# SQL (Structured Query Language)

## ADMINISTRATION SYNTAX

Administration syntax digunakan untuk mengatur *user account*.

### MEMBUAT USER

**CREATE USER** *‘<nama\_user>*’**@**’*<localhost> atau <%> atau <nama\_host> atau <IP>*’

**IDENTIFIED BY** ‘*<password>*’;

### MENGHAPUS USER

**DROP USER** 'user\_name'@'localhost *atau <%> atau <nama\_host> atau <IP>*’;

### MEMBERI HAK AKSES kepada USER

**GRANT** *<Hak akses>, <Hak akses>, …*

**ON** *<nama\_database>*.*<nama\_table> atau <<nama\_database>*.*\**

**TO** 'user\_name'@'localhost *atau <%> atau <nama\_host> atau <IP>*’;

### MENCABUT HAK AKSES dari USER

**REVOKE** *<Hak akses>, <Hak akses>, …*

**ON** *<nama\_database>*.*<nama\_table> atau <<nama\_database>*.*\**

**TO** 'user\_name'@'localhost *atau <%> atau <nama\_host> atau <IP>*’;

**Hak akses**:

|  |  |  |
| --- | --- | --- |
| Hak terhadap Data | Hak terhadap Structure | Hak terhadap Administration |
| Hak\_thd\_data:   * SELECT * INSERT * UPDATE * DELETE * FILE | Hak\_thd\_structure:   * CREATE * ALTER * INDEX * DROP * CREATE TEMPORARY TABLES * SHOW VIEW * CREATE ROUTINE * ALTER ROUTINE * EXECUTE * CREATE VIEW * EVENT * TRIGGER | Hak\_thd\_structure:   * GRANT * SUPER * PROCESS * RELOAD * SHUTDOWN * SHOW DATABASES * LOCK TABLES * REFERENCES * REPLICATION CLIENT * REPLICATION SLAVE * CREATE USER |

Contoh:

C:>MySql –u root –h 192.168.70.160

Mysql>CREATE DATABASE augury;

Mysql>CREATE USER ‘augury’@’%’ IDENTIFIED BY ‘1234’;

Mysql>GRANT ALL ON augury.\* TO ‘augury’@’%’;

***Keterangan****:*

*Baris 1🡪 root membuat database augury*

*Baris 2🡪 root membuat user augury dengan hak masuk dari mana saja, dan password 1234*

*Baris 3🡪 root memberikan hak akses penuh terhadap database augury kepada user augury*

Mysql> CREATE USER ‘tes’@’localhost’ IDENTIFIED BY ‘tes1234’;

Mysql>GRANT CREATE ON augury.\* TO ‘tes’@’localhost’;

***Keterangan****:*

*Baris 2🡪 root membuat user tes dengan hak masuk dari localhost, dan password tes1234*

*Baris 3🡪 root memberikan hak akses membuat table terhadap database augury kepada user tes*

Mysql> CREATE USER ‘tes2’@’%’ IDENTIFIED BY ‘tes234’;

Mysql>GRANT SELECT(‘isi’) ON augury.test TO ‘tes2’@’%’;

***Keterangan****:*

*Baris 2🡪 root membuat user tes2 dengan hak masuk dari mana saja, dan password tes234*

*Baris 3🡪 root memberikan hak akses select hanya pada field isi terhadap database augury table test kepada user tes*

Mysql>EXIT

## DDL (Data Definition Language)

DDL digunakan untuk mendeskripsikan skema database.

### DDL DATABASE

CREATE DATABASE *<nama\_database>*

Contoh:

CREATE DATABASE kampus

DROP DATABASE *<nama\_database>*

ALTER DATABASE *<nama\_database>*

alter\_specification ...

### DDL TABLE

**CREATE TABLE**

CREATE TABLE <nama\_tabel> (<nama\_kolom> <tipe\_kolom> [atribut\_kolom], … … … ,

[definisi\_key] (<nama\_kolom>) );

Contoh:

CREATE TABLE panduan(

id\_panduan INT NOT NULL AUTO\_INCREMENT,

judul\_panduan VARCHAR(100) NOT NULL,

penulis VARCHAR(40) NOT NULL,

tanggal\_publikasi DATE,

PRIMARY KEY (id\_panduan )

);

Keterangan:

[atribut\_kolom]:

**NOT NULL** : digunakan jika kita ingin agar **kolom ini tidak boleh NULL** (atau tidak diisi). Jadi jika user membuat record data, maka kolom dengan atribut\_kolom NOT NULL harus diisi, jika tidak maka MySQL akan memberitahu terjadi error.

**AUTO\_INCREMENT** : digunakan jika kita ingin **kolom ini berisi angka yang merupakan penambahan dari angka record sebelumnya**, dan penambahan ini otomatis dilakukan oleh MySQL.

[definisi\_key]

**PRIMARY KEY** : digunakan untuk mendefinisikan suatu kolom sebagai primary key. Kita bisa mendefinisikan lebih dari satu kolom sebagai primary key (dengan menggunakan koma sebagai pemisah kolom).

**DESCRIBE TABLE**

Untuk melihat struktur tabel yang sudah dibuat

DESCRIBE *<nama\_tabel>*

**DROP TABLE**

DROP TABLE *<nama\_tabel>*

**ALTER TABLE**

ALTER [IGNORE] TABLE tbl\_name

alter\_specification [, alter\_specification] ,...

***alter\_specification***:

ADD [COLUMN] column\_definition [FIRST | AFTER col\_name ]

| ADD [COLUMN] (column\_definition,...)

| ADD INDEX [index\_name] [index\_type] (index\_col\_name,...)

| ADD [CONSTRAINT [symbol]] PRIMARY KEY [index\_type] (index\_col\_name,...)

| ADD [CONSTRAINT [symbol]] UNIQUE [index\_name] [index\_type] (index\_col\_name,...)

| ADD [FULLTEXT|SPATIAL] [index\_name] (index\_col\_name,...)

| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index\_name] (index\_col\_name,...) [reference\_definition]

| ALTER [COLUMN] col\_name {SET DEFAULT literal | DROP DEFAULT}

| CHANGE [COLUMN] old\_col\_name column\_definition [FIRST|AFTER col\_name]

| MODIFY [COLUMN] column\_definition [FIRST | AFTER col\_name]

| DROP [COLUMN] col\_name

| DROP PRIMARY KEY

| DROP INDEX index\_name

| DROP FOREIGN KEY fk\_symbol

| DISABLE KEYS

| ENABLE KEYS

| RENAME [TO] new\_tbl\_name

| ORDER BY col\_name

| CONVERT TO CHARACTER SET charset\_name [COLLATE collation\_name]

| [DEFAULT] CHARACTER SET charset\_name [COLLATE collation\_name]

| DISCARD TABLESPACE

| IMPORT TABLESPACE

| table\_options

## DML (Data Manipulation Language)

DML digunakan untuk memproses dan memanipulasi data pada tabel.

**INSERT**

Untuk memasukkan atau menambah baris data pada tabel

**INSERT INTO** <nama\_tabel> **(** <field\_1>, <field\_2>, … <field\_N> **)**

**VALUES (** nilai\_1, nilai\_2, … nilai\_N **);**

Atau

**INSERT INTO** <nama\_tabel>

**VALUES (** nilai\_1, nilai\_2, … nilai\_N **);**

Contoh:

**INSERT INTO** Mahasiswa (nim, nama,thnMsk)

**VALUES** (‘2015081001’, ‘Stefanus Setya Wahyudi’, 2015 );

**INSERT INTO** Mahasiswa

**VALUES** (‘2015081007’, ‘Dava Eranda Rahmatza’, 2015 );

**UPDATE**

Untuk merubah data pada suatu baris data dengan kriteria tertentu dalam tabel

**UPDATE** <nama\_tabel>

**SET** <field\_1>=nilai\_1, <field\_2>=nilai\_2

**WHERE** <kriteria> ;

Contoh:

**UPDATE** Mahasiswa

**SET** thnMsk=2013

**WHERE** nim=‘2013011001’;

**UPDATE** Mahasiswa

**SET** thnMsk=2015

**WHERE** nim=‘2015101028’;

**DELETE**

Untuk menghapus baris data dengan kriteria tertentu pada tabel

**DELETE FROM** <nama\_tabel>

**WHERE** <kriteria> ;

Contoh:

**DELETE FROM** Mahasiswa

**WHERE** nim=‘2013011101’;

**SELECT**

**SELECT** field1, field2,...fieldN table\_name1, table\_name2...

[**WHERE** Clause]

[OFFSET M ][LIMIT N]

Atau

**SELECT** \*

[**WHERE** Clause]

[OFFSET M ][LIMIT N]

Contoh:

**SELECT \* FROM** Mahasiswa;

**SELECT** nama, thnMsk **FROM** Mahasiswa **WHERE** nim=‘2013011101’;

**SELECT** nim, thnMsk FROM Mahasiswa;

**SELECT** nim,nama FROM Mahasiswa WHERE thnMsk=2013;

**SELECT** nama FROM Mahasiswa WHERE nim=‘2015081007’;