

# 1team project

---

학사 관리 데이터 베이스

# 프로젝트 구성

---

각 테이블들의 연관 관계



## 학사관리

### 테이블

kmajor

prop

pupil

mclass

classinfo

### kmajor

<u>num</u>	number	not null
name	varchar2	not null

### prop

<u>num</u>	number	not null
name	varchar2	not null
majornum	number	not null

### mclass

<u>num</u>	number	not null
name	varchar2	not null
score	number	
<u>mpnum</u>	number	not null

### pupil

<u>num</u>	number	not null
name	varchar2	not null
addr	varchar2	not null
classes	number	
ent	date	
<u>majornum</u>	number	not null

### pupil

<u>stunum</u>	number	not null
<u>clsnum</u>	number	not null
score	number	
ranks	varchar2	

학과

교수

과목

학생

수강 내역

## 개발 정보

### 테이블

kmajor	prop	pupil	mclass	classinfo
--------	------	-------	--------	-----------

### kmajor

num	number	not null	primary key	
name	varchar2	not null		

### prop

num	number	not null	primary key	
name	varchar2	not null		
majornum	number	not null	foreign key references kmajor (num)	

### pupil

num	number	not null	primary key	
name	varchar2	not null		
addr	varchar2	not null		
classes	number			
ent	date			
majornum	number	not null	foreign key references kmajor (num)	

### mclass

num	number	not null	primary key	
name	varchar2	not null		
score	number			
mpnum	number	not null	foreign key references prop (num)	

### classinfo

stunum	number	notnull	pk	foreign pupil (num)
clsnum	number	notnull	pk	foreign mclass (num)
score	number			
ranks	varchar2			

kmajor

- Num - pk

- Name

prop

- Num - pk

- name

- Major num - fk

- Num - pk

Pupil

- name

- Addr

- Classes

- Ent

- Num - pk

mclass

- Name

- Score

- Mpnum - fk

- Stunum - pk

classinfo

- Clsnum - pk

- Score

- ranks

# Table Cord

```
CREATE TABLE kmajor (      --학과
    num  NUMBER NOT NULL PRIMARY KEY,  --학과번호
    name VARCHAR2(50) NOT NULL        --학과명
);
```

```
CREATE TABLE prop (      --교수
    num      NUMBER NOT NULL PRIMARY KEY,  --교수번호
    name     VARCHAR2(50) NOT NULL,        --교수명
    majornum NUMBER NOT NULL,              --전공번호
    FOREIGN KEY ( majornum )
        REFERENCES kmajor ( num )
);
```

```
CREATE TABLE pupil (      --학생
    num      NUMBER NOT NULL PRIMARY KEY,  --학생번호
    name     VARCHAR2(50) NOT NULL,        --학생명
    addr     VARCHAR2(100) NOT NULL,       --주소
    classes  NUMBER,                      --학년
    ent      DATE,                        --입력 일자
    majornum NUMBER NOT NULL,              --학과번호
    FOREIGN KEY ( majornum )
        REFERENCES kmajor ( num )
);
```

```
CREATE TABLE mclass (      --과목
    num      NUMBER NOT NULL PRIMARY KEY,  --과목번호
    name     VARCHAR2(50) NOT NULL,        --과목명
    score    NUMBER,                      --학점
    mpnum    NUMBER NOT NULL,              --담당교수번호
    FOREIGN KEY ( mpnum )
        REFERENCES prop ( num )
);
```

```
CREATE TABLE classinfo (      --수강 내역
    stunum   NUMBER NOT NULL, --학생번호
    clsnum   NUMBER NOT NULL, --과목번호
    score    NUMBER(4, 1),    --점수
    ranks    VARCHAR2(10),    --학점
    CONSTRAINT classinfo_pk PRIMARY KEY ( stunum,
                                            clsnum ),
    FOREIGN KEY ( stunum )
        REFERENCES pupil ( num ),
    FOREIGN KEY ( clsnum )
        REFERENCES mclass ( num )
);
```

# Insert Procedure Cord

```
/      --수강내역
create or replace procedure inclassinfo(
    in_stunum in number,
    in_clsnum number,
    in_score number,
    in_ranks varchar2
)is begin
insert into classinfo values(
    in_stunum,
    in_clsnum,
    in_score,
    in_ranks
);
end inclassinfo;
/
```

```
/ -- 학과등록
CREATE OR REPLACE PROCEDURE inmajor (
    in_num  IN NUMBER,
    in_name VARCHAR2
) IS
BEGIN
    INSERT INTO kmajor VALUES (
        in_num,
        in_name
    );

END inmajor;
/
/ --교수등록
CREATE OR REPLACE PROCEDURE inprop (
    in_num in number,
    in_name varchar2,
    in_majornum number
) IS
BEGIN
    INSERT INTO prop VALUES (
        in_num,
        in_name,
        in_majornum
    );

END inprop;
/
```

```
/ --학생등록
create or replace procedure inpupil(
    in_num in number,
    in_name varchar2,
    in_addr varchar2,
    in_classes number,
    in_ent date,
    in_majornum number
)is begin
insert into pupil values(
    in_num,
    in_name,
    in_addr,
    in_classes,
    in_ent,
    in_majornum
);
end inpupil;
/
/ -- 과목등록
create or replace procedure inclass(
    in_num in number,
    in_name varchar2,
    in_score number,
    in_mpnum number
)is begin
insert into mclass values(
    in_num,
    in_name,
    in_score,
    in_mpnum
);
end inclass;
/
```



# Delete Procedure Cord

```
--교수
create or replace procedure delprop(
    in_num in number
)is begin
delete from prop where num = in_num;
end delprop;
/

--학과
create or replace procedure delmajor(
    in_num in number
)is begin
delete from kmajor where num = in_num;
end delmajor;
/
```

```
--수강
/
create or replace procedure delclassinfo(
    in_stunum in number
)is begin
delete from classinfo where stunum = in_stunum;
end delclassinfo;
/

--과목
create or replace procedure delclass(
    in_num in number
)is begin
delete from mclass where num = in_num;
end delclass;
/

--학생
create or replace procedure delpupil(
    in_num in number
)is begin
delete from pupil where num = in_num;
end delpupil;
/
```

# Update Procedure Cord

```
/
create or replace procedure upclassinfo( -- 수강내역 점수, 학점 변경
    in_stunum in number,
    in_score number,
    in_ranks varchar2
)is begin
update classinfo set
    score = in_score,
    ranks = in_ranks
where stunum = in_stunum;
end upclassinfo;
/
/
create or replace procedure upclass( -- 과목 수강학점 변경
    in_num in number,
    in_score number
)is begin
update mclass set
    score = in_score
where num = in_num;
end upclass;
/
```

```
/
create or replace procedure uppupil( --학생 정보 수정
    in_num in number,
    in_addr varchar2,
    in_classes number,
    in_ent date,
    in_majornum number
)is begin
update pupil set
    addr = in_addr,
    classes = in_classes,
    ent = in_ent,
    majornum = in_majornum
where num=in_num;
end uppupil;
/
/
create or replace procedure upprop(
    in_num in number,
    in_majornum number
)is begin
update prop set
    majornum = in_majornum
where num = in_num;
end upprop;
/
```



# Trigger Cord

```
/
CREATE OR REPLACE TRIGGER in_classinfo BEFORE --수강 내역 등록
    insert ON classinfo
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 수강내역 등록 완료. ***');
END;
/

-----삭제 -----
/
CREATE OR REPLACE TRIGGER in_major BEFORE --학과
    delete ON kmajor
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 학과 삭제 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_prop BEFORE --교수
    delete ON prop
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 교수 삭제 완료. ***');
END;
/
```

```
CREATE OR REPLACE TRIGGER in_major BEFORE --학과 등록
    insert ON kmajor
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 학과 등록 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_prop BEFORE --교수 등록
    insert ON prop
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 교수 등록 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_pupil BEFORE --학생 등록
    insert ON pupil
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 학생 등록 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_class BEFORE --과목 등록
    insert ON mclass
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 과목 등록 완료. ***');
END;
/
```

# Trigger Cord

```
/
CREATE OR REPLACE TRIGGER in_pupil BEFORE --학생
    delete ON pupil
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 학생 삭제 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_class BEFORE --과목
    delete ON mclass
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 과목 삭제 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_classinfo BEFORE --수강 내역
    delete ON classinfo
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 수강내역 삭제 완료. ***');
END;
/
```

```
/
CREATE OR REPLACE TRIGGER in_major BEFORE --학과
    update ON kmajor
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 학과 수정 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_prop BEFORE --교수
    update ON prop
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 교수 수정 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_pupil BEFORE --학생
    update ON pupil
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 학생 수정 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_class BEFORE --과목
    update ON mclass
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 과목 수정 완료. ***');
END;
/
/
CREATE OR REPLACE TRIGGER in_classinfo BEFORE --수강 내역
    update ON classinfo
    FOR EACH ROW
DECLARE BEGIN
    dbms_output.put_line('*** 수강내역 수정 완료. ***');
end;
/
```

# Test Cord

```
exec inclassinfo(1,2,80,'B');
```

```
SELECT
    *
FROM
    classinfo;
```

```
select name as "이름", addr as "주소" from pupil;
```

```
select * from pupil where name like '이%';
```

```
select * from pupil where by name asc;
```

```
select clsnum, avg(score) as "과목평균점수" from classinfo group by clsnum order by clsnum;
select * from pupil where group by addr like '%대구%' having name like '이%';
```

```
create or replace view 학생 as
(
    select num, name, classes, ent
    from pupil
);
```

```
exec inpupil(sq.nextval, 'kim', 'asdf asdfas',1,'2023-10-20',2);
exec inpupil(sq.nextval, 'im', 'asdf asdfas',2,'2023-10-20',1);
exec inpupil(sq.nextval, 'moon', 'asdf asdfas',3,'2023-10-20',3);
```

```
SELECT
    *
FROM
    pupil;
```

```
exec inclass(sq.nextval, 'SQL활용',3,5);
exec inclass(sq.nextval, 'java활용',3,1);
exec inclass(sq.nextval, 'C활용',2,2);
```

```
SELECT
    *
FROM
    mclass;
```

```
CREATE SEQUENCE sq;
```

```
exec inmajor(sq.nextval, 'Java');
exec inmajor(sq.nextval, 'C');
exec inmajor(sq.nextval, 'C++');
exec inmajor(sq.nextval, 'python');
exec inmajor(sq.nextval, 'sql');
exec inmajor(sq.nextval, 'script');
```

```
SELECT
    *
FROM
    kmajor;
```

```
exec inprop(sq.nextval, 'Lee', 1);
exec inprop(sq.nextval, 'kim', 1);
exec inprop(sq.nextval, 'im', 2);
exec inprop(sq.nextval, 'jung', 2);
exec inprop(sq.nextval, 'Lee', 3);
exec inprop(sq.nextval, 'Lee', 4);
exec inprop(sq.nextval, 'Lee', 5);
exec inprop(sq.nextval, 'Lee', 6);
```

```
SELECT
    *
FROM
    prop;
```



## 차후 계획

후에 있을 **C#** 프로젝트에서 학사관리 프로그램을 제작  
위 데이터 베이스를 연동하여 작동하여 실행 내용을 저장 및  
불러오기가 가능하도록 제작하는 것이 목표입니다

---

## SQL 프로젝트 소감



권 병 진

•타 언어와 다르게 문법이 직설적이며, 영어 문법 그 자체같은 느낌이 강해 생각보다 어려웠다.



백 정 윤

•기존 C언어와 JAVA와 달리 개념이 다른 프로그램이라서 생소했지만 구현해보니까 즐거웠습니다.



홍 주 환

•SQL문법과 기능에 대해서 새로 알게 되었고,몰랐던 기능들을 배우는 것이 흥미로웠습니다



신 예 원

•데이터를 직접 다루는 프로그래밍이 매우 흥미롭게 느껴졌습니다.



END

