=========================================================================

DB 응용

학번 : 201907033  
 이름 : 김혜인  
 제출일 : 20/06/05

=========================================================================

1) 칼럼레벨과 테이블 레벨로 테이블을 각각 생성하여라.  
1. 테이블 이름 : DEPT학번c(칼럼 레벨)  
SQL> CREATE TABLE DEPT201907033c  
 2 (no NUMBER(7) CONSTRAINT DEPT201907033c\_no\_pk PRIMARY KEY,  
 3 name VARCHAR(25));  
  
테이블이 생성되었습니다.

2. 테이블 이름 : DEPT학번t(테이블 레벨)  
SQL> CREATE TABLE DEPT201907033t  
 2 (no NUMBER(7),  
 3 name VARCHAR(25),  
 4 CONSTRAINT DEPT201907033t\_no\_pk PRIMARY KEY(no));  
  
테이블이 생성되었습니다.

3. 테이블 이름: EMP학번c(칼럼 레벨)  
SQL> CREATE TABLE EMP201907033c  
 2 (id NUMBER(7) CONSTRAINT EMP201907033c\_id\_pk PRIMARY KEY,  
 3 lname VARCHAR2(25) CONSTRAINT EMP201907033c\_lname\_nn NOT NULL,  
 4 fname VARCHAR2(25),  
 5 deptid NUMBER(7) CONSTRAINT EMP201907033c\_deptid\_nn NOT NULL  
 6 CONSTRAINT EMP201907033c\_deptid\_fk REFERENCES DEPT201907033c(NO));

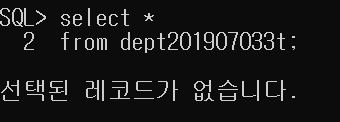
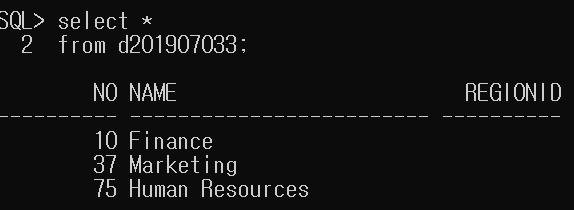
테이블이 생성되었습니다.

4. 테이블 이름 : EMP학번t(테이블 레벨)  
SQL> CREATE TABLE EMP201907033t  
 2 (id NUMBER(7),  
 3 lname VARCHAR(25) CONSTRAINT EMP201907033t\_lname\_nn NOT NULL,  
 4 fname VARCHAR(25),  
 5 deptid NUMBER(7) CONSTRAINT EMP201907033t\_deptid\_nn NOT NULL,  
 6 CONSTRAINT EMP201907033t\_id\_pk PRIMARY KEY(id),  
 7 CONSTRAINT EMP201907033t\_deptid\_fk FOREIGN KEY(deptid) REFERENCES DEPT201907033t(NO));  
  
테이블이 생성되었습니다.  
  
  
2) 위에서 생성한 테이블들의 구조를 확인하여라.  
1. 테이블 이름 : DEPT학번c(칼럼 레벨)  
SQL> desc DEPT201907033c  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME VARCHAR2(25)  
  
2. 테이블 이름 : DEPT학번t(테이블 레벨)  
SQL> desc DEPT201907033t  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME VARCHAR2(25)  
  
3. 테이블 이름: EMP학번c(칼럼 레벨)

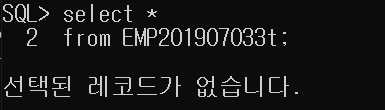
SQL> desc EMP201907033c

이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(7)  
 LNAME NOT NULL VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)

4. 테이블 이름 : EMP학번t(테이블 레벨)  
SQL> desc EMP201907033t  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(7)  
 LNAME NOT NULL VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)

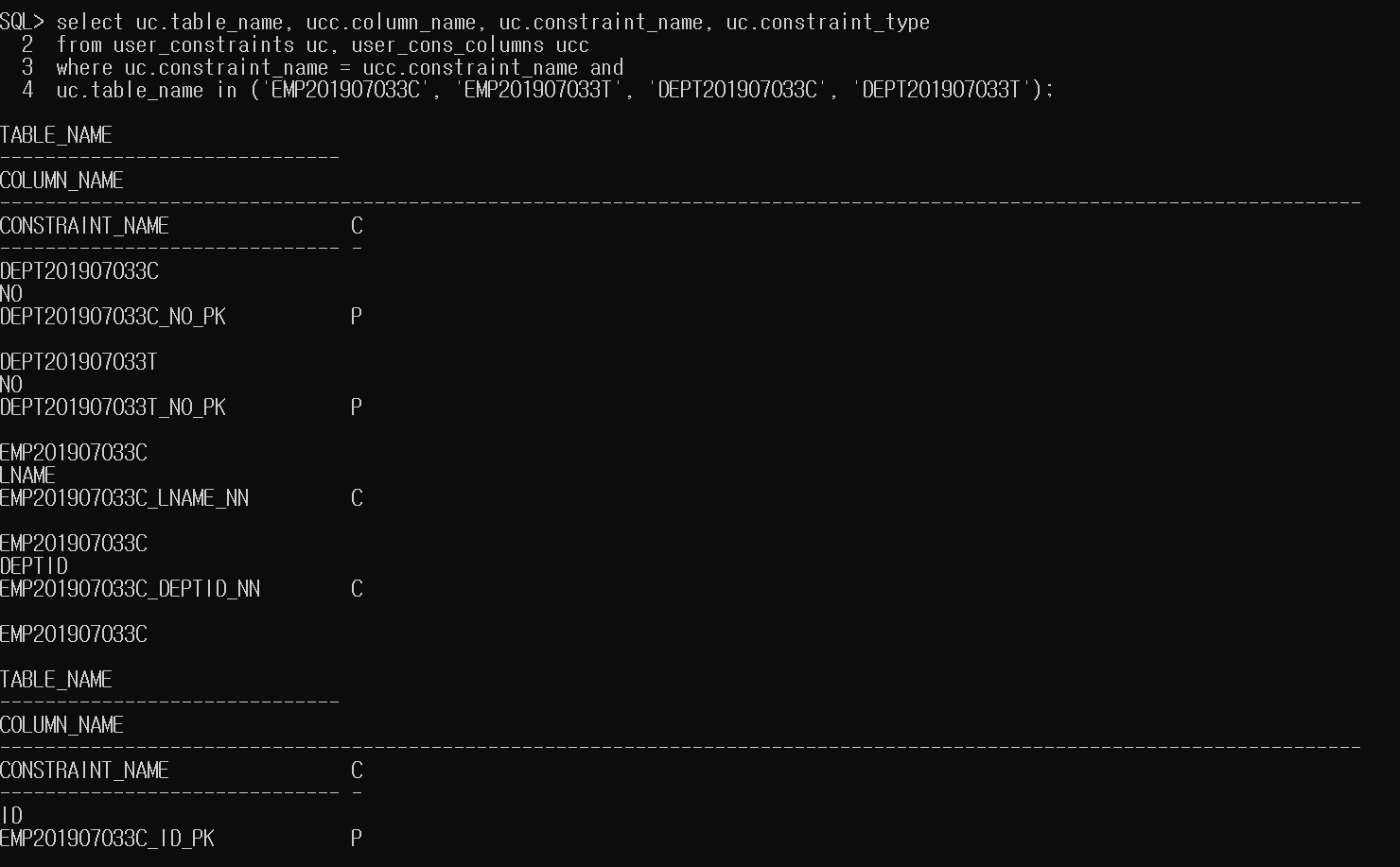
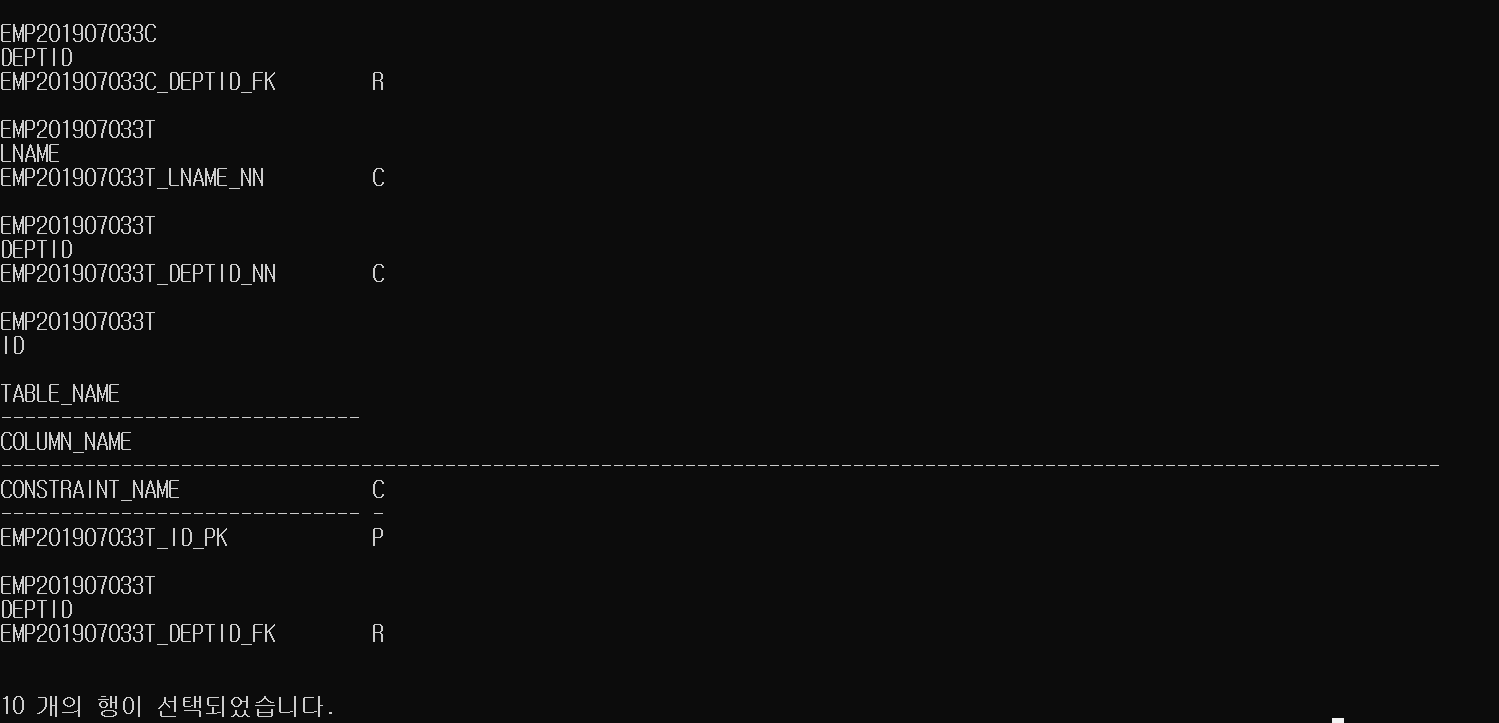
3) 위에서 생성한 테이블의 데이터를 확인하여라.  
1. 테이블 이름 : DEPT학번c(칼럼 레벨)  
SQL> select \*  
 2 from d201907033;  
  
 NO NAME REGIONID  
---------- ------------------------- ----------  
 10 Finance  
 37 Marketing  
 75 Human Resources  
  
2. 테이블 이름 : DEPT학번t(테이블 레벨)  
SQL> select \*  
 2 from DEPT201907033t;  
  
선택된 레코드가 없습니다.

3. 테이블 이름: EMP학번c(칼럼 레벨)  
SQL> select \*  
 2 from e201907033;  
  
 ID LNAME FNAME DEPTID  
---------- ------------------------- ------------------------- ----------  
 200 Donna Smith 10  
 202 Korsgaard Chin 75  
 203 Rey Guiliani 37  
  
4. 테이블 이름 : EMP학번t(테이블 레벨)  
SQL> select \*  
 2 from EMP201907033t;  
  
선택된 레코드가 없습니다.

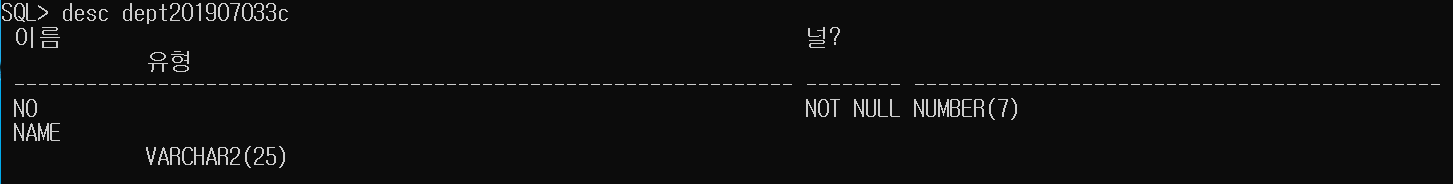


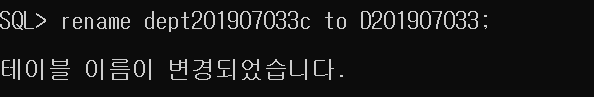
1. 데이터 사전을 이용해서 8주차에 생성한 EMP학번c, EMP학번t, DEPT학번c, DEPT학번t 테이블에   
대해 테이블명, 칼럼명, 제약조건명, 제약조건유형을 검색하여라.   
(4개의 테이블에 대해서만 한 개의 SQL문에 표시하여라.)  
테이블명(table\_name), 칼럼명(column\_name), 제약조건명(constraint\_name),   
제약조건유형(constraint\_type)

SQL> select uc.table\_name, ucc.column\_name, uc.constraint\_name, uc.constraint\_type  
 2 from user\_constraints uc, user\_cons\_columns ucc  
 3 where uc.constraint\_name = ucc.constraint\_name and  
 4 uc.table\_name in ('EMP201907033C', 'EMP201907033T', 'DEPT201907033C', 'DEPT201907033T');

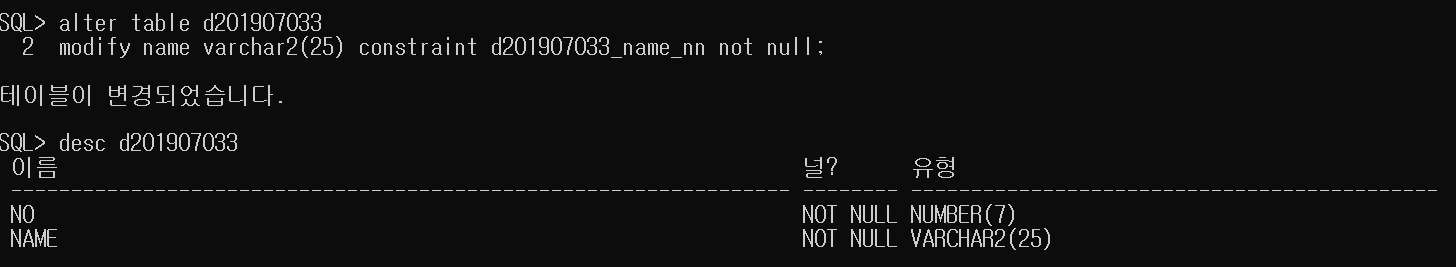


**1 . DEPT학번c 테이블을 다음과 같이 변경하여라.**

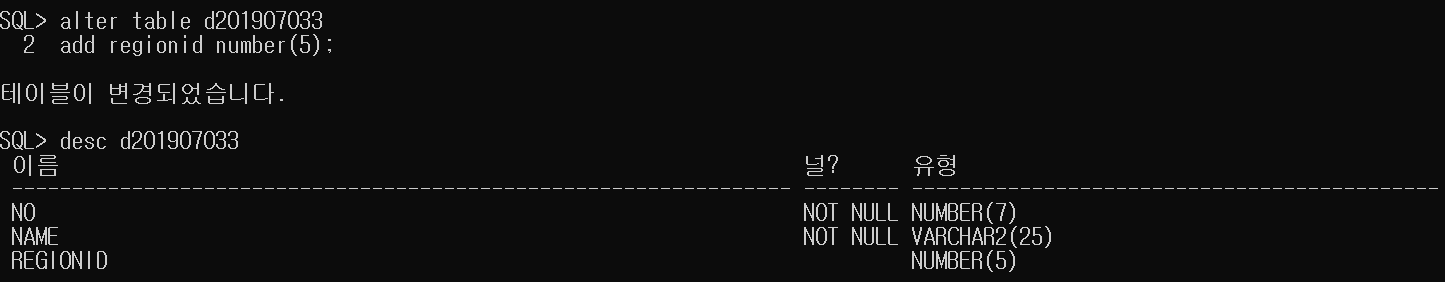
 1) DEPT학번c테이블의 구조를 확인하여라.  
SQL> desc dept201907033c  
 이름 널?  
 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME  
 VARCHAR2(25)

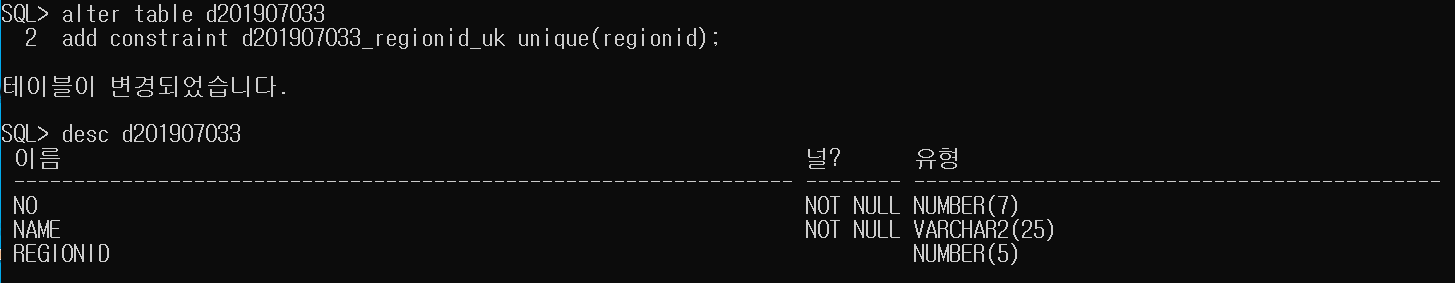
2) 테이블 이름을 D학번으로 변경하여라.  
  
  
  
  
SQL> rename dept201907033c to D201907033;  
  
테이블 이름이 변경되었습니다.

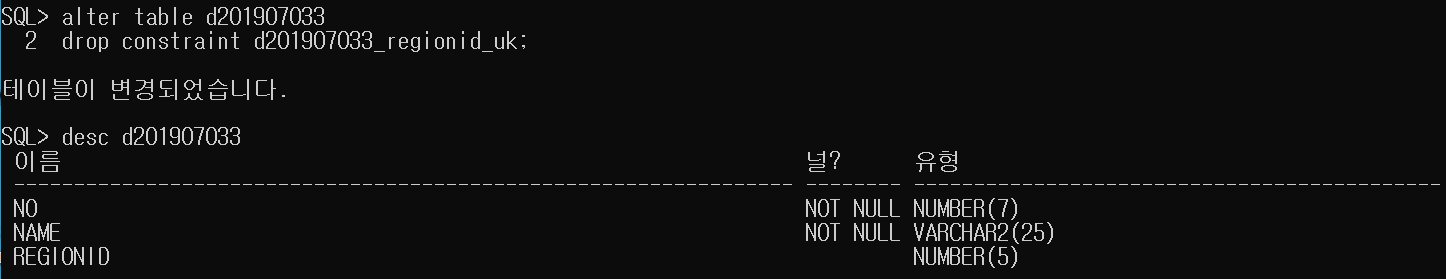
  
 3) D학번 테이블의 구조를 확인하여라.  
SQL> desc d201907033|  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME VARCHAR2(25)

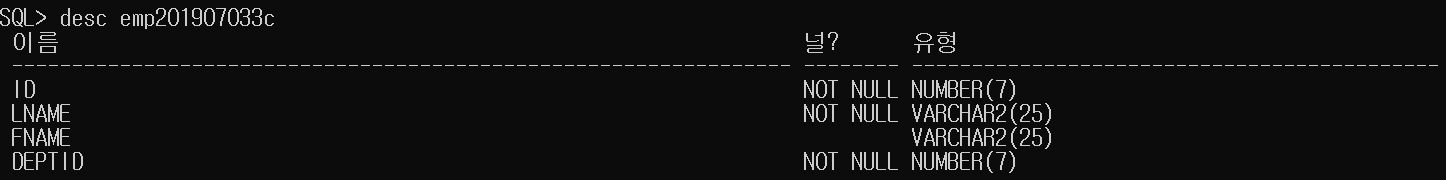
 4) D학번 테이블의 NAME 칼럼에 Not Null 제약조건을 추가하고 테이블의 구조를 확인하여라.  
SQL> alter table d201907033  
 2 modify name varchar2(25) constraint d201907033\_name\_nn not null;  
  
테이블이 변경되었습니다.

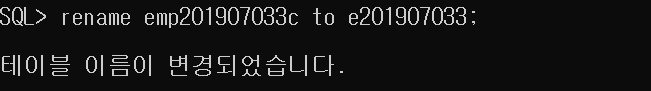
SQL> desc d201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME NOT NULL VARCHAR2(25)

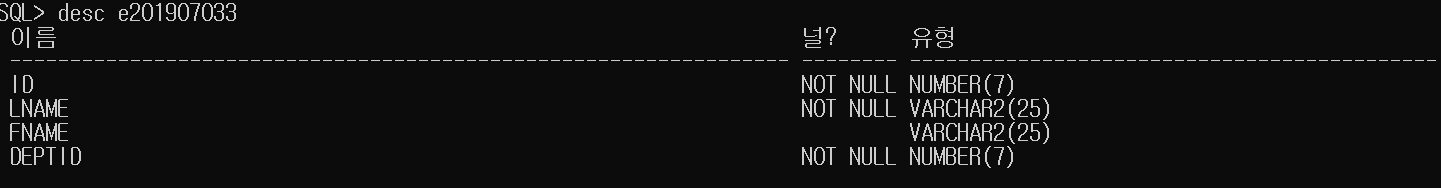
 5) D학번 테이블의 regionid 칼럼을 Num(5)로 추가하고 테이블의 구조를 확인하여라.   
SQL> alter table d201907033  
 2 add regionid number(5);  
  
테이블이 변경되었습니다.  
  
SQL> desc d201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME NOT NULL VARCHAR2(25)  
 REGIONID NUMBER(5)

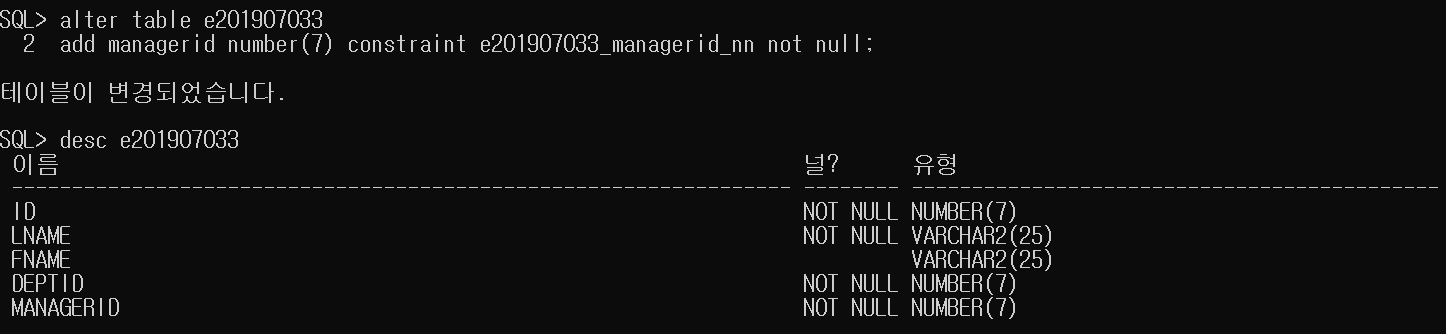
 6) D학번 테이블의 regionid 칼럼에 Unique제약조건을 추가하고 테이블의 구조를 확인하여라.   
SQL> alter table d201907033  
 2 add constraint d201907033\_regionid\_uk unique(regionid);  
  
테이블이 변경되었습니다.  
  
SQL> desc d201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)\  
 NAME NOT NULL VARCHAR2(25)  
 REGIONID NUMBER(5)

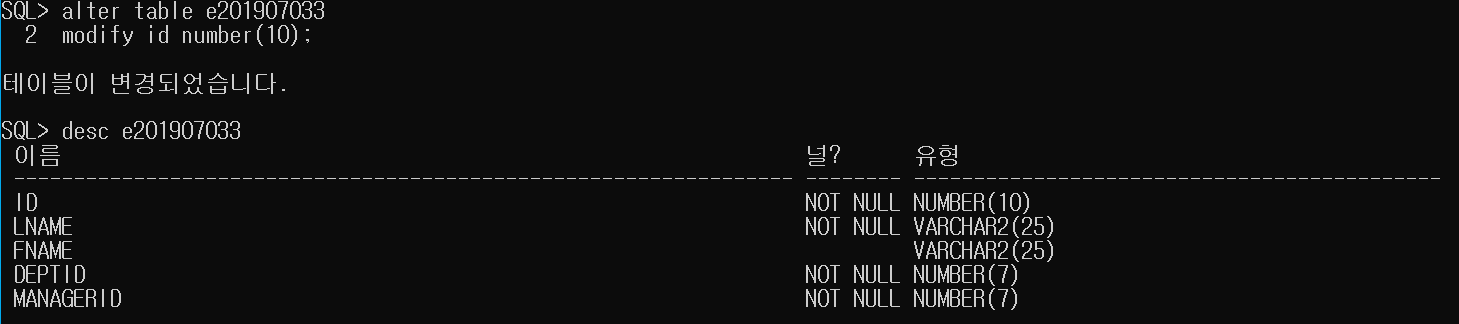
 7) D학번 테이블의 regionid 칼럼에 Unique제약조건을 삭제하고 테이블의 구조를 확인하여라.  
SQL> alter table d201907033  
 2 drop constraint d201907033\_regionid\_uk;  
  
테이블이 변경되었습니다.  
  
SQL> desc d201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME NOT NULL VARCHAR2(25)  
 REGIONID NUMBER(5)  
**2 . EMP학번c 테이블을 다음과 같이 변경하여라.**

1) EMP학번c 테이블의 구조를 확인하여라.  
SQL> desc emp201907033c  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(7)  
 LNAME NOT NULL VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)

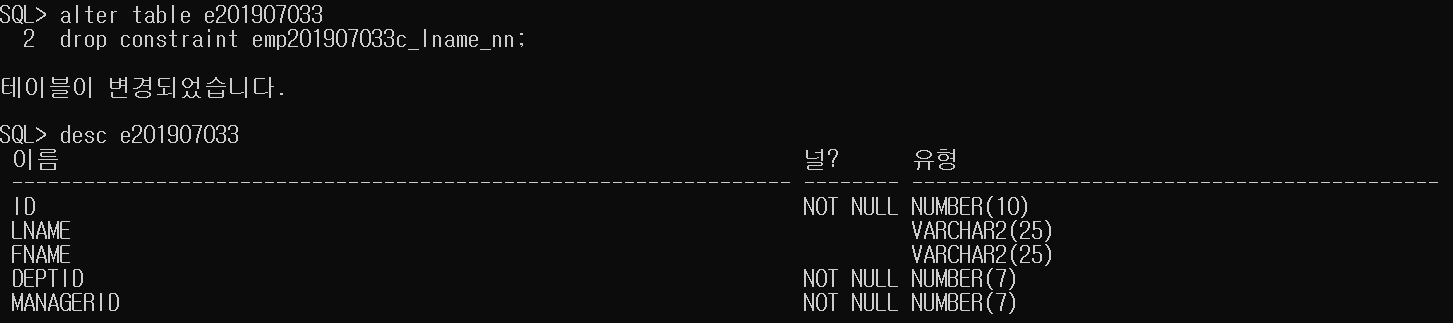
 2) 테이블 이름을 E학번으로 변경하여라.  
  
  
  
  
SQL> rename emp201907033c to e201907033;  
  
테이블 이름이 변경되었습니다.

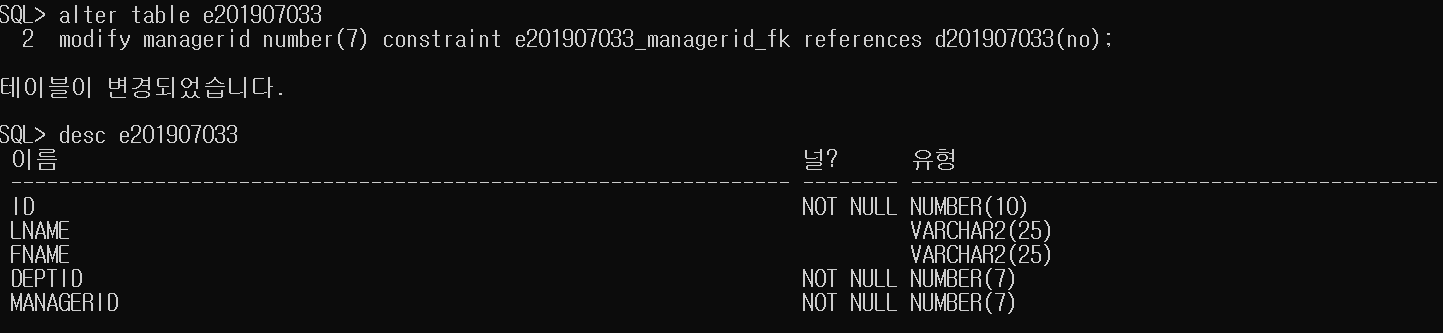
  
 3) E학번 테이블의 구조를 확인하여라.  
SQL> desc e201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(7)  
 LNAME NOT NULL VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)

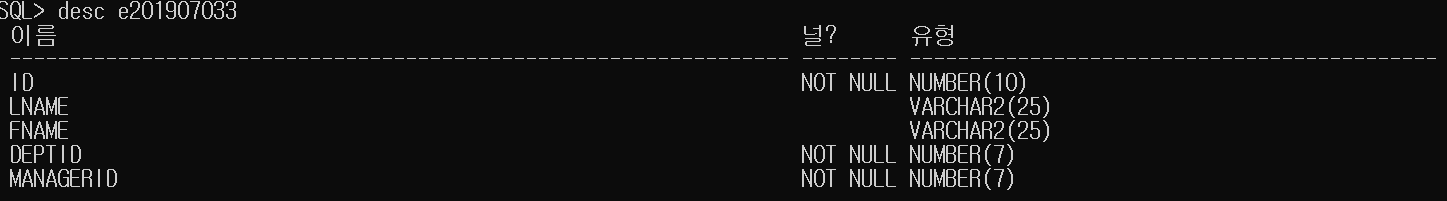
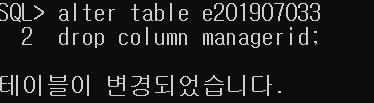
4) E학번 테이블의 managerid 칼럼을 Num(7) Not Null 제약조건으로 추가하고 테이블의 구조를 확인하여라.  
SQL> alter table e201907033  
 2 add managerid number(7) constraint e201907033\_managerid\_nn not null;  
  
테이블이 변경되었습니다.  
  
SQL> desc e201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(7)  
 LNAME NOT NULL VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 MANAGERID NOT NULL NUMBER(7)

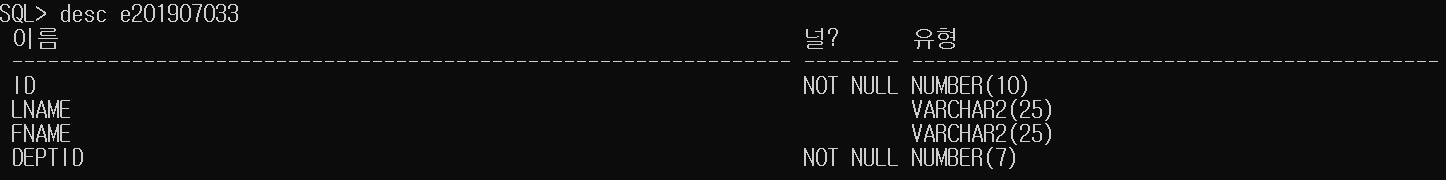
5) E학번 테이블의 ID 칼럼의 길이를 NUM(10)으로 변경하고 테이블의 구조를 확인하여라.  
SQL> alter table e201907033  
 2 modify id number(10);  
  
테이블이 변경되었습니다.

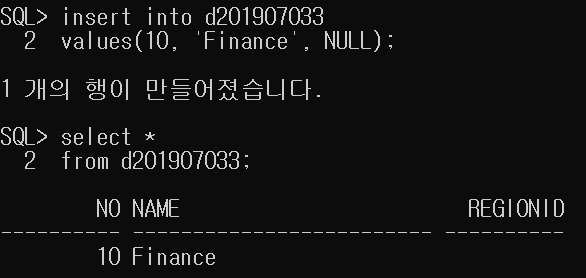
SQL> desc e201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(10)  
 LNAME NOT NULL VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 MANAGERID NOT NULL NUMBER(7)

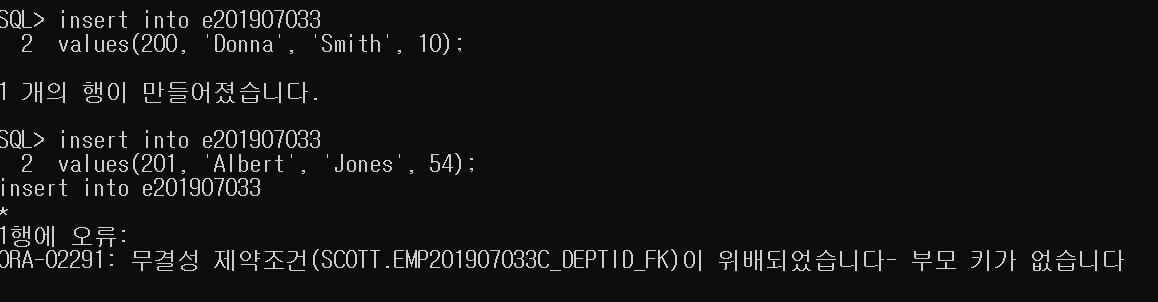
 6) E학번 테이블의 LNAME 칼럼의 Not Null 제약조건을 삭제하고 테이블의 구조를 확인하여라.  
SQL> alter table e201907033  
 2 drop constraint emp201907033c\_lname\_nn;  
  
테이블이 변경되었습니다.  
  
SQL> desc e201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(10)  
 LNAME VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 MANAGERID NOT NULL NUMBER(7)

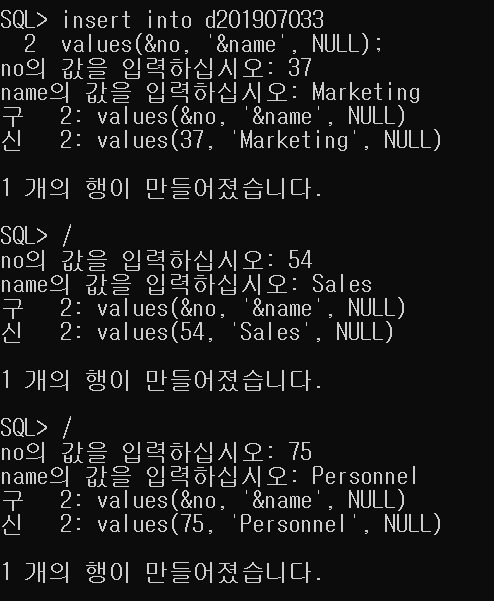
 7) E학번 테이블의 managerid 칼럼에 D학번 테이블의 no 칼럼을 참조하는 Foreign Key 제약조건을 추가하고, 테이블의 구조를 확인하여라.  
SQL> alter table e201907033  
 2 modify managerid number(7) constraint e201907033\_managerid\_fk references d201907033(no);  
  
테이블이 변경되었습니다.  
  
SQL> desc e201907033  
이름 널? 유형  
----------------------------------------------------------------- -------- --------------------------------------------  
ID NOT NULL NUMBER(10)  
LNAME VARCHAR2(25)  
FNAME VARCHAR2(25)  
DEPTID NOT NULL NUMBER(7)  
MANAGERID NOT NULL NUMBER(7)

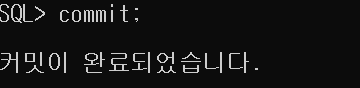
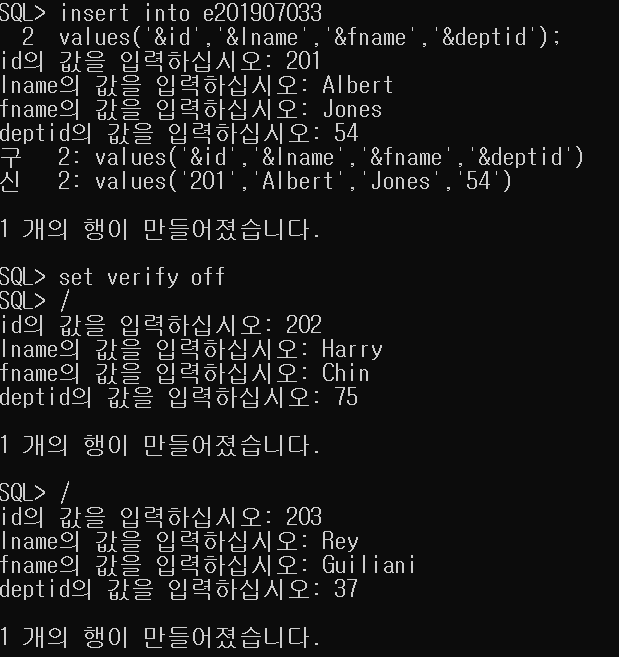
**\*\*\* 5장에서 생성한 E학번 테이블과 D학번 테이블 사용 \*\*\***   
  
**1. E학번 테이블의 구조를 확인하여라.**SQL> desc e201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(10)  
 LNAME VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 MANAGERID NOT NULL NUMBER(7) **2. E학번 테이블의 manager\_id 칼럼을 삭제하여라.**  
  
  
SQL> alter table e201907033  
 2 drop column managerid;  
  
테이블이 변경되었습니다.

**3. E학번 테이블의 구조를 확인하여라.**SQL> desc e201907033  
이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(10)  
 LNAME VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 **4. D학번 테이블의 구조를 확인하여라.**SQL> desc d201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME NOT NULL VARCHAR2(25)  
 REGIONID NUMBER(5)

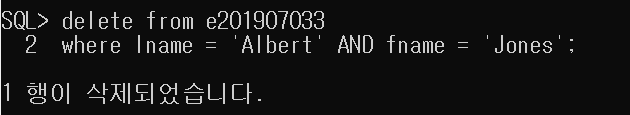
**5. E학번 테이블과 D학번 테이블에 데이터를 추가하여라.**  
1) D학번 테이블에 NO는 10 이고 NAME은 Finance인 행을 추가하고, 데이터를 확인하여라.   
  
  
  
  
  
  
  
  
  
  
  
SQL> insert into d201907033  
 2 values(10, 'Finance', NULL);  
  
1 개의 행이 만들어졌습니다.  
  
SQL> select \*  
 2 from d201907033;  
  
 NO NAME REGIONID  
---------- ------------------------- ----------  
 10 Finance

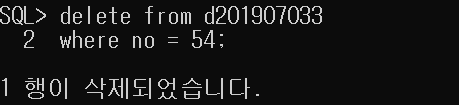
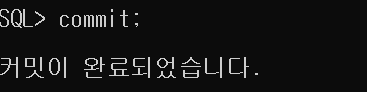
2) E학번 테이블에 두 개의 행을 추가하여라.   
첫 번째 사원은 DEPTID가 10인 Donna(LNAME) Smith(FNAME)이고 ID는 200이다.   
두 번째 사원은 DEPTID가 54인 Albert Jones이고 ID는 201 이다.   
결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
\* 부모 테이블의 칼럼 deptid에 데이터 54가 없기 때문에 foreign key 제약조건이 위배되어 에러 메시지가 발생하게 되었다고 생각합니다.  
  
  
3) NO가 10, NAME이 Marketing인 데이터를 D학번 테이블에 추가하여라.   
결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
\* 위에 오류 사항을 보면 무결성 제약 조건에서 에러가 발생하였습니다. No\_pk라는 것을 보면 primary key 제약조건을 가진다는 것을 알 수 있습니다.   
primary key는 유일한 값만 허용하므로 중복된 값을 갖지 않습니다.   
  
  
  
  
  
근데 위를 보면 이미 no 값이 10을 가지고 있기 때문에   
primary key 제약조건에 위배되어 에러가 발생하게 되었다고 생각합니다.

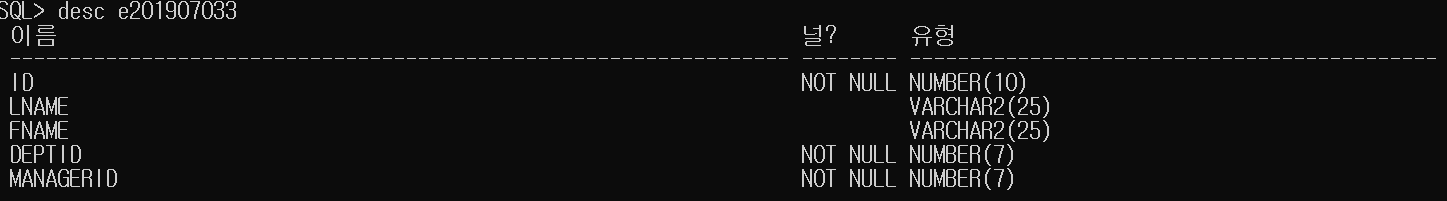
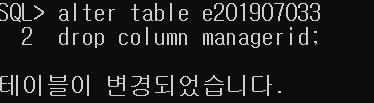
4) 치환매개변수를 이용하여 D학번 테이블에 Marketing은 NO를 37번으로,   
Sales는 54번, Personnel은 75번으로 추가하고, 데이터를 확인하여라.

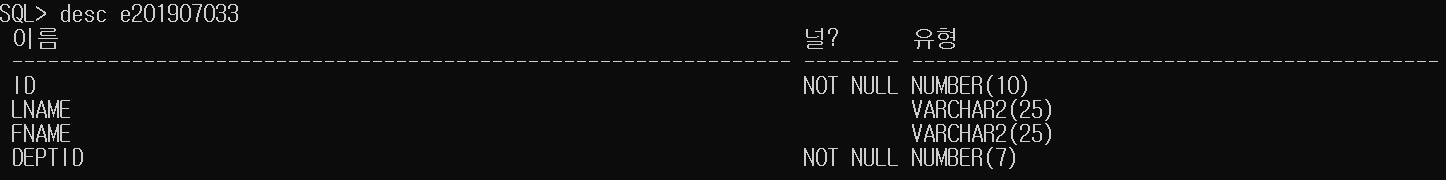
5) Albert Jones의 DEPTID는 54이고 ID는 201, Harry Chin은 각각 75와 202,   
Rey Guiliani는 37과 203인 행을 E학번 테이블에 추가하고, 데이터를 확인하여라.   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
6) 추가한 데이터가 영구적으로 반영되도록 하여라.

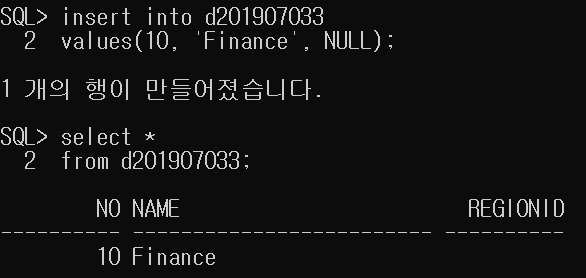
**6. E학번 테이블과 D학번 테이블의 데이터를 변경하고 삭제하여라.**  
1) D학번 테이블의 Personnel NAME을 Human Resources로 변경하고, 데이터를 확인하여라.   
  
2) E학번 테이블의 202번 LNAME을 Korsgaard로 변경하고, 데이터를 확인하여라.

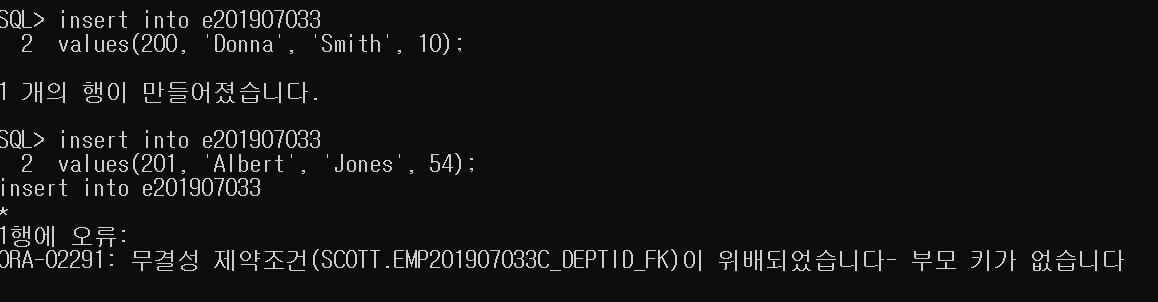
3) D학번 테이블에서 NO가 54인 데이터를 삭제하여라. 결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
SQL> delete from d201907033  
 2 where no = 54;  
delete from d201907033  
\*  
1행에 오류:  
ORA-02292: 무결성 제약조건(SCOTT.EMP201907033C\_DEPTID\_FK)이 위배되었습니다- 자식 레코드가 발견되었습니다  
\* D201907033 테이블의 칼럼 no와 e201907033 테이블의 칼럼 deptid와 foreign key 제약조건을 맺고 있는데 ,  
부모 레코드인 no 54에 참조하고 있는 자식 레코드(deptid)가 존재하므로   
제거가 불가능하며 무결성 제약조건을 위배하여 삭제할 수 없는 것이라고 생각합니다.  
  
4) E학번 테이블에서 Albert Jones에 해당하는 데이터를 삭제하고, 데이터를 확인하여라.

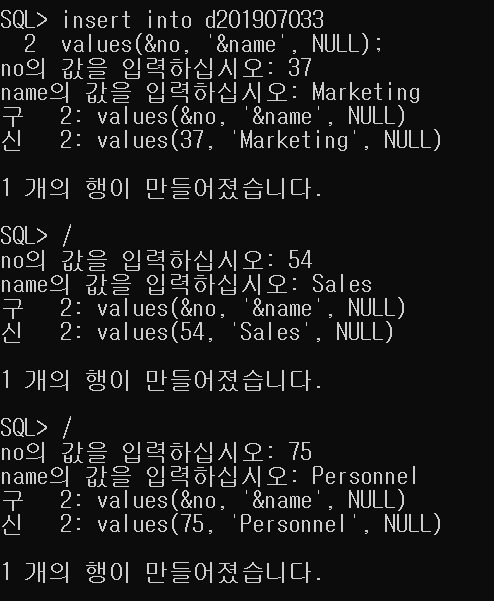
5) D학번 테이블에서 NO가 54인 데이터를 다시 삭제하여라.   
결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
\* 아까의 결과와는 다르게 삭제가 되는 것을 볼 수 있다. 그 이유는   
4)을 하기 전 e201907033의 테이블의 데이터이다. Deptid는 d201907033의 no 값 54를 참조하고 있었기 때문에 삭제가 불가능하였던 것이고   
위와 같이 4)를 실행하므로 인해 deptid와 no의 값 54 참조관계가 사라졌기 때문에 삭제가 가능해진 것이라고 생각합니다.  
  
6) 변경한 데이터가 영구적으로 반영되도록 하여라.

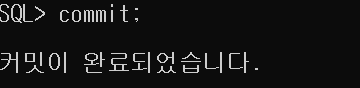
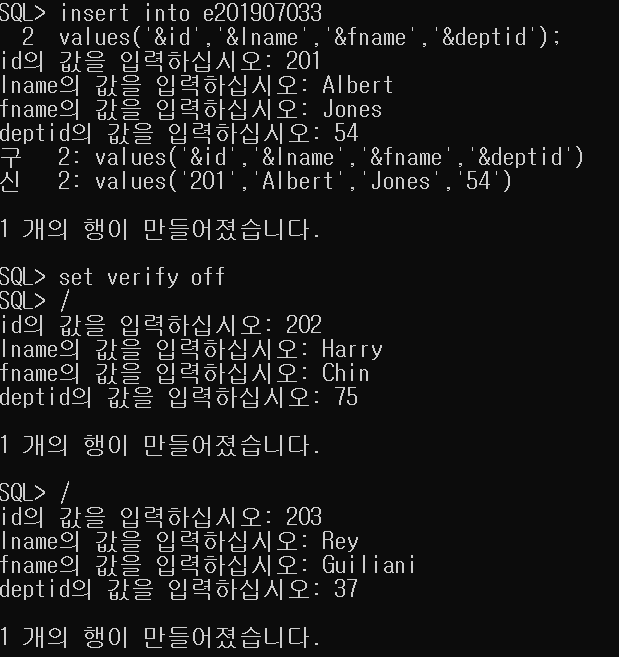
**\*\*\* 5장에서 생성한 E학번 테이블과 D학번 테이블 사용 \*\*\***   
  
**1. E학번 테이블의 구조를 확인하여라.**SQL> desc e201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(10)  
 LNAME VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 MANAGERID NOT NULL NUMBER(7) **2. E학번 테이블의 manager\_id 칼럼을 삭제하여라.**  
  
  
SQL> alter table e201907033  
 2 drop column managerid;  
  
테이블이 변경되었습니다.

**3. E학번 테이블의 구조를 확인하여라.**SQL> desc e201907033  
이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 ID NOT NULL NUMBER(10)  
 LNAME VARCHAR2(25)  
 FNAME VARCHAR2(25)  
 DEPTID NOT NULL NUMBER(7)  
 **4. D학번 테이블의 구조를 확인하여라.**SQL> desc d201907033  
 이름 널? 유형  
 ----------------------------------------------------------------- -------- --------------------------------------------  
 NO NOT NULL NUMBER(7)  
 NAME NOT NULL VARCHAR2(25)  
 REGIONID NUMBER(5)

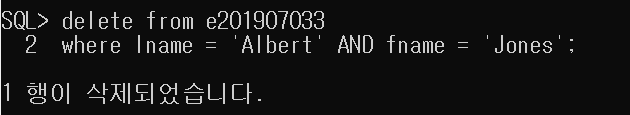
**5. E학번 테이블과 D학번 테이블에 데이터를 추가하여라.**  
1) D학번 테이블에 NO는 10 이고 NAME은 Finance인 행을 추가하고, 데이터를 확인하여라.   
  
  
  
  
  
  
  
  
  
  
  
SQL> insert into d201907033  
 2 values(10, 'Finance', NULL);  
  
1 개의 행이 만들어졌습니다.  
  
SQL> select \*  
 2 from d201907033;  
  
 NO NAME REGIONID  
---------- ------------------------- ----------  
 10 Finance

2) E학번 테이블에 두 개의 행을 추가하여라.   
첫 번째 사원은 DEPTID가 10인 Donna(LNAME) Smith(FNAME)이고 ID는 200이다.   
두 번째 사원은 DEPTID가 54인 Albert Jones이고 ID는 201 이다.   
결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
\* 부모 테이블의 칼럼 deptid에 데이터 54가 없기 때문에 foreign key 제약조건이 위배되어 에러 메시지가 발생하게 되었다고 생각합니다.  
  
  
3) NO가 10, NAME이 Marketing인 데이터를 D학번 테이블에 추가하여라.   
결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
\* 위에 오류 사항을 보면 무결성 제약 조건에서 에러가 발생하였습니다. No\_pk라는 것을 보면 primary key 제약조건을 가진다는 것을 알 수 있습니다.   
primary key는 유일한 값만 허용하므로 중복된 값을 갖지 않습니다.   
  
  
  
  
  
근데 위를 보면 이미 no 값이 10을 가지고 있기 때문에   
primary key 제약조건에 위배되어 에러가 발생하게 되었다고 생각합니다.

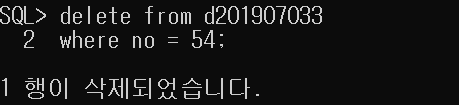
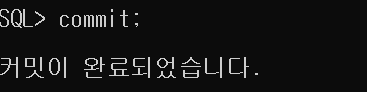
4) 치환매개변수를 이용하여 D학번 테이블에 Marketing은 NO를 37번으로,   
Sales는 54번, Personnel은 75번으로 추가하고, 데이터를 확인하여라.

5) Albert Jones의 DEPTID는 54이고 ID는 201, Harry Chin은 각각 75와 202,   
Rey Guiliani는 37과 203인 행을 E학번 테이블에 추가하고, 데이터를 확인하여라.   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
6) 추가한 데이터가 영구적으로 반영되도록 하여라.

**6. E학번 테이블과 D학번 테이블의 데이터를 변경하고 삭제하여라.**  
1) D학번 테이블의 Personnel NAME을 Human Resources로 변경하고, 데이터를 확인하여라.   
  
2) E학번 테이블의 202번 LNAME을 Korsgaard로 변경하고, 데이터를 확인하여라.   
SQL> update e201907033  
 2 set lname = ‘Korsgaard’  
 3 where id = 202  
  
1 행이 갱신되었습니다.

3) D학번 테이블에서 NO가 54인 데이터를 삭제하여라. 결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
SQL> delete from d201907033  
 2 where no = 54;  
delete from d201907033  
\*  
1행에 오류:  
ORA-02292: 무결성 제약조건(SCOTT.EMP201907033C\_DEPTID\_FK)이 위배되었습니다- 자식 레코드가 발견되었습니다  
\* D201907033 테이블의 칼럼 no와 e201907033 테이블의 칼럼 deptid와 foreign key 제약조건을 맺고 있는데 ,  
부모 레코드인 no 54에 참조하고 있는 자식 레코드(deptid)가 존재하므로   
제거가 불가능하며 무결성 제약조건을 위배하여 삭제할 수 없는 것이라고 생각합니다.  
  
4) E학번 테이블에서 Albert Jones에 해당하는 데이터를 삭제하고, 데이터를 확인하여라.   
SQL> delete from e201907033  
 2 where lname = ‘Albert’ AND fname = ‘Jones’;  
  
1 행이 삭제되었습니다.



5) D학번 테이블에서 NO가 54인 데이터를 다시 삭제하여라.   
결과와 그 결과의 이유를 설명하여라.   
(이유는 컴퓨터에 나오는 에러메시지가 아닌 본인이 이해한 것을 직접 설명)   
\* 아까의 결과와는 다르게 삭제가 되는 것을 볼 수 있다. 그 이유는   
4)을 하기 전 e201907033의 테이블의 데이터이다. Deptid는 d201907033의 no 값 54를 참조하고 있었기 때문에 삭제가 불가능하였던 것이고   
위와 같이 4)를 실행하므로 인해 deptid와 no의 값 54 참조관계가 사라졌기 때문에 삭제가 가능해진 것이라고 생각합니다.  
  
6) 변경한 데이터가 영구적으로 반영되도록 하여라.  
SQL> commit;  
커밋이 완료되었습니다.