

DASAR SQL

MI2143

MODUL PRAKTIKUM



giving and caring the world

Penyusun:
Wardani Muhamad, MT
wdm@politeknikt Telkom.ac.id

Program Studi Manajemen Informatika
Politeknik Telkom
2013

“Untuk digunakan secara terbatas di lingkungan Politeknik Telkom”

Daftar Isi

| | |
|--|---|
| Daftar Isi | i |
| 1 Bab 8 Join..... | 1 |
| 1.1 Identitas | 1 |
| 1.2 Praktik | 2 |
| 1.2.1 Klausa Natural Join..... | 3 |
| 1.2.2 Klausa USING | 4 |
| 1.2.3 Klausa ON..... | 5 |
| 1.2.4 Membatasi Data pada Join | 6 |
| 1.2.5 Menggunakan Group Function pada Join | 7 |

1 BAB 8 JOIN

1.1 IDENTITAS

Kajian

Data Manipulation Language

Topik

- Join

Referensi

1. Oracle Database 10g. SQL Fundamental 1 – Student Guide Volume 1. California: Oracle Publisher 2004.
2. Oracle Database 10g. SQL Reference 10g Release 2. California: Oracle Publisher 2005.
3. Oracle Database 11g. SQL Language Reference 11g Release 2. July 2013

Kompetensi Utama

1. Mahasiswa menguasai konsep seleksi data dari beberapa table yang digabung
2. Mahasiswa mampu melakukan equijoin dengan menggunakan natural join, klausa ON, dan klausa USING

Lama Kegiatan Praktikum

1. Pertemuan Terbimbing : 1 x 120 menit
2. Kegiatan Mandiri : 2 x 120 menit

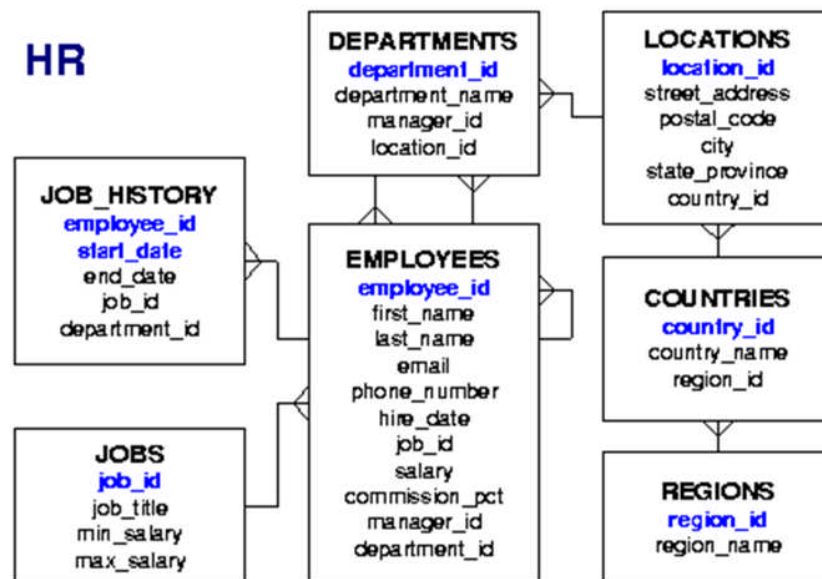
Parameter Penilaian

- | | |
|----------------------|-------|
| 1. Tugas Pendahuluan | (20%) |
| 2. Tes Awal | (20%) |
| 3. Tes Akhir | (60%) |

1.2 PRAKTIK

Klausula join pada SQL digunakan untuk mengkombinasikan (menggabung) baris data (record) dari dua table atau lebih pada basisdata. Umumnya, ketika menggunakan join, ditentukan kondisi-kondisi yang dapat mendefinisikan keterhubungan antar table.

Pada praktikum Mata Kuliah Dasar SQL, kita menggunakan schema HR. Keterhubungan (relasi) antar table pada schema HR disajikan pada gambar berikut:



Pada gambar di atas, disajikan keterhubungan antar table melalui hubungan Primary Key dengan Foreign Key. Sebagai contoh, table departments dan locations terhubung melalui column location_id. Column location_id pada table departments berperan sebagai foreign key, sedangkan pada table locations berperan sebagai primary key.

Hal pertama yang harus dikuasai ketika menggabungkan beberapa table (join) pada sebuah SQL Statement adalah menentukan jembatan antar table. Jembatan merupakan pasangan column yang terdapat pada table yang digabungkan dan digunakan untuk menentukan keterhubungan antar table. Umumnya, jembatan merupakan column yang memiliki constraint primary key dan foreign key, akan tetapi tidak menutup kemungkinan jika pasangan column tersebut bukan merupakan kedua constraint tersebut.

Dengan menggunakan join, kita dapat menyajikan informasi dengan lebih rinci atau lengkap, karena sebagian informasi mungkin berada di sebuah table sedangkan sebagian informasi lainnya berada pada table yang menjadi pasangannya. Salah satu contoh penggunaan join untuk menyajikan informasi secara lengkap adalah sebagai berikut: jika kita ingin menyajikan informasi berupa alamat lengkap (nama jalan, kode pos, kota, dan provinsi) sebuah departemen sesuai dengan data nama departemen yang kita ketahui, kita tidak dapat menyajikan informasi secara lengkap jika hanya menyajikannya dari table departemen. Hal ini dikarenakan informasi alamat lengkap tersimpan pada table locations. Untuk menyajikan alamat lengkap, berarti kita harus menyajikan nilai pada column street_address, postal_code, city, dan state_province sesuai dengan nilai pada column location_id yang bersesuaian antara table departments dan locations.

Sintaks umum klausa join sesuai dengan standard SQL:1999 adalah

```
SELECT table1.column, table2.column
FROM   table1
[NATURAL JOIN table2] |
[JOIN table2 USING (column_name)] |
[JOIN table2
  ON (table1.column_name = table2.column_name)] |
[LEFT|RIGHT|FULL OUTER JOIN table2
  ON (table1.column_name = table2.column_name)] |
[CROSS JOIN table2];
```

Melalui sintaks umum join diatas, dapat diperhatikan bahwa untuk menggabungkan beberapa table, kita harus menambahkan klausa join pada SQL Statement. Pada praktikum Mata Kuliah Dasar SQL, hanya tiga klausa join yang akan digunakan, yaitu:

1. Klausa NATURAL JOIN
2. Klausa USING
3. Klausa ON

1.2.1 Klausa Natural Join

Natural Join merupakan operasi join yang menciptakan klausa join secara implisit dengan membandingkan kolom-kolom umum yang berada pada table yang digabungkan. Maksud klausa join secara implisit adalah bahwa kita tidak perlu mendefinisikan secara eksplisit keterhubungan (kolom yang menghubungkan antara satu table dengan table yang lainnya) antar table yang digabung. Oracle secara otomatis akan menghubungkan table tersebut melalui column-column yang memiliki nama yang sama pada semua table yang digabung. Column yang dibandingkan, selain memiliki nama yang sama juga harus memiliki tipe data yang sama.

Contoh: jika kita menggabungkan table countries dan regions menggunakan natural join, maka Oracle secara otomatis akan membandingkan column region_id yang terdapat pada kedua table.

1.2.1.1 Soal

Tampilkan nama departemen dan alamat lengkap departemen tersebut. Alamat lengkap terdiri atas nama jalan, kode pos, kota, dan provinsi

1.2.1.2 Langkah Penyelesaian

Untuk menyelesaikan permasalahan diatas, kita harus menentukan table yang menyimpan informasi yang dibutuhkan:

1. Table yang menyimpan informasi nama departemen adalah table departments
2. Table yang menyimpan informasi alamat lengkap (nama jalan, kode pos, kota, dan provinsi) adalah table locations
3. Untuk mendapatkan informasi dari kedua table tersebut, kita harus melakukan join kepada table departments dan locations.

1.2.1.3 SQL Command

Klausa natural join yang digunakan untuk menghasilkan informasi tersebut adalah

```
SELECT department_name,street_address,postal_code,city,
       state_province
FROM departments
NATURAL JOIN locations
```

Hasil eksekusi SQL Command di atas adalah menyajikan informasi sebagai berikut:

| DEPARTMENT_NAME | STREET_ADDRESS | POSTAL_CODE | CITY | STATE_PROVINCE |
|--------------------|---------------------|-------------|---------------------|----------------|
| IT | 2014 Jabberwocky Rd | 26192 | Southlake | Texas |
| Shipping | 2011 Interiors Blvd | 99236 | South San Francisco | California |
| Administration | 2004 Charade Rd | 98199 | Seattle | Washington |
| Purchasing | 2004 Charade Rd | 98199 | Seattle | Washington |
| Executive | 2004 Charade Rd | 98199 | Seattle | Washington |
| Finance | 2004 Charade Rd | 98199 | Seattle | Washington |
| Accounting | 2004 Charade Rd | 98199 | Seattle | Washington |
| Treasury | 2004 Charade Rd | 98199 | Seattle | Washington |
| Corporate Tax | 2004 Charade Rd | 98199 | Seattle | Washington |
| Control And Credit | 2004 Charade Rd | 98199 | Seattle | Washington |

1.2.2 Klausa USING

Penggunaan klausa NATURAL JOIN memiliki kekurangan, yaitu tidak dapat mencocokkan column yang berpasangan jika tipe data yang dimiliki oleh kedua column tersebut berbeda, meskipun memiliki kesamaan nama column. Kekurangan ini dapat diantisipasi dengan penggunaan klausa USING. Ketika menggunakan klausa USING, kita harus menentukan secara eksplisit column yang menjadi jembatan antar table yang digabungkan. Klausa USING hanya bisa melibatkan sebuah column saja. Jika terdapat lebih dari satu column yang sama, kita harus memilih salah satunya. Klausa USING tidak dapat digunakan secara bersamaan dengan klausa NATURAL JOIN.

1.2.2.1 Soal

Tampilkan nama departemen dan alamat lengkap departemen tersebut. Alamat lengkap terdiri atas nama jalan, kode pos, kota, dan provinsi.

Selesaikan soal tersebut menggunakan klausa USING

1.2.2.2 Langkah Penyelesaian

Untuk menyelesaikan permasalahan diatas, kita harus menentukan table yang menyimpan informasi yang dibutuhkan:

1. Table yang menyimpan informasi nama departemen adalah table departments
2. Table yang menyimpan informasi alamat lengkap (nama jalan, kode pos, kota, dan provinsi) adalah table locations
3. Untuk mendapatkan informasi dari kedua table tersebut, kita harus melakukan join kepada table departments dan locations melalui column location_id.

1.2.2.3 SQL Command

Untuk menyelesaikan persoalan diatas, digunakan SQL Statement sebagai berikut:

```
SELECT department_name,street_address,postal_code,city,state_province
FROM departments
JOIN locations
USING (location_id)
```

Hasil eksekusi SQL Command di atas adalah menyajikan informasi sebagai berikut:

| DEPARTMENT_NAME | STREET_ADDRESS | POSTAL_CODE | CITY | STATE_PROVINCE |
|--------------------|---------------------|-------------|---------------------|----------------|
| IT | 2014 Jabberwocky Rd | 26192 | Southlake | Texas |
| Shipping | 2011 Interiors Blvd | 99236 | South San Francisco | California |
| Administration | 2004 Charade Rd | 98199 | Seattle | Washington |
| Purchasing | 2004 Charade Rd | 98199 | Seattle | Washington |
| Executive | 2004 Charade Rd | 98199 | Seattle | Washington |
| Finance | 2004 Charade Rd | 98199 | Seattle | Washington |
| Accounting | 2004 Charade Rd | 98199 | Seattle | Washington |
| Treasury | 2004 Charade Rd | 98199 | Seattle | Washington |
| Corporate Tax | 2004 Charade Rd | 98199 | Seattle | Washington |
| Control And Credit | 2004 Charade Rd | 98199 | Seattle | Washington |

1.2.3 Klausa ON

Klausa ketiga yang digunakan untuk melakukan join adalah klausa ON. Pada prinsipnya, ketika melakukan penggabungan beberapa table, informasi yang ingin dihasilkan adalah gabungan informasi dari table-table yang digabungkan dan memenuhi kesesuaian nilai pada column yang digunakan sebagai penghubung. Identik dengan penggunaan klausa USING, pada klausa ON kita juga harus menentukan secara eksplisit pasangan column yang menghubungkan kedua table. Bedanya, jika pada klausa USING, kita hanya bisa menggunakan sebuah column yang memiliki nama yang sama, dengan menggunakan klausa USING, jumlah column yang berpasangan dapat lebih dari satu dan memiliki nama column yang berbeda.

1.2.3.1 Soal

Tampilkan nama departemen dan alamat lengkap departemen tersebut. Alamat lengkap terdiri atas nama jalan, kode pos, kota, dan provinsi.

Selesaikan soal tersebut menggunakan klausa ON

1.2.3.2 Langkah Penyelesaian

Untuk menyelesaikan permasalahan diatas, kita harus menentukan table yang menyimpan informasi yang dibutuhkan:

1. Table yang menyimpan informasi nama departemen adalah table departments
2. Table yang menyimpan informasi alamat lengkap (nama jalan, kode pos, kota, dan provinsi) adalah table locations
3. Untuk mendapatkan informasi dari kedua table tersebut, kita harus melakukan join kepada table departments dan locations melalui column location_id, dimana informasi yang akan disajikan hanyalah yang memenuhi kondisi yaitu nilai column location_id pada kedua table sama.

1.2.3.3 SQL Command

Untuk menyelesaikan persoalan diatas, digunakan SQL Statement sebagai berikut:

```
SELECT department_name, street_address, postal_code, city, state_province
FROM departments
JOIN locations
ON departments.location_id = locations.location_id
```

Hasil eksekusi SQL Command di atas adalah menyajikan informasi sebagai berikut:

| DEPARTMENT_NAME | STREET_ADDRESS | POSTAL_CODE | CITY | STATE_PROVINCE |
|--------------------|---------------------|-------------|---------------------|----------------|
| IT | 2014 Jabberwocky Rd | 26192 | Southlake | Texas |
| Shipping | 2011 Interiors Blvd | 99236 | South San Francisco | California |
| Administration | 2004 Charade Rd | 98199 | Seattle | Washington |
| Purchasing | 2004 Charade Rd | 98199 | Seattle | Washington |
| Executive | 2004 Charade Rd | 98199 | Seattle | Washington |
| Finance | 2004 Charade Rd | 98199 | Seattle | Washington |
| Accounting | 2004 Charade Rd | 98199 | Seattle | Washington |
| Treasury | 2004 Charade Rd | 98199 | Seattle | Washington |
| Corporate Tax | 2004 Charade Rd | 98199 | Seattle | Washington |
| Control And Credit | 2004 Charade Rd | 98199 | Seattle | Washington |

1.2.4 Membatasi Data pada Join

Pada penggabungan table melalui join, kita juga dapat menambahkan kondisi (syarat) tambahan yang harus dipenuhi pada saat menyajikan informasi. Klausa yang digunakan untuk menambahkan kondisi tersebut adalah klausa WHERE dan AND.

Klausa WHERE dan AND dapat digunakan untuk membatasi informasi yang disajikan melalui join menggunakan klausa ON, sedangkan join yang menggunakan klausa NATURAL JOIN dan USING hanya dapat menggunakan klausa WHERE untuk membatasi penyajian informasinya.

1.2.4.1 Soal

Tampilkan nama departemen dan alamat lengkap departemen tersebut. Alamat lengkap terdiri atas nama jalan, kode pos, kota, dan provinsi hanya untuk departemen yang berada di kota Suothlake

1.2.4.2 Langkah Penyelesaian

Untuk menyelesaikan permasalahan diatas, kita harus menentukan table yang menyimpan informasi yang dibutuhkan:

1. Table yang menyimpan informasi nama departemen adalah table departments
2. Table yang menyimpan informasi alamat lengkap (nama jalan, kode pos, kota, dan provinsi) adalah table locations
3. Untuk mendapatkan informasi dari kedua table tersebut, kita harus melakukan join kepada table departments dan locations
4. Untuk membatasi penyajian informasi perlu ditambahkan kondisi dimana city='Southlake'

1.2.4.3 SQL Command

SQL Command yang dapat digunakan untuk menyelesaikan soal diatas adalah:

1. Menggunakan klausa NATURAL JOIN

```
SELECT department_name,street_address,postal_code,city,
       state_province
FROM   departments
NATURAL JOIN locations
WHERE  city = 'Southlake'
```


2. Menggunakan klausa KLAUSA USING

```
SELECT department_name, street_address, postal_code, city,
       state_province
FROM departments
JOIN locations
USING (location_id)
WHERE city = 'Southlake'
```

3. Menggunakan klausa KLAUSA ON

```
SELECT department_name, street_address, postal_code, city,
       state_province
FROM departments
JOIN locations
ON departments.location_id = locations.location_id
WHERE city = 'Southlake'
```

Atau

```
SELECT department_name, street_address, postal_code, city,
       state_province
FROM departments
JOIN locations
ON departments.location_id = locations.location_id
AND city = 'Southlake'
```

Eksekusi ketiga SQL Command diatas akan mengembalikan hasil yang sama sebagai berikut:

| DEPARTMENT_NAME | STREET_ADDRESS | POSTAL_CODE | CITY | STATE_PROVINCE |
|-----------------|---------------------|-------------|-----------|----------------|
| IT | 2014 Jabberwocky Rd | 26192 | Southlake | Texas |

1.2.5 Menggunakan Group Function pada Join

1.2.5.1 Soal

Tampilkan nama kota dan jumlah departemen yang berada pada kota tersebut.

1.2.5.2 Langkah Penyelesaian

Untuk menyelesaikan permasalahan diatas, kita harus menentukan table yang menyimpan informasi yang dibutuhkan:

1. Table yang menyimpan informasi nama departemen adalah table departments
2. Table yang menyimpan informasi nama kota adalah table locations
3. Untuk mendapatkan informasi dari kedua table tersebut, kita harus melakukan join kepada table departments dan locations
4. Untuk menghitung jumlah departemen, digunakan group function count dan mengelompokkan data menggunakan GROUP BY berdasar nama kota

1.2.5.3 SQL Command

SQL Statement yang digunakan untuk menyelesaikan soal diatas adalah:

```
SELECT city,count(department_id) as "Jml Departemen"  
FROM departments  
JOIN locations  
USING (location_id)  
GROUP BY city
```

Hasil eksekusi SQL Command di atas adalah menyajikan informasi sebagai berikut:

| CITY | Jml Departemen |
|---------------------|----------------|
| London | 1 |
| Munich | 1 |
| Oxford | 1 |
| Seattle | 21 |
| South San Francisco | 1 |
| Southlake | 1 |
| Toronto | 1 |