

오라클을 이용한 DB 구축 초급

강의 노트

데이터 조작

Lesson 1. 행 삽입

행 삽입

1. 데이터 조작어(DML)와 트랜잭션
2. 행 삽입
3. 날짜 값을 갖는 행의 삽입
4. 대체 변수를 이용한 행 삽입
5. 다른 테이블의 행을 복사한 행 삽입

1. 데이터 조작어(DML)와 트랜잭션

(1) 데이터 조작어(DML : Data Manipulation Language)

→ 이미 정의된 데이터를 검색, 삽입, 삭제, 변경하기 위해 사용되는 언어로, 다음과 같은 네 가지 명령어가 있다.

명령어		기능
SELECT		테이블에 있는 기존 행을 변경
INSERT		테이블에서 행을 검색
UPDATE		테이블에 있는 기존 행을 삭제
DELETE		테이블에 새로운 행을 삽입

(2) 트랜잭션(Transaction)이란?

- 작업의 논리적인 단위로, 하나 이상의 DML 문장으로 구성됨
- 실제로 DB에서 한번에 처리되는 일의 단위
즉, 실제로 DB에 변화를 유발하는 일의 단위
- 실행 시 트랜잭션을 구성하는 한 문장이라도 실패하면, 다른 모든 문장의 실행이 취소됨

2. 행 삽입

→ INSERT 명령으로 기존 테이블에 새로운 행을 추가한다.

(1) INSERT 명령의 형식

→ 삽입할 테이블 이름과 열 이름들, 그리고 각 열의 값을 기술한다.

```
INSERT INTO 테이블이름 [(열 이름1 [, 열 이름2 ...])]
VALUES (값1 [, 값2 ...]);
```

(2) INSERT 명령의 특징

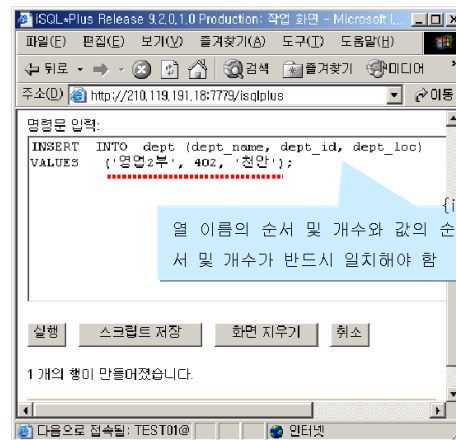
VALUES 절에 열의 값을 명시하므로, 단 한 행만 삽입할 수 있다.

열 이름을 명시하지 않으면, 테이블에 정의된 열의 순서대로 값을 나열해야 한다.

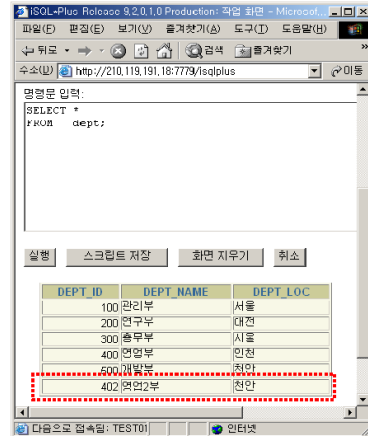
열 이름을 명시하면, 열 이름의 순서 및 개수와 대응하는 값의 순서 및 개수가 일치해야 한다.

(3) 행 삽입 예제 : 열 이름을 명시한 경우

→ dept 테이블에 부서이름이 '영업2부', 부서번호가 402, 부서위치가 '천안'인 행을 삽입하되, 열을 부서이름, 번호, 위치 순으로 삽입하고, 이 행이 제대로 삽입되었는지 확인한다.

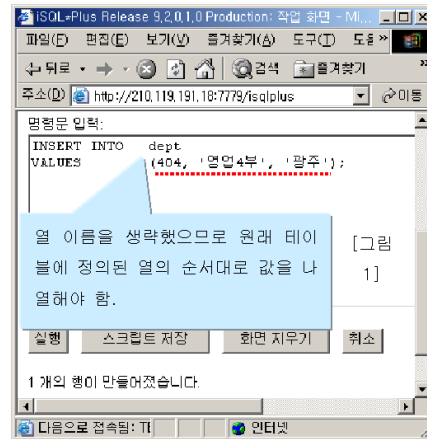


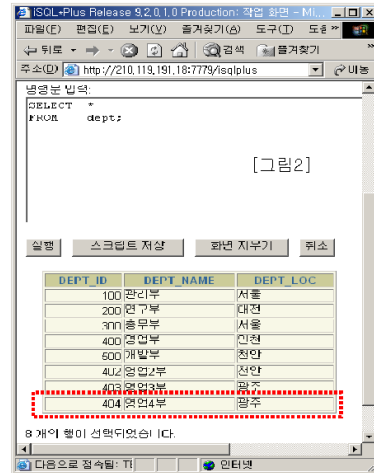
[그림1]



(4) 행 삽입 예제 : 열 이름을 생략한 경우

→ dept 테이블에 부서번호가 404, 부서이름이 '영업4부', 부서위치가 '광주'인 행을 삽입하되, 열 이름을 나열하지 않고 삽입한 다음, 이 행이 제대로 삽입되었는지 확인한다.



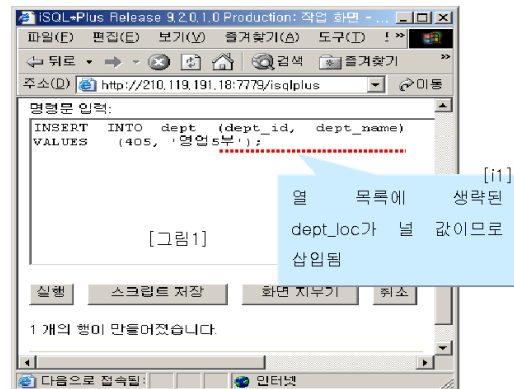


(5) 행 삽입 예제 : 널 값을 갖는 행을 삽입하는 경우의 예

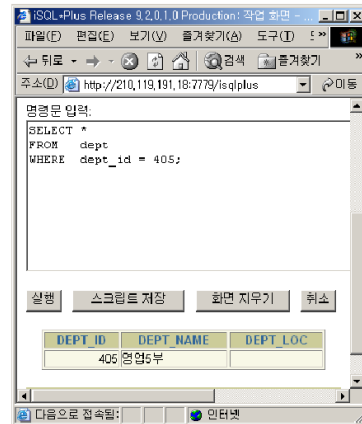
→ 기본 키인 열을 제외하고, 현재 값을 알 수 없는 열의 경우, 널 값을 삽입할 수 있다.

① 암시적인 방법 : 열 목록에서 널 값을 갖는 열 이름을 생략한다.

→ dept 테이블에 부서번호가 405이고, 부서이름이 '영연5부'인 행을 삽입한 다음,



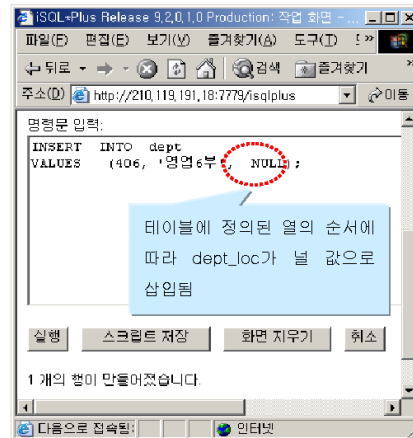
[그림2]



새로 삽입한 행의 부서위치(dept_loc)가 어떤 값을 갖고 있는지 확인한다.

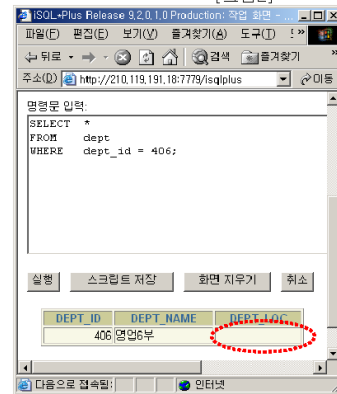
② 명시적인 방법 : VALUES 절에 해당 열의 값을 'NULL'로 명시한다. (문자나 날짜 스트링인 경우, NULL 대신 공백 스트링(' ')을 명시해도 된다.)

→ dept 테이블에 부서번호가 406이고, 부서이름이 '영역6부', 부서위치의 값은 널인 행을 삽입한 다음, 새로 삽입한 행의 각 열의 값을 확인한다.



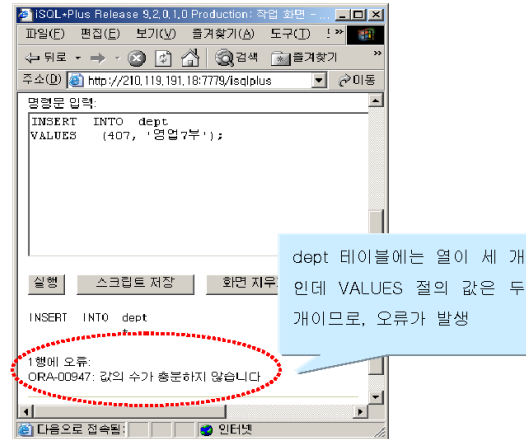
[그림1]

[그림2]



☞ 참고 : VALUES 절에 널 값 지정을 생략한 경우

앞의 INSERT 문에서 VALUES 절에 NULL을 생략하면, dept 테이블의 열 수는 세 개인 반면 VALUES 절의 열 값이 두 개이기 때문에 오류가 발생한다.

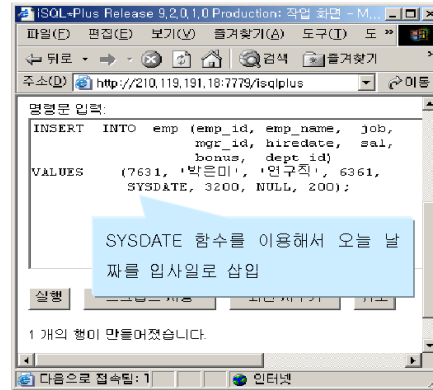


3. 날짜 값을 갖는 행의 삽입

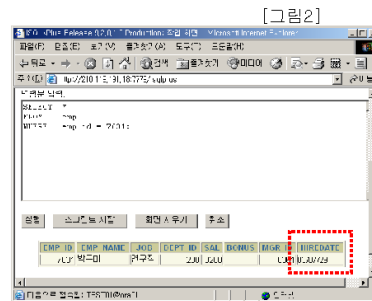
- 오라클에서는 반드시 TO_DATE 함수를 사용해서 날짜 값을 삽입해야 한다.
- SYSDATE 함수를 사용해서 오늘 날짜를 삽입할 수도 있다.

[예 1 : SYSDATE 함수를 사용한 날짜 값 삽입의 예]

→ emp 테이블에 사원번호가 7631, 이름이 '박은미', 업무가 '연구직', 관리자번호가 6361, 입사일은 오늘 날짜, 급여는 3200, 보너스는 없고, 부서번호가 200번 사원을 삽입한 다음, 새로 삽입한 행의 각 열의 값을 확인한다. 특히 입사일로 오늘 날짜가 제대로 삽입되었는지 확인한다.



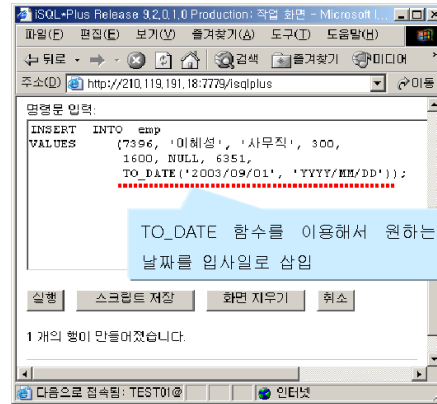
[그림1]



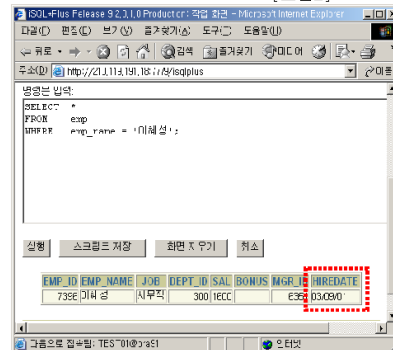
[그림2]

[예 2 : TO_DATE 함수를 사용한 날짜 값 삽입의 예]

→ emp 테이블에 사원번호가 7396, 이름인 '이혜성', 업무가 '사무직', 부서번호가 300, 급여는 1600, 보너스는 없고, 관리자번호가 6361, 입사일은 2003/09/01인 사원을 삽입한 다음, 새로 삽입한 행의 각 열의 값을 확인한다. 특히, 입사일이 제대로 삽입되었는지 확인한다.



[그림2]



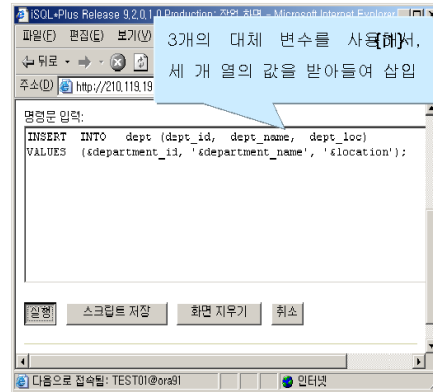
4. 대체 변수를 이용한 행 삽입

- 대체 변수를 사용하여 열의 값을 사용자로부터 받아들이어서 행을 삽입할 수 있다.
- 대체 변수를 사용한 하나의 INSERT 문을 여러 번 실행시켜서 서로 다른 행을 삽입할 수 있다.

[예 1 : 3개의 대체변수를 이용해서 행 삽입]

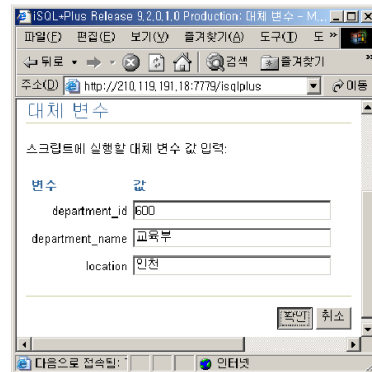
- dept 테이블에 여러 행을 삽입하기 위해서 INSERT 문에 각 열의 값을 고정시키는 대신, 대체 변수를 사용해 세 개 열의 값을 사용자로부터 입력 받아서 삽입하도록

INSERT 문을 작성한 다음, 부서번호는 600, 이름은 '교육부', 위치는 '인천'인 삽입하고, 삽입된 행을 확인한다.

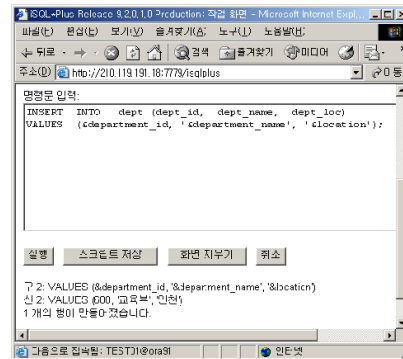


[그림1]

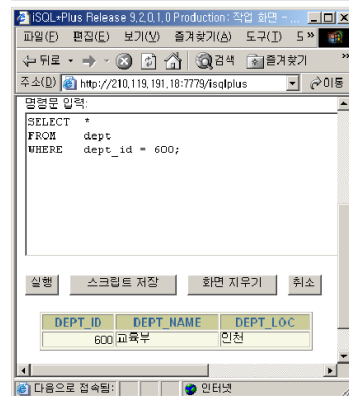
[그림2]



[그림3]

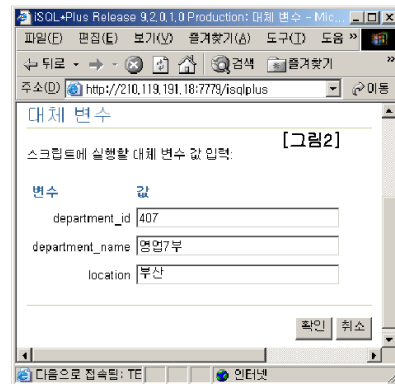
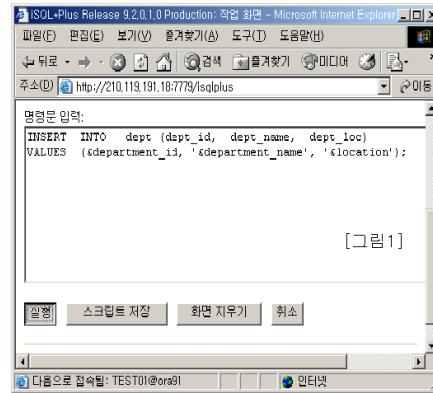


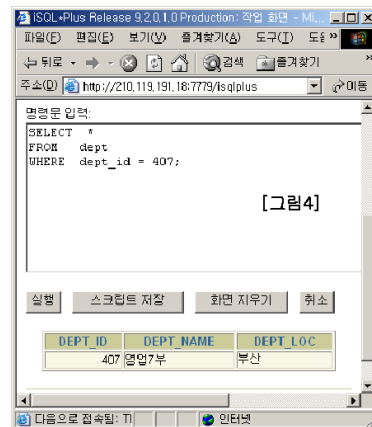
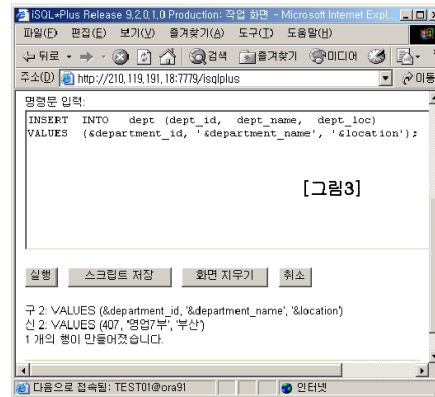
[그림4]



[예 2 : 3개의 대체변수를 이용해서 랭 삽입]

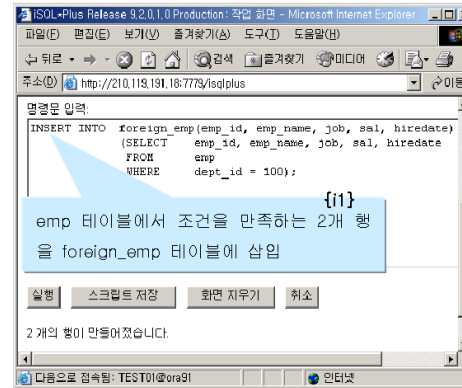
→ 앞의 INSERT 문을 다시 실행해서, 부서번호가 470이고, 이름은 '영업7부', 위치는 '부산'인 새로운 부서를 삽입한 다음, 제대로 삽입되었는지 확인한다.





6. 다른 테이블의 행을 복사한 행 삽입

- 서브쿼리를 이용해서 INSERT 문을 작성한다.
- VALUES 절 대신 열의 값을 반환할 서브쿼리를 기술한다.
- 서브쿼리가 반환한 열 수와 INSERT 절의 열 수가 반드시 일치해야 한다.
- emp 테이블에서 100번 부서에서 근무하는 모든 사원을 복사해서 foreign_emp 테이블에 삽입한 다음, foreign_emp 테이블에 제대로 삽입되었는지 확인한다.



[그림1]

