

MODUL IV

Tabel dan Data Manipulation Language (DML)

Part 2

Tujuan :

Setelah menyelesaikan modul ini, anda diharapkan dapat :

1. Membuat database dan tabel dengan menggunakan *primary key*.
2. Mampu memodifikasi tabel.

Dasar Teori

Dalam membuat sebuah database, kita akan menemukan sebuah record yang data nya tidak boleh sama dengan record yang lain. Agar data tidak kembar maka harus membuat sebuah kolom yang di deklarasikan sebagai kunci primer (*primary key*). *Primary key* adalah nilai yang menjadi pembeda antara satu *record* data dengan *record* data yang lainnya. Untuk memastikan dalam sebuah tabel agar tidak memiliki data yang sama, maka kita dapat menggunakan *primary key*. *Primary key* hanya diperbolehkan dibuat satu kunci.

1.1 Tabel

Tabel adalah obyek utama yang harus ada pada sebuah basis data karena di dalamnya semua data akan disimpan. Tabel terletak pada sebuah database, sehingga pembuatan tabel dilakukan setelah sebuah database telah dibuat. Dalam tabel terdapat baris dan kolom. Baris diistilahkan dengan recordset dan kolom dengan field.

a. Membuat Tabel dengan FOREIGN KEY

Mysql memiliki beberapa engine, antara lain adalah InnoDB. Pembuatan tabel dengan menggunakan InnoDB dapat membantu kita dalam melakukan *referential integrity*. *Foreign key* merupakan *key* yang merupakan *field* dari tabel yang lain, yang dijadikan *primary key* pada tabel lain. Contoh pembuatan *foreign key* dapat dilihat pada tampilan di bawah ini :

```
mysql> create table models(  
-> modelid smallint not null auto_increment,  
-> name varchar(40) not null,  
-> primary key (modelid)  
-> )  
-> engine=InnoDB;  
Query OK, 0 rows affected (0.11 sec)  
  
mysql> desc models;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| modelid | smallint(6) | NO | PRI | NULL | auto_increment |  
| name | varchar(40) | NO | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
2 rows in set (0.01 sec)
```

Pada tampilan di atas, kita membuat tabel 'models' dengan 'modelid' sebagai *primary key*. Kemudian kita akan membuat tabel lain dengan nama 'orders' yang memiliki *primary key* 'id' dan *foreign key* 'modelid' yang merupakan *primary key* pada tabel 'models', hasil dapat dilihat pada tampilan berikut ini :

```
mysql> create table orders(  
-> id smallint not null primary key,  
-> modelid smallint not null,  
-> description varchar(40),  
-> foreign key (modelid) references models (modelid) on update cascade on delete cascade  
-> ) engine=InnoDB;  
Query OK, 0 rows affected (0.16 sec)  
  
mysql> desc orders;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key  | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| id         | smallint(6)   | NO   | PRI  | NULL    |       |  
| modelid    | smallint(6)   | NO   | MUL  | NULL    |       |  
| description | varchar(40)   | YES  |      | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

Pada tampilan di atas dapat kita lihat bahwa tabel 'orders' memiliki *primary key* 'id' dan memiliki *foreign key* 'modelid' yang memiliki relasi dengan *primary key* pada tabel 'models'. Untuk lebih jelas, maka kita akan mengisi tabel tersebut dengan data sebagai berikut :

```
mysql> insert into models values (1, 'testsatu');  
Query OK, 1 row affected (0.05 sec)  
  
mysql> select * from models;  
+-----+-----+  
| modelid | name      |  
+-----+-----+  
| 1       | testsatu  |  
+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> insert into orders values (1, 1, 'deskripsisatu');  
Query OK, 1 row affected (0.05 sec)  
  
mysql> select * from orders;  
+-----+-----+-----+  
| id | modelid | description |  
+-----+-----+-----+  
| 1  | 1       | deskripsisatu |  
+-----+-----+-----+  
1 row in set (0.00 sec)
```

Pada perintah sql di atas, kita memasukkan data ke tabel models dan orders. Data yang dimasukkan pada tabel orders memiliki referensi terhadap *primary key* pada tabel models, yaitu modelid yang memiliki nilai 1. Untuk melihat keterkaitan antara tabel models dan tabel orders, maka kita akan lakukan hapus data pada tabel models, sebagai berikut :

```
mysql> delete from models where modelid=1;  
Query OK, 1 row affected (0.05 sec)  
  
mysql> select * from models;  
Empty set (0.00 sec)
```

Data dengan modelid = 1 terhapus, maka tabel models kosong/tidak ada data. Karena tabel orders memiliki *foreign key* 'modelid', maka data pada tabel orders yang memiliki

modelid=1 juga akan terhapus/seharusnya tabel orders juga kosong/tidak ada data, seperti pada tampilan berikut :

```
mysql> select * from models;
Empty set (0.00 sec)

mysql> select * from orders;
Empty set (0.00 sec)
```

b. Memodifikasi Tabel dengan ALTER TABLE

Tabel yang telah dibuat dapat dimodifikasi dengan menggunakan perintah ALTER TABLE.

1. Menamhakan Kolom pada Tabel

Salah satu cara modifikasi tabel adalah dengan menambahkan kolom pada tabel yang sebelumnya telah dibuat sebagai berikut :

- Buat tabel baru, pada contoh ini diberi nama 'table5' sebagai berikut :

```
mysql> create table table5(
-> id int
-> );
Query OK, 0 rows affected (0.08 sec)

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

- Sisipkan *field* atau kolom 'name' pada table5.

```
mysql> alter table table5
-> add name varchar(5);
Query OK, 0 rows affected (0.30 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id    | int(11)    | YES  |     | NULL    |       |
| name  | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

- Tambahkan *field* atau kolom 'first_name' setelah *field* 'id', sebagai berikut :

```
mysql> alter table table5
-> add first_name varchar(30) after id;
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)    | YES  |     | NULL    |       |
| first_name | varchar(30) | YES  |     | NULL    |       |
| name       | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

2. Memodifikasi Nama Kolom pada Tabel

Kita akan memodifikasi/mengubah *field* 'name' dengan 'last_name' dengan menggunakan perintah berikut ini :

```
mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | YES  |     | NULL    |       |
| first_name | varchar(30)| YES  |     | NULL    |       |
| name       | varchar(5) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

mysql> alter table table5
-> change name last_name varchar(30);
Query OK, 0 rows affected (0.19 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | YES  |     | NULL    |       |
| first_name | varchar(30)| YES  |     | NULL    |       |
| last_name  | varchar(30)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Dapat dilihat bahwa *field* 'name' sudah berubah menjadi 'last_name'.

3. Memodifikasi Definisi Tabel

Kita dapat mengubah definisi tabel, antara lain adalah *length* tabel, dengan menggunakan perintah sebagai berikut :

```
mysql> alter table table5
-> modify first_name varchar(15),
-> modify last_name varchar(15);
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | YES  |     | NULL    |       |
| first_name | varchar(15)| YES  |     | NULL    |       |
| last_name  | varchar(15)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Sebelumnya first_name dan last_name memiliki *length* (30), dilakukan modifikasi menjadi (15) dengan menggunakan perintah ALTER TABLE ...MODIFY.

4. Menambah Primary Key

Kita dapat menambah *primary key* dari tabel sebelumnya yang tidak memiliki *primary key*, dengan perintah sebagai berikut :

```
mysql> alter table table5
-> add primary key (id);
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | 0        |       |
| first_name | varchar(15)| YES  |     | NULL     |       |
| last_name  | varchar(15)| YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Setelah diubah, maka *field* 'id' menjadi *primary key*, dengan menggunakan perintah ALTER TABLE.

5. Menghapus Kolom

Selain menambahkan kolom, kita juga dapat menghapus kolom, dengan menggunakan perintah sebagai berikut :

```
mysql> alter table table5
-> drop first_name;
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc table5;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id         | int(11)   | NO   | PRI | 0        |       |
| last_name  | varchar(15)| YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Dengan menggunakan ALTER TABLE DROP, kita dapat menghapus kolom yang kita inginkan. Pada contoh di atas yang dihapus adalah kolom first_name.

6. Mengganti Nama Tabel

Untuk mengganti nama tabel yang sudah dibuat, kita dapat melakukan dengan menggunakan perintah ALTER TABLE. RENAME, contoh sebagai berikut :

```
mysql> show tables;
+-----+
| Tables_in_modul3 |
+-----+
| cars              |
| employee          |
| employee_copy     |
| models            |
| orders            |
| table1            |
| table2            |
| table3            |
| table4            |
| table5            |
+-----+
10 rows in set (0.02 sec)

mysql> alter table employee_copy RENAME senior_employee;
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+-----+
| Tables_in_modul3 |
+-----+
| cars              |
| employee          |
| models            |
| orders            |
| senior_employee   |
| table1            |
| table2            |
| table3            |
| table4            |
| table5            |
+-----+
10 rows in set (0.00 sec)
```

Pada contoh di atas, sebelum dilakukan ubah nama tabel, terdapat tabel 'employee_copy', kemudian dilakukan RENAME menjadi 'senior_employee'. Maka setelah dilakukan rename, tabel employee_copy berubah menjadi senior_employee.

7. Menghapus Semua Data dalam Tabel

Untuk mengosongkan tabel atau menghapus semua data dalam tabel, dapat dilakukan dengan menggunakan perintah TRUNCATE. Contoh sebagai berikut :

```
mysql> create table employee_copy AS
-> select * from employee;
Query OK, 4 rows affected (0.11 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> select * from employee_copy;
+----+-----+-----+-----+-----+-----+-----+-----+
| 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 |
+----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> TRUNCATE TABLE employee_copy;
Query OK, 0 rows affected (0.05 sec)

mysql> select * from employee_copy;
Empty set (0.00 sec)
```

Pada contoh di atas, dibuat tabel employee_copy dari tabel employee. Setelah query dijalankan, maka isi tabel employee_copy sama dengan isi tabel employee. Kemudian dilakukan TRUNCATE atau hapus semua isi tabel employee-copy, maka tabel tersebut kosong.

Praktikum

Praktekkan perintah-perintah di atas, kemudia buat dokumentasi dengan menggunakan screenshot atau dengan menggunakan note atau catatan atau \T yang disimpan pada folder yang sama seperti pada praktikum modul 3.

Tugas

Dengan menggunakan *command line*,

1. Buat database dengan nama modul4.
2. Pada database tersebut buatlah tabel :
 - Mahasiswa, dengan *field* : NIM, nama, alamat, no_telp (dengan NIM sebagai *primary key*);
 - Buku, dengan *field* : id_buku, judul_buku, penerbit, tahun (id_buku sebagai *primary key*);
 - Pinjam, dengan *field* : id_pinjam, NIM, id_buku (id_pinjam sebagai *primary key*, NIM dan id_buku sebagai *foreign key*)
3. Isikan data masing-masing tabel dengan tiga baris data.
4. Buat tabel baru dengan nama 'pinjam_copy', dari tabel pinjam beserta semua datanya.
5. Ubah nama tabel 'pinjam_copy' menjadi kembali.
6. Tambahkan kolom 'tgl_kembali' pada tabel kembali.
7. Hapus semua data pada tabel kembali.
8. Isikan tabel kembali dengan tiga baris data dengan menggunakan perintah load data.
9. Tambahkan isi tabel kembali dengan dua baris data dengan perintah insert.