

Konsep dasar SQL, konsep DDL dan DML, serta Query SELECT

Sesi 2-Bootcamp Data Analyst with
SQL and Python using Google Platform



Memahami perintah DDL

Data Definition Language (DDL) merupakan perintah SQL untuk membuat, merubah atau menghapus struktur database.

Perintah DDL antara lain

- CREATE -> Membuat database
- DROP -> Menghapus table/database

Perintah CREATE

Database berisi table untuk menyimpan entitas. Table terdiri dari field (kolom) dan record (baris data). Perintah CREATE digunakan untuk membuat table.

```
CREATE TABLE table_name(  
    column_name_1 data_type default value column_constraint,  
    column_name_2 data_type default value column_constraint,  
    ...,  
    table_constraint  
);
```

Contoh SQL CREATE TABLE

Berikut adalah pernyataan yang membuat tabel courses:

```
CREATE TABLE courses (  
    course_id INT AUTO_INCREMENT PRIMARY KEY,  
    course_name VARCHAR(50) NOT NULL  
);
```

courses	
* course_id	
course_name	

course_id adalah kolom primary key dari tabel courses. Setiap tabel memiliki satu dan hanya satu primary key yang secara unik mengidentifikasi setiap baris dalam tabel. Menentukan primary key adalah praktik yang baik untuk setiap tabel.

Tipe data dari course_id adalah integer yang ditandai dengan kata kunci INT. Selain itu, nilai kolom course_id diatur sebagai AUTO_INCREMENT. Ini berarti ketika Anda menyisipkan baris baru ke dalam tabel courses tanpa memberikan nilai untuk kolom course_id, sistem database akan menghasilkan nilai integer untuk kolom tersebut.

course_name menyimpan nama-nama kursus. Tipe data kolom ini adalah string karakter (VARCHAR) dengan panjang maksimum 50. NOT NULL memastikan bahwa tidak ada nilai NULL disimpan dalam kolom course_name.

Perintah DROP TABLE

Untuk menghapus tabel yang sudah ada, Anda menentukan nama tabel setelah klausa DROP TABLE. Jika tabel yang dihapus tidak ada, sistem database akan mengeluarkan pesan error.

Berikut adalah contoh sintaks dari pernyataan DROP TABLE:

```
DROP TABLE [IF EXISTS] table_name;
```

Memahami perintah DML

Data Manipulation Language (DML) adalah perintah SQL untuk manipulasi data dalam table

Perintah DML antara lain

- INSERT -> menambah record di database
- UPDATE -> mengubah record di database
- DELETE -> menghapus record di database

SQL INSERT statement

Untuk menyisipkan satu baris ke dalam sebuah tabel, Anda menggunakan sintaks berikut dari pernyataan INSERT.

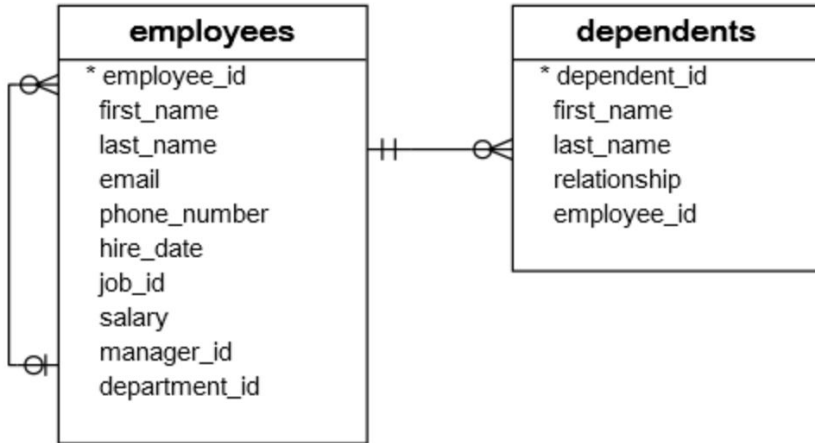
```
INSERT INTO table1  
VALUES  
    (value1, value2,...);
```

```
INSERT INTO table1 (column1, column2,...)  
VALUES  
    (value1, value2,...);
```

```
INSERT INTO (column1, column3)  
VALUES  
    (column1, column3);
```

Contoh SQL INSERT statement

Untuk menyisipkan baris baru ke tabel “dependents”.



```
INSERT INTO dependents (  
    first_name,  
    last_name,  
    relationship,  
    employee_id  
)  
VALUES  
    (  
        'Dustin',  
        'Johnson',  
        'Child',  
        178  
    );
```


SQL Multiple INSERT statement

Untuk menyisipkan banyak baris menggunakan satu pernyataan INSERT, Anda menggunakan konstruk berikut:

```
INSERT INTO table1
VALUES
    (value1, value2,...),
    (value1, value2,...),
    (value1, value2,...),
    ...;
```

Contoh SQL Multiple INSERT statement

Misalnya, untuk menyisipkan dua baris ke tabel tanggungan, Anda menggunakan query berikut.

```
INSERT INTO dependents (  
    first_name,  
    last_name,  
    relationship,  
    employee_id  
)  
VALUES  
    (  
        'Cameron',  
        'Bell',  
        'Child',  
        192  
    ),  
    (  
        'Michelle',  
        'Bell',  
        'Child',  
        192  
    );
```

SQL INSERT dari select statement

Kita dapat menggunakan pernyataan INSERT untuk query data dari satu atau beberapa tabel dan menyisipkannya ke dalam tabel lain sebagai berikut:

```
INSERT INTO table1 (column1, column2)
SELECT
    column1,
    column2
FROM
    table2
WHERE
    condition1;
```

Contoh SQL INSERT dari select statement

Misalnya, Kita memiliki tabel bernama dependents_archive yang memiliki struktur yang sama dengan tabel dependen. Pernyataan berikut menyalin semua baris dari tabel dependen ke tabel dependents_archive.

```
INSERT INTO dependents_archive
SELECT
    *
FROM
    dependents;
```

UPDATE statement

Untuk mengubah data yang sudah ada dalam sebuah tabel, Anda menggunakan pernyataan UPDATE. Berikut adalah sintaks dari pernyataan UPDATE:

```
UPDATE table_name
SET column1 = value1,
    column2 = value2
WHERE
    condition;
```

Dalam sintaks di atas:

Pertama, tentukan tabel yang ingin Anda perbarui dalam klausa UPDATE.

Kedua, tentukan kolom-kolom yang ingin Anda ubah nilainya dalam klausa SET. Kolom-kolom yang tidak terdaftar dalam klausa SET akan mempertahankan nilai-nilai aslinya.

Ketiga, tentukan baris-baris yang ingin Anda perbarui dalam klausa WHERE.

Contoh UPDATE statement

Kita akan menggunakan tabel employee dan dependents untuk menunjukkan pernyataan UPDATE.

Misalkan id karyawan 192 Sarah Bell mengubah nama belakangnya dari Bell menjadi Lopez dan Anda perlu memperbarui catatannya di tabel employee.

	employee_id	first_name	last_name
▶	192	Sarah	Bell

	employee_id	first_name	last_name
	192	Sarah	Lopez

```
UPDATE employees
SET
    last_name = 'Lopez'
WHERE
    employee_id = 192;
```

DELETE Syntax

Untuk menghapus satu atau lebih baris dari sebuah tabel, Anda menggunakan pernyataan DELETE. Sintaks umum dari pernyataan DELETE adalah sebagai berikut:

```
DELETE
FROM
    table_name
WHERE
    condition;
```

Pertama, berikan nama tabel di mana Anda ingin menghapus baris-baris.

Kedua, tentukan kondisi dalam klausa WHERE untuk mengidentifikasi baris-baris yang perlu dihapus. Jika Anda mengabaikan klausa WHERE, semua baris dalam tabel akan dihapus. Oleh karena itu, Anda harus selalu menggunakan pernyataan DELETE dengan hati-hati.

Contoh DELETE Syntax

Kita akan menggunakan tabel karyawan dan tanggungan untuk mendemonstrasikan pernyataan DELETE.

Misalkan David, yang memiliki id karyawan 105, ingin menghapus Fred dari daftar dependents. Kita tahu bahwa Fred memiliki id dependen 16, jadi kita gunakan pernyataan DELETE berikut untuk menghapus Fred dari tabel dependents.

```
DELETE FROM dependents  
WHERE  
    dependent_id = 16;
```


SELECT Statement

Perintah SQL SELECT memilih data dari satu atau lebih tabel. Berikut adalah sintaks dasar pernyataan SELECT yang memilih data dari satu tabel:

```
SELECT
    select_list
FROM
    table_name;
```

Dalam sintaks ini:

Pertama, tentukan daftar kolom yang dipisahkan oleh koma dari tabel dalam klausa SELECT.

Kemudian, tentukan nama tabel dalam klausa FROM.

Ketika mengevaluasi pernyataan SELECT, sistem database mengevaluasi klausa FROM terlebih dahulu dan kemudian klausa SELECT. Ini seperti dari sebuah tabel, pilih data dari kolom-kolom ini.

Titik koma (;) bukan bagian dari query. Server database menggunakannya untuk memisahkan dua pernyataan SQL.

SQL SELECT statement examples

Kita akan menggunakan tabel employees [dalam database contoh](#) untuk tujuan demonstrasi.

employees
* employee_id
first_name
last_name
email
phone_number
hire_date
job_id
salary
manager_id
department_id

SQL SELECT – Semua Columns

Contoh berikut menggunakan pernyataan SQL SELECT untuk mengambil data dari semua baris dan kolom dalam tabel employees:

```
SELECT * FROM table_name;
```

Query ini akan mengembalikan semua data yang ada di tabel employees, termasuk semua kolom.

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	manager_id	department_id
100	Steven	King	steven.king@sqltutorial.org	515.123.4567	1987-06-17	4	24000	null	9
101	Neena	Kochhar	neena.kochhar@sqltutorial.org	515.123.4568	1989-09-21	5	17000	100	9
102	Lex	De Haan	lex.de haan@sqltutorial.org	515.123.4569	1993-01-13	5	17000	100	9
103	Alexander	Russell	alexander.russell@sqltutorial.org	515.123.4567	1989-01-02	6	16000	100	6

QUIZ - soal

Berdasarkan use case pada <https://www.sqltutorial.org/sql-sample-database/>, buatlah query untuk menampilkan isi tabel employee

QUIZ – jawaban

select * from employee

Thanks!

