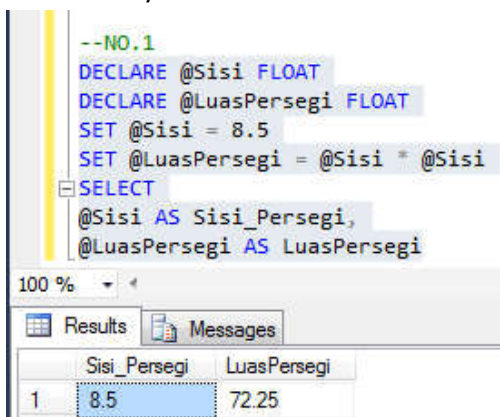


Nama : Arya Bima Mulya Putra
Kelas :2KA08
NPM :10123188
ACTIVITY

PERTEMUAN 7 BACKUP AND RECOVERY

Universitas X baru saja mengembangkan sistem informasi kampus yang akan digunakan untuk mengelola data mahasiswa, dosen, dan mata kuliah. Sistem ini menggunakan SQL Server sebagai database manajemennya dan universitas ingin memastikan data tersebut aman dengan membuat backup database secara teratur. Sebagai seorang administrator database, anda memiliki beberapa tugas sebagai berikut :

1. Buatlah script T-SQL untuk menghitung luas persegi dengan ketentuan berikut:
 - a. Deklarasikan variable sisi bertipe float
 - b. Berikan nilai untuk sisi adalah 8,5
 - c. Rumus luas persegi = $S \times S$
 - d. Simpan hasilnya dalam variable luasPersegi dengan tipe data float
 - e. Tampilkan hasilnya



```
--NO.1
DECLARE @Sisi FLOAT
DECLARE @LuasPersegi FLOAT
SET @Sisi = 8.5
SET @LuasPersegi = @Sisi * @Sisi
SELECT
    @Sisi AS Sisi_Persegi,
    @LuasPersegi AS LuasPersegi
```

	Sisi_Persegi	LuasPersegi
1	8.5	72.25

Jawab:

2. Buat stored procedure dengan nama TambahDataMahasiswa dengan ketentuan berikut ini:
 - a. Memiliki parameter @npm (Varchar), @nama (Varchar), @kelas (Varchar) dan @jurusan (varchar)
 - b. Masukkan data kedalam table Mahasiswa (buat table mahasiswa terlebih dahulu dengan ketentuan fieldnya adalah:
 - ❑ NPM varchar (8) primary key
 - ❑ NAMA varchar(50)
 - ❑ KELAS varchar(5)
 - ❑ JURUSAN varchar(30)

```
--NO.2
CREATE TABLE MAHASISWA(
  NPM VARCHAR(8) PRIMARY KEY,
  NAMA VARCHAR(50),
  KELAS VARCHAR(5),
  JURUSAN VARCHAR(30))
```

100 %

Messages

Command(s) completed successfully.

c. Setelah berhasil membuat table dan stored procedure, masukan data berikut kedalam table Mahasiswa melalui stored procedure

- ☐ NPM : npm masing-masing praktikan
- ☐ NAMA : nama masing-masing praktikan
- ☐ KELAS : kelas masing-masing praktikan
- ☐ JURUSAN : jurusan masing-masing praktikan

```
CREATE PROCEDURE TambahDataMahasiswa
  @NPM VARCHAR(8),
  @NAMA VARCHAR(50),
  @KELAS VARCHAR(5),
  @JURUSAN VARCHAR(30)
AS
BEGIN
  INSERT INTO MAHASISWA(NPM, NAMA, KELAS, JURUSAN)
  VALUES (@NPM, @NAMA, @KELAS, @JURUSAN)
END
```

100 %

Messages

Command(s) completed successfully.

d. Buktikan bahwa data berhasil dimasukkan kedalam table Mahasiswa

```
EXEC TambahDataMahasiswa
  @NPM = '10123188',
  @NAMA = 'Arya Bima',
  @KELAS = '2KA08',
  @JURUSAN = 'Sistem Informasi'

SELECT * FROM MAHASISWA
```

100 %

Results Messages

	NPM	NAMA	KELAS	JURUSAN
1	10123188	Arya Bima	2KA08	Sistem Informasi

3. Dengan menggunakan table Mahasiswa yang sudah dibuat pada nomor 2, buatlah clustered index untuk table Mahasiswa pada kolom NAMA.

a. Apa yang hasil yang didapat? Apakah error atau berhasil?

b. Jika error berikan penjelasannya dan bagaimana caranya agar clustered index dapat berhasil terbuat?

Jawab:

```
--NO.3
CREATE CLUSTERED INDEX idx_nama
ON MAHASISWA(NAMA)

EXEC sp_helpindex 'MAHASISWA'
```

100 %

Messages

Msg 1902, Level 16, State 3, Line 41
Cannot create more than one clustered index on table 'MAHASISWA'. Drop the existing clustered index 'PK__MAHASISW_

karena Clustered index akan terbuat secara otomatis dalam table, sehingga ketika membuat clustered index tersendiri akan error.

```
EXEC sp_helpindex 'MAHASISWA'
```

100 %

Results Messages

	index_name	index_description	index_keys
1	PK__MAHASISW__C7DE9A2F1F8060EC	clustered, unique, primary key located on PRIMARY	NPM

```
SELECT
CONSTRAINT_NAME,
CONSTRAINT_TYPE
FROM
INFORMATION_SCHEMA.TABLE_CONSTRAINTS
WHERE
TABLE_NAME = 'MAHASISWA'
```

100 %

Results Messages

	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	PK__MAHASISW__C7DE9A2F1F8060EC	PRIMARY KEY

```
ALTER TABLE MAHASISWA
DROP CONSTRAINT PK__MAHASISW__C7DE9A2F1F8060EC
```

100 %

Messages

Command(s) completed successfully.

```
CREATE CLUSTERED INDEX idx_nama
ON MAHASISWA(NAMA)

EXEC sp_helpindex 'MAHASISWA'
```

100 %

Results Messages

	index_name	index_description	index_keys
1	idx_nama	clustered located on PRIMARY	NAMA

4. Dengan menggunakan table Mahasiswa yang sudah dibuat pada nomor 2, buatlah nonclustered index untuk table Mahasiswa pada kolom JURUSAN.

a. Buktikan bahwa non-clustered index berhasil dibuat

Jawab:

```
--NO.4
create nonclustered index idx_jurusan
ON MAHASISWA(JURUSAN)

exec sp_helpindex 'MAHASISWA'
```

100 %

Results Messages

	index_name	index_description	index_keys
1	idx_jurusan	nonclustered located on PRIMARY	JURUSAN
2	idx_nama	clustered located on PRIMARY	NAMA