

LAPORAN PRAKTIKUM PERTEMUAN 3 MBD A2 KELOMPOK 4

Amara Putri Soniaji (24060123120004)

Akmal Fauzan Azima (24060123120022)

Cacania Pasu N.S. (24060123120033)

Brilianita Eva Syafitri (24060123130051)

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.

```
CREATE TRIGGER tr_UpdateAccountBalance_BeforeTransaction
ON transactions
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @account_id CHAR(36), @amount DECIMAL(10,2), @total_saldo DECIMAL(10,2);

    SELECT @account_id = account_id, @amount = amount
    FROM inserted;

    SELECT @total_saldo = balance
    FROM accounts
    WHERE account_id = @account_id;

    IF @total_saldo < @amount
    BEGIN
        RAISERROR('Saldo tidak mencukupi', 10, 1)
        ROLLBACK TRANSACTION;
        RETURN;
    END
END
```

%

Messages

Commands completed successfully.

```
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;

UPDATE accounts
SET balance = @total_saldo - @amount
WHERE account_id = @account_id;
END

INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description)
VALUES ('6edc13ce-6ed2-48a7-b73d-ef6ed8b73e38', 1, 10000.00, GETDATE(), 'Penarikan ATM');
```

%

Messages

Saldo tidak mencukupi
Msg 3609, Level 16, State 1, Line 38
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2025-05-18T13:54:50.0933406+07:00

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.

```
USE BANKDB;

CREATE TRIGGER tr_Deposit_AfterTransaction
ON transactions
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO transactions (account_id, transaction_type_id, amount, description)
    SELECT a.account_id, 1, i.amount, 'Penerimaan transfer dari akun lain'
    FROM inserted i
    JOIN accounts a ON a.account_number = i.reference_account
    WHERE i.transaction_type_id = 3;
END

-- Tambah dua akun untuk pengujian
INSERT INTO customers (first_name, last_name, email, phone_number, address)
VALUES ('Akmal', 'FA', 'akmal@gmail.com', '08991234567', 'Bandung');

DECLARE @cust_id CHAR(36) = (SELECT TOP 1 customer_id FROM customers WHERE first_name = 'Akmal');

-- Akun asal (111111111)
INSERT INTO accounts (customer_id, account_number, account_type, balance)
VALUES (@cust_id, '111111111', 'savings', 200000.00);

-- Akun tujuan (222222222)
INSERT INTO accounts (customer_id, account_number, account_type, balance)
VALUES (@cust_id, '222222222', 'savings', 100000.00);

-- Lakukan transaksi transfer
DECLARE @acc_id CHAR(36) = (
    SELECT account_id FROM accounts WHERE account_number = '111111111'
);

INSERT INTO transactions (account_id, transaction_type_id, amount, description, reference_account)
VALUES (@acc_id, 3, 50000.00, 'Transfer ke 222222222', '222222222');

-- Lihat hasilnya (daftar transaksi)
SELECT t.transaction_id, a.account_number, t.transaction_type_id, t.amount, t.description
FROM transactions t
JOIN accounts a ON t.account_id = a.account_id
ORDER BY t.transaction_date DESC;
```

100 %

Results Messages

	transaction_id	account_number	transaction_type_id	amount	description
1	310DD593-8810-4608-8500-65557A77ECBF	222222222	1	50000.00	Penerimaan transfer dari akun lain
2	5861D318-5008-4CC5-B9ED-40AE74FE40A9	111111111	3	50000.00	Transfer ke 222222222

Query executed successfully. WINDOWS-PCPNN7P\MS

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` \geq maka transaksi dibatalkan. Gunakan syntax Rollback.

```
CREATE TRIGGER tr_LimitAccountsPerCustomer
ON accounts
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1 FROM inserted i
        JOIN (SELECT customer_id, COUNT(*) AS jml FROM accounts GROUP BY customer_id) c
        ON i.customer_id = c.customer_id
        WHERE c.jml >= 3
    )
    BEGIN
        RAISERROR('Customer sudah punya 3 akun', 16, 1);
        ROLLBACK;
        RETURN;
    END

    INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance, created_at)
    SELECT NEWID(), customer_id, account_number, account_type, balance, GETDATE()
    FROM inserted;
END;
```

%

Messages

Commands completed successfully.

```
INSERT INTO accounts (customer_id, account_number, account_type, balance)
VALUES ('d01d990c-8cbd-4b0f-87b3-4c31fafbf3d5', '0000000003', 'savings', 500000);

SELECT account_id, account_number, account_type, balance
FROM accounts
WHERE customer_id = 'd01d990c-8cbd-4b0f-87b3-4c31fafbf3d5';
```

0 %

Results Messages

	account_id	account_number	account_type	balance
1	A159BFC8-224A-46DA-B010-FCE39CC2AF31	0000000003	savings	500000.00
2	da3b43bc-55d1-4f0b-80d1-494ccfc9acc5	0587333038	savings	10257.16

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.

```
CREATE TRIGGER tr_BlockInvalidReferenceAccount
ON transactions
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted i
        WHERE i.reference_account IS NOT NULL
        AND NOT EXISTS (
            SELECT 1
            FROM accounts a
            WHERE a.account_id = i.reference_account
        )
    )
    BEGIN
        RAISERROR('Reference_account tidak ditemukan.', 16, 1);
        ROLLBACK;
        RETURN;
    END
END;

-- Pengujian
INSERT INTO transactions (account_id, transaction_type_id, amount, reference_account)
VALUES ('6edc13ce-6ed2-48a7-b73d-ef6ed8b73e38', 3, 100000, '4734b927-4bba-4591-a8fc');
```

Messages

Msg 50000, Level 16, State 1, Procedure tr_BlockInvalidReferenceAccount, Line 19 [Batch Start Line 25]
Reference_account tidak ditemukan.
Msg 3609, Level 16, State 1, Line 26
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2025-05-20T00:07:02.0923314+07:00

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.

```

USE BANKDB;

CREATE TRIGGER tr_EnsureMinimumBalance
ON transactions
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    -- Batalkan jika transaksi menyebabkan saldo akhir < 100.000
    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 Rp100.000.', 16, 1);
        ROLLBACK;
        RETURN;
    END

    -- Jika saldo mencukupi, lanjutkan transaksi
    INSERT INTO transactions (account_id, transaction_type_id, amount, description, reference_account)
    SELECT account_id, transaction_type_id, amount, description, reference_account
    FROM inserted;
END

-- Tambahkan akun untuk pengujian
INSERT INTO customers (first_name, last_name, email, phone_number, address)
VALUES ('Fauzan', 'A', 'fauzan@gmail.com', '08111222333', 'Surabaya');

DECLARE @cust_id CHAR(36) = (
    SELECT TOP 1 customer_id FROM customers WHERE first_name = 'Fauzan'
);

-- Akun dengan saldo Rp150.000
INSERT INTO accounts (customer_id, account_number, account_type, balance)
VALUES (@cust_id, '3333333333', 'savings', 150000.00);

-- Uji Transaksi DITOLAK (karena saldo akhir < 100.000)
DECLARE @acc_id CHAR(36) = (
    SELECT account_id FROM accounts WHERE account_number = '3333333333'
);

-- Tarik Rp60.000, hasil akhir = 90.000 < 100.000 (DITOLAK)
INSERT INTO transactions (account_id, transaction_type_id, amount, description)
VALUES (@acc_id, 2, 60000.00, 'Tarik tunai Rp60.000');

```

Messages

Msg 50000, Level 16, State 1, Procedure tr_EnsureMinimumBalance, Line 17 [Batch Start Line 40]
Saldo setelah transaksi harus minimal Rp100.000.
Msg 3609, Level 16, State 1, Line 46
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2025-05-18T22:50:04.1304224+07:00

100 %

Query completed with errors. WINDOWS-PC

Di bawah ini adalah pengujian jika transaksi diterima:

```
-- Uji Transaksi DITERIMA (saldo akhir >= 100.000)
DECLARE @acc_id CHAR(36) = (
    SELECT account_id FROM accounts WHERE account_number = '3333333333'
);
-- Tarik Rp30.000, hasil akhir = 120.000 (DITERIMA)
INSERT INTO transactions (account_id, transaction_type_id, amount, description)
VALUES (@acc_id, 2, 30000.00, 'Tarik tunai Rp30.000');
-- Lihat hasilnya (daftar transaksi)
SELECT t.transaction_id, a.account_number, t.transaction_type_id, t.amount, t.description
FROM transactions t
JOIN accounts a ON t.account_id = a.account_id
WHERE a.account_number = '3333333333'
ORDER BY t.transaction_date DESC;
```

100 %

Results **Messages**

	transaction_id	account_number	transaction_type_id	amount	description
1	930C43C4-2DA7-47C7-A8A1-0FF9BF6E5C34	3333333333	2	30000.00	Tarik tunai Rp30.000

Query executed successfully. WINDOWS-PC