

TUGAS TRIGGERS  
PRAKTIKUM MATA KULIAH MANAJEMEN BASIS DATA  
PERTEMUAN ke-3



Disusun oleh:  
Kelompok 3

Anggota:

1. Anisa Rafa Puspita (24060123120001)
2. Angel Mentari Bulawan Pariakan (24060123140200)
3. Alisha Fatima Putri Kurniawan (24060123140048)
4. Aninditya Nabilah Hendarto (24060123140134)
5. Arya Pradana (24060123140125)

PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN MATEMATIKA  
UNIVERSITAS DIPONEGORO

Buatlah kemudian aplikasikan query trigger untuk kasus berikut:

i. 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.

- Query:

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

    -- Cek saldo setiap transaksi yang akan dimasukkan
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN accounts a ON i.account_id = a.account_id
        WHERE i.amount > a.balance
    )
    BEGIN
        RAISERROR('Saldo tidak mencukupi untuk transaksi ini.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika saldo mencukupi, izinkan insert
    INSERT INTO transactions (
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    )
    SELECT
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    FROM inserted;

    END;

INSERT INTO transactions (transaction_id, account_id, transaction_type_id,
amount,
transaction_date, description, reference_account)
VALUES (NEWID(), '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 1,
10000.00, GETDATE(), 'Penarikan Besar', NULL);

-- coba transaksi berhasil
SELECT balance FROM accounts WHERE account_id =
'9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7';
-- Saldo: 9038.34

INSERT INTO transactions (transaction_id, account_id, transaction_type_id,
amount,
transaction_date, description, reference_account)
VALUES (NEWID(), '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 1,
1000.00, GETDATE(), 'Pembayaran Tagihan', NULL);
```

- Screenshot:

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

    -- Cek saldo setiap transaksi yang akan dimasukkan
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN accounts a ON i.account_id = a.account_id
        WHERE i.amount > a.balance
    )
    BEGIN
        RAISERROR('Saldo tidak mencukupi untuk transaksi ini.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika saldo mencukupi, izinkan insert
    INSERT INTO transactions (
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-05-22T14:11:00.9020564+07:00

```
INSERT INTO transactions (transaction_id, account_id, transaction_type_id, amount,
transaction_date, description, reference_account)
VALUES (NEWID(), '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 1,
1000.00, GETDATE(), 'Penarikan Besar', NULL);
```

100 %

Messages

Msg 50000, Level 16, State 1, Procedure tr\_UpdateAccountBalance\_BeforeTransaction, Line 16 [Batch Start Line 38]  
Saldo tidak mencukupi untuk transaksi ini.  
Msg 3609, Level 16, State 1, Line 39  
The transaction ended in the trigger. The batch has been aborted.

Completion time: 2025-05-22T14:15:44.6517300+07:00

```
-- coba transaksi berhasil
SELECT balance FROM accounts WHERE account_id = '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7';
-- Saldo: 9038.34

INSERT INTO transactions (transaction_id, account_id, transaction_type_id, amount,
transaction_date, description, reference_account)
VALUES (NEWID(), '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 1,
1000.00, GETDATE(), 'Pembayaran Tagihan', NULL);
```

100 %

Results

balance
9038.34

```
INSERT INTO transactions (transaction_id, account_id, transaction_type_id, amount,
transaction_date, description, reference_account)
VALUES (NEWID(), '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 1,
1000.00, GETDATE(), 'Pembayaran Tagihan', NULL);
```

100 %

Messages

(1 row affected)

Completion time: 2025-05-22T14:18:32.2629500+07:00

ii. 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.

- Query:

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

    INSERT INTO transactions (
        transaction_id,
        account_id,
        transaction_type_id,
        amount,
        transaction_date,
        description,
        reference_account
    )
    SELECT
        NEWID(),
        i.reference_account, 1, i.amount, GETDATE(), LEFT(CONCAT('Transfer
received from account ', i.account_id), 50),
        i.account_id
    FROM inserted i
    WHERE i.transaction_type_id = 2 AND i.reference_account IS NOT NULL;
END;
```

```
-- coba transaksi
INSERT INTO transactions (
    transaction_id,
    account_id,
    transaction_type_id,
    amount,
    transaction_date,
    description,
    reference_account
)
VALUES (
    NEWID(),
    '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', -- pengirim
    2, -- transfer
    500.00,
    GETDATE(),
    'Transfer ke akun lain',
    '905660c8-d6a7-40e9-9be3-e6711d22efb6' -- penerima
);
```

```
-- cek hasil
SELECT *
FROM transactions
WHERE account_id = '905660c8-d6a7-40e9-9be3-e6711d22efb6';
```

- Screenshoot:

The first screenshot shows the creation of a trigger named `tr_deposit_AfterTransaction` on the `transactions` table. The trigger is set to fire `AFTER INSERT` and contains logic to insert a new record into the `transactions` table with a new ID, a reference account ID, a transaction type ID of 1, the amount, the current date, a description, and the reference account. The trigger is set to `SET NOCOUNT ON;` and ends with `END;`.

The second screenshot shows an `INSERT INTO transactions` statement with the following values: `NEWID()`, `'9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7'` (commented as `-- pengirim`), `2` (commented as `-- transfer`), `500.00`, `GETDATE()`, `'Transfer ke akun lain'`, and `'905660c8-d6a7-40e9-9be3-e6711d22efb6'` (commented as `-- penerima`). The message below the statement indicates `(1 row affected)`.

The third screenshot shows a query result for the `transactions` table. The query is `SELECT * FROM transactions WHERE account_id = '905660c8-d6a7-40e9-9be3-e6711d22efb6'`. The results table has columns: `transaction_id`, `account_id`, `transaction_type_id`, `amount`, `transaction_date`, and `description`. The first row shows a transaction with ID `1800E6CB-A4DE-4E0D-8436-AF010D8EF3B8`, account ID `905660c8-d6a7-40e9-9be3-e6711d22efb6`, type ID `1`, amount `500.00`, date `2025-05-22 14:35:16.850`, and description `Transfer received from account 9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7`. The second row shows a transaction with ID `149e2d66-5f0c-45fb-ac4c-6c51b226f78d`, account ID `905660c8-d6a7-40e9-9be3-e6711d22efb6`, type ID `3`, amount `118.44`, date `2025-03-21 05:47:34.000`, and description `Affectus et firmitatem animi nec mortem nec dol`.

iii. 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`  $\geq 3$ , maka transaksi dibatalkan. Gunakan syntax `Rollback`.

- Query:

```
CREATE TRIGGER tr_LimitAccountsPerCustomer
ON accounts
INSTEAD OF INSERT
AS
```

```

BEGIN
    SET NOCOUNT ON;

    -- Cek apakah ada customer yang sudah memiliki >= 3 akun
    IF EXISTS (
        SELECT i.customer_id
        FROM inserted i
        JOIN accounts a ON i.customer_id = a.customer_id
        GROUP BY i.customer_id
        HAVING COUNT(a.account_id) >= 3
    )
    BEGIN
        RAISERROR('Customer ini sudah memiliki 3 akun. Tidak dapat
menambahkan akun baru.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika lolos validasi, lanjutkan insert
    INSERT INTO accounts (
        account_id, customer_id, account_number, account_type, balance,
        created_at
    )
    SELECT
        account_id, customer_id, account_number, account_type, balance,
        created_at
    FROM inserted;
END;

-- cari customer dengan 3 akun
SELECT customer_id, COUNT(*) AS account_count
FROM accounts
GROUP BY customer_id
HAVING COUNT(*) = 3;

-- coba masukkan akun baru
INSERT INTO accounts (account_id, customer_id, account_number, account_type,
balance)
VALUES
('new-account-id-1', '9d78736f-df51-4622-9cb1-c4db88dca2d0', '1234567890',
'savings', 1000.00),
('new-account-id-2', '9d78736f-df51-4622-9cb1-c4db88dca2d0', '1234567891',
'current', 1500.00),
('new-account-id-3', 'f8628b73-c900-4973-849e-22aa8e42fc33', '1234567892',
'credit', 2000.00),
('new-account-id-4', 'f8628b73-c900-4973-849e-22aa8e42fc33', '1234567893',
'savings', 2500.00),
('new-account-id-5', '3f5ab011-f526-4bbd-8ab8-8a67ad95c5bf', '1234567894',
'current', 3000.00),
('new-account-id-6', '3f5ab011-f526-4bbd-8ab8-8a67ad95c5bf', '1234567895',
'credit', 3500.00);

```

- Screenshot:

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

    -- Cek apakah ada customer yang sudah memiliki >= 3 akun
    IF EXISTS (
        SELECT i.customer_id
        FROM inserted i
        JOIN accounts a ON i.customer_id = a.customer_id
        GROUP BY i.customer_id
        HAVING COUNT(a.account_id) >= 3
    )
    BEGIN
        RAISERROR('Customer ini sudah memiliki 3 akun. Tidak dapat menambahkan akun baru.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika lolos validasi, lanjutkan insert
    INSERT INTO accounts (
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-05-22T14:56:49.8972513+07:00

```
        -- Jika lolos validasi, lanjutkan insert
        INSERT INTO accounts (
            account_id, customer_id, account_number, account_type, balance, created_at
        )
        SELECT
            account_id, customer_id, account_number, account_type, balance, created_at
        FROM inserted;
    END;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-05-22T14:56:49.8972513+07:00

```
-- cari customer dengan 3 akun
SELECT customer_id, COUNT(*) AS account_count
FROM accounts
GROUP BY customer_id
HAVING COUNT(*) = 3;
```

100 %

Results Messages

customer_id	account_count
-------------	---------------

```
-- coba masukkan akun baru
INSERT INTO accounts (account_id, customer_id, account_number, account_type, balance)
VALUES
    ('new-account-id-1', '9d78736f-df51-4622-9cb1-c4db88dca2d0', '1234567890', 'savings', 1000.00),
    ('new-account-id-2', '9d78736f-df51-4622-9cb1-c4db88dca2d0', '1234567891', 'current', 1500.00),
    ('new-account-id-3', 'f8628b73-c900-4973-849e-22aa8e42fc33', '1234567892', 'credit', 2000.00),
    ('new-account-id-4', 'f8628b73-c900-4973-849e-22aa8e42fc33', '1234567893', 'savings', 2500.00),
    ('new-account-id-5', '3f5ab011-f526-4bbd-8ab8-8a67ad95c5bf', '1234567894', 'current', 3000.00),
    ('new-account-id-6', '3f5ab011-f526-4bbd-8ab8-8a67ad95c5bf', '1234567895', 'credit', 3500.00);
```

100 %

Messages

Msg 50000, Level 16, State 1, Procedure tr\_LimitAccountsPerCustomer, Line 17 [Batch Start Line 149]  
Customer ini sudah memiliki 3 akun. Tidak dapat menambahkan akun baru.  
Msg 3609, Level 16, State 1, Line 150

iv. 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`.

- Query:

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

    -- Cek apakah ada transaksi transfer yang reference_account-nya tidak
    valid
    IF EXISTS (
        SELECT 1
        FROM inserted i
        WHERE i.transaction_type_id = 2 -- Asumsikan 2 = transfer
        AND (
            i.reference_account IS NULL
            OR NOT EXISTS (
                SELECT 1
                FROM accounts a
                WHERE a.account_id = i.reference_account
            )
        )
    )
    BEGIN
        RAISERROR('Akun tujuan (reference_account) tidak valid atau tidak
        ditemukan.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika semua valid, lanjutkan insert
    INSERT INTO transactions (
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    )
    SELECT
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    FROM inserted;
END;

-- pengujian
INSERT INTO transactions (
    transaction_id,
    account_id,
    transaction_type_id,
    amount,
    transaction_date,
    description,
    reference_account
)
VALUES (
    NEWID(),
```



```

        '8c46fae1-9789-4427-8440-7cf48a0272cd',
        2,
        100.00,
        GETDATE(),
        'Transfer ke akun tidak valid',
        '00000000-0000-0000-0000-000000000000'
    );

```

- Screenshoot:

The first screenshot shows the creation of a trigger named `tr_BlockInvalidReferenceAccount` on the `transactions` table. The trigger is an `INSTEAD OF INSERT` trigger. It begins by setting `NOCOUNT ON`. A comment indicates the purpose: "-- Cek apakah ada transaksi transfer yang reference\_account-nya tidak valid". The logic uses an `IF EXISTS` clause to check if there is an inserted transaction with `transaction_type_id = 2` (assumed to be transfer) and either a `NULL` `reference_account` or no corresponding account in the `accounts` table. If this condition is met, it raises an error: `RAISERROR('Akun tujuan (reference account) tidak valid atau tidak ditemukan.', 16, 1);`. The second screenshot shows the continuation of the trigger logic. After the error-raising block, there is a `ROLLBACK TRANSACTION;` and a `RETURN;` statement. If the condition is not met, it proceeds to insert the data from the `inserted` table into the `transactions` table. The third screenshot shows a test query. A comment `-- pengujian` is followed by an `INSERT INTO transactions` statement with values: a new ID, the UUID from the first screenshot, 2, 100.00, the current date, the description 'Transfer ke akun tidak valid', and the zero UUID. The messages pane shows two messages: "Msg 50000, Level 16, State 1, Procedure tr\_BlockInvalidReferenceAccount, Line 23 [Batch Start Line 203] Akun tujuan (reference account) tidak valid atau tidak ditemukan." and "Msg 3609, Level 16, State 1, Line 204".

```

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

    -- Cek apakah ada transaksi transfer yang reference_account-nya tidak valid
    IF EXISTS (
        SELECT 1
        FROM inserted i
        WHERE i.transaction_type_id = 2 -- Asumsikan 2 = transfer
        AND (
            i.reference_account IS NULL
            OR NOT EXISTS (
                SELECT 1
                FROM accounts a
                WHERE a.account_id = i.reference_account
            )
        )
    )
    BEGIN
        RAISERROR('Akun tujuan (reference account) tidak valid atau tidak ditemukan.', 16, 1);
    END
    BEGIN
        ROLLBACK TRANSACTION;
        RETURN;
    END
    -- Jika semua valid, lanjutkan insert
    INSERT INTO transactions (
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    )
    SELECT
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    FROM inserted;
END;

```

```

-- pengujian
INSERT INTO transactions (
    transaction_id,
    account_id,
    transaction_type_id,
    amount,
    transaction_date,
    description,
    reference_account
)
VALUES (
    NEWID(),
    '8c46fae1-9789-4427-8440-7cf48a0272cd',
    2,
    100.00,
    GETDATE(),
    'Transfer ke akun tidak valid',
    '00000000-0000-0000-0000-000000000000'
);

```

Msg 50000, Level 16, State 1, Procedure tr\_BlockInvalidReferenceAccount, Line 23 [Batch Start Line 203]  
 Akun tujuan (reference account) tidak valid atau tidak ditemukan.  
 Msg 3609, Level 16, State 1, Line 204

v. 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.

- Query:

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

    -- Trigger hanya aktif untuk transaksi withdrawal (id=3) atau transfer
    (id=2)
    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) -- 2: Transfer, 3: Withdrawal
        AND (a.balance - i.amount) < 100000
    )
    BEGIN
        RAISERROR('Saldo tidak mencukupi. Minimal saldo setelah transaksi
        harus Rp100.000.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika validasi lolos, izinkan insert
    INSERT INTO transactions (
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    )
    SELECT
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    FROM inserted;
END;

-- pengujian
BEGIN TRANSACTION;
INSERT INTO transactions (account_id, transaction_type_id, amount,
transaction_date, description)
VALUES ('9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 3, 9000, GETDATE(), 'Failed
Withdrawal');
COMMIT TRANSACTION;
```

- Screenshoot:

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

    -- Trigger hanya aktif untuk transaksi withdrawal (id=3) atau transfer (id=2)
    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) -- 2: Transfer, 3: Withdrawal
        AND (a.balance - i.amount) < 100000
    )
    BEGIN
        RAISERROR('Saldo tidak mencukupi. Minimal saldo setelah transaksi harus Rp100.000.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    -- Jika validasi lolos, izinkan insert
    INSERT INTO transactions (
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    )
    SELECT
        transaction_id, account_id, transaction_type_id, amount,
        transaction_date, description, reference_account
    FROM inserted;
END;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-05-22T15:34:26.1465003+07:00

```
-- Jika validasi lolos, izinkan insert
INSERT INTO transactions (
    transaction_id, account_id, transaction_type_id, amount,
    transaction_date, description, reference_account
)
SELECT
    transaction_id, account_id, transaction_type_id, amount,
    transaction_date, description, reference_account
FROM inserted;
END;
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-05-22T15:34:26.1465003+07:00

```
-- pengujian
BEGIN TRANSACTION;
INSERT INTO transactions (account_id, transaction_type_id, amount, transaction_date, description)
VALUES ('9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7', 3, 9000, GETDATE(), 'Failed Withdrawal');
COMMIT TRANSACTION;
```

100 %

Messages

Msg 50000, Level 16, State 1, Procedure tr\_EnsureMinimumBalance, Line 17 [Batch Start Line 263]  
Saldo tidak mencukupi. Minimal saldo setelah transaksi harus Rp100.000.  
Msg 3609, Level 16, State 1, Line 265