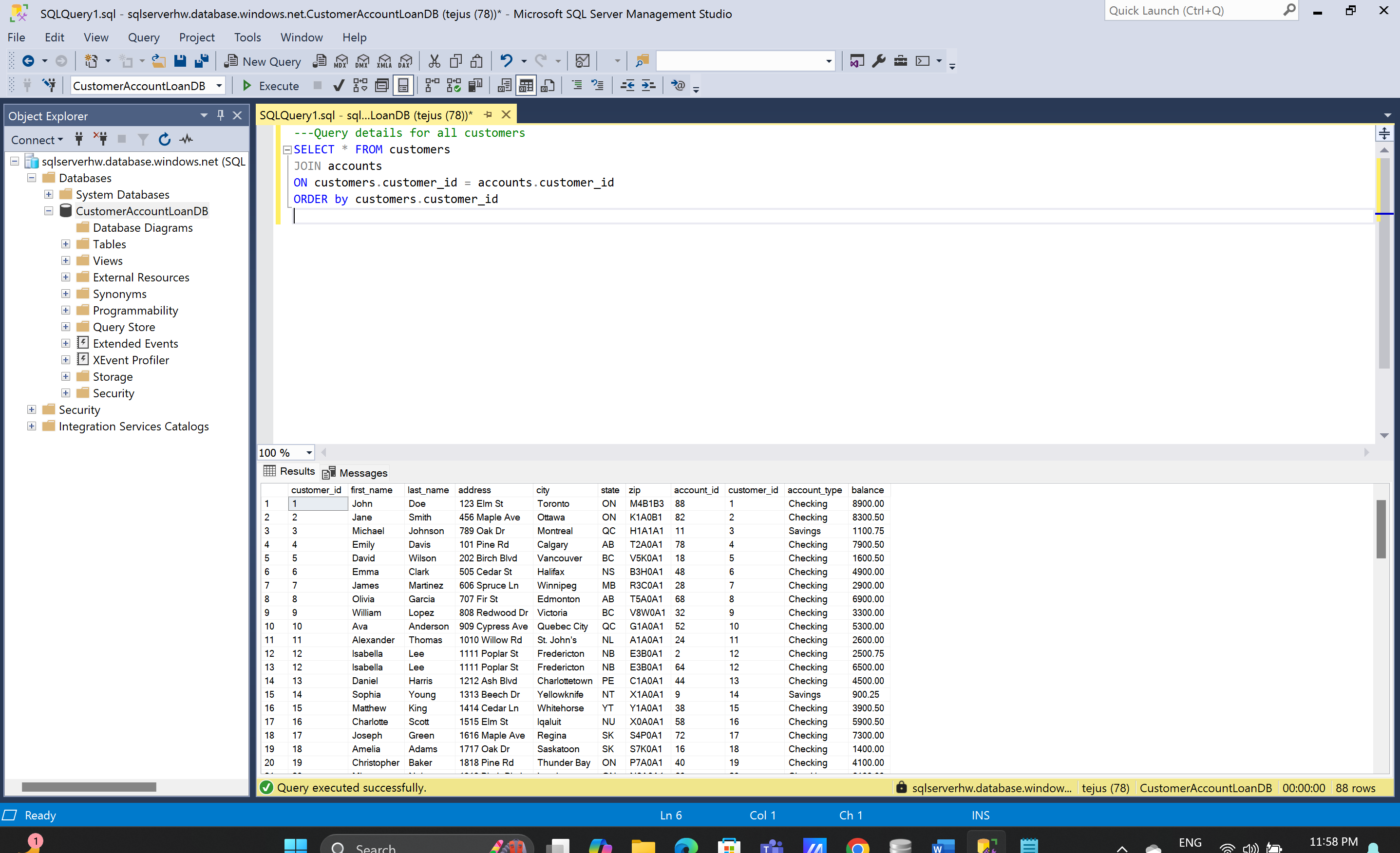
1. ---Query details for all customers

SELECT \* FROM customers

JOIN accounts

ON customers.customer\_id = accounts.customer\_id

ORDER by customer\_id



1. ---Query details of specific customers

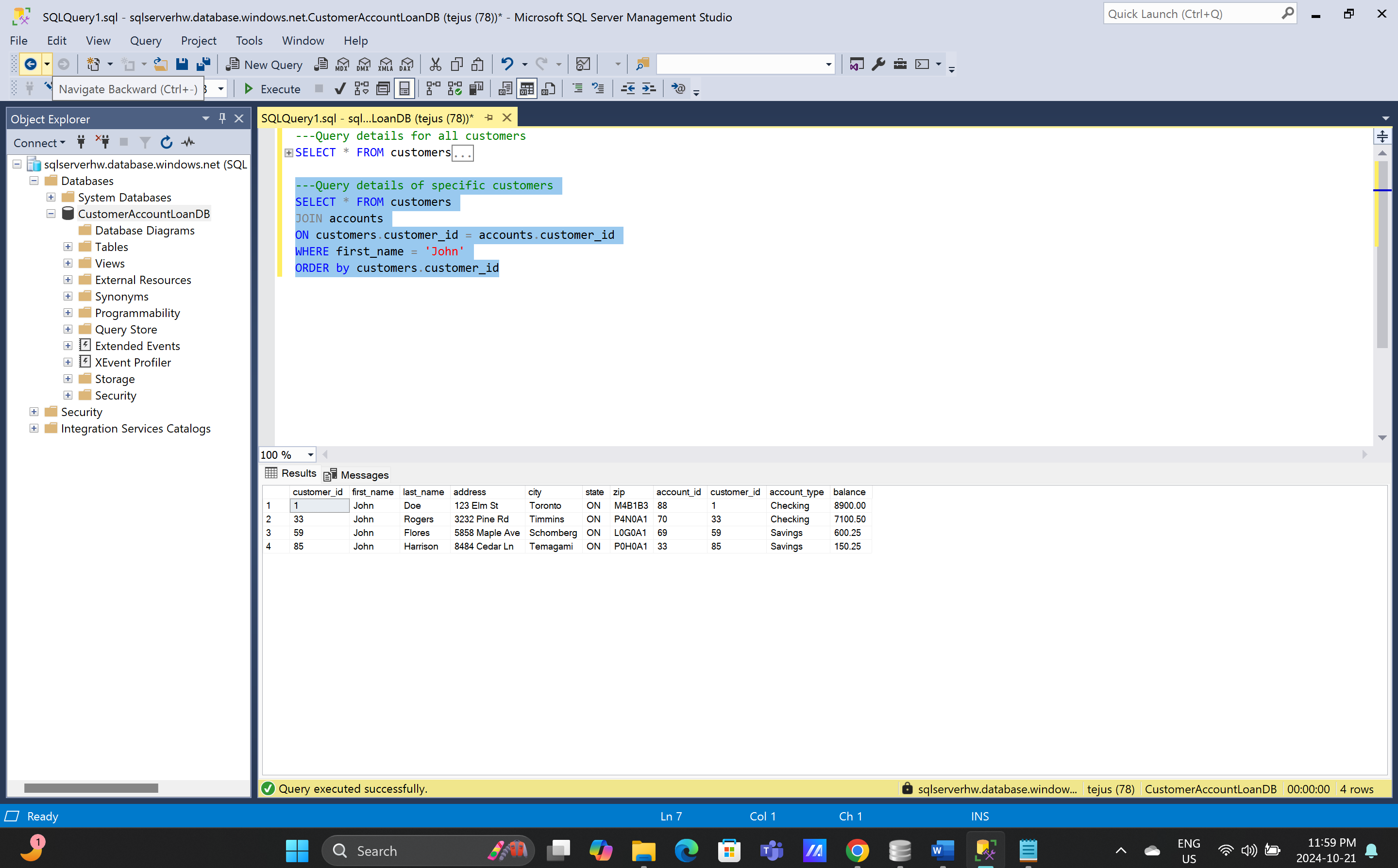
SELECT \* FROM customers

JOIN accounts

ON customers.customer\_id = accounts.customer\_id

WHERE first\_name = 'John'

ORDER by customers.customer\_id



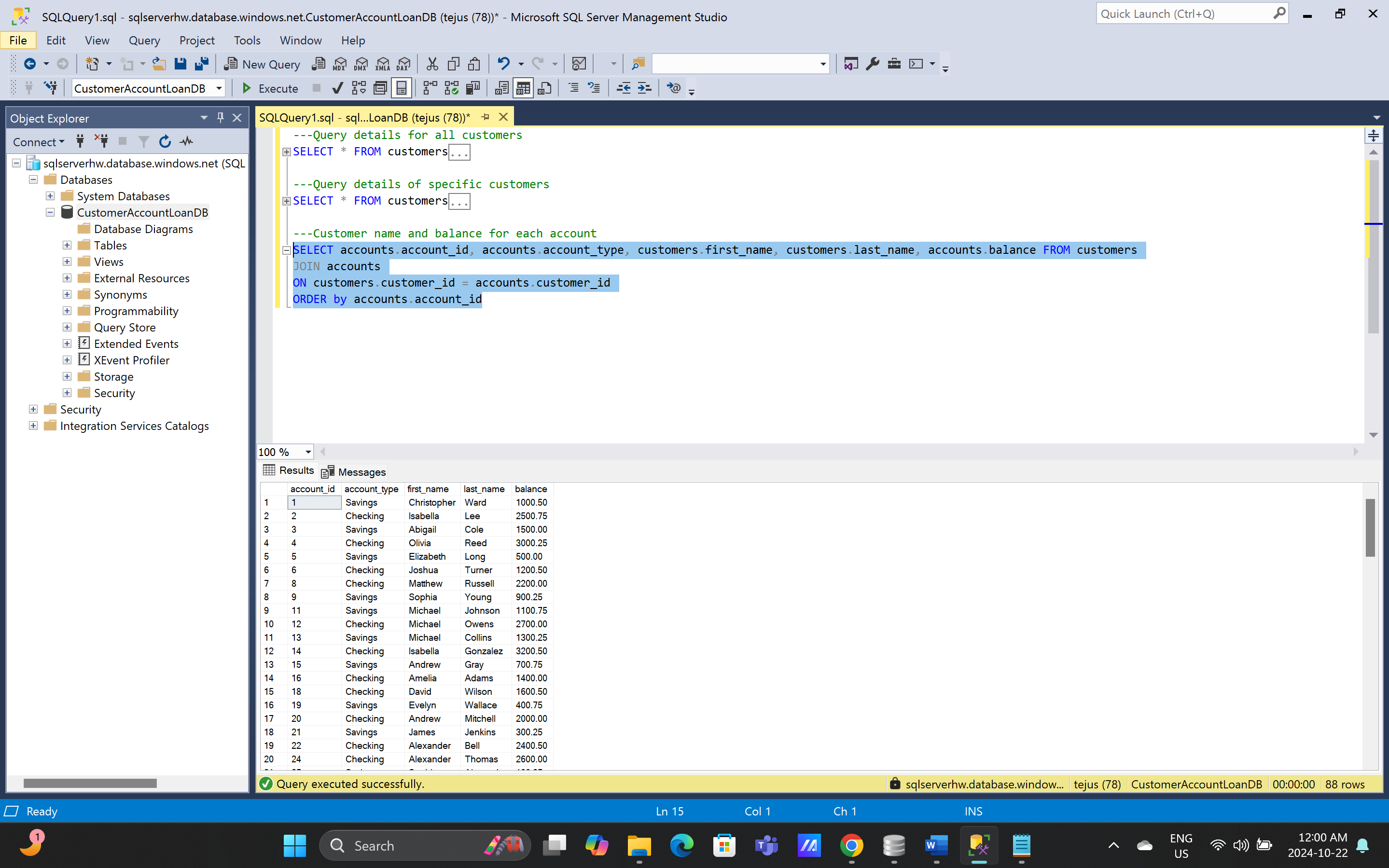
1. ---Customer name and balance for each account

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



1. ---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

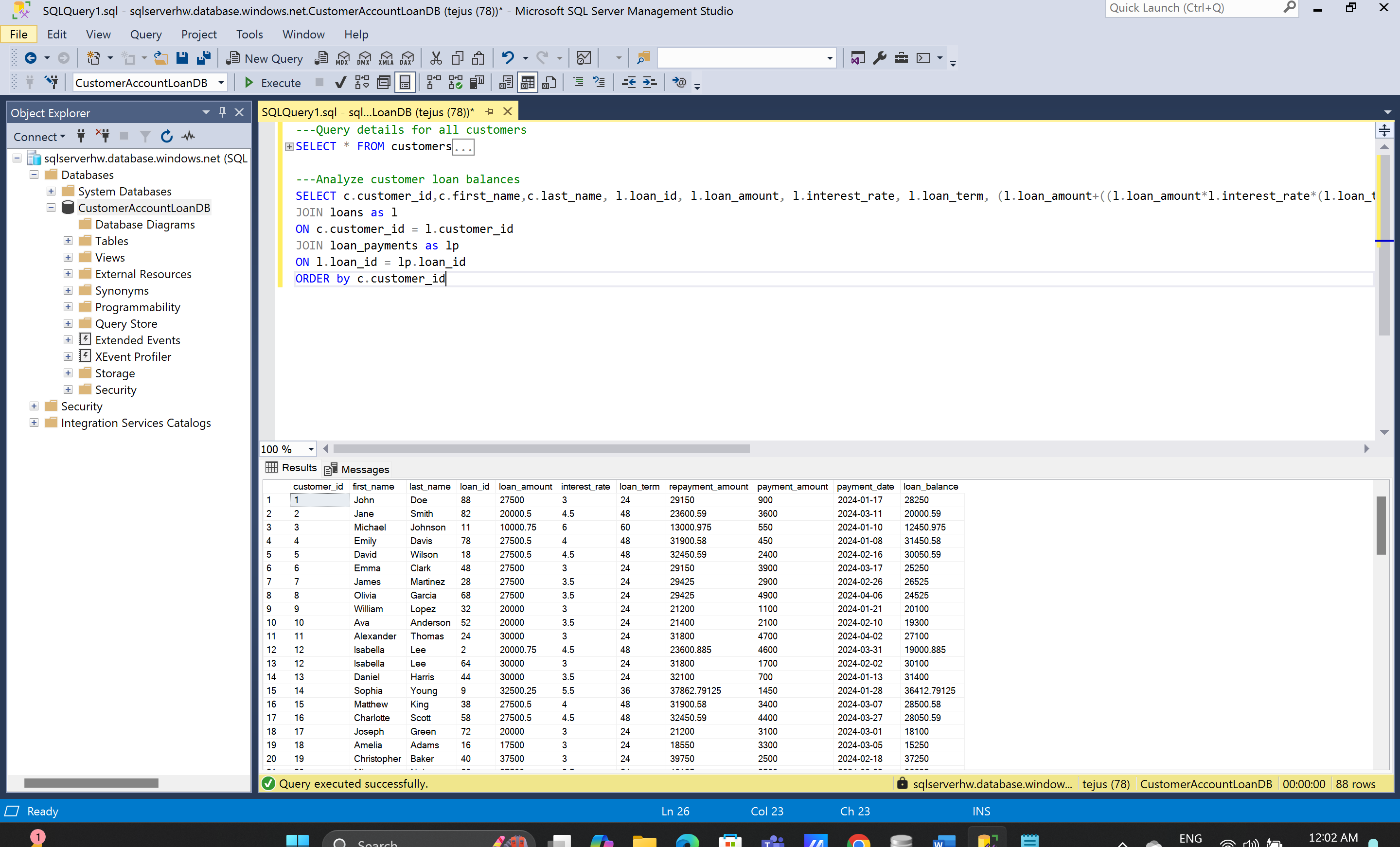
JOIN loans as l

ON c.customer\_id = l.customer\_id

JOIN loan\_payments as lp

ON l.loan\_id = lp.loan\_id

ORDER by c.customer\_id



1. --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

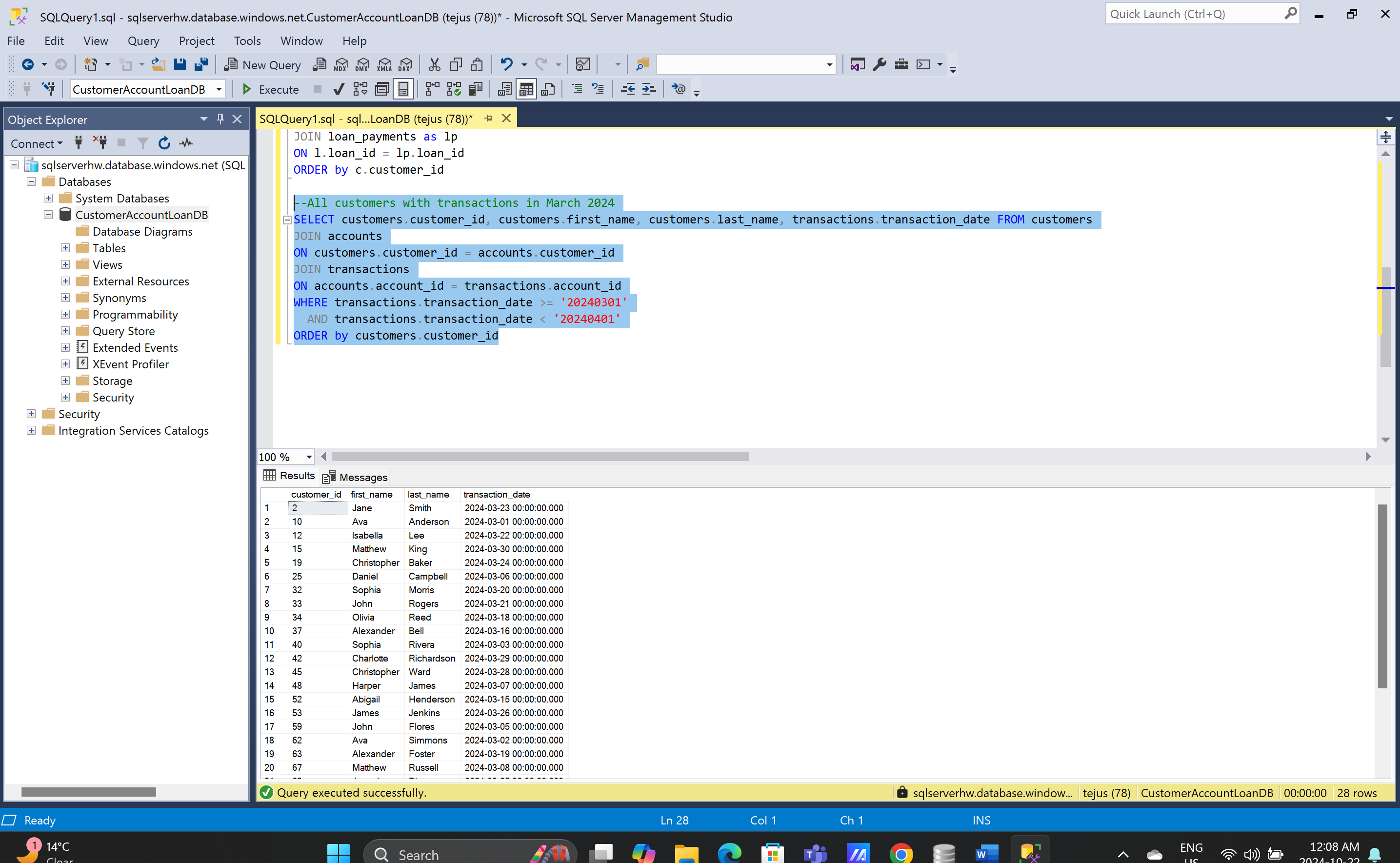
JOIN transactions

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

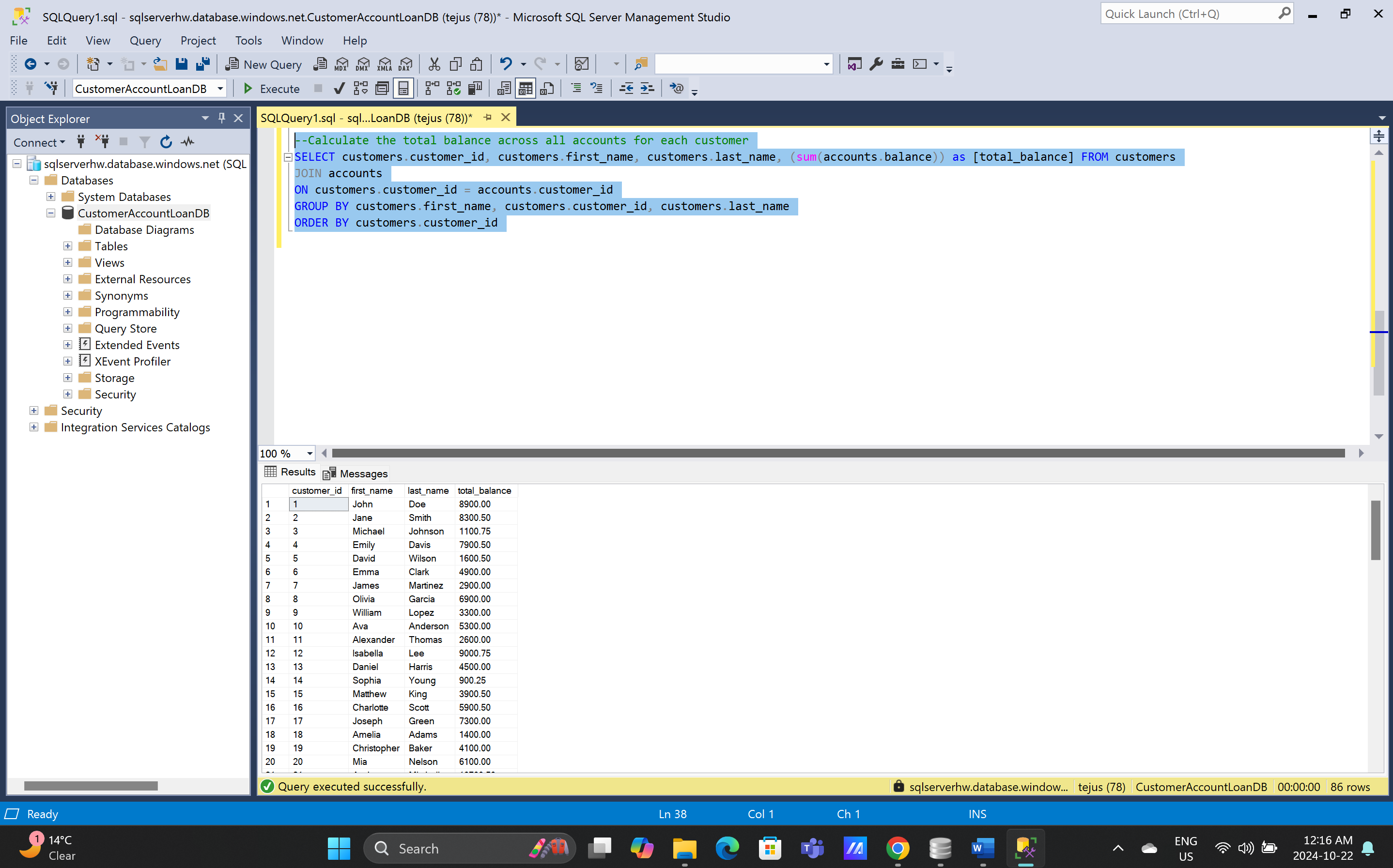
SELECT customers.customer\_id, customers.first\_name, customers.last\_name, (sum(accounts.balance)) as [total\_balance] FROM customers

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



1. --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

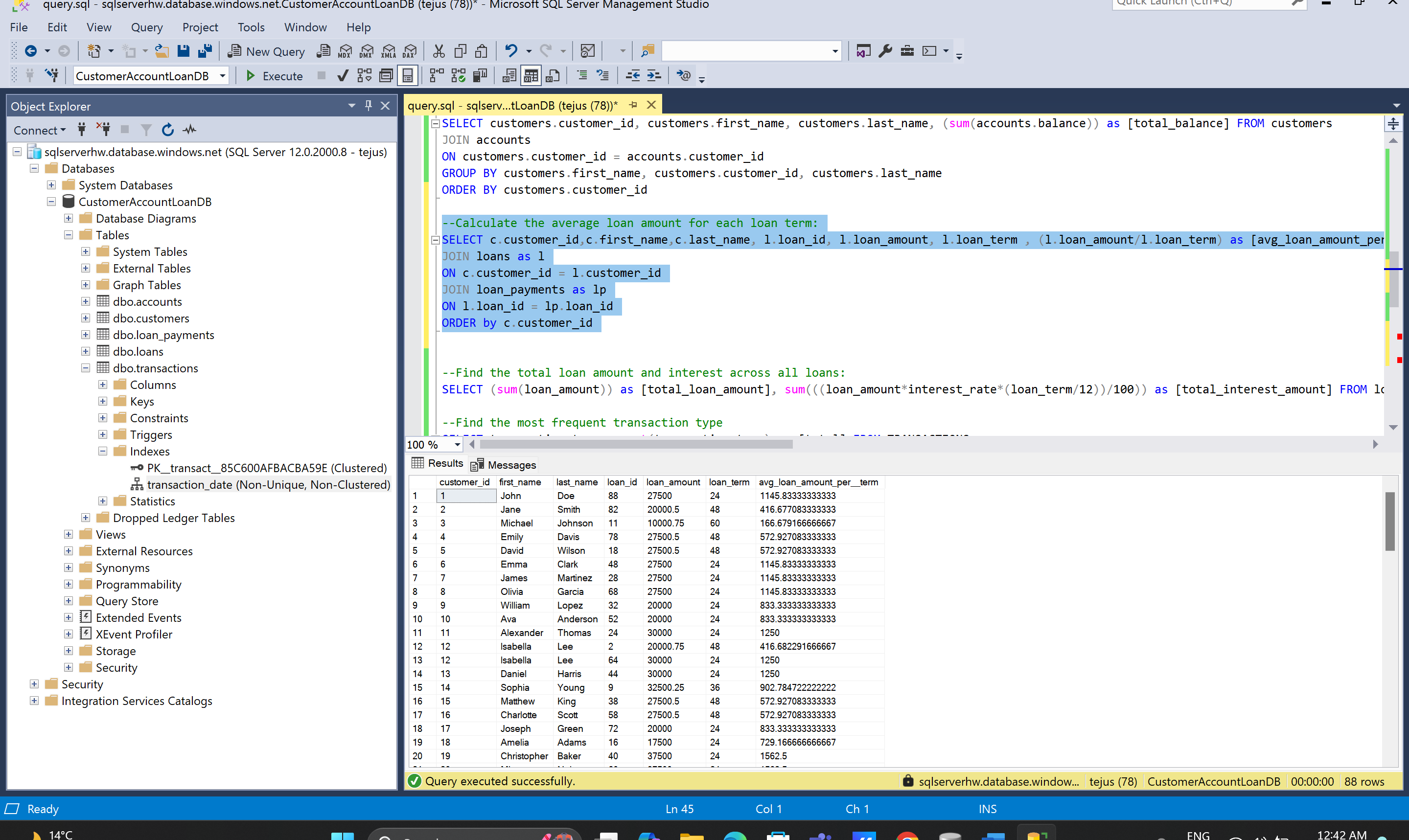
JOIN loans as l

ON c.customer\_id = l.customer\_id

JOIN loan\_payments as lp

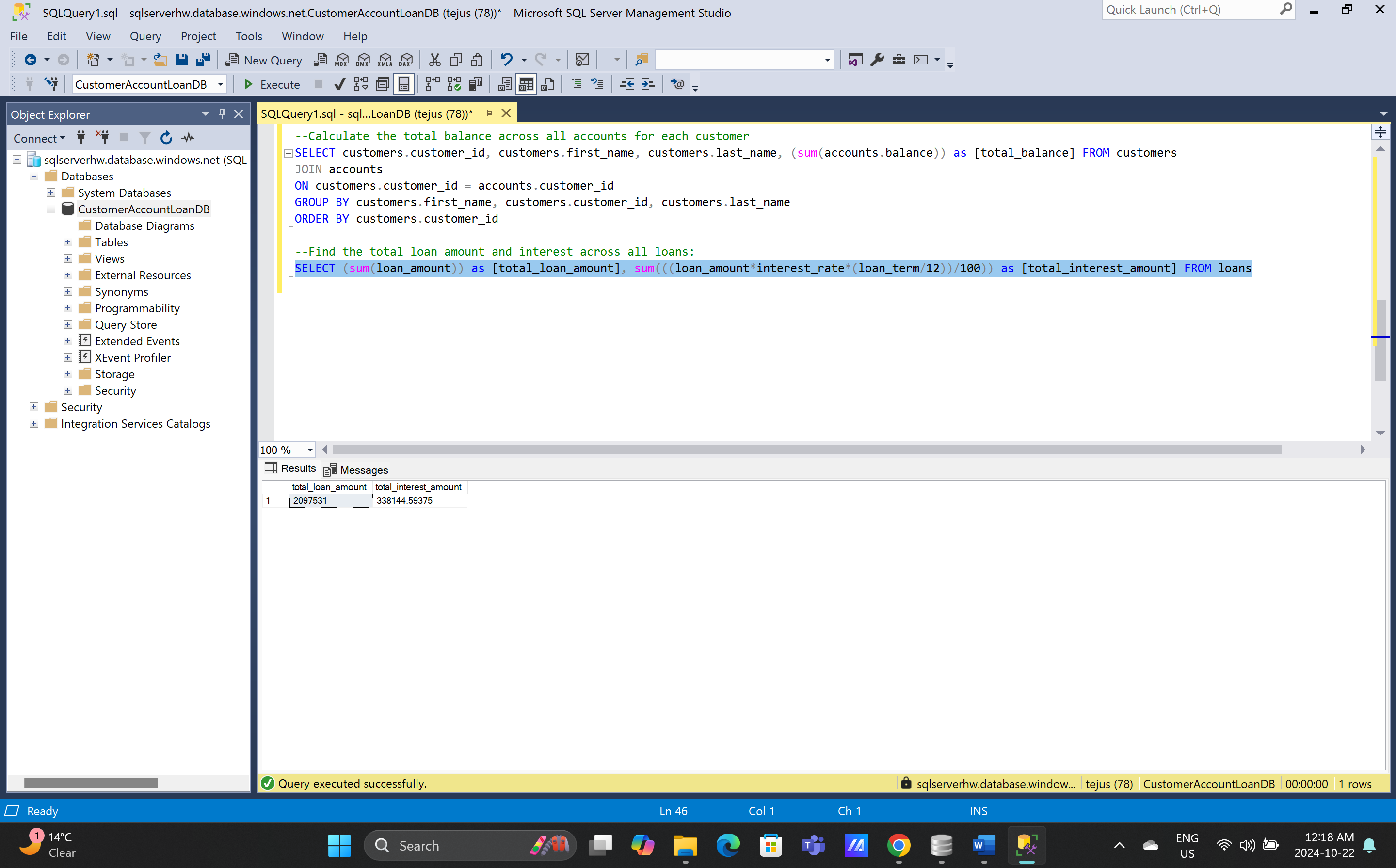
ON l.loan\_id = lp.loan\_id

ORDER by c.customer\_id



1. --Find the total loan amount and interest across all loans:

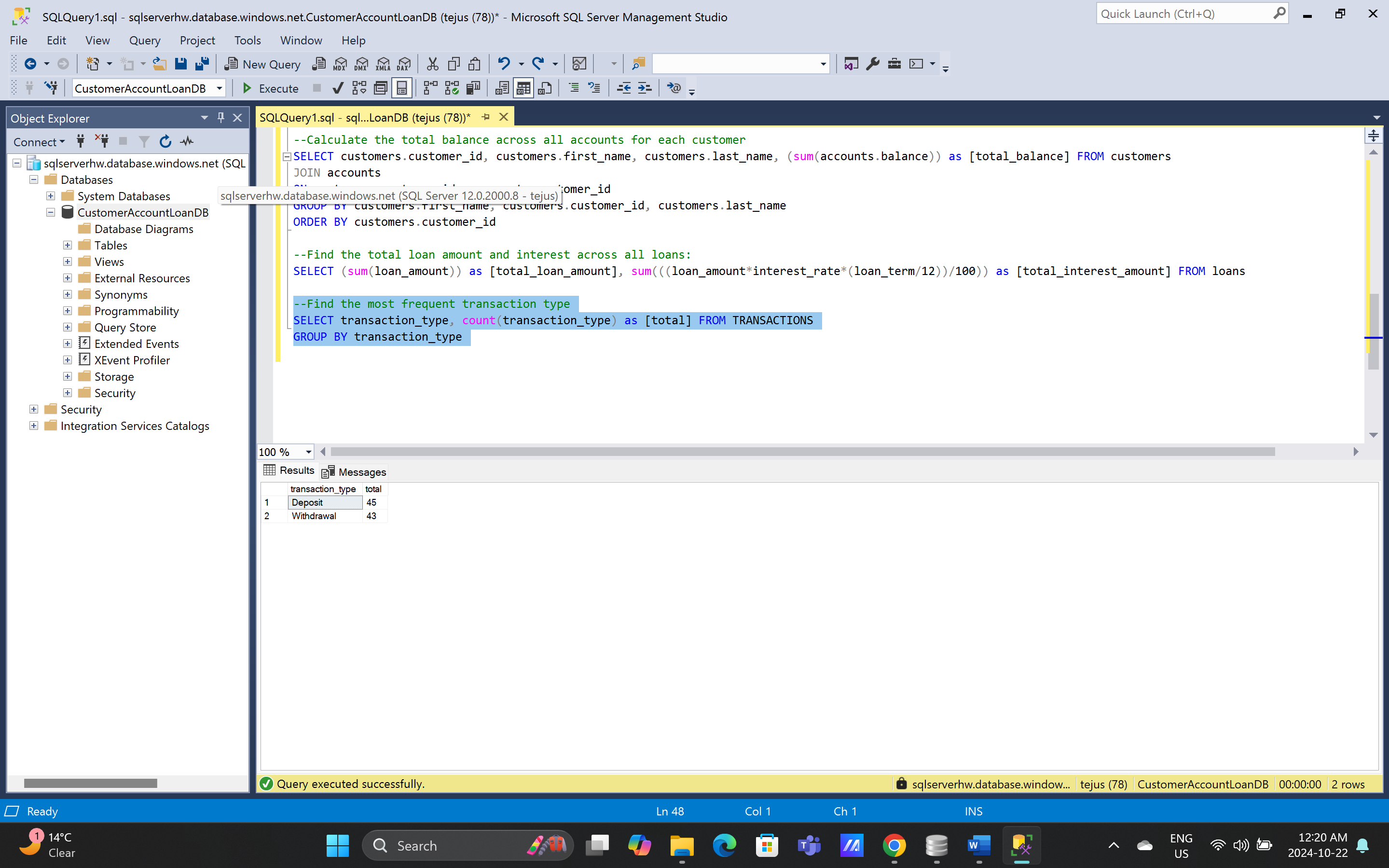
SELECT (sum(loan\_amount)) as [total\_loan\_amount], sum(((loan\_amount\*interest\_rate\*(loan\_term/12))/100)) as [total\_interest\_amount] FROM loans



1. --Find the most frequent transaction type

SELECT transaction\_type, count(transaction\_type) as [total] FROM TRANSACTIONS

GROUP BY transaction\_type



1. --Analyze transactions by account and transaction type:

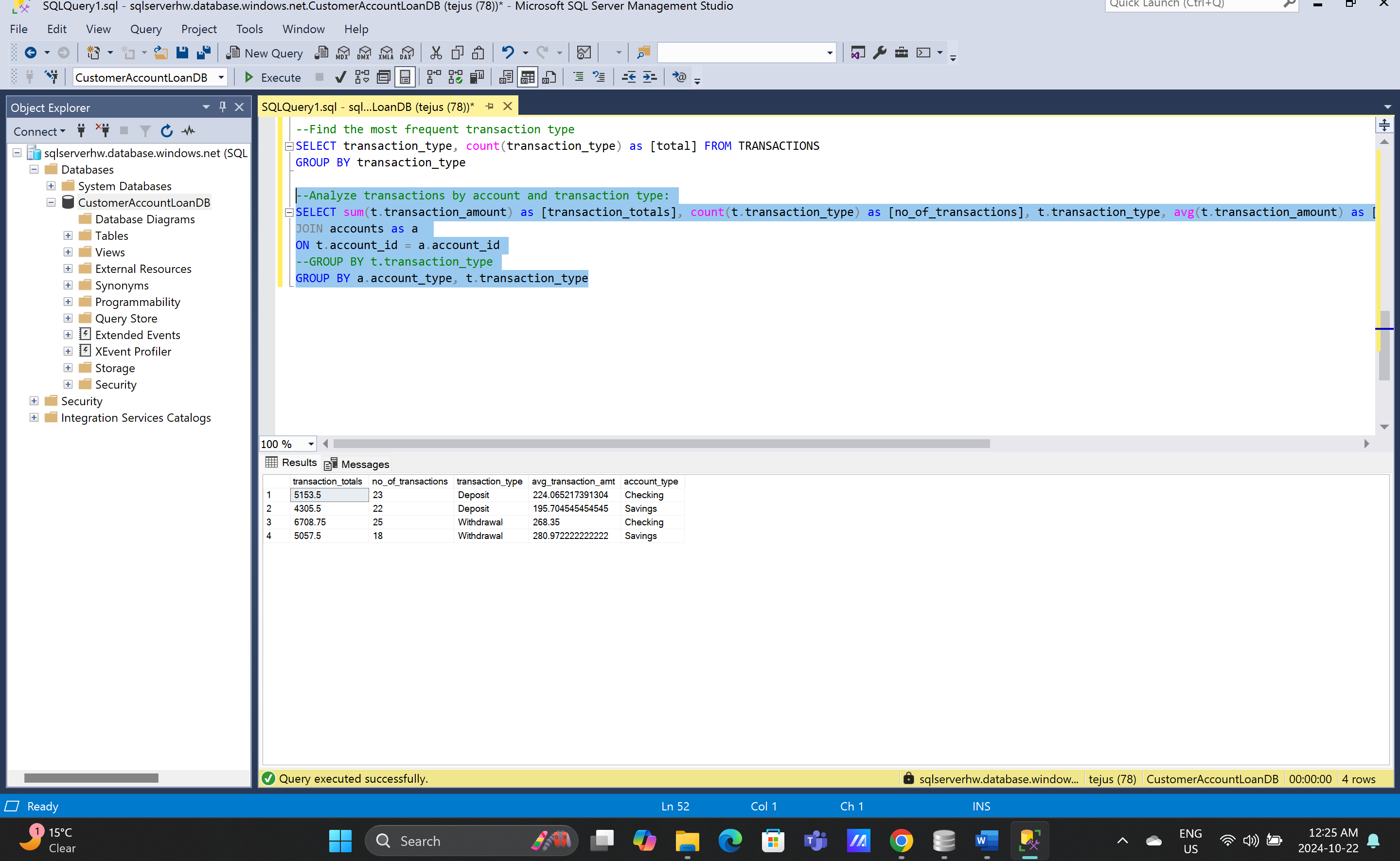
SELECT sum(t.transaction\_amount) as [transaction\_totals], count(t.transaction\_type) as [no\_of\_transactions], t.transaction\_type, avg(t.transaction\_amount) as [avg\_transaction\_amt], a.account\_type FROM TRANSACTIONS as t

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:

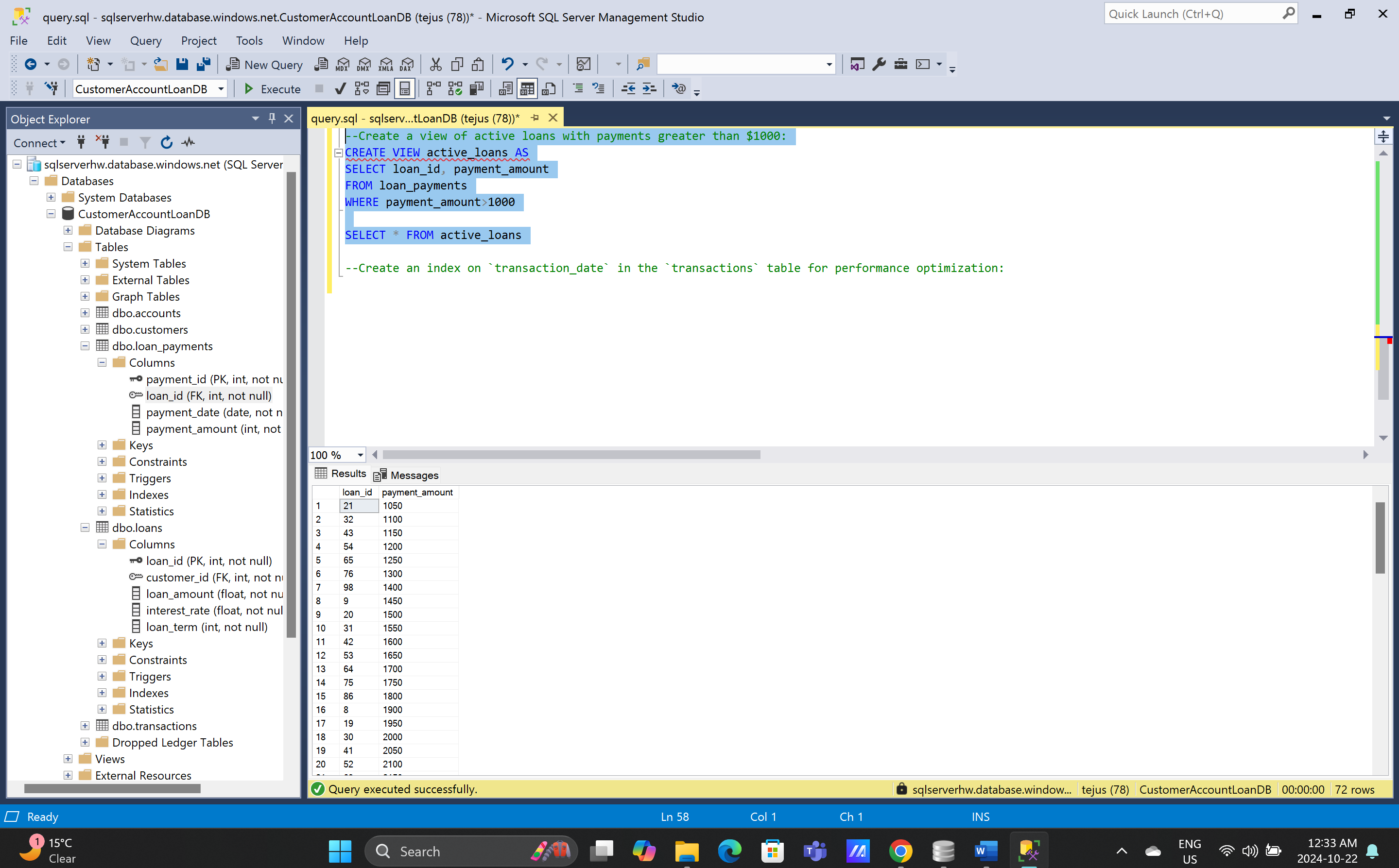
CREATE VIEW active\_loans AS

SELECT loan\_id, payment\_amount

FROM loan\_payments

WHERE payment\_amount>1000

SELECT \* FROM active\_loans



1. --Create an index on `transaction\_date` in the `transactions` table for performance optimization:

CREATE INDEX transaction\_date

ON transactions (transaction\_date)