



2023-10-29(일)

🕒 Created	@2023년 10월 29일 오전 8:26
☰ 수업유형	실습

수업활동일지

교과목명	데이터베이스응용	분반	002
수업일자	2023-10-30 (수)	교수자명	이중화 교수님
이름	이종민	학번	20193174

10/29 실습문제

- 외래키에서 on delete set null, on delete cascade 옵션 연습
1. emp테이블에서 내용을 복사하여 emp_c 테이블을 생성하시오. 단 emp_c는 emp 테이블의 empno, ename, deptno 속성으로만 이루어짐

```
create table emp_c as
select empno, ename, deptno from emp;
```

2. dept테이블을 복사하여 dept_c테이블을 생성하시오.

```
create table dept_c as
select * from dept;
```

3. dept_c테이블에서 deptno 속성을 primary key로 설정하시오.

```
alter table dept_c
```

```
add constraint dept_c_deptno_pk primary key(deptno)
```

4. emp_c테이블에서 deptno 속성이 dept_c 테이블의 deptno를 참조하는 외래키로 설정 하시오.

```
alter table emp_c  
add constraint emp_c_deptno_fk foreign key(deptno)  
references dept_c(deptno);
```

5. dept_c 테이블에서 10번 부서를 삭제하고 두 테이블의 변경 사항을 관찰하시오.

```
delete from dept_c  
where deptno = 10;
```



delete from dept_c where deptno = 10

오류 보고 -

ORA-02292: 무결성 제약조건(A20193174.EMP_C_DEPTNO_FK)이 위반되었습니다- 자식 레코드가 발견되었습니다.

- emp_c 테이블의 deptno가 dept_c의 deptno를 참조중이기에 삭제할 수 없다. (참조무결성 위반)

6. emp_c_deptno_fk 제약조건을 삭제하시오.

```
alter table emp_c  
drop constraint emp_c_deptno_fk;
```

7. emp_c테이블에서 deptno 속성이 dept_c 테이블의 deptno를 참조하는 외래키로 설정 하시오.

단, dept테이블의 deptno가 삭제될 때 emp_c테이블의 deptno는 NULL로 바꾸도록 하시오.

```
alter table emp_c
add constraint emp_c_deptno_fk foreign key(deptno)
references dept_c(deptno) on delete set null;
```

8. dept_c 테이블에서 10번 부서를 삭제하고 두 테이블의 변경 사항을 관찰하시오.

```
delete from dept_c
where deptno = 10;
```

	EMPNO	ENAME	DEPTNO
1	7369	SMITH	20
2	7499	ALLEN	30
3	7521	WARD	30
4	7566	JONES	20
5	7654	MARTIN	30
6	7698	BLAKE	30
7	7782	CLARK	(null)
8	7839	KING	(null)
9	7844	TURNER	30
10	7900	JAMES	30
11	7902	FORD	20
12	7934	MILLER	(null)

emp_c 테이블

	DEPTNO	DNAME	LOC
1	20	RESEARCH	DALLAS
2	30	SALES	CHICAGO
3	40	OPERATIONS	BOSTON

dept_c 테이블

7번에서 설정한 **on delete set null** 로 dept_c의 10번 부서를 삭제했을때 이 10번 부서를 참조하고 있던 emp_c 테이블값들이 null을 가지는것을 볼수있다.

9. emp_c_deptno_fk 제약조건을 삭제하시오.

```
alter table emp_c
drop constraint emp_c_dept_fk;
```

10. emp_c테이블에서 deptno 속성이 dept_c 테이블의 deptno를 참조하는 외래키로 설정 하시오. 단, dept테이블의 deptno가 삭제될 때 emp_c테이블의 해당 deptno를 연쇄 삭제하도록 하시오

```
alter table emp_c
add constraint emp_c_deptno_fk foreign key(deptno)
references dept(deptno) on delete cascade;
```

11. dept_c 테이블에서 20번 부서를 삭제하고 두 테이블의 변경 사항을 관찰하시오.

```
delete from dept_c
where deptno = 20;
```

	EMPNO	ENAME	DEPTNO
1	7499	ALLEN	30
2	7521	WARD	30
3	7654	MARTIN	30
4	7698	BLAKE	30
5	7782	CLARK	(null)
6	7839	KING	(null)
7	7844	TURNER	30
8	7900	JAMES	30
9	7934	MILLER	(null)

emp_c 테이블

	DEPTNO	DNAME	LOC
1	30	SALES	CHICAGO
2	40	OPERATIONS	BOSTON

dept_c 테이블

10번에서 설정한 **on delete cascade** 옵션으로 인해 dept_c의 20번 부서가 삭제될 때 이를 참조하고 있는 emp_c의 데이터도 함께 삭제되었다.

12. 10번의 외래키 제약 조건을 disable 하시오. 단 diable 후에 insert/update/delete 등의 문장은 수행되도록 설정하시오.

```
alter table emp_c
disable novalidate constraint emp_c_deptno_fk;
```

- disable을 통해 현재 부여되어있는 제약조건을 해제한다.



외래키를 해제하여 사용하는것은 위험하다.

▼ 외래키 해제 위험 사유

FOREIGN KEY를 DISABLE(비활성화) 시키면 해당 외래 키 제약 조건이 무시되거나 체크되지 않게 됩니다. 즉, FOREIGN KEY 제약 조건을 확인하지 않고 데이터를 변경할 수 있게 됩니다.

FOREIGN KEY는 관계형 데이터베이스에서 테이블 간의 참조 무결성을 유지하기 위한 제약 조건입니다. 이 제약 조건은 일반적으로 부모 테이블의 기본 키와 자식 테이블의 외래 키 간의 일치 여부를 확인합니다. FOREIGN KEY가 활성화된 상태에서는 데이터베이스 시스템은 해당 제약 조건을 검사하여 부모-자식 관계가 유효한지 확인하고, 잘못된 데이터 변경을 방지합니다.

하지만 FOREIGN KEY를 DISABLE 시키면 이러한 검사가 수행되지 않으므로, 다음과 같은 결과가 발생할 수 있습니다:

부모-자식 관계에 대한 일관성 손실: 비활성화된 FOREIGN KEY로 인해 잘못된 참조가 발생할 수 있으며, 이로 인해 부모-자식 관계에 대한 일관성 손실이 발생할 수 있습니다. 데이터 무결성 위험: 비활성화된 FOREIGN KEY로 인해 유효하지 않은 데이터가 삽입되거나 업데이트될 수 있으며, 이는 데이터 무결성 위험을 초래할 수 있습니다. 참조 무결성 보장 문제: 비활성화된 FOREIGN KEY로 인해 외래 키와 기본 키 간의 참조 무결성 보장이 되지 않습니다.

FOREIGN KEY를 DISABLE 상태로 유지하는 것은 예외적인 경우에만 권장됩니다. 주로 대규모 데이터 마이그레이션 작업 등에서 사용되며, 작업 후에는 반드시 다시 활성화하여 데이터 일관성과 참조 무결성을 복원해야 합니다.

13. 아래 문장을 수행해서 dept_c 에 없는 부서번호인 50번 부서를 입력할 때 어떤 상황이 생기는지 관찰하시오.

```
insert into emp_c values(9000, '홍길동', 50) ;
```



disable novalidate로 인해서 값이 삽입된다.

원래이면, 없는 부서번호 50번이 들어왔을때 참조 무결성 위배를 방지하기 위해 제한된다.

14. 12번에서 disable 한 제약조건을 다시 enable하시오.

단, 이전에 입력된 값들에 대한 검사를 하지 않도록 하시오.

```
alter table emp_c  
enable novalidate constraint emp_c_deptno_fk;
```

- **novalidate**를 하지 않으면 현재 이미, 참조 무결성을 위배하는 값이 삽입되어 있으므로 허용하지 제약조건 enable을 허용하지 않는다.
- 따라서, novalidate를 통해 이전값들에 대해 검사하지 않는 조건으로 enable 가능하게 된다.