

## 9장. 데이터 조작과 트랜잭션

이번 장에서는 새로운 데이터를 테이블에 입력하고 수정하고 삭제하는 명령어를 배웁니다. 그리고 트랜잭션으로 데이터의 일관성을 유지하는 방법을 학습합니다.

## 학습 내용

- ❖ 테이블에 내용을 추가하는 INSERT 문
- ❖ 테이블에 내용을 수정하는 UPDATE 문
- ❖ 테이블에 내용을 삭제하는 DELETE 문
- ❖ 트랜잭션 관리

## 학습목표

- ❖ 테이블에 새로운 내용을 삽입할 수 있습니다.
- ❖ 테이블의 내용을 갱신할 수 있습니다.
- ❖ 테이블의 내용을 삭제할 수 있습니다.
- ❖ 트랜잭션을 제어할 수 있습니다.

# 1. 테이블에 내용을 추가하는 INSERT 문

- ❖ DML(데이터 조작어 : Data Manipulation Language)은 테이블에 새로운 데이터를 삽입하거나 기존의 데이터를 수정, 삭제하기 위한 명령어의 집합입니다.
- ❖ INSERT 문은 테이블에 데이터를 입력하기 위한 명령어입니다.

형식	INSERT INTO table_name (column_name, ... ) VALUES (column_value, ... ) ;
----	---

- 한 번에 하나의 로우만 INTO 뒤에 명시한 테이블에 삽입됩니다.
- 테이블명 뒤에 기술한 칼럼에 VALUES 절에서 지정한 칼럼 값을 순서대로 입력합니다.
- 삽입할 칼럼의 데이터 타입이 문자(Char, VARCHAR2)나 날짜(Date) 타입일 경우, 반드시 작은 따옴표(' ')를 함께 사용해야 합니다.

# 1. 테이블에 내용을 추가하는 INSERT 문

[예제] 실습을 위한 부서테이블을 새로이 생성한 후 ACCOUNTING 부서의 정보를 입력하여 보시다.

예 제

```
CREATE TABLE dept_copy  
AS  
SELECT * FROM department WHERE 0 = 1 ;  
(데이터를 제외한 구조만 복사)
```

예 제

```
INSERT INTO dept_copy  
VALUES      (10, 'ACCOUNTING', 'NEW YORK') ;
```

예 제

```
INSERT INTO dept_copy (dno, loc, dname)  
VALUES      (20, 'DALLAS', 'RESEARCH') ;
```

## 1-1. NULL 값을 갖는 로우 삽입

- ❖ 데이터를 입력하는 시점에서 해당 칼럼 값을 모르거나 확정되지 않았을 경우에는 칼럼의 값으로 NULL을 입력합니다.
- ❖ 암시적 방법은 입력 시 칼럼명 리스트에 해당 칼럼을 생략합니다. 생략한 칼럼에는 NULL이 삽입 됩니다.

예 제

```
INSERT INTO dept_copy (dno, dname)
VALUES      (30, 'SALES') ;
```

- ❖ 명시적 방법은 입력 시 VALUES 절에 명시적으로 NULL을 입력합니다. 문자나 날짜 타입의 경우 공백 문자열("")을 지정할 수 있습니다.

예 제

```
INSERT INTO dept_copy
VALUES      (40, 'OPERATIONS', NULL) ;
```

## 1-1.

- ❖ 날짜 데이터를 입력하기 위해 해당 시스템에서 요구하는 기본 날짜 형식을 사용해야 합니다. oracle의 경우 'YY/MM/DD'의 형식을 따릅니다.

예 제

```
INSERT INTO emp_copy  
VALUES (7000, 'CANDY', 'MANAGER', '2012/05/01', 10);
```

- ❖ TO\_DATE 함수를 사용해서 날짜를 입력하기도 합니다.

예 제

```
SELECT * FROM emp_copy ;
```

## 1-2. 다른 테이블에서 데이터 복사하기

- ❖ INSERT INTO 다음에 VALUES 절 대신 서브쿼리를 사용함으로써 다른 테이블의 데이터를 복사할 수 있습니다.

형 식

```
INSERT INTO table [(column1, column2, ...)]  
subquery ;
```

[예제] 서브 쿼리를 통해 다중행을 입력하여 복사다.

예 제

```
INSERT INTO dept_copy  
SELECT * FROM department ;
```



## 2. 테이블의 내용을 수정하는 UPDATE 문

- ❖ UPDATE 문은 테이블에 저장된 데이터를 수정하기 위한 DML입니다.

형 식

```
UPDATE table_name  
SET column_name1 = value1, column_name2 = value2, ...  
WHERE conditions ;
```

[예제] WHERE 절로 특정 로우만 수정하여 봅시다.

예 제

```
UPDATE dept_copy  
SET dname = 'PROGRAMMING'  
WHERE dno = 10 ;
```

## 2. 테이블의 내용을 수정하는 UPDATE 문

[예제] WHERE 절을 생략하여 모든 로우 수정하여 보시다

예 제

```
UPDATE dept_copy  
SET dname = 'HR';
```

[예제] 칼럼 값 여러 개를 한 번에 수정하여 보시다.

예 제

```
UPDATE dept_copy  
SET dname = 'PROGRAMMING', loc = 'SEOUL'  
WHERE dno = 10 ;
```

## 2-1. 다른 테이블을 기반으로 데이터 수정하기

- ❖ UPDATE 문의 SET 절에서 서브쿼리를 기술하면 서브 쿼리를 수행한 결과로 내용이 변경됩니다.

[예제] 10번 부서의 지역명을 20번 부서의 지역명으로 변경하여 봅시다.

예 제

```
UPDATE dept_copy  
SET loc = (SELECT loc FROM dept_copy  
           WHERE dno = 20 )  
WHERE dno = 10 ;
```

[예제] 여러 개의 칼럼 값을 동시에 변경해봅시다.

예 제

```
UPDATE dept_copy  
SET dname = (SELECT dname FROM dept_copy  
             WHERE dno = 30 ),  
    loc = (SELECT loc FROM dept_copy  
           WHERE dno = 30 )  
WHERE dno = 10 ;
```

### 3. 테이블의 내용을 삭제하는 DELETE 문

- ❖ DELETE 문을 사용하여 기존에 저장된 데이터를 삭제할 수 있습니다.
- ❖ WHERE 절을 생략하면 테이블에 있는 모든 행이 제거됩니다.

형 식

```
DELETE [FROM] table_name  
WHERE conditions ;
```

[예제] WHERE 절로 특정 로우만 제거하기

예 제

```
DELETE dept_copy  
WHERE dno = 10 ;
```

### 3-1. 다른 테이블을 기반으로 로우 삭제

- ❖ DELETE 문에 서브 쿼리를 사용하여 테이블에서 다른 테이블의 값을 기반으로 로우를 삭제할 수 있습니다.

[예제] 사원들 중 SALES 부서에 근무하는 사원을 모두 삭제하여 봅시다.

예 제

```
DELETE emp_copy  
WHERE dno = (SELECT dno FROM department  
             WHERE dname = 'SALES');
```

## 4. 트랜잭션 관리

- ❖ 트랜잭션(Transaction)은 데이터 처리에서 논리적으로 하나의 작업 단위를 의미합니다.
- ❖ oracle 명령어 중 DDL과 DCL은 하나의 명령어가 하나의 트랜잭션이며, DML은 데이터의 변동의 일관성을 위해 하나 이상의 DML문이 모여 하나의 트랜잭션을 구성합니다.
- ❖ All – OR – Nothing : 여러 개의 명령어 집합이 정상적으로 처리되면 정상 종료하고, 명령어들 중 하나라도 잘못된다면 전체를 취소해 버립니다.

## 4. 트랜잭션 관리

- ❖ DML문은 실행됨과 동시에 트랜잭션이 진행됩니다. DML 작업이 성공적으로 처리되었다면 COMMIT을, 취소해야 한다면 ROLLBACK 명령을 실행합니다.
- ❖ COMMIT은 트랜잭션의 처리 과정을 반영하여 변경된 내용을 모두 영구 저장합니다.
- ❖ ROLLBACK은 트랜잭션의 처리 과정 중 발생한 변경 사항을 취소하는 명령어입니다. ROLLBACK을 실행하게 되면 하나의 묶음 처리가 시작되기 이전 상태로 돌아갑니다.