11 11:00:00 |

| 203 | 103 | savings | deposit | 2000.00 | 2024-03-12 09:15:00 |

| 204 | 104 | zero\_balance | deposit | 1500.00 | 2024-03-12 09:15:00 |

| 205 | 105 | current | withdrawal | 200.00 | 2024-03-14 16:50:00 |

| 206 | 106 | savings | deposit | 5000.00 | 2024-03-15 12:40:00 |

| 207 | 107 | savings | withdrawal | 3000.00 | 2024-03-16 18:00:00 |

| 208 | 108 | current | deposit | 2500.00 | 2024-03-12 09:15:00 |

| 209 | 109 | current | deposit | 4000.00 | 2024-03-18 20:15:00 |

+----------------+------------+--------------+------------------+---------+---------------------+

9 rows in set (0.00 sec)

6. Write a SQL query to Get a list of customers along with their account details.

mysql> SELECT c.customer\_id, c.first\_name, c.last\_name, a.account\_id, a.account\_type, a.balance

-> FROM Customers c

-> JOIN Accounts a ON c.customer\_id = a.customer\_id

-> ORDER BY c.customer\_id;

+-------------+------------+-----------+------------+--------------+----------+

| customer\_id | first\_name | last\_name | account\_id | account\_type | balance |

+-------------+------------+-----------+------------+--------------+----------+

| 1 | Amit | Sharma | 101 | savings | 5500.00 |

| 1 | Amit | Sharma | 102 | current | 2000.00 |

| 2 | Priya | Verma | 103 | savings | 10000.00 |

| 3 | Rahul | Patel | 104 | zero\_balance | 0.00 |

| 4 | Sneha | Iyer | 105 | current | 1500.00 |

| 5 | Vikram | Reddy | 106 | savings | 8000.00 |

| 6 | Ananya | Singh | 107 | savings | 12000.00 |

| 6 | Ananya | Singh | 108 | current | 5000.00 |

| 7 | Rohan | Das | 109 | current | 3000.00 |

+-------------+------------+-----------+------------+--------------+----------+

9 rows in set (0.00 sec)

7. Write a SQL query to Retrieve transaction details along with customer information for a specific account.

mysql> SELECT t.transaction\_id, t.account\_id, t.transaction\_type, t.amount, t.transaction\_date,

-> c.customer\_id, c.first\_name, c.last\_name

-> FROM Transactions t

-> JOIN Accounts a ON t.account\_id = a.account\_id

-> JOIN Customers c ON a.customer\_id = c.customer\_id

-> WHERE t.account\_id = 101

ID

+----------------+------------+------------------+---------+---------------------+-------------+------------+-----------+

| transaction\_id | account\_id | transaction\_type | amount | transaction\_date | customer\_id | first\_name | last\_name |

+----------------+------------+------------------+---------+---------------------+-------------+------------+-----------+

| 201 | 101 | deposit | 1000.00 | 2024-03-10 10:30:00 | 1 | Amit | Sharma |

+----------------+------------+------------------+---------+---------------------+-------------+------------+-----------+

1 row in set (0.00 sec)

8. Write a SQL query to Identify customers who have more than one account.

mysql> SELECT customer\_id, COUNT(account\_id) AS total\_accounts

-> FROM Accounts

-> GROUP BY customer\_id

-> HAVING COUNT(account\_id) > 1;

+-------------+----------------+

| customer\_id | total\_accounts |

+-------------+----------------+

| 1 | 2 |

| 6 | 2 |

+-------------+----------------+

2 rows in set (0.00 sec)

9. Write a SQL query to Calculate the difference in transaction amounts between deposits and withdrawals.

mysql> SELECT account\_id,

-> SUM(CASE WHEN transaction\_type = 'deposit' THEN amount ELSE 0 END) -

-> SUM(CASE WHEN transaction\_type = 'withdrawal' THEN amount ELSE 0 END) AS balance\_difference

-> FROM Transactions

-> GROUP BY account\_id;

+------------+--------------------+

| account\_id | balance\_difference |

+------------+--------------------+

| 101 | 1000.00 |

| 102 | -500.00 |

| 103 | 2000.00 |

| 104 | 1500.00 |

| 105 | -200.00 |

| 106 | 5000.00 |

| 107 | -3000.00 |

| 108 | 2500.00 |

| 109 | 4000.00 |

+------------+--------------------+

9 rows in set (0.01 sec)

10. Write a SQL query to Calculate the average daily balance for each account over a specified period.

mysql> SELECT account\_id, AVG(balance) AS average\_daily\_balance

-> FROM Accounts

-> WHERE account\_id IN (SELECT DISTINCT account\_id FROM Transactions WHERE transaction\_date BETWEEN '2024-03-11' AND '2024-03-16')

-> GROUP BY account\_id;

+------------+-----------------------+

| account\_id | average\_daily\_balance |

+------------+-----------------------+

| 102 | 2000.000000 |

| 103 | 10000.000000 |

| 104 | 0.000000 |

| 105 | 1500.000000 |

| 106 | 8000.000000 |

| 108 | 5000.000000 |

+------------+-----------------------+

6 rows in set (0.01 sec)

11. Calculate the total balance for each account type.

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

-> FROM Accounts

-> GROUP BY account\_type;

+--------------+---------------+

| account\_type | total\_balance |

+--------------+---------------+

| savings | 35500.00 |

| current | 11500.00 |

| zero\_balance | 0.00 |

+--------------+---------------+

3 rows in set (0.00 sec)

12. Identify accounts with the highest number of transactions order by descending order.

mysql> SELECT account\_id, COUNT(transaction\_id) AS transaction\_count

-> FROM Transactions

-> GROUP BY account\_id

-> ORDER BY transaction\_count DESC;

+------------+-------------------+

| account\_id | transaction\_count |

+------------+-------------------+

| 101 | 1 |

| 102 | 1 |

| 103 | 1 |

| 104 | 1 |

| 105 | 1 |

| 106 | 1 |

| 107 | 1 |

| 108 | 1 |

| 109 | 1 |

+------------+-------------------+

9 rows in set (0.00 sec)

13. List customers with high aggregate account balances, along with their account types.

mysql> SELECT c.customer\_id, c.first\_name, c.last\_name, a.account\_type, SUM(a.balance) AS total\_balance

-> FROM Customers c

-> JOIN Accounts a ON c.customer\_id = a.customer\_id

-> GROUP BY c.customer\_id, a.account\_type

-> HAVING SUM(a.balance) > 10000 -- Adjust threshold as needed

-> ORDER BY total\_balance DESC;

+-------------+------------+-----------+--------------+---------------+

| customer\_id | first\_name | last\_name | account\_type | total\_balance |

+-------------+------------+-----------+--------------+---------------+

| 6 | Ananya | Singh | savings | 12000.00 |

+-------------+------------+-----------+--------------+---------------+

1 row in set (0.00 sec)

14. Identify and list duplicate transactions based on transaction amount, date, and account.

mysql> SELECT transaction\_date, account\_id, amount, COUNT(\*) AS duplicate\_count

-> FROM Transactions

-> GROUP BY transaction\_date, account\_id, amount

-> HAVING COUNT(\*) > 1;

Empty set (0.00 sec)