

MODUL V

Tabel dan Data Manipulation Language (DML)

Part 3

Tujuan :

Setelah menyelesaikan modul ini, anda diharapkan dapat :

1. Membuat database dan tabel dengan menggunakan *primary key*.
2. Mampu memodifikasi tabel dengan menggunakan *query-query* yang dimiliki DML.

Dasar Teori

DML adalah sebuah metode Query yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari Query DML ini untuk melakukan pemanipulasian database yang telah dibuat. Query yang dimiliki DML adalah INSERT, UPDATE, DELETE.

1.2 Data Manipulation Language (DML)

a. Memasukkan Data dengan INSERT

Untuk memasukkan data ke dalam tabel dapat dilakukan dengan menggunakan perintah INSERT. Aturan penulisan/*syntax* dalam penulisan perintah INSERT adalah sebagai berikut :

```
INSERT INTO <tablename> values (value1,value2,....)
```

Bentuk insert seperti di atas dapat digunakan ketika jumlah value sudah sama dengan jumlah kolom yang ada dalam tabel. Contoh sebagai berikut :

```
mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| first_name | varchar(15)   | YES  |     | NULL    |                |
| last_name  | varchar(15)   | YES  |     | NULL    |                |
| start_date | date          | YES  |     | NULL    |                |
| end_date   | date          | YES  |     | NULL    |                |
| salary     | float(8,2)    | YES  |     | NULL    |                |
| city       | varchar(10)   | YES  |     | NULL    |                |
| description | varchar(15)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.08 sec)
```

Tabel employee memiliki 8 rows, kemudian akan diisi data pada tabel tersebut, dengan 8 nilai sesuai dengan jumlah kolom. Hasilnya sebagai berikut :

```
mysql> insert into employee values(
-> 9,
-> 'James',
-> 'Bond',
-> '1982-04-21',
-> '2002-09-23',
-> 1234.56,
-> 'London',
-> 'Spy',
-> );
Query OK, 1 row affected (0.03 sec)
```

Selain menggunakan perintah INSERT di atas, kita juga dapat menggunakan perintah INSERT seperti berikut ini :

```
mysql> insert into employee
-> (first_name, last_name, start_date, end_date, salary, city, description)
-> values
-> ('Hercules',
-> 'Pairot',
-> '1973-05-23',
-> '2001-08-09',
-> 4321.98,
-> 'Brussels',
-> 'Detective'
-> );
Query OK, 1 row affected (0.06 sec)

mysql> select * from employee;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | start_date | end_date | salary | city | description |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1 | alison | martin | 1996-07-25 | 2006-07-25 | 1234.56 | toronto | programmer |
| 2 | alison | mathews | 1976-03-21 | 1986-02-21 | 6661.78 | vancouver | tester |
| 3 | james | smith | 1978-12-12 | 1990-03-15 | 6544.78 | vancouver | tester |
| 4 | celia | rice | 1982-10-24 | 1999-04-21 | 2344.78 | vancouver | manager |
| 9 | James | Bond | 1982-04-21 | 2002-09-23 | 1234.56 | London | Spy |
| 10 | Hercules | Pairot | 1973-05-23 | 2001-08-09 | 4321.98 | Brussels | Detective |
+----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Menggunakan perintah di atas, maka kita dapat menentukan kolom mana saja yang diisi. Kolom yang tidak diisi, secara otomatis akan diisi dengan menggunakan nilai default yang telah dispesifikasikan saat pembuatan tabel. Kolom id dispesifikasikan AUTO INCREMENT, sehingga kolom tersebut akan diisi dengan increment dari nilai sebelumnya.

b. Memasukkan lebih dari satu baris dalam satu INSERT

Memasukkan data lebih dari satu baris dapat dilakukan dengan menggunakan satu perintah INSERT. Hal ini dilakukan dengan memisahkan input tiap baris dengan tanda koma (,). Berikut ini adalah contohnya :

```
mysql> insert into employee (first_name, last_name, start_date, end_date, salary, city, description) values
-> ('Lincoln', 'Rhyme', '1999-05-25', '2011-07-13', 3212.98, 'New York', 'Forensics'),
-> ('Sherlock', 'Holmes', '1923-08-12', '1945-07-21', 4123.21, 'London', 'Detective');
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select * from employee;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | start_date | end_date | salary | city | description |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1 | alison | martin | 1996-07-25 | 2006-07-25 | 1234.56 | toronto | programmer |
| 2 | alison | mathews | 1976-03-21 | 1986-02-21 | 6661.78 | vancouver | tester |
| 3 | james | smith | 1978-12-12 | 1990-03-15 | 6544.78 | vancouver | tester |
| 4 | celia | rice | 1982-10-24 | 1999-04-21 | 2344.78 | vancouver | manager |
| 9 | James | Bond | 1982-04-21 | 2002-09-23 | 1234.56 | London | Spy |
| 10 | Hercules | Pairot | 1973-05-23 | 2001-08-09 | 4321.98 | Brussels | Detective |
| 11 | Lincoln | Rhyme | 1999-05-25 | 2011-07-13 | 3212.98 | New York | Forensics |
| 12 | Sherlock | Holmes | 1923-08-12 | 1945-07-21 | 4123.21 | London | Detective |
+----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

c. INSERT menggunakan data dari tabel lain

Untuk melakukan INSERT, kita juga dapat mengambil data dari tabel lain, yaitu dengan menggunakan perintah INSERT....INTO...SELECT. Berikut ini adalah contohnya :

```
mysql> create table employee2(
-> id int auto_increment primary key,
-> name varchar(15),
-> city varchar(20)
-> );
Query OK, 0 rows affected (0.06 sec)

mysql> desc employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)       | NO   | PRI | NULL    | auto_increment |
| first_name | varchar(15)   | YES  |     | NULL    |                |
| last_name  | varchar(15)   | YES  |     | NULL    |                |
| start_date | date          | YES  |     | NULL    |                |
| end_date   | date          | YES  |     | NULL    |                |
| salary     | float(8,2)    | YES  |     | NULL    |                |
| city       | varchar(10)   | YES  |     | NULL    |                |
| description| varchar(15)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.02 sec)

mysql> INSERT INTO employee2
-> (id, name, city)
-> SELECT id, CONCAT (first_name, ' ', last_name), city
-> FROM employee;
Query OK, 8 rows affected (0.05 sec)
Records: 8  Duplicates: 0  Warnings: 0

mysql> select * from employee2;
+----+-----+-----+
| id | name           | city      |
+----+-----+-----+
| 1  | alison martin  | toronto   |
| 2  | alison mathews | vancouver |
| 3  | james smith    | vancouver |
| 4  | celia rice     | vancouver |
| 9  | James Bond     | London    |
| 10 | Hercules Poirot | Brussels  |
| 11 | Lincoln Rhyme  | New York  |
| 12 | Sherlock Holmes | London    |
+----+-----+-----+
8 rows in set (0.00 sec)
```

Pada tampilan di atas, dibuat tabel 'employee2' yang memiliki *field* (id, name, city). Untuk mengisi tabel employee2, akan mengambil data dari tabel employee. Untuk kolom 'name' pada tabel employee2, merupakan gabungan antara *field* 'first_name dan last_name', digabungkan dengan menggunakan perintah CONCAT.

d. Mengubah data dengan UPDATE

Data yang telah kita masukkan ke dalam tabel dapat kita ubah menggunakan perintah UPDATE. Bentuk umum perintah UPDATE adalah sebagai berikut :

```
UPDATE <table_name>
SET <column_name>='new_value'
WHERE (<column name> = 'some value');
```

Berikut ini adalah contohnya :

```
mysql> select * from employee2;
+----+-----+-----+
| id | name       | city      |
+----+-----+-----+
| 1  | alison martin | toronto   |
| 2  | alison mathews | vancouver |
| 3  | james smith   | vancouver |
| 4  | celia rice    | vancouver |
| 9  | James Bond    | London    |
| 10 | Hercules Poirot | Brussels  |
| 11 | Lincoln Rhyme  | New York  |
| 12 | Sherlock Holmes | London    |
+----+-----+-----+
8 rows in set (0.00 sec)

mysql> UPDATE employee2
-> SET city='Ottawa'
-> WHERE city='Vancouver';
Query OK, 3 rows affected (0.05 sec)
Rows matched: 3 Changed: 3 Warnings: 0

mysql> select * from employee2;
+----+-----+-----+
| id | name       | city      |
+----+-----+-----+
| 1  | alison martin | toronto   |
| 2  | alison mathews | Ottawa    |
| 3  | james smith   | Ottawa    |
| 4  | celia rice    | Ottawa    |
| 9  | James Bond    | London    |
| 10 | Hercules Poirot | Brussels  |
| 11 | Lincoln Rhyme  | New York  |
| 12 | Sherlock Holmes | London    |
+----+-----+-----+
8 rows in set (0.00 sec)
```

Pada contoh di atas dapat dilihat bahwa terdapat 3 baris data city='vancouver' berubah menjadi 'Ottawa'.

e. Mengubah nilai berdasarkan nilai sekarang

Kita dapat mengubah nilai yang dimiliki dengan menggunakan perintah UPDATE. Berikut ini adalah contohnya :

```
mysql> select salary from employee;
+-----+
| salary |
+-----+
| 123456 |
| 666178 |
| 65478  |
| 23478  |
| 123456 |
| 432198 |
| 321298 |
| 412321 |
+-----+
8 rows in set (0.00 sec)

mysql> UPDATE employee SET salary = salary + 1;
Query OK, 8 rows affected (0.01 sec)
Rows matched: 8 Changed: 8 Warnings: 0

mysql> select salary from employee;
+-----+
| salary |
+-----+
| 123457 |
| 666179 |
| 65479  |
| 23479  |
| 123457 |
| 432199 |
| 321299 |
| 412322 |
+-----+
8 rows in set (0.00 sec)
```

f. Mengganti data dari tabel

Data yang telah dimasukkan dalam tabel dapat kita ganti dengan menggunakan perintah REPLACE. Format penulisan perintah REPLACE sama dengan INSERT, namun

perintah ini akan menggantikan data yang memiliki nilai sama pada indeks PRIMARY KEY atau UNIQUE. Berikut ini adalah contohnya :

```
mysql> select * from employee2;
+----+-----+-----+
| id | name       | city    |
+----+-----+-----+
| 1  | alison martin | toronto |
| 2  | alison mathews | Ottawa |
| 3  | james smith  | Ottawa |
| 4  | celia rice   | Ottawa |
| 9  | James Bond   | London  |
| 10 | Hercules Poirot | Brussels |
| 11 | Lincoln Rhyme | New York |
| 12 | Sherlock Holmes | London  |
+----+-----+-----+
8 rows in set (0.00 sec)
```

```
mysql> REPLACE INTO employee2 (id,name,city) VALUES (12,'sHERLOCK hOLMES','Manchester');
Query OK, 2 rows affected (0.05 sec)
```

```
mysql> select * from employee2;
+----+-----+-----+
| id | name       | city    |
+----+-----+-----+
| 1  | alison martin | toronto |
| 2  | alison mathews | Ottawa |
| 3  | james smith  | Ottawa |
| 4  | celia rice   | Ottawa |
| 9  | James Bond   | London  |
| 10 | Hercules Poirot | Brussels |
| 11 | Lincoln Rhyme | New York |
| 12 | sHERLOCK hOLMES | Manchester |
+----+-----+-----+
8 rows in set (0.00 sec)
```

g. Menghapus data dari tabel

Menghapus data tertentu dari sebuah tabel dalam database dapat dilakukan menggunakan perintah DELETE. Berikut ini adalah aturan penulisannya :

```
DELETE FROM <table_name>
WHERE (<column_name> = 'some_value');
```

Berikut ini adalah contoh menghapus data dengan menggunakan DELETE :

```
mysql> SELECT * FROM employee2;
+----+-----+-----+
| id | name       | city    |
+----+-----+-----+
| 1  | alison martin | toronto |
| 2  | alison mathews | Ottawa |
| 3  | james smith  | Ottawa |
| 4  | celia rice   | Ottawa |
| 9  | James Bond   | London  |
| 10 | Hercules Poirot | Brussels |
| 11 | Lincoln Rhyme | New York |
| 12 | sHERLOCK hOLMES | Manchester |
+----+-----+-----+
8 rows in set (0.00 sec)
```

Setelah dilakukan DELETE, maka hasilnya adalah sebagai berikut :

```
mysql> select * from employee2;
+----+-----+-----+
| id | name       | city    |
+----+-----+-----+
| 1  | alison martin | toronto |
| 2  | alison mathews | Ottawa |
| 3  | james smith  | Ottawa |
| 4  | celia rice   | Ottawa |
| 9  | James Bond   | London  |
| 10 | Hercules Poirot | Brussels |
| 12 | sHERLOCK hOLMES | Manchester |
+----+-----+-----+
7 rows in set (0.00 sec)
```

Maka data dengan city = 'New York' sudah tidak ada pada tabel.

Praktikum

1. Cek service MySQL pada komputer yang Anda gunakan, pastikan service MySQL pada komputer yang Anda gunakan berjalan. Buat database baru dengan nama modul_5.
2. Buat tabel baru dengan ketentuan sesuai dengan Gambar 1, Gambar 2 dan Gambar 3.
3. Buat tabel sementara sesuai dengan ketentuan pada Gambar . Gunakan temp_ diikuti dengan nama tabel sementara yang dibuat. Contoh : temp_dokter.
4. Tambahkan kolom dengan nama jenis_kelamin pada tabel Pasien. Kolom baru yang Anda buat, Anda letakkan diantara kolom nama_pasien dengan kolom alamat_pasien. Kolom tersebut bertipe data Char dengan panjang karakter sebanyak 3.
5. Tambahkan kolom dengan nama no_telepon pada tabel Dokter. Kolom baru tersebut Anda letakkan setelah kolom alamat_dokter. Kolom baru tersebut bertipe data Varchar dengan panjang karakter sebanyak 15.
6. Ubah nama kolom alamat_dokter pada tabel Dokter menjadi almt_dokter. Nama kolom yang diubah tersebut bertipe data sama dengan sebelumnya, hanya saja kolom tersebut bersifat tidak wajib diisi.
7. Ubah kolom id_dokter pada tabel Dokter, kolom id_pasien pada tabel Pasien, dan kolom id_periksa pada tabel Periksa menjadi kolom sebagai kunci utama.
8. Ubah kolom id_pasien pada tabel Pasien menjadi kolom dengan kunci asing yang merujuk pada tabel Pasien kolom id_pasien.
9. Ubah nama tabel pada tabel Dokter menjadi tabel Tenaga_Medis.

Field	Type	Null	Key	Default	Extra
id_dokter	varchar(6)	NO		NULL	
nama_dokter	varchar(25)	NO		NULL	
alamat_dokter	varchar(35)	NO		NULL	

Gambar 1. Tabel Dokter

Field	Type	Null	Key	Default	Extra
id_pasien	varchar(9)	NO		NULL	
nama_pasien	varchar(25)	NO		NULL	
alamat_pasien	varchar(35)	NO		NULL	

Gambar 2. Tabel Pasien

Field	Type	Null	Key	Default	Extra
id_periksa	varchar(7)	NO		NULL	
tgl_periksa	date	NO		NULL	
id_pasien	varchar(9)	NO		NULL	

Gambar 3. Tabel Periksa

Tugas

Buatlah dokumentasi hasil praktikum yang anda lakukan dengan menggunakan note atau catatan atau dengan perintah \T pada *command line* anda, letakkan pada drive 'd:\namaanda\note\tgsmodul5.doc'. gunakan komentar untuk mempermudah memberikan catatan pada setiap kelompok perintah dengan menggunakan perintah # atau --.