Tasks 2:

1. Write a SQL query to retrieve the name, account type and email of all customers.

SELECT first_name, account_type, email from Customers c join Accounts a on a.customer_id = c.customer_id;

first_name	account_type	email
Alice	Savings	alice.steve@gmail.com
bob	Current	bob.ford@gmail.com
Cathy	Zero_balance	cathy.pant@gmail.com
David	Current	david.warner@gmail.com
Eoin	Zero_balance	eoin.morgan@gmail.com
Freddy	Savings	freddy.salt@gmail.com
George	Savings	george.antony@gmail.com
Harry	Current	harry.brook@gmail.com
Ian	Zero_balance	ian.bishop@gmail.com
Jack	Savings	jack.mcgreth@gmail.com

2. Write a SQL query to list all transaction corresponding customer.

SELECT c.first_name, c.last_name, t.transaction_type, t.amount, t.transaction_date FROM Customers c JOIN Accounts a ON c.customer_id = a.customer_id JOIN Transactions t ON a.account_id = t.account_id;

first_name	last_name	transaction_type	amount	transaction_date
Alice	Steve	deposit	1000.00	2025-01-10
bob	ford	withdrawal	250.00	2024-09-23
Cathy	Pant	transfer	5500.00	2025-03-18
David	Warner	withdrawal	700.00	2025-01-31
Eoin	Morgan	transfer	25000.00	2023-12-17
Freddy	Salt	deposit	50000.00	2022-08-15
George	Antony	deposit	9000.00	2023-06-13
Harry	Brook	withdrawal	1600.00	2024-09-01
Ian	Bishop	transfer	400.00	2025-02-14
Jack	Mcgreth	deposit	75000.00	2023-07-07

Tasks 3: Aggregate functions, Having, Order By, GroupBy and Joins:

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

Select Avg(balance) as average_balance from accounts;

average_balance
82360.000000

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

Select account_id, balance FROM accounts order by balance desc;

account_id	balance
2226	500000.00
2230	73000.00
2227	65000.00
2225	54000.00
2222	50000.00
2224	30000.00
2221	22000.00
2228	12000.00
2229	9800.00
2223	7800.00

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

SELECT SUM(amount) AS total_deposits FROM Transactions WHERE transaction_type = 'deposit' AND transaction_date = '2023-06-13';

total_deposits
9000.00

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

SELECT min(DOB) as oldest_customer, max(dob) as newest_customer from Customers;

oldest_customer	newest_customer
2000-05-09	2003-12-09

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

SELECT t.* , a.account_type from Transactions t join accounts a on t.account_id = a.account_id;

transaction_id	account_id	transaction_type	amount	transaction_date	account_type
5001	2221	deposit	1000.00	2025-01-10	Savings
5002	2222	withdrawal	250.00	2024-09-23	Current
5003	2223	transfer	5500.00	2025-03-18	Zero_balance
5004	2224	withdrawal	700.00	2025-01-31	Current
5005	2225	transfer	25000.00	2023-12-17	Zero_balance
5006	2226	deposit	50000.00	2022-08-15	Savings
5007	2227	deposit	9000.00	2023-06-13	Savings
5008	2228	withdrawal	1600.00	2024-09-01	Current
5009	2229	transfer	400.00	2025-02-14	Zero_balance
5010	2230	deposit	75000.00	2023-07-07	Savings

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

SELECT first_name, a.* from accounts a join Customers c on c.customer_id = a.customer_id;

first_name	account_id	customer_id	account_type	balance
Alice	2221	101	Savings	22000.00
bob	2222	102	Current	50000.00
Cathy	2223	103	Zero_balance	7800.00
David	2224	104	Current	30000.00
Eoin	2225	105	Zero_balance	54000.00
Freddy	2226	106	Savings	500000.00
George	2227	107	Savings	65000.00
Harry	2228	108	Current	12000.00
Ian	2229	109	Zero_balance	9800.00
Jack	2230	110	Savings	73000.00

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

SELECT t.*, first_name from Customers c join Accounts a on a.customer_id = c.customer_id join Transactions t on t.account_id = a.account_id where a.account_id = 2221;

transaction_id	account_id	transaction_type	amount	transaction_date	first_name
5001	2221	deposit	1000.00	2025-01-10	Alice

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

SELECT customer_id, COUNT(account_id) AS num_accounts FROM accounts GROUP BY customer_id HAVING COUNT(account_id) > 1;

customer_id	num_accounts

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

SELECT SUM(CASE WHEN transaction_type = 'deposit' THEN amount ELSE 0 END) - SUM(CASE WHEN transaction_type = 'withdrawal' THEN amount ELSE 0 END) AS difference FROM Transactions;

difference
132450.00

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

select account_id,AVG(amount) as average_daily_balance from Transactions where transaction_date BETWEEN '2025-01-10' AND '2025-01-31' group by account_id;

account_id	average_daily_balance
2221	1000.000000
2224	700.000000

11. Calculate the total balance for each account type.

SELECT account_type, SUM(balance) AS total_balance from Accounts GROUP BY account_type;

account_type	total_balance
Savings	660000.00
Current	92000.00
Zero_balance	71600.00

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

SELECT account_id, COUNT(transaction_id) AS transaction_count FROM Transactions GROUP BY account_id ORDER BY transaction_count DESC;

account_id	transaction_count
2221	1
2222	1
2223	1
2224	1
2225	1
2226	1
2227	1
2228	1
2229	1
2230	1

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

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,c.first_name,c.last_name, a.account_type HAVING SUM(a.balance) > 10000 ORDER BY total_balance DESC;

customer_id	first_name	last_name	account_type	total_balance
106	Freddy	Salt	Savings	500000.00
110	Jack	Mcgreth	Savings	73000.00
107	George	Antony	Savings	65000.00
105	Eoin	Morgan	Zero_balance	54000.00
102	bob	ford	Current	50000.00
104	David	Warner	Current	30000.00
101	Alice	Steve	Savings	22000.00
108	Harry	Brook	Current	12000.00

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

SELECT t.account_id, t.amount, t.transaction_date, COUNT(*) AS duplicate_count FROM Transactions t

GROUP BY t.account_id, t.amount, t.transaction_date HAVING COUNT(*) > 1 ORDER BY t.transaction_date, t.account_id;

account_id	amount	transaction_date	duplicate_count