



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA YOGYAKARTA**

**MODUL 10
BAHASA QUERY**

Manipulasi Data

A. TUJUAN

Mahasiswa dapat:

1. Menjelaskan setiap perintah *Data Manipulation Language (DML)*.
2. Menyisipkan baris dalam tabel.
3. Mengupdate baris dalam tabel.
4. Menghapus baris dalam tabel.

B. LANDASAN TEORI & LANGKAH PRAKTIKUM

Untuk mendukung latihan pada modul ini, setelah login ke basisdata Anda, lakukan perintah di bawah ini untuk membuat tabel yang dapat Anda gunakan untuk latihan:

```
CREATE TABLE copy_employees AS SELECT * FROM employees;
CREATE TABLE copy_departments AS SELECT * FROM departments;
```

Data Manipulation Language (DML) adalah bagian inti dalam SQL. Perintah DML diperlukan ketika kita melakukan hal-hal berikut:

- a. Menambah baris baru dalam tabel
- b. Memodifikasi baris yang ada dalam tabel
- c. Menghapus baris yang ada dalam tabel

Sekumpulan perintah DML yang membentuk satu unit logikal disebut **transaksi**.

B.1. Menyisipkan baris baru ke dalam tabel dengan perintah INSERT

Gambar 10.1 berikut ini mengilustrasikan penambahan baris baru dalam tabel DEPARTMENTS.

Menambah baris baru dalam tabel

75	Information Technology	100	1700
----	------------------------	-----	------

Baris baru

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	2000	1700
20	Manufacturing	2000	1700
30	Marketing	2000	1700
40	Human Resources	2000	1700
50	Finance	2000	1700
60	IT	2000	1700
70	Public Relations	2000	1700
80	Legal	2000	1700
90	Compliance	2000	1700
100	Procurement	2000	1700
110	Accounting	2000	1700
120	Corporate Risk	2000	1700

Menambah baris baru ke dalam tabel DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	2000	1700
20	Manufacturing	2000	1700
30	Marketing	2000	1700
40	Human Resources	2000	1700
50	Finance	2000	1700
60	IT	2000	1700
70	Public Relations	2000	1700
80	Legal	2000	1700
90	Compliance	2000	1700
100	Procurement	2000	1700
110	Accounting	2000	1700
120	Corporate Risk	2000	1700
75	Information Technology	100	1700

Gambar 10.1. Ilustrasi Penambahan Baris Baru dalam Tabel (© Oracle 2004)

B.1.1. Sintaks Perintah INSERT

Untuk menyisipkan baris baru dalam tabel, digunakan perintah INSERT. Sintaks dari perintah INSERT adalah sebagai berikut:

```
INSERT INTO table [(column [, column....])]
VALUES (value [, value....]);
```

Keterangan:

- *table* adalah nama tabel
- *column* adalah nama kolom dalam tabel yang akan diisi datanya
- *value* adalah nilai dari kolom tersebut
- Perintah INSERT dengan klausa VALUES ini hanya dipakai untuk menyisipkan **satu baris data dalam satu waktu**.

B.1.2. Penggunaan Perintah INSERT

Contoh 1

- Menyisipkan sebuah baris baru yang berisi nilai tertentu ke dalam tabel DEPARTMENTS dengan menyebut nama kolom.

```
INSERT INTO COPY_DEPARTMENTS(department_id, department_name,
                               manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
```

Dalam perintah di atas, nilai-nilai yang mengikuti klausa VALUES akan dimasukkan ke dalam kolom yang bersesuaian secara urut. Gunakan tanda petik tunggal ‘ ‘ untuk menuliskan nilai karakter dan tanggal. Sedangkan untuk nilai numerik tidak perlu memakai tanda ‘ ‘. Jika tanda petik ditambahkan pada data numerik, maka data tersebut akan dikonversi menjadi karakter. Cek hasil penyisipan data tersebut dengan menggunakan perintah:

```
Select * from COPY_DEPARTMENTS;
```

Kita dapat menggunakan perintah INSERT hanya dengan menyebut nama tabel yang akan kita sisipi data, **tanpa menyebut nama kolom** tempat kita memasukkan nilai. Akan tetapi, nilai-nilai yang dimasukkan mengikuti klausa VALUES harus dituliskan dalam urutan default sesuai struktur tabel, dan setiap kolom harus diisi nilai tertentu (bisa juga NULL). Untuk memastikan urutan default kolom, struktur tabel bisa dilihat menggunakan perintah DESCRIBE.

Contoh 2

- Misal dengan perintah DESC COPY_DEPARTMENTS kita mendapatkan struktur tabel DEPARTMENTS sebagai berikut:

```
desc copy_departments;
+-----+-----+-----+
Name          Null    Type
+-----+-----+-----+
DEPARTMENT_ID          NUMBER(4)
DEPARTMENT_NAME        NOT NULL VARCHAR2(30)
MANAGER_ID              NUMBER(6)
LOCATION_ID               NUMBER(4)
4 rows selected
```

Untuk menyisipkan sebuah baris baru yang berisi nilai tertentu ke dalam tabel COPY_DEPARTMENTS tanpa menyebut nama kolom, digunakan perintah berikut:

```
INSERT INTO COPY_DEPARTMENTS
VALUES (65, 'Marketing', 110, 1600);
```

Cobalah cek hasilnya dengan perintah `select`. Dengan perintah di atas, maka nilai 60 akan dimasukkan dalam kolom `department_id`, 'Marketing' dimasukkan dalam kolom `department_name`, 110 ke dalam `manager_id`, dan 1600 ke dalam `location_id`.

B.1.3. Penyisipan baris baru yang memiliki nilai NULL

Untuk memasukkan nilai NULL ke dalam tabel, dapat dilakukan dengan dua cara berikut:

- Implisit:** dilakukan dengan mengabaikan (tidak menuliskan) kolom yang akan diisi NULL
- Eksplisit:** dilakukan dengan menspesifikasikan keyword NULL dalam daftar VALUES, atau menspesifikasikan string kosong (") dalam daftar VALUES untuk data bertipe string dan date.

Contoh 3:

- Menyisipkan sebuah baris baru yang berisi nilai NULL ke dalam tabel COPY_DEPARTMENTS secara implisit.

```
INSERT INTO COPY_DEPARTMENTS(department_id, department_name)
VALUES (30, 'Purchasing');
```

Cobalah cek hasilnya dengan perintah `select`.

Contoh 4:

- Menyisipkan sebuah baris baru yang berisi nilai NULL ke dalam tabel COPY_DEPARTMENTS secara eksplisit.

```
INSERT INTO COPY_DEPARTMENTS
VALUES (100, 'Finance', NULL, NULL);
```

Cobalah cek hasilnya dengan perintah `select`.

B.1.4. Penyisipan nilai-nilai khusus

Untuk memasukkan nilai-nilai khusus, digunakan fungsi (*function*). Sebagai contoh untuk memasukkan tanggal dan waktu sekarang dapat digunakan fungsi `SYSDATE()`.

Contoh 5:

- Menyisipkan sebuah baris baru untuk employee bernama Louis Popp ke dalam tabel COPY_EMPLOYEES. Nilai `hire_date` diisi tanggal sekarang yang diambil dengan fungsi `SYSDATE`. Verifikasilah hasil insert dengan mengecek isi tabel.

```

INSERT INTO COPY_EMPLOYEES (employee_id,
                             first_name, last_name,
                             email, phone_number,
                             hire_date, job_id, salary,
                             commission_pct, manager_id,
                             department_id)
VALUES (113,
        'Louis', 'Popp',
        'LPOPP', '515.124.4567',
        SYSDATE, 'AC_ACCOUNT', 6900,
        NULL, 205, 100);

```

B.2. Menyalin (*copy*) baris dari tabel yang lain

Untuk menambah baris ke dalam tabel dengan isi yang disalin dari tabel lain, gunakan *subquery* dengan sintaks sbb:

```
INSERT INTO table [ (column [, column....])] subquery;
```

Keterangan:

- *table* adalah nama tabel
- *column* adalah nama kolom dalam tabel yang akan diisi datanya
- *subquery* adalah *subquery* yang mengembalikan baris-baris data
- Jumlah kolom dan tipe data dalam daftar kolom yang mengikuti klausa INSERT harus sesuai dengan jumlah nilai dan tipe data hasil *subquery*.
- Jangan gunakan klausa VALUES.

Contoh 6:

- Menyisipkan baris-baris baru hasil *subquery* ke dalam tabel SALES_REPS. Cek hasilnya dengan perintah select. **Sebelumnya, buatlah lebih dahulu tabel SALES_REPS dengan menggunakan perintah berikut ini.**

```

CREATE TABLE SALES_REPS
(id_name NUMBER(6),
 name     VARCHAR2(25),
 salary   NUMBER(8,2),
 commission_pct NUMBER(2,2));

```

```

INSERT INTO SALES_REPS(id_name, name, salary, commission_pct)
SELECT employee_id, last_name, salary, commission_pct
FROM employees
WHERE job_id LIKE '%REP%';

```

B.3. Mengubah data dalam tabel

Kita dapat mengubah data dalam tabel seperti ilustrasi dalam gambar 10.2 berikut ini, yang mengubah `department_id` dari seluruh employee dengan `department_id` 60 menjadi 30.

Mengubah data dalam tabel

EMPLOYEES

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	DEPARTMENT_ID	JOB_ID
1	King	17-SEP-07	90 AD_PRES	
2	Kochhar	21-SEP-08	90 AD_VP	
3	De Haan	13-MAR-03	90 AD_VP	
4	Hudnell	03-MAR-00	60 IT_PROG	
5	Smith	21-MAY-01	60 IT_PROG	
6	Austin	25-JUN-07	60 IT_PROG	
7	Patel	05-FEB-08	60 IT_PROG	
8	Lorel	07-FEB-08	60 IT_PROG	
9	Greenberg	17-AUG-04	100 FI_MGR	
10	Favet	16-AUG-04	100 FI_ACCOUNT	
11	Chen	28-SEP-07	100 FI_ACCOUNT	
12	Serra	30-SEP-07	100 FI_ACCOUNT	

Update baris data dalam tabel EMPLOYEES

EMPLOYEE_ID	LAST_NAME	HIRE_DATE	DEPARTMENT_ID	JOB_ID
1	King	17-SEP-07	90 AD_PRES	
2	Kochhar	21-SEP-08	90 AD_VP	
3	De Haan	13-MAR-03	90 AD_VP	
4	Hudnell	03-MAR-00	30 IT_PROG	
5	Smith	21-MAY-01	30 IT_PROG	
6	Austin	25-JUN-07	30 IT_PROG	
7	Patel	05-FEB-08	30 IT_PROG	
8	Lorel	07-FEB-08	30 IT_PROG	
9	Greenberg	17-AUG-04	100 FI_MGR	
10	Favet	16-AUG-04	100 FI_ACCOUNT	
11	Chen	28-SEP-07	100 FI_ACCOUNT	
12	Serra	30-SEP-07	100 FI_ACCOUNT	

Baris dalam Tabel

© Oracle 2004

B.3.1. Sintaks Perintah UPDATE

Untuk mengubah (modifikasi) baris dalam tabel, digunakan perintah UPDATE. Perintah ini dapat digunakan untuk mengupdate lebih dari satu baris. Sintaks perintah UPDATE adalah sebagai berikut:

```
UPDATE    table
SET       column = value [, column = value, ...]
[WHERE    condition];
```

Keterangan:

- *table* adalah nama tabel
- *column* adalah nama kolom dalam tabel yang akan diisi datanya
- *value* adalah nilai atau *subquery* dari kolom terkait
- *condition* adalah baris-baris yang akan diupdate dan terdiri dari nama kolom, konstanta, *subquery*, dan operator pembanding.
- Jika klausa WHERE tidak digunakan, maka semua baris dalam tabel akan dimodifikasi.
- Sebaiknya gunakan kunci primer untuk mengidentifikasi suatu baris. Jika digunakan atribut bukan kunci primer, ada kemungkinan duplikat.

B.3.2. Penggunaan Perintah UPDATE

Contoh 7

- Mengupdate nilai `department_id` menjadi 70, untuk employee dengan `employee_id` = 113. Cobalah cek hasilnya dengan menampilkan baris dari `employee_id` = 113.

```
UPDATE copy_employees
SET    department_id = 70
WHERE  employee_id = 113;
```

Contoh 8

- Mengupdate semua baris dalam `COPY_EMPLOYEES` sehingga `department_id` = 110. Cobalah cek hasilnya dengan menampilkan isi `department_id` dalam `COPY_EMPLOYEES`.

```
UPDATE copy_employees
SET    department_id = 110;
```

B.3.3. Mengupdate dua kolom dengan *subquery*

Kita dapat mengupdate beberapa kolom dengan menggunakan klausa `SET` dalam perintah `UPDATE`, dengan memakai *subquery*. Jika tidak ada baris yang diupdate, maka muncul pesan “0 rows updated”. Sintaks untuk mengupdate dua kolom dengan *subquery* adalah sebagai berikut:

```
UPDATE    table
SET       column =
           (SELECT column
            FROM   table
            WHERE  condition)
        [ ,
          column =
           (SELECT column
            FROM   table
            WHERE  condition)
        [WHERE condition ] ;
```

Contoh 9

- Mengupdate job dan salary dari employee dengan `employee_id` = 114 , dengan job dan salary yang sama dengan job dan salary dari employee 205. Cek hasil update dengan menampilkan isi record dari `employee_id` 114 dan 205.

```

UPDATE copy_employees
SET    job_id = (SELECT job_id
                  FROM    copy_employees
                  WHERE   employee_id = 205),
       salary = (SELECT salary
                  FROM    copy_employees
                  WHERE   employee_id = 205)
WHERE  employee_id = 114;

```

B.4. Menghapus data dari tabel

Kita dapat menghapus baris-baris data dari dalam tabel. Gambar 10.3 berikut ini mengilustrasikan penghapusan department Finance dari tabel DEPARTMENTS (asumsi: tidak ada kendala/*constraints* yang didefinisikan dalam tabel DEPARTMENTS).

Menghapus data dalam tabel

DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10 Administration	200	1700
2	20 Marketing	201	1800
3	30 Purchasing	114	1700
4	40 Human Resources	203	2400
5	50 Shipping	121	1500
6	60 IT	103	1400
7	70 Public Relations	204	2700
8	80 Sales	145	2500
9	90 Executive	100	1700
10	100 Finance	106	1700
11	110 Accounting	205	1700
12	120 Treasury	(null)	1700

Menghapus satu baris data dari tabel DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10 Administration	200	1700
2	20 Marketing	201	1800
3	30 Purchasing	114	1700
4	40 Human Resources	203	2400
5	50 Shipping	121	1500
6	60 IT	103	1400
7	70 Public Relations	204	2700
8	80 Sales	145	2500
9	90 Executive	100	1700
10	100 Finance	106	1700
11	110 Accounting	205	1700
12	120 Corporate Tax	(null)	1700

Gambar 10.3. Ilustrasi Menghapus Data dari Tabel
© Oracle 2004

B.4.1 Sintaks Perintah DELETE

Untuk menghapus baris dalam tabel, digunakan perintah DELETE. Sintaks perintah DELETE adalah sebagai berikut:

```

DELETE [FROM]      table
[WHERE             condition];

```

Keterangan:

- *table* adalah nama tabel
- *condition* mengidentifikasi baris-baris yang akan dihapus dan terdiri dari nama kolom, konstanta, *subquery*, dan operator pembandingan.
- Jika tidak memakai klausa WHERE, maka akan menghapus semua baris dalam tabel tersebut.
- Jika tidak ada baris yang dihapus, maka akan muncul pesan "0 rows deleted".

B.4.2. Penggunaan Perintah DELETE

Contoh 10

- Menghapus baris dari COPY_EMPLOYEES untuk data employee yang memiliki employee_id = 114. Cobalah cek hasilnya dengan menampilkan baris data yang sudah Anda hapus.

```
DELETE FROM copy_employees
WHERE employee_id = 114;
```

B.4.3. Menghapus baris berdasarkan tabel yang lain

Kita dapat menggunakan *subquery* untuk menghapus baris berdasar nilai dari tabel yang lain.

Contoh 11

- Menghapus semua employee dari tabel COPY_EMPLOYEES yang bekerja di department yang department_name nya memuat string "Public". *Subquery* akan mencari terlebih dahulu dari tabel DEPARTMENTS untuk mencari department_id dari department yang memiliki department_name memuat string "Public". Hasil *subquery* kemudian dikirimkan ke *main query*, yang kemudian menghapus baris data dari tabel EMPLOYEES berdasar pada department_id. Cek hasilnya dengan menampilkan isi tabel COPY_EMPLOYEES.

```
DELETE FROM copy_employees
WHERE department_id =
      (SELECT department_id
       FROM departments
       WHERE department_name LIKE '%Public%');
```

C. TUGAS

1. Buatlah sebuah tabel bernama MY_EMPLOYEES dengan mengetik dan menjalankan script berikut. Simpan script dalam file lab_10_01.sql.

```
CREATE TABLE my_employees (
    id          NUMBER(4)
        CONSTRAINT emp_id_pk          PRIMARY KEY
    ,first_name  VARCHAR2(20)
    ,last_name   VARCHAR2(25)
    ,userid      VARCHAR2(8)
    ,salary      NUMBER(8,2));
```


2. Tampilkan struktur tabel MY_EMPLOYEES yang baru saja Anda buat.
3. Diketahui terdapat sejumlah data sebagai berikut:

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dance	Betty	bdance	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	aropebur	1550

Tambahkan **baris pertama** dari tabel di atas ke dalam tabel MY_EMPLOYEES **tanpa** menuliskan daftar nama kolomnya. Cek hasilnya dengan menampilkan isi tabel.

4. Tambahkan **baris kedua** dari tabel di atas ke dalam tabel MY_EMPLOYEES **dengan** menuliskan daftar nama kolomnya. Cek hasilnya dengan menampilkan isi tabel.
5. Tambahkan **tiga baris data berikutnya** dari tabel di atas ke dalam tabel MY_EMPLOYEES **dengan** menuliskan daftar nama kolomnya. Cek hasilnya dengan menampilkan isi tabel.
6. Ubahlah last_name dari employee ke-3 dalam tabel MY_EMPLOYEES menjadi Drexler. Cek hasilnya dengan menampilkan isi tabel.
7. Untuk semua employee dalam tabel MY_EMPLOYEES yang salarynya lebih kecil dari 900, ubahlah salary menjadi 1000. Cek hasil perubahan dengan menampilkan isi tabel.
8. Hapuslah employee Betty Dancs dari tabel MY_EMPLOYEES. Cek hasil penghapusan dengan menampilkan isi tabel.

D. DAFTAR PUSTAKA

1. *Oracle Database 10g : SQL Fundamental I*, Oracle Inc. 2004

☺☺☺ *You can if you think you can* !☺☺☺