

Introduction to Oracle9i : SQL

Chapter15. DML과 DDL문의 확장

- ▶ 다중테이블 INSERT 기능을 기술한다.
- ▶ 다중테이블 INSERT의 종류를 파악하고 상황별 다중테이블 INSERT 문장을 수행한다.

1. INSERT문

- 테이블에 새로운 행을 입력할 때 사용
- 하나의 INSERT문을 통해서 한 행만 입력 가능

(1) INSERT 구문

```
INSERT INTO 테이블명 [(컬럼명, 컬럼명...)]  
VALUES      (값, 값...);
```

```
INSERT INTO departments (department_id, department_name, manager_id)  
VALUES      (120, 'CRM', 100);
```

1행이 입력되었습니다.

2. UPDATE문

- 테이블에 존재하는 행의 내용을 변경할 때 사용
- 하나의 UPDATE문을 통해서 한 행 이상의 행 변경 가능

1) UPDATE 구문

```
UPDATE      테이블명  
SET          컬럼명 = 값  
[WHERE      조건 ] ;
```

```
UPDATE employees  
SET      department_id = 80  
WHERE    employee_id = 100 ;
```

1행이 변경되었습니다.

3. 다중 테이블 INSERT

(1) 정의 및 특징

- INSERT ...SELECT 문을 사용
- 데이터 웨어하우징 등의 용도로 사용
- 성능 향상의 효과

(2) 구문

```
INSERT [ALL] [조건적인 INSERT 절]  
[INSERT INTO 절 VALUES 절] ( 서브쿼리 )
```

3. 다중 테이블 INSERT

(3) 종류

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

3. 다중 테이블 INSERT

(3) 종류

1) Unconditional INSERT ALL

· 여러 테이블에 동시에 무조건 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 ;
```

3. 다중 테이블 INSERT

(3) 종류

2) Conditional INSERT ALL

- 조건적으로 INSERT 하되 우선 제시된 조건이 이후 조건의 수행에 영향을 끼치지 않음

```
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 ;
```


3. 다중 테이블 INSERT

(3) 종류

3) Conditional FIRST INSERT

- 조건적으로 INSERT하며 우선 제시된 조건에 의해 이미 INSERT된 값은 이후 조건을 만족하더라도 INSERT 불가

```
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 ;
```

3. 다중 테이블 INSERT

(3) 종류

4) Pivoting INSERT

- Unconditional INSERT ALL과 사실상 같음
- 잘못된 컬럼 구조를 갖고 있는 테이블로부터 데이터를 옮길 때 사용

EMPLOYEE_ID	WEEK_ID	SALES_ MON	SALES_ TUE	SALES_ WED	SALES_ THU	SALES_ FRI
176	6	2000	3000	4000	5000	6000

3. 다중 테이블 INSERT

(3) 종류

4) Pivoting INSERT

INSERT ALL

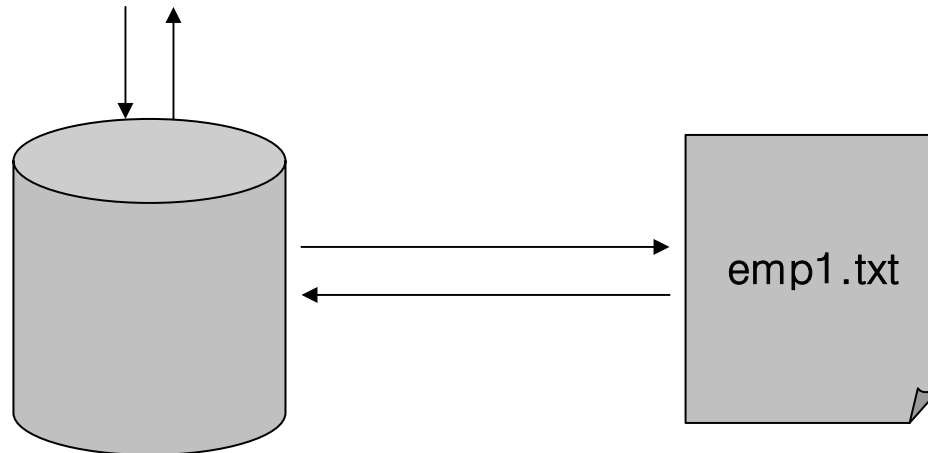
```
INTO sales_info VALUES (employee_id, week_id, sales_MON)
INTO sales_info VALUES (employee_id, week_id, sales_TUE)
INTO sales_info VALUES (employee_id, week_id, sales_WED)
INTO sales_info VALUES (employee_id, week_id, sales_THU)
INTO sales_info VALUES (employee_id, week_id, sales_FRI)
SELECT employee_id, week_id, sales_MON, sales_TUE,
       sales_WED, sales_THU, sales_FRI
FROM   sales_source_data ;
```

4. 외부 테이블

(1) 정의

- 데이터베이스 외부 파일에 저장된 데이터를 읽어들이기 때 사용
- SELECT만 할 수 있는 READ ONLY 테이블

```
SELECT *  
FROM oldemp ;
```



4. 외부 테이블

(2) 외부 테이블 생성

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

```
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 ;
```

5. 테이블 생성 시 인덱스 생성

```
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)) ;
```