

# **Oracle Academic Initiative**

## **Oracle9i Introduction to SQL**



**Oleh:**

**Tessy Badriyah, SKom.MT**

**Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember  
Surabaya**

## BAB 20 : Oracle9i Perluasan ke Statement DML dan DDL

### 20.1. Sasaran

- Dapat menggambarkan fitur dari perintah insert pada banyak table (multitable insert)
- Dapat menggunakan tipe-tipe insert pada banyak table berikut :
  - Unconditional INSERT
  - Pivoting INSERT
  - Conditional ALL INSERT
  - Conditional FIRST INSERT
- Dapat membuat dan menggunakan tabel eksternal
- Memberi nama indeks pada waktu pembuatan primary key constraint

### 20.2. Review : Statement INSERT

Menambahkan baris baru ke dalam tabel dengan menggunakan statement INSERT.

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

Hanya satu baris yang bisa ditambahkan setiap kali menggunakan perintah ini.

```
INSERT INTO departments(department_id, department_name,
                        manager_id, location_id)
VALUES (70, 'Public Relations', 100, 1700);
1 row created.
```

### 20.3. Review : Statement UPDATE

Untuk memodifikasi baris yang sudah ada dengan perintah UPDATE.

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

UPDATE dapat juga dilakukan pada lebih dari satu baris setiap kali diberikan perintah. Beberapa baris akan dimodifikasi sesuai dengan kondisi yang memenuhi pada klausa WHERE.

```
UPDATE employees
SET department_id = 70
WHERE employee_id = 142;
1 row updated.
```

### 20.4. Pendahuluan : Statement INSERT untuk Banyak Tabel

Statement INSERT ... SELECT dapat digunakan untuk menyisipkan baris ke dalam banyak tabel sebagai bagian dari statement DML tunggal. Multitable INSERT dapat digunakan pada sistem datawarehousing untuk mentransfer data dari satu atau lebih sumber operasional ke dalam himpunan tabel sasaran. Perintah ini menyediakan peningkatan performansi atas :

- DML tunggal vs. multiple INSERT...SELECT Statement
- DML tunggal vs. prosedur untuk melakukan multiple insert dengan menggunakan IF...THEN

### 20.5. Tipe dari Statement INSERT untuk Banyak Tabel

Oracle9i menyediakan bermacam tipe untuk multitable insert, sebagai berikut :

- Unconditional INSERT
- Pivoting INSERT
- Conditional ALL INSERT
- Conditional FIRST INSERT

## 20.6. Statement INSERT untuk Banyak Tabel

Sintak :

```
INSERT [ALL] [conditional_insert_clause]
[insert_into_clause values_clause] (subquery)
```

Conditional\_insert\_clause

```
[ALL] [FIRST]
[WHEN condition THEN] [insert_into_clause values_clause]
[ELSE] [insert_into_clause values_clause]
```

## 20.7. Unconditional INSERT ALL

Pilih EMPLOYEE\_ID, HIRE\_DATE, SALARY dan MANAGER\_ID dari tabel EMPLOYEES untuk pegawai yang EMPLOYEE\_ID nya lebih besar dari 200. Kemudian masukkan nilainya ke dalam tabel SAL\_HISTORY dan MGR\_HISTORY dengan menggunakan multitable INSERT.

```
INSERT ALL
  INTO sal_history VALUES (EMPID, HIREDATE, SAL)
  INTO mgr_history VALUES (EMPID, MGR, SAL)
  SELECT employee_id EMPID, hire_date HIREDATE,
         salary SAL, manager_id MGR
  FROM employees
  WHERE employee_id > 200;
8 rows created.
```

## 20.8. Conditional INSERT ALL

Pilih EMPLOYEE\_ID, HIRE\_DATE, SALARY dan MANAGER\_ID dari tabel EMPLOYEES untuk pegawai yang EMPLOYEE\_ID nya lebih besar dari 200. Jika salary-nya lebih dari \$10,000, sisipkan nilainya ke dalam tabel SAL\_HISTORY dengan menggunakan conditional multitable INSERT. Jika MANAGER\_ID lebih dari 200, sisipkan nilainya ke dalam tabel MGR\_HISTORY dengan menggunakan conditional multitable INSERT.

```
INSERT ALL
  WHEN SAL > 10000 THEN
    INTO sal_history VALUES (EMPID, HIREDATE, SAL)
  WHEN MGR > 200 THEN
    INTO mgr_history VALUES (EMPID, MGR, SAL)
  SELECT employee_id EMPID, hire_date HIREDATE,
         salary SAL, manager_id MGR
  FROM employees
  WHERE employee_id > 200;
4 rows created.
```

## 20.9. Conditional FIRST INSERT

Pilih DEPARTMENT\_ID, SUM(SALARY) dan MAX(HIRE\_DATE) dari tabel EMPLOYEE. Jika SUM(SALARY) lebih dari \$25,000 maka masukkan nilainya ke dalam SPECIAL\_SAL, dengan menggunakan conditional FIRST multitable INSERT. Jika klausa WHEN pertama kali dievaluasi = true, maka klausa WHEN berikutnya untuk baris ini akan dilompati. Untuk baris yang tidak

memenuhi kondisi WHEN pertama, masukkan nilainya ke dalam tabel HIREDATE\_HISTORY\_00, atau HIREDATE\_HISTORY\_99, atau HIREDATE\_HISTORY, berdasarkan nilai kolom HIRE\_DATE dengan menggunakan conditional multitable INSERT.

```

INSERT FIRST
  WHEN SAL > 25000 THEN
    INTO special_sal VALUES (DEPTID, SAL)
  WHEN HIREDATE like ('%00%') THEN
    INTO hiredate_history_00 VALUES (DEPTID, HIREDATE)
  WHEN HIREDATE like ('%99%') THEN
    INTO hiredate_history_99 VALUES (DEPTID, HIREDATE)
  ELSE
    INTO hiredate_history VALUES (DEPTID, HIREDATE)
SELECT department_id DEPTID, SUM(salary) SAL,
       MAX(hire_date) HIREDATE
FROM   employees
GROUP BY department_id;
8 rows created.

```

### 20.10. Pivoting INSERT

Misal kita dapatkan himpunan data sales dari tabel-tabel nonrelational database Yaitu SALES\_SOURCE\_DATA dengan struktur :

EMPLOYEE\_ID, WEEK\_ID, SALES\_MON, SALES\_TUE, SALES\_WED, SALES\_THUR, SALES\_FRI.

Kemudian kita ingin menyimpan data ini ke dalam tabel SALES\_INFO dengan format relasional yaitu EMPLOYEE\_ID, WEEK, SALES

Dengan menggunakan pivoting INSERT, kita bisa mengkonversi dari nonrelational database tabel ke dalam format relasional.

```

INSERT ALL
  INTO sales_info VALUES (employee_id, week_id, sales_MON)
  INTO sales_info VALUES (employee_id, week_id, sales_TUE)
  INTO sales_info VALUES (employee_id, week_id, sales_WED)
  INTO sales_info VALUES (employee_id, week_id, sales_THUR)
  INTO sales_info VALUES (employee_id, week_id, sales_FRI)
SELECT EMPLOYEE_ID, week_id, sales_MON, sales_TUE,
       sales_WED, sales_THUR, sales_FRI
FROM   sales_source_data;
5 rows created.

```

### 20.11. Tabel Eksternal

Tabel eksternal bersifat *read-only* dimana data yang disimpan berada diluar database dalam bentuk flat file. Metadata untuk tabel eksternal dibuat dengan menggunakan Statement CREATE TABLE. Dengan bantuan tabel eksternal, data Oracle dapat disimpan atau di-unload sebagai flat file. Data dapat dilakukan query dengan menggunakan SQL, tapi kita tidak bisa menggunakan DML Statement dan tidak ada indeks yang bisa dibuat.

### 20.12. Pembuatan Tabel Eksternal

Gunakan external\_table\_clause pada sintak CREATE TABLE untuk membuat tabel eksternal. Tentukan ORGANIZATION sebagai EXTERNAL yang menunjukkan bahwa tabel tersebut ditempatkan diluar database. External\_table\_clause berisi TYPE dari access driver, external\_data\_properties, dan REJECT LIMIT. External\_data\_properties berisi : DEFAULT\_DIRECTORY, ACCESS\_PARAMETER dan LOCATION.

### 20.13. Contoh Pembuatan Tabel Eksternal

Buat object DIRECTORY yang berkorespondensi dengan directory yang ada pada file system dimana external data source ditempatkan

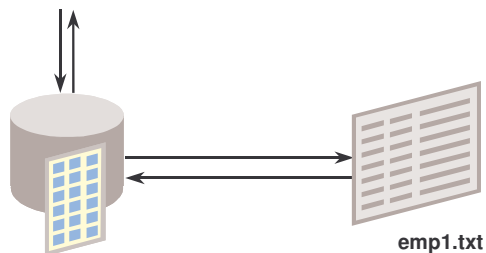
```
CREATE DIRECTORY emp_dir AS '/flat_files' ;
```

Contoh pembuatan tabel eksternal :

```
CREATE TABLE oldemp (
  empno NUMBER, empname CHAR(20), birthdate DATE)
  ORGANIZATION EXTERNAL
  (TYPE ORACLE_LOADER
  DEFAULT DIRECTORY emp_dir
  ACCESS PARAMETERS
  (RECORDS DELIMITED BY NEWLINE
  BADFILE 'bad_emp'
  LOGFILE 'log_emp'
  FIELDS TERMINATED BY ','
  (empno CHAR,
  empname CHAR,
  birthdate CHAR date_format date mask "dd-mon-yyyy"))
  LOCATION ('emp1.txt'))
  PARALLEL 5
  REJECT LIMIT 200;
Table created.
```

### 20.14. Query pada Tabel Eksternal

```
SELECT *
FROM oldemp
```



### 20.15. CREATE INDEX dengan Statement CREATE TABLE

```
CREATE TABLE NEW_EMP
(employee_id NUMBER(6)
  PRIMARY KEY USING INDEX
  (CREATE INDEX emp_id_idx ON
  NEW_EMP(employee_id)),
first_name VARCHAR2(20),
last_name VARCHAR2(25));
Table created.
```

```
SELECT INDEX_NAME, TABLE_NAME
FROM USER_INDEXES
WHERE TABLE_NAME = 'NEW_EMP';
```

INDEX_NAME	TABLE_NAME
EMP_ID_IDX	NEW_EMP

**20.16. Latihan**