

Muhammad azhar utama  
223040077  
B

#### 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 mahasiswa dan dosen.
- 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 mahasiswa, 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 buatlah 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 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.

```
USE PRA_UTS_223040077
GO

----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.-----

CREATE TABLE mahasiswa
(
    ID_Mahasiswa int NOT NULL,
    nim varchar(50) NOT NULL,
    nama varchar(50) NOT NULL,
    alamat varchar(50) NULL,
    kota varchar(50) NULL,

    CONSTRAINT PK_MahasiswaID PRIMARY KEY (ID_Mahasiswa)
);

CREATE TABLE matakuliah
(
    ID_matakuliah int NOT NULL,
    nama_matakuliah varchar(50) NOT NULL,
    SKS int NOT NULL,

    CONSTRAINT PK_MatakuliahID PRIMARY KEY (ID_matakuliah)
);

CREATE TABLE nilai
(
    nilai int,
    grade char(2),
    MahasiswaID_Mahasiswa int,
    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,
);
```

```

CREATE TABLE perkuliahan
(
    ID_perkuliahan int NOT NULL,
    ruangan        int NULL,
    ID_Mahasiswa   int,
    ID_matakuliah   int,
    ID_Dosen        int,
    FOREIGN KEY (ID_Mahasiswa) REFERENCES mahasiswa (ID_Mahasiswa) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (ID_matakuliah) REFERENCES matakuliah (ID_matakuliah) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (ID_Dosen) REFERENCES dosen (ID_dosen) ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT PK_ID_PERKULIAHAN PRIMARY KEY (ID_perkuliahan)
);

CREATE TABLE dosen
(
    ID_dosen int NOT NULL,
    NIP      varchar(10) NOT NULL,
    nama     varchar(50) NOT NULL,
    alamat   varchar(50) NULL,
    kota     varchar(50) NULL
    CONSTRAINT PK_ID_dosen PRIMARY KEY (ID_dosen)
);

```

```

ALTER TABLE perkuliahan
    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  
  
ALTER TABLE mahasiswa  
ADD CONSTRAINT UQ_Mahasiswa UNIQUE (nim);  
  
ALTER TABLE dosen  
ADD CONSTRAINT UQ_Dosen UNIQUE (NIP);
```

B) tambahkan kolom email ber-constraint unique untuk tabel mahasiswa dan dosen.

```
-- b) Tambahkan kolom Email ber-Constraint UNIQUE untuk tabel Mahasiswa dan Dosen.  
  
ALTER TABLE mahasiswa  
ADD email varchar(50) not null;  
  
ALTER TABLE mahasiswa  
ADD CONSTRAINT UC_Mahasiswa UNIQUE (email);  
  
ALTER TABLE dosen  
ADD email varchar(50) not null;  
  
ALTER 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
INSERT INTO mahasiswa (ID_Mahasiswa, nim, nama, alamat, kota, email)
VALUES (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

ID_Mahasiswa	nim	nama	alamat	kota	email
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

## Dosen

```
--dosen
INSERT 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');
```

## Output

ID_dosen	NIP	nama	alamat	kota	email
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
INSERT INTO matakuliah (ID_matakuliah, nama_matakuliah, SKS)
VALUES (1, 'Matematika Informatika', 3),
       (2, 'Basis Data', 3),
       (3, 'Infrastruktur Jaringan', 2);
```

## Output

Output PRA_UTC_223040077.dbo.matakuliah X			
3 rows			
	ID_matakuliah	nama_matakuliah	SKS
1	1	Matematika Informatika	3
2	2	Basis Data	3
3	3	Infrastruktur Jaringan	2

## Nilai

```
--nilai
INSERT INTO nilai (MahasiswaID_Mahasiswa, MatakuliahID_Matakuliah, nilai, grade)
VALUES (1, 1, 85, 'A'),
       (1, 2, 55, 'CD'),
       (1, 3, 80, 'AB'),
       (2, 1, 88, 'A'),
       (2, 2, 78, 'B'),
       (2, 3, 78, 'B'),
       (3, 1, 92, 'A'),
       (3, 2, 90, 'A'),
       (3, 3, 78, 'B'),
       (4, 1, 68, 'C'),
       (4, 2, 68, 'C'),
       (4, 3, 78, 'B'),
       (5, 1, 85, 'A'),
       (5, 2, 90, 'A'),
       (5, 3, 75, 'B'),
       (6, 1, 68, 'C'),
       (6, 2, 88, 'A'),
       (6, 3, 80, 'AB'),
       (7, 1, 85, 'A'),
       (7, 2, 68, 'C'),
       (7, 3, 75, 'B'),
       (8, 1, 0, 'T'),
       (8, 2, 0, 'T'),
       (8, 3, 0, 'T'),
       (9, 1, 55, 'CD'),
       (9, 2, 68, 'C'),
       (9, 3, 80, 'AB'),
       (10, 1, 50, 'D'),
       (10, 2, 50, 'D'),
       (10, 3, 50, 'D');
```

## Output

	nilai	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	B	2	2
6	78	B	2	3
7	92	A	3	1
8	90	A	3	2
9	78	B	3	3
10	68	C	4	1
11	68	C	4	2
12	78	B	4	3
13	85	A	5	1
14	90	A	5	2
15	75	B	5	3
16	68	C	6	1
17	88	A	6	2
18	80	AB	6	3
19	85	A	7	1
20	68	C	7	2
21	75	B	7	3
22	0	T	8	1
23	0	T	8	2
24	0	T	8	3
25	55	CD	9	1
26	68	C	9	2
27	80	AB	9	3
28	50	D	10	1
29	50	D	10	2
30	50	D	10	3

## Perkuliahan

```
INSERT INTO perkuliahan(ID_perkuliahan, ID_Mahasiswa, ID_matakuliah, ID_Dosen, ruangan)
VALUES (1, 1, 1, 1, 'SB410'),
(2, 2, 1, 1, 'SB410'),
(3, 3, 1, 1, 'SB410'),
(4, 4, 1, 1, 'SB410'),
(5, 5, 1, 1, 'SB410'),
(6, 6, 1, 1, 'SB410'),
(7, 7, 1, 1, 'SB410'),
(8, 9, 1, 1, 'SB410'),
(9, 10, 1, 1, 'SB410'),
(10, 1, 2, 2, 'SB601'),
(11, 2, 2, 2, 'SB601'),
(12, 3, 2, 2, 'SB601'),
(13, 4, 2, 2, 'SB601'),
(14, 5, 2, 2, 'SB601'),
(15, 6, 2, 2, 'SB601'),
(16, 7, 2, 2, 'SB601'),
(17, 9, 2, 2, 'SB601'),
(18, 10, 2, 2, 'SB601'),
(19, 1, 3, 3, 'SB603'),
(20, 2, 3, 3, 'SB603'),
(21, 3, 3, 3, 'SB603'),
(22, 4, 3, 3, 'SB603'),
(23, 5, 3, 3, 'SB603'),
(24, 6, 3, 3, 'SB603'),
(25, 7, 3, 3, 'SB603'),
(26, 9, 3, 3, 'SB603'),
(27, 10, 3, 3, 'SB603');
```



## Output

ID_perkuliahan	ID_Mahasiswa	ID_matakuliah	ID_Dosen	ruangan
3	3	1	1	SB410
4	4	1	1	SB410
5	5	1	1	SB410
6	6	1	1	SB410
7	7	1	1	SB410
8	8	1	1	SB410
9	9	1	1	SB410
10	10	1	2	SB601
11	11	2	2	SB601
12	12	3	2	SB601
13	13	4	2	SB601
14	14	5	2	SB601
15	15	6	2	SB601
16	16	7	2	SB601
17	17	9	2	SB601
18	18	10	2	SB601
19	19	1	3	SB603
20	20	2	3	SB603
21	21	3	3	SB603
22	22	4	3	SB603
23	23	5	3	SB603
24	24	6	3	SB603
25	25	7	3	SB603
26	26	9	3	SB603
27	27	10	3	SB603

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;
```

## Output

ID_Mahasiswa	nim	nama	alamat	kota	email
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

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

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

Output

Output PRA_UTS_223040077.dbo.mahasiswa						
ID_Mahasiswa	nim	nama	alamat	kota	email	
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	bandung	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	bandung	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	bandung	email9@example.com	
10	223040010	Nama Mahasiswa 10	Alamat 10	Cimahi	email10@example.com	

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

```
DELETE FROM Dosen  
WHERE NIP = '04304004';
```

Output

Output PRA_UTS_223040077.dbo.dosen						
ID_dosen	NIP	nama	alamat	kota	email	
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	

### 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
SELECT Nama FROM Mahasiswa
UNION ALL
SELECT Nama FROM Dosen;
```

Output

Output	
PRA_UTS_223040077	
13 rows	
	Nama
1	Nama Mahasiswa 1
2	Nama Mahasiswa 2
3	Nama Mahasiswa 3
4	Nama Mahasiswa 4
5	Nama Mahasiswa 5
6	Nama Mahasiswa 6
7	Nama Mahasiswa 7
8	Nama Mahasiswa 8
9	Nama Mahasiswa 9
10	Nama Mahasiswa 10
11	Nama Dosen 1
12	Nama Dosen 2
13	Nama Dosen 3

b) tampilkan kota yang ada pada dosen dan mahasiswa

- Tanpa duplikasi

```
--- TANPA DUPLIKASI
SELECT DISTINCT Kota FROM Dosen
UNION
SELECT DISTINCT Kota FROM Mahasiswa;
```

## Output

Output		PRA_UTS	
		6 rows	
	Kota		
1	bandung		
2	Cimahi		
3	Cirebon		
4	Jakarta		
5	Pangandaran		
6	Surabaya		

- Dengan duplikasi

```
--- DENGAN DUPLIKASI
SELECT Kota FROM Dosen
UNION ALL
SELECT Kota FROM Mahasiswa;
```

## Output

Output		- DENGAN
13 rows		
	Kota	
1	Bandung	
2	Jakarta	
3	Cimahi	
4	Jakarta	
5	Surabaya	
6	bandung	
7	Cirebon	
8	Jakarta	
9	bandung	
10	Pangandaran	
11	Cimahi	
12	bandung	
13	Cimahi	

C) tampilkan kota mahasiswa yang ada tidak ada pada dosen

```
---c) Tampilkan Kota Mahasiswa yang ada tidak ada pada Dosen  
SELECT DISTINCT Kota  
FROM Mahasiswa  
WHERE Kota NOT IN (SELECT DISTINCT Kota FROM Dosen);
```

Output

	Kota
1	Cirebon
2	Pangandaran
3	Surabaya

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;
```

Output

	Nama	NIM	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

	Nama	NIM
1	Nama Mahasiswa 10	223040010

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

	JumlahMahasiswa
1	8

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;
```

Output



	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 mahasiswa, 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.

```

SELECT
    Mahasiswa>Nama,
    Mahasiswa.NIM,
    Nilai.MahasiswaID_Mahasiswa AS ID_Mahasiswa,
    Nilai.nilai,
    Nilai.MatakuliahID_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;

```

Output

Output Result 148 X							
CSV							
	Nama	NIM	ID_Mahasiswa	nilai	ID_Matakuliah	RataRataNilai	Peringkat
4	Nama Mahasiswa 5	223040005	5	85	1	69	4
5	Nama Mahasiswa 7	223040007	7	85	1	69	5
6	Nama Mahasiswa 6	223040006	6	68	1	69	6
7	Nama Mahasiswa 4	223040004	4	68	1	69	7
8	Nama Mahasiswa 9	223040009	9	55	1	69	8
9	Nama Mahasiswa 8	223040008	8	0	1	69	9
10	Nama Mahasiswa 5	223040005	5	90	2	67	1
11	Nama Mahasiswa 3	223040003	3	90	2	67	2
12	Nama Mahasiswa 6	223040006	6	88	2	67	3
13	Nama Mahasiswa 2	223040002	2	78	2	67	4
14	Nama Mahasiswa 4	223040004	4	68	2	67	5
15	Nama Mahasiswa 7	223040007	7	68	2	67	6
16	Nama Mahasiswa 9	223040009	9	68	2	67	7
17	Nama Mahasiswa 1	223040001	1	55	2	67	8
18	Nama Mahasiswa 8	223040008	8	0	2	67	9
19	Nama Mahasiswa 6	223040006	6	80	3	69	1
20	Nama Mahasiswa 1	223040001	1	80	3	69	2
21	Nama Mahasiswa 9	223040009	9	80	3	69	3
22	Nama Mahasiswa 2	223040002	2	78	3	69	4
23	Nama Mahasiswa 4	223040004	4	78	3	69	5
24	Nama Mahasiswa 3	223040003	3	78	3	69	6
25	Nama Mahasiswa 5	223040005	5	75	3	69	7
26	Nama Mahasiswa 7	223040007	7	75	3	69	8
27	Nama Mahasiswa 8	223040008	8	0	3	69	9

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

```

SELECT
    Mahasiswa.Nama,
    Mahasiswa.NIM,
    Nilai.MahasiswaID_Mahasiswa AS ID_Mahasiswa,
    Nilai.nilai,
    Nilai.MatakuliahID_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 PeringkatTanpaLoncatan
FROM
    Mahasiswa
JOIN
    Nilai ON Mahasiswa.ID_Mahasiswa = Nilai.MahasiswaID_Mahasiswa
WHERE
    Nilai.MatakuliahID_Matakuliah = 2 -- ID Mata Kuliah untuk Basis Data
ORDER BY
    Nilai.MatakuliahID_Matakuliah, Nilai.nilai DESC;

```

## Output

	Nama	NIM	ID_Mahasiswa	nilai	ID_Matakuliah	RataRataNilai	Peringkat	PeringkatTanpaLoncatan
1	Nama Mahasiswa 1	223040001	1	55	2	67	8	5
2	Nama Mahasiswa 2	223040002	2	78	2	67	4	3
3	Nama Mahasiswa 3	223040003	3	90	2	67	1	1
4	Nama Mahasiswa 4	223040004	4	68	2	67	5	4
5	Nama Mahasiswa 5	223040005	5	90	2	67	2	1
6	Nama Mahasiswa 6	223040006	6	88	2	67	3	2
7	Nama Mahasiswa 7	223040007	7	68	2	67	6	4
8	Nama Mahasiswa 8	223040008	8	0	2	67	9	6
9	Nama Mahasiswa 9	223040009	9	68	2	67	7	4

## 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
INSERT 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;

```

Output

	ID_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		6 223040006	Nama Mahasiswa 6	Alamat 6	bandung	email6@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

```

DELETE FROM mahasiswa
WHERE ID_Mahasiswa = 11;

```

Output

	ID_Mahasiswa	nim	nama	alamat	kota	email
Select All		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		6 223040006	Nama Mahasiswa 6	Alamat 6	bandung	email6@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		12 223040085	Reza Ageng	jl.Hegarmana	jawa	reza@gmail.com
12		13 223040086	Muhammad Azhar Lutfiadi	jl.gatau	subang	azharLutfiafi@gmail.com

SKEMA DIAGRAM

