

PRAKTIKUM SISTEM BASIS DATA

Modul 4: SQL Data Manipulation Language (DML)

Tujuan dan Capaian Pembelajaran Praktikum:

1. Mahasiswa dapat menjelaskan konsep Data Manipulation Language, yang berupa pernyataan INSERT, UPDATE dan DELETE terhadap isi dari suatu tabel.
2. Mahasiswa dapat menambahkan, menghapus dan mengubah satu baris atau lebih dari suatu tabel dengan atau tanpa kondisi tertentu.

INSTRUKSI UMUM: Bacalah dahulu modul ini sebelum praktikum, sehingga ketika praktikum bisa langsung mencoba latihan dan menjelaskan variasi kasus yang mungkin muncul.

Alat dan Bahan:

1. Modul Praktikum
2. Komputer
3. Software MySQL (**masih menggunakan mode CLI**)
4. Viewer LCD

Materi dan Pembagian Waktu:

Materi dan pembagian waktu pelaksanaan praktikum meliputi:

No	Materi	Waktu
1	Pretest	10 menit
2	DML INSERT	15 menit
3	DML UPDATE	20 menit
4	DML DELETE	20 menit
5	Latihan Mandiri	40 menit
6	Posttest	70 menit
Total		170 menit

Materi Praktikum

Seperti yang sudah kita pelajari sebelumnya, CRUD memegang peranan penting dalam basis data. Apa yang akan kita pelajari di modul ini merupakan bentuk nyata dari CRUD yang diimplementasikan dalam bahasa SQL.

1. Pengertian DML INSERT

INSERT adalah statemen yang digunakan untuk menambahkan satu baris atau lebih ke dalam satu tabel. Sintaks utamanya adalah sebagai berikut:

```

INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
  [INTO] tbl_name
  [PARTITION (partition_name,...)]
  [(col_name,...)]
  {VALUES | VALUE} ({expr | DEFAULT},...),(...),...
  [ ON DUPLICATE KEY UPDATE
    col_name=expr
    [, col_name=expr] ... ]

```

Sintaks sederhana yang standar adalah:

```

INSERT INTO nama_tabel (kolom1, kolom2, ...) VALUES ('nilai1', 'nilai2 ', ...);

```

Kita bisa langsung menuliskan INSERT INTO nama_tabel VALUES (...) apabila nilai-nilai yang kita masukkan sesuai dengan urutan kolom dari tabel yang dituju. Sebaliknya, bila kita hanya akan mengisi kolom tertentu saja (dan membiarkan kolom yang tidak disebutkan agar bernilai DEFAULT atau NULL) maka kita bisa menyebutkan kolom yang diinginkan setelah nama tabel dan sebelum klausa VALUES. Berikut ini beberapa contoh cara penggunaan sintaks INSERT:

CARA INSERT 1

```

INSERT INTO dosen (kodeDosen, namaDosen, prodi) VALUES (1, 'Agata', 'Informatika');

```

Apabila kolom kodeDosen adalah AUTO_INCREMENT maka bisa ditulis:

```

INSERT INTO dosen (namaDosen, prodi) VALUES ('Agata', 'Informatika');

```

CARA INSERT 2

```

INSERT INTO dosen VALUES (1, 'Agata', 'Informatika');

```

CARA INSERT 3

```

INSERT INTO dosen VALUES
(1, 'Agata', 'Informatika'),
(2, 'Nila', 'Informatika'),
(3, 'Lukas', 'Informatika');

```

Apabila kolom kodeDosen adalah AUTO_INCREMENT maka bisa ditulis:

```

INSERT INTO (namaDosen, prodi) dosen VALUES
('Agata', 'Informatika'),
('Nila', 'Informatika'),
('Lukas', 'Informatika');

```

Klausa IGNORE digunakan untuk mengabaikan pesan kesalahan dari DBMS apabila INSERT melanggar aturan entity integrity, misalnya menambahkan record dengan kunci yang sudah ada. Dengan menggunakan klausa ini, record tetap tidak akan ditambahkan namun program tidak akan berhenti karena error. Misalnya pada tabel dosen yang sudah memiliki data berikut ini:

kodeDosen	namaDosen	prodi
DSN1	Agata	Informatika
DSN2	Nila	Informatika
DSN3	Lukas	Informatika

Dilakukan INSERT beberapa data baru berikut ini:

```
INSERT IGNORE INTO dosen VALUES
('DSN1', 'Agata', 'Informatika'),
('DSN4', 'Gloria', 'Informatika');
```

Maka data yang akan masuk hanya data DSN4 karena DSN1 sudah ada. Apabila tidak menggunakan sintaks IGNORE, maka proses INSERT akan berhenti karena error.

Klausula ON DUPLICATE KEY UPDATE digunakan apabila kita menemui kasus akan menambahkan satu baris baru namun tidak ingat apakah baris dengan primary key tersebut sudah ada di tabel atau belum. Ketika sudah ada di tabel maka agar tidak menyalahi konstrain integritas, maka tidak dilakukan penambahan baris baru namun cukup memperbarui data baris dengan primary key sesuai yang dimaksud. Di beberapa DBMS, fungsi ini digantikan oleh statemen MERGE. Contoh penggunaan dari sintaks ini dengan data dosen:

```
INSERT INTO dosen VALUES
('DSN1', 'Agata', 'Informatika')
ON DUPLICATE KEY UPDATE prodi = 'Arsitektur';
```

Kita bisa menggunakan statemen INSERT untuk memasukkan banyak baris yang berasal dari tabel lain ke tabel yang dituju dengan sintaks sebagai berikut:

```
INSERT [LOW_PRIORITY | HIGH_PRIORITY] [IGNORE]
[INTO] tbl_name
[PARTITION (partition_name,...)]
[(col_name,...)]
SELECT ...
[ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

Sintaks ini akan kita coba setelah kita belajar tentang statemen SELECT.

Latihan:

1.1. Buatlah tabel BUKU dengan struktur dan data sebagai berikut:

idBuku (CHAR, 6) PRIMARY KEY	judulBuku (VARCHAR, 50)	bahasa (CHAR, 2)	stok (INT)	tglTerbit DATE
BUKU01	Laut Bercerita	ID	5	2017-08-10
BUKU02	Thinking, Fast and Slow	EN	3	2011-08-25
BUKU03	The Bandit Queens	EN	7	NULL

Tulis SQL:

1.2. Isikan semua data di tabel *buku* dengan satu statemen INSERT.

1.3. Ujilah klausa IGNORE dengan menambahkan dua record berikut:

BUKU01	The Mountains Sing	EN	3	2020-03-17
BUKU04	Ariadne	EN	10	2021-03-30

Berapa buku yang berhasil masuk?

1.4. Ujilah klausa ON DUPLICATE dengan menambahkan dua record berikut:

BUKU03	The Bandit Queens	EN	7	2023-01-03
BUKU01	Gadis Kretek	ID	8	2012-02-23

2. Pengertian DML UPDATE

Sesuai namanya, statemen UPDATE akan memperbarui baris yang sudah ada di tabel dengan data yang kita masukkan. Sintaks UPDATE adalah sebagai berikut:

```
UPDATE [LOW_PRIORITY] [IGNORE] table_references
  SET col_name1={expr1|DEFAULT} [, col_name2={expr2|DEFAULT}] ...
  [WHERE where_condition]
  [LIMIT row_count]
```

Kita bisa memperbarui beberapa kolom sekaligus pada suatu tabel yang memenuhi kondisi tertentu. Apabila kita tidak memasukkan kondisi tertentu maka seluruh baris akan diperbarui sesuai dengan nilai yang baru dimasukkan.

Latihan:

- 2.1. Buatlah perbaikan pada tabel BUKU untuk semua buku dengan bahasa Indonesia (ID), stok diupdate menjadi 10.
- 2.2. Buatlah perbaikan dari tabel BUKU, naikan semua stok ditambah 3x untuk buku dengan bahasa Inggris (EN).
- 2.3. Buatlah perbaikan dari tabel BUKU, bagi yang tanggal terbitnya di atas tanggal 1 Januari 2023, tanggal terbit dibuat NULL.
- 2.4. Buatlah perbaikan dari tabel BUKU, untuk kode buku BUKU04 nama buku diupdate menjadi "Pulang", bahasa diganti "ID" dan tanggal terbit 1 Desember 2012 dalam satu statement UPDATE.

3. Pengertian DML DELETE

DELETE digunakan untuk menghapus baris yang ada di satu tabel, dengan atau tanpa kondisi tertentu. Sintaks DELETE adalah sebagai berikut:

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name
  [PARTITION (partition_name,...)]
  [WHERE where_condition]
  [ORDER BY ...]
  [LIMIT row_count]
```

Untuk menghapus seluruh isi tabel cukup menuliskan DELETE FROM nama_tabel. Statemen ini sama dengan TRUNCATE nama_tabel. Namun apabila kita menggunakan *transaction* (dibahas pada materi teori pertemuan XI), penggunaan DELETE FROM lebih aman karena hasil DML tidak berpengaruh secara fisik ke tabel sebelum ada perintah COMMIT. Hasil DML juga dapat ditarik ke belakang (*undo*) dengan perintah ROLLBACK.

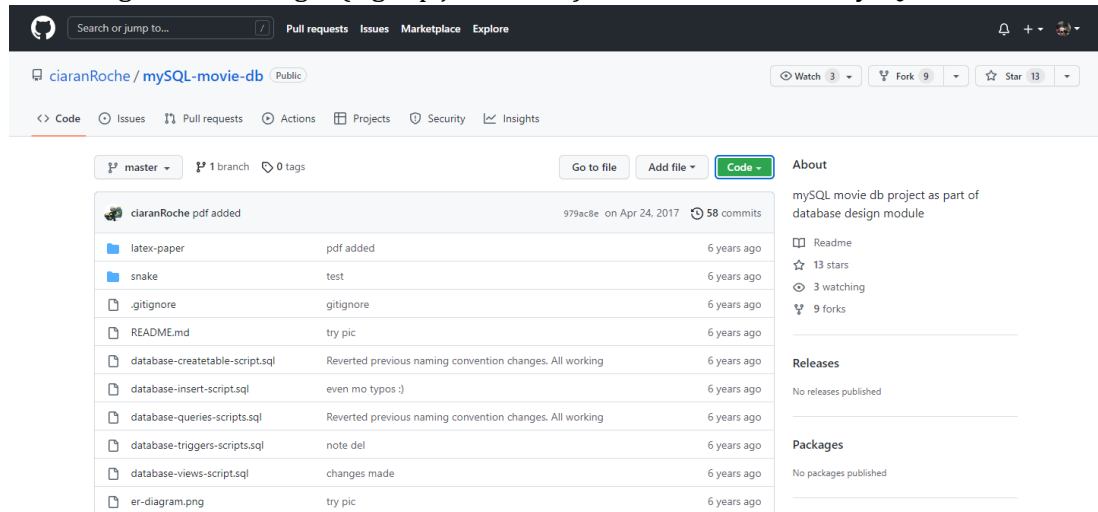
Latihan:

- 3.1. Hapuslah data buku dengan tanggal terbit NULL.
- 3.2. Hapuslah data buku dengan stok kurang dari 10.

4. Latihan Mandiri

Lakukan langkah berikut:

- i. Bukalah github.com, login (signup jika belum), kemudian carilah “mySQL-movie-db”.



- ii. Klik button Code kemudian Download ZIP.
- iii. Setelah download selesai, ekstraklah di folder pilihan Anda. Saran, ekstrak di d:\db_p4.
- iv. Buka file database-createtable-script.sql lalu beri remark (--) di baris pertama (menjadi -- drop database sequelmovie;)
- v. Kemudian buka shell console, lakukan impor untuk file database-createtable-script.sql dan file database-insert-script.sql. Cara impor adalah dengan mengetikkan perintah source [path]\[nama file.sql].

Contoh:

```
source d:\db_p4\database-createtable-script.sql
```

```
source d:\db_p4\database-insert-script.sql
```

- 4.1. Ubah skala *movieRating* menjadi sesuai yang ada di IMDb (1-10 dan bisa menampung angka di belakang koma).
- 4.2. Ubah *movieCertificate* untuk The Godfather menjadi N/A dan *movieRating* menjadi 9.2.
- 4.3. Hapus data artiste yang berasal dari German. Perhatikan mengapa Hans Zimmer belum terhapus?
- 4.4. Isikan dua film kesukaan Anda yang belum ada di tabel, masing-masing diproduksi oleh studio yang berbeda (satu dari Warner Bros, satu dari Marvel Studios).
- 4.5. Jelaskan statemen relationship antara tabel *movie* dengan tabel *movie_studio*, serta tabel *studio* dengan tabel *movie_studio*.
- 4.6. Jelaskan kegunaan tabel *movie_studio* kaitannya terhadap hubungan antara *movie* dengan *studio*.

REFERENSI:

MySQL INSERT Statement - <https://dev.mysql.com/doc/refman/5.7/en/insert.html>

MySQL UPDATE Statement - <https://dev.mysql.com/doc/refman/5.7/en/update.html>

MySQL DELETE Statement - <https://dev.mysql.com/doc/refman/5.7/en/delete.html>

MySQL SELECT Statement - <https://dev.mysql.com/doc/refman/5.7/en/select.html>

Lynn Beighley, “Head First SQL”, 2007, O’Reilly. ISBN: 978-0-596-52684-9

CiaranRoche, “Database Design Module”, <https://github.com/ciaranRoche/mySQL-movie-db>