

**TUGAS KELOMPOK PRAKTIKUM MANAJEMEN BASIS
DATA PERTEMUAN-3 TENTANG TRIGGERS**



Asisten Praktikum :

Alya Safina & Izazava Putri Asari

Disusun oleh Kelompok 6 Lab A2 :

Aditya Arif Setiawan (24060121140144)

Alfonso Michael N H (24060123140156)

Anggita Mahardika Syarifa (24060119130089)

**DEPARTEMEN ILMU KOMPUTER/INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO**

2025

TOPIK 4 TRIGGERS

1. Buat trigger **tr_UpdateAccountBalance_BeforeTransaction** ke tabel transactions yang akan mencegah transaksi dilakukan jika saldo akun tidak cukup. Jika saldo < jumlah amount, maka transaksi harus dibatalkan. Gunakan syntax Rollback.

Screenshot:

The screenshot displays a SQL Server environment. At the top, a window titled 'SQLQuery1.sql - LA...F6QIKPV\adity (78))' contains a SQL script. The script includes comments for a trigger validation task, a data reset command, initial data insertion for customers, accounts, and transaction types, and a final data display command. Below the script, the 'Results' pane shows the output of the 'SELECT * FROM accounts;' query, displaying two rows of account data. The 'Messages' pane is also visible but empty.

```
-- =====
-- SOAL 1: TRIGGER VALIDASI SALDO CUKUP (tr_UpdateAccountBalance_BeforeTransaction)
-- =====

-- RESET DATA AWAL (Hapus semua data terkait)
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

-- INISIALISASI DATA AWAL UNTUK TESTING (SOAL 1-4)
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST001', 'Andi', 'Wijaya', 'andi@mail.com', '081234567890', 'Jakarta'),
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance) VALUES
('ACC001', 'CUST001', '001', 'savings', 500000),
('ACC002', 'CUST002', '002', 'savings', 150000);
INSERT INTO transaction_types (transaction_type_id, name) VALUES
(1, 'Deposit'), (2, 'Transfer'), (3, 'Withdrawal');
GO

-- TAMPILKAN DATA SEBELUM TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;
```

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 13:08:34.140
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 13:08:34.140

transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
----------------	------------	---------------------	--------	------------------	-------------	-------------------

SQLQuery1.sql - LA...F6QIKPV\adity (78))*

```
SELECT * FROM transactions;

-- BUAT TRIGGER: Validasi saldo cukup & update saldo untuk withdrawal/transfer
CREATE OR ALTER TRIGGER tr_UpdateAccountBalance_BeforeTransaction
ON transactions
INSTEAD OF INSERT
AS
BEGIN
    -- Validasi saldo cukup untuk withdrawal/transfer
    IF EXISTS (
        SELECT 1 FROM inserted i
        JOIN accounts a ON i.account_id = a.account_id
        WHERE i.transaction_type_id IN (2,3) AND a.balance < i.amount
    )
    BEGIN
        RAISERROR('Saldo tidak cukup.', 16, 1); ROLLBACK TRANSACTION; RETURN;
    END

    -- Update saldo akun
    UPDATE a
    SET a.balance =
        CASE
            WHEN i.transaction_type_id = 1 THEN a.balance + i.amount -- Deposit
```

90 %

Messages

Commands completed successfully.

Completion time: 2025-05-21T13:09:45.9422178+07:00

SQLQuery1.sql - LA...F6QIKPV\adity (78))*

```
GO

-- TESTING TRIGGER
-- Gagal: saldo tidak cukup
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 3, 600000, GETDATE(), 'Test gagal', NULL);
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH

-- Berhasil: saldo cukup (withdrawal)
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC002', 3, 100000, GETDATE(), 'Test berhasil', NULL);
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH

-- Berhasil: deposit
BEGIN TRY
```

90 %

Messages

(0 rows affected)
Saldo tidak cukup.

Completion time: 2025-05-21T13:11:30.2701927+07:00

```
SQLQuery1.sql - LA...F6QIKPV\adity (78)) *  X
GO

-- TESTING TRIGGER
-- Gagal: saldo tidak cukup
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 3, 600000, GETDATE(), 'Test gagal', NULL);
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH

-- Berhasil: saldo cukup (withdrawal)
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC002', 3, 100000, GETDATE(), 'Test berhasil', NULL);
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH

-- Berhasil: deposit
BEGIN TRY
```

90 %

Messages

(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2025-05-21T13:12:06.5312412+07:00

```
SQLQuery1.sql - LA...F6QIKPV\adity (78)) *  X
    PRINT ERROR_MESSAGE();
END CATCH

-- Berhasil: deposit
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 1, 200000, GETDATE(), 'Test deposit', NULL);
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA (untuk soal berikutnya)
DROP TRIGGER tr_UpdateAccountBalance_BeforeTransaction;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO
```

90 %

Messages

(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2025-05-21T13:12:33.1199910+07:00

SQLQuery1.sql - LA...F6QIKPV\adity (78)) *

```
        PRINT ERROR_MESSAGE();
    END CATCH

    -- Berhasil: deposit
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 1, 200000, GETDATE(), 'Test deposit', NULL);
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA (untuk soal berikutnya)
DROP TRIGGER tr_UpdateAccountBalance_BeforeTransaction;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO
```

90 %

Results

Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	700000.00	2025-05-21 13:08:34.140
2	ACC002	CUST002	002	savings	50000.00	2025-05-21 13:08:34.140

	transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
1	5B0E5174-05C1-46E2-9858-1B1ADD78BDE4	ACC002	3	100000.00	2025-05-21 13:12:06.513	Test berhasil	NULL
2	8169C421-0417-4940-899F-3769E73582BE	ACC001	1	200000.00	2025-05-21 13:12:33.090	Test deposit	NULL

2. Buat trigger **tr_Deposit_AfterTransaction** ke tabel transactions yang aktif jika ada transaksi transfer yang menambahkan transaksi baru ke akun tujuan (reference_account) dengan tipe transaksi deposit dan nilai amount yang sama.

Screenshot:

SQLQuery1.sql - LA...F6QIKPV\adity (78)*

```
-- =====
-- SOAL 2: TRIGGER AUTO DEPOSIT SETELAH TRANSFER (tr_Deposit_AfterTransaction)
-- =====

-- RESET DATA AWAL
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

-- INISIALISASI DATA AWAL
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST001', 'Andi', 'Wijaya', 'andi@mail.com', '081234567890', 'Jakarta'),
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance) VALUES
('ACC001', 'CUST001', '001', 'savings', 500000),
('ACC002', 'CUST002', '002', 'savings', 150000);
INSERT INTO transaction_types (transaction_type_id, name) VALUES
(1, 'Deposit'), (2, 'Transfer'), (3, 'Withdrawal');
GO

-- TAMPILKAN DATA SEBELUM TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;
```

90 %

Results Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 13:14:59.947
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 13:14:59.947

transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
----------------	------------	---------------------	--------	------------------	-------------	-------------------

```
SQLQuery1.sql - LA...F6QIKPV\adity (78)) * X
-- BUAT TRIGGER: Auto deposit ke akun tujuan setelah transfer
CREATE OR ALTER TRIGGER tr_Deposit_AfterTransaction
ON transactions
AFTER INSERT
AS
BEGIN
    -- Insert transaksi deposit otomatis ke akun tujuan transfer
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    SELECT
        i.reference_account, 1, i.amount, GETDATE(), 'Auto deposit from transfer', i.account_id
    FROM inserted i
    WHERE i.transaction_type_id = 2 AND i.reference_account IS NOT NULL;

    -- Update saldo akun tujuan transfer
    UPDATE a
    SET a.balance = a.balance + i.amount
    FROM accounts a
    JOIN inserted i ON a.account_id = i.reference_account
    WHERE i.transaction_type_id = 2 AND i.reference_account IS NOT NULL;
END
GO

90 %
Messages
Commands completed successfully.
Completion time: 2025-05-21T13:16:20.3016489+07:00
```

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78)) * X
        JOIN inserted i ON a.account_id = i.reference_account
        WHERE i.transaction_type_id = 2 AND i.reference_account IS NOT NULL;
    END
    GO

-- TESTING TRIGGER
-- Transfer (harus auto deposit ke tujuan)
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 2, 50000, GETDATE(), 'Test transfer', 'ACC002');

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_Deposit_AfterTransaction;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

-- =====
-- SOAL 3: TRIGGER MAKSIMAL 3 AKUN PER CUSTOMER (tr_LimitAccountsPerCustomer)

90 %
Messages
(1 row affected)
(1 row affected)
(1 row affected)
Completion time: 2025-05-21T14:12:02.0222146+07:00
```

Pert-03_Kel-06.sql...7F6QIKPV\adity (78)

```

JOIN inserted i ON a.account_id = i.reference_account
WHERE i.transaction_type_id = 2 AND i.reference_account IS NOT NULL;
END
GO

-- TESTING TRIGGER
-- Transfer (harus auto deposit ke tujuan)
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 2, 50000, GETDATE(), 'Test transfer', 'ACC002');

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_Deposit_AfterTransaction;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

-- =====
-- SOAL 3: TRIGGER MAKSIMAL 3 AKUN PER CUSTOMER (tr_LimitAccountsPerCustomer)

```

90 %

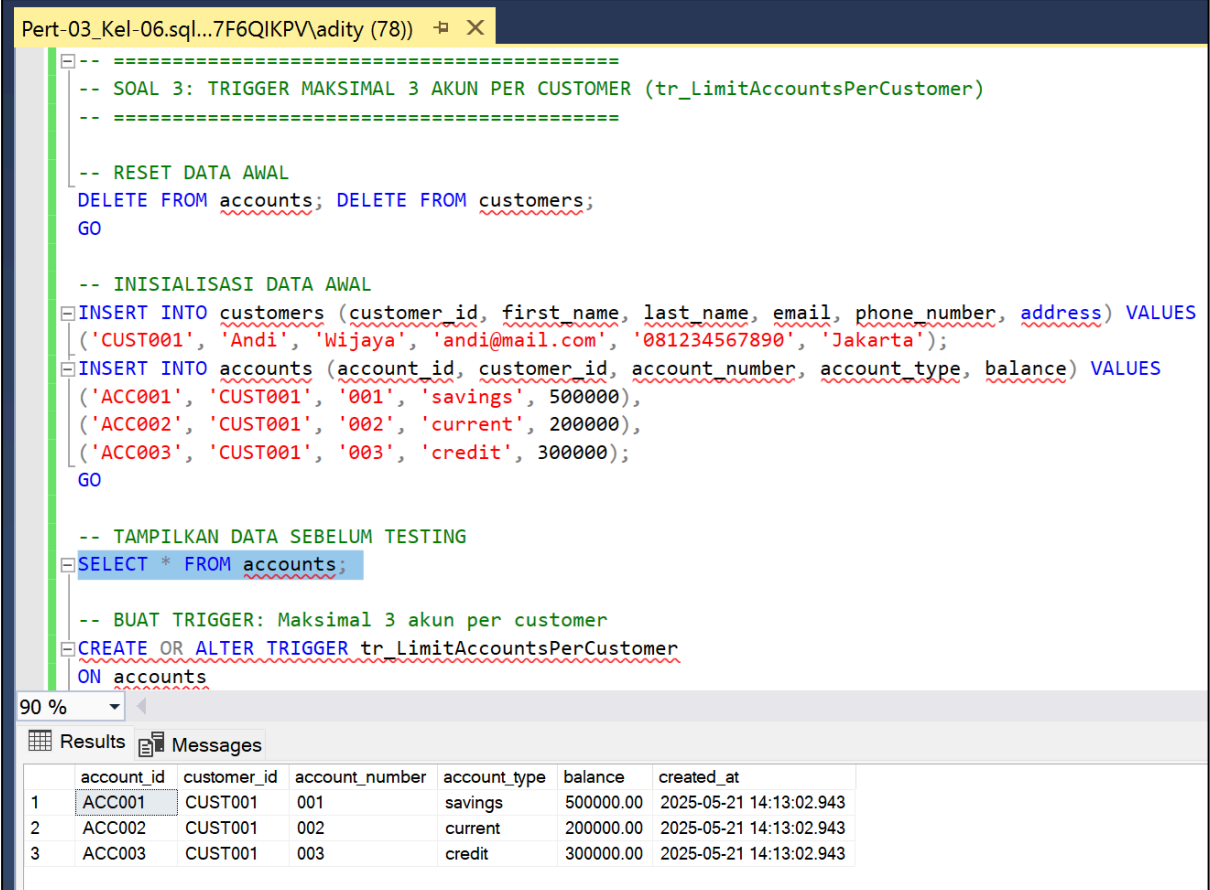
Results Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 13:14:59.947
2	ACC002	CUST002	002	savings	200000.00	2025-05-21 13:14:59.947

	transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
1	E64E4132-8861-4B0C-9413-586F8619BA5F	ACC002	1	50000.00	2025-05-21 14:12:01.983	Auto deposit from transfer	ACC001
2	FAD52C49-6960-4CA1-9C9B-3D6D188F8B9D	ACC001	2	50000.00	2025-05-21 14:12:01.970	Test transfer	ACC002

3. Buat trigger **tr_LimitAccountsPerCustomer** yang akan mencegah penambahan akun baru (accounts) jika seorang nasabah (customer_id) sudah memiliki 3 akun. Trigger ini dijalankan sebelum insert ke tabel accounts. Jika jumlah akun untuk customer_id ≥ 3 , maka transaksi dibatalkan. Gunakan syntax Rollback.

Screenshot:



The screenshot shows a SQL script in a text editor. The script includes comments for a trigger named 'tr_LimitAccountsPerCustomer' which is intended to limit the number of accounts per customer to 3. The script also includes SQL commands to reset and initialize data, insert a customer and three accounts, and create the trigger. Below the script, a 'Results' pane shows the output of a SELECT statement, displaying three accounts for customer CUST001.

```
-- =====
-- SOAL 3: TRIGGER MAKSIMAL 3 AKUN PER CUSTOMER (tr_LimitAccountsPerCustomer)
-- =====

-- RESET DATA AWAL
DELETE FROM accounts; DELETE FROM customers;
GO

-- INISIALISASI DATA AWAL
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST001', 'Andi', 'Wijaya', 'andi@mail.com', '081234567890', 'Jakarta');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance) VALUES
('ACC001', 'CUST001', '001', 'savings', 500000),
('ACC002', 'CUST001', '002', 'current', 200000),
('ACC003', 'CUST001', '003', 'credit', 300000);
GO

-- TAMPILKAN DATA SEBELUM TESTING
SELECT * FROM accounts;

-- BUAT TRIGGER: Maksimal 3 akun per customer
CREATE OR ALTER TRIGGER tr_LimitAccountsPerCustomer
ON accounts
```

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:13:02.943
2	ACC002	CUST001	002	current	200000.00	2025-05-21 14:13:02.943
3	ACC003	CUST001	003	credit	300000.00	2025-05-21 14:13:02.943

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78) X
SELECT * FROM accounts;

-- BUAT TRIGGER: Maksimal 3 akun per customer
CREATE OR ALTER TRIGGER tr_LimitAccountsPerCustomer
ON accounts
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted i
        JOIN (SELECT customer_id, COUNT(*) AS cnt FROM accounts GROUP BY customer_id) a
        ON i.customer_id = a.customer_id
        WHERE a.cnt >= 3
    )
    BEGIN
        RAISERROR('Customer sudah memiliki 3 akun.', 16, 1); ROLLBACK TRANSACTION; RETURN;
    END
    INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
    SELECT account_id, customer_id, account_number, account_type, balance FROM inserted;
END
GO

-- TESTING TRIGGER
```

90 %

Messages

Commands completed successfully.

Completion time: 2025-05-21T14:15:07.5313967+07:00

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78) X
SELECT account_id, customer_id, account_number, account_type, balance FROM inserted;
END
GO

-- TESTING TRIGGER
-- Gagal: akun ke-4 customer sama
BEGIN TRY
    INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
    VALUES ('ACC004', 'CUST001', '004', 'savings', 100000);
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH

-- Berhasil: akun pertama customer baru
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
VALUES ('ACC005', 'CUST002', '005', 'savings', 150000);

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_LimitAccountsPerCustomer;
DELETE FROM accounts; DELETE FROM customers;
```

90 %

Messages

(0 rows affected)
Customer sudah memiliki 3 akun.

Completion time: 2025-05-21T14:15:36.0519709+07:00

Pert-03_Kel-06.sql...7F6QIKPV\adity (78)

```

SELECT account_id, customer_id, account_number, account_type, balance FROM inserted;
END
GO

-- TESTING TRIGGER
-- Gagal: akun ke-4 customer sama
BEGIN TRY
    INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
    VALUES ('ACC004', 'CUST001', '004', 'savings', 100000);
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH

-- Berhasil: akun pertama customer baru
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
VALUES ('ACC005', 'CUST002', '005', 'savings', 150000);

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_LimitAccountsPerCustomer;
DELETE FROM accounts; DELETE FROM customers;

```

90 %

Messages

Msg 2627, Level 14, State 1, Line 208
Violation of PRIMARY KEY constraint 'PK_customer_CD65CB857FCE98C7'. Cannot insert duplicate key in object 'dbo.customers'. The duplicate key value is (CUST002)
The statement has been terminated.

(1 row affected)

(1 row affected)

Completion time: 2025-05-21T14:16:04.7582139+07:00

Pert-03_Kel-06.sql...7F6QIKPV\adity (78)

```

-- TESTING TRIGGER
-- Gagal: akun ke-4 customer sama
BEGIN TRY
    INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
    VALUES ('ACC004', 'CUST001', '004', 'savings', 100000);
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH

-- Berhasil: akun pertama customer baru
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
VALUES ('ACC005', 'CUST002', '005', 'savings', 150000);

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_LimitAccountsPerCustomer;
DELETE FROM accounts; DELETE FROM customers;
GO

```

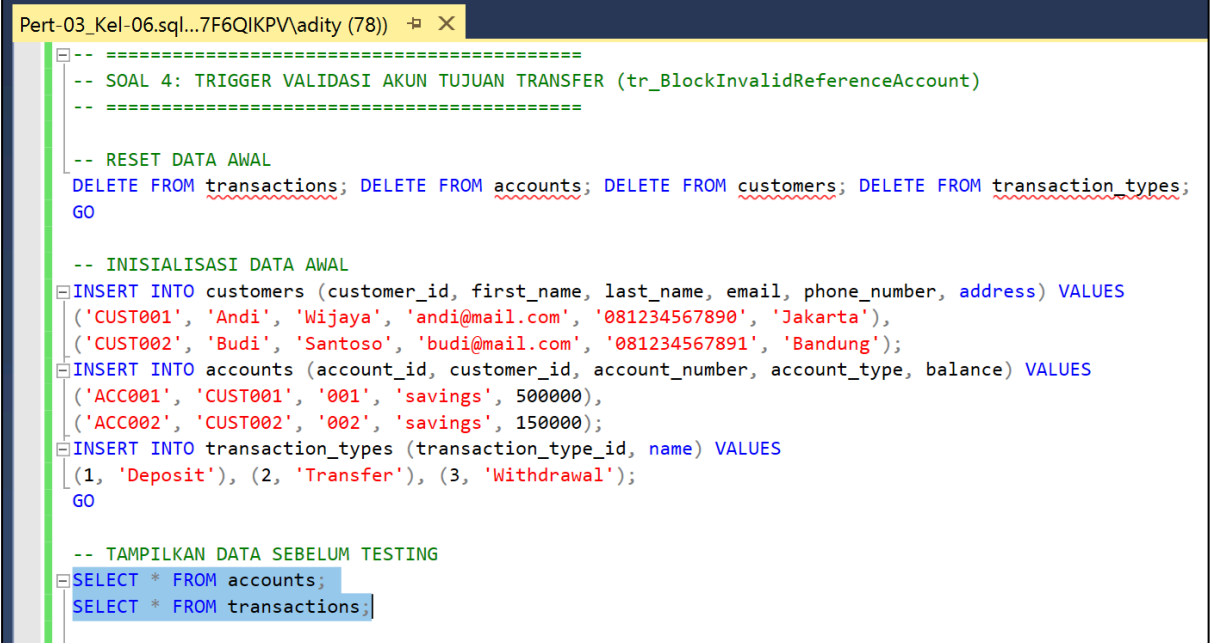
90 %

Results Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:13:02.943
2	ACC002	CUST001	002	current	200000.00	2025-05-21 14:13:02.943
3	ACC003	CUST001	003	credit	300000.00	2025-05-21 14:13:02.943
4	ACC005	CUST002	005	savings	150000.00	2025-05-21 14:16:04.713

4. Buat trigger **tr_BlockInvalidReferenceAccount** pada tabel transactions yang aktif jika terdapat insert transaksi transfer, sistem harus mengecek apakah akun tujuan (reference_account) benar-benar ada di tabel accounts. Jika tidak ditemukan, batalkan transaksi. Gunakan syntax Rollback.

Screenshot:



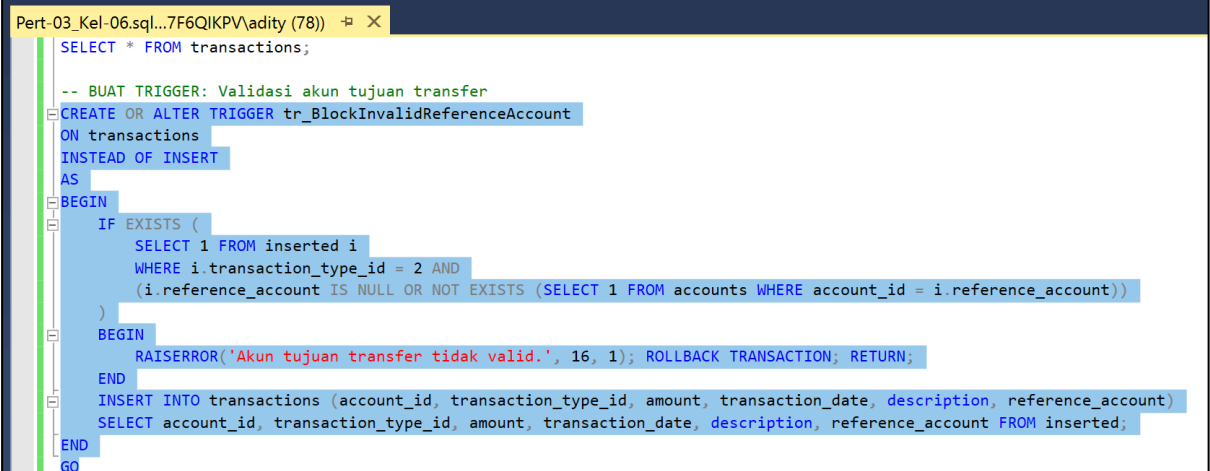
```
-- =====
-- SOAL 4: TRIGGER VALIDASI AKUN TUJUAN TRANSFER (tr_BlockInvalidReferenceAccount)
-- =====

-- RESET DATA AWAL
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

-- INISIALISASI DATA AWAL
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST001', 'Andi', 'Wijaya', 'andi@mail.com', '081234567890', 'Jakarta'),
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance) VALUES
('ACC001', 'CUST001', '001', 'savings', 500000),
('ACC002', 'CUST002', '002', 'savings', 150000);
INSERT INTO transaction_types (transaction_type_id, name) VALUES
(1, 'Deposit'), (2, 'Transfer'), (3, 'Withdrawal');
GO

-- TAMPILKAN DATA SEBELUM TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;
```

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:17:08.980
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:17:08.980



```
SELECT * FROM transactions;

-- BUAT TRIGGER: Validasi akun tujuan transfer
CREATE OR ALTER TRIGGER tr_BlockInvalidReferenceAccount
ON transactions
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted i
        WHERE i.transaction_type_id = 2 AND
        (i.reference_account IS NULL OR NOT EXISTS (SELECT 1 FROM accounts WHERE account_id = i.reference_account))
    )
    BEGIN
        RAISERROR('Akun tujuan transfer tidak valid.', 16, 1); ROLLBACK TRANSACTION; RETURN;
    END
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    SELECT account_id, transaction_type_id, amount, transaction_date, description, reference_account FROM inserted;
END
GO

-- TESTING TRIGGER
-- Gagal: akun tujuan tidak valid
```

Commands completed successfully.

Completion time: 2025-05-21T14:17:56.0428173+07:00

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78)  X
END
GO

-- TESTING TRIGGER
-- Gagal: akun tujuan tidak valid
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 2, 50000, GETDATE(), 'Test gagal', 'ACC999');
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH
-- Berhasil: akun tujuan valid
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 2, 50000, GETDATE(), 'Test berhasil', 'ACC002');

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_BlockInvalidReferenceAccount;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO
```

90 %

Messages

(0 rows affected)
Akun tujuan transfer tidak valid.
Completion time: 2025-05-21T14:18:23.4198122+07:00

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78)  X
END
GO

-- TESTING TRIGGER
-- Gagal: akun tujuan tidak valid
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 2, 50000, GETDATE(), 'Test gagal', 'ACC999');
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH
-- Berhasil: akun tujuan valid
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 2, 50000, GETDATE(), 'Test berhasil', 'ACC002');

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_BlockInvalidReferenceAccount;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO
```

90 %

Messages

(1 row affected)
(1 row affected)
Completion time: 2025-05-21T14:18:46.3740598+07:00

Pert-03_Kel-06.sql...7F6QIKPV\adity (78)

END
GO

-- TESTING TRIGGER
-- Gagal: akun tujuan tidak valid
BEGIN TRY
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 2, 50000, GETDATE(), 'Test gagal', 'ACC999');
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH
-- Berhasil: akun tujuan valid
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 2, 50000, GETDATE(), 'Test berhasil', 'ACC002');

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_BlockInvalidReferenceAccount;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

90 %

Results Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:17:08.980
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:17:08.980

	transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
1	68151202-365E-4983-9D8C-AF2FBEC13F31	ACC001	2	50000.00	2025-05-21 14:18:46.320	Test berhasil	ACC002

5. Buat trigger **tr_EnsureMinimumBalance** di tabel transactions yang aktif saat transaksi withdrawal atau transfer dilakukan. Trigger harus memeriksa apakah saldo akun setelah transaksi memiliki minimal Rp100.000. Jika tidak, transaksi dibatalkan. Gunakan syntax Rollback.

Screenshot:

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78))
-- =====
-- SOAL 5: TRIGGER SALDO MINIMAL 100.000 (tr_EnsureMinimumBalance)
-- =====

-- RESET DATA AWAL
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

-- INISIALISASI DATA AWAL
INSERT INTO customers (customer_id, first_name, last_name, email, phone_number, address) VALUES
('CUST001', 'Andi', 'Wijaya', 'andi@mail.com', '081234567890', 'Jakarta'),
('CUST002', 'Budi', 'Santoso', 'budi@mail.com', '081234567891', 'Bandung');
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance) VALUES
('ACC001', 'CUST001', '001', 'savings', 500000),
('ACC002', 'CUST002', '002', 'savings', 150000);
INSERT INTO transaction_types (transaction_type_id, name) VALUES
(1, 'Deposit'), (2, 'Transfer'), (3, 'Withdrawal');
GO

-- TAMPILKAN DATA SEBELUM TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- BUAT TRIGGER: Saldo minimal 100.000 setelah transaksi
```

90 %

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:19:44.893
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:19:44.893

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78))
-- TAMPILKAN DATA SEBELUM TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- BUAT TRIGGER: Saldo minimal 100.000 setelah transaksi
CREATE OR ALTER TRIGGER tr_EnsureMinimumBalance
ON transactions
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted i JOIN accounts a ON i.account_id = a.account_id
        WHERE i.transaction_type_id IN (2,3) AND (a.balance - i.amount) < 100000
    )
    BEGIN
        RAISERROR('Saldo setelah transaksi harus minimal 100.000.', 16, 1); ROLLBACK TRANSACTION; RETURN;
    END
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    SELECT account_id, transaction_type_id, amount, transaction_date, description, reference_account FROM inserted;
END
GO

-- TESTING TRIGGER
```

90 %

Messages

Commands completed successfully.

Completion time: 2025-05-21T14:20:29.3083998+07:00

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78) X
SELECT account_id, transaction_type_id, amount, transaction_date, description, reference_account FROM inserted;
END
GO

-- TESTING TRIGGER
-- Gagal: saldo akhir < 100.000
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 3, 150000, GETDATE(), 'Test gagal', NULL);
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH
-- Berhasil: saldo akhir cukup
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 3, 50000, GETDATE(), 'Test berhasil', NULL);

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_EnsureMinimumBalance;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

90 %
Messages
(1 row affected)
(1 row affected)
Completion time: 2025-05-21T14:20:54.7930856+07:00
```

```
Pert-03_Kel-06.sql...7F6QIKPV\adity (78) X
SELECT account_id, transaction_type_id, amount, transaction_date, description, reference_account FROM inserted;
END
GO

-- TESTING TRIGGER
-- Gagal: saldo akhir < 100.000
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 3, 150000, GETDATE(), 'Test gagal', NULL);
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH
-- Berhasil: saldo akhir cukup
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 3, 50000, GETDATE(), 'Test berhasil', NULL);

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_EnsureMinimumBalance;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

90 %
Messages
(1 row affected)
(1 row affected)
Completion time: 2025-05-21T14:21:20.4101573+07:00
```


Pert-03_Kel-06.sql...7F6QIKPV\adity (78)

```

SELECT account_id, transaction_type_id, amount, transaction_date, description, reference_account FROM inserted;
END
GO

-- TESTING TRIGGER
-- Gagal: saldo akhir < 100.000
BEGIN TRY
    INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
    VALUES ('ACC001', 3, 150000, GETDATE(), 'Test gagal', NULL);
END TRY BEGIN CATCH PRINT ERROR_MESSAGE(); END CATCH
-- Berhasil: saldo akhir cukup
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description, reference_account)
VALUES ('ACC001', 3, 50000, GETDATE(), 'Test berhasil', NULL);

-- TAMPILKAN DATA SESUDAH TESTING
SELECT * FROM accounts;
SELECT * FROM transactions;

-- HAPUS TRIGGER & RESET DATA
DROP TRIGGER tr_EnsureMinimumBalance;
DELETE FROM transactions; DELETE FROM accounts; DELETE FROM customers; DELETE FROM transaction_types;
GO

```

90 % ◀

Results

Messages

	account_id	customer_id	account_number	account_type	balance	created_at
1	ACC001	CUST001	001	savings	500000.00	2025-05-21 14:19:44.893
2	ACC002	CUST002	002	savings	150000.00	2025-05-21 14:19:44.893

	transaction_id	account_id	transaction_type_id	amount	transaction_date	description	reference_account
1	382D1116-3045-4493-B11B-40E93F828FFE	ACC001	3	50000.00	2025-05-21 14:21:20.357	Test berhasil	NULL
2	9BA28F66-94B5-4017-8404-31AFDE661B9B	ACC001	3	150000.00	2025-05-21 14:20:54.777	Test gagal	NULL