

MANAJEMEN BASIS DATA SUB QUERY

A. Tujuan

- Memahami konsep subquery dan keterhubungan entitas di dalam basis data
- Memahami implementasi subquery di dalam basis data
- Mampu menyelesaikan operasi-operasi data spesifik dengan memanfaatkan subquery

B. Dasar Teori

Sub Query

Subquery adalah statement SELECT didalam SELECT. Subquery dinyatakan dalam tanda kurung (). Subquery dapat menyelesaikan persoalan rumit berkaitan dengan query data.

Contoh :

“Diperlukan data mahasiswa yang memiliki alamat yang sama dengan seorang mahasiswa dengan nim 001”

Menurut Anda, bagaimana menyelesaikan permasalahan tersebut ?

Melihat kasus diatas diperlukan dua tahapan untuk menyelesaikan permasalahan tersebut, yaitu:

Pertama, mendapatkan data alamat dari mahasiswa dengan nim 001.

Kedua, setelah mengetahui alamat mahasiswa nim 001, barulah bisa mengetahui data mahasiswa yang memiliki alamat yang sama dengan mahasiswa nim 001.

Dengan menggunakan subquery, maka penyelesaian permasalahan di atas dapat dipermudah karena hanya memerlukan satu tahapan saja,

Sintaks cara penulisan subquery adalah sebagai berikut :

```
SELECT select_list FROM table WHERE expr operator  
(SELECT select_list FROM table);
```

Klasifikasi 3 jenis subquery :

1. *Scalar Subquery*
Scalar subquery atau subquery baris tunggal hanya menghasilkan satu baris data.
Operator yang dapat digunakan ialah =, >, >=, <, <=, <>.
2. *Multiple-Row Subquery*
Multiple-row subquery atau subquery baris ganda mengembalikan lebih dari satu baris data.
Multiple-row subquery dapat menggunakan operator IN, ANY, SOME, ALL.
3. *Multiple-Colomn Subquery*
Multiple-colomn subquery atau subquery kolom ganda ialah mengembalikan lebih dari satu baris dan satu kolom data.

C. Latihan

1. Buatlah beberapa tabel dengan struktur sebagai berikut :

Tabel Mahasiswa : **tbMhs**

#	Name	Type
1	NIM	varchar(9)
2	Nama	varchar(35)
3	Alamat	varchar(50)
4	Tempat_lahir	varchar(25)
5	Tgl_lahir	date
6	JenKel	enum('L', 'P')

Tabel Dosen : **tbDosen**

#	Name	Type
1	kd_dosen	varchar(7)
2	nama	varchar(27)
3	alamat	varchar(50)

Tabel Jurusan ; **tbJurusan**

#	Name	Type
1	kode_jur	varchar(3)
2	nama_jur	varchar(35)
3	kd_dosen	varchar(7)

Tabel Mata Kuliah : **tbMataKuliah**

#	Name	Type
1	kd_mk	varchar(5)
2	nama_mk	varchar(25)
3	sks	int(2)
4	semester	int(2)
5	kd_dosen	varchar(7)

Tabel Ambil Mata Kuliah : **tbAmbilMK**

#	Name	Type
1	NIM	varchar(9)
2	kd_mk	varchar(5)

2. Inputkan data berikut kedalam tabel

tbMhs

NIM	Nama	Alamat	Tempat_lahir	Tgl_lahir	JenKel
190011001	Banyu Biru	Jl. Sukarno Hatta	Samarinda	2000-08-06	L
190011002	Sekar Harum	Jl. Juanda	Balikpapan	2000-11-15	P
190012001	Dian Sari	Jl. Jakarta	Jakarta	2000-01-30	P
190012002	Jaka Lanang	Jl. Sukarno Hatta	Surabaya	1999-10-12	L
190013001	Harum Manis	Jl. Sukarno Hatta	Tenggarong	2000-06-08	P
190013002	Donna N	Jl. Juanda	Balikpapan	2000-05-05	P
190013003	Novel	Jl. Ahmad Yani	Samarinda	2000-06-06	L

tbJurusan

kode_jur	nama_jur	kd_dosen
JMT	Matematika dan Teknologi Informasi	115005
JTE	Teknik Elektro	116003
JTS	Teknik Sipil	116001

tbDosen

kd_dosen	nama	alamat
1115005	Lili	Jl. Patimura
1116001	Sukarno	Jl. Sukarno Hatta
1116002	Suharto	Jl. Juanda
1116003	Edward	Jl. Jakarta
1117001	Rose	Jl. Sukarno Hatta

tbMataKuliah

kd_mk	nama_mk	sks	semester	kd_dosen
IF112	Basis Data	3	3	115005
IF123	Algoritma dan Pemrograman	3	2	115005
IF131	Komputer Grafik	3	4	1117001
KM101	Matematika Dasar	3	1	1116001
KM201	Matematika Lanjut	3	2	1116001
TE113	Medan Elektromagnetik	4	4	1116003
TE131	Pengolahan Sinyal Digital	3	4	116002

tbAmbilMK

id	NIM	kd_mk
1	190011001	IF123
2	190011001	IF112
3	190011002	IF112
4	190013001	TE113
5	190013001	TE131
6	190013003	TE131

3. Scalar Subquery

Contoh *subquery* baris tunggal : mendapatkan data mahasiswa yang memiliki jenis kelamin sama dengan mahasiswa bernama “Novel”

```
SELECT * FROM tbMhs
WHERE JenKel =
      (SELECT JenKel FROM tbMhs WHERE Nama = "Novel")
```

Sebagai hasilnya akan didapatkan jenis kelamin mahasiswa bernama “Novel”, yang selanjutnya digunakan oleh *main query* sehingga menghasilkan :

NIM	Nama	Alamat	Tempat_lahir	Tgl_lahir	JenKel
190011001	Banyu Biru	Jl. Sukarno Hatta	Samarinda	2000-08-06	L
190012002	Jaka Lanang	Jl. Sukarno Hatta	Surabaya	1999-10-12	L
190013003	Novel	Jl. Ahmad Yani	Samarinda	2000-06-06	L

4. Multiple-Row Subquery

Subquery ini menggunakan operator komparasi IN, ANY/ SOME, ALL.

a. Operator IN

Operator IN berarti sama dengan member di dalam list. Operator IN bisa digunakan untuk mendapatkan data dosen yang mengajar matakuliah.

```
SELECT kd_dosen, nama FROM tbDosen a
WHERE a.kd_dosen IN
      (SELECT kd_dosen FROM tbMatKul)
```

maka hasilnya :

kd_dosen	nama
1116001	Sukarno
1116003	Edward
1117001	Rose

b. Operator ANY/ SOME

Operator ANY/ SOME digunakan untuk membandingkan suatu nilai dengan setiap nilai yang dikembalikan oleh *subquery*.

Contoh : Mendapatkan data matakuliah yang memiliki sks lebih besar dari sembarang sks di semester 3.

```
SELECT * FROM tbMataKuliah
WHERE sks > ANY
      (SELECT sks FROM tbMataKuliah WHERE semester = 3)
```

hasilnya :

kd_mk	nama_mk	sks	semester	kd_dosen
TE113	Medan Elektromagnetik	4	4	1116003

Informasi :

1. Operator = ANY ekuivalen dengan IN
2. Operator < ANY ekuivalen dengan MAX (lebih kecil dibanding MAX)
3. Operator > ANY ekuivalen dengan MIN (lebih besar dibanding MIN)

c. Operator ALL

Operator ALL digunakan untuk membandingkan suatu nilai dengan semua nilai yang dikembalikan oleh *subquery*.

Contoh : Mendapatkan data matakuliah yang memiliki sks lebih besar dari semua sks di semester 3.

```
SELECT * FROM tbMataKuliah
WHERE sks > ALL
      (SELECT sks FROM tbMataKuliah WHERE semester = 3)
```

hasilnya :

kd_mk	nama_mk	sks	semester	kd_dosen
TE113	Medan Elektromagnetik	4	4	1116003

Informasi :

1. Operator < ALL ekuivalen dengan MAX (lebih kecil dibanding MIN)
2. Operator > ALL ekuivalen dengan MIN (lebih besar dibanding MAX)

5. Multiple-Column Subquery

Subquery ini menggunakan operator komparasi IN, ANY/ SOME atau ALL

6. Operator EXISTS dan NOT EXISTS

Operator *EXISTS* dan *NOT EXISTS* digunakan pada *correlated subquery* untuk memeriksa apakah *subquery* mengembalikan hasil atau tidak. Apabila *subquery* mengembalikan hasil, *EXIST* akan mengembalikan nilai *true*. Begitu pula sebaliknya, jika tidak mengembalikan hasil.

a. EXISTS

Operator ini bisa digunakan untuk mendapatkan data matakuliah yang diambil oleh mahasiswa.

```
SELECT * FROM tbMataKuliah a
WHERE EXISTS
      (SELECT * FROM tbAmbilMK b WHERE a.kd_mk = b.kd_mk)
```

kd_mk	nama_mk	sks	semester	kd_dosen
IF112	Basis Data	3	3	115005
IF123	Algoritma dan Pemrograman	3	2	115005
TE113	Medan Elektromagnetik	4	4	1116003
TE131	Pengolahan Sinyal Digital	3	4	116002

b. NOT EXISTS

Pernyataan berikut akan mendapatkan data matakuliah yang tidak diambil oleh mahasiswa.

```
SELECT * FROM tbMataKuliah a
WHERE NOT EXISTS
      (SELECT * FROM tbAmbilMK b WHERE a.kd_mk = b.kd_mk)
```

kd_mk	nama_mk	sks	semester	kd_dosen
IF131	Komputer Grafik	3	4	1117001
KM101	Matematika Dasar	3	1	1116001
KM201	Matematika Lanjut	3	2	1116001

D. Tugas

1. Dapatkan kode dan nama matakuliah dosen yang menjadi Ketua Jurusan Teknik Sipil.
2. Dapatkan data mahasiswa yang tidak mengambil matakuliah.
3. Dapatkan data dosen yang mengajar matakuliah di atas semester 3.
4. Dapatkan data matakuliah dosen yang bukan merupakan Ketua Jurusan Matematika dan Teknologi Informasi.
5. Dapatkan data dosen pengajar yang tidak diambil oleh mahasiswa
6. Dapatkan data mahasiswa yang tinggal satu wilayah dengan dosen yang bukan Ketua Jurusan Teknik Sipil