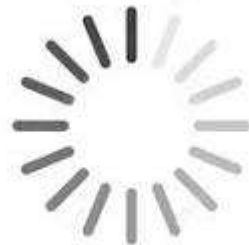




# 학생 관리 프로그램 DB



Loading

3조  
추형욱 한형빈 조민석

# contents

1. 프로젝트 개요
2. 구성원 역할 소개
3. 프로젝트 수행 절차
4. 프로젝트 수행 결과
5. 자체 평가 의견

# 프로젝트 개요

프로젝트 명 : 학생 관리 프로그램  
DataBase

목표 : sql 문법 , 함수, 프로시저를 사용  
하여  
학생, 반, 성적 을 관리하기 위한 데이터베  
이스 구축

기간 : 2024 - 02 - 29 ~ 2024 - 03 -  
07



**추형욱(조장)**

기능 구현

데이터 수집

GUI

다이아그램

트리거



**조민석**

기능 구현

데이터 수집

테이블 코드

기타 문법

함수

프로시저



**한형빈**

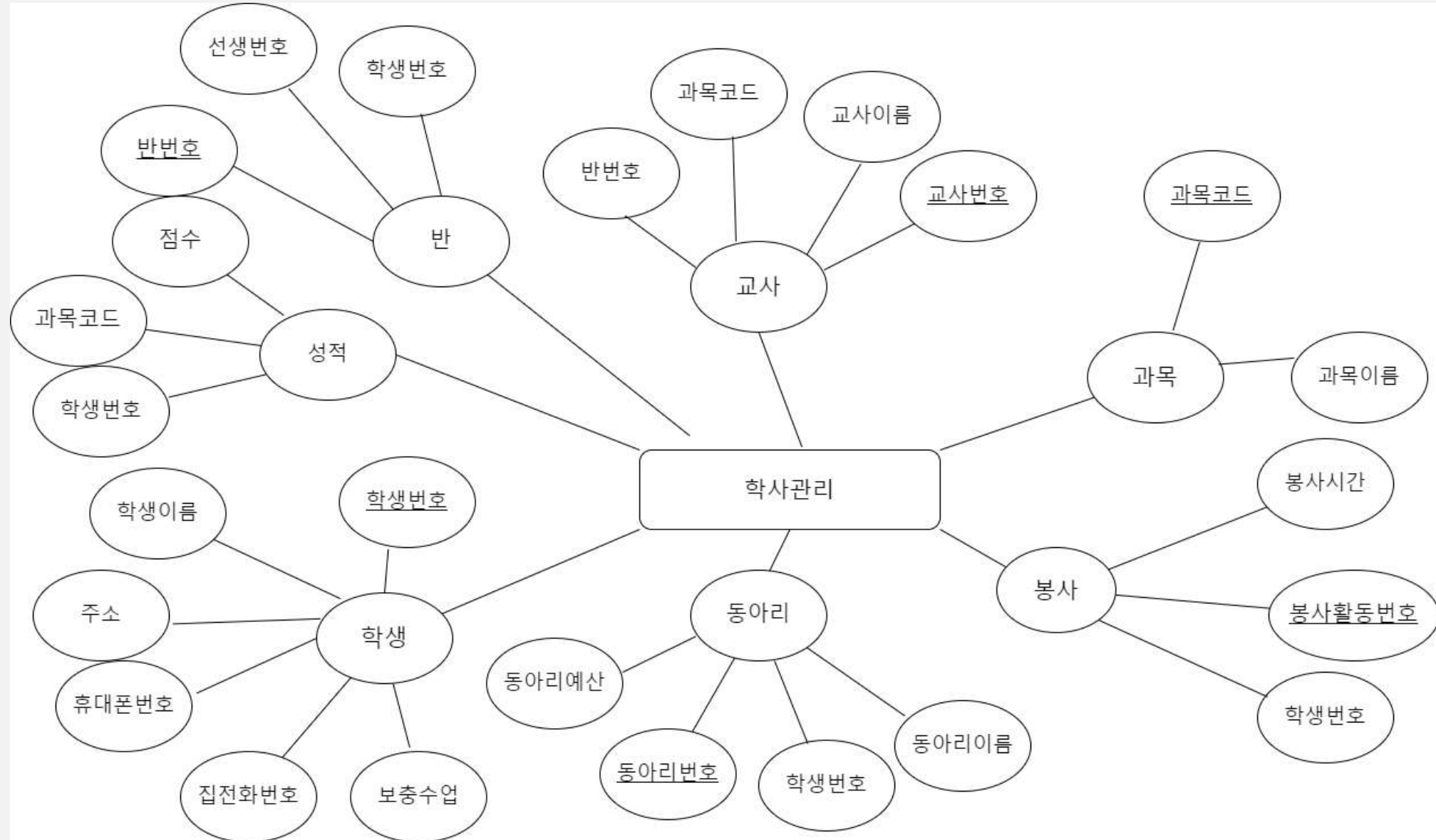
기능 구현

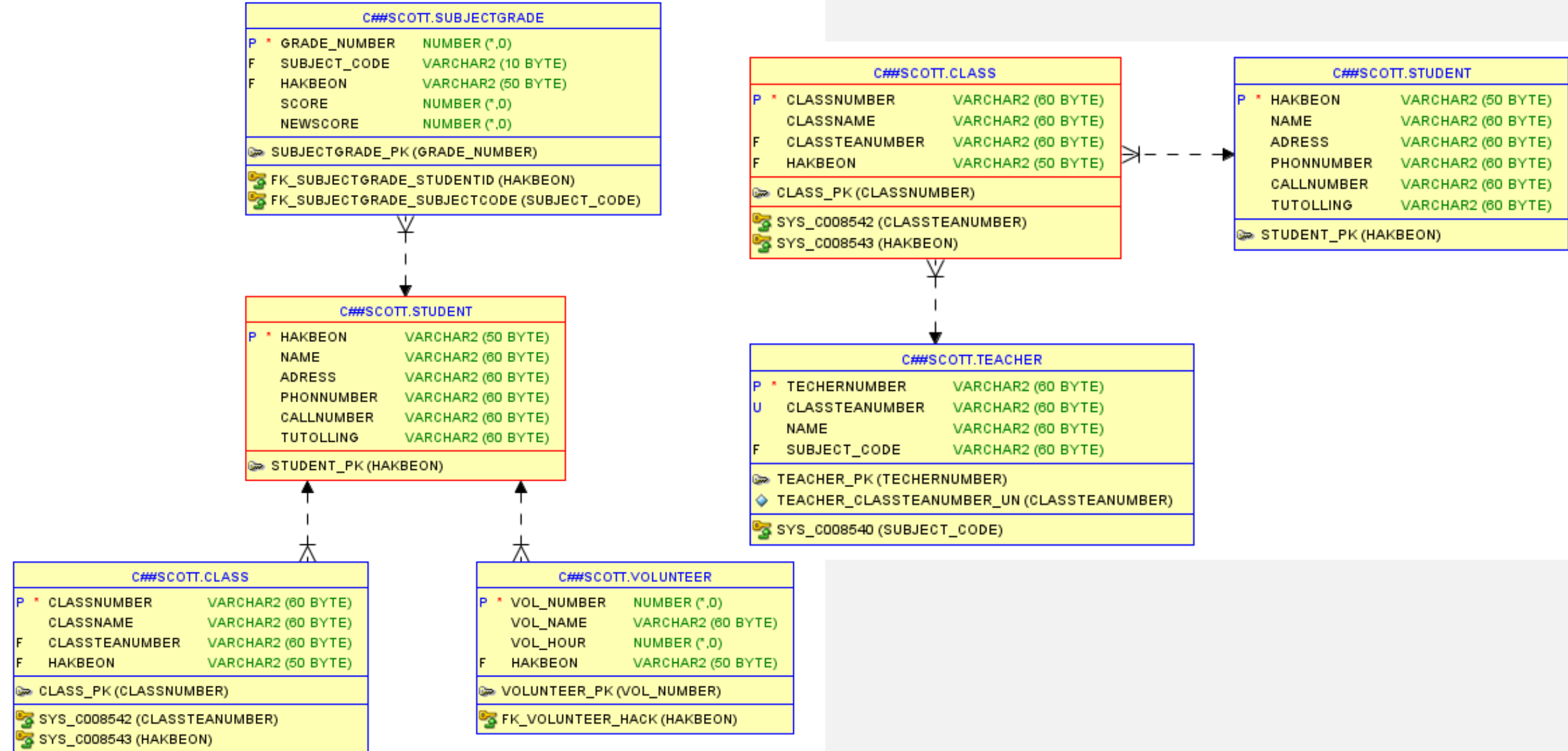
데이터 수집

요구사항 정의서

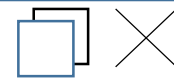
테이블 정의서

회의록 작성





# 프로젝트 절차



```
CREATE TABLE Student (  
    hakbeon VARCHAR2(50) PRIMARY KEY,  
    name VARCHAR2(60) not null,  
    address VARCHAR2(60) not null,  
    phonenumber VARCHAR2(60) not null unique,  
    callnumber VARCHAR2(60) unique,  
    tutolling VARCHAR2(60)  
);
```

| 테이블 명 | Student         |             |              |    |         |        |  |
|-------|-----------------|-------------|--------------|----|---------|--------|--|
| 설명    | 학생 정보를 관리       |             |              |    |         |        |  |
| No    | 컬럼 id           | 컬럼 명        | 타입           | 길이 | Null    | Key    |  |
| 1     | hakbeon         | 학생<br>고유 번호 | VARCHAR<br>2 | 50 | NotNull | PK     |  |
| 2     | name            | 이름          | VARCHAR<br>2 | 60 | NotNull |        |  |
| 3     | address         | 주소          | VARCHAR<br>2 | 60 | NotNull |        |  |
| 4     | Phonen<br>umber | 전화번호        | VARCHAR<br>2 | 60 | NotNull | unique |  |
| 5     | Callnum<br>ber  | 집<br>전화번호   | VARCHAR<br>2 | 60 | Null    | Unique |  |
| 6     | tutolling       | 보충 수업<br>시간 | VARCHAR<br>2 | 60 | Null    |        |  |

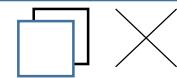


```
CREATE TABLE subject(  
subject_code varchar2(60) primary key,  
subject_name varchar2(60) not null  
);
```

```
CREATE TABLE teacher(  
teachernumber varchar2(60) primary key,  
classteaumber varchar2(60) unique,  
name varchar2(60) not null,  
subject_code varchar2(60),  
foreign key(subject_code)  
references subject(subject_code)  
);
```

| 테이블 명 | subject      |          |           |    |         |     |
|-------|--------------|----------|-----------|----|---------|-----|
| 설명    | 과목 정보를 관리    |          |           |    |         |     |
| No    | 컬럼 id        | 컬럼 명     | 타입        | 길이 | Null    | Key |
| 1     | Subject_code | 과목 고유 번호 | VARCHAR 2 | 60 | NotNull | PK  |
| 2     | Subject_name | 과목 이름    | VARCHAR 2 | 60 | NotNull |     |

| 테이블 명 | Teacher       |            |           |    |         |        |
|-------|---------------|------------|-----------|----|---------|--------|
| 설명    | 교사 정보를 관리     |            |           |    |         |        |
| No    | 컬럼 id         | 컬럼 명       | 타입        | 길이 | Null    | Key    |
| 1     | teachernumber | 교사 고유 번호   | VARCHAR 2 | 60 | NotNull | PK     |
| 2     | classteaumber | 교사 반 배정 번호 | VARCHAR 2 | 60 | Null    | unique |
| 3     | name          | 교사 이름      | VARCHAR 2 | 60 | NotNull |        |
| 4     | Subject_code  | 과목코드       | VARCHAR 2 | 60 | Null    | FK     |



```
CREATE TABLE class (  
    classnumber varchar2(60) PRIMARY KEY,  
    classname varchar2(60) not null,  
    classtea number varchar2(60) not null,  
    foreign key(classtea number) references  
teacher(classtea number),  
    hakbeon varchar2(50),  
    foreign key(hakbeon)  
references student(hakbeon)  
);
```

| 테이블 명 | class           |            |           |    |         |        |
|-------|-----------------|------------|-----------|----|---------|--------|
| 설명    | 반 정보를 관리        |            |           |    |         |        |
| No    | 컬럼 id           | 컬럼 명       | 타입        | 길이 | Null    | Key    |
| 1     | class number    | 반 고유 번호    | VARCHAR 2 | 60 | NotNull | PK     |
| 2     | class name      | 반 이름       | VARCHAR 2 | 60 | Null    | unique |
| 3     | classtea number | 교사 반 배정 번호 | VARCHAR 2 | 60 | NotNull | FK     |
| 4     | hakbeon         | 학생 고유 번호   | VARCHAR 2 | 50 | Null    | FK     |

```
CREATE TABLE volunteer(  
vol_number int primary key,  
vol_name varchar2(60) not null,  
vol_hour int,  
hakbeon varchar2(50),  
constraint fk_volunteer_hack foreign key (hakbeon)  
references Student(hakbeon)  
);
```

| 테이블 명 | volunteer       |            |          |    |         |        |
|-------|-----------------|------------|----------|----|---------|--------|
| 설명    | 반 정보를 관리        |            |          |    |         |        |
| No    | 컬럼 id           | 컬럼 명       | 타입       | 길이 | Null    | Key    |
| 1     | classnumber     | 반 고유 번호    | VARCHAR2 | 60 | NotNull | PK     |
| 2     | classname       | 반 이름       | VARCHAR2 | 60 | Null    | unique |
| 3     | classtea number | 교사 반 배정 번호 | VARCHAR2 | 60 | NotNull | FK     |
| 4     | hakbeon         | 학생 고유 번호   | VARCHAR2 | 50 | Null    | FK     |

```
CREATE TABLE SubjectGrade (  
  grade_number INT PRIMARY KEY,  
  subject_code VARCHAR2(10) NotNull,  
  hakbeon VARCHAR2(50) NotNull,  
  score INT NotNull,  
  CONSTRAINT FK_SubjectGrade_SubjectCode FOREIGN KEY  
    (subject_code) REFERENCES Subject(subject_code),  
  CONSTRAINT FK_SubjectGrade_StudentID FOREIGN KEY  
    (hakbeon) REFERENCES Student(hakbeon)  
);
```

| 테이블 명 | SubjectGrade |          |           |    |         |     |  |
|-------|--------------|----------|-----------|----|---------|-----|--|
| 설명    | 점수를 관리       |          |           |    |         |     |  |
| No    | 컬럼 id        | 컬럼 명     | 타입        | 길이 | Null    | Key |  |
| 1     | Grade_number | 성적 고유 번호 | INT       | 1  | NotNull | PK  |  |
| 2     | score        | 점수       | INT       | 1  | NotNull |     |  |
| 3     | hakbeon      | 학생 고유 번호 | VARCHAR 2 | 50 | NotNull | FK  |  |
| 4     | Subject_code | 과목 고유 번호 | VARCHAR 2 | 10 | NotNull | FK  |  |

# 프로젝트 결과

각 반 당 점수가 가장 높은 학생

```
CREATE VIEW TopScorerPerClassView AS
SELECT
    className,
    studentName,
    hakbeon,
    score
FROM (
    SELECT
        c.className,
        s.name AS studentName,
        sg.hakbeon,
        sg.score,
        RANK() OVER (PARTITION BY c.className ORDER BY sg.score DESC) AS ranking
    FROM
        Class c
    JOIN
        Student s ON c.hakbeon = s.hakbeon
    JOIN
        SubjectGrade sg ON s.hakbeon = sg.hakbeon
) ranked
WHERE
    ranking = 1;

SELECT * FROM TopScorerPerClassView;
```

|   | CLASSNAME | STUDENTNAME | HAKBEON | SCORE |
|---|-----------|-------------|---------|-------|
| 1 | 1학년 1반    | 강예준         | 1006    | 95    |
| 2 | 1학년 2반    | 정하윤         | 2005    | 93    |
| 3 | 1학년 3반    | 정민서         | 3005    | 94    |

```
CREATE VIEW TopVolunteers AS
SELECT hakbeon, vol_hour
FROM Volunteer
ORDER BY vol_hour DESC
FETCH FIRST 10 PERCENT ROWS ONLY;

CREATE VIEW TopGrades AS
SELECT hakbeon, score
FROM SubjectGrade
ORDER BY score DESC
FETCH FIRST 10 PERCENT ROWS ONLY;
```

```
CREATE VIEW TopStudents AS
SELECT s.hakbeon, s.name
FROM Student s
JOIN TopVolunteers v ON s.hakbeon = v.hakbeon
JOIN TopGrades g ON s.hakbeon = g.hakbeon;
```

```
-- TopStudents 뷰 조회
```

```
SELECT * FROM TopStudents;
```

|   | HAKBEON | NAME |
|---|---------|------|
| 1 | 3005    | 정민서  |
| 2 | 2005    | 정하윤  |

```
SELECT s.name as 학생_이름, c.classname as 반_이름, SUM(v.vol_hour) as 총_봉사_시간
FROM Student s
JOIN class c ON s.hakbeon = c.hakbeon
JOIN volunteer v ON s.hakbeon = v.hakbeon
GROUP BY s.name, c.classname
HAVING SUM(v.vol_hour) >= 30;
```

|   | 학생_이름 | 반_이름   | 총_봉사_시간 |
|---|-------|--------|---------|
| 1 | 정민서   | 1학년 1반 | 30      |
| 2 | 정하윤   | 1학년 2반 | 30      |
| 3 | 최도윤   | 1학년 3반 | 30      |
| 4 | 정민서   | 1학년 3반 | 35      |



--트리거용 테이블 생성

```
create table scorediffrence(  
hakbeon2 VARCHAR2(50),  
oldscore int,  
newscore int,  
regdate timestamp default sysdate  
);
```

```
CREATE OR REPLACE TRIGGER UPDATE_SCORE  
after update of score ON SubjectGrade  
FOR EACH ROW  
BEGIN  
insert into scorediffrence(hakbeon2,oldscore,newscore) values (:old.hakbeon,:old.score,:new.score);  
END;  
/
```

-- 트리거를 생성한다 hakbeon2는 수정한학번을 oldscore는 원래입력되어있던 성적값 newscore는 새로입력한값이다

```
create view gapmax as select hakbeon2,max(newscore-oldscore)gap2 from scorediffrence group by hakbeon2;
```

--view를 만들어각반의 max값을 찾는다

```
create view scoregapbyclass as select distinct classname,name,score as 이번성적,  
gap2 as 지난시험과성적차이 from SubjectGrade  
join student on student.hakbeon=SubjectGrade.hakbeon  
join class on class.hakbeon=SubjectGrade.hakbeon  
join gapmax on SubjectGrade.hakbeon=gapmax.hakbeon2;
```

|   | CLASSNAME | NAME | 이번성적 | 지난시험과성적차이 |
|---|-----------|------|------|-----------|
| 1 | 1학년 1반    | 김지우  | 85   | 15        |
| 2 | 1학년 2반    | 서예성  | 98   | 27        |
| 3 | 1학년 3반    | 김하울  | 100  | 27        |

프로시저 사용

```
CREATE OR REPLACE PROCEDURE GetStudentsByTutolling(  
    tutolling_value VARCHAR2  
) AS  
BEGIN  
    FOR student_rec IN (SELECT * FROM Student WHERE tutolling = tutolling_value) LOOP  
        DBMS_OUTPUT.PUT_LINE('Hakbeon: ' || student_rec.hakbeon || ', Name: ' || student_rec.name || ', Tutolling: ' || student_rec.tutolling);  
    END LOOP;  
END GetStudentsByTutolling;  
/  
  
DECLARE  
    tutolling_value VARCHAR2(10) := '2시간'; -- 원하는 tutolling 값으로 설정  
BEGIN  
    GetStudentsByTutolling(tutolling_value);  
END;  
/  
  
-- DBMS_OUTPUT을 활성화  
SET SERVEROUTPUT ON;  
  
-- 프로시저 호출  
EXEC GetStudentsByTutolling('2시간');
```

PL/SQL 프로시저가 성공적으로 완료되었습니다.

Hakbeon: 2003, Name: 박지영, Tutolling: 2시간  
Hakbeon: 3004, Name: 최도윤, Tutolling: 2시간

## 함수 사용

```
CREATE OR REPLACE FUNCTION AddStudent(
    hakbeon_in IN VARCHAR2,
    name_in IN VARCHAR2,
    address_in IN VARCHAR2,
    phone_number_in IN VARCHAR2
)
RETURN VARCHAR2
AS
BEGIN
    -- 전화번호 형식 체크
    IF NOT REGEXP LIKE(phone_number_in, '^\d{3}-\d{4}-\d{4}$') THEN
        RAISE_APPLICATION_ERROR(-20001, '전화번호 형식이 올바르지 않습니다. (예: 010-1234-5678)');
        RETURN NULL;
    END IF;

    -- 학번이 입력되었는지 체크
    IF hakbeon_in IS NULL OR hakbeon_in = '' THEN
        RAISE_APPLICATION_ERROR(-20002, '학번은 필수 입력 사항입니다.');
```

```
        RETURN NULL;
    END IF;

    -- 학번이 유니크한지 체크
    DECLARE
        hakbeon_count INT;
    BEGIN
        SELECT COUNT(1) INTO hakbeon_count
        FROM Student
        WHERE hakbeon = hakbeon_in;

        IF hakbeon_count > 0 THEN
            RAISE_APPLICATION_ERROR(-20003, '이미 존재하는 학번입니다. 다른 학번을 입력하세요.');
```

```
            RETURN NULL;
        END IF;
    END;
```

-- 새로운 학생을 추가

```
INSERT INTO Student (hakbeon, name, address, phonnumber)
VALUES (hakbeon_in, name_in, address_in, phone_number_in);
```

```
RETURN hakbeon_in;
```

```
END;
```

```
/
```

```
student_name VARCHAR2(60) := '조민석';  
student_address VARCHAR2(60) := '대구 북구';  
student_phone_number VARCHAR2(60) := '010-178';  
student_hakbeon VARCHAR2(50) := '3014'; -- 사용자 입력 또는 자동 생성
```

오류 보고 -

ORA-20001: 전화번호 형식이 올바르지 않습니다. (예: 010-1234-5678)  
ORA-06512: "C##SCOTT.ADDSTUDENT", 12행  
ORA-06512: 9행

```
student_name VARCHAR2(60) := '조민석';  
student_address VARCHAR2(60) := '대구 북구';  
student_phone_number VARCHAR2(60) := '010-0000-1780';  
student_hakbeon VARCHAR2(50) := '3013'; -- 사용자 입력 또는 자동 생성
```

오류 보고 -

ORA-20003: 이미 존재하는 학번입니다. 다른 학번을 입력하세요.  
ORA-06512: "C##SCOTT.ADDSTUDENT", 31행  
ORA-06512: 9행

## 함수 사용

```

DECLARE
  student_name VARCHAR2(60) := '조민석';
  student_address VARCHAR2(60) := '주소';
  student_phone_number VARCHAR2(60) := '010-1111-5678';
  student_hakbeon VARCHAR2(50) := '3016'; -- 사용자 입력 또는 자동 생성

  result_hakbeon VARCHAR2(50);
BEGIN
  result_hakbeon := AddStudent(student_hakbeon, student_name, student_address, student_phone_number);
  IF result_hakbeon IS NOT NULL THEN
    DBMS_OUTPUT.PUT_LINE('새로운 학생이 추가되었습니다. 학번: ' || result_hakbeon);
  END IF;
END;
/

```

PL/SQL 프로시저가 성공적으로 완료되었습니다.

|    |      |     |    |               |        |        |  |
|----|------|-----|----|---------------|--------|--------|--|
| 31 | 3016 | 조민석 | 주소 | 010-1111-5678 | (null) | (null) |  |
|----|------|-----|----|---------------|--------|--------|--|

# 인덱스

```
create index idx_score on SubjectGrade(score DESC);
```

```
select hakbeon, score
from subjectgrade
where score >=0;
```

|    | HAKBEON | SCORE |
|----|---------|-------|
| 1  | 1006    | 95    |
| 2  | 3005    | 94    |
| 3  | 2005    | 93    |
| 4  | 2007    | 91    |
| 5  | 1010    | 90    |
| 6  | 3008    | 90    |
| 7  | 3001    | 89    |
| 8  | 1005    | 88    |
| 9  | 2002    | 87    |
| 10 | 2009    | 86    |
| 11 | 3010    | 85    |
| 12 | 1009    | 84    |
| 13 | 3006    | 83    |
| 14 | 2006    | 82    |
| 15 | 3003    | 81    |
| 16 | 1007    | 80    |
| 17 | 2001    | 79    |
| 18 | 1003    | 78    |
| 19 | 2010    | 77    |
| 20 | 3007    | 76    |
| 21 | 2004    | 75    |

Index는 where을 사용할 때 발동

```
select hakbeon, score
from subjectgrade;
```

|    | HAKBEON | SCORE |
|----|---------|-------|
| 1  | 1002    | 70    |
| 2  | 1003    | 78    |
| 3  | 1004    | 70    |
| 4  | 1005    | 88    |
| 5  | 1006    | 95    |
| 6  | 1007    | 80    |
| 7  | 1008    | 72    |
| 8  | 1009    | 84    |
| 9  | 1010    | 90    |
| 10 | 2001    | 79    |
| 11 | 2002    | 87    |
| 12 | 2003    | 68    |
| 13 | 2004    | 75    |
| 14 | 2005    | 93    |
| 15 | 2006    | 82    |
| 16 | 2007    | 91    |
| 17 | 2008    | 73    |
| 18 | 2009    | 86    |
| 19 | 2010    | 77    |
| 20 | 3001    | 89    |
| 21 | 3002    | 74    |

```
select /*+ INDEX(student SYS_C008536) */
      a.hakbeon as 학번,
      a.name as 이름
      ,a.adress as 주소
      ,a.phonnumber as 전화번호,
      a.tutolling as 보충수업
from student a
where a.tutolling IN ('1시간', '2시간');
```

/\*+ \*/ 구절을 사용하면 인덱스 힌트를 사용할 수 있다.

|    | 학번   | 이름  | 주소      | 전화번호          | 보충수업 |
|----|------|-----|---------|---------------|------|
| 1  | 1003 | 박하준 | 서울시 서초구 | 010-3456-7890 | 1시간  |
| 2  | 1004 | 최도윤 | 서울시 송파구 | 010-4567-8901 | 1시간  |
| 3  | 1008 | 서시현 | 서울시 강동구 | 010-8901-2345 | 1시간  |
| 4  | 2001 | 김민지 | 서울시 관악구 | 010-9876-5432 | 1시간  |
| 5  | 2003 | 박지영 | 서울시 성북구 | 010-3116-7890 | 2시간  |
| 6  | 2004 | 최민준 | 서울시 강남구 | 010-4567-8901 | 1시간  |
| 7  | 2008 | 서예성 | 서울시 강서구 | 010-8901-2345 | 1시간  |
| 8  | 2010 | 박하준 | 서울시 성동구 | 010-0123-4567 | 1시간  |
| 9  | 3002 | 이서윤 | 서울시 서초구 | 010-2345-6789 | 1시간  |
| 10 | 3004 | 최도윤 | 서울시 강북구 | 010-4567-8901 | 2시간  |
| 11 | 3007 | 윤시우 | 서울시 관악구 | 010-7890-1234 | 1시간  |
| 12 | 3009 | 신예성 | 서울시 구로구 | 010-4714-3456 | 1시간  |



Form1 [www.BANDICAM.COM](http://www.BANDICAM.COM)

|    | 학번 | 이름 | 주소 | 전화번호 |
|----|----|----|----|------|
| ▶▶ |    |    |    |      |

학번

이름

주소

h.p

출력

입력

삭제(학번입력)

# 자체 평가 의견



### 추형욱(조장)

데이터베이스의 특성과 다른 언어와의 차이를 느낄수있었다  
팀원이만든데이터를 관리하는법을배울수있는좋은기회였다



### 조민석

데이터베이스에 관해 관심도 별로 없었고 내용도 제대로 알지 못했었는데 프로젝트를 진행하며 내용에 대해 많이 알게 되고 관심도 생긴 것 같아 좋습니다.



### 한형빈

데이터베이스 실습을 하며 문법에 어려움을 느끼고 있었는데 프로젝트를 진행하며 실력이 상승한 것 같아 좋습니다.

감사합니다.