

11.1 DML (Data Manipulation Language)

■ DML 용도 및 종류

문장	설명
SELECT	데이터베이스로부터 데이터를 검색
INSERT UPDATE DELETE MERGE	데이터베이스 내의 테이블에 새로운 행을 입력하거나, 기존의 행을 수정 또는 삭제하는 명령어로 데이터 조작용어(DML : Data Manipulation Language)라고 함
CREATE ALTER DROP RENAME TRUNCATE	테이블을 생성, 변경, 삭제하는 명령어로 데이터 정의어(DDL : Data Definition Language)라고 함
COMMIT ROLLBACK SAVEPOINT	DML 문장에 의한 변경 사항을 관리하거나, 변경사항을 하나의 논리적 트랜잭션으로 포함시키는 명령어
GRANT REVOKE	데이터베이스와 데이터베이스를 구성하는 구조(테이블, 뷰 등)에 접근 권한을 부여하거나 회수하는 명령어로 데이터 제어어(DCL : Data Control Language)라고 함

11.2 DML (Data Manipulation Language)

■ INSERT 용도

- : 테이블에 새로운 행을 삽입하는 기능이다.
- : 한번에 하나의 행만을 입력할 수 있다.
- : 입력값이 문자/날짜 데이터인 경우에는 " 사용한다.

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

```
SQL> DESC DEPT;
```

이름	널?	유형
DEPTNO	NOT NULL	NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

```
SQL> INSERT INTO DEPT(DEPTNO,DNAME,LOC)
2 VALUES( 90,'인사과','서울');
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO DEPT( LOC, DNAME, DEPTNO)
2 VALUES( '서울','인사과',70 );
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO DEPT
2 VALUES( 80 , '인사과','서울');
```

1 개의 행이 만들어졌습니다.

11.3 DML (Data Manipulation Language)

■ NULL 값 입력

1. 입력될 컬럼을 INSERT 문장에 기술하지 않는다. (묵시적 방법)

```
SQL> INSERT INTO DEPT( DEPTNO, DNAME )  
2 VALUES( 91, '인사과' );
```

1 개의 행이 만들어졌습니다.

2. 명시적으로 NULL 키워드 기술하는 방법
(명시적 방법)

```
SQL> INSERT INTO DEPT  
2 VALUES( 92, '인사과', NULL );
```

1 개의 행이 만들어졌습니다.

```
SQL> SELECT *  
2 FROM DEPT;
```

DEPTNO	DNAME	LOC
90	인사과	서울
80	인사과	서울
70	인사과	서울
91	인사과	
92	인사과	
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

11.4 DML (Data Manipulation Language)

■ INSERT 의 특수한 형태

1. 특수 값 입력 (SYSDATE , USER)

```
SQL> INSERT INTO EMP(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)  
2 VALUES(9000, USER, '연구원', 7839, SYSDATE, 5000, NULL, 90);
```

1 개의 행이 만들어졌습니다.

2. 특정 데이터 타입으로 입력 (날짜타입)

```
SQL> INSERT INTO EMP(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)  
2 VALUES (9001, '홍길동', 'MANAGER', 7839, '2000/01/01', 2000, NULL, 30 );
```

1 개의 행이 만들어졌습니다.

```
SQL> INSERT INTO EMP(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)  
2 VALUES (9001, '임걱정', 'MANAGER', 7839, TO_DATE('1999-12-13', 'YYYY/MM/DD'),  
, 2000, NULL, 30 );
```

1 개의 행이 만들어졌습니다.

11.5 DML (Data Manipulation Language)

- INSERT 의 특수한 형태

3. 다른 테이블로부터의 데이터 입력

```
SQL> CREATE TABLE DEPT_1  
2  ( DEPTNO NUMBER(2),  
3    DNAME VARCHAR2(10),  
4    LOC VARCHAR2(10)  
5  );
```

테이블이 생성되었습니다.

```
SQL> INSERT INTO DEPT_1( DEPTNO , DNAME, LOC )  
2  SELECT DEPTNO, DNAME, LOC  
3  FROM DEPT;
```

4 개의 행이 만들어졌습니다.

11.6 DML (Data Manipulation Language)

■ UPDATE 용도

- : 테이블에 저장된 행들을 변경하는 문장이다.
- : 한 번에 여러 개의 행들을 변경할 수 있다.

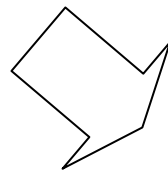
```
UPDATE table  
SET column = value [, column = value, ... ]  
[WHERE condition];
```

```
SQL> SELECT *  
2 FROM DEPT;
```

DEPTNO	DNAME	LOC
90	인사과	서울
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

```
SQL> UPDATE DEPT  
2 SET DNAME = '경리과', LOC = '부산'  
3 WHERE DEPTNO = 90;
```

1 행이 갱신되었습니다.



DEPTNO	DNAME	LOC
90	경리과	부산
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

11.7 DML (Data Manipulation Language)

■ UPDATE 의 특수한 형태

1. 서브쿼리를 이용한 복수 컬럼 변경

```
SQL> UPDATE EMP
  2 SET JOB = ( SELECT JOB
  3               FROM EMP
  4               WHERE EMPNO = 7900 ),
  5     SAL = ( SELECT SAL
  6               FROM EMP
  7               WHERE EMPNO = 7844 )
  8 WHERE EMPNO = 9001;
```

1 행이 갱신되었습니다.

2. 다른 테이블 기반으로 데이터 수정

```
SQL> UPDATE DEPT_1
  2 SET DNAME = ( SELECT DNAME
  3                 FROM DEPT
  4                 WHERE DEPTNO = 30 )
  5 WHERE DEPTNO = ( SELECT DEPTNO
  6                   FROM DEPT
  7                   WHERE DNAME = 'SALES' );
```

1 행이 갱신되었습니다.

11.8 DML (Data Manipulation Language)

■ DELETE 용도

- : 테이블에 저장된 행들을 삭제한다.
- : 한 번에 여러 개의 행들을 삭제할 수 있다.

```
DELETE [FROM] table
[WHERE condition];
```

```
SQL> DELETE FROM DEPT
      2 WHERE DEPTNO = 91;
```

1 행이 삭제되었습니다.

■ DELETE 의 특수한 형태

- 서브쿼리 및 다른 테이블 기반을 이용한 데이터 삭제

```
SQL> DELETE FROM EMP
      2 WHERE DEPTNO = ( SELECT DEPTNO
      3                   FROM DEPT
      4                   WHERE DNAME = '경리과' );
```

1 행이 삭제되었습니다.

11.9 Transaction (트랜잭션)

■ Transaction 정의 및 특징

- : SQL 문장을 이용하여 수행해야 할 작업 단위.
- : 작업단위는 하나의 SQL 문 또는 여러 SQL 문장이 될 수 있다.
- : Transaction 는 반드시 시작과 종료로 구성된다.
- : Transaction의 수행 결과는 반드시 성공 또는 실패(실행 취소함)이다.

■ Transaction 타입

타입	설명
DML(Data Manipulation Language)	데이터베이스의 논리적 작업 단위로서 여러 개의 DML 문장으로 구성
DDL(Data Definition Language)	하나의 DDL 문장으로 구성
DCL(Data Control Language)	하나의 DCL 문장으로 구성

■ Transaction 제어 명령어

명령	설명
COMMIT	현재 진행 중인 트랜잭션을 종료하며 모든 변경사항을 데이터베이스에 영구히 반영
SAVEPOINT <i>name</i>	현재 진행 중인 트랜잭션의 중간에 저장점 <i>name</i> 을 표시
ROLLBACK	현재 진행 중인 트랜잭션을 종료하며 모든 변경사항을 취소
ROLLBACK TO <i>name</i>	현재 진행 중인 트랜잭션의 저장점으로 복귀하고 저장점 이후의 모든 데이터 변경사항을 취소

11.10 Transaction (트랜잭션)

■ Transaction 시작 및 종료

1. Transaction 시작

: DML 명령을 시작한 경우에 Transaction 는 자동으로 시작된다.

2. Transaction 종료

: 사용자가 COMMIT 또는 ROLLBACK 명령을 명시적으로 실행한 경우.

: DDL문장을 실행한 경우

: DCL 문장을 실행한 경우

: 사용자가 SQL*PLUS 또는 iSQL*PLUS 를 종료한 경우

: 하드웨어 고장 또는 시스템 오류시.

11.11 Transaction (트랜잭션)

■ 명시적인 COMMIT / ROLLBACK 장점

- : 데이터의 일관성을 보장해 준다.
- : 데이터의 변경사항을 데이터베이스에 영구히 반영하기 전에 데이터 변경사항을 미리 볼 수 있다.
- : 논리적으로 연관된 작업을 그룹화 할 수 있다.

■ Transaction 종료 전의 데이터 상태

- : 트랜잭션내의 모든 데이터 변경 사항은 트랜잭션이 종료되기 전까지는 모두 임시적이다. 따라서 데이터의 변경 전 데이터로 복구 될 수 있다.
- : 변경된 행은 Lock(잠금)이 걸리며, 해당 사용자를 제외한 나머지 사용자는 해당 행들을 변경할 수 없다.
- : 현재 사용자는 SELECT 문장을 이용하여 데이터 변경 후의 결과를 확인 할 수 있다.
- : 다른 사용자는 현재 사용자에게 의해 변경된 데이터 결과를 확인할 수 없다.
(데이터 일관성)

11.12 Transaction (트랜잭션)

■ COMMIT

- : 모든 데이터 변경사항을 데이터베이스에 영구히 반영시키는 명령어.
- : 변경전의 데이터는 모든 잃게 된다.
- : 모든 사용자들이 트랜잭션 종료 후의 결과를 확인 할 수 있다.
- : 트랜잭션이 진행 중이었던 행들에 대한 잠금이 모두 해소되며, 다른 사용자에 의해서 변경이 가능해진다.

```
SQL> SELECT * FROM DEPT;
```

DEPTNO	DNAME	LOC
90	경리과	부산
92	인사과	
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

6 개의 행이 선택되었습니다.

```
SQL> DELETE FROM DEPT  
2 WHERE DEPTNO = 92;
```

1 행이 삭제되었습니다.

```
SQL> COMMIT;
```

커밋이 완료되었습니다.

11.13 Transaction (트랜잭션)

■ ROLLBACK

- : 모든 데이터 변경사항을 취소하는 명령어.
- : 변경전의 데이터가 복원된다.
- : 모든 사용자들이 트랜잭션 종료 후의 결과를 확인 할 수 있다.
- : 트랜잭션이 진행 중이었던 행들에 대한 잠금이 모두 해소되며, 다른 사용자에 의해서 변경이 가능해진다.

```
SQL> DELETE FROM EMP;
```

14 행이 삭제되었습니다.

```
SQL> ROLLBACK;
```

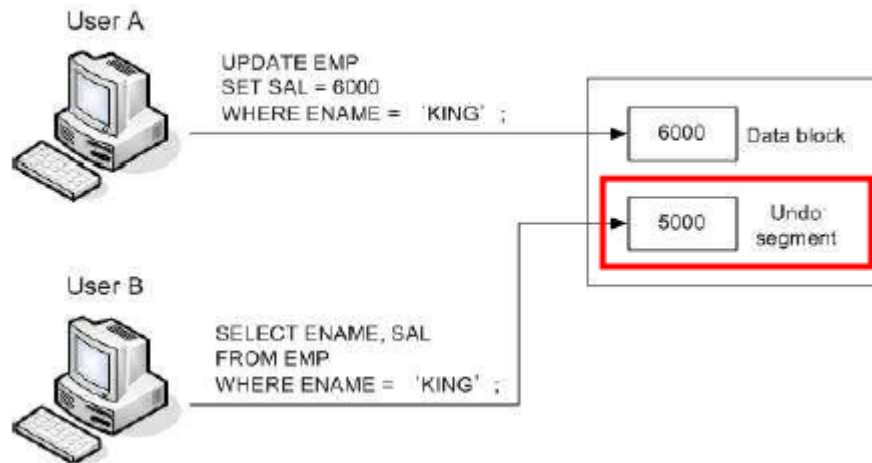
롤백이 완료되었습니다.

11.14 Transaction (트랜잭션)

■ 읽기 일관성 (Read Consistency)

- : 사용자들에게 가장 최근에 커밋된 데이터를 보여주는 것.
- : 데이터를 검색하는 사용자와 변경하는 사용자들 사이에 일관적인 관점을 제공한다.
즉, 다른 사용자들이 변경중인 데이터를 볼 수 없게 한다.
- : 일관적인 데이터베이스 변경방법을 제공함으로써 동일한 데이터를 동시에 변경할 때 발생할 수 있는 혼란을 방지한다.

■ 읽기 일관성 구현 원리



11.15 Transaction (트랜잭션)

LOCK 경합

① SQL> SELECT * FROM DEPT;

DEPTNO	DNAME	LOC
90	경리과	부산
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

② SQL> UPDATE DEPT
2 SET DNAME = '인사과'
3 WHERE DEPTNO = 90;

1 행이 갱신되었습니다.

③ SQL> SELECT * FROM DEPT;

DEPTNO	DNAME	LOC
90	인사과	부산
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

⑥ SQL> ROLLBACK;

롤백이 완료되었습니다.

④ SQL> SELECT * FROM DEPT;

DEPTNO	DNAME	LOC
90	경리과	부산
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

⑤ SQL> UPDATE DEPT
2 SET LOC = '서울'
3 WHERE DEPTNO = 90; **Wait!!**

⑦ 1 행이 갱신되었습니다.