

UTS BASIS DATA

(Fayyadh Ahmad Murhali 2023071065)

No 1.

1. Data Definition Language (DDL)

DDL adalah perintah yang digunakan untuk mendefinisikan atau memodifikasi struktur objek di dalam database,

seperti tabel, indeks, dan skema. Perintah DDL mengubah struktur dari database secara langsung.

Contoh Perintah DDL

```
CREATE TABLE karyawan (
```

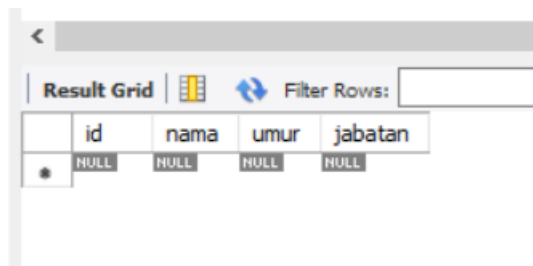
```
    id INT PRIMARY KEY,
```

```
    nama VARCHAR(50),
```

```
    umur INT,
```

```
    jabatan VARCHAR(50)
```

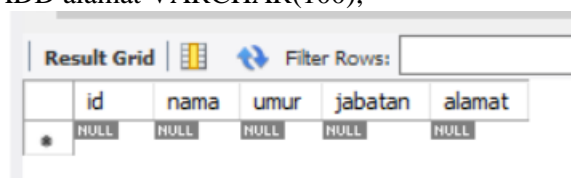
```
);
```



	id	nama	umur	jabatan
*	NULL	NULL	NULL	NULL

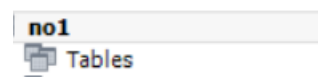
```
ALTER TABLE karyawan
```

```
ADD alamat VARCHAR(100);
```



	id	nama	umur	jabatan	alamat
*	NULL	NULL	NULL	NULL	NULL

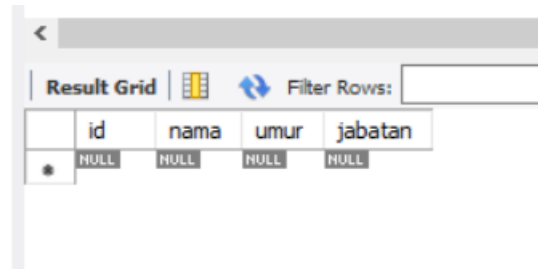
```
DROP TABLE karyawan;
```



TRUNCATE TABLE karyawan;



	id	nama	umur	jabatan
▶	1	Ahmad	30	Manager
	2	Budi	25	Staff
	3	Citra	28	HR
	4	Dewi	32	Developer
	5	Eko	29	Designer
*	NULL	NULL	NULL	NULL



	id	nama	umur	jabatan
*	NULL	NULL	NULL	NULL

2.DQL adalah perintah yang digunakan untuk mengambil data dari database,

umumnya melalui perintah SELECT. Fokus DQL adalah mengambil data tanpa melakukan modifikasi.

Contoh Perintah DQL

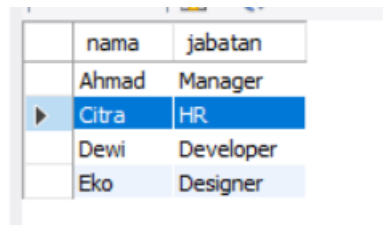
SELECT nama, jabatan

FROM karyawan

WHERE umur > 25;



	id	nama	umur	jabatan
▶	1	Ahmad	30	Manager
	2	Budi	25	Staff
	3	Citra	28	HR
	4	Dewi	32	Developer
	5	Eko	29	Designer
*	NULL	NULL	NULL	NULL



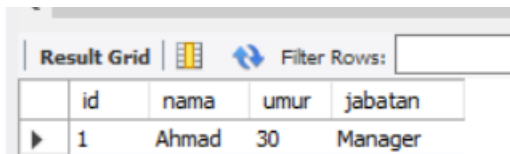
	nama	jabatan
	Ahmad	Manager
▶	Citra	HR
	Dewi	Developer
	Eko	Designer

3.Data Manipulation Language (DML)

DML adalah perintah yang digunakan untuk memanipulasi data di dalam tabel, termasuk menambah, mengubah, atau menghapus data.

Contoh perintah DML

INSERT INTO karyawan (id, nama, umur, jabatan) VALUES
(1, 'Ahmad', 30, 'Manager'),



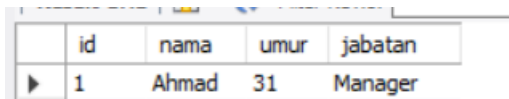
The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains one row of data with the following values:

	id	nama	umur	jabatan
▶	1	Ahmad	30	Manager

UPDATE karyawan

SET umur = 31

WHERE id = 1;

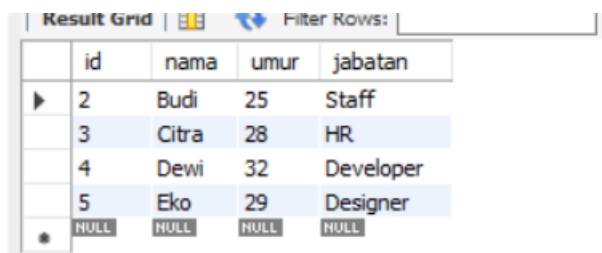


The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains one row of data with the following values:

	id	nama	umur	jabatan
▶	1	Ahmad	31	Manager

DELETE FROM karyawan

WHERE id = 1;



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains five rows of data with the following values:

	id	nama	umur	jabatan
▶	2	Budi	25	Staff
	3	Citra	28	HR
	4	Dewi	32	Developer
	5	Eko	29	Designer
•	NULL	NULL	NULL	NULL

Perbedaan

DDL digunakan untuk mendefinisikan dan mengubah struktur database dan tabelnya, berfokus pada objek di dalam database, seperti membuat atau menghapus tabel.

DQL berfungsi untuk mengambil atau membaca data yang ada di database tanpa melakukan modifikasi data.

DML digunakan untuk manipulasi atau modifikasi data di dalam tabel, seperti menambah atau memperbarui data.

Data Mahasiswa

[illegible]

```
CREATE TABLE uts_basisdata.Kartu_Rencana_Studi (
    No INT AUTO_INCREMENT PRIMARY KEY,          -- Nomor urut sebagai primary key dan auto-increment
    Kode CHAR(6) NOT NULL,                       -- Kode mata kuliah, tipe CHAR(6)
    NIM CHAR(10) NOT NULL,                       -- NIM mahasiswa, tipe CHAR(10) untuk relasi dengan "Data_Mahasiswa"
    Nama_Mata_Kuliah VARCHAR(100) NOT NULL,      -- Nama mata kuliah, tipe VARCHAR
    Nama_Kelas VARCHAR(10) NOT NULL,            -- Nama kelas, tipe VARCHAR
    SKS TINYINT UNSIGNED NOT NULL,              -- Jumlah SKS mata kuliah, tipe TINYINT
    Jadwal DATETIME NOT NULL,                   -- Jadwal mata kuliah, tipe DATETIME
    Dosen_Pengajar VARCHAR(100) NOT NULL,       -- Nama dosen pengajar, tipe VARCHAR
    Keterangan TEXT,                             -- Keterangan tambahan, tipe TEXT
    UNIQUE (NIM, Kode)                          -- Key unik gabungan untuk menghindari duplikasi NIM dan Kode mata kuliah
);

-- Menambahkan foreign key pada kolom NIM untuk menghubungkan tabel "Kartu_Rencana_Studi" dengan tabel "Data_Mahasiswa"
ALTER TABLE uts_basisdata.Kartu_Rencana_Studi
ADD CONSTRAINT fk_nim_krs FOREIGN KEY (NIM) REFERENCES uts_basisdata.Data_Mahasiswa(NIM);
```

[illegible]

Kartu rencana studi

```
CREATE TABLE uts_basisdata.Kartu_Hasil_Studi (  
    No INT AUTO_INCREMENT PRIMARY KEY,      -- Nomor urut sebagai primary key dan auto-increment  
    Kode CHAR(6) NOT NULL,                  -- Kode mata kuliah, tipe CHAR(6)  
    NIM CHAR(10) NOT NULL,                  -- NIM mahasiswa, tipe CHAR(10)  
    Nama_Mata_Kuliah VARCHAR(100) NOT NULL, -- Nama mata kuliah, tipe VARCHAR  
    SKS TINYINT UNSIGNED NOT NULL,          -- Jumlah SKS mata kuliah, tipe TINYINT  
    Nilai_Mutu DECIMAL(3, 2),               -- Nilai mutu, tipe DECIMAL(3,2)  
    Bobot DECIMAL(3, 2),                   -- Bobot nilai berdasarkan nilai mutu, tipe DECIMAL(3,2)  
    Nilai CHAR(1),                          -- Nilai huruf (A, B, C, dst.), tipe CHAR(1)  
    Keterangan TEXT,                       -- Keterangan tambahan, tipe TEXT  
    UNIQUE (NIM, Kode)                     -- Key unik gabungan untuk mencegah duplikasi NIM dan Kode mata kuliah  
);  
  
-- Menambahkan foreign key pada kolom NIM untuk menghubungkan tabel "Kartu_Hasil_Studi" dengan tabel "Data_Mahasiswa"  
ALTER TABLE uts_basisdata.Kartu_Hasil_Studi  
ADD CONSTRAINT fk_nim_khs FOREIGN KEY (NIM) REFERENCES uts_basisdata.Data_Mahasiswa(NIM);  
  
-- Menambahkan foreign key pada kolom Kode untuk menghubungkan tabel "Kartu_Hasil_Studi" dengan tabel "Kartu_Rencana_Studi"  
ALTER TABLE uts_basisdata.Kartu_Hasil_Studi  
ADD CONSTRAINT fk_kode_khs FOREIGN KEY (Kode) REFERENCES uts_basisdata.Kartu_Rencana_Studi(Kode);
```

Result Grid										
Filter Rows:										
	No	Kode	NIM	Nama_Mata_Kuliah	SKS	Nilai_Mutu	Bobot	Nilai	Keterangan	
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

No 3

1. Buatlah table dengan struktur kolom diatas dan tambahkan data

```
1 CREATE TABLE employee_salary (  
2     employee_ID INT,  
3     employee_name VARCHAR(50),  
4     salary INT,  
5     allowance INT  
6 );  
7  
8 INSERT INTO employee_salary (employee_ID, employee_name, salary, allowance)  
9 VALUES  
10 (1, 'Alex', 25000, 1000),  
11 (2, 'John', 55000, 1000),  
12 (3, 'James', 52000, 1000),  
13 (4, 'Sam', 30000, 1000);  
14
```

Result Grid				
Filter Rows:				
	employee_ID	employee_name	salary	allowance
▶	1	Alex	25000	1000
	2	John	55000	1000
	3	James	52000	1000
	4	Sam	30000	1000

2. SELECT employee_ID, employee_name, (salary + allowance) AS total_compensation

FROM kariawan.employee_salary;

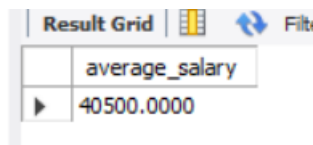


The screenshot shows a 'Result Grid' window with a table containing four rows of employee data. The columns are employee_ID, employee_name, and total_compensation. The data is as follows:

employee_ID	employee_name	total_compensation
1	Alex	26000
2	John	56000
3	James	53000
4	Sam	31000

3. SELECT AVG(salary) AS average_salary

FROM kariawan.employee_salary;



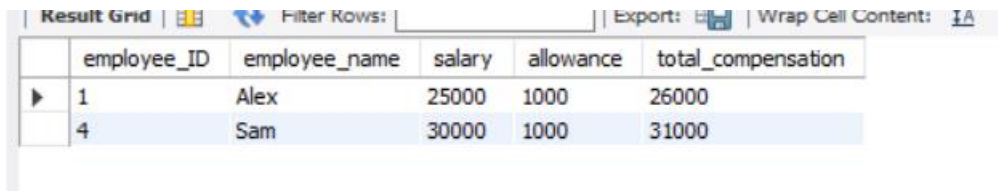
The screenshot shows a 'Result Grid' window with a single row containing the average salary.

average_salary
40500.0000

4. SELECT employee_ID, employee_name, salary, allowance, (salary + allowance) AS total_compensation

FROM kariawan.employee_salary

WHERE salary + allowance < 40000;



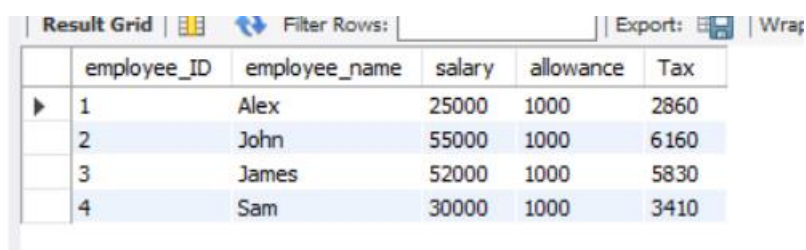
The screenshot shows a 'Result Grid' window with a table containing two rows of employee data. The columns are employee_ID, employee_name, salary, allowance, and total_compensation. The data is as follows:

employee_ID	employee_name	salary	allowance	total_compensation
1	Alex	25000	1000	26000
4	Sam	30000	1000	31000

5. ALTER TABLE kariawan.employee_salary ADD Tax INT;

UPDATE kariawan.employee_salary

SET Tax = ROUND((salary + allowance) * 0.11);

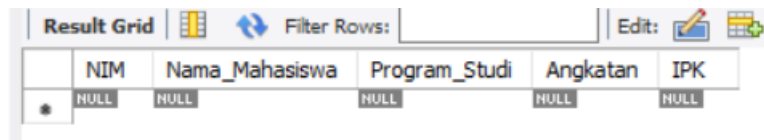


The screenshot shows a 'Result Grid' window with a table containing four rows of employee data. The columns are employee_ID, employee_name, salary, allowance, and Tax. The data is as follows:

employee_ID	employee_name	salary	allowance	Tax
1	Alex	25000	1000	2860
2	John	55000	1000	6160
3	James	52000	1000	5830
4	Sam	30000	1000	3410

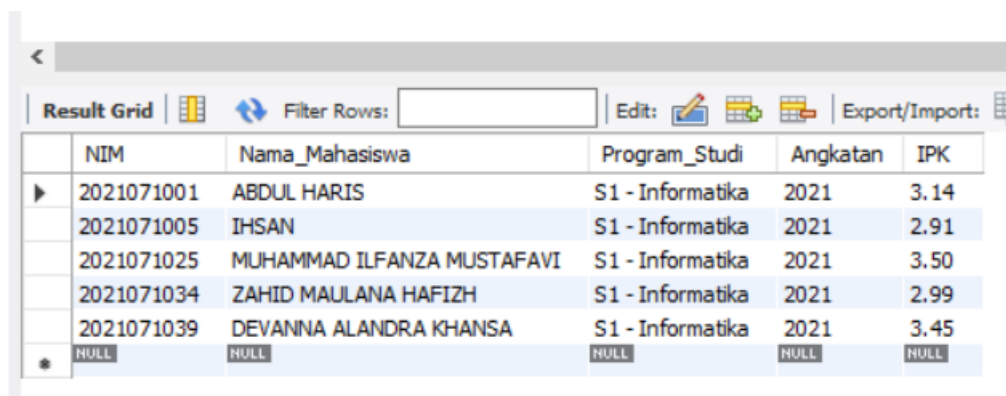
No 4.

a. CREATE TABLE Data_IPK (
NIM VARCHAR(10) PRIMARY KEY,
Nama_Mahasiswa VARCHAR(100),
Program_Studi VARCHAR(50),
Angkatan INT,
IPK DECIMAL(3,2)
);



	NIM	Nama_Mahasiswa	Program_Studi	Angkatan	IPK
*	NULL	NULL	NULL	NULL	NULL

b. INSERT INTO Data_IPK (NIM, Nama_Mahasiswa, Program_Studi, Angkatan, IPK) VALUES
('2021071001', 'ABDUL HARIS', 'S1 - Informatika', 2021, 3.14),
('2021071025', 'MUHAMMAD ILFANZA MUSTAFAVI', 'S1 - Informatika', 2021, 3.50),
('2021071034', 'ZAHID MAULANA HAFIZH', 'S1 - Informatika', 2021, 2.99),
('2021071039', 'DEVANNA ALANDRA KHANSA', 'S1 - Informatika', 2021, 3.45),
('2021071005', 'IHSAN', 'S1 - Informatika', 2021, 2.91);

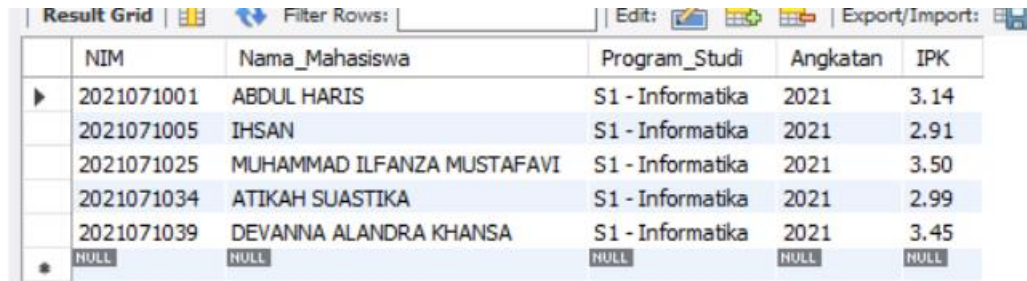


	NIM	Nama_Mahasiswa	Program_Studi	Angkatan	IPK
▶	2021071001	ABDUL HARIS	S1 - Informatika	2021	3.14
	2021071005	IHSAN	S1 - Informatika	2021	2.91
	2021071025	MUHAMMAD ILFANZA MUSTAFAVI	S1 - Informatika	2021	3.50
	2021071034	ZAHID MAULANA HAFIZH	S1 - Informatika	2021	2.99
	2021071039	DEVANNA ALANDRA KHANSA	S1 - Informatika	2021	3.45
*	NULL	NULL	NULL	NULL	NULL

C. UPDATE Data_IPK

SET Nama_Mahasiswa = 'ATIKAH SUASTIKA'

WHERE NIM = '2021071034';

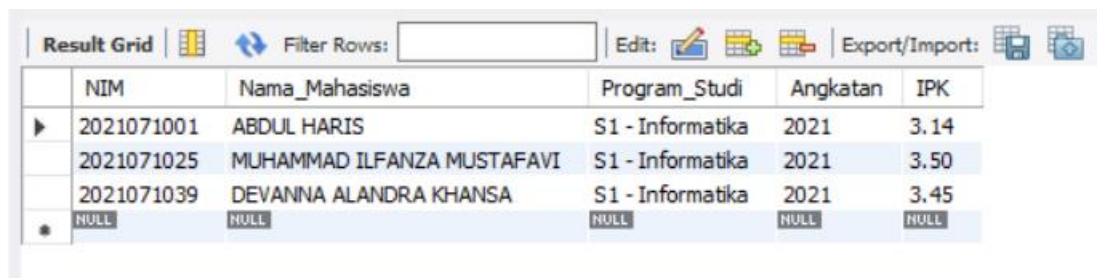


	NIM	Nama_Mahasiswa	Program_Studi	Angkatan	IPK
▶	2021071001	ABDUL HARIS	S1 - Informatika	2021	3.14
	2021071005	IHSAN	S1 - Informatika	2021	2.91
	2021071025	MUHAMMAD ILFANZA MUSTAFAVI	S1 - Informatika	2021	3.50
	2021071034	ATIKAH SUASTIKA	S1 - Informatika	2021	2.99
	2021071039	DEVANNA ALANDRA KHANSA	S1 - Informatika	2021	3.45
*	NULL	NULL	NULL	NULL	NULL

d. SELECT *

FROM Data_IPK

WHERE IPK > 3;

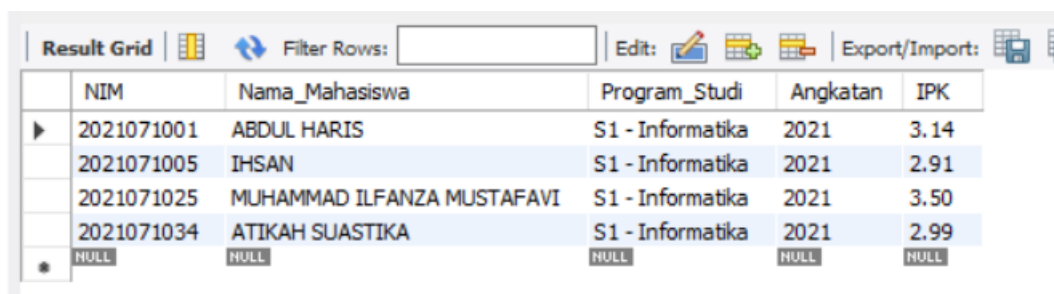


	NIM	Nama_Mahasiswa	Program_Studi	Angkatan	IPK
▶	2021071001	ABDUL HARIS	S1 - Informatika	2021	3.14
	2021071025	MUHAMMAD ILFANZA MUSTAFAVI	S1 - Informatika	2021	3.50
	2021071039	DEVANNA ALANDRA KHANSA	S1 - Informatika	2021	3.45
*	NULL	NULL	NULL	NULL	NULL

E.

DELETE FROM Data_IPK

WHERE NIM = '2021071039';



	NIM	Nama_Mahasiswa	Program_Studi	Angkatan	IPK
▶	2021071001	ABDUL HARIS	S1 - Informatika	2021	3.14
	2021071005	IHSAN	S1 - Informatika	2021	2.91
	2021071025	MUHAMMAD ILFANZA MUSTAFAVI	S1 - Informatika	2021	3.50
	2021071034	ATIKAH SUASTIKA	S1 - Informatika	2021	2.99
*	NULL	NULL	NULL	NULL	NULL