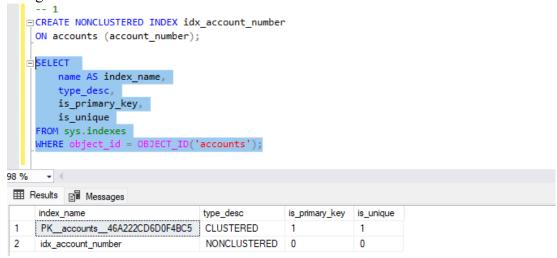
## Kelompok 05 Lab A2

| 1. | Bagus Athallah        | (24060123120002) |
|----|-----------------------|------------------|
| 2. | Akbar Maulana P       | (24060123140199) |
| 3. | Axel Natakesuma       | (24060123130058) |
| 4. | Agathan Khairy Bowo L | (24060123140129) |
| 5. | Ahmad Rafi Alhammam   | (24060123140138) |

## Praktikum 4

1. Query untuk membuat index pada tabel accounts untuk mempercepat pencarian. Pilih salah satu kolom pada table tersebut dan berikan alasan pemilihan kolom tersebut sebagai index!



Kolom account\_number dipilih karena merupakan kolom pencarian yang umum digunakan dalam transaksi perbankan. Selain itu, nilainya unik (mirip nomor rekening) dan Sering digunakan dalam WHERE clause saat mencari akun tertentu.

Sebelum

```
| Results | | Results | | | | Results | | | Results | | | Results | | | Results | Result
```

Sesudah

```
## Results | Messages |
SQL Server parse and compile time:
CFU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CFU time = 0 ms, elapsed time = 0 ms.

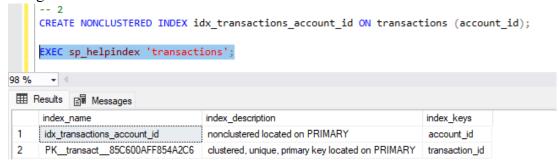
SQL Server Execution Times:
CFU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CFU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CFU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times:
CFU time = 0 ms, elapsed time = 0 ms.
```

2. Query untuk membuat index pada tabel transactions untuk mempercepat pencarian. Pilih salah satu kolom pada table tersebut dan berikan alasan pemilihan kolom tersebut sebagai index!



Index dibuat pada kolom account\_id dalam tabel transactions karena kolom ini merupakan foreign key yang sering digunakan dalam pencarian data transaksi berdasarkan akun tertentu. Penggunaan account\_id dalam klausa WHERE dan operasi JOIN dengan tabel accounts sangat umum dalam sistem perbankan, terutama saat menelusuri riwayat transaksi per akun. Dengan demikian, pembuatan index pada kolom ini dapat secara signifikan mempercepat proses pencarian dan meningkatkan kinerja query yang terkait.

• Sebelum

```
CREATE NONCLUSTERED INDEX idx_transactions_account_id ON transactions (account_id);
       SET STATISTICS TIME ON;
SET STATISTICS IO ON;
      SELECT *
      FROM transactions
WHERE account_id = '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7';

    ■ Results    ■ Messages

    SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 0 ms.

SQL Server parse and compile time:

CPU time = 0 ms, elapsed time = 0 ms.
    (4 rows affected)
Table "transactions". Scan count 1, logical reads 6, physical reads 0, page server read-ahead reads 0, page server read-ahead reads 0, lob logical reads
     SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms.
    Completion time: 2025-05-23T09:43:49.3487100+07:00
Sesudah
       -- 2
CREATE NONCLUSTERED INDEX idx_transactions_account_id ON transactions (account_id);
       SET STATISTICS TIME ON;
SET STATISTICS IO ON;
      SELECT FROM transactions

MHERE account_id = '9cd97ecb-58c9-4610-b4b1-d9f72bcae7f7';
 SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 0 ms.

SQL Server parse and compile time:

CPU time = 0 ms, elapsed time = 0 ms.
    (4 rows affected)
Table 'transactions'. Scan count 1, logical reads 6, physical reads 0, page server read-ahead reads 0, page server read-ahead reads 0, lob logical reads
      SQL Server Execution Times:
CPU time = 0 ms, elapsed time = 0 ms
      Completion time: 2025-05-23T09:45:19.5685143+07:00
```

Hasilnya terlihat sama karena query yang dijalankan terlalu ringan. Jika waktu < 1 ms, maka akan ditampilkan sebagai 0 ms.

3. Query untuk membuat composite index pada tabel accounts. Pilih lebih dari satu kolom pada tabel tersebut dan uji performa query sebelum dan setelah menggunakan index! (query pengujian berdasarkan kolom yang dipilih sebagai index)



Hasilnya terlihat sama karena query yang dijalankan terlalu ringan. Jika waktu < 1 ms, maka akan ditampilkan sebagai 0 ms.

4. Query untuk membuat cursor untuk membaca semua akun dari tabel accounts, lalu periksa balance masing-masing. Jika balance < 9.000, tampilkan pesan: "Saldo rendah untuk account id: [account id]".

```
-- 4
DECLARE @account_id CHAR(36);
DECLARE @balance DECIMAL(18, 2);
             DECLARE cur_accounts CURSOR FOR
             SELECT account_id, balance
             FROM accounts;
OPEN cur_accounts;
              FETCH NEXT FROM cur_accounts INTO @account_id, @balance;
              MHILE
                                                              = 0
                     IF @balance < 9000
                                PRINT 'Saldo rendah untuk account_id: ' + @account_id;
                        FETCH NEXT FROM cur_accounts INTO @account_id, @balance;
                 OSE cur_accounts;
             DEALLOCATE cur_accounts;
98 %
  Messages
       Messages
Saldo rendah untuk account_id: 0345E97A-8DEF-42FE-AB36-5124366D027A
Saldo rendah untuk account_id: 07a36a4b-7943-4634-b17a-7c7eb4407cd3
Saldo rendah untuk account_id: 0937445F-d5eb-4ef9-887f-4173fe662dd4
Saldo rendah untuk account_id: 1d37bcel-15c94-49e4-622a-6ad56b9afb0
Saldo rendah untuk account_id: 1d37bcel-15c94-49e4-622a-6ad56b9afb0
Saldo rendah untuk account_id: 316c43bf-9511-4700-alc3-f3e747b72f20
Saldo rendah untuk account_id: 31e643bf-9511-4030-bd13-81e47a4b8cac
Saldo rendah untuk account_id: 34e9be76-4674-4d0f-b7c0-a39e43548747
Saldo rendah untuk account_id: 36f3d280-2886-4e51-adcd-dfc644e59d80
Saldo rendah untuk account_id: 36f3d280-2886-4e51-adcd-dfc644e59d80
       Saldo rendah untuk account_id: 3B6B0001-52C3-4FE1-B930-70DAA90C9F31
Saldo rendah untuk account_id: 418a133e-5e1f-4377-8d48-63de24b5b08e
Saldo rendah untuk account_id: 418fedb1-da8f-4c67-8073-4a147e2f5ded
         Saldo rendah untuk account_id: 441aa9d2-bdf2-4b65-a79f-c18e66de0e17
```

5. Query untuk membuat cursor untuk membaca semua pelanggan (customers) dan gabungkan first\_name dan last\_name untuk ditampilkan dengan format: "Customer: [Nama Lengkap]".

```
DECLARE @first_name VARCHAR(50);
DECLARE @last_name VARCHAR(50);
DECLARE @full_name VARCHAR(101);
       SELECT first_name, last_name
        FROM customers;
        OPEN CustomerCursor;
        FETCH NEXT FROM CustomerCursor INTO @first_name, @last_name;
        WHILE
                                      = 0
             SET @full_name = C
                                                (@first_name, ' ', @last_name);
             PRINT 'Customer: ' + @full_name;
             FETCH NEXT FROM CustomerCursor INTO @first_name, @last_name;
        LOSE CustomerCursor:
       DEALLOCATE CustomerCursor;
98 %

    Messages

    Customer: Sergeant Maith
Customer: Shaine Matic
    Customer: Buddie Sandison
Customer: Stephanus Frear
Customer: Corette Vicent
    Customer: Devon McQuode
    Customer: Jacquenette Hilldrup
Customer: Bobette Shambrooke
Customer: Donnajean Amerighi
    Customer: Barbabra Swires
Customer: Sandro Alten
    Customer: Nester Niccolls
Customer: Rebekah Conkie
```