QUERIES

1. Display the names of customers who have made transactions on stocks associated with companies headquartered in New York.

SELECT DISTINCT c.Name

FROM CUSTOMER c

JOIN WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

JOIN TRANSACTION\_HISTORY th ON w.stock\_acc\_id = th.stock\_acc\_id

JOIN IPO i ON th.trans\_id = i.trans\_id

JOIN COMPANY\_DETAILS cd ON i.comp\_name = cd.name

WHERE cd.hq = 'New York';

2. List the names of customers along with their total wallet balances (sum of wallet balances) and average bank account balances.

SELECT c.Name, SUM(w.balance) AS total\_wallet\_balance, AVG(b.balance) AS avg\_bank\_balance

FROM CUSTOMER c

INNER JOIN WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

INNER JOIN BANK\_ACC b ON c.bank\_acc\_no = b.acc\_no

GROUP BY c.Name;

3. Retrieve the names and quantities of stocks in the portfolio along with their associated current market price, sorted by the product of quantity and current market price in descending order.

SELECT p.stock\_name, p.qty, sd.cur\_mp, (p.qty \* sd.cur\_mp) AS total\_value

FROM PORTFOLIO p

JOIN STOCK\_DETAILS sd ON p.stock\_name = sd.stock\_name

ORDER BY total\_value DESC;

4. List the names of mutual funds along with the count of customers who have invested in them.

SELECT mf.name AS mutual\_fund\_name, COUNT(th.fund\_name) AS investor\_count

FROM MUTUAL\_FUNDS mf

LEFT JOIN TRANSACTION\_HISTORY th ON mf.name = th.fund\_name

GROUP BY mf.name;

5. Retrieve the names of customers along with their total wallet balances and the total balance of all their bank accounts.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance,

COALESCE(SUM(ba.balance), 0) AS total\_bank\_balance

FROM

CUSTOMER c

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

LEFT JOIN

BANK\_ACC ba ON c.bank\_acc\_no = ba.acc\_no

GROUP BY

c.Name;

6. Show the names of customers along with the total quantity of stocks in their portfolio and the sum of their wallet balances.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(p.qty), 0) AS total\_stock\_quantity,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance

FROM

CUSTOMER c

LEFT JOIN

PORTFOLIO p ON c.demat\_id = p.stock\_acc\_id

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

GROUP BY

c.Name;

7. List the names of customers along with their total wallet balances and the total balance of all their bank accounts, sorted by the total wallet balance in descending order.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance,

COALESCE(SUM(b.balance), 0) AS total\_bank\_balance

FROM

CUSTOMER c

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

LEFT JOIN

BANK\_ACC b ON c.bank\_acc\_no = b.acc\_no

GROUP BY

c.Name

ORDER BY

total\_wallet\_balance DESC;

8. Show the names of customers along with the total quantity of stocks in their portfolio and the sum of their wallet balances, sorted by the sum of wallet balances in descending order.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(p.qty), 0) AS total\_stock\_quantity,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance

FROM

CUSTOMER c

LEFT JOIN

PORTFOLIO p ON c.demat\_id = p.stock\_acc\_id

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

GROUP BY

c.Name

ORDER BY

total\_wallet\_balance DESC;

9. List the names of mutual funds along with their return percentages, but only consider mutual funds with a return percentage greater than 7%.

SELECT name, return\_perc

FROM MUTUAL\_FUNDS

WHERE return\_perc > 7;

10. Show the names of customers along with their total wallet balances and the average bank account balances, sorted by the total wallet balance in descending order.

SELECT c.Name, SUM(w.balance) AS total\_wallet\_balance, AVG(b.balance) AS avg\_bank\_balance

FROM CUSTOMER c

INNER JOIN WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

INNER JOIN BANK\_ACC b ON c.bank\_acc\_no = b.acc\_no

GROUP BY c.Name

ORDER BY total\_wallet\_balance DESC;

11. Retrieve the names of all customers.

SELECT Name FROM CUSTOMER;

12. Count the total number of transactions.

SELECT COUNT(\*) AS total\_transactions FROM TRANSACTIONS;

13.Retrieve the names and balances of all bank accounts.

SELECT acc\_no, balance

FROM BANK\_ACC;

14. Find the names and current market prices of all stocks in the stock details table.

SELECT stock\_name, cur\_mp

FROM STOCK\_DETAILS;

15. What is the total investment required for each mutual fund along with its return percentage?

SELECT name, investment\_req, return\_perc

FROM MUTUAL\_FUNDS;

16. Show the email addresses of all customers.

SELECT email\_id

FROM CUSTOMER;

17. Retrieve the quantity and premium listed for all IPOs.

SELECT qty, premium\_listed

FROM IPO;

18. List the names and headquarters of all companies.

SELECT name, hq

FROM COMPANY\_DETAILS;

19. Get the wallet balances for all customers.

SELECT balance

FROM WALLET;

20. Show the profit and quantity of stocks in each portfolio.

SELECT profit, qty

FROM PORTFOLIO;

21. Find the stock names and dividends in the watchlist.

SELECT stock\_name, dividends

FROM WATCHLIST;

22. Retrieve the IDs and account numbers of all demat accounts.

SELECT id, acc\_no

FROM DEMAT;

23. Retrieve the names and balances of bank accounts with balances greater than 10000.

SELECT acc\_no, balance FROM BANK\_ACC WHERE balance > 10000;

24. Find the total investment requirement for all mutual funds in a specific stock account.

SELECT SUM(investment\_req) FROM MUTUAL\_FUNDS WHERE stock\_acc\_id = 2339619165;

25. Get the transaction IDs and corresponding fund names for transactions with balances below 5000.

SELECT trans\_id, fund\_name FROM TRANSACTION\_HISTORY WHERE curr\_balance < 5000;

26. Find the company names and headquarters of companies with major shareholders holding more than 20%.

SELECT name, hq FROM COMPANY\_DETAILS WHERE name IN (SELECT name FROM PRIVATE\_SHAREHOLDINGS WHERE percentage > 7);

27. Find the stock names and dividends in the watchlist with dividends greater than 2.

SELECT stock\_name, dividends FROM WATCHLIST WHERE dividends > 0.75;

28. Get the contact numbers of customers residing in a specific state.

SELECT contact\_no FROM CUSTOMER WHERE cust\_id IN (SELECT cust\_id FROM CUSTOMER WHERE state = 'California');

29. Retrieve the names of customers who have a bank account and contact numbers.

SELECT Name FROM CUSTOMER WHERE bank\_acc\_no IS NOT NULL AND contact\_no IS NOT NULL;

30. Get the stock names and current market prices from the stock details table for stocks listed in a specific exchange.

SELECT stock\_name, cur\_mp FROM STOCK\_DETAILS WHERE excahnge = 'NASDAQ';

QUERIES

1. Display the names of customers who have made transactions on stocks associated with companies headquartered in New York.

SELECT DISTINCT c.Name

FROM CUSTOMER c

JOIN WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

JOIN TRANSACTION\_HISTORY th ON w.stock\_acc\_id = th.stock\_acc\_id

JOIN IPO i ON th.trans\_id = i.trans\_id

JOIN COMPANY\_DETAILS cd ON i.comp\_name = cd.name

WHERE cd.hq = 'New York';

2. List the names of customers along with their total wallet balances (sum of wallet balances) and average bank account balances.

SELECT c.Name, SUM(w.balance) AS total\_wallet\_balance, AVG(b.balance) AS avg\_bank\_balance

FROM CUSTOMER c

INNER JOIN WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

INNER JOIN BANK\_ACC b ON c.bank\_acc\_no = b.acc\_no

GROUP BY c.Name;

3. Retrieve the names and quantities of stocks in the portfolio along with their associated current market price, sorted by the product of quantity and current market price in descending order.

SELECT p.stock\_name, p.qty, sd.cur\_mp, (p.qty \* sd.cur\_mp) AS total\_value

FROM PORTFOLIO p

JOIN STOCK\_DETAILS sd ON p.stock\_name = sd.stock\_name

ORDER BY total\_value DESC;

4. List the names of mutual funds along with the count of customers who have invested in them.

SELECT mf.name AS mutual\_fund\_name, COUNT(th.fund\_name) AS investor\_count

FROM MUTUAL\_FUNDS mf

LEFT JOIN TRANSACTION\_HISTORY th ON mf.name = th.fund\_name

GROUP BY mf.name;

5. Retrieve the names of customers along with their total wallet balances and the total balance of all their bank accounts.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance,

COALESCE(SUM(ba.balance), 0) AS total\_bank\_balance

FROM

CUSTOMER c

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

LEFT JOIN

BANK\_ACC ba ON c.bank\_acc\_no = ba.acc\_no

GROUP BY

c.Name;

6. Show the names of customers along with the total quantity of stocks in their portfolio and the sum of their wallet balances.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(p.qty), 0) AS total\_stock\_quantity,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance

FROM

CUSTOMER c

LEFT JOIN

PORTFOLIO p ON c.demat\_id = p.stock\_acc\_id

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

GROUP BY

c.Name;

7. List the names of customers along with their total wallet balances and the total balance of all their bank accounts, sorted by the total wallet balance in descending order.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance,

COALESCE(SUM(b.balance), 0) AS total\_bank\_balance

FROM

CUSTOMER c

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

LEFT JOIN

BANK\_ACC b ON c.bank\_acc\_no = b.acc\_no

GROUP BY

c.Name

ORDER BY

total\_wallet\_balance DESC;

8. Show the names of customers along with the total quantity of stocks in their portfolio and the sum of their wallet balances, sorted by the sum of wallet balances in descending order.

SELECT

c.Name AS customer\_name,

COALESCE(SUM(p.qty), 0) AS total\_stock\_quantity,

COALESCE(SUM(w.balance), 0) AS total\_wallet\_balance

FROM

CUSTOMER c

LEFT JOIN

PORTFOLIO p ON c.demat\_id = p.stock\_acc\_id

LEFT JOIN

WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

GROUP BY

c.Name

ORDER BY

total\_wallet\_balance DESC;

9. List the names of mutual funds along with their return percentages, but only consider mutual funds with a return percentage greater than 7%.

SELECT name, return\_perc

FROM MUTUAL\_FUNDS

WHERE return\_perc > 7;

10. Show the names of customers along with their total wallet balances and the average bank account balances, sorted by the total wallet balance in descending order.

SELECT c.Name, SUM(w.balance) AS total\_wallet\_balance, AVG(b.balance) AS avg\_bank\_balance

FROM CUSTOMER c

INNER JOIN WALLET w ON c.bank\_acc\_no = w.bank\_acc\_no

INNER JOIN BANK\_ACC b ON c.bank\_acc\_no = b.acc\_no

GROUP BY c.Name

ORDER BY total\_wallet\_balance DESC;

11. Get the contact numbers of customers residing in a specific state.

SELECT contact\_no FROM CUSTOMER WHERE cust\_id IN (SELECT cust\_id FROM CUSTOMER WHERE state = 'California');

12. Retrieve the names of customers who have made transactions on stocks associated with companies headquartered in states with more than 2 stock details.

SELECT DISTINCT c.Name

FROM customer c JOIN bank\_acc ba ON c.cust\_id = ba.cust\_id

JOIN stock\_acc sa ON ba.acc\_no = sa.acc\_no

JOIN watchlist w ON sa.stock\_id = w.stock\_id

JOIN stock\_details sd ON w.stock\_name = sd.stock\_name

JOIN company\_details cd ON cd.comp\_name = sd.comp\_name

GROUP BY cd.hq, c.name

HAVING COUNT(cd.comp\_name) > 2;

13. Display the names of customers along with the total wallet balances and the average bank account balances, but only for customers whose total wallet balance exceeds twice their average bank account balance.

SELECT c.Name, SUM(w.balance) AS total\_wallet\_balance, AVG(b.balance) AS avg\_bank\_balance

FROM CUSTOMER c

JOIN bank\_acc b ON c.cust\_id = b.cust\_id

JOIN wallet w ON w.bank\_acc\_no = b.acc\_no

group by c.name

having sum(w.balance) > (2 \* AVG(b.balance));

14. List the names of companies with an IPO premium listed greater than the average premium listed, along with the total quantity of stocks held by customers for those companies.

SELECT comp\_name, sum(qty) AS total\_stock\_qty

FROM IPO i

GROUP BY comp\_name

HAVING premium\_listed > (SELECT AVG(premium\_listed) FROM IPO);

15. Retrieve the names of customers who have made transactions on stocks associated with companies headquartered in states with an average public shareholding percentage greater than 50%.

SELECT DISTINCT c.Name

FROM CUSTOMER c

JOIN bank\_acc ba on ba.cust\_id = c.cust\_id

JOIN stock\_acc sa on sa.bank\_acc\_no = ba.bank\_acc\_no

JOIN portfolio p on p.stock\_acc\_id = sa.stock\_id

JOIN stock\_details sd on sd.stock\_name = p.stock\_name

JOIN company\_details cd on cd.comp\_name = sd.comp\_name

GROUP BY cd.hq, c.Name

HAVING AVG(hd.public\_shareholding\_perc) > 50;

16.Show the names of customers along with the total quantity of stocks in their portfolio and the total balance of all their bank accounts, but only for customers who have made transactions in the last 60 days.

SELECT c.Name, SUM(p.qty) AS total\_stock\_qty, SUM(b.balance) AS total\_bank\_balance

FROM CUSTOMER c

JOIN bank\_acc b on b.cust\_id = c.cust\_id

JOIN stock\_acc sa on sa.acc\_no = b.acc\_no

JOIN portfolio p on p.stock\_acc\_id = sa.stock\_id

JOIN transaction\_history th on sa.stock\_id = th.stock\_acc\_id

WHERE (CURRENT\_DATE-th.date\_of\_trans) >= 60

GROUP BY c.Name;

17. List the names of customers who have made transactions on stocks with a growth rate higher than the average growth rate of stocks associated with companies headquartered in states where the total bank balance of customers is less than the total wallet balance of customers.

SELECT DISTINCT c.Name

FROM CUSTOMER c

JOIN bank\_acc ba on ba.cust\_id = c.cust\_id

JOIN stock\_acc sa on sa.acc\_no = ba.acc\_no

JOIN wallet w on w.bank\_acc\_no = ba.acc\_no

JOIN portfolio p on p.stock\_acc\_id = sa.stock\_id

JOIN transaction\_history th on sa.stock\_id = th.stock\_acc\_id

JOIN stock\_details sd on sd.stock\_name = p.stock\_name

JOIN company\_details cd on cd.comp\_name = sd.comp\_name

WHERE cd.hq IN (

SELECT cd.hq

FROM CUSTOMER c2

JOIN bank\_acc ba on ba.cust\_id = c.cust\_id

JOIN wallet w on w.bank\_acc\_no = ba.acc\_no

JOIN stock\_acc sa on sa.acc\_no = ba.acc\_no

JOIN portfolio p on p.stock\_acc\_id = sa.stock\_id

JOIN stock\_details sd on sd.stock\_name = p.stock\_name

JOIN company\_details cd on cd.comp\_name = sd.comp\_name

GROUP BY cd.hq

HAVING SUM(ba.balance) < SUM(w.balance)

)

AND sd.growth > (SELECT AVG(sd2.growth) FROM STOCK\_DETAILS sd2 WHERE cd.comp\_name = sd2.stock\_name);

18. List the names of customers who have made transactions on stocks with a growth rate higher than the growth rate of stocks associated with companies headquartered in New York.

SELECT DISTINCT c.Name

FROM CUSTOMER c

JOIN bank\_acc b on b.cust\_id = c.cust\_id

JOIN stock\_acc s on s.acc\_no = b.acc\_no

JOIN portfolio p on s.stock\_id = p.stock\_acc\_id

JOIN stock\_details sd on sd.stock\_name = p.stock\_name

JOIN company\_details cd on cd.comp\_name = sd.comp\_name

WHERE cd.hq = 'New York' AND sd.growth > (SELECT AVG(growth) FROM STOCK\_DETAILS sd2 WHERE sd2.comp\_name = cd.comp\_name);

19. Retrieve the names of customers who have invested in mutual funds with a return percentage higher than the average return percentage of all mutual funds

SELECT DISTINCT c.Name

FROM CUSTOMER c

JOIN bank\_acc ba on ba.cust\_id = c.cust\_id

JOIN stock\_acc sa on sa.acc\_no = ba.acc\_no

JOIN mutual\_funds m on m.stock\_acc\_id = sa.stock\_id

WHERE m.return\_perc > (SELECT AVG(return\_perc) FROM MUTUAL\_FUNDS)

AND m.return\_perc > (SELECT AVG(return\_perc) FROM MUTUAL\_FUNDS WHERE return\_perc > 15);

20. List the name, email\_id and wallet balance of people who have sh in their name

select name, email\_id, w.balance from customer c

JOIN bank\_acc b on b.cust\_id = c.cust\_id

JOIN wallet w on w.bank\_acc\_no = b.acc\_no

where name like '%sh%'

21. List all the transaction IDs and company names done on the date 19-04-2024

SELECT trans\_id, comp\_name

FROM transaction\_history

WHERE date\_of\_trans= '04-19-2024'

22. List the portfolio details of 3 people living in california state

SELECT stock\_name, qty, purchasing\_price from portfolio p

JOIN stock\_acc s on p.stock\_acc\_id = s.stock\_id

JOIN bank\_acc b on s.acc\_no = b.acc\_no

JOIN customer c on c.cust\_id = b.cust\_id

WHERE state = 'California'

limit 3

23. Give the stock name having highest market price right now.

SELECT stock\_name, cur\_mp

FROM stock\_details

GROUP BY stock\_name

ORDER BY cur\_mp desc

limit 1

24. List the maximum dividend stock name from each exchange

SELECT w.exchange, w.stock\_name, w.dividends AS max\_dividend

FROM WATCHLIST w

INNER JOIN (

SELECT exchange, MAX(dividends) AS max\_dividend

FROM WATCHLIST

GROUP BY exchange

) max\_div ON w.exchange = max\_div.exchange AND w.dividends = max\_div.max\_dividend;

25. List the balance of bank account for the customer having demat id equal to 14400448

SELECT b.balance, c.name

FROM customer c

JOIN demat d on d.cust\_id = c.cust\_id

JOIN bank\_acc b on b.cust\_id = c.cust\_id

WHERE d.id = 14400448