

MODUL 7

RELASI TABEL

I. TUJUAN

1. Mahasiswa dapat membuat relasi tabel
2. Mahasiswa mampu membuktikan integritas data dengan tabel yang saling berhubungan (relasi)

II. DASAR TEORI

Relasional Database dan RDBMS, Relational Database sebenarnya adalah salah satu konsep penyimpanan data, sebelum konsep database relasional muncul sebenarnya sudah ada dua model database yaitu Network Database dan Hierarchie Database. Dalam database relasional, data disimpan dalam bentuk relasi atau tabel dua dimensi, dan antar tabel satu dengan tabel lainnya terdapat hubungan atau relationship, sehingga sering kita baca di berbagai literatur, database didefinisikan sebagai “kumpulan dari sejumlah tabel yang saling hubungan atau keterkaitan”. Nah, kumpulan dari data yang diorganisasikan sebagai tabel tadi disimpan dalam bentuk data elektronik di dalam hardisk komputer. Untuk membuat struktur tabel, mengisi data ke tabel, mengubah data jika diperlukan dan menghapus data dari tabel diperlukan software.

Software yang digunakan membuat tabel, isi data, ubah data dan hapus data disebut Relational Database Management System atau dikenal dengan singkatan RDBMS sedangkan perintah yang digunakan untuk membuat tabel, isi, ubah dan hapus data disebut perintah SQL yang merupakan singkatan dari Structure Query Language. Jadi, setiap software RDBMS pasti bisa digunakan untuk menjalankan perintah SQL.

Sebenarnya fungsi RDBMS bukan cuma buat tabel, isi data, ubah dan hapus data, untuk manajemen data dalam skala besar dan agar bisa mendukung proses bisnis yang kontinyu dan real time. Suatu RDBMS dituntut untuk mempunyai kemampuan manajemen user dan keamanan data, backup dan recovery data serta kemampuan lainnya yang berkaitan dengan kecepatan pemrosesan data (performance).

Primary key adalah suatu field atau kombinasi field yang secara unik mengidentifikasi setiap record di dalam tabel. Contoh NIM yang terdapat pada tabel mahasiswa.

Foreign key adalah primary key pada suatu tabel yang di masukkan pada tabel lain dan dijadikan salah satu key pada tabel tersebut.

References menunjukkan bahwa foreign key di suatu tabel merupakan key pada tabel lain. Pada perubahan suatu record baik itu update atau delete, dapat diikuti dengan kata: cascade, restrict atau set null. Kita bisa atur sesuai kebutuhan.

Contoh :

- on update CASCADE on delete RESTRICT
- on update RESTRICT on delete SET NULL

III. PRAKTIKUM

1. Buat sebuah database dengan nama Akademik !
2. Buat tabel Mahasiswa dengan tipe innodb !

```
mysql> create table mahasiswa(nim varchar(20) not null,
-> nama varchar(25), primary key (nim));
Query OK, 0 rows affected (0.36 sec)

mysql> desc mahasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nim   | varchar(20)   | NO   | PRI |          |       |
| nama  | varchar(25)   | YES  |     |          |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> alter table mahasiswa type=innodb;
Query OK, 0 rows affected, 1 warning (0.38 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

3. Buat tabel dosen pembimbing dengan tipe innodb dan dan relasikan ke tabel mahasiswa dengan mengambil key nim dari tabel mahasiswa sebagai foreign key di tabel dosen_pembimbing :

```
mysql> create table dosen_pembimbing(idx int(11) not null auto_increment,
-> kode_dosen varchar(5),
-> nama_dosen varchar(20),
-> nim varchar(20),
-> primary key(idx),
-> index (nim),
-> foreign key(nim) references mahasiswa(nim) on update cascade on delete ca
-> scade)type=innodb;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

4. Isi data di tabel seperti data berikut :

nim	nama
001	Ferry Gustiawa
002	Zarkasih
003	Aris
004	Obet

dan

idx	kode_dosen	nama_dosen	nim
1	D001	Deris	001
2	D002	Dian	002
3	D001	Deris	003

Note : untuk mengisi data yang fieldnya AUTO_INCREMENT maka sebutkan NULL untuk field tersebut

5. Kemudian update satu record di tabel mahasiswa :

```
mysql> update mahasiswa set nim='005' where nama='obet';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from mahasiswa;
+----+-----+
| nim | nama |
+----+-----+
| 001 | Ferry Gustiawa |
| 002 | Zarkasih |
| 003 | Aris |
| 005 | Obet |
+----+-----+
4 rows in set (0.00 sec)

mysql> select * from dosen_pembimbing;
+----+-----+-----+-----+
| idx | kode_dosen | nama_dosen | nim |
+----+-----+-----+-----+
| 1 | D001 | Deris | 001 |
| 2 | D002 | Dian | 002 |
| 3 | D001 | Deris | 003 |
| 4 | D002 | Dian | 005 |
+----+-----+-----+-----+
4 rows in set (0.00 sec)
```

6. Hapus sebuah record di tabel mahasiswa :

```
mysql> delete from mahasiswa where nim='005';
Query OK, 1 row affected (0.00 sec)

mysql> select * from mahasiswa;
+----+-----+
| nim | nama |
+----+-----+
| 001 | Ferry Gustiawa |
| 002 | Zarkasih |
| 003 | Aris |
+----+-----+
3 rows in set (0.00 sec)

mysql> select * from dosen_pembimbing;
+----+-----+-----+-----+
| idx | kode_dosen | nama_dosen | nim |
+----+-----+-----+-----+
| 1 | D001 | Deris | 001 |
| 2 | D002 | Dian | 002 |
| 3 | D001 | Deris | 003 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

IV. EVALUASI DAN PERTANYAAN

Tuliskan perintah-perintah SQL percobaan di atas beserta outputnya !

1. Tulis semua perintah-perintah SQL percobaan di atas beserta outputnya !
2. Berikan kesimpulan Anda!

Kesimpulan

