CHAPTER

14

SQL
OPERASI JOIN

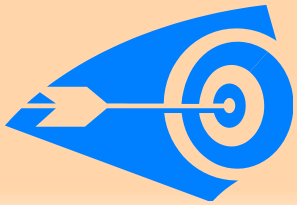
Arif Basofi
PENS 2015

Objectives

Tujuan:

Mengenal perintah SQL dengan operasi **JOIN**:

- Equijoin (Inner Join atau Simple Join)
- Non-Equijoin
- Outer Join (Left Outer Join & Right Outer Join)
- Self Join



S Q L – JOIN

- SQL tidak hanya menyediakan mekanisme query dan operasi modifikasi database saja, tetapi SQL juga menyediakan mekanisme untuk menggabungkan (**join**) relasi-relasi.
- Saat data yang dibutuhkan berasal lebih dari satu table, maka kondisi **join** dibutuhkan.
- Umumnya dalam men-**join** table berdasarkan pada kolom yang bersesuaian **Primary Key** dari table-1dengan **Foreign Key** dari table-2, atau yang disebut dengan **join** atau **equi-join**.
- Kondisi Join meliputi:
  - Equijoin (Inner Join atau Simple Join)
  - Non-Equijoin
  - Outer Join (Left Outer Join & Right Outer Join)
  - Self Join

S Q L – JOIN

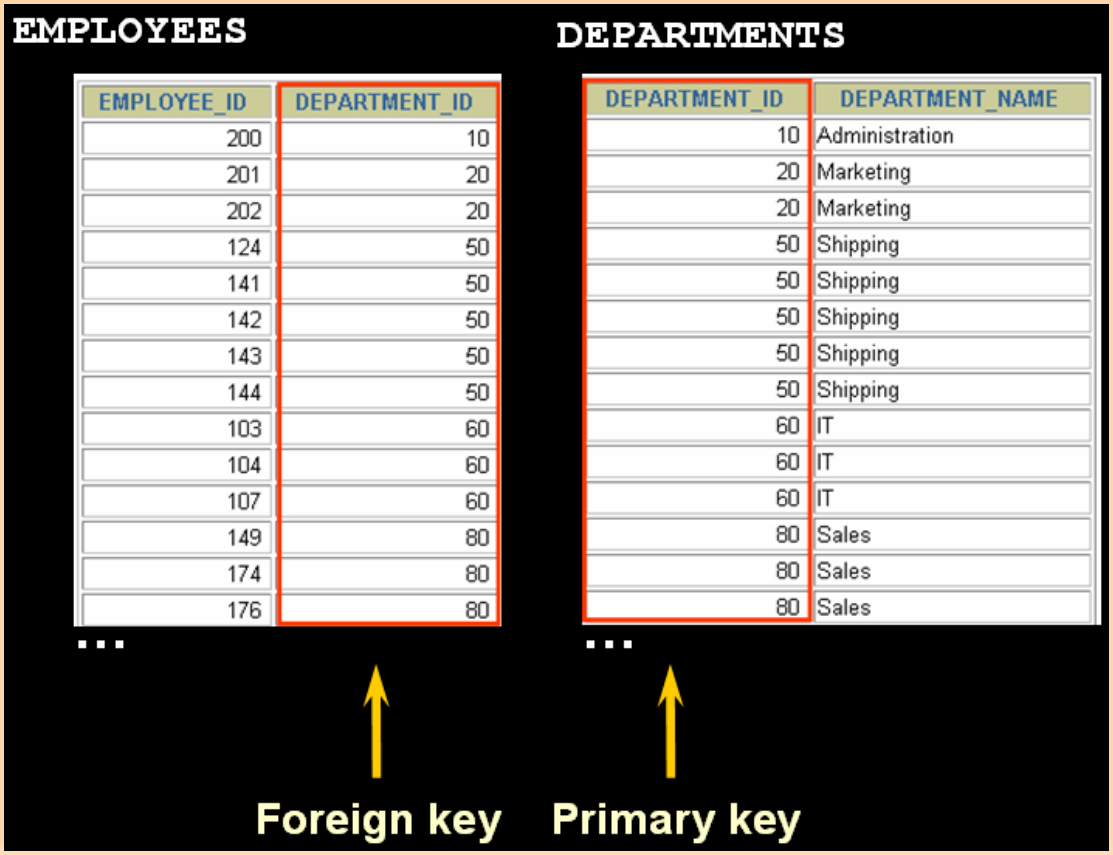
- **Syntax Join SQL JOIN / EQUI-JOIN**:

```
SELECT    table1.column, table2.column
FROM      table1, table2
WHERE     table1.column1 = table2.column2;
```

S Q L – EquiJoin (Inner Join atau Simple Join)

- **Join / Equijoin** atau biasa disebut sebagai **Inner Join** atau **Simple Join** adalah bentuk kondisi join dimana nilai relasi yang terjadi antar dua table (binary relation) adalah **sama** (terdapat hubungan antara **Primary Key** dan **Foreign Key**)

Contoh:



EMPLOYEE_ID	DEPARTMENT_ID
200	10
201	20
202	20
124	50
141	50
142	50
143	50
144	50
103	60
104	60
107	60
149	80
174	80
176	80
...	

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
20	Marketing
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
60	IT
60	IT
60	IT
80	Sales
80	Sales
80	Sales
...	

S Q L – EquiJoin (Inner Join atau Simple Join)

Retrieving Records with Equijoins

```
SELECT employees.employee_id, employees.last_name,
       employees.department_id, departments.department_id,
       departments.location_id
FROM   employees, departments
WHERE  employees.department_id = departments.department_id;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500
143	Matos	50	50	1500
144	Vargas	50	50	1500

...

19 rows selected.

S Q L – EquiJoin (Inner Join atau Simple Join)

Kondisi Join dengan Operator AND

```
SELECT last_name, employees.department_id,
       department_name
  FROM   employees, departments
  WHERE  employees.department_id = departments.department_id
  AND    last_name = 'Matos';
```

EMPLOYEES

LAST_NAME	DEPARTMENT_ID
Whalen	10
Hartstein	20
Fay	20
Mourgos	50
Rajs	50
Davies	50
Matos	50
Vargas	50
Hunold	60
Ernst	60

...

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
20	Marketing
50	Shipping
50	Shipping
50	Shipping
50	Shipping
50	Shipping
60	IT
60	IT

...

S Q L – EquiJoin (Inner Join atau Simple Join)

Kondisi Join Menggunakan Table Alias

- Menyederhanakan queries dengan menggunakan table alias.
- Meningkatkan performance.

```
SELECT e.employee_id, e.last_name, e.department_id,
       d.department_id, d.location_id
FROM   employees e , departments d
WHERE  e.department_id = d.department_id;
```

S Q L – EquiJoin (Inner Join atau Simple Join)

Kondisi Join Menggunakan Table Alias

```
SELECT e.last_name, d.department_name, l.city
FROM   employees e, departments d, locations l
WHERE  e.department_id = d.department_id
AND    d.location_id = l.location_id;
```

EMPLOYEES

LAST_NAME	DEPARTMENT_ID
King	90
Kochhar	90
De Haan	90
Hunold	60
Ernst	60
Lorentz	60
Mourgos	50
Rajs	50
Davies	50
Matos	50
Vargas	50
Zlotkey	80
Abel	80
Taylor	80

...

20 rows selected.

DEPARTMENTS

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

8 rows selected.

LOCATIONS

LOCATION_ID	CITY
1400	Southlake
1500	South San Francisco
1700	Seattle
1800	Toronto
2500	Oxford

S Q L – Non - EquiJoin

- **Non - Equijoin** adalah kondisi join yang terkadang tidak mengandung operator sama dengan (=).

  Contoh:

EMPLOYEES

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Hunold	9000
Ernst	6000
Lorentz	4200
Mourgos	5800
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500
Zlotkey	10500
Abel	11000
Taylor	8600

...

20 rows selected.

JOB_GRADES

GRA	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

Salary dalam table EMPLOYEES harus bernilai antara lowest salary dan highest salary pada table JOB_GRADES.

S Q L – Non - EquiJoin

Retrieving Records with Non-Equijoins

```
SELECT e.last_name, e.salary, j.grade_level
FROM   employees e, job_grades j
WHERE  e.salary
       BETWEEN j.lowest_sal AND j.highest_sal;
```

LAST_NAME	SALARY	GRA
Matos	2600	A
Vargas	2500	A
Lorentz	4200	B
Mourgos	5800	B
Rajs	3500	B
Davies	3100	B
Whalen	4400	B
Hunold	9000	C
Ernst	6000	C

...

20 rows selected.

S Q L – Self Join

- **Self Join** adalah bentuk kondisi join yang terjadi pada table diri sendiri (recursive).
- Misal. Ingin mencari nama manager dari tiap employee, tentunya akan mencari pada table yang sama yaitu **EMPLOYEES**.

EMPLOYEES (WORKER)

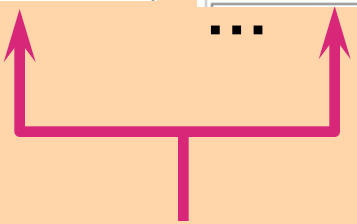
EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

...

EMPLOYEES (MANAGER)

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

...



MANAGER_ID in the WORKER table is equal to EMPLOYEE_ID in the MANAGER table.

S Q L – Self Join

- **Self Join** adalah bentuk kondisi join yang terjadi pada table diri sendiri (recursive).
- Misal. Ingin mencari nama manager dari tiap employee, tentunya akan mencari pada table yang sama yaitu **EMPLOYEES**.

```
SELECT worker.last_name || ' works for '
       || manager.last_name
FROM   employees worker, employees manager
WHERE  worker.manager_id = manager.employee_id ;
```

WORKER.LAST_NAME||'WORKSFOR'||MANAGER.LAST_NAME
Kochhar works for King
De Haan works for King
Mourgos works for King
Zlotkey works for King
Hartstein works for King
Whalen works for Kochhar
Higgins works for Kochhar
Hunold works for De Haan
Ernst works for Hunold

...

19 rows selected.

S Q L – Excercise

Latihan:

1. Buat SQL Query untuk menampilkan last name, department number, dan department name untuk semua pegawai.
2. Tampilkan semua job (job id) pegawai secara unik yang berada pada department 80 termasuk nama lokasinya.
3. Buat query yang menampilkan last name, nama department, location id dan kota dari semua pegawai yang memiliki komisi.
4. Tampilkan last name pegawai dan nama department untuk semua pegawai yang memiliki huruf ‘a’ pada last name.
5. Buat query yang menampilkan last name, department number, department name untuk semua pegawai yang bekerja di kota Toronto.

S Q L – Exercise…

Latihan…:

6. Tampilkan data nama terakhir pegawai, email, gaji, nomor departemen dan nama departemen untuk pegawai yang memiliki gaji antara $500 hingga @2500.
7. Tampilkan data nama pertama pegawai, email, no telepon, gaji, kode job dan nama jobnya untuk pegawai yang berada di departemen 'Information Technology', 'Financial' dan 'Human Resource Development'.
8. Tampilkan data no pegawai, nama terakhir pegawai, no departemen, nama departemen untuk pegawai yang memiliki kode departemen 'HRD', 'TI' dan 'FIN'. Urutkan menurut kode departementnya.
9. Tampilkan data nama terakhir pegawai, nama pekerjaannya, nama departemen, kota dan negara dimana departemen tersebut berada, urutkan menurut nama departemen dan nama terakhir pegawai.
10. Tampilkan data no departemen, nama departemen, rata-rata gaji pegawai per-departemen, untuk pegawai yang memiliki kode departemen 'HRD', 'TI' dan 'FIN'.

Skema Database Company