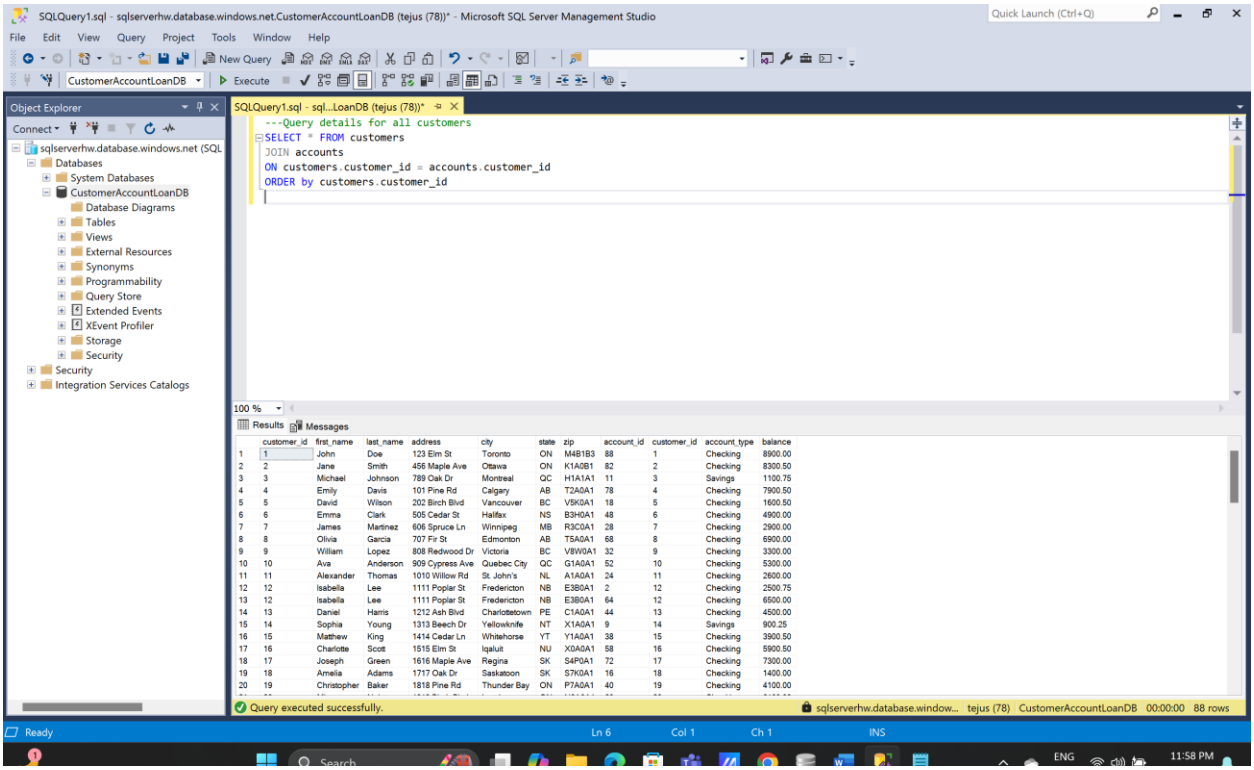
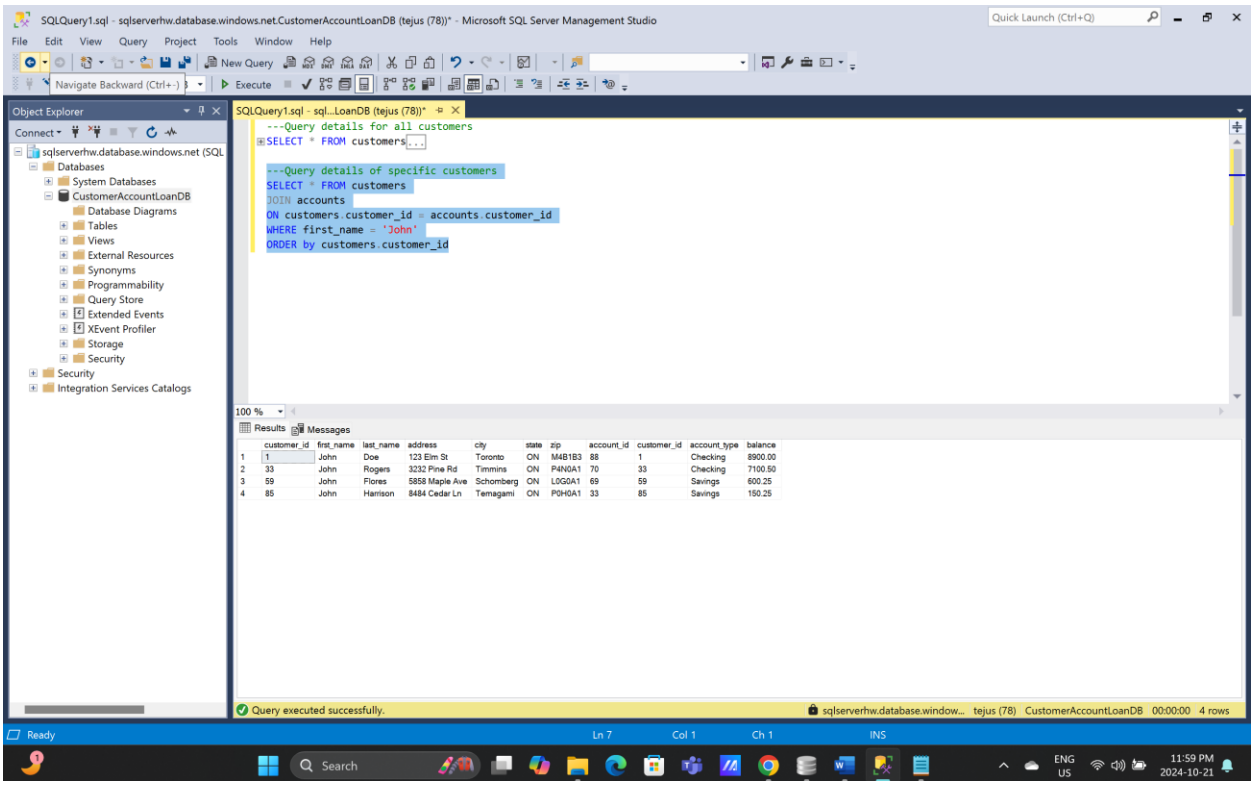
1) ---Query details for all customers

```
SELECT * FROM customers
JOIN accounts
ON customers.customer_id = accounts.customer_id
ORDER by customer_id
```



2) ---Query details of specific customers

```
SELECT * FROM customers
JOIN accounts
ON customers.customer_id = accounts.customer_id
WHERE first_name = 'John'
ORDER by customers.customer_id
```



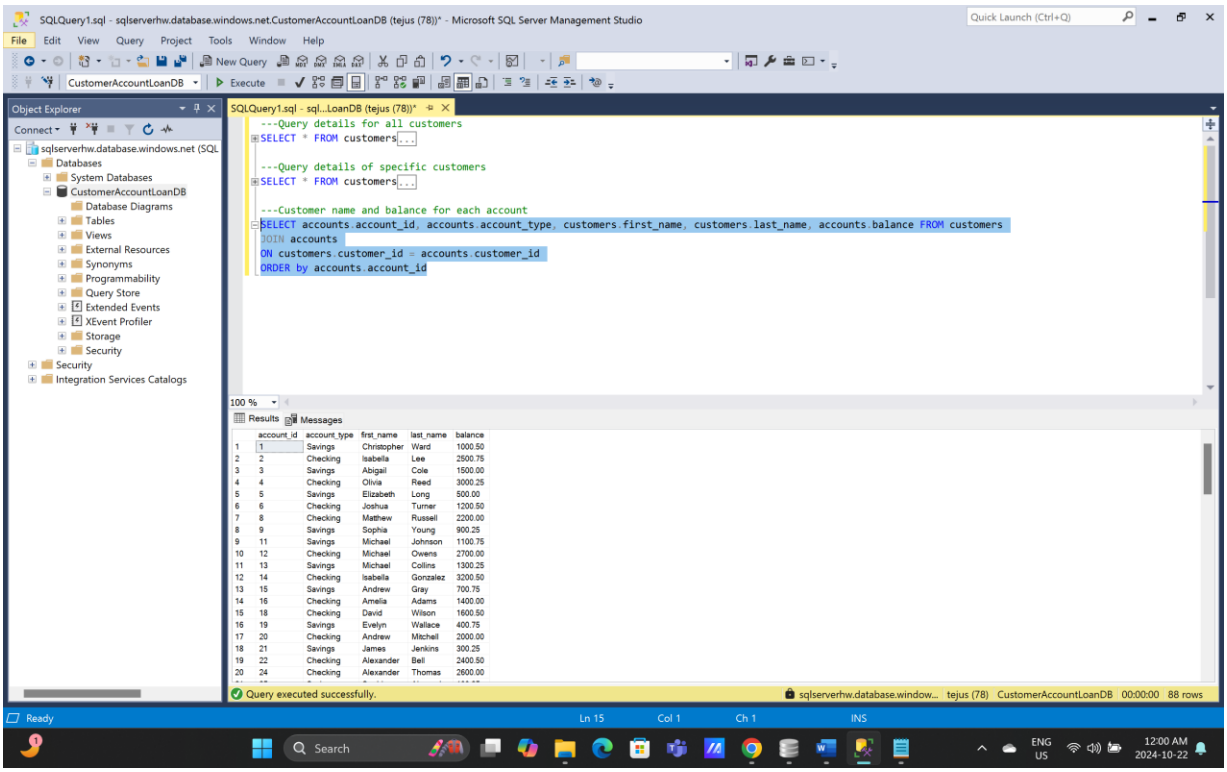
3) ---Customer name and balance for each account

SELECT accounts.account_id, accounts.account_type, customers.first_name, customers.last_name, accounts.balance FROM customers

JOIN accounts

ON customers.customer_id = accounts.customer_id

ORDER by accounts.account_id



4) ---Analyze customer loan balances

SELECT c.customer_id,c.first_name,c.last_name, l.loan_id, l.loan_amount, l.interest_rate, l.loan_term, (l.loan_amount+((l.loan_amount*l.interest_rate*(l.loan_term/12))/100)) as [repayment_amount], lp.payment_amount, lp.payment_date, ((l.loan_amount+((l.loan_amount*l.interest_rate*(l.loan_term/12))/100))-lp.payment_amount) as [loan_balance] FROM customers as c

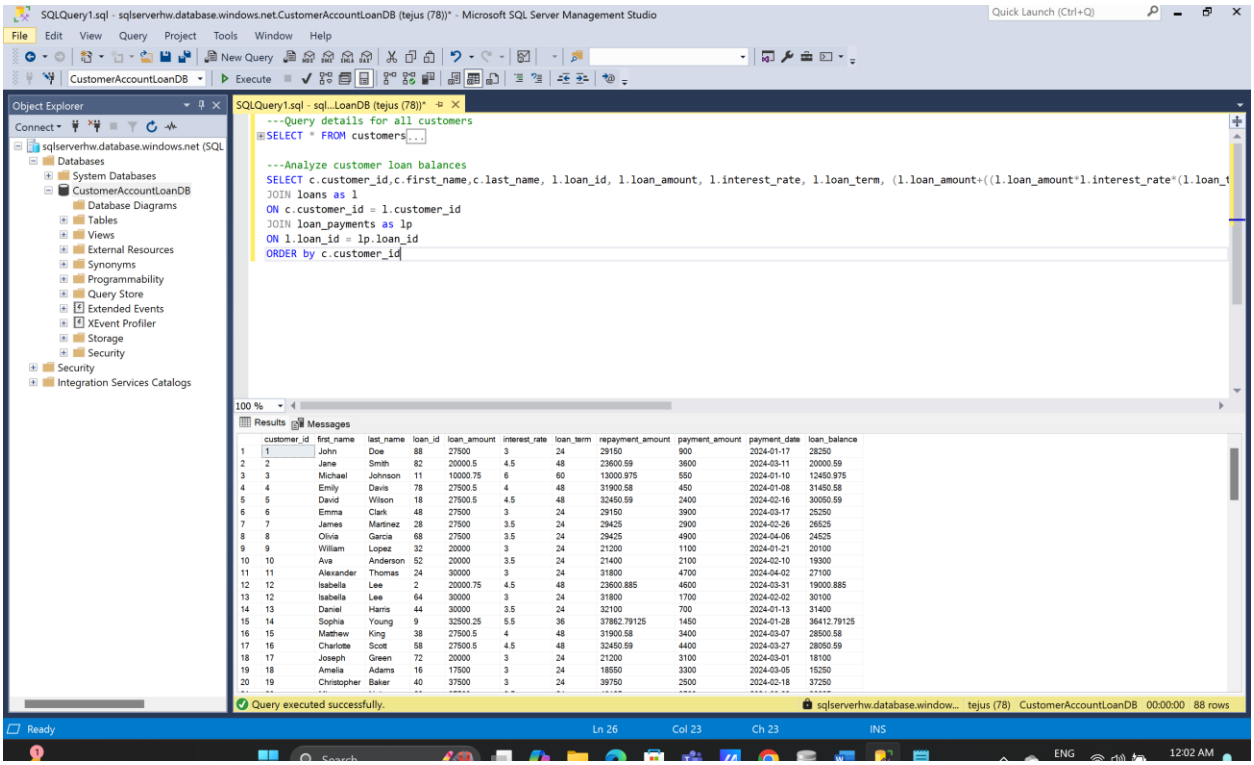
JOIN loans as l

ON c.customer_id = l.customer_id

JOIN loan_payments as lp

ON l.loan_id = lp.loan_id

ORDER by c.customer_id



5) --All customers with transactions in March 2024

SELECT customers.customer_id, customers.first_name, customers.last_name, transactions.transaction_date FROM customers

JOIN accounts

ON customers.customer_id = accounts.customer_id

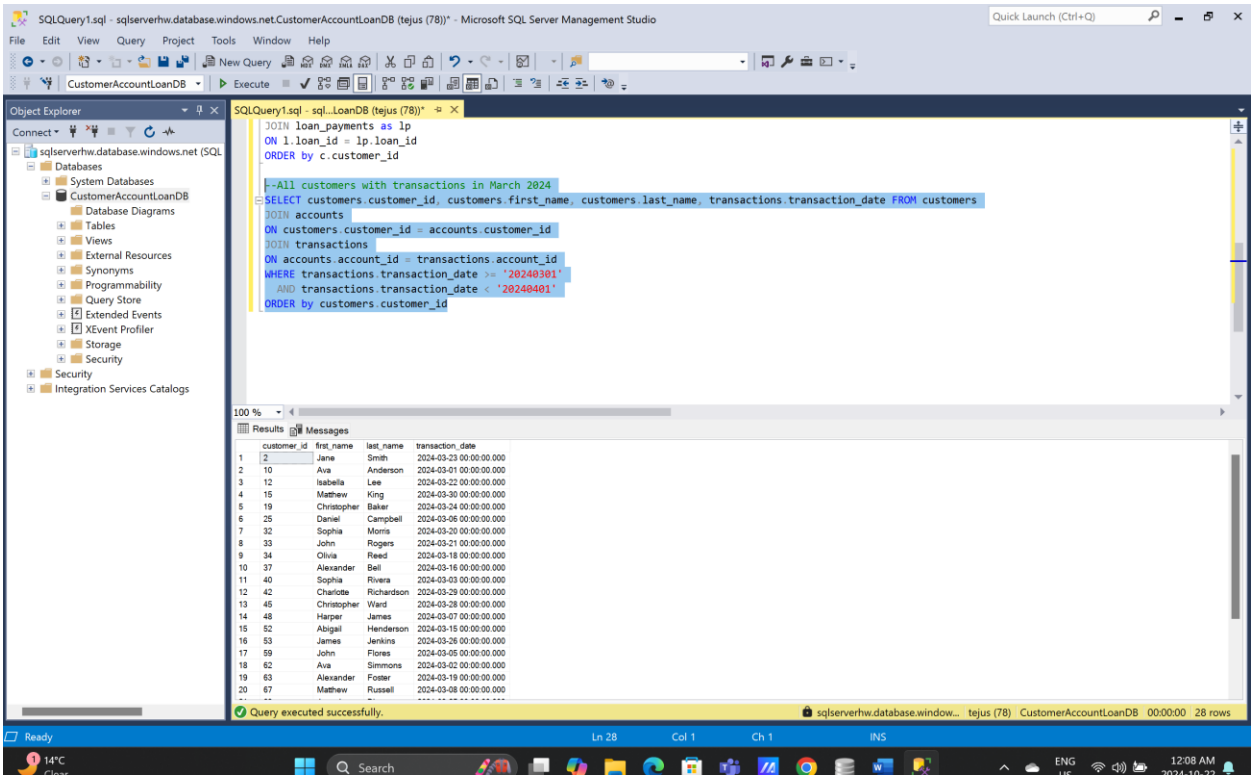
JOIN transactions

ON accounts.account_id = transactions.account_id

WHERE transactions.transaction_date >= '20240301'

AND transactions.transaction_date < '20240401'

ORDER by customers.customer_id



1) --Calculate the total balance across all accounts for each customer

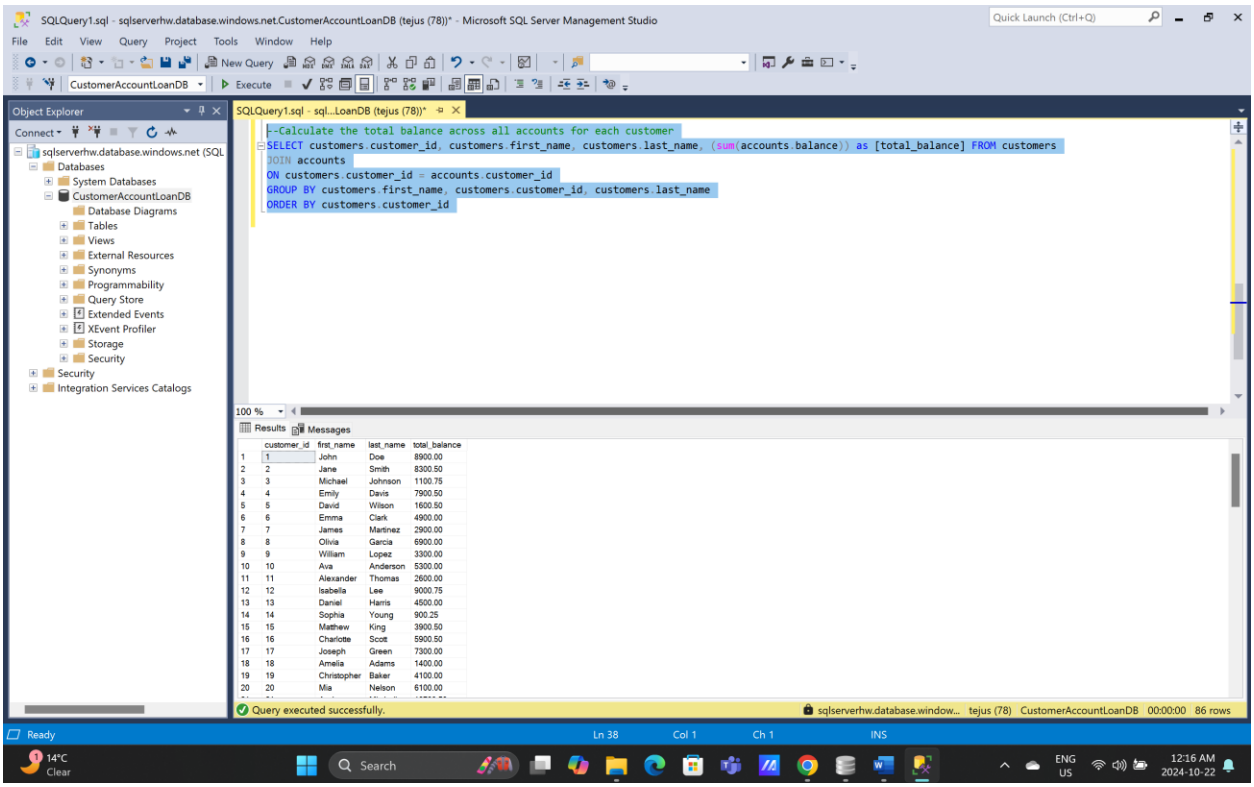
SELECT customers.customer_id, customers.first_name, customers.last_name, (sum(accounts.balance)) as [total_balance] FROM customers

JOIN accounts

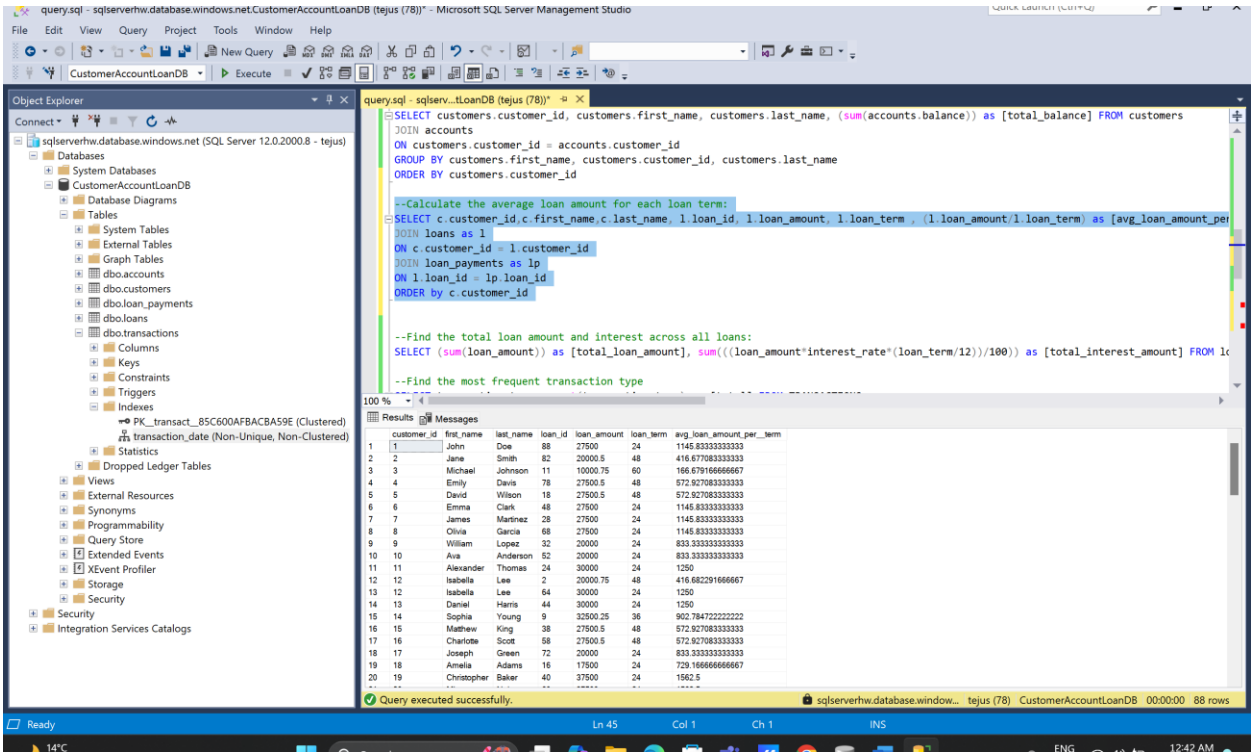
ON customers.customer_id = accounts.customer_id

GROUP BY customers.first_name, customers.customer_id, customers.last_name

ORDER BY customers.customer_id

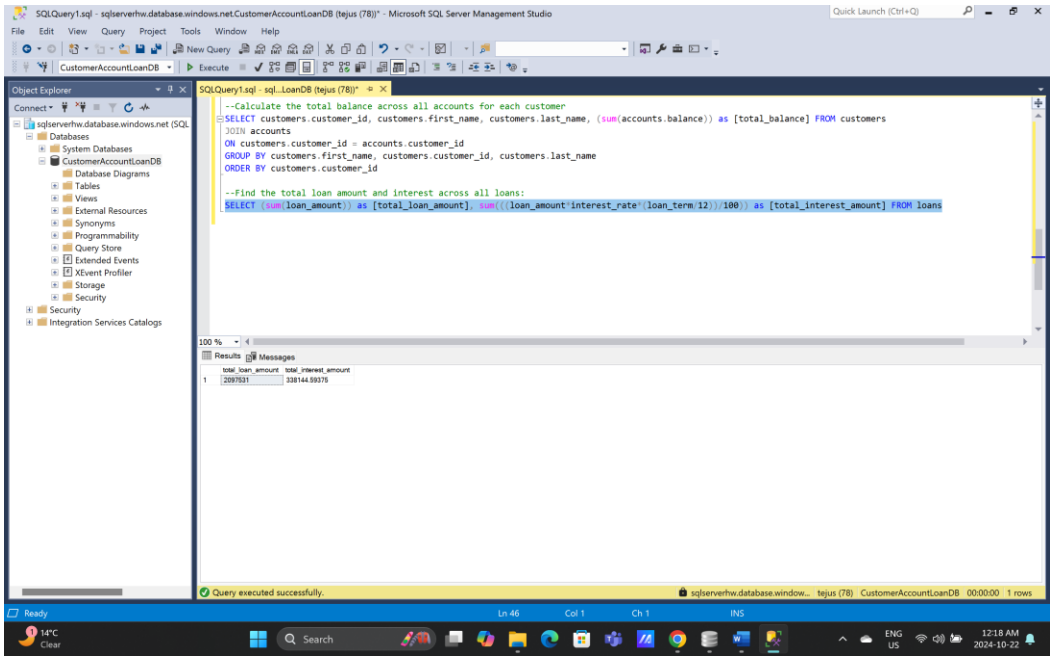


2) --Calculate the average loan amount for each loan term:
SELECT c.customer_id,c.first_name,c.last_name, l.loan_id, l.loan_amount, l.loan_term ,
(l.loan_amount/l.loan_term) as [avg_loan_amount_per__term] FROM customers as c
JOIN loans as l
ON c.customer_id = l.customer_id
JOIN loan_payments as lp
ON l.loan_id = lp.loan_id
ORDER by c.customer_id



3) --Find the total loan amount and interest across all loans:

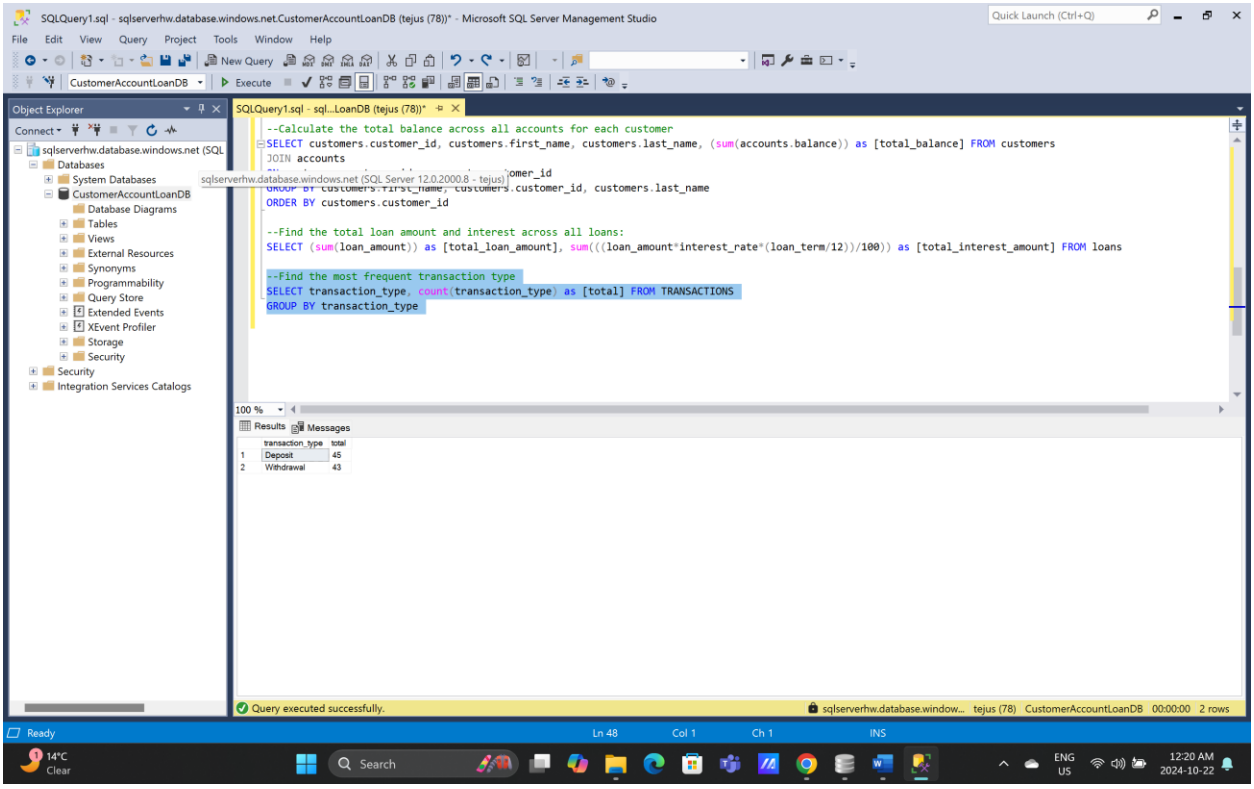
SELECT (sum(loan_amount)) as [total_loan_amount],
sum(((loan_amount*interest_rate*(loan_term/12))/100)) as [total_interest_amount] FROM loans



4) --Find the most frequent transaction type

SELECT transaction_type, count(transaction_type) as [total] FROM TRANSACTIONS

GROUP BY transaction_type



5) --Analyze transactions by account and transaction type:

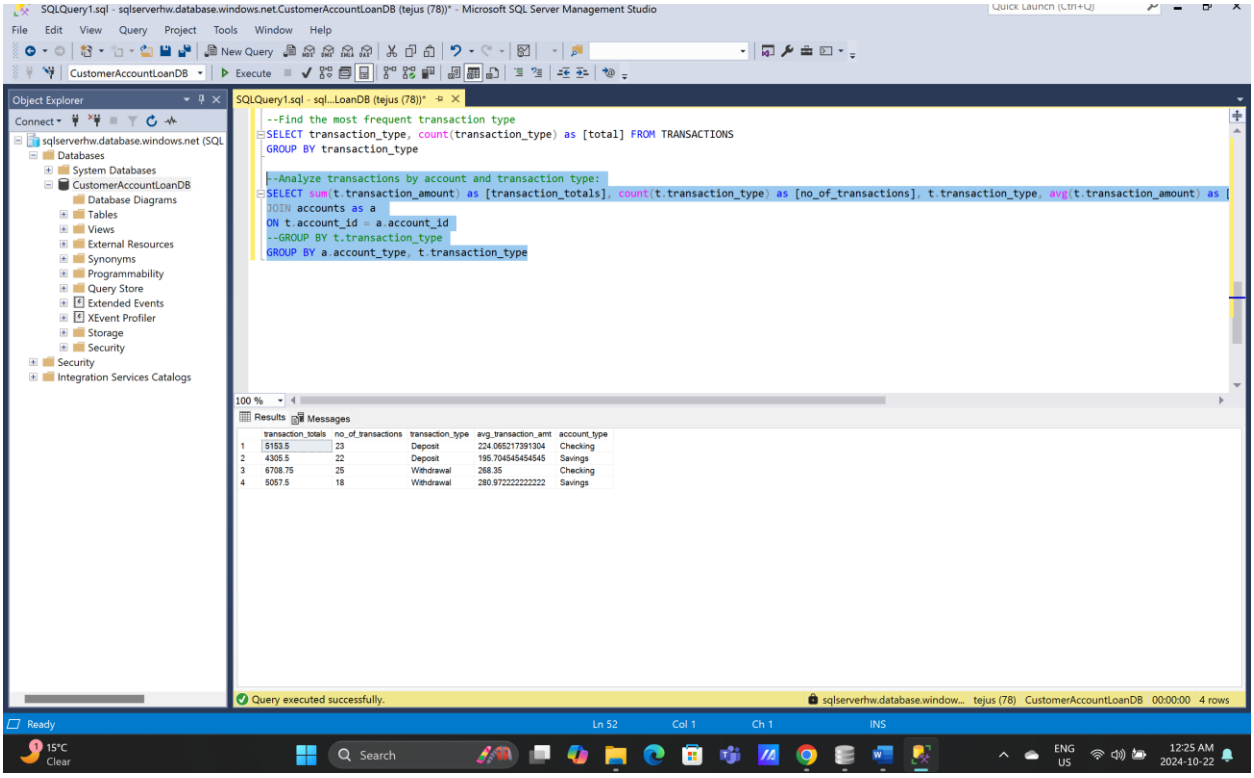
SELECT sum(t.transaction_amount) as [transaction_totals], count(t.transaction_type) as [no_of_transactions], t.transaction_type, avg(t.transaction_amount) as [avg_transaction_amt], a.account_type FROM TRANSACTIONS as t

JOIN accounts as a

ON t.account_id = a.account_id

--GROUP BY t.transaction_type

GROUP BY a.account_type, t.transaction_type



1) --Create a view of active loans with payments greater than $1000:

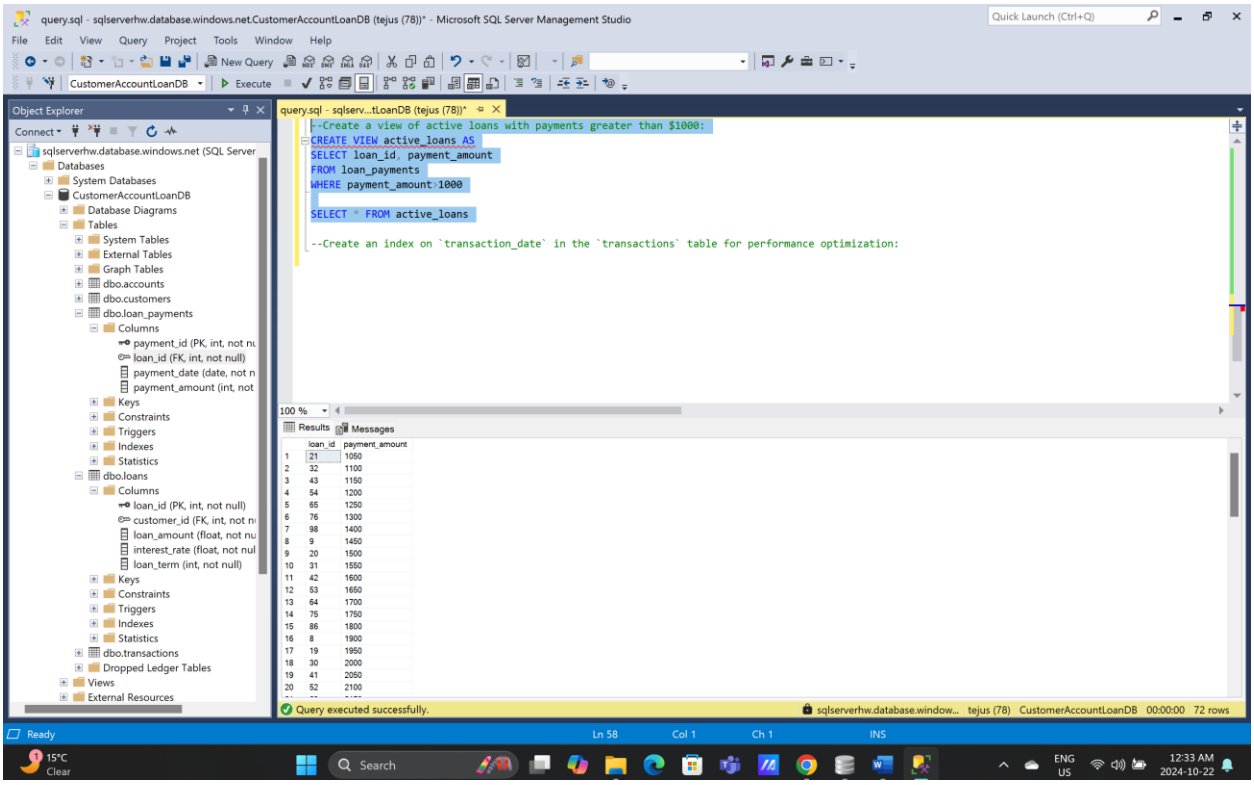
CREATE VIEW active_loans AS

SELECT loan_id, payment_amount

FROM loan_payments

WHERE payment_amount>1000

SELECT * FROM active_loans



2) --Create an index on `transaction_date` in the `transactions` table for performance optimization:

CREATE INDEX transaction_date

ON transactions (transaction_date)