**Query** adalah permintaan informasi. Begitu juga dalam bahasa pemrograman komputer, query atau kueri mengacu pada permintaan informasi, tapi informasi ini diambil dari database. query merupakan syntax atau perintah yang dipakai agar dapat saling berinteraksi.

**Query** lebih dikenal dengan sebutan SQL (Structured **Query** Language) yang artinya adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional agar bisa berinteraksi dengan suatu sistem database atau lebih tepatnya suatu sistem manajemen database relasional

**Query-query pada SQL (Structured Query Language)**

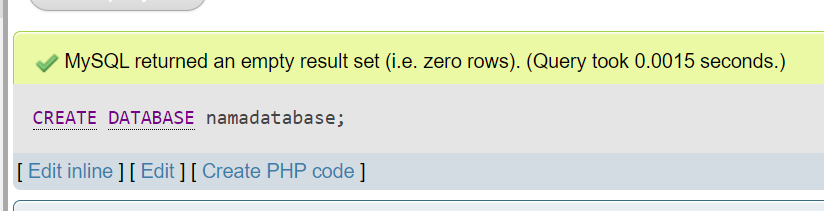
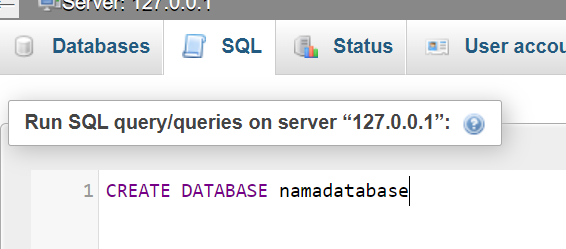
* **Create Database**

Digunakan untuk membuat database baru.

Syntax dasar:

CREATE DATABASE database\_nama

#membuat database dengan nama namadatabase



* **Create Table**

Digunakan untuk membuat tabel data baru dalam sebuah database.

Syntax dasar:

CREATE TABLE table\_name (

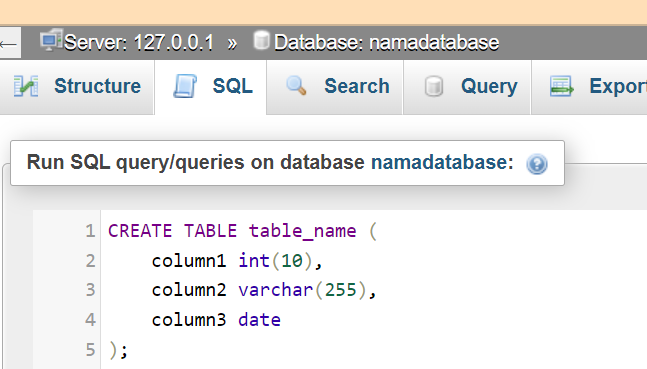
column1 datatype,

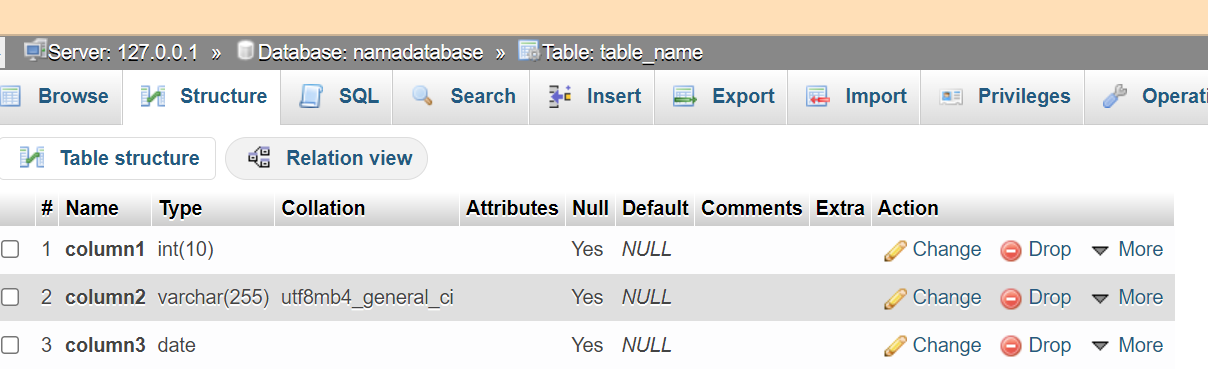
column2 datatype,

column3 datatype

);

#membuat tabel dengan nama table\_name yang berisi column 1 sampai 3 dengan tipe data dan jumlah values nya





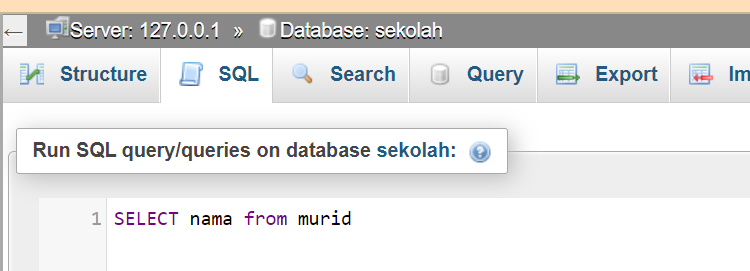
* **Select**

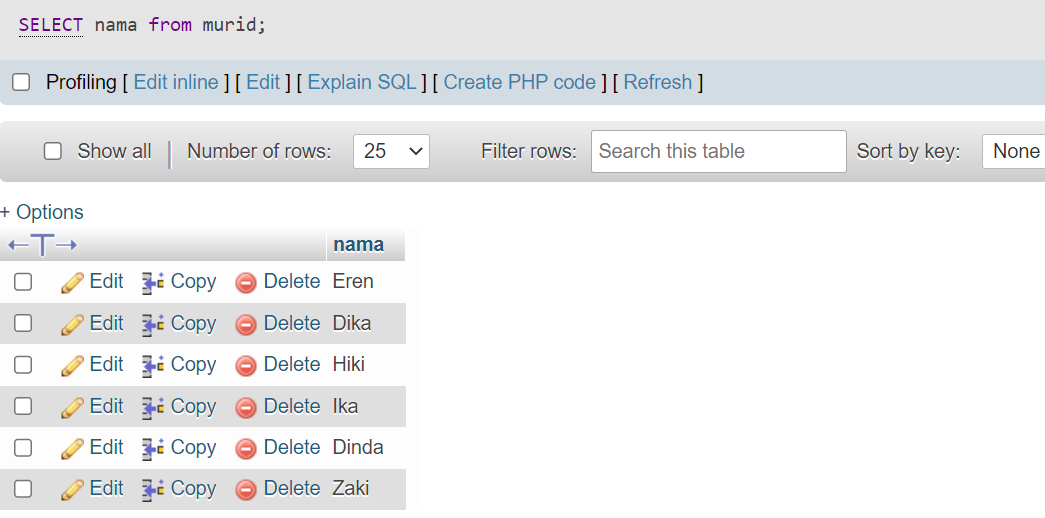
Digunakan untuk memilih data dari table database.

Syntax dasar:

SELECT column\_name(s) FROM table\_name

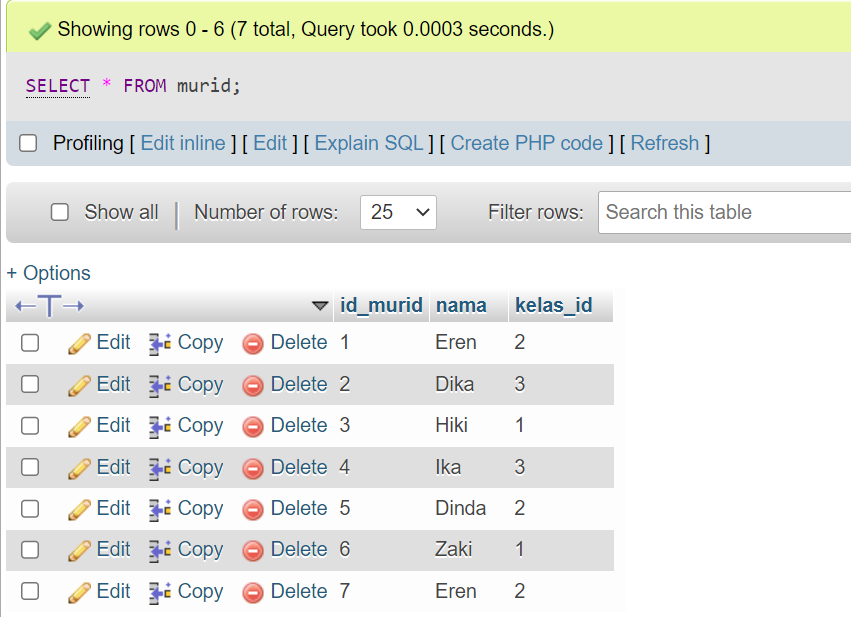
#memilih field nama dari tabel murid





Atau SELECT \* FROM table\_name

#memilih semua field yang ada dari murid



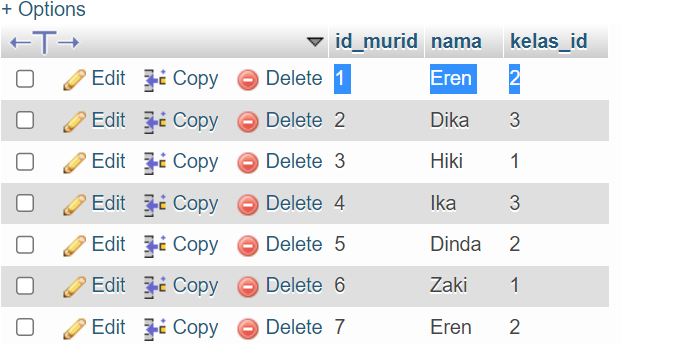
* **Select Distinct**

Digunakan untuk memilih data-data yang berbeda (menghilangkan duplikasi) dari sebuah table database.

Syntax dasar:

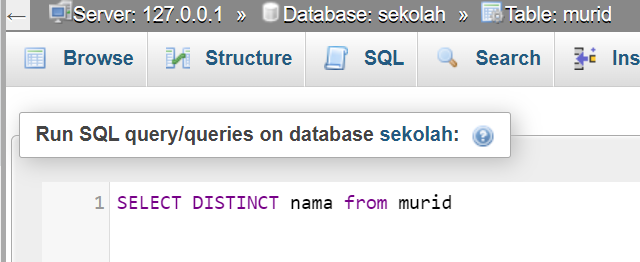
SELECT DISTINCT column\_name(s) FROM table\_name

isi tabel sebelum SELECT DISTINCT:

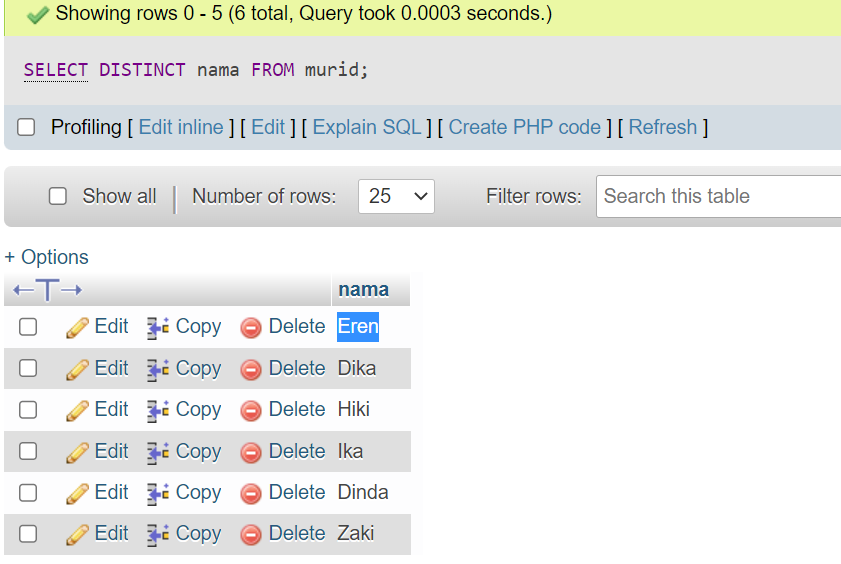


melakukan SELECT DISTINCT:

#memilih dan menghilangkan duplikasi isi nilai dari field nama yang sama dari tabel murid



setelah SELECT DISTINCT:



* **Where**

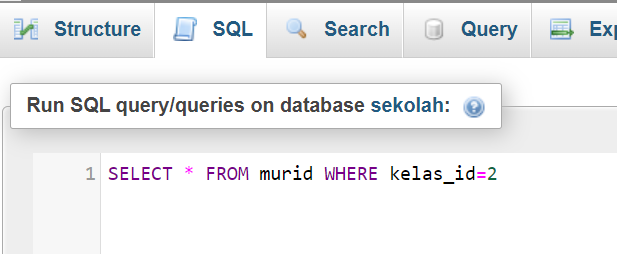
Digunakan untuk memfilter data pada perintah Select

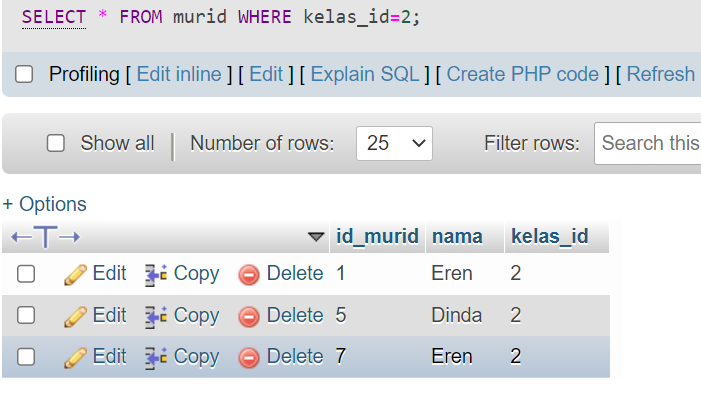
Syntax dasar:

SELECT column name(s) FROM table\_name WHERE column\_name operator value

atau SELECT \* FROM table\_name WHERE column\_name operator value

#memilih semua dari tabel murid dan memfilter berdasarkan field kelas\_id yang ber id 2





* **Order By**

Digunakan untuk mengurutkan data berdasarkan kolom (field) tertentu. Secara default, urutan tersusun secara ascending (urut kecil ke besar). Anda dapat mengubahnya menjadi descending (urut besar ke kecil) dengan menambahkan perintah DESC.

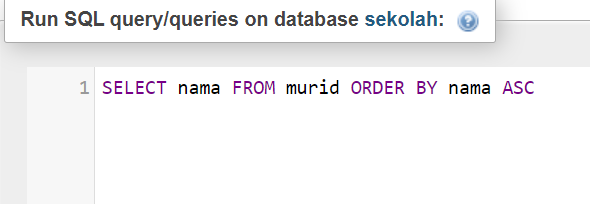
Syntax dasar:

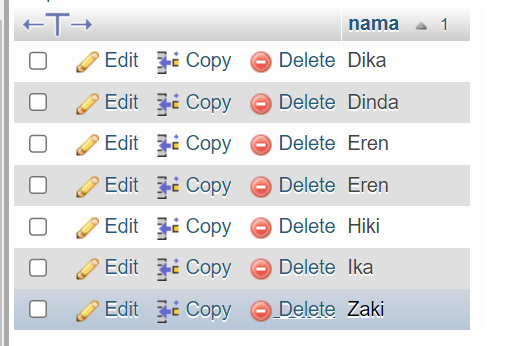
SELECT column\_name(s)

FROM table\_name

ORDER BY column\_name(s) ASC|DESC

#memilih field nama dari tabel murid lalu mengurutkannya berdasarkan ASC





* **Like**

Digunakan bersama dengan perintah Where, untuk proses pencarian data dengan spesifikasi tertentu.

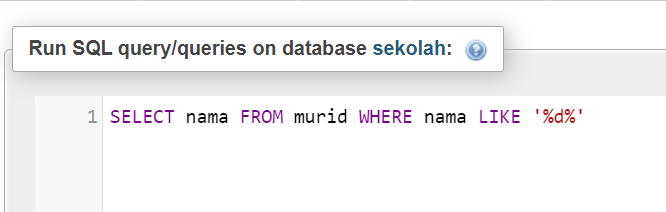
Syntax dasar:

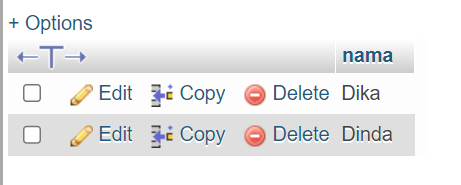
SELECT column\_name(s)

FROM table\_name

WHERE column\_name LIKE pattern

#memilih field nama dari tabel murid berdasarkan field nama untuk mencari inisial huruf d





* **In**

Digunakan untuk pencarian data menggunakan lebih dari satu filter pada perintah Where.

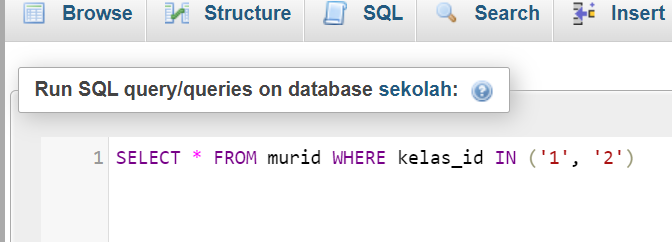
Syntax dasar :

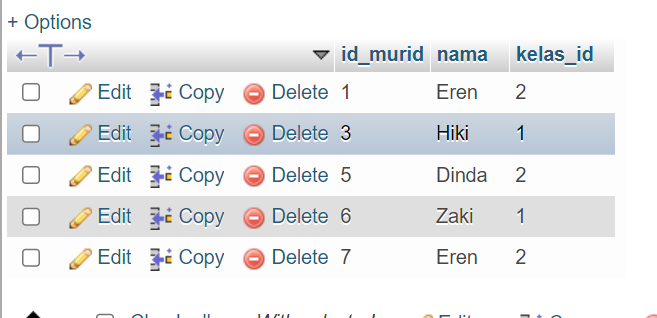
SELECT column\_name(s)

FROM table\_name

WHERE column\_name IN (value1,value2, . . .)

#mencari semua field dari tabel murid berdasarkan kelas\_id dan di filter berdasarkan kelas\_id yang valuesnya 1 dan 2





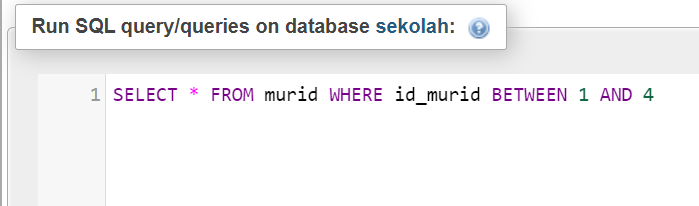
* **Between**

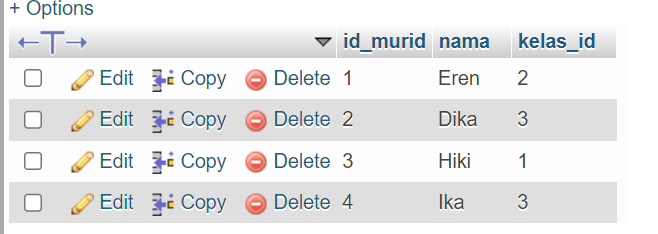
Digunakan untuk menentukan jangkauan pencarian.

Syntax dasar:

SELECT column\_name FROM table\_name WHERE column\_name BETWEEN value1 AND value2

#mencari semua field dari tabel murid berdasarkan id\_murid antara 1 sampai 4





* **Insert Into**

Digunakan untuk menambahkan data baru di tabel database.

Syntax dasar :

INSERT INTO table\_name

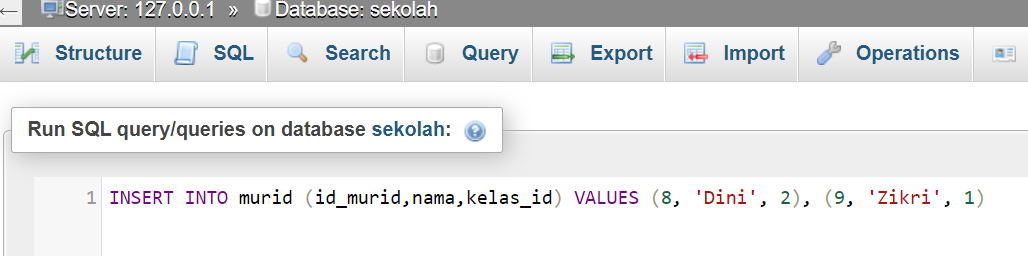
VALUES (value1,value2,value3, . . .)

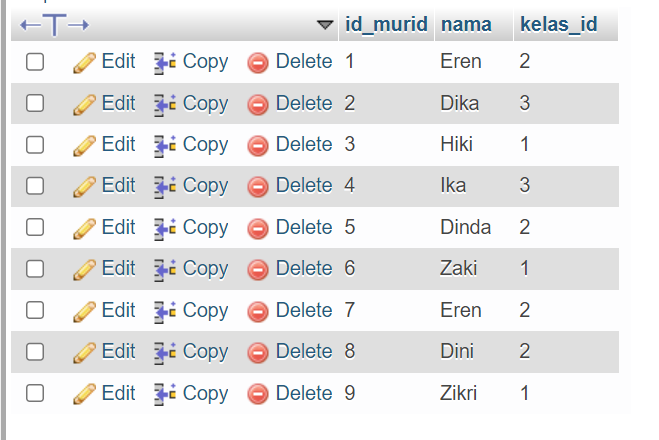
Atau

INSERT INTO table\_name (column1,column2,column3, . . .)

VALUES (value1,value2,value3, . . .)

#menambah data di tabel murid untuk mengisi values field id\_murid, nama, dan kelas\_id





* **Update**

Digunakan untuk mengubah/memperbarui data di tabel database.

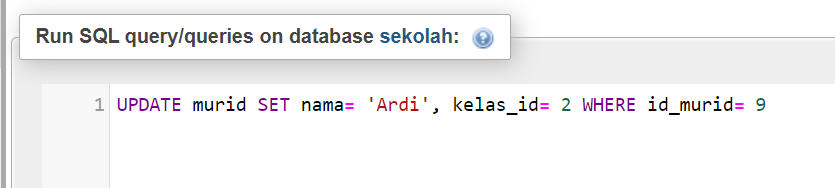
Syntax dasar:

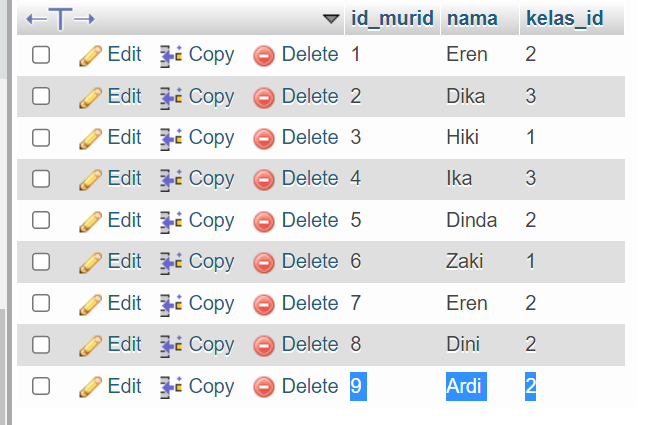
UPDATE table\_name

SET column1=value,column2=value, . . .

WHERE some\_column=some\_value

#mengubah/ memperbarui data di tabel murid dan memilih field nama dan kelas\_id nya untuk di ubah values nya berdasarkan id\_murid ke 9





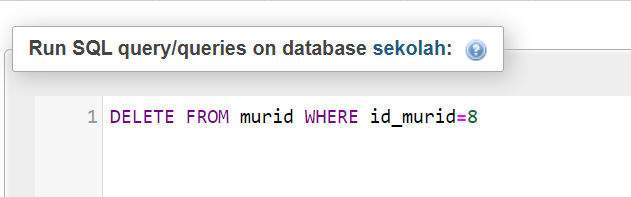
* **Delete**

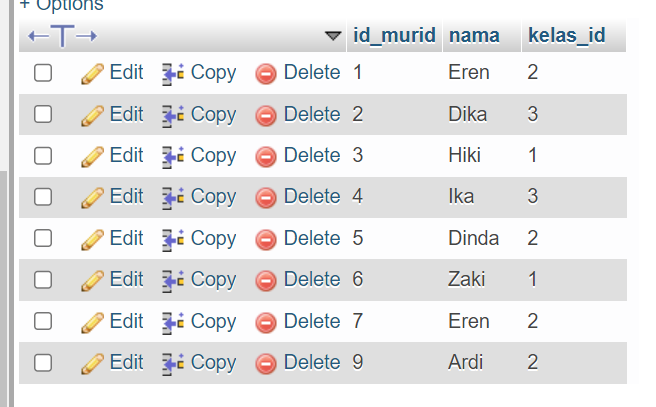
Digunakan untuk menghapus data di table database. Tambahkan perintah Where untuk memfilter data-data tertentu yang akan dihapus. Jika tanpa perintah Where, maka seluruh data dalam tabel akan terhapus.

Syntax dasar :

DELETE FROM table\_name WHERE some\_column=some\_value

#menghapus dari tabel murid berdasarkan id\_murid ke 8





* **Inner Join**

Digunakan untuk menghasilkan baris data dengan cara menggabungkan 2 buah tabel atau lebih menggunakan pasangan data yang match pada masing-masing tabel. Perintah ini sama dengan perintah join yang sering digunakan.

Syntax dasar :

SELECT column\_name(s)

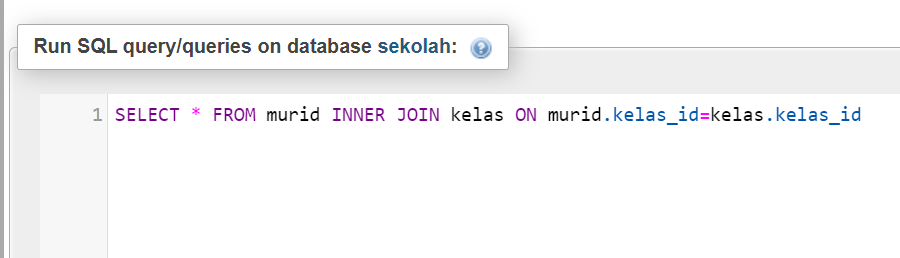
FROM table\_name1

INNER JOIN table\_name2

ON table\_name1.column\_name=table\_name2

column-name

#memilih semua field dari tabel murid lalu digabung dengan tabel kelas lalu tampilkan data yang sama antara field kelas\_id pada tabel murid dan field kelas\_id pada tabel kelas





* **Left Join**

Digunakan untuk menghasilkan baris data dari tabel kiri (nama tabel pertama) yang tidak ada pasangan datanya pada tabel kanan (nama tabel kedua).

Syntax dasar :

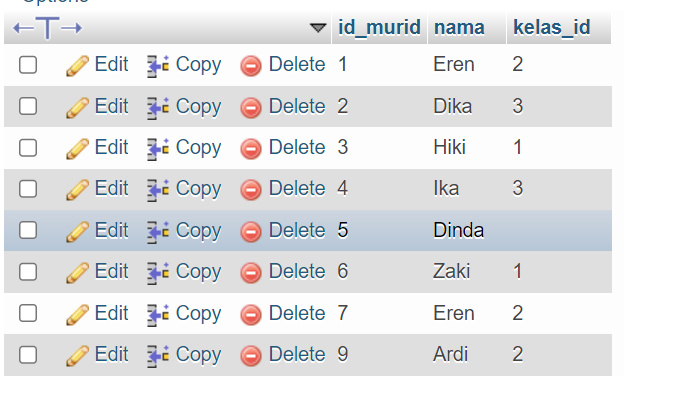
SELECT column\_name(s)

FROM table\_name1

LEFT JOIN table\_name2

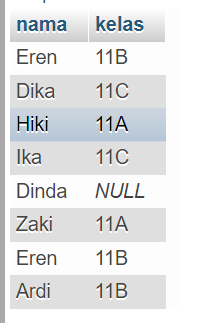
ON table\_name1.column\_name=table\_name2.

column\_name



#memilih field nama tabel murid dan field kelas tabel kelas dari tabel murid digabung dengan field kelas\_id pada tabel murid dan field kelas\_id pada tabel kelas dan ditampilkan datanya berdasarkan tabel pertama ( di bagian kiri )





* **Right Join**

Digunakan untuk menghasilkan baris data dari tabel kanan (nama tabel kedua) yang tidak ada pasangan datanya pada tabel kiri (nama tabel pertama).

Syntax dasar :

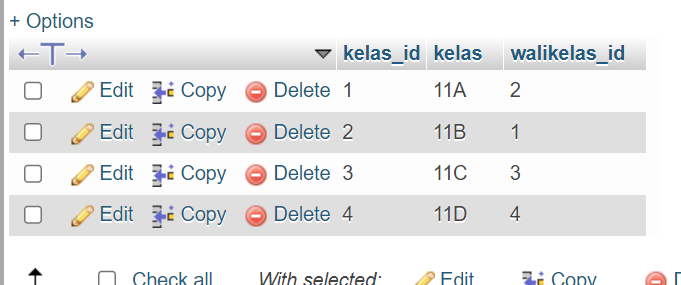
SELECT column\_name(s)

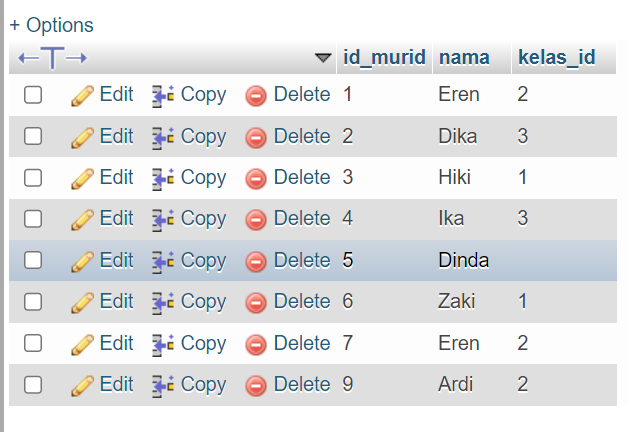
FROM table\_name1

RIGHT JOIN table\_name2

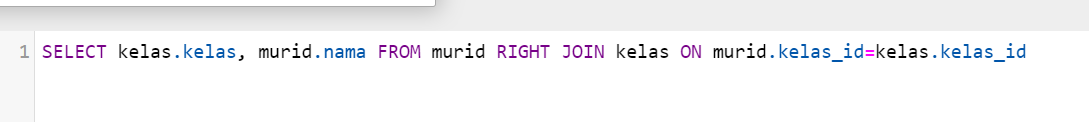
ON table\_name1.column\_name=table\_name2

column\_name





#memilih field nama tabel murid dan field kelas tabel kelas dari tabel murid digabung dengan field kelas\_id pada tabel murid dan field kelas\_id pada tabel kelas dan ditampilkan datanya berdasarkan tabel kedua( di bagian kanan)

****

****

* **Alter Table**

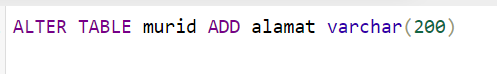
Digunakan untuk menambah, menghapus, atau mengubah kolom (field) pada tabel yang sudah ada.

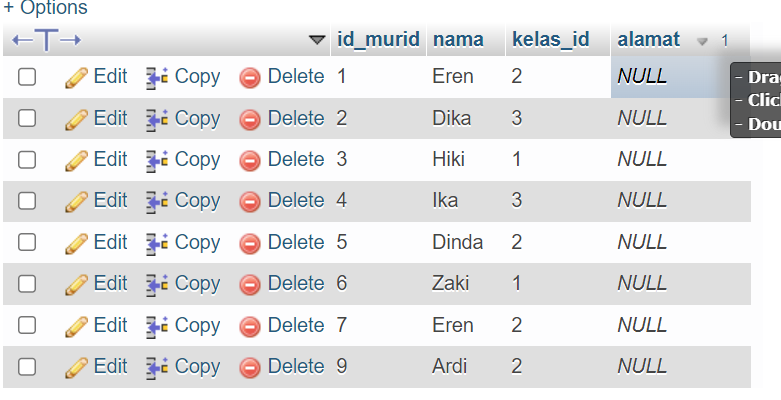
Syntax untuk menambah kolom :

ALTAR TABLE table\_name

ADD column\_name datatype

#menambah field alamat dan tipe data varchar pada tabel murid



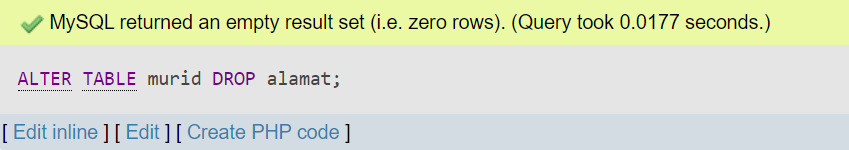


Syntax untuk menghapus kolom :

ALTER TABLE table\_name

DROP COLUMN column\_name

#menghapus field alamat pada tabel murid



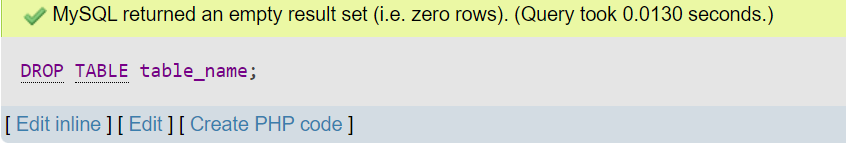
* **Drop Table**

Digunakan untuk menghapus tabel beserta seluruh datanya.

Syntax dasar :

DROP TABLE table\_name

#menghapus tabel



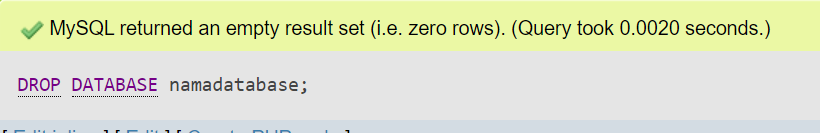
* **Drop Database**

Digunakan untuk menghapus database.

Syntax dasar :

DROP DATABASE database\_name

#menghapus database



**~Izanagi**