Muhammad azhar utama 223040077

I. Query dasar

1. Tuliskan query tabel-tabel dari gambar erd kemudian tampilkan hasilnya di diagram microsoft sql server.

Untuk setiap primary key dan foreign key gunakan perintah constraint.

Dan untuk setiap foreign key gunakan perintah on delete no cascade on update cascade.

li. Manipulasi data

- 2. Manipulasi data menggunakan perintah alter, update, insert dan delete.
 - a) tambahkan constraint unique untuk nim pada tabel mahasiswa dan nip pada tabel dosen.
 - b) tambahkan kolom email ber-constraint unique untuk tabel mhasiswa dan dosesn.
 - c) berikan constraint check untuk tabel matakuliah agar nilai sks lebih dari sama dengan 0 dan pada tabel nilai agar tidak negatif dan tidak melebihi 100.
 - d) masukkan data dari file data.txt menggunakan perintah insert into.
 - e) pada tabel mahasiswa tampilkan semua informasi kolom (hindari penggunaan wildcard(*)). Tampilkan semua data dimana alamat kota bernilai "null" tampilkan hasil querynya.
 - f) update alamat tabel mahasiswa yang bernilai "null" menjadi "bandung".
 - g) hapus data dosen yang memiliki nip "04304004"

lii. Join dan set operator

- 3. Menggunakan fungsi set operator dan join untuk menampilkan/menggabungkan data dari 2 atau lebih tabel yang berbeda.
 - a) tampilkan semua nama mahasiswa dan dosen menggunakan set operator
 - b) tampilkan kota yang ada pada dosen dan mahasiswa
 - c) tampilkan kota mahasiswa yang ada tidak ada pada dosen
 - d) tampilkan nama, nim dan nama mata kuliah untuk mahasiswa yang mengambil mata kuliah tertentu (misalnya id matakuliah = 1).
 - e) tampilkan nama dan nim mahasiswa yang tidak mengambil mata kuliah.

Iv. Fungsi agregat dan window functions

4.

- a) hitung jumlah mahasiswa yang memiliki nilai lebih dari 80 (fungsi agregat).
- b) tampilkan nama mahasiswa dan rata-rata nilai (fungsi agregat).
- c) tampilkan nama, nim, id mahasiwa, nilai dan id mata kuliah, kemudian lakukan perintah window functions untuk memberikan nilai rata-rata berdasarkan id matakuliah serta peringkat dengan dan tanpa loncatan berdasarkan id matakuliah yang diurutkan dari nilai terbesar ke terkecil.
- d) buatlah urutan menggunakan id mata kuliah dan buatkanlah kelompok 3 berdasarkan nilai untuk matakuliah basis data.

V. Output statement

5.

- a) masukkan minimal 1 data mahasiswa baru dan munculkanlah data yang baru tersebut
- b) hapus minimal 1 data mahasiswa dan munculkanlah data yang terhapus tadi

1. Tuliskan query tabel-tabel dari gambar erd kemudian tampilkan hasilnya di diagram microsoft sgl server.

Untuk setiap primary key dan foreign key gunakan perintah constraint.

Dan untuk setiap foreign key gunakan perintah on delete no cascade on update cascade.

```
USE PRA_UTS_223040077
---I. Query Dasar
----1. Tuliskan Query tabel-tabel dari gambar ERD kemudian Tampilkan hasilnya di diagram Microsoft SQL server.----
----Untuk setiap PRIMARY KEY dan FOREIGN KEY gunakan perintah CONSTRAINT.----
----Dan <u>untuk setiap</u> FOREIGN KEY <u>gunakan perintah</u> ON DELETE NO CASCADE ON UPDATE CASCADE.-----
CREATE TABLE mahasiswa
   ID_Mahasiswa int
               varchar(50) NOT NULL,
   CONSTRAINT PK_MahasiswaID PRIMARY KEY (ID_Mahasiswa)
CREATE TABLE matakuliah
   nama_matakuliah varchar(50) NOT NULL,
   CONSTRAINT PK_MatakuliahID PRIMARY KEY (ID_matakuliah)
CREATE TABLE nilai
   MahasiswaID_Mahasiswa
   MatakuliahID_Matakuliah int,
   FOREIGN KEY (MahasiswaID_Mahasiswa) REFERENCES mahasiswa (ID_Mahasiswa) ON DELETE CASCADE ON UPDATE CASCADE,
   FOREIGN KEY (MatakuliahID_Matakuliah) REFERENCES matakuliah (ID_matakuliah) ON DELETE CASCADE ON UPDATE CASCADE,
```

```
DROP
COLUMN ruangan;

ALTER TABLE perkuliahan
ADD ruangan varchar(50);
```

• Mengubah tipe data ruangan karena data yang diberikan bertipe string sedangkan dalam erd data ruangan bertipe integer

- 2. Manipulasi data menggunakan perintah alter, update, insert dan delete.
- A) tambahkan constraint unique untuk nim pada tabel mahasiswa dan nip pada tabel dosen.

```
---A .Tambahkan Constraint UNIQUE untuk NIM pada tabel Mahasiswa dan NIP pada Tabel Dosen.---

USE PRA_UTS_223040077

GO

PALTER TABLE mahasiswa

ADD CONSTRAINT UQ_Mahasiswa UNIQUE (nim);

PALTER TABLE dosen

ADD CONSTRAINT UQ_Dosen UNIQUE (NIP);
```

B) tambahkan kolom email ber-constraint unique untuk tabel mhasiswa dan dosesn.

```
-- b) Tambahkan kolom Email ber-Constraint UNIQUE untuk tabel Mhasiswa dan Dosesn.

PALTER TABLE mahasiswa

PALTER TABLE mahasiswa

ADD constraint UC_Mahasiswa UNIQUE (email);

PALTER TABLE dosen

ADD email varchar(50) not null;

PALTER TABLE dosen

ADD constraint UC_dosen UNIQUE (email);
```

C) berikan constraint check untuk tabel matakuliah agar nilai sks lebih dari sama dengan 0 dan pada tabel nilai agar tidak negatif dan tidak melebihi 100.

```
ALTER TABLE matakuliah

ADD CONSTRAINT CK_SKS CHECK (SKS >= 0);

ALTER TABLE nilai

ADD CONSTRAINT CK_Nilai CHECK (Nilai >= 0 AND Nilai <= 100);
```

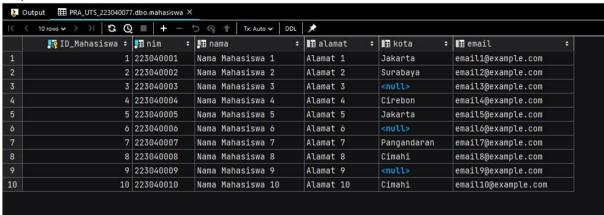
D) masukkan data dari file data.txt menggunakan perintah insert into.

Mahasiswa

```
--mahasiswa
FINSERT INTO mahasiswa (ID_Mahasiswa, nim, nama, alamat, kota, email)
FVALUES (1, '223040001', 'Nama Mahasiswa 1', 'Alamat 1', 'Jakarta', 'email1@example.com'),
(2, '223040002', 'Nama Mahasiswa 2', 'Alamat 2', 'Surabaya', 'email2@example.com'),
(3, '223040003', 'Nama Mahasiswa 3', 'Alamat 3', NULL, 'email3@example.com'),
(4, '223040004', 'Nama Mahasiswa 4', 'Alamat 4', 'Cirebon', 'email4@example.com'),
(5, '223040005', 'Nama Mahasiswa 5', 'Alamat 5', 'Jakarta', 'email5@example.com'),
(6, '223040006', 'Nama Mahasiswa 6', 'Alamat 6', NULL, 'email6@example.com'),
(7, '223040007', 'Nama Mahasiswa 7', 'Alamat 7', 'Pangandaran', 'email7@example.com'),
(8, '223040008', 'Nama Mahasiswa 8', 'Alamat 8', 'Cimahi', 'email8@example.com'),
(9, '223040009', 'Nama Mahasiswa 9', 'Alamat 9', NULL, 'email9@example.com'),
(10, '223040010', 'Nama Mahasiswa 10', 'Alamat 10', 'Cimahi', 'email10@example.com');

SELECT ID_Mahasiswa, nim, nama, alamat, kota, email FROM mahasiswa;
```

Output



Dosen

```
--dosen

JINSERT INTO dosen (ID_dosen, NIP, nama, alamat, kota, email)

VALUES (1, '040304001', 'Nama Dosen 1', 'Alamat Dosen 1', 'Bandung', 'dosen1@example.com'),

(2, '040304002', 'Nama Dosen 2', 'Alamat Dosen 2', 'Jakarta', 'dosen2@example.com'),

(3, '040304003', 'Nama Dosen 3', 'Alamat Dosen 3', 'Cimahi', 'dosen3@example.com'),

(4, '040304004', 'Nama Dosen 4', 'Alamat Dosen 4', 'Bandung', 'dosen4@example.com');
```



Matakuliah

```
--matakuliah

JINSERT INTO matakuliah (ID_matakuliah, nama_matakuliah, SKS)

VALUES (1, 'Matematika Informatika', 3),

(2, 'Basis Data', 3),

(3, 'Infrastruktur Jaringan', 2);
```

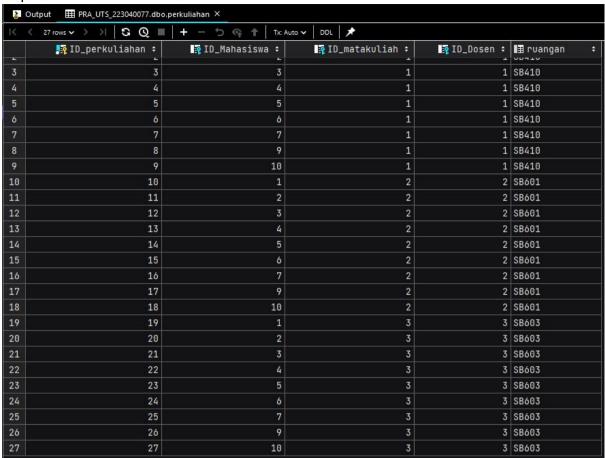
Output

K	<	3 rows ✔	>	>1	3	Q		+	-	D	Q	1	Tx: Auto ✔	DDL	*
		🃭 ID	_ma	taku	liah	1 ‡	#	nama.	mat	aku	lia	h		題	SKS ÷
1			1 Matematika Informatika							3					
2						2	Bas	is D	ata						3
3						3	Inf	rast	ruk	tur	Jar	ing	an		2

Nilai

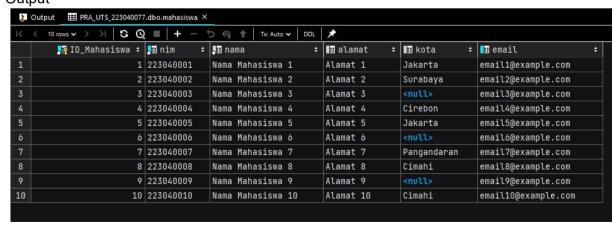
<	< 30 rows ∨ >)	S Q ■	+ - 5 № ↑ Tx: Auto > DDL 🖈	
	⊞ nilai ‡	I grade ‡	MahasiswaID_Mahasiswa ÷	🌠 MatakuliahID_Matakuliah 🕏
1	85	A	1	1
2	55	CD	1	2
3	80	AB	1	3
4	88	A	2	1
5	78	В	2	2
6	78	В	2	3
7	92	A	3	1
8	90	A	3	2
9	78	В	3	3
LO	68	С	4	1
1	68	С	4	-
2	78	В	4	
3	85	A	5	
4	90	A	5	
.5	75	В	5	:
.6	68	С	6	1
17	88	A	ó	:
.8	80	AB	6	;
9	85	A	7	1
0	68	С	7	
1	75	В	7	:
2	θ	T	8	
3	0	T	8	:
24	θ	T	8	
5	55	CD	9	
6	68	С	9	
7	80	AB	9	
8	50	D	10	:
9	50		10	
30	50	D	10	

Perkuliahan



E) pada tabel mahasiswa tampilkan semua informasi kolom (hindari penggunaan wildcard(*)). Tampilkan semua data dimana alamat kota bernilai "null" tampilkan hasil querynya.

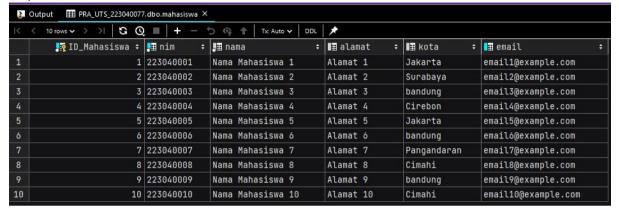
```
SELECT ID_Mahasiswa, nim, nama, alamat, kota, email
FROM mahasiswa;
```



F) update alamat tabel mahasiswa yang bernilai "null" menjadi "bandung".

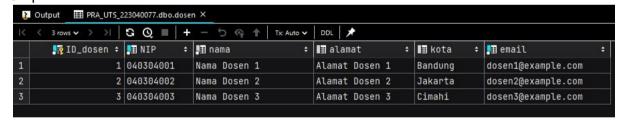
```
--f) UPDATE alamat tabel Mahasiswa yang bernilai "null" menjadi "Bandung".
JUPDATE mahasiswa
SET kota = 'Bandung'
ЭWHERE kota IS NULL;
```

Output



g) hapus data dosen yang memiliki nip "04304004".





3. Join dan set operator

A) tampilkan semua nama mahasiswa dan dosen menggunakan set operator B)

```
---a) Tampilkan semua nama Mahasiswa dan Dosen menggunakan SET OPERATOR
DSELECT Nama FROM Mahasiswa
UNION ALL
DSELECT Nama FROM Dosen;
```

Output



b) tampilkan kota yang ada pada dosen dan mahasiswa

• Tanpa duplikasi

```
--- TANPA DUPLIKASI

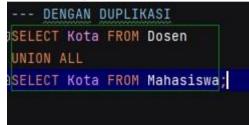
SELECT DISTINCT Kota FROM Dosen

UNION

SELECT DISTINCT Kota FROM Mahasiswa;
```

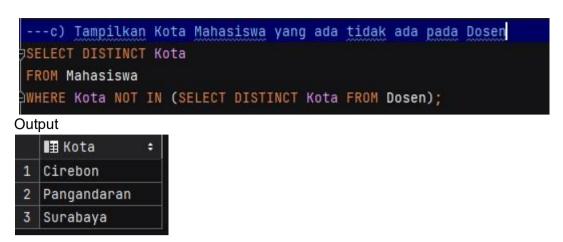


• Dengan duplikasi





C) tampilkan kota mahasiswa yang ada tidak ada pada dosen



D) tampilkan nama, nim dan nama mata kuliah untuk mahasiswa yang mengambil mata kuliah tertentu (misalnya id_matakuliah = 1).

```
SELECT mahasiswa.Nama, mahasiswa.NIM, matakuliah.nama_matakuliah

FROM mahasiswa

JOIN nilai ON mahasiswa.ID_mahasiswa = nilai.MahasiswaID_Mahasiswa

JOIN mataKuliah ON nilai.MatakuliahID_Matakuliah = MataKuliah.ID_Matakuliah

WHERE matakuliah.ID_Matakuliah = 1;
```

	III Nama ≎	■■ NIM ÷	II nama_matakuliah		
1	Nama Mahasiswa 1	223040001	Matematika Informatika		
2	Nama Mahasiswa 2	223040002	Matematika Informatika		
3	Nama Mahasiswa 3	223040003	Matematika Informatika		
4	Nama Mahasiswa 4	223040004	Matematika Informatika		
5	Nama Mahasiswa 5	223040005	Matematika Informatika		
6	Nama Mahasiswa 6	223040006	Matematika Informatika		
7	Nama Mahasiswa 7	223040007	Matematika Informatika		
8	Nama Mahasiswa 8	223040008	Matematika Informatika		
9	Nama Mahasiswa 9	223040009	Matematika Informatika		

E) tampilkan nama dan nim mahasiswa yang tidak mengambil mata kuliah

```
SELECT Nama, NIM
FROM Mahasiswa
WHERE NOT EXISTS (
SELECT 1
FROM Nilai
WHERE Nilai.MahasiswaID_Mahasiswa = Mahasiswa.ID_Mahasiswa

Output
```



4.fungsi agregat dan window functions

A) hitung jumlah mahasiswa yang memiliki nilai lebih dari 80 (fungsi agregat).

```
-- a) Hitung jumlah mahasiswa yang memiliki nilai lebih dari 80 (Fungsi Agregat).

SELECT COUNT(*) AS JumlahMahasiswa
FROM Mahasiswa
JOIN Nilai ON Mahasiswa.ID_Mahasiswa = Nilai.MahasiswaID_Mahasiswa
WHERE Nilai.nilai > 80;
```

Output



B) tampilkan nama mahasiswa dan rata-rata nilai (fungsi agregat).

```
-- b) Tampilkan Nama Mahasiswa dan Rata-rata Nilai (Fungsi Agregat).

SELECT Mahasiswa.Nama, AVG(Nilai.nilai) AS RataRataNilai

FROM Mahasiswa

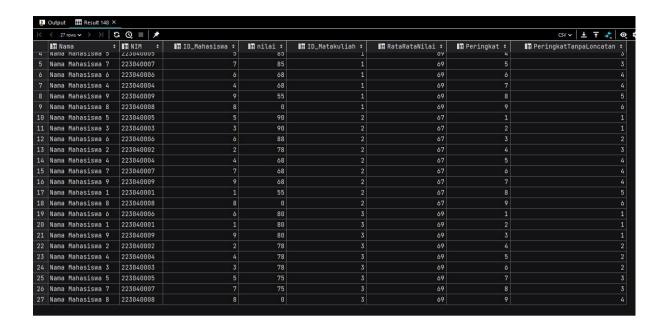
JOIN Nilai ON Mahasiswa.ID_Mahasiswa = Nilai.MahasiswaID_Mahasiswa

GROUP BY Mahasiswa.ID_Mahasiswa, Mahasiswa.Nama;
```

	■ Nama		■ RataRataNilai ÷
1	Nama Mahasiswa 1		73
2	Nama Mahasiswa 2		81
3	Nama Mahasiswa 3		86
4	Nama Mahasiswa 4		71
5	Nama Mahasiswa 5	(83
6	Nama Mahasiswa 6		78
7	Nama Mahasiswa 7		76
8	Nama Mahasiswa 8		0
9	Nama Mahasiswa 9		67

C) tampilkan nama, nim, id mahasiwa, nilai dan id mata kuliah, kemudian lakukan perintah window functions untuk memberikan nilai rata-rata berdasarkan id matakuliah serta peringkat dengan dan tanpa loncatan berdasarkan id matakuliah yang diurutkan dari nilai terbesar ke terkecil.

```
Mahasiswa.Nama,
Mahasiswa.NIM,
Nilai.MahasiswaID_Mahasiswa AS ID_Mahasiswa,
Nilai.nilai,
Nilai.MahasiswaID_Matakuliah AS ID_Matakuliah,
AVG(Nilai.nilai) OVER (PARTITION BY Nilai.MatakuliahID_Matakuliah) AS RataRataNilai,
ROW_NUMBER() OVER (PARTITION BY Nilai.MatakuliahID_Matakuliah ORDER BY Nilai.nilai DESC) AS Peringkat,
DENSE_RANK() OVER (PARTITION BY Nilai.MatakuliahID_Matakuliah ORDER BY Nilai.nilai DESC) AS PeringkatTanpaloncat
FROM
Mahasiswa
JOIN
Nilai ON Mahasiswa.ID_Mahasiswa = Nilai.MahasiswaID_Mahasiswa
ORDER BY
Nilai.MatakuliahID_Matakuliah, Nilai.nilai DESC;
```



D) buatlah urutan menggunakan id mata kuliah dan buatkanlah kelompok 3 berdasarkan nilai untuk matakuliah basis data.

```
Mahasiswa.Nama,
Mahasiswa.NIM,
Nitai.MahasiswaID_Mahasiswa AS ID_Mahasiswa,
Nitai.MahasiswaID_Mahasiswa AS ID_Mahasiswa,
Nitai.nitai,
Nitai.MatakuliahID_Matakuliah AS ID_Matakuliah,
AVG(Nitai.nitai) OVER (PARTITION BY Nitai.MatakuliahID_Matakuliah) AS RataRataNitai,
ROW_NUMBER() OVER (PARTITION BY Nitai.MatakuliahID_Matakuliah ORDER BY Nitai.nitai DESC) AS Peringkat,
DENSE_RANK() OVER (PARTITION BY Nitai.MatakuliahID_Matakuliah ORDER BY Nitai.nitai DESC) AS PeringkatTanpaLoncatan
FROM
Mahasiswa
JOIN
Nitai ON Mahasiswa.ID_Mahasiswa = Nitai.MahasiswaID_Mahasiswa
WHERE
Nitai.MatakuliahID_Matakuliah = 2 -- ID Mata Kuliah untuk Basis Data
ORDER BY

■ Nitai.MatakuliahID_Matakuliah, Nitai.nitai DESC;
```

I ≣ Nama	÷	≣NIM ÷	II ID_Mahasiswa ▲ 1	I⊞ nilai ÷	II ID_Matakuliah ÷	I≣ RataRataNilai ≎	⊞ Peringkat ≎	I⊞ PeringkatTanpaLoncatan ≎
1 Nama Mahasiswa 1		23040001	1	55	2	67	8	5
2 Nama Mahasiswa 2	2	23040002	2	78	2	67	4	3
3 Nama Mahasiswa 3	2	23040003	3	90	2	67	1	1
4 Nama Mahasiswa 4	2	23040004	4	68	2	67	5	4
5 Nama Mahasiswa 5	2	23040005	5	90	2	67	2	1
6 Nama Mahasiswa 6	2	23040006	6	88	2	67	3	2
7 Nama Mahasiswa 7	2	23040007	7	68	2	67	6	4
8 Nama Mahasiswa 8	2	23040008	8	0	2	67	9	6
9 Nama Mahasiswa 9	2	23040009	9	68	2	67	7	4
The state of the s	1/4		1					

5.output statement

A) masukkan minimal 1 data mahasiswa baru dan munculkanlah data yang baru tersebut

```
--- a) Masukkan minimal 1 data Mahasiswa baru dan munculkanlah data yang baru tersebut
JINSERT INTO Mahasiswa (ID_Mahasiswa, nim, nama, alamat, kota, email)

VALUES (11, '223040077', 'Muhammad Azhar Utama', 'jl.galuga', 'bogor', 'azharutama.com'),

(12, '223040085', 'Reza Ageng', 'jl.Hegarmana', 'jawa', 'reza@gmail.com'),

(13, '223040086', 'Muhammad Azhar Lutfiadi', 'jl.gatau', 'subang', 'azharLutfiafi@gmail.com');

SELECT ID_Mahasiswa, nim, nama, alamat, kota, email FROM Mahasiswa;
```

		∰ nim ÷	,⊞ nama ÷	∎alamat ÷	∎≣ kota ‡	∎ email ÷
1	1	223040001	Nama Mahasiswa 1	Alamat 1	Jakarta	email1@example.com
2	2	223040002	Nama Mahasiswa 2	Alamat 2	Surabaya	email2@example.com
3	3	223040003	Nama Mahasiswa 3	Alamat 3	bandung	email3@example.com
4	4	223040004	Nama Mahasiswa 4	Alamat 4	Cirebon	email4@example.com
5	5	223040005	Nama Mahasiswa 5	Alamat 5	Jakarta	email5@example.com
6	ó	223040006	Nama Mahasiswa 6	Alamat 6	bandung	emailó@example.com
7	7	223040007	Nama Mahasiswa 7	Alamat 7	Pangandaran	email7@example.com
8	8	223040008	Nama Mahasiswa 8	Alamat 8	Cimahi	email8@example.com
9	9	223040009	Nama Mahasiswa 9	Alamat 9	bandung	email9@example.com
10	10	223040010	Nama Mahasiswa 10	Alamat 10	Cimahi	email10@example.com
11	11	223040077	Muhammad Azhar Utama	jl.galuga	bogor	azharutama.com
12	12	223040085	Reza Ageng	jl.Hegarmana	jawa	reza@gmail.com
13	13	223040086	Muhammad Azhar Lutfiadi	jl.gatau	subang	azharLutfiafi@gmail.com

B) hapus minimal 1 data mahasiswa dan munculkanlah data yang terhapus tadi

```
DoLETE FROM mahasiswa
WHERE ID_Mahasiswa = 11;
```

