

MODUL PRAKTIKUM VI

Constraint II

Tujuan :

- Mampu memahami dan mengerti bermacam – macam batasan – batasan (constraint) di Database
- Mampu menggunakan dan menerapkan constraint pada SQL Server

Materi :

- FOREIGN KEY
- IDENTITY
- DEFAULT VALUE

Persiapan

- Membaca buku literature, referensi atau dari sumber lain tentang penggunaan Constraint
- Modul Praktikum VII

Landasan Teori

Pengantar

Selain constraint/batasan-batasan yang sudah dibahas pada modul sebelumnya, batasan lain yang biasa diberlakukan di dalam table antara lain :

- FOREIGN KEY
- DEFAULT VALUE
- IDENTITY

FOREIGN KEY

FOREIGN KEY atau KUNCI TAMU merupakan kunci yang digunakan sebagai penghubung antara satu tabel dengan tabel lainnya. Jika sebuah tabel memiliki FOREIGN KEY maka table tersebut dapat di sebut juga dengan tabel kedua dimana tabel utamanya adalah tabel yang memiliki kolom PRIMARY KEY dari FOREIGN KEY tersebut

CONSTRAINT fk_nama-tabel_nama-kolom FOREIGN KEY
REFERENCES table-referensi (kolom referensi)

Ke:

Modul Praktikum Sistem Basis Data I S1-SI

Fk_nama-tabel_nama-kolom = Nama FOREIGN KEY

FOREIGN KEY = Jenis constraint-nya, yaitu FOREIGN KEY

REFERENCES = Kata kunci untuk merelasikan tabel ini ke tabel utamanya (tabel yang memiliki PRIMAR KEY)

Tabel-Referensi = Tabel yang akan dijadikan referensi (tabel utama).

Kolom-Referensi = Kolom yang menjadi referensi dari tabel utama.

Ex :

```
CREATE TABLE mahasiswa
(
  nim INTEGER CONSTRAINT pk_mahasiswa_nim PRIMARY KEY,
  nip INTEGER,
  nama_mhs VARCHAR(45),
  alamat_mhs VARCHAR(255)
  CONSTRAINT fk_mahasiswa_nip
  FOREIGN KEY REFERENCES dosen(nip)
)
```

Dalam SQL Server ada beberapa pilihan yang dapat digunakan ketika FOREIGN KEY di berikan pada sebuah kolom dalam tabel, yaitu:

[ON DELETE {CASCADE | NO ACTION}]

Pilihan ini akan menentukan aksi apa yang akan di lakukan terhadap sebuah baris/ record dalam tabel, jika record tersebut memiliki hubungan referensi (referential relationship) dan record yang di referensi di hapus (DELETE) dari tabel induk. Defaultnya adalah NO ACTION. Jika pilihan CASCADE di gunakan, sebuah record akan di hapus dari tabel referensinya jika record tersebut di hapus dari tabel induknya. Jika pilihan NO ACTION yang digunakan, SQL Server akan menampilkan pesan kesalahan dan proses penghapusan record pada tabel induk di batalkan (ROLLED BACK).

Contohnya, pada table – table diatas, tabel **mahasiswa** memiliki hubungan referensi (referential relationship) dengan tabel **dosen**. **FOREIGN KEY mahasiswa.nip** memiliki hubungan referensi dengan **PRIMARY dosen.nip**.

Jika perintah **DELETE** di lakukan pada sebuah record di tabel **dosen**, dan properties **ON DELETE CASCADE** di berikan pada kolom **mahasiswa.nip**, SQL Server melakukan pengecekan terhadap satu atau lebih record yang memiliki ketergantungan di tabel **mahasiswa**. Jika ada record yang memiliki ketergantungan tersebut, maka record tersebut akan di hapus dari table **mahasiswa** sesuai dengan record yang di hapus di tabel **dosen**. Dengan kata lain, jika pilihan **NO ACTION** yang digunakan, SQL Server akan menampilkan pesan kesalahan dan membatalkan penghapusan data untuk record di tabel **dosen** tersebut jika terdapat paling tidak satu record di tabel **mahasiswa** yang berhubungan dengan record tersebut

[ON UPDATE {CASCADE | NO ACTION}]

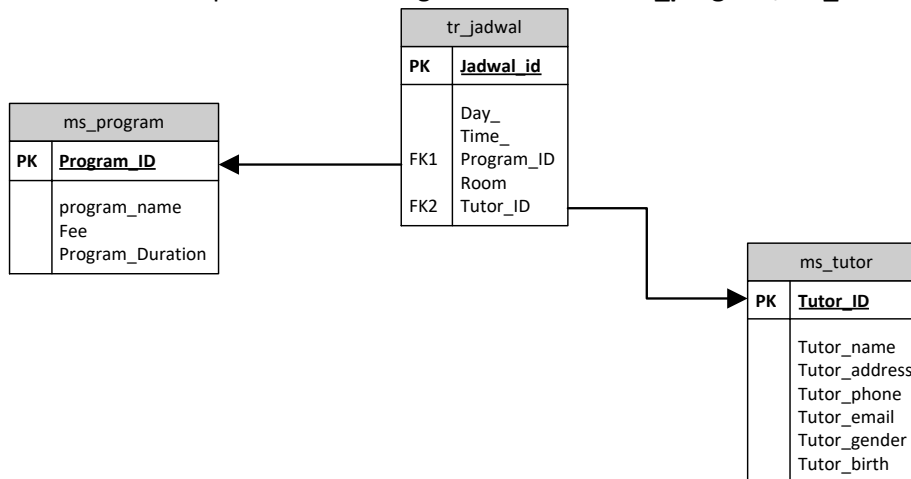
Sama dengan [ON DELETE {CASCADE | NO ACTION}], yang membedakan jika **ON DELETE** aksi dilakukan jika record di hapus (DELETE), pada **ON UPDATE** aksi dilakukan jika record di rubah (UPDATE). Berikut adalah contoh pemberian pilihan ON DELETE CASCADE dan ON UPDATE CASCADE pada sebuah tabel untuk kolom yang menjadi FOREIGN KEY

```
CREATE TABLE mahasiswa
```

```
(  
nim INTEGER CONSTRAINT pk_mahasiswa_nim PRIMARY KEY,  
nip INTEGER  
CONSTRAINT fk_mahasiswa_nim  
FOREIGN KEY REFERENCES dosen(nip)  
ON DELETE CASCADE ON UPDATE CASCADE,  
nama_mhs VARCHAR(45),  
alamat_mhs VARCHAR(255)  
)
```

Langkah – Langkah Praktikum

1. Perhatikan Konseptual atau hubungan antara table **ms_program**, **ms_tutor** dan **tr_jadwal**



Penjelasan :

- Sebuah program bisa memiliki atau ada di beberapa jadwal. Contoh, program TOEFL bisa dijadwalkan di hari senin, selasa, kamis atau hari lainnya
 - Seorang tutor bisa mengajar di beberapa jadwal. Contoh, tutor Mr Dedy jadwalnya mengajarnya di hari senin, selasa, kamis dan lain sebagainya
 - Tabel **ms_program** mempunyai relasi **one to many** dengan table **tr_jadwal** pada kolom **program_id**.
 - Tabel **ms_program.Program_id** sebagai **Primary Key** merupakan table referensi dari table **tr_jadwal.jadwal_id** yang berfungsi sebagai **foreign key**
 - Tabel **ms_tutor** mempunyai relasi **one to many** dengan table **tr_jadwal** pada kolom **tutor_id**.
 - Tabel **ms_tutor.tutor_id** sebagai **Primary Key** merupakan table referensi dari table **tr_jadwal.tutor_id** yang berfungsi sebagai **foreign key**
- a. Untuk keseragaman, mulailah dari awal yaitu dengan membuat ke-3 buah table tersebut. Sebelumnya, hapus atau drop dulu table – table diatas bila sudah ada sebelumnya

```
Kursus  
drop table ms_tutor  
drop table ms_program  
The command(s) completed successfully.
```

- b. Create ulang table `ms_tutor` dan table `ms_program` terlebih dahulu. Pada contoh dibawah adalah cara lain membuat constraint **primary key** dan **not null** selain yang diajarkan pada modul sebelumnya

```
create table ms_tutor
(
  Tutor_id          char (5) primary key not null,
  Tutor_name        varchar (30),
  Tutor_address      varchar (50),
  Tutor_phone        varchar (13),
  Tutor_email        varchar (50),
  Tutor_gender       varchar (6),
  Tutor_birth        datetime
)
```

The command(s) completed successfully.

```
create table ms_program
(
  Program_id        char (5) primary key not null,
  Program_name       varchar (50),
  Fee               numeric,
  Program_duration   varchar (20),
)
```

The command(s) completed successfully.

- c. Setelah table utama atau table yang menjadi referensi bagi table lainnya dibuat, barulah membuat table kedua atau table sekunder, misalnya disini adalah table `tr_jadwal`

```
create table tr_jadwal
(
  Jadwal_id          char (5) primary key not null,
  Day_               varchar (10),
  Time_              varchar (50),
  Program_id         char (5) foreign key references ms_program (Program_id),
  Room               char (3),
  tutor_id           char (5) foreign key references ms_tutor (tutor_id),
)
```

The command(s) completed successfully.

- d. Masukkan data berikut ke table `tr_jadwal`

```
insert into tr_jadwal
values ('J0001', 'Mon', '13.00-15.00', 'PC001', '101', 'D1908')
```

Server: Msg 547, Level 16, State 1, Line 1
INSERT statement conflicted with COLUMN FOREIGN KEY constraint 'FK_tr_jadwal_Progr_3C69FB99'.
The statement has been terminated.

Hasil eksekusi diatas menunjukkan adanya kesalahan pada constraint **foreign key**. Hal ini terjadi karena table `tr_jadwal` adalah table sekunder yang data – data atau record – record didalamnya merujuk/mengacu pada table utama/induk yang dalam hal ini adalah table `ms_tutor` dan table `ms_program`. Istilah mengacu pada table utama maksudnya data yang ada pada table `tr_jadwal` .`program_id` harus juga ada di table `ms_program`.`program_id` dan data pada table `tr_jadwal`.`tutor_id` juga harus ada di table `ms_program`.`tutor_id`.

Modul Praktikum Sistem Basis Data I S1-SI

Sebelumnya table **ms_tutor** dan **ms_program** isinya masih kosong sehingga tidak ada data yang dirujuk

- e. Untuk mengatasi hal diatas, masukkan data pada table **ms_program** dan **ms_tutor**. Perhatikan disini, data yang dimasukkan pada table **ms_program.program_id** harus **'PC001'** dan pada table **ms_tutor.tutor_id** harus **'D1908'**

```
insert into ms_program
values('PC001','conversation',1500000,'12')

INSERT INTO ms_tutor
VALUES ('D1908','Robert','Sukawati 56','0818277345','Robert@yahoo.com','Male','10/09/1970')
```

(1 row(s) affected)

(1 row(s) affected)

- f. Ulangi langkah no 5 dan perhatikan bagaimana hasilnya. Kalau sudah sukses, cek data yang ada pada table **tr_jadwal** dengan perintah

SELECT * FROM tr_jadwal

```
insert into tr_jadwal
values('J0001','Mon','13.00-15.00','PC001','101','D1908')
```

(1 row(s) affected)

```
select * from tr_jadwal
```

	Jadwal_id	Day_	Time_	Program_id	Room	tutor_id
1	J0001	Mon	13.00-15.00	PC001	101	D1908

- g. Hapus data di table **ms_program** yang **program_id = 'PC001'**. Perhatikan hasilnya.

```
delete from ms_program
where program_id = 'PC001'
```

Server: Msg 547, Level 16, State 1, Line 1
DELETE statement conflicted with COLUMN REFERENCE constraint 'FK_tr_jadwal_Progr_3C69FB99'.
The statement has been terminated.

Error diatas dikarenakan adanya upaya penghapusan data **program_id 'PC001'** pada table **ms_program** dimana data ini merupakan data yang dirujuk oleh table **tr_jadwal.program_id** yang pada saat ini juga berisi data **'PC001'**.

Demikian pula halnya bila ingin menghapus data pada table **ms_tutor**

```
delete from ms_tutor
where tutor_id = 'D1908'
```

Server: Msg 547, Level 16, State 1, Line 1
DELETE statement conflicted with COLUMN REFERENCE constraint 'FK_tr_jadwal_tutor_3D5E1FD2'.
The statement has been terminated.

- h. Supaya bisa menghapus data diatas maka langkah pertama adalah menghapus data – data yang ada di table kedua/sekunder pada kolom dan data yang bersesuaian

```
delete from tr_jadwal
where jadwal_id = 'J0001'
```

(1 row(s) affected)

- i. Sekarang, ulangi langkah no 1g

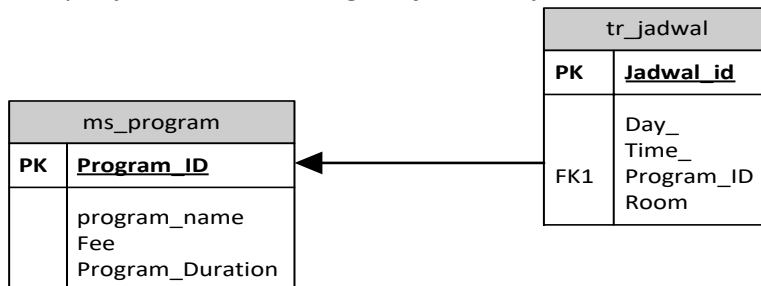
```
delete from ms_program
where program_id = 'PC001'
```

(1 row(s) affected)

```
delete from ms_tutor
where tutor_id = 'D1908'
```

(1 row(s) affected)

2. Constraint **Foreign Key** mempunyai opsi **on delete cascade on update cascade**. Untuk memudahkan, pada gambar dibawah merupakan desain table yang akan digunakan untuk mempelajari constraint **foreign key** lebih lanjut.



- a. hapus dulu table **tr_jadwal**

```
drop table tr_jadwal
```

The command(s) completed successfully.

- b. Create ulang table tr_jadwal, dengan struktur yang sedikit berbeda. Kolom **tutor_ID** dihilangkan

```
create table tr_jadwal
(
  Jadwal_ID      char(5) primary key not null,
  Day_           varchar(10),
  Time_          varchar(50),
  Program_ID     char(5) foreign key references ms_program(program_ID) on update cascade on delete cascade,
  Room           char(3)
)
```

The command(s) completed successfully.

- c. Masukkan data pada table **ms_program** dan **tr_jadwal**

```
INSERT INTO ms_program
values ('PC001', 'conversation', 1500000, '12')
```

```
INSERT INTO tr_jadwal
values ('J0001', 'Mon', '13.00-15.00', 'PC001', '101')
```

(1 row(s) affected)

- d. Hapus data di table **ms_program**

```
delete from ms_program
where program_ID = 'PC001'
```

(1 row(s) affected)

Hasil eksekusi menunjukan data bisa dihapus (hasil berbeda kita dapatkan pada percobaan penghapusan data di langkah no 1g)

- e. Cek data di table **tr_jadwal**

```
select *
from tr_jadwal
```

Jadwal_ID	Day_	Time_	Program_ID	Room
-----------	------	-------	------------	------

Terlihat data di table **tr_jadwal** ikut terhapus. Hal ini disebabkan opsi **on delete cascade** yang dibuat waktu create table **tr_jadwal**

- f. Ulangi langkah no 2 c

- g. Ubah data di table **ms_program**

```
update ms_program
set program_ID = 'XXXXXX'
where program_ID = 'PC001'
```

(1 row(s) affected)

```
select *
from ms_program
```

	Program_id	Program_name	Fee	Program_duration
1	XXXXXX	conversation	1500000	12

- h. Cek data kembali di table **tr_jadwal**

```
select *
from tr_jadwal
```

	Jadwal_ID	Day_	Time_	Program_ID	Room
1	J0001	Mon	13.00-15.00	XXXXX	101

Terlihat kolom **program_ID** yang semula datanya 'PC001' ikut berubah menjadi 'XXXXX' yang disebabkan **on update cascade** yang dibuat waktu create table **tr_jadwal**

3. Constraint **default** digunakan untuk memberikan nilai pada kolom yang tidak secara langsung diberikan nilai saat klausa **insert**
- a. Buat tabel **ms_grade**, dengan kolom **discount** diset **default '50%'**

```
create table ms_grade
(
  grade_id          char(1) primary key not null,
  nilai             varchar(7),
  discount           varchar(4) default '50%'
)
```

- b. Masukkan data di table **ms_grade** dan cek datanya

```
insert into ms_grade (grade_id, nilai)
values ('B', '85-100')
```

(1 row(s) affected)

```
select * from ms_grade
```

	grade_id	nilai	discount
1	B	85-100	50%

Terlihat meskipun kolom **discount** tidak diberi data saat klausa **insert**, secara otomatis diberi nilai **50%** yang merupakan default value untuk kolom **discount**

- c. Tambahkan beberapa record data lagi di table **ms_grade**

```
insert into ms_grade (grade_id, nilai, discount)
values ('C', '50-84', '10%')

insert into ms_grade (grade_id, nilai, discount)
values ('D', '0-49', null)
```

(1 row(s) affected)

(1 row(s) affected)

```
select * from ms_grade
```

	grade_id	nilai	discount
1	B	85-100	50%
2	C	50-84	10%
3	D	0-49	NULL

4. Constraint **identity** digunakan untuk memberikan nilai secara otomatis dan teratur secara menaik pada kolom yang tertentu dan bertipe angka. Identity biasanya digunakan untuk membuat **autonumber** atau **auto increment**

- a. Buat tabel **mhs**, dengan kolom **nomor** diset **identity(1,1)**

```
create table mhs  
(  
  nomor int identity(1,1),  
  nim varchar(9) primary key not null,  
  nama varchar(50)  
)
```

- b. Masukkan data seperti dibawah

```
insert into mhs  
values('100010100', 'Ayu Pratiwi')
```

(1 row(s) affected)

```
select * from mhs
```

	nomor	nim	nama
1	1	100010100	Ayu Pratiwi

```
insert into mhs  
values('100010200', 'Bambang')
```

(1 row(s) affected)

```
select * from mhs
```

	nomor	nim	nama
1	1	100010100	Ayu Pratiwi
2	2	100010200	Bambang

```
insert into mhs  
values(5, '100010300', 'Joko')
```

Server: Msg 8101, Level 16, State 1, Line 1

An explicit value for the identity column in table 'mhs' can only be specified when a column list is

```
insert into mhs  
values(3, '100010300', 'Joko')
```

Server: Msg 8101, Level 16, State 1, Line 1

An explicit value for the identity column in table 'mhs' can only be specified when a column list is

