



Topik

Select JOIN

Tujuan

Mahasiswa diharapkan dapat:

1. Memahami keterhubungan entitas di dalam basis data
2. Memahami jenis-jenis operasi pengambilan data di beberapa entitas
3. Mampu menyelesaikan kasus-kasus *retrieval* yang melibatkan lebih dari satu entitas

Teori

✓ **Relationship**

Relationship adalah suatu hubungan antara beberapa entitas. Konsep ini sangat penting sekali di dalam basis data, di mana memungkinkan entitas-entitas untuk saling berhubungan satu sama lain. Di dalam sebuah relationship, primary key memiliki peran penting untuk mengaitkan entitas. Selain itu, primary key juga digunakan untuk mendefinisikan batasan keterhubungan.

✓ **Join**

Join merupakan salah satu konstruksi dasar dari SQL dan basis data. Join dapat didefinisikan sebagai kombinasi record dari dua atau lebih tabel di dalam basis data relasional dan menghasilkan sebuah tabel (temporary) baru yang disebut sebagai joined table. Join dapat diklasifikasikan ke dalam dua jenis: inner dan outer.

a. Inner Join

Inner join pada dasarnya adalah menemukan persimpangan (intersection) antara dua buah tabel.

Sintaks *inner join* diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1
    INNER JOIN r2
    ON r1.join_key = r2.join_key
```

Cara 1

```
SELECT A1, A2, ..., An
FROM r1
INNER JOIN r2
ON r1.join_key = r2.join_key
```

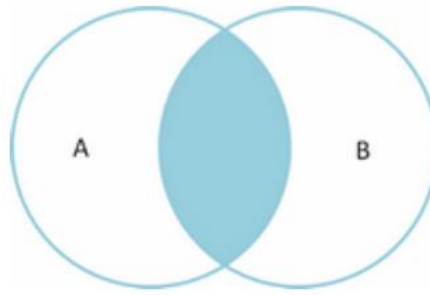
Inner join juga dapat direpresentasikan dalam bentuk implisit.

```
SELECT A1, A2, ..., An
FROM r1, r2
WHERE r1.key = r2.key
```

Cara 2

```
SELECT A1, A2, ..., An
FROM r1, r2
WHERE r1.key = r2.key
```

Misalkan terdapat tabel A dan B, maka hasil *inner join* dapat diperlihatkan—sebagai bidang terarsir—dalam diagram Venn seperti Gambar 1.



Gambar 1. Inner Join

b. Outer Join

- **Left Outer Join**

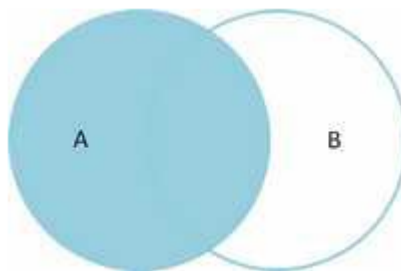
Left outer join (atau left join) mengembalikan semua nilai dari tabel kiri ditambah dengan nilai dari tabel kanan yang sesuai (atau NULL jika tidak ada nilai yang sesuai).

```
SELECT A1, A2, ..., An
FROM r1
  LEFT OUTER JOIN r2
    ON r1.join_key = r2.join_key
```

Syntax

```
SELECT A1, A2, ..., An
FROM r1
LEFT OUTER JOIN r2
ON r1.join_key = r2.join_key
```

Left outer join antara tabel A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar 2.



Gambar 2. Left Outer Join

- **Right Outer Join**

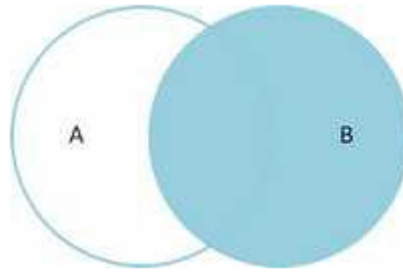
Right outer join (atau right join) pada dasarnya sama seperti left join, namun dalam bentuk terbalik—kanan dan kiri. Sintaks *right outer join* diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1
  RIGHT OUTER JOIN r2
    ON r1.join_key = r2.join_key
```

Syntax

```
SELECT A1, A2, ..., An
FROM r1
RIGHT OUTER JOIN r2
ON r1.join_key = r2.join_key
```

3. *Right outer join* antara tabel A dan B dapat diilustrasikan dalam diagram Venn seperti Gambar



Gambar 3. Right Outer Join

- **Full Outer Join**

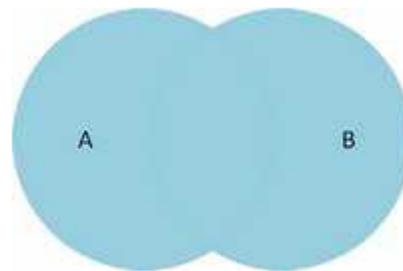
Full outer join (atau full join) pada hakekatnya merupakan kombinasi dari left dan right join. Sintaks *full outer join* diperlihatkan sebagai berikut:

```
SELECT A1, A2, ..., An
FROM r1
FULL OUTER JOIN r2
ON r1.join_key = r2.join_key
```

Syntax

```
SELECT A1, A2, ..., An
FROM r1
FULL OUTER JOIN r2
ON r1.join_key = r2.join_key
```

Bentuk visual dari *full outer join* dapat diperlihatkan menggunakan diagram Venn seperti Gambar 4.



Gambar 4. Full Outer Join

Selain empat jenis join yang utama di atas, masih ada beberapa variasi join lainnya, seperti CROSS JOIN (cartesian product), NATURAL JOIN, dan sebagainya. Perlu juga diperhatikan, join bisa diimplementasikan dalam bentuk bersarang (nested join). Jadi, di dalam sebuah operasi join bisa terdapat operasi join lainnya.

PRAKTIKUM

1. Relationship

Buat database nama_kantor (isi dengan nama anda)

```
MariaDB [(none)]> create database mayang_kantor;  
Query OK, 1 row affected (0.06 sec)
```

Buat tabel karyawan dan tabel departemen dengan struktur sebagai berikut:

```
mysql> \u _100533405409  
Database changed  
mysql> create table karyawan(  
-> nama varchar (30) NOT NULL,  
-> id_dep int (5) NOT NULL)  
-> ENGINE = myISAM;  
Query OK, 0 rows affected (0.11 sec)  
  
mysql> create table departemen(  
-> id_dep int (5) NOT NULL,  
-> nama_dep varchar (30) NOT NULL,  
-> PRIMARY KEY(id_dep))  
-> ENGINE = myISAM;  
Query OK, 0 rows affected (0.34 sec)
```

```
CREATE TABLE karyawan ( nama varchar(30) NOT NULL, id_dep int(5)  
NOT NULL  
) ENGINE=MyISAM;
```

```
CREATE TABLE departemen ( id_dep int(5) NOT NULL, nama_dep  
varchar(30) NOT NULL, PRIMARY KEY (id_dep)  
) ENGINE=MyISAM;
```

Data yang digunakan adalah sebagai berikut:

- Tabel Karyawan

Nama	id_dep
Agus	10
Budi	16
Citra	12
Dani	17

- Tabel Departmen

id_dep	nama_dep
10	Penelitian
11	Pemasaran
12	SDM
13	Keuangan

```
mysql> insert into karyawan (nama, id_dep)  
-> values ('Agus', 10), ('Budi', 16), ('Citra', 12), ('Dani', 17);  
Query OK, 4 rows affected (0.00 sec)  
Records: 4 Duplicates: 0 Warnings: 0  
  
mysql> insert into departemen (id_dep, nama_dep)  
-> values (10, 'Penelitian'), (11, 'Pemasaran'), (12, 'SDM'), (13, 'Keuangan'  
-> );  
Query OK, 4 rows affected (0.02 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

2. Inner Join

Sebagaimana dijelaskan, *inner join* akan mengembalikan data di tabel A dan B yang sesuai. Sebagai contoh, kita bisa mendapatkan data karyawan yang memiliki departemen.

a. Data karyawan yang memiliki departemen

Cara 1

```
mysql> select * from karyawan
-> INNER JOIN departemen
-> ON karyawan.id_dep = departemen.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Citra	12	12	SDM

2 rows in set (0.03 sec)

Cara 2

```
mysql> select * from karyawan, departemen
-> where karyawan.id_dep = departemen.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Citra	12	12	SDM

2 rows in set (0.00 sec)

Dalam pengambilan data ini, kita juga bisa menspesifikasikan field terkait. Sebagai contoh, hanya mengambil nama karyawan dan nama departemen saja.

```
mysql> select karyawan.nama, departemen.nama_dep
-> from karyawan
-> inner join departemen
-> on karyawan.id_dep = departemen.id_dep;
```

nama	nama_dep
Agus	Penelitian
Citra	SDM

2 rows in set (0.00 sec)

Agar penulisan SQL lebih efisien, kita dapat memanfaatkan fitur “derived table” (atau alias).

```
mysql> select k.nama, d.nama_dep
-> from karyawan k
-> inner join departemen d
-> on k.id_dep = d.id_dep;
```

nama	nama_dep
Agus	Penelitian
Citra	SDM

2 rows in set (0.02 sec)

Pada pernyataan SQL di atas, tabel karyawan dinotasikan dengan huruf k dan tabel departemen menggunakan huruf d.

3. **Outer Join**

a. Left Outer Join

Data seluruh karyawan (yang memiliki departemen maupun tidak)

```
mysql> select * from karyawan k left outer join departemen d
-> on k.id_dep = d.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Budi	16	NULL	NULL
Citra	12	12	SDM
Dani	17	NULL	NULL

4 rows in set (0.03 sec)

Data karyawan yang tidak memiliki departemen

```
mysql> select * from karyawan k left outer join departemen d
-> on k.id_dep = d.id_dep
-> where d.id_dep is null;
```

nama	id_dep	id_dep	nama_dep
Budi	16	NULL	NULL
Dani	17	NULL	NULL

2 rows in set (0.02 sec)

b. Right Outer Join

```
mysql> select * from karyawan k right outer join departemen d
-> on k.id_dep = d.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
NULL	NULL	11	Pemasaran
Citra	12	12	SDM
NULL	NULL	13	Keuangan

4 rows in set (0.00 sec)

c. Full Outer Join

```
mysql> select * from karyawan k left outer join departemen d
-> on k.id_dep = d.id_dep
-> union
-> select * from karyawan k right outer join departemen d
-> on k.id_dep = d.id_dep;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Budi	16	NULL	NULL
Citra	12	12	SDM
Dani	17	NULL	NULL
NULL	NULL	11	Pemasaran
NULL	NULL	13	Keuangan

6 rows in set (0.00 sec)

d. Cross Join

Cara 1

```
mysql> select * from karyawan cross join departemen;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Budi	16	10	Penelitian
Citra	12	10	Penelitian
Dani	17	10	Penelitian
Agus	10	11	Pemasaran
Budi	16	11	Pemasaran
Citra	12	11	Pemasaran
Dani	17	11	Pemasaran
Agus	10	12	SDM
Budi	16	12	SDM
Citra	12	12	SDM
Dani	17	12	SDM
Agus	10	13	Keuangan
Budi	16	13	Keuangan
Citra	12	13	Keuangan
Dani	17	13	Keuangan

16 rows in set (0.00 sec)

Cara 2

```
mysql> select * from karyawan, departemen;
```

nama	id_dep	id_dep	nama_dep
Agus	10	10	Penelitian
Budi	16	10	Penelitian
Citra	12	10	Penelitian
Dani	17	10	Penelitian
Agus	10	11	Pemasaran
Budi	16	11	Pemasaran
Citra	12	11	Pemasaran
Dani	17	11	Pemasaran
Agus	10	12	SDM
Budi	16	12	SDM
Citra	12	12	SDM
Dani	17	12	SDM
Agus	10	13	Keuangan
Budi	16	13	Keuangan
Citra	12	13	Keuangan
Dani	17	13	Keuangan

16 rows in set (0.00 sec)

Tugas

Perhatikan, dalam mengerjakan tugas praktikum ini, sebaiknya pernyataan SQL disimpan di file untuk kemudian dieksekusi. Berikut adalah data-data tabel yang akan digunakan (**sesuaikan nilainya agar sama persis**).

- Tabel mahasiswa

nim	Nama	jenis_kelamin	alamat
101	Arif	L	Jl. Kenangan
102	Budi	L	Jl. Jombang
103	Wati	P	Jl. Surabaya
104	Ika	P	Jl. Jombang
105	Tono	L	Jl. Jakarta
106	Iwan	L	Jl. Bandung
107	Sari	P	Jl. Malang

- Tabel ambil_mk

nim	kode_mk
101	PTI447
103	TIK333
104	PTI333
104	PTI777
111	PTI123
123	PTI999

- Tabel matakuliah

kode_mk	nama_mk	sks	semester
PTI447	Praktikum Basis Data	1	3
TIK342	Praktikum Basis Data	1	3
PTI333	Basis Data Terdistribusi	3	5
TIK123	Jaringan Komputer	2	5
TIK333	Sistem Operasi	3	5
PTI123	Grafika Multimedia	3	5
PTI777	Sistem Informasi	2	3

Soal

1. Dapatkan data mahasiswa yang mengambil matakuliah. Selesaikan dengan pendekatan join eksplisit dan implisit. Hindari duplikasi data.

nim	nama
101	Arif
103	Wati
104	Ika

3 rows in set (0.00 sec)

2. Kelompokkan data mahasiswa yang tidak mengambil matakuliah berdasarkan jenis kelaminnya, kemudian hitung banyaknya.

jenis_kelamin	jumlah
L	3
P	1

2 rows in set (0.00 sec)

3. Dapatkan nim dan nama mahasiswa yang mengambil matakuliah beserta kode_mk dan nama_mk yang diambilnya. Selesaikan dengan pendekatan join eksplisit dan implisit.

nim	nama	kode_mk	nama_mk
101	Arif	PTI447	Praktikum Basis Data
103	Wati	TIK333	Sistem Operasi
104	Ika	PTI333	Basis Data Terdistribusi
104	Ika	PTI777	Sistem Informasi

4 rows in set (0.00 sec)

4. Dapatkan nim, nama, dan total sks yang diambil oleh mahasiswa, di mana total sksnya lebih dari 4 dan kurang dari 10.

nim	nama	total_sks
104	Ika	5

1 row in set (0.00 sec)

5. Dapatkan matakuliah yang tidak diambil oleh mahasiswa terdaftar (mahasiswa di tabel mahasiswa).

kode_mk	nama_mk	sks	semester
PTI123	Grafika Multimedia	3	5
TIK123	Jaringan Komputer	2	5
TIK342	Praktikum Basis Data	1	3

3 rows in set (0.00 sec)

-- Selamat Mengerjakan --