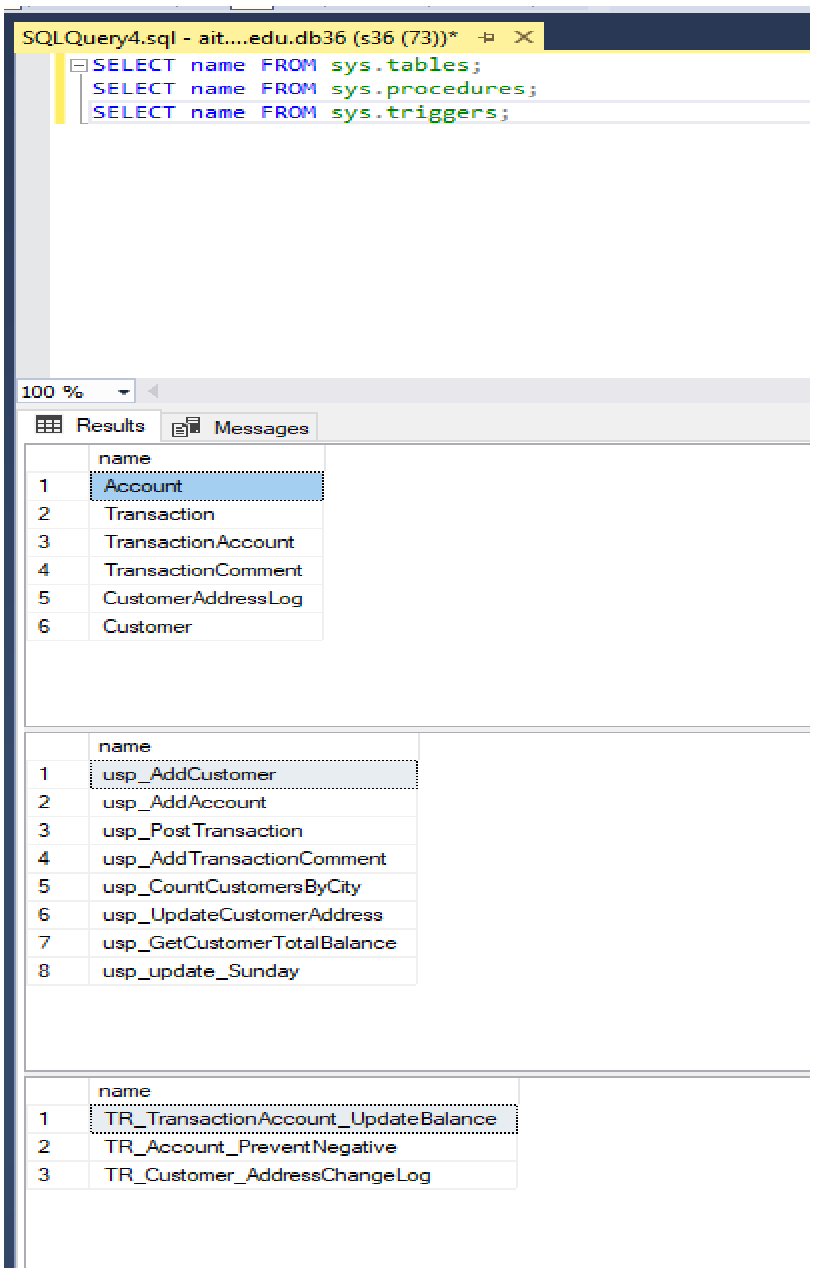

SQLQuery4.sql - ait....edu.db36 (s36 (73))*

```sql
SELECT name FROM sys.tables;
SELECT name FROM sys.procedures;
SELECT name FROM sys.triggers;
```

100 %

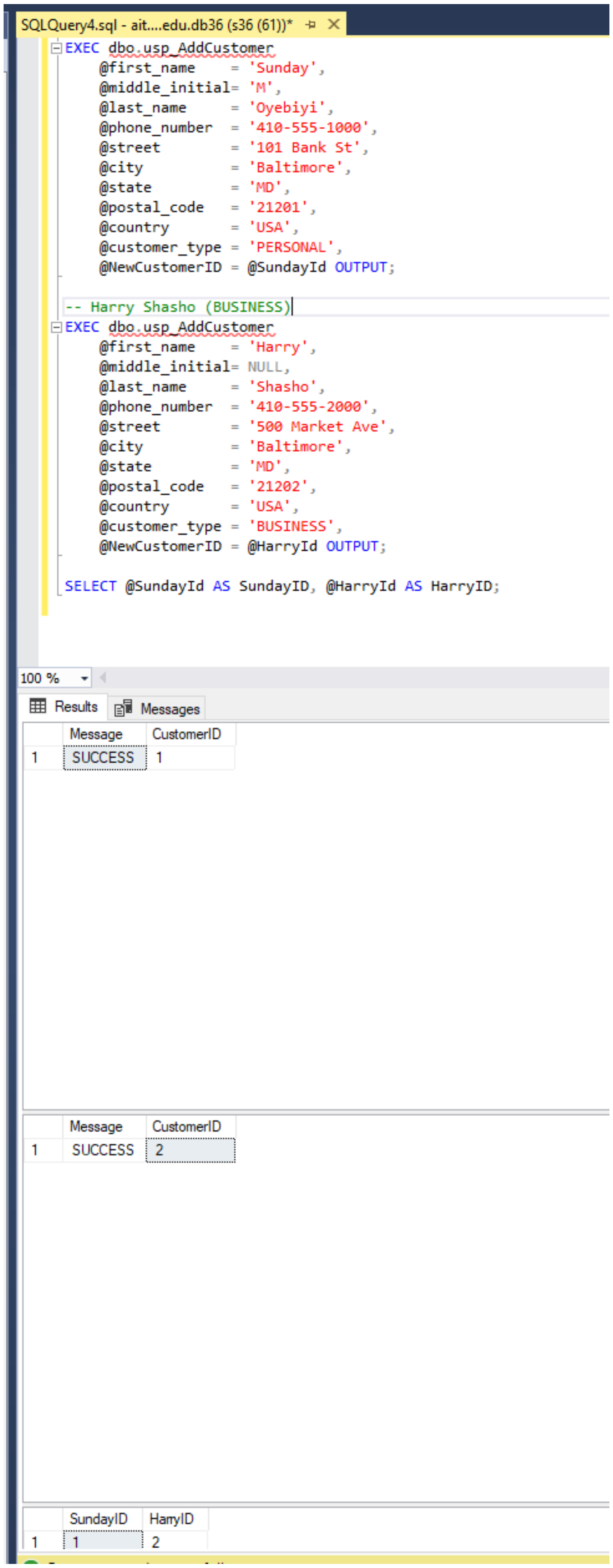
Results | Messages

	name
1	Account
2	Transaction
3	TransactionAccount
4	TransactionComment
5	CustomerAddressLog
6	Customer

	name
1	usp_AddCustomer
2	usp_AddAccount
3	usp_PostTransaction
4	usp_AddTransactionComment
5	usp_CountCustomersByCity
6	usp_UpdateCustomerAddress
7	usp_GetCustomerTotalBalance
8	usp_update_Sunday

	name
1	TR_TransactionAccount_UpdateBalance
2	TR_Account_PreventNegative
3	TR_Customer_AddressChangeLog

1A – Working examples



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
EXEC dbo.usp_AddCustomer
    @first_name     = 'Sunday',
    @middle_initial= 'M',
    @last_name      = 'Oyebiyi',
    @phone_number   = '410-555-1000',
    @street         = '101 Bank St',
    @city           = 'Baltimore',
    @state          = 'MD',
    @postal_code    = '21201',
    @country        = 'USA',
    @customer_type = 'PERSONAL',
    @NewCustomerID = @SundayId OUTPUT;

-- Harry Shasho (BUSINESS)
EXEC dbo.usp_AddCustomer
    @first_name     = 'Harry',
    @middle_initial= NULL,
    @last_name      = 'Shasho',
    @phone_number   = '410-555-2000',
    @street         = '500 Market Ave',
    @city           = 'Baltimore',
    @state          = 'MD',
    @postal_code    = '21202',
    @country        = 'USA',
    @customer_type = 'BUSINESS',
    @NewCustomerID = @HarryId OUTPUT;

SELECT @SundayId AS SundayID, @HarryId AS HarryID;
```

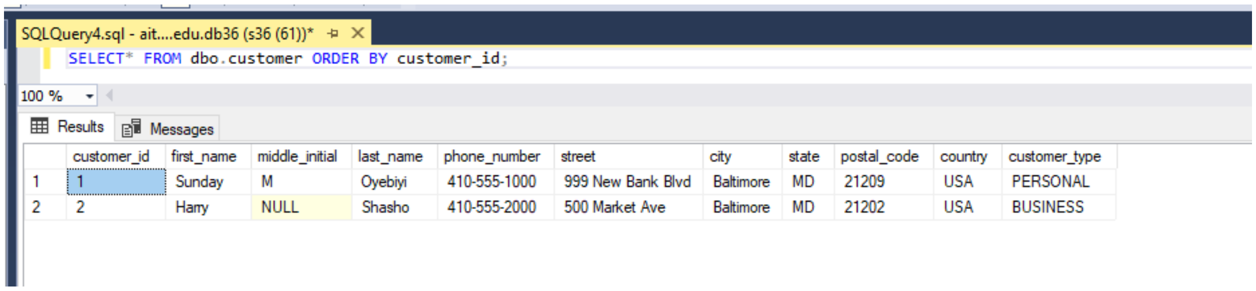
100 %

Results | Messages

	Message	CustomerID
1	SUCCESS	1

	Message	CustomerID
1	SUCCESS	2

	SundayID	HarryID
1	1	2


SQLQuery4.sql - ait....edu.db36 (s36 (61))*

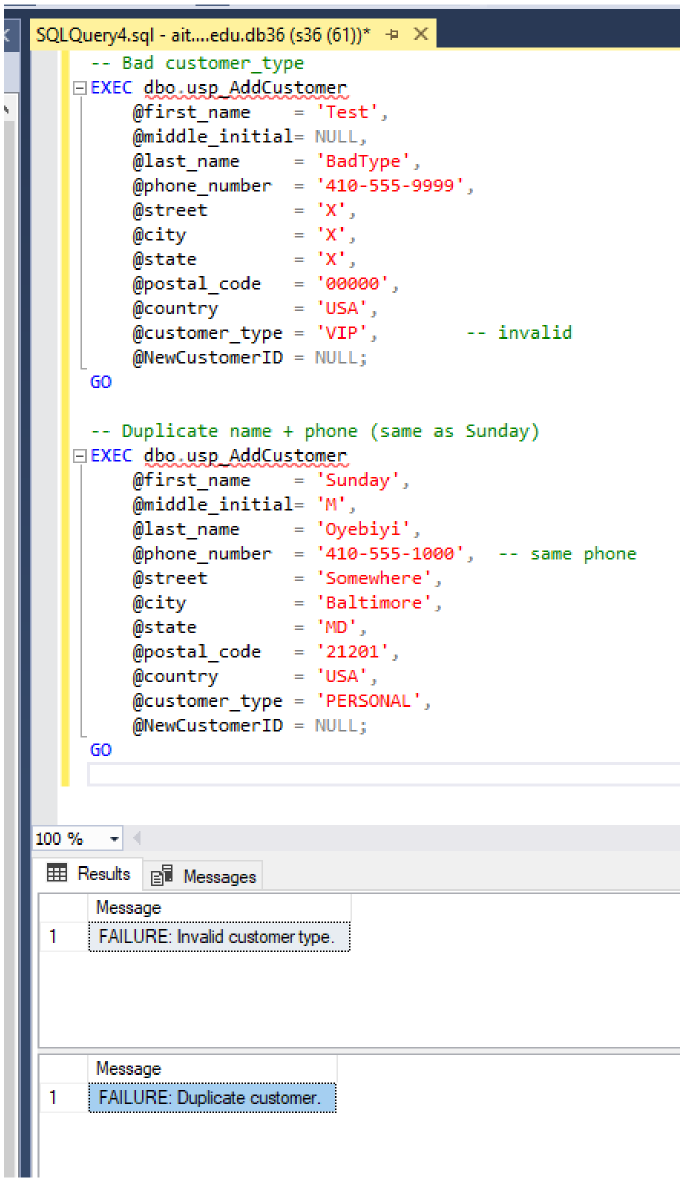
```
SELECT* FROM dbo.customer ORDER BY customer_id;
```

100 %

Results | Messages

	customer_id	first_name	middle_initial	last_name	phone_number	street	city	state	postal_code	country	customer_type
1	1	Sunday	M	Oyebiyi	410-555-1000	999 New Bank Blvd	Baltimore	MD	21209	USA	PERSONAL
2	2	Harry	NULL	Shasho	410-555-2000	500 Market Ave	Baltimore	MD	21202	USA	BUSINESS

1B – Non-working examples (should FAIL)


SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```
-- Bad customer_type
EXEC dbo.usp_AddCustomer
    @first_name     = 'Test',
    @middle_initial= NULL,
    @last_name      = 'BadType',
    @phone_number   = '410-555-9999',
    @street         = 'X',
    @city           = 'X',
    @state          = 'X',
    @postal_code    = '00000',
    @country        = 'USA',
    @customer_type = 'VIP',          -- invalid
    @NewCustomerID = NULL;
GO

-- Duplicate name + phone (same as Sunday)
EXEC dbo.usp_AddCustomer
    @first_name     = 'Sunday',
    @middle_initial= 'M',
    @last_name      = 'Oyebiyi',
    @phone_number   = '410-555-1000',   -- same phone
    @street         = 'Somewhere',
    @city           = 'Baltimore',
    @state          = 'MD',
    @postal_code    = '21201',
    @country        = 'USA',
    @customer_type = 'PERSONAL',
    @NewCustomerID = NULL;
GO
```

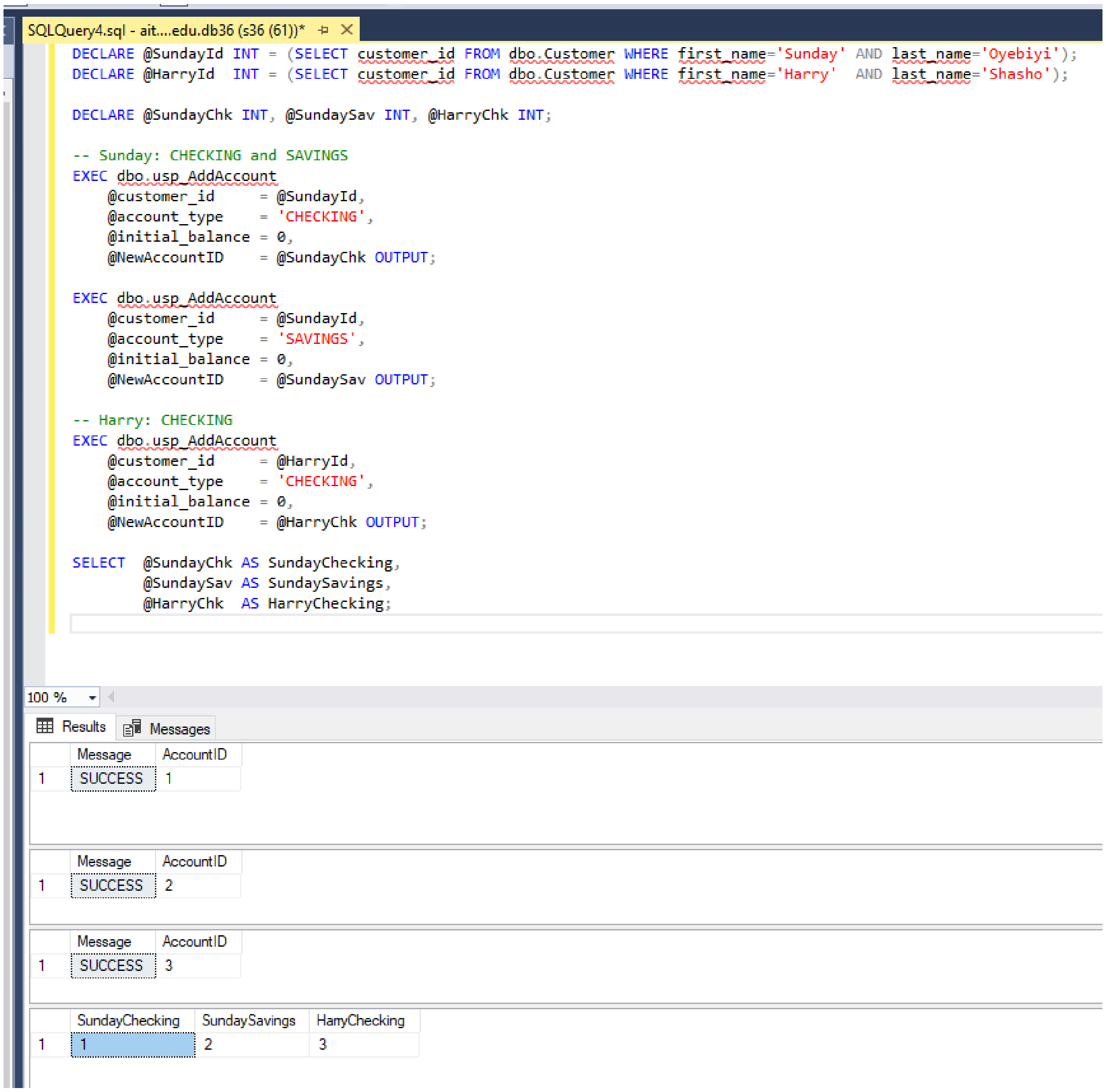
100 %

Results | Messages

	Message
1	FAILURE: Invalid customer type.

	Message
1	FAILURE: Duplicate customer.

REQ 2 – Add accounts for Sunday and Harry



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
DECLARE @SundayId INT = (SELECT customer_id FROM dbo.Customer WHERE first_name='Sunday' AND last_name='Oyebiyi');
DECLARE @HarryId  INT = (SELECT customer_id FROM dbo.Customer WHERE first_name='Harry'  AND last_name='Shasho');

DECLARE @SundayChk INT, @SundaySav INT, @HarryChk INT;

-- Sunday: CHECKING and SAVINGS
EXEC dbo.usp_AddAccount
    @customer_id     = @SundayId,
    @account_type    = 'CHECKING',
    @initial_balance = 0,
    @NewAccountID    = @SundayChk OUTPUT;

EXEC dbo.usp_AddAccount
    @customer_id     = @SundayId,
    @account_type    = 'SAVINGS',
    @initial_balance = 0,
    @NewAccountID    = @SundaySav OUTPUT;

-- Harry: CHECKING
EXEC dbo.usp_AddAccount
    @customer_id     = @HarryId,
    @account_type    = 'CHECKING',
    @initial_balance = 0,
    @NewAccountID    = @HarryChk OUTPUT;

SELECT  @SundayChk AS SundayChecking,
        @SundaySav AS SundaySavings,
        @HarryChk  AS HarryChecking;
```

100 %

Results | Messages

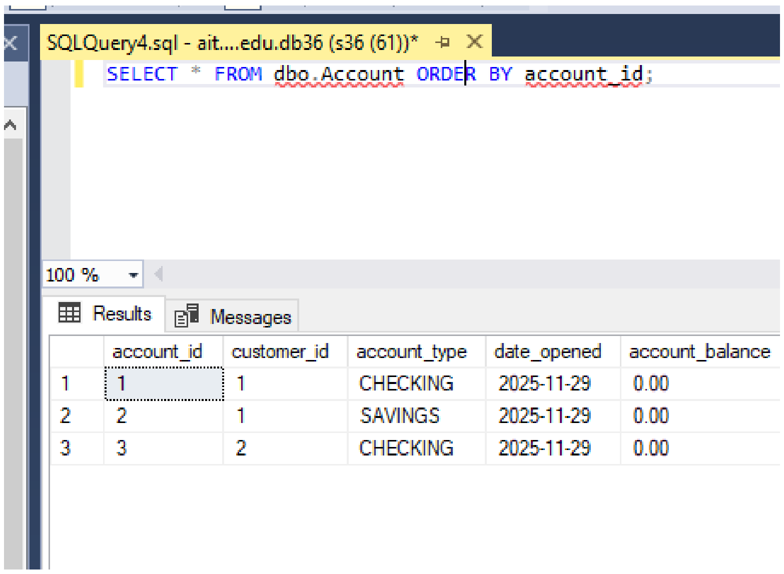
	Message	AccountID
1	SUCCESS	1

	Message	AccountID
1	SUCCESS	2

	Message	AccountID
1	SUCCESS	3

	SundayChecking	SundaySavings	HarryChecking
1	1	2	3

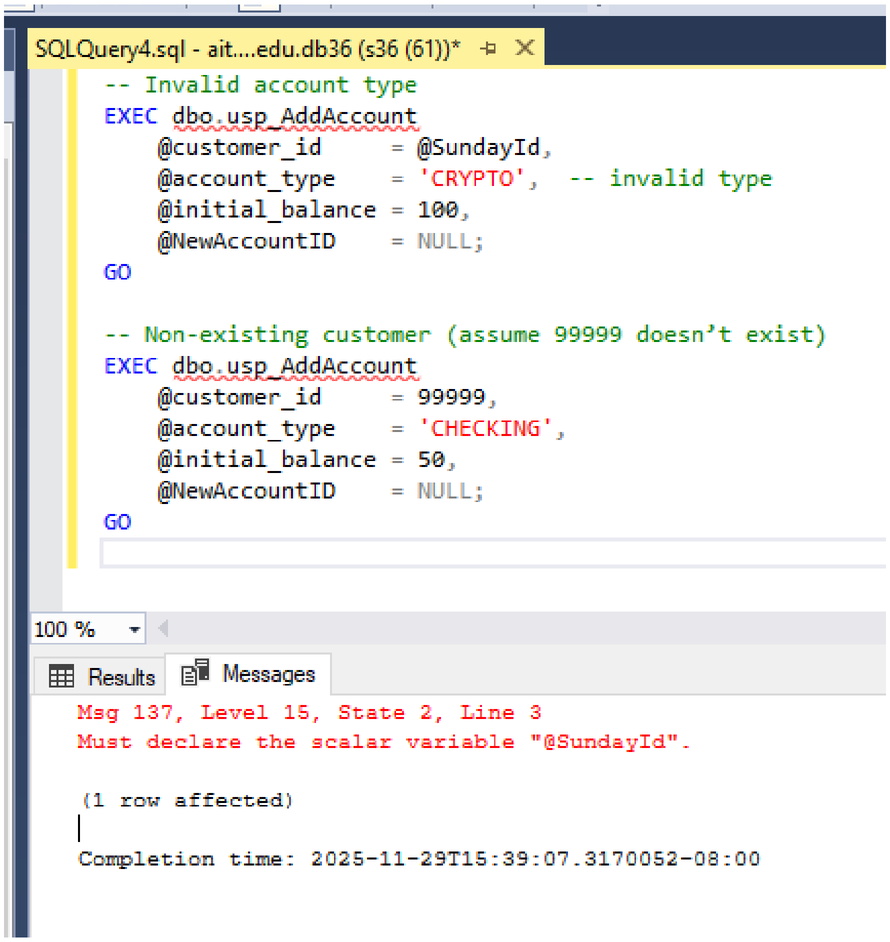
Check accounts:



```
SELECT * FROM dbo.Account ORDER BY account_id;
```

	account_id	customer_id	account_type	date_opened	account_balance
1	1	1	CHECKING	2025-11-29	0.00
2	2	1	SAVINGS	2025-11-29	0.00
3	3	2	CHECKING	2025-11-29	0.00

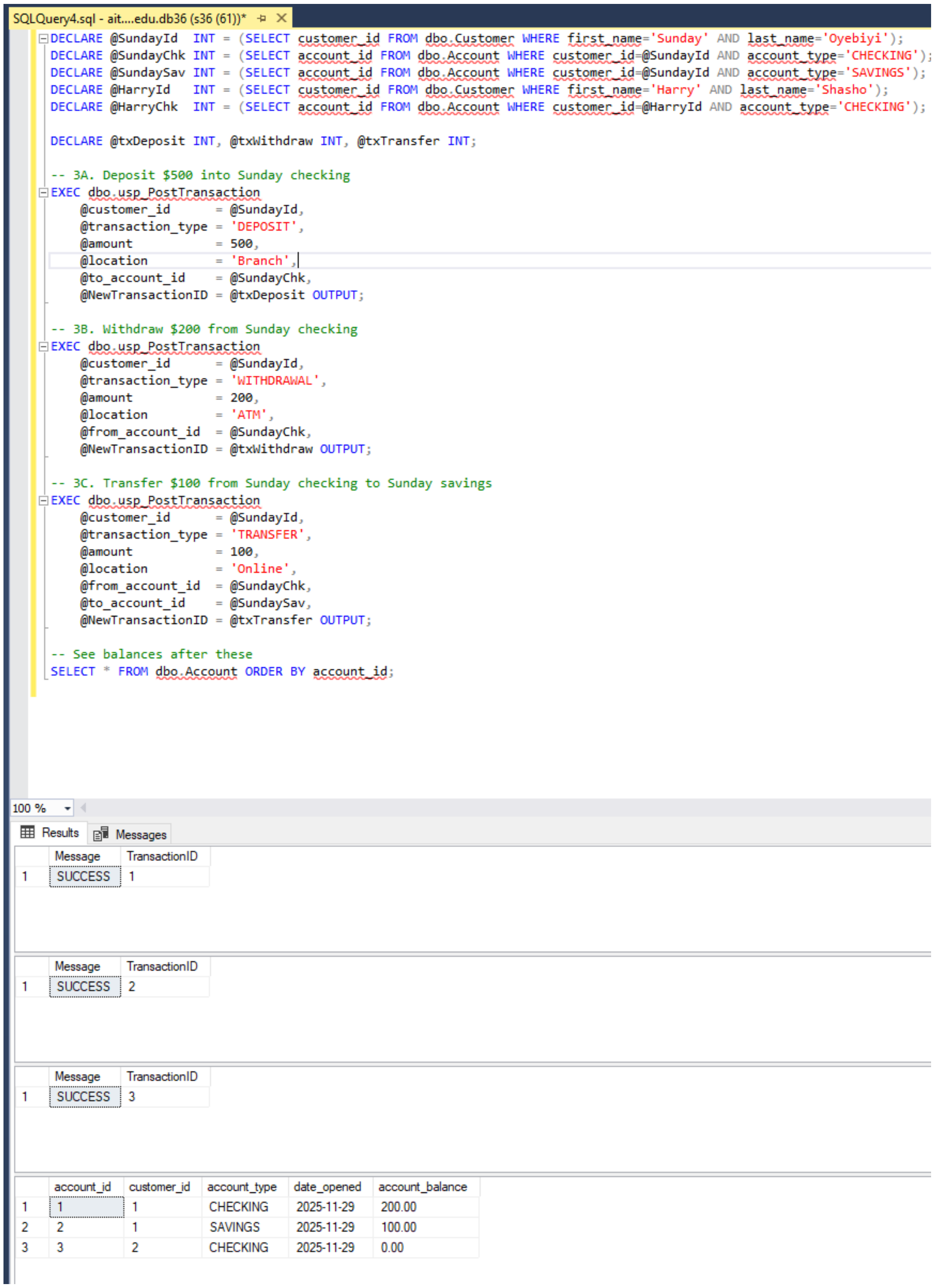
2B – Non-working examples (should fail)



```
-- Invalid account type
EXEC dbo.usp_AddAccount
    @customer_id     = @SundayId,
    @account_type    = 'CRYPTO',  -- invalid type
    @initial_balance = 100,
    @NewAccountID    = NULL;
GO

-- Non-existing customer (assume 99999 doesn’t exist)
EXEC dbo.usp_AddAccount
    @customer_id     = 99999,
    @account_type    = 'CHECKING',
    @initial_balance = 50,
    @NewAccountID    = NULL;
GO
```

```
Msg 137, Level 15, State 2, Line 3
Must declare the scalar variable "@SundayId".

(1 row affected)

Completion time: 2025-11-29T15:39:07.3170052-08:00
```

REQ 3 – Transactions (deposit, withdrawal, transfer) + triggers


SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
DECLARE @SundayId  INT = (SELECT customer_id FROM dbo.Customer WHERE first_name='Sunday' AND last_name='Oyebiyi');
DECLARE @SundayChk INT = (SELECT account_id FROM dbo.Account WHERE customer_id=@SundayId AND account_type='CHECKING');
DECLARE @SundaySav INT = (SELECT account_id FROM dbo.Account WHERE customer_id=@SundayId AND account_type='SAVINGS');
DECLARE @HarryId   INT = (SELECT customer_id FROM dbo.Customer WHERE first_name='Harry' AND last_name='Shasho');
DECLARE @HarryChk  INT = (SELECT account_id FROM dbo.Account WHERE customer_id=@HarryId AND account_type='CHECKING');

DECLARE @txDeposit INT, @txWithdraw INT, @txTransfer INT;

-- 3A. Deposit $500 into Sunday checking
EXEC dbo.usp_PostTransaction
    @customer_id      = @SundayId,
    @transaction_type = 'DEPOSIT',
    @amount           = 500,
    @location         = 'Branch',
    @to_account_id    = @SundayChk,
    @NewTransactionID = @txDeposit OUTPUT;

-- 3B. Withdraw $200 from Sunday checking
EXEC dbo.usp_PostTransaction
    @customer_id      = @SundayId,
    @transaction_type = 'WITHDRAWAL',
    @amount           = 200,
    @location         = 'ATM',
    @from_account_id  = @SundayChk,
    @NewTransactionID = @txWithdraw OUTPUT;

-- 3C. Transfer $100 from Sunday checking to Sunday savings
EXEC dbo.usp_PostTransaction
    @customer_id      = @SundayId,
    @transaction_type = 'TRANSFER',
    @amount           = 100,
    @location         = 'Online',
    @from_account_id  = @SundayChk,
    @to_account_id    = @SundaySav,
    @NewTransactionID = @txTransfer OUTPUT;

-- See balances after these
SELECT * FROM dbo.Account ORDER BY account_id;
```

	Message	TransactionID
1	SUCCESS	1

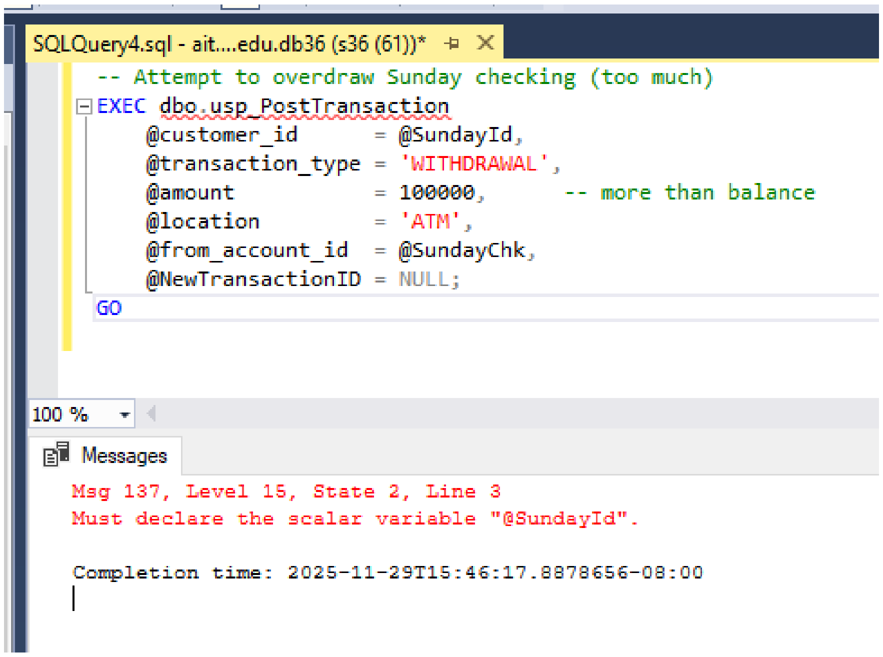
	Message	TransactionID
1	SUCCESS	2

	Message	TransactionID
1	SUCCESS	3

	account_id	customer_id	account_type	date_opened	account_balance
1	1	1	CHECKING	2025-11-29	200.00
2	2	1	SAVINGS	2025-11-29	100.00
3	3	2	CHECKING	2025-11-29	0.00

3D – Non-working transaction examples

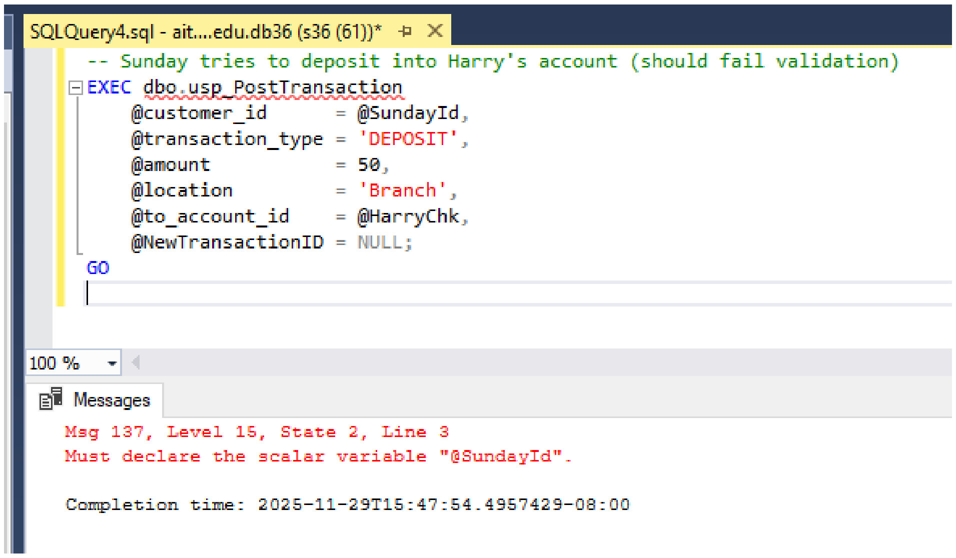
Try to overdraft → trigger should prevent negative balance and proc returns failure:



```
-- Attempt to overdraw Sunday checking (too much)
EXEC dbo.usp_PostTransaction
    @customer_id       = @SundayId,
    @transaction_type = 'WITHDRAWAL',
    @amount            = 100000,      -- more than balance
    @location          = 'ATM',
    @from_account_id  = @SundayChk,
    @NewTransactionID = NULL;
GO
```

```
Msg 137, Level 15, State 2, Line 3
Must declare the scalar variable "@SundayId".

Completion time: 2025-11-29T15:46:17.8878656-08:00
```

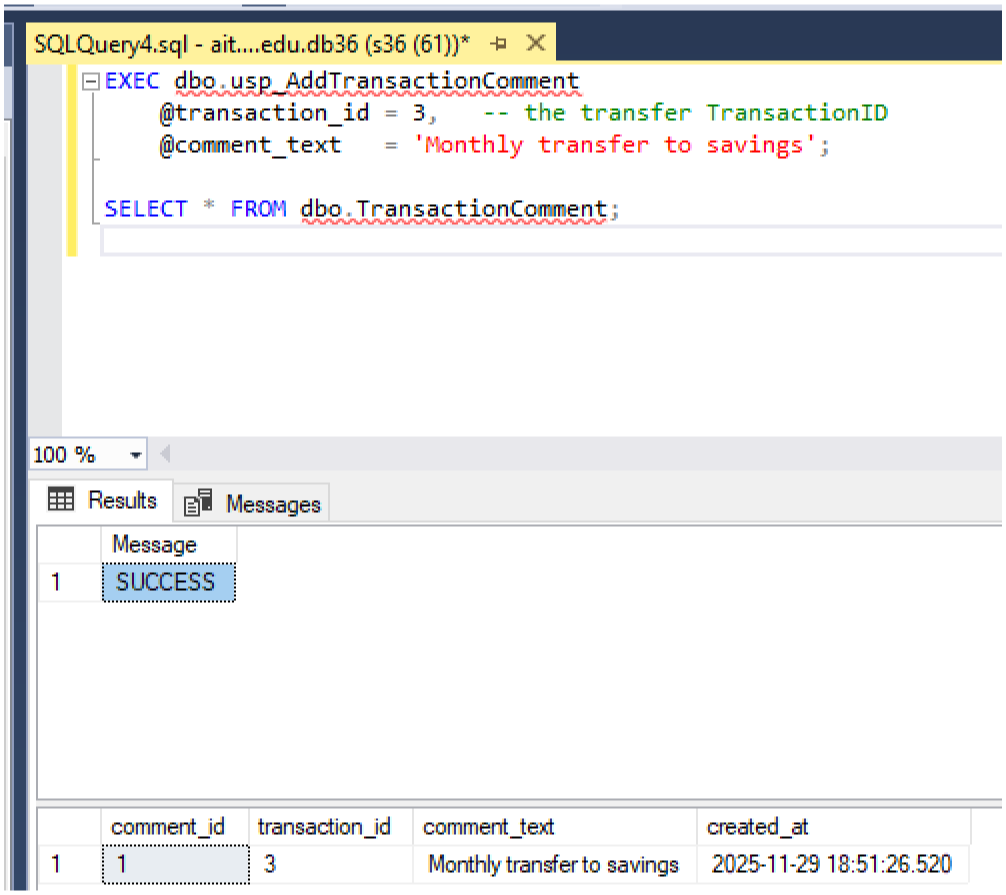
Try to deposit into an account Sunday does NOT own (Harry's):



```
-- Sunday tries to deposit into Harry's account (should fail validation)
EXEC dbo.usp_PostTransaction
    @customer_id       = @SundayId,
    @transaction_type = 'DEPOSIT',
    @amount            = 50,
    @location          = 'Branch',
    @to_account_id    = @HarryChk,
    @NewTransactionID = NULL;
GO
```

```
Msg 137, Level 15, State 2, Line 3
Must declare the scalar variable "@SundayId".

Completion time: 2025-11-29T15:47:54.4957429-08:00
```

REQ 3 – Comments



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```
EXEC dbo.usp_AddTransactionComment
    @transaction_id = 3,    -- the transfer TransactionID
    @comment_text   = 'Monthly transfer to savings';

SELECT * FROM dbo.TransactionComment;
```

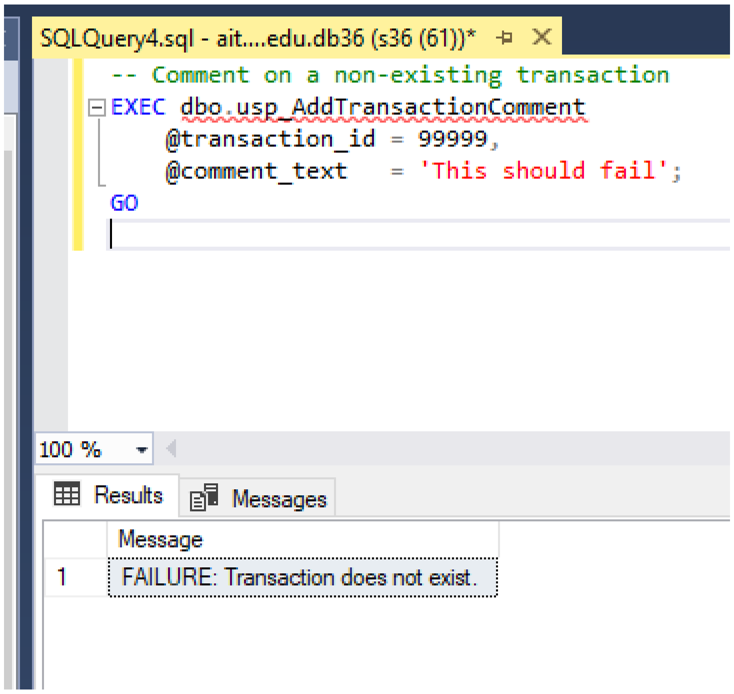
100 %

Results | Messages

	Message
1	SUCCESS

	comment_id	transaction_id	comment_text	created_at
1	1	3	Monthly transfer to savings	2025-11-29 18:51:26.520

Non-working example:



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

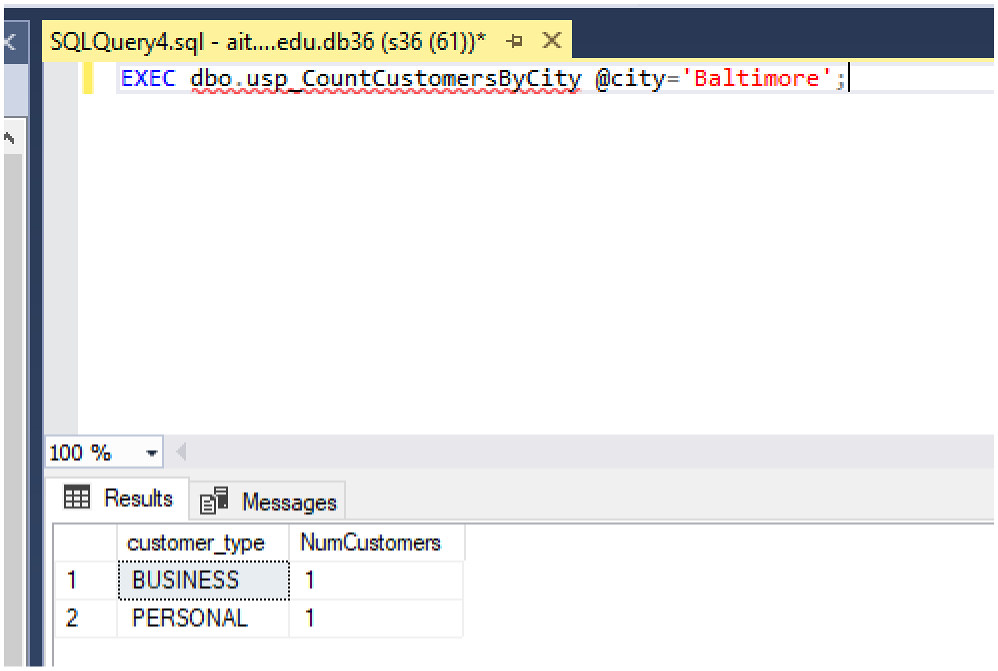
```
-- Comment on a non-existing transaction
EXEC dbo.usp_AddTransactionComment
    @transaction_id = 99999,
    @comment_text   = 'This should fail';
GO
```

100 %

Results | Messages

	Message
1	FAILURE: Transaction does not exist.

REQ 4 – Customers by city



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

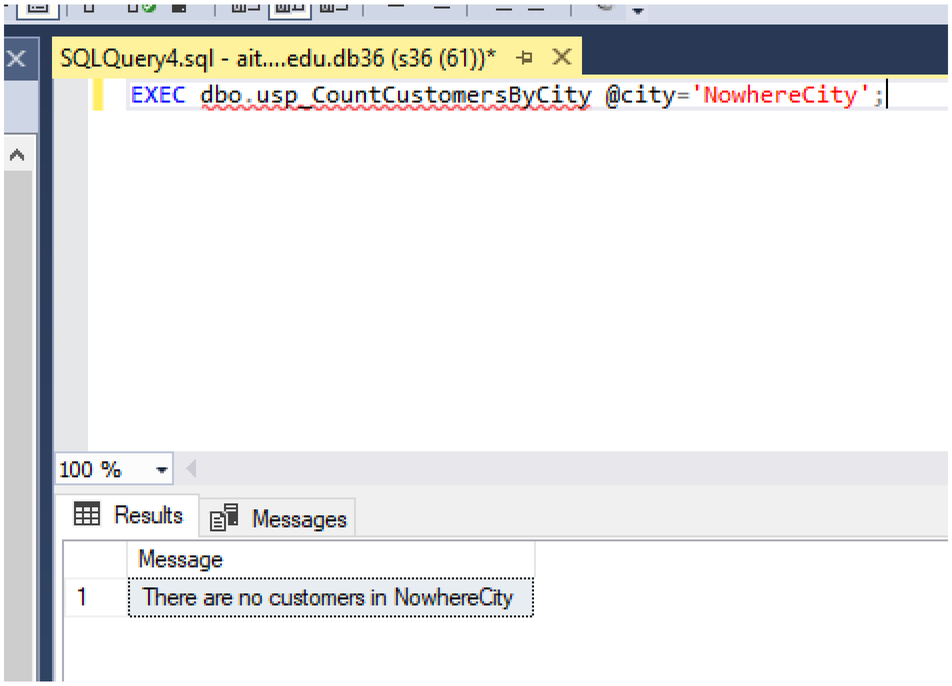
```
EXEC dbo.usp_CountCustomersByCity @city='Baltimore';
```

100 %

Results | Messages

	customer_type	NumCustomers
1	BUSINESS	1
2	PERSONAL	1

Non-working / empty cas



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

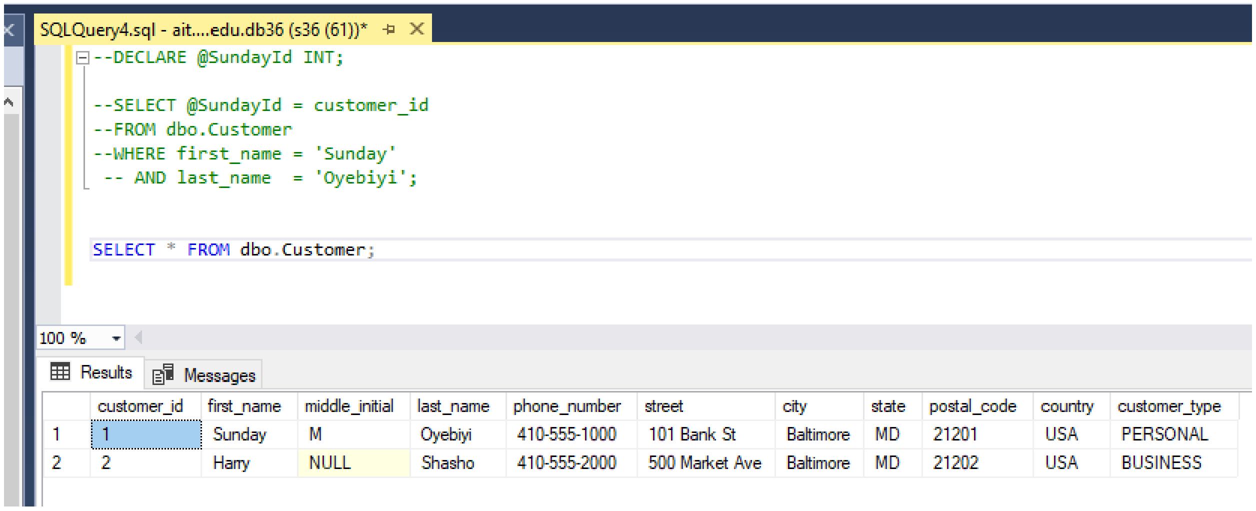
```
EXEC dbo.usp_CountCustomersByCity @city='NowhereCity';
```

100 %

Results | Messages

	Message
1	There are no customers in NowhereCity

REQ 5 – Update Sunday’s address (and log trigger)



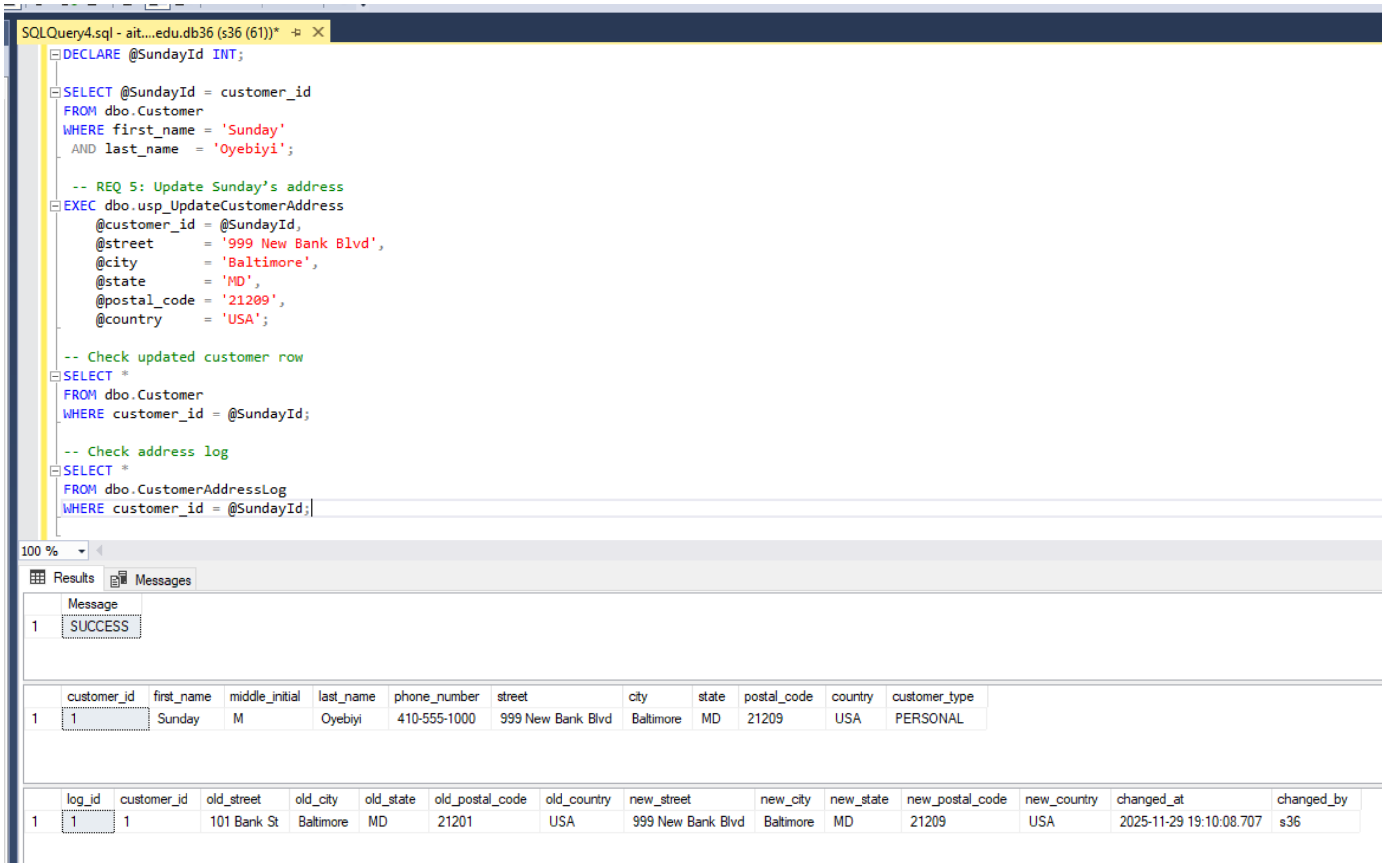
SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
--DECLARE @SundayId INT;

--SELECT @SundayId = customer_id
--FROM dbo.Customer
--WHERE first_name = 'Sunday'
 -- AND last_name  = 'Oyebiyi';

SELECT * FROM dbo.Customer;
```

100 %

Results | Messages

	customer_id	first_name	middle_initial	last_name	phone_number	street	city	state	postal_code	country	customer_type
1	1	Sunday	M	Oyebiyi	410-555-1000	101 Bank St	Baltimore	MD	21201	USA	PERSONAL
2	2	Harry	NULL	Shasho	410-555-2000	500 Market Ave	Baltimore	MD	21202	USA	BUSINESS



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
DECLARE @SundayId INT;

SELECT @SundayId = customer_id
FROM dbo.Customer
WHERE first_name = 'Sunday'
  AND last_name  = 'Oyebiyi';

 -- REQ 5: Update Sunday’s address
EXEC dbo.usp_UpdateCustomerAddress
    @customer_id = @SundayId,
    @street      = '999 New Bank Blvd',
    @city        = 'Baltimore',
    @state       = 'MD',
    @postal_code = '21209',
    @country     = 'USA';

-- Check updated customer row
SELECT *
FROM dbo.Customer
WHERE customer_id = @SundayId;

-- Check address log
SELECT *
FROM dbo.CustomerAddressLog
WHERE customer_id = @SundayId;
```

100 %

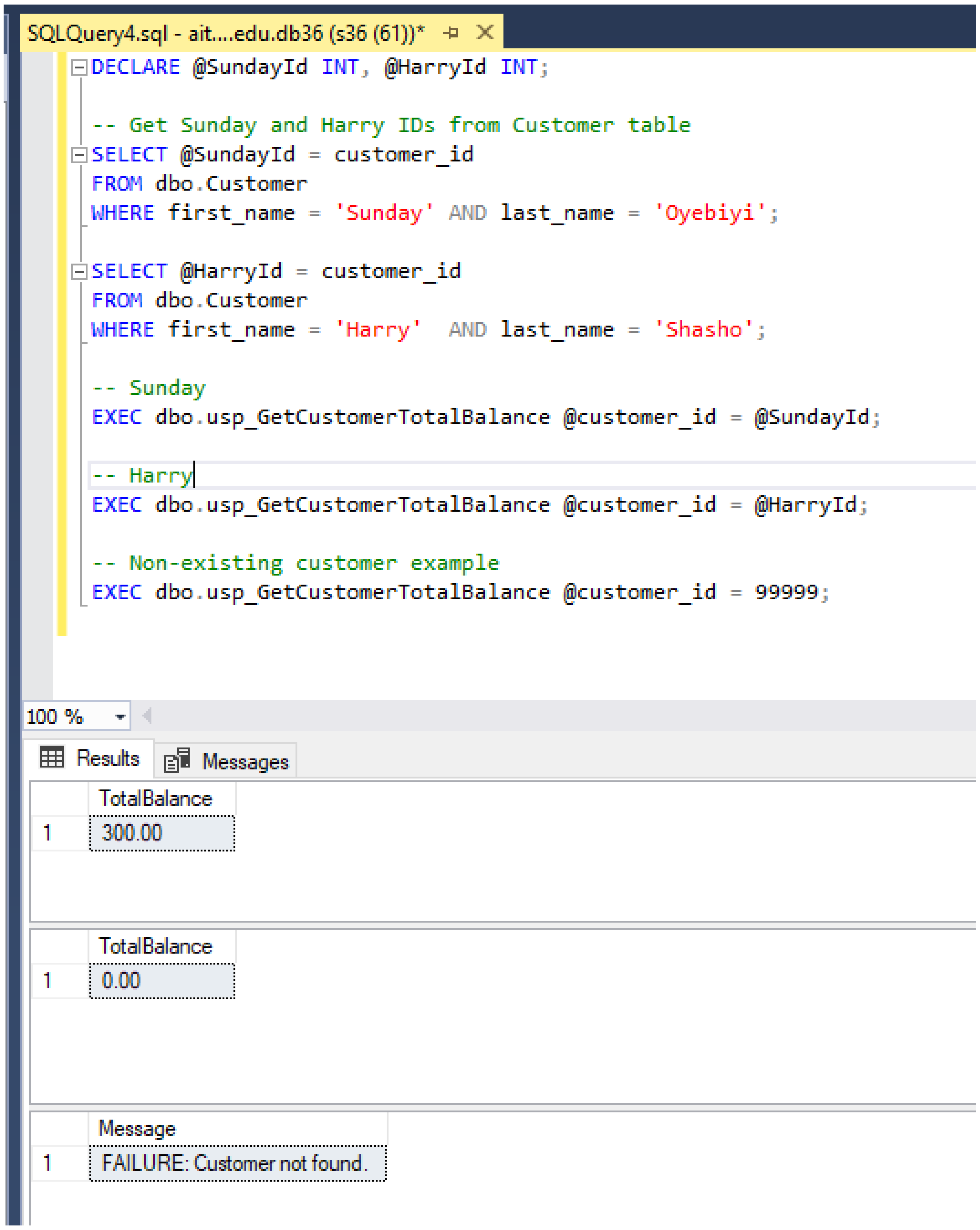
Results | Messages

	Message
1	SUCCESS

	customer_id	first_name	middle_initial	last_name	phone_number	street	city	state	postal_code	country	customer_type
1	1	Sunday	M	Oyebiyi	410-555-1000	999 New Bank Blvd	Baltimore	MD	21209	USA	PERSONAL

	log_id	customer_id	old_street	old_city	old_state	old_postal_code	old_country	new_street	new_city	new_state	new_postal_code	new_country	changed_at	changed_by
1	1	1	101 Bank St	Baltimore	MD	21201	USA	999 New Bank Blvd	Baltimore	MD	21209	USA	2025-11-29 19:10:08.707	s36

REQ 6 – Total balances



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
DECLARE @SundayId INT, @HarryId INT;

-- Get Sunday and Harry IDs from Customer table
SELECT @SundayId = customer_id
FROM dbo.Customer
WHERE first_name = 'Sunday' AND last_name = 'Oyebiyi';

SELECT @HarryId = customer_id
FROM dbo.Customer
WHERE first_name = 'Harry'  AND last_name = 'Shasho';

-- Sunday
EXEC dbo.usp_GetCustomerTotalBalance @customer_id = @SundayId;

-- Harry
EXEC dbo.usp_GetCustomerTotalBalance @customer_id = @HarryId;

-- Non-existing customer example
EXEC dbo.usp_GetCustomerTotalBalance @customer_id = 99999;
```

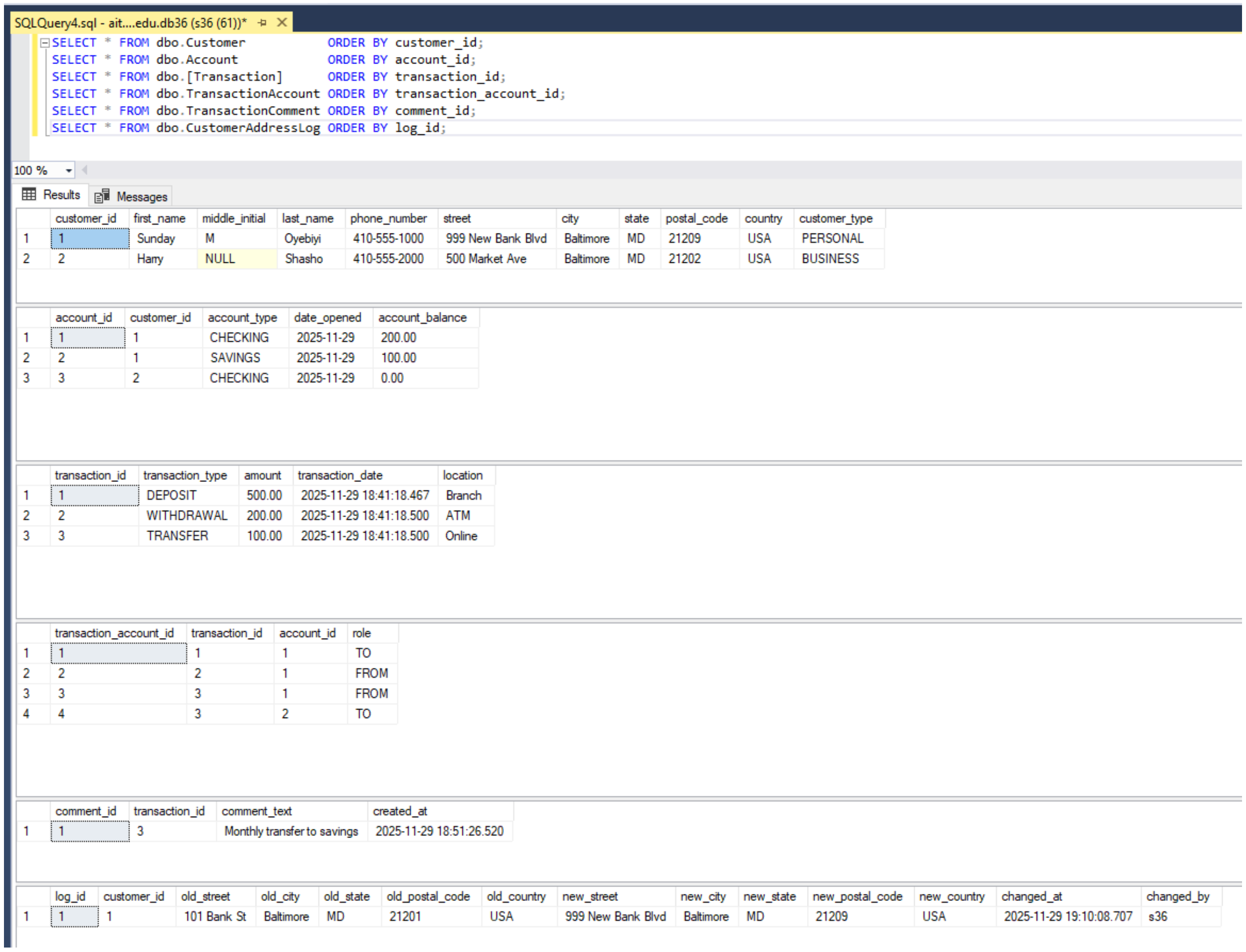
100 %

Results | Messages

	TotalBalance
1	300.00

	TotalBalance
1	0.00

	Message
1	FAILURE: Customer not found.


SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
SELECT * FROM dbo.Customer           ORDER BY customer_id;
SELECT * FROM dbo.Account            ORDER BY account_id;
SELECT * FROM dbo.[Transaction]      ORDER BY transaction_id;
SELECT * FROM dbo.TransactionAccount ORDER BY transaction_account_id;
SELECT * FROM dbo.TransactionComment ORDER BY comment_id;
SELECT * FROM dbo.CustomerAddressLog ORDER BY log_id;
```

100 %

Results | Messages

	customer_id	first_name	middle_initial	last_name	phone_number	street	city	state	postal_code	country	customer_type
1	1	Sunday	M	Oyebiyi	410-555-1000	999 New Bank Blvd	Baltimore	MD	21209	USA	PERSONAL
2	2	Harry	NULL	Shasho	410-555-2000	500 Market Ave	Baltimore	MD	21202	USA	BUSINESS

	account_id	customer_id	account_type	date_opened	account_balance
1	1	1	CHECKING	2025-11-29	200.00
2	2	1	SAVINGS	2025-11-29	100.00
3	3	2	CHECKING	2025-11-29	0.00

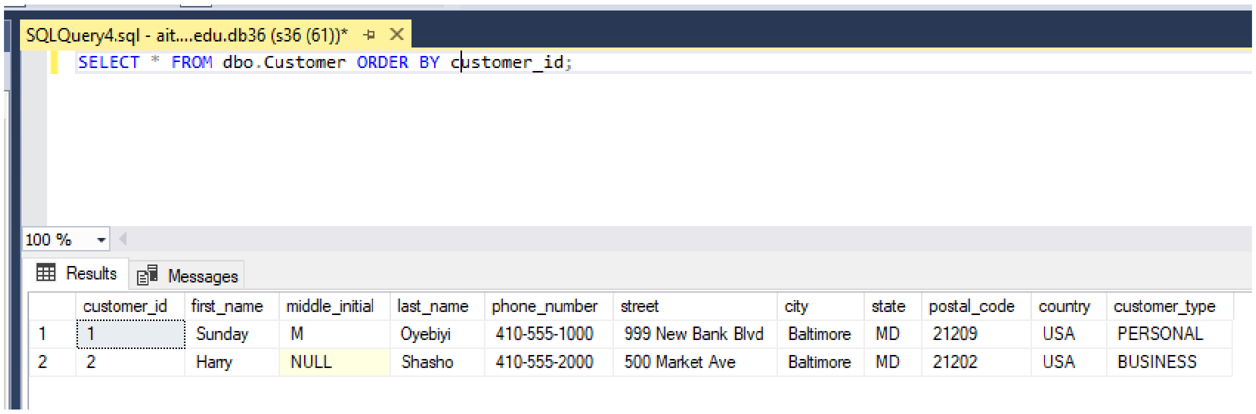
	transaction_id	transaction_type	amount	transaction_date	location
1	1	DEPOSIT	500.00	2025-11-29 18:41:18.467	Branch
2	2	WITHDRAWAL	200.00	2025-11-29 18:41:18.500	ATM
3	3	TRANSFER	100.00	2025-11-29 18:41:18.500	Online

	transaction_account_id	transaction_id	account_id	role
1	1	1	1	TO
2	2	2	1	FROM
3	3	3	1	FROM
4	4	3	2	TO

	comment_id	transaction_id	comment_text	created_at
1	1	3	Monthly transfer to savings	2025-11-29 18:51:26.520

	log_id	customer_id	old_street	old_city	old_state	old_postal_code	old_country	new_street	new_city	new_state	new_postal_code	new_country	changed_at	changed_by
1	1	1	101 Bank St	Baltimore	MD	21201	USA	999 New Bank Blvd	Baltimore	MD	21209	USA	2025-11-29 19:10:08.707	s36

SELECT TABLES

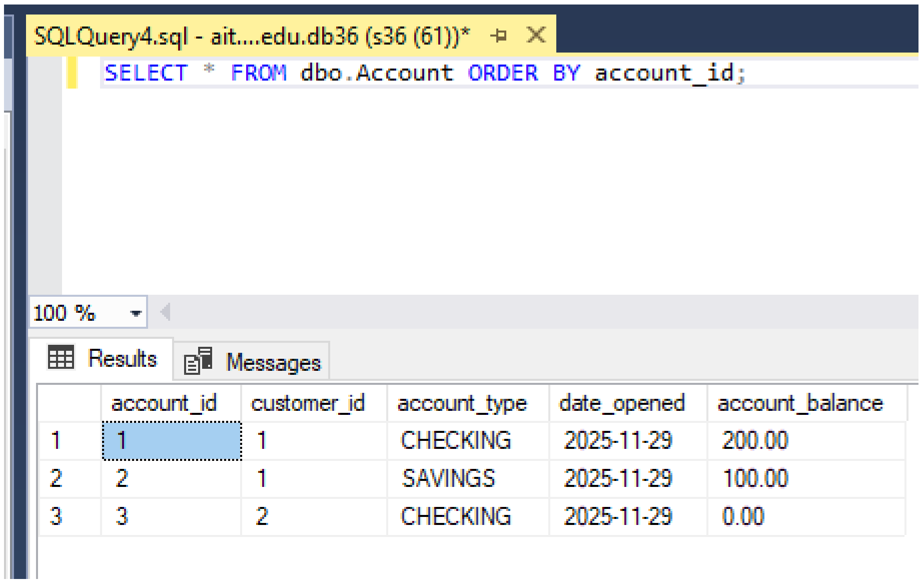

SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```sql
SELECT * FROM dbo.Customer ORDER BY customer_id;
```

100 %

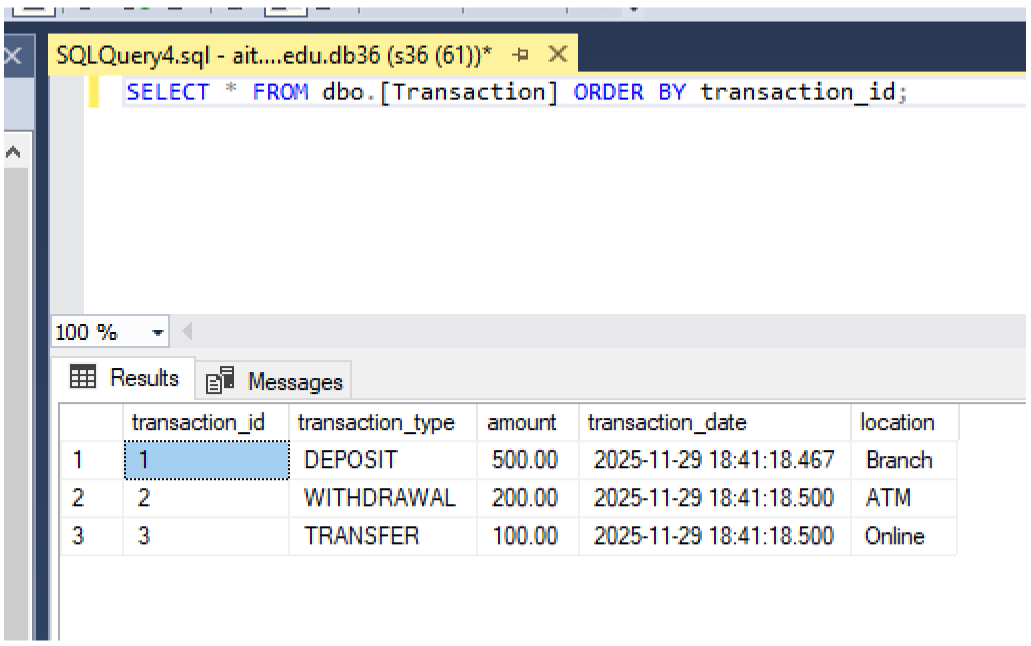
Results | Messages

	customer_id	first_name	middle_initial	last_name	phone_number	street	city	state	postal_code	country	customer_type
1	1	Sunday	M	Oyebiyi	410-555-1000	999 New Bank Blvd	Baltimore	MD	21209	USA	PERSONAL
2	2	Harry	NULL	Shasho	410-555-2000	500 Market Ave	Baltimore	MD	21202	USA	BUSINESS



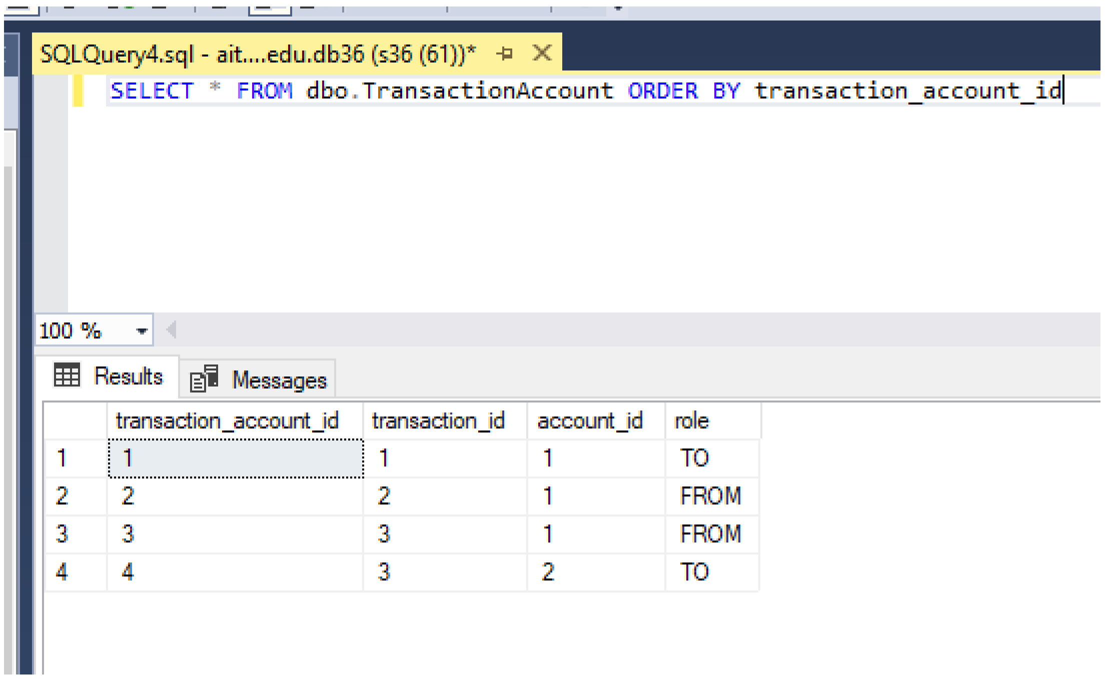
```
SELECT * FROM dbo.Account ORDER BY account_id;
```

	account_id	customer_id	account_type	date_opened	account_balance
1	1	1	CHECKING	2025-11-29	200.00
2	2	1	SAVINGS	2025-11-29	100.00
3	3	2	CHECKING	2025-11-29	0.00



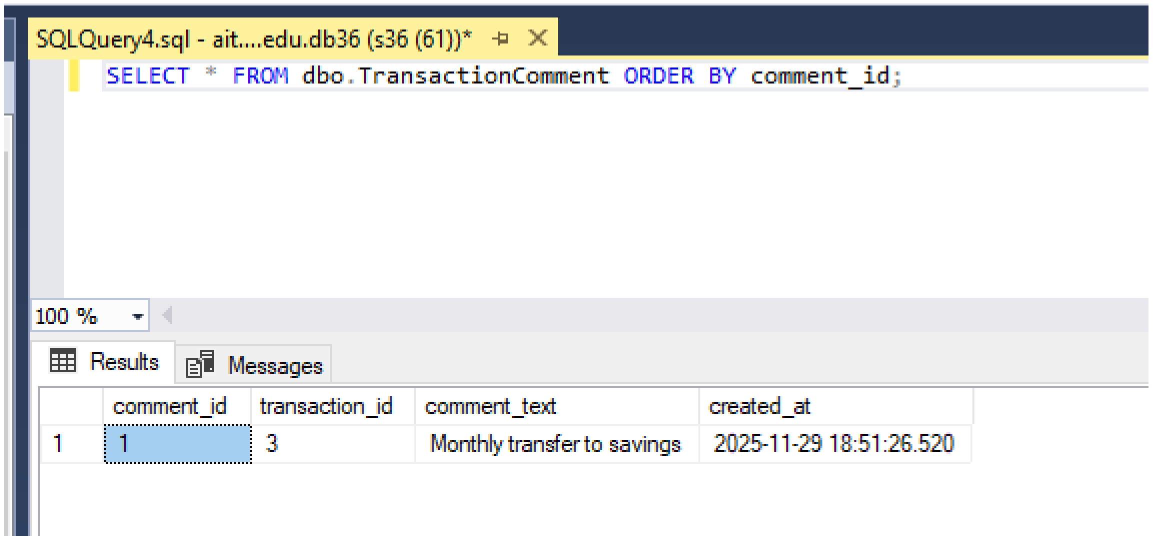
```
SELECT * FROM dbo.[Transaction] ORDER BY transaction_id;
```

	transaction_id	transaction_type	amount	transaction_date	location
1	1	DEPOSIT	500.00	2025-11-29 18:41:18.467	Branch
2	2	WITHDRAWAL	200.00	2025-11-29 18:41:18.500	ATM
3	3	TRANSFER	100.00	2025-11-29 18:41:18.500	Online



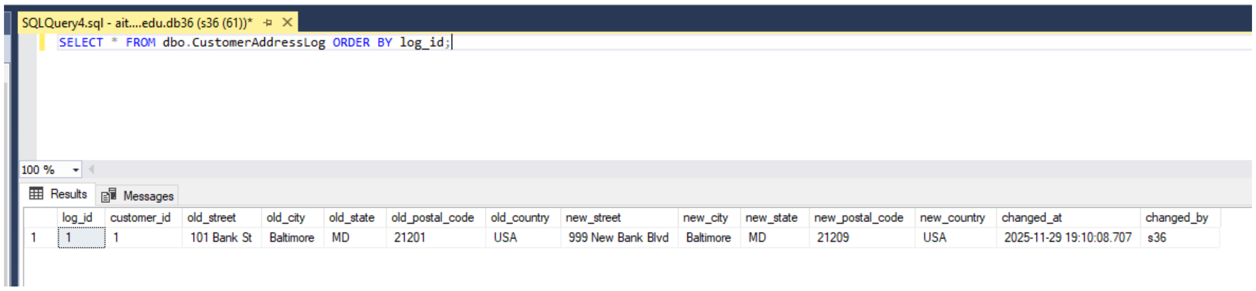
```sql
SELECT * FROM dbo.TransactionAccount ORDER BY transaction_account_id
```

	transaction_account_id	transaction_id	account_id	role
1	1	1	1	TO
2	2	2	1	FROM
3	3	3	1	FROM
4	4	3	2	TO



```sql
SELECT * FROM dbo.TransactionComment ORDER BY comment_id;
```

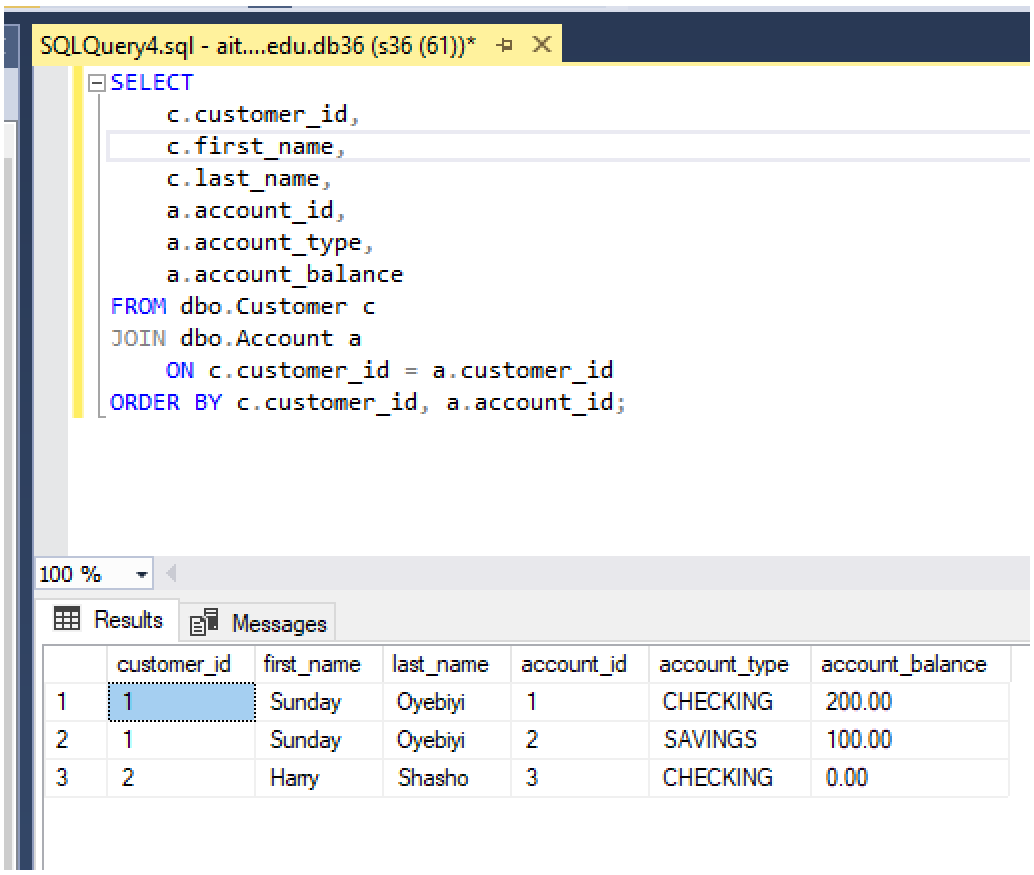
	comment_id	transaction_id	comment_text	created_at
1	1	3	Monthly transfer to savings	2025-11-29 18:51:26.520



```sql
SELECT * FROM dbo.CustomerAddressLog ORDER BY log_id;
```

	log_id	customer_id	old_street	old_city	old_state	old_postal_code	old_country	new_street	new_city	new_state	new_postal_code	new_country	changed_at	changed_by
1	1	1	101 Bank St	Baltimore	MD	21201	USA	999 New Bank Blvd	Baltimore	MD	21209	USA	2025-11-29 19:10:08.707	s36

Accounts WITH customer name (JOIN)



SQLQuery4.sql - ait....edu.db36 (s36 (61))*

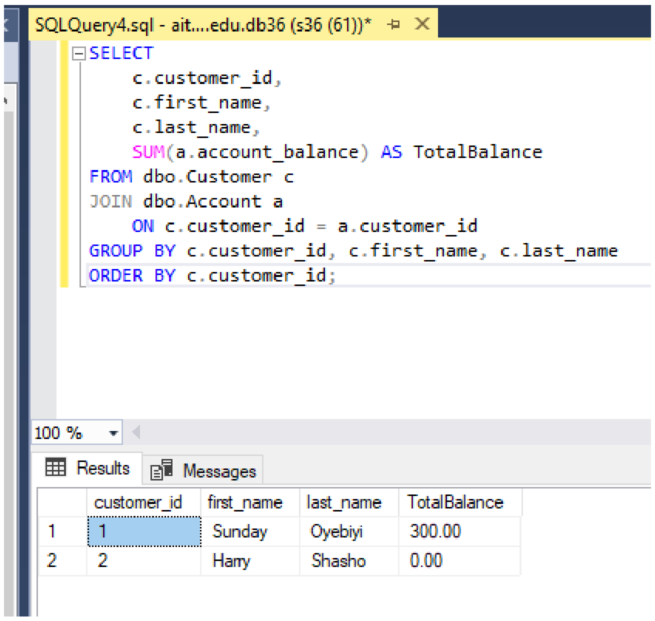
```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    a.account_id,
    a.account_type,
    a.account_balance
FROM dbo.Customer c
JOIN dbo.Account a
    ON c.customer_id = a.customer_id
ORDER BY c.customer_id, a.account_id;
```

100 %

Results | Messages

	customer_id	first_name	last_name	account_id	account_type	account_balance
1	1	Sunday	Oyebiyi	1	CHECKING	200.00
2	1	Sunday	Oyebiyi	2	SAVINGS	100.00
3	2	Harry	Shasho	3	CHECKING	0.00

Extra: Total balance per customer (GROUP BY)



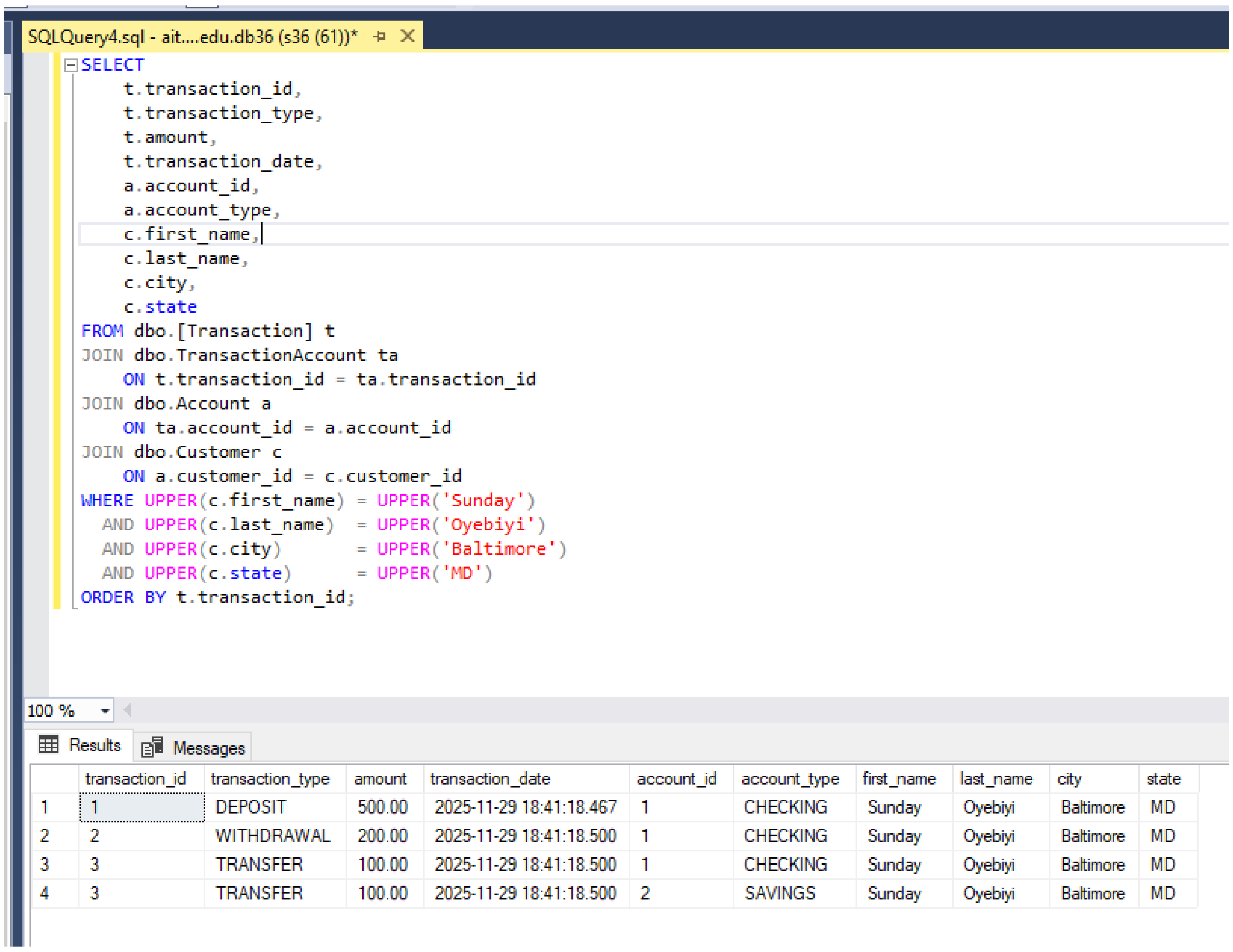
SQLQuery4.sql - ait....edu.db36 (s36 (61))*

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    SUM(a.account_balance) AS TotalBalance
FROM dbo.Customer c
JOIN dbo.Account a
    ON c.customer_id = a.customer_id
GROUP BY c.customer_id, c.first_name, c.last_name
ORDER BY c.customer_id;
```

100 %

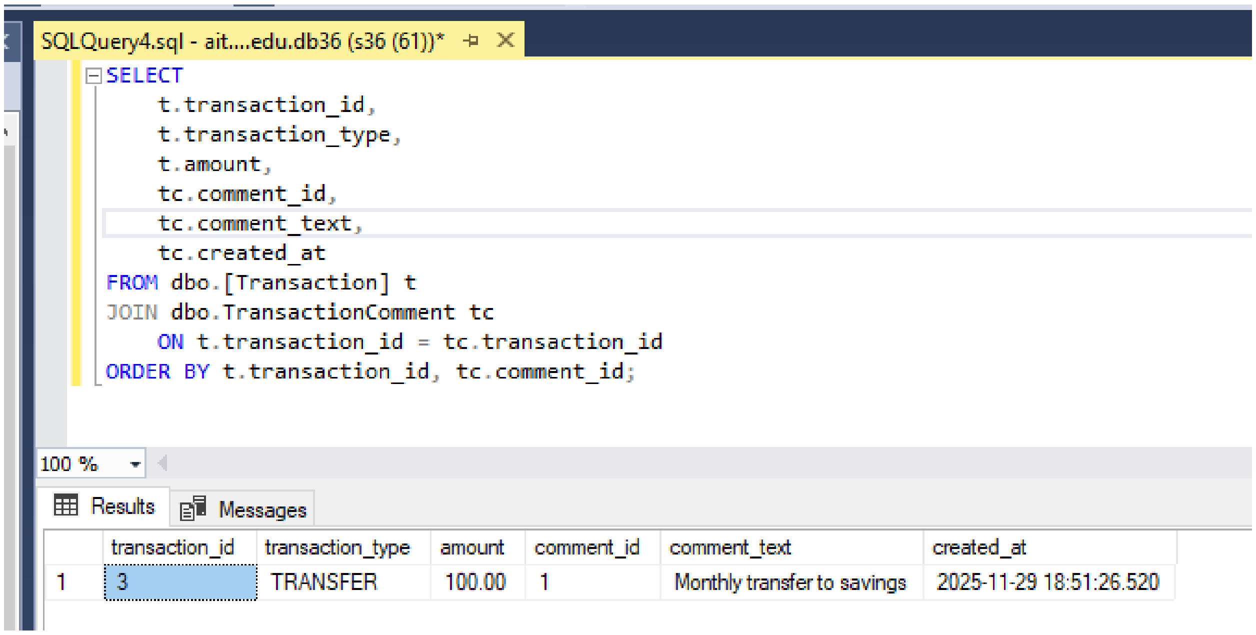
Results | Messages

	customer_id	first_name	last_name	TotalBalance
1	1	Sunday	Oyebiyi	300.00
2	2	Harry	Shasho	0.00



```sql
SELECT
    t.transaction_id,
    t.transaction_type,
    t.amount,
    t.transaction_date,
    a.account_id,
    a.account_type,
    c.first_name,
    c.last_name,
    c.city,
    c.state
FROM dbo.[Transaction] t
JOIN dbo.TransactionAccount ta
    ON t.transaction_id = ta.transaction_id
JOIN dbo.Account a
    ON ta.account_id = a.account_id
JOIN dbo.Customer c
    ON a.customer_id = c.customer_id
WHERE UPPER(c.first_name) = UPPER('Sunday')
  AND UPPER(c.last_name)  = UPPER('Oyebiyi')
  AND UPPER(c.city)       = UPPER('Baltimore')
  AND UPPER(c.state)      = UPPER('MD')
ORDER BY t.transaction_id;
```

	transaction_id	transaction_type	amount	transaction_date	account_id	account_type	first_name	last_name	city	state
1	1	DEPOSIT	500.00	2025-11-29 18:41:18.467	1	CHECKING	Sunday	Oyebiyi	Baltimore	MD
2	2	WITHDRAWAL	200.00	2025-11-29 18:41:18.500	1	CHECKING	Sunday	Oyebiyi	Baltimore	MD
3	3	TRANSFER	100.00	2025-11-29 18:41:18.500	1	CHECKING	Sunday	Oyebiyi	Baltimore	MD
4	3	TRANSFER	100.00	2025-11-29 18:41:18.500	2	SAVINGS	Sunday	Oyebiyi	Baltimore	MD



```sql
SELECT
    t.transaction_id,
    t.transaction_type,
    t.amount,
    tc.comment_id,
    tc.comment_text,
    tc.created_at
FROM dbo.[Transaction] t
JOIN dbo.TransactionComment tc
    ON t.transaction_id = tc.transaction_id
ORDER BY t.transaction_id, tc.comment_id;
```

	transaction_id	transaction_type	amount	comment_id	comment_text	created_at
1	3	TRANSFER	100.00	1	Monthly transfer to savings	2025-11-29 18:51:26.520