



MODUL VIII

PRAKTIKUM SISTEM BASIS DATA I

Semester Genap 2015

CACA E. SUPRIANA, S.Si.,MT. (*caca.e.supriana@unpas.ac.id*)

I. Tabel Pegawai

Untuk modul ini, gunakan kembali tabel Pegawai dari modul VII, struktur database dari tabel tersebut adalah :

```
mysql> desc pegawai;
```

| Field | Type | Null | Key | Default | Extra |
|-----------|-------------|------|-----|---------|-------|
| nip | varchar(5) | NO | PRI | NULL | |
| nama | varchar(25) | YES | | NULL | |
| alamat | varchar(30) | YES | | NULL | |
| tgl_lahir | date | YES | | NULL | |
| no_telp | varchar(15) | YES | | NULL | |
| thn_masuk | year(4) | YES | | NULL | |
| golongan | char(1) | YES | | NULL | |
| gaji | int(11) | YES | | NULL | |

```
8 rows in set (0.11 sec)
```

II. Case statement

Perintah **CASE** yang berupa pilihan-pilihan jabatan berdasarkan isi dari field golongan di tabel Pegawai (1=supervisor, 2=asisten supervisor, 3=operator dan 4=staff), pilihannya ditulis setelah '**when**' lalu ditampilkan di kolom baru dengan nama 'jabatan'. Buatlah instruksi SQL berikut :

```
mysql> select nama,
-> case golongan
-> when '1' then 'supervisor'
-> when '2' then 'asisten supervisor'
-> when '3' then 'operator'
-> when '4' then 'staff'
-> end as jabatan
-> from pegawai;
```

```

+-----+-----+
| nama          | jabatan          |
+-----+-----+
| Ahmad Burhanuddin | operator          |
| Amin Imsyorry    | asisten supervisor |
| Budhy Bungaox    | staff            |
| Zulkarnaen       | supervisor        |
| Dewi Sudewa      | supervisor        |
| Ina Nurlian      | staff            |
| Cheppy Chardut   | staff            |
| Dodong M         | operator          |
+-----+-----+
8 rows in set (0.06 sec)

```

III. If statement

Perhatikan instruksi SQL berikut, dimana setiap pegawai mengikuti asuransi yang premi (angsuran) perbulannya tergantung dari gaji. Pegawai yang gajinya kurang atau sama dengan rp. 1.250.000 harus membayar rp. 25.000 per bulan sedangkan yang gajinya lebih besar dari rp. 1.250.000 harus membayar rp. 40.000 per bulan:

```

mysql> select nip, nama, gaji,
        -> if (gaji <= 1250000, 'premi 25rb/bln', 'premi 40rb/bln')
        -> as premi_asuransi
        -> from pegawai;

```

```

+-----+-----+-----+-----+
| nip | nama          | gaji    | premi_asuransi |
+-----+-----+-----+-----+
| 001 | Ahmad Burhanuddin | 1250000 | premi 25rb/bln |
| 002 | Amin Imsyorry    | 1350000 | premi 40rb/bln |
| 003 | Budhy Bungaox    | 1050000 | premi 25rb/bln |
| 004 | Zulkarnaen       | 1450000 | premi 40rb/bln |
| 005 | Dewi Sudewa      | 1450000 | premi 40rb/bln |
| 006 | Ina Nurlian      | 1050000 | premi 25rb/bln |
| 007 | Cheppy Chardut   | 1050000 | premi 25rb/bln |
| 008 | Dodong M         | 1250000 | premi 25rb/bln |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)

```

Intruksi if dapat dipergunakan juga untuk menangani pilihan, sebaiknya dipergunakan jika pilihan tidak sebanyak pilihan dalam instruksi case seperti contoh diatas yang hanya mempunyai 2 pilihan untuk iuran_pegawai berdasarkan gaji.

IV. *Referential integrity*

Dalam sebuah klinik, seorang dokter akan melayani lebih dari satu pasien, sehingga jika akan dibuat relasi antara tabel **Dokter** dengan tabel **Pasien** maka kardinalitasnya adalah **satu ke banyak** (*one to many*). Dengan menggunakan MySQL buatlah relasi antara tabel Dokter dengan tabel Pasien. Beri nama databasenya dengan **NRP** anda !

```
mysql> create database klinik;  
Query OK, 1 row affected (0.05 sec)  
  
mysql> use klinik;  
Database changed  
mysql> create table dokter(  
    -> kode_dr char(5) primary key not null,  
    -> nama_dr char(25) not null,  
    -> spesialis char(20));  
Query OK, 0 rows affected (0.11 sec)
```

Gantilah nama
'**klinik**' menjadi
NRP anda !

Konsekuensi dari relasi antar tabel, terutama relasi antar tabel dengan kardinalitas '*one to many*' adalah dibutuhkannya *field foreign key*. *Foreign key* adalah fasilitas integritas referensial (*referential integrity*) yang dibutuhkan untuk menjaga konsistensi hubungan antar tabel. Integritas referensial akan menjamin bahwa nilai yang merujuk pada tabel lain adalah memang benar ada di tabel lain tersebut.

```
mysql> create table pasien(  
    -> no_pasien char(5) primary key not null,  
    -> nama_pasien char(25) not null,  
    -> alamat_pasien char(25),  
    -> umur int,  
    -> kode_dr char(5),  
    -> foreign key (kode_dr) references dokter(kode_dr)  
    -> on update cascade  
    -> on delete restrict)  
    -> engine=InnoDB;  
Query OK, 0 rows affected (0.06 sec)
```

Tindakan yang dapat dikenakan pada saat sebuah field di UPDATE atau di DELETE adalah :

- a. RESTRICT : update atau delete akan ditolak
- b. CASCADE : pada *update* jika nilai kunci induk diubah maka isi *foreign key* akan ikut diubah, sedangkan pada *delete* jika record dengan kunci induk dihapus maka record yang mempunyai *foreign key* yang sama dengan yang dihapus akan ikut dihapus.
- c. SET NULL : jika *record* dengan kunci induk dihapus maka *record* yang mempunyai *foreign key* yang sama dengan yang dihapus akan diisi dengan NULL.
- d. NO ACTION : sama dengan RESTRICT

Tipe basis data (*database engine*) dalam MySQL yang dapat menggunakan FOREIGN KEY adalah **InnoDB**.

V. *Insert Record*

Langkah selanjutnya adalah mengisi tabel Dokter terlebih dahulu. Agar diingat, tabel yang menjadi referensi, dalam hal ini adalah tabel Dokter harus diisi terlebih dahulu.

```
mysql> insert into dokter
-> (kode_dr, nama_dr, spesialis)
-> values
-> ('dr01','dr. Buddy','Umum'),
-> ('dr02','drg. Emilia','Gigi'),
-> ('dr03','dr. Endang','THT'),
-> ('dr04','dr. Maliq','Kandungan');
Query OK, 4 rows affected (0.05 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> select * from dokter;
```

| kode_dr | nama_dr | spesialis |
|---------|-------------|-----------|
| dr01 | dr. Buddy | Umum |
| dr02 | drg. Emilia | Gigi |
| dr03 | dr. Endang | THT |
| dr04 | dr. Maliq | Kandungan |

```
4 rows in set (0.00 sec)
```

Setelah tabel Dokter diisi lalu lanjutkan dengan mengisi record pada tabel Pasien. Pastikan isi *foreign key* kode_dr memang telah ada di tabel Dokter.

```
mysql> insert into pasien
-> (no_pasien, nama_pasien, alamat_pasien, umur, kode_dr)
-> values
-> ('p001','Abdul','Cihaji 16 Bandung',23,'dr01');
Query OK, 1 row affected (0.02 sec)
```

```
mysql> insert into pasien
-> (no_pasien, nama_pasien, alamat_pasien, umur, kode_dr)
-> values
-> ('p002','Amin','Cikaso 67 Bandung',28,'dr02'),
-> ('p003','Neneng','Cikadut 74 Bandung',25,'dr04'),
-> ('p004','Nenden','Gatot Subroto 7 Bandung',35,'dr04'),
-> ('p005','Azwar','Cijalu 5 Subang',30,'dr03'),
-> ('p006','Hanif','Cibaduyut 556 Bandung',39,'dr01'),
-> ('p007','Yusuf','A. Yani 889 Bandung',36,'dr03');
Query OK, 6 rows affected (0.01 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

```
mysql> select * from pasien;
```

| no_pasien | nama_pasien | alamat_pasien | umur | kode_dr |
|-----------|-------------|-------------------------|------|---------|
| p001 | Abdul | Cihaji 16 Bandung | 23 | dr01 |
| p002 | Amin | Cikaso 67 Bandung | 28 | dr02 |
| p003 | Neneng | Cikadut 74 Bandung | 25 | dr04 |
| p004 | Nenden | Gatot Subroto 7 Bandung | 35 | dr04 |
| p005 | Azwar | Cijalu 5 Subang | 30 | dr03 |
| p006 | Hanif | Cibaduyut 556 Bandung | 39 | dr01 |
| p007 | Yusuf | A. Yani 889 Bandung | 36 | dr03 |

```
7 rows in set (0.00 sec)
```

VI. Query SQL

Tampilkan nama dokter dan pasien yang dirawatnya, *query* ini akan menampilkan nama_dr dari tabel Dokter dan nama_pasien dari tabel Pasien. Untuk membuat query dari 2 tabel atau lebih tidak jauh berbeda dengan membuat query dari satu tabel, perbedaannya yang paling tampak tentu saja dari klausa FROM yang dalam modul sebelumnya berisi 1 tabel, dalam mencari data dari 2 tabel atau lebih klausa FROM berisi nama 2 tabel atau lebih. Perhatikan juga penulisan field pada klausa SELECT, harus diawali oleh nama tabel tempat field tersebut berada, jika field tersebut adalah primary key yang menjadi foreign key di tabel lain. *Query* cara pertama :

```
mysql> select nama_dr, nama_pasien
-> from dokter, pasien
-> where dokter.kode_dr = pasien.kode_dr;
```

```

+-----+-----+
| nama_dr      | nama_pasien |
+-----+-----+
| dr. Buddy    | Abdul       |
| dr. Buddy    | Hanif       |
| drg. Emilia  | Amin        |
| dr. Endang   | Azwar       |
| dr. Endang   | Yusuf       |
| dr. Maliq    | Neneng      |
| dr. Maliq    | Nenden      |
+-----+-----+
7 rows in set (0.06 sec)

```

Query cara kedua :

```

mysql> select nama_dr, nama_pasien
      -> from dokter natural join pasien;

```

```

+-----+-----+
| nama_dr      | nama_pasien |
+-----+-----+
| dr. Buddy    | Abdul       |
| dr. Buddy    | Hanif       |
| drg. Emilia  | Amin        |
| dr. Endang   | Azwar       |
| dr. Endang   | Yusuf       |
| dr. Maliq    | Neneng      |
| dr. Maliq    | Nenden      |
+-----+-----+
7 rows in set (0.02 sec)

```

Kedua cara *query* tersebut diatas menghasilkan output yang sama !

Tampilkan nama pasien dari dokter spesialis THT !

```

mysql> select nama_pasien
      -> from dokter, pasien
      -> where spesialis = 'THT'
      -> and dokter.kode_dr = pasien.kode_dr;

```

```

+-----+
| nama_pasien |
+-----+
| Azwar       |
| Yusuf       |
+-----+
2 rows in set (0.00 sec)

```

Cari nama pasien dari dokter spesialis kandungan dan berumur lebih dari 30 tahun !

```
mysql> select nama_pasien, umur
-> from dokter, pasien
-> where spesialis = 'Kandungan'
-> and umur > 30
-> and dokter.kode_dr = pasien.kode_dr;
```

```
+-----+-----+
| nama_pasien | umur |
+-----+-----+
| Nenden      | 35   |
+-----+-----+
1 row in set (0.00 sec)
```

VII. Latihan

1. Cari nama dokter yang merawat pasien yang namanya dimulai dengan huruf 'A'.
2. Tampilkan nama dan umur pasien yang berumur lebih dari 20 tahun dan dirawat oleh 'dr. Buddy'.
3. Cari nomor dan nama pasien yang dirawat oleh 'drg. Emilia'.
4. Tampilkan nama dokter yang merawat pasien yang beralamat di 'Cijalu'.
5. Cari seluruh data pasien yang dirawat di dokter spesialis 'kandungan'.