**2020학년도 2학기 데이터베이스 설계 및 구축 최종 보고서**

**( 생활관 상벌점 관리 )의 데이터베이스**

**설계 및 구축**

|  |  |
| --- | --- |
| **학 과** | **컴퓨터정보과** |
| **분 반** | **3 반** |
| **과목명** | **데이터베이스** |
| **학 번** | **201823886** |
| **이 름** | **김 산** |
| **담당교수** | **김 경 민** |
| **제 출 일** | **2020년 12월 22일** |

****

**목 차**

1. 프로젝트 개요 ··························································································································· 3

2. 요구사항 분석 ··························································································································· 4

3. 개념적 데이터 모델 ················································································································ 5

4. 논리적 데이터 모델 ················································································································ 6

5. 용어 사전 정의 ························································································································· 7

6. 물리적 데이터 모델 ················································································································ 8

7. Table 기술서 ······························································································································· 9

8. SQL문 사용하기

1) 기본 SQL ···························································································································· 12

2) JOIN ······································································································································ 15

3) Sub Query ·························································································································· 16

4) Trigger ·································································································································· 19

9. 별첨(프로젝트 후기, Databace 구축 Dump 자료) ················································· 20

# 1. 프로젝트 개요

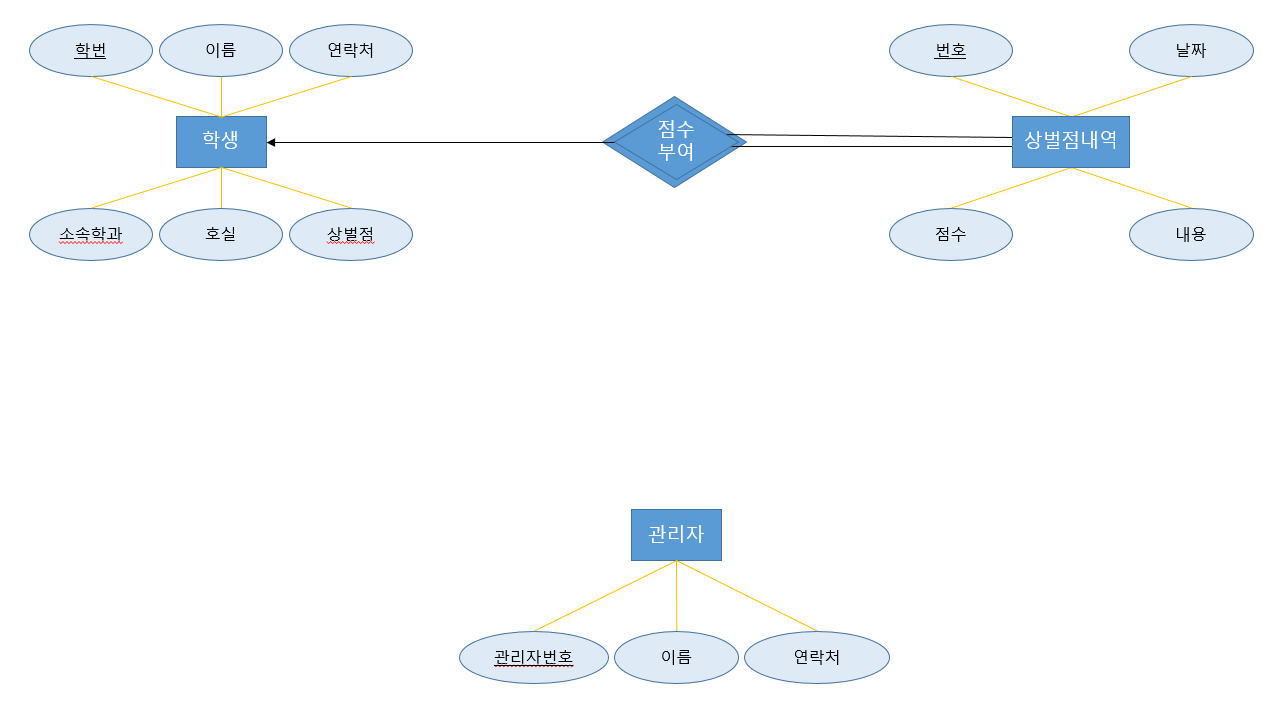
|  |  |
| --- | --- |
| 프로젝트 개요 | |
| **프로젝트 명** | 생활관 상벌점 관리 |
| 주제선정 이유 : 수업이 끝나고 무엇을 주제로 할지 고민하며 기숙사로 돌아가고 있을 때  엘리베이터에 기숙사 학생들 중 벌점이6점이상인 학생들을 알리는 알림문이  붙어있는것을 보고 ‘아 이거다! 이거야!!’라고 생각해서 선정하게 되었습니다.  프로젝트 소개 : 기숙사학생들의 상벌점을 관리해주는 시스템입니다.  개대효과 : 상벌점 관리 시스템을 통해 더 쉽게 학생들의 상벌점을 관리할수 있고  더 쉽고 보기좋게 나타낼수 있습니다. | |

# 2. 요구사항 분석

|  |  |
| --- | --- |
| 요구사항 분석 | |
| **프로젝트 명** | 생활관 상벌점 관리 |
| < 저장할 정보들 >   * 학생은 유일한정보인 학번(PK)과 이름, 연락처, 소속학과, 호실 그리고 상벌점을 가진다. * 상벌점내역은 유일한정보인 번호(PK)와 날짜, 점수, 내용 그리고 학번(FK)을 가진다. * 관리자는 유일한정보인 관리자번호(PK)와 이름 그리고 연락처를 가진다. * 마름모는 점수부여이며 트리거를 이용해 details에 데이터를 저장하면 student의 데이터가 자동으로 업데이트되어 저장된다.   < 제공할 기능들 >   * 학생들의 정보를 제공 * 추가된 상벌점내역들의 정보를 제공 * 관리자들의 정보를 제공 * 학번에 따른 학생들의 정보를 제공 * 관리자의 정보를 제공 * 총학생수의 정보를 제공 * 총호실수의 정보를 제공 * 상벌점점수가 n점 이상인 학생의 정보를 제공 * 학생들의 상벌점합의 정보를 제공 * 전체학생들의 상벌점내역의 정보를 제공 * 이름에 따른 학생의 상벌점내역의 정보를 제공 * 지정한 한 학생과 학과가 같은 학생의 정보를 제공 * 상점이 가장 높은 학생의 정보를 제공 * 벌점이 가장 높은 학생의 정보를 제공 * 지정한 한 학생보다 상점이 더 높은 학생들의 정보를 제공 * 지정한 한 학생보다 벌점이 더 높은 학생들의 정보를 제공 * 상벌점내역을 추가하면 자동으로 업데이트되는 기능을 제공 | |

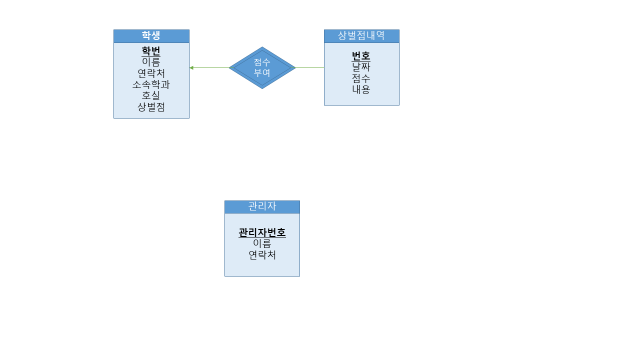
# 3. 개념적 데이터 모델

|  |  |
| --- | --- |
| 개념적 데이터 모델(ERD) | |
| **프로젝트 명** | 생활관 상벌점 관리 |



# 4. 논리적 데이터 모델

|  |  |
| --- | --- |
| 논리적 데이터 모델 | |
| **프로젝트 명** | 생활관 상벌점 관리 |



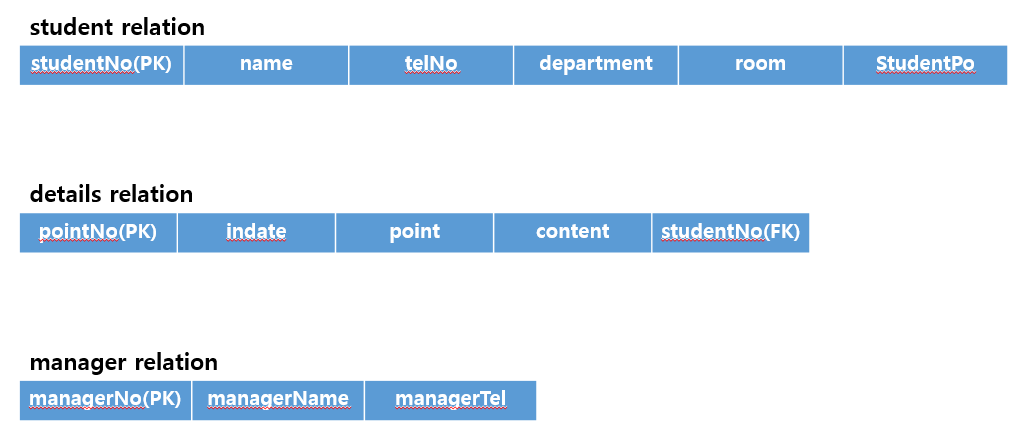


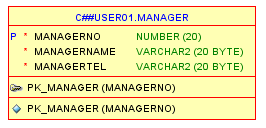
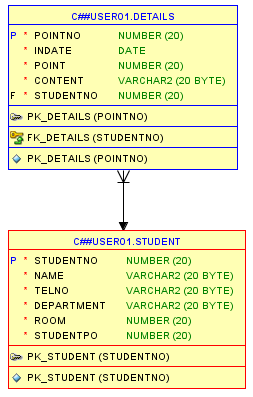
# 5. 용어 사전 정의

|  |  |  |  |
| --- | --- | --- | --- |
| 용어 사전 | | | |
| **프로젝트 명** | 생활관 상벌점 관리 | | |
| **논리명** | **물리명** | **약어** | **설 명** |
| **학생** | student |  |  |
| 학번 | studentNo |  | 학번 |
| 이름 | name |  | 학생이름 |
| 연락처 | telNo |  | 학생연락처 |
| 소속학과 | department |  | 소속학과 |
| 호실 | room |  | 호실 |
| 상벌점 | studentPo |  | 상벌점점수 |
| **상벌점내역** | details |  |  |
| 번호 | pointNo |  | 상벌점번호 |
| 날짜 | indate |  | 날짜 |
| 점수 | point |  | 상벌점부여점수 |
| 내용 | content |  | 상벌점내역 |
| 학번 | studentPo |  | 학번 |
| **관리자** | manager |  |  |
| 관리자번호 | managerNo |  | 관리자번호 |
| 이름 | managerName |  | 관리자이름 |
| 연락처 | managerTel |  | 관리자연락처 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 6. 물리적 데이터 모델

|  |  |
| --- | --- |
| 물리적 데이터 모델 | |
| **프로젝트 명** | 생활관 상벌점 관리 |





# 7. Table 기술서

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | student | | | | |
| **테이블 설명** | | 학생 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | studentNo | number(20) | Y | PK |  | 학번 |
| 2 | name | varchar2(20) | Y |  |  | 학생이름 |
| 3 | telNo | varchar2(20) | Y |  |  | 학생연락처 |
| 4 | department | varchar2(20) | Y |  |  | 소속학과 |
| 5 | room | number(20) | Y |  |  | 호실 |
| 6 | studentPo | number(20) | Y |  |  | 상벌점점수 |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | details | | | | |
| **테이블 설명** | | 상벌점내역 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | pointNo | number(20) | Y | PK |  | 상벌점번호 |
| 2 | indate | date | Y |  |  | 날짜 |
| 3 | point | Number(20) | Y |  |  | 상벌점부여점수 |
| 4 | content | varchar2(20) | Y |  |  | 상벌점내역 |
| 5 | studentNo | number(20) | Y | FK |  | 학번 |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 기술서 | | | | | | |
| **테이블 명** | | manager | | | | |
| **테이블 설명** | | 관리자 | | | | |
| **No** | **Attribute** | **Data Type** | **NN** | **Ky** | **Default** | **Description** |
| 1 | managerNo | number(20) | Y | PK |  | 관리자번호 |
| 2 | managerName | varchar2(20) | Y |  |  | 관리자이름 |
| 3 | managerTel | varchar2(20) | Y |  |  | 관리자연락처 |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |
| 12 |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |
| **비 고** | | | | | | |
|  | | | | | | |

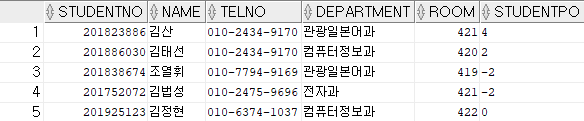
# 8. SQL문 사용하기

## **1) 기본 SQL(select, where, 그룹함수, group by, having, order by) - 기본 5개**

1. student테이블을 나타내주는 SQL문

select \*

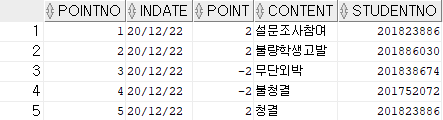
from student;



2. details테이블을 나타내주는 SQL문

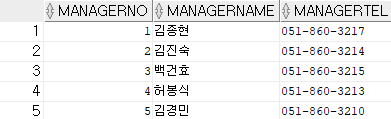
select \*

from details;



3. manager테이블을 나타내주는 SQL문

select \*

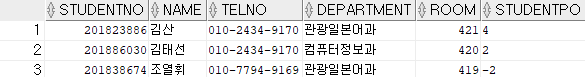
from manager;  


4. student테이블의 학번에 2018이 포함되는 학생들을 나타내주는 SQL문

select \*

from student

where substr(studentNo, 1, 4) = '2018';



5. manager테이블의 관리자이름이 김경민인 관리자를 나타내주는 SQL문

select \*

from manager

where managerName = '김경민';



6. student테이블의 총학생수를 나타내주는 SQL문

select count(\*) as "학생의 수"

from student;



7. student테이블의 총호실수를 나타내주는 SQL문

select count(room) as "호실의 수"

from student;



8. student테이블의 상벌점점수가 2점 이상인 학생의 학번과 그 학생의 상벌점점수를 나타내주는 SQL문

select studentNo, max(studentPo)

from student

group by studentNo

having max(studentPo) >= 2;



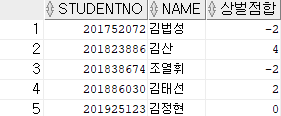
9. student테이블의 학생의 학번, 이름 그리고 상벌점합을 나타내주는 SQL문

select studentNo, name, sum(studentPo) as 상벌점합

from student

group by studentNo, name

order by studentNo, name;



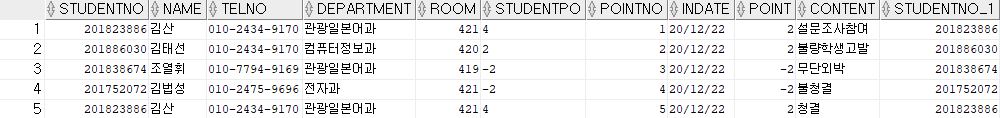
## **2) JOIN – 기본 2개**

1. 전체학생들의 상벌점내역정보를 나타내주는 SQL문

select \*

from student, details

where student.studentNo = details.studentNo;



2. 이름이 김산인 학생의 상벌점내역정보를 나타내주는 SQL문

select \*

from student, details

where student.studentNo = details.studentNo

and name = '김산';



## **3) Sub Query – 기본 3개**

1. student테이블에서 이름이 김산인 학생과 학과가 같은 학생을 나타내주는 SQL문

select studentNo, name, department

from student

where department = (select department

from student

where name = '김산');



2. student테이블에서 이름이 김태선인 학생과 학과가 같은 학생을 나타내주는 SQL문

select studentNo, name, department

from student

where department = (select department

from student

where name = '김태선');



3. student테이블에서 상점이 가장 높은 학생을 나타내주는 SQL문

select studentNo, name, studentPo

from student

where studentPo = ( select max(studentPo)

from student );



4. student테이블에서 벌점이 가장 높은 학생을 나타내주는 SQL문

select studentNo, name, studentPo

from student

where studentPo = ( select min(studentPo)

from student );



5. student테이블에서 이름이 김정현인 학생보다 상점이 더 높은 학생들을 나타내주는 SQL문

select studentno, name, studentPo

from student

where studentPo > ( select studentPo

from student

where name = '김정현');



6. student테이블에서 이름이 김정현인 학생보다 벌점이 더 높은 학생들을 나타내주는 SQL문

select studentno, name, studentPo

from student

where studentPo < ( select studentPo

from student

where name = '김정현');



## **4) Trigger**

create or replace trigger trg\_point

after insert

on details

for each row

begin

update student

set student.studentPo = (student.studentPo + :new.point)

where student.studentNo = :new.studentNo;

end;

/

insert into details

values(1, sysdate, 2, '설문조사참여', 201823886);

insert into details

values(2, sysdate, 2, '불량학생고발', 201886030);

insert into details

values(3, sysdate, -2, '무단외박', 201838674);

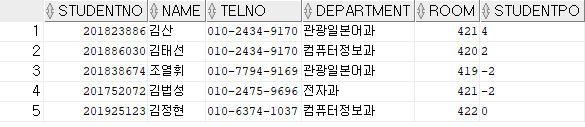
insert into details

values(4, sysdate, -2, '불청결', 201752072);

insert into details

values(5, sysdate, 2, '청결', 201823886);

select \* from student;



|  |
| --- |
| 별 첨 |

**별첨 1. 프로젝트 후기**

**별첨 2. Database 구축 dump 자료**

**1. 프로젝트 후기**

생애 처음 프로젝트를 진행해보았다.

한학기동안 배운 것을 되새기며 자신이 정한 주제를 오라클로 구현하는 프로젝트였다.

프로젝트를 어떻게 진행해야할지 모르고 주제도 고민되고 도저히 감이 잡히지않았지만 교수님께서

각각 개인 1:1로 지도해주셔서 더 집중되었고 프로젝트진행도 수월했던것같다.

특히 정규수업이 다 끝났는데도 zoom으로 모르는 것을 알려주실 때 정말 감동했다.

그때 Trigger를 처음 접하고 배웠는데 Trigger는 정말 유용하고 필수적인 것 같다.

남들보다 난이도가 높지않은 프로젝트를 진행했지만 너무 재미있었고 노력했고 열심히했으며 오타나 틀린부분이 있는지 없는지 몇번이고 되풀이 하며 고치고 고민했다. 그만큼 교수님께 완성도있는 보고서로 보답하고 싶었다. 스스로는 만족하지만 좀더 열심히했다면 더 알고있었다면 더 괜찮은 프로젝트를 진행했을거라는 아쉬움은 남는다. 한학기가 정말 짧고 아쉽다. 왜냐하면 데이터베이스는 내가 가장 좋아하는 교과목이었기떄문이다. 이번학기가 끝나 군문제로 휴학하게되지만 데이터베이스만은 스스로 계속 공부할것같다.

**2. Database 구축 dump 자료**

**< 계정 생성하기 >**

conn system/sys;  
  
create user c##user01 identified by user01  
default tablespace users  
temporary tablespace temp  
quota unlimited on users;  
  
grant connect to c##user01;  
grant resource to c##user01;  
  
conn c##user01/user01;  
  
show user;

**< 테이블 생성하기 >**

create table student(

studentNo number(20) not null,

name varchar2(20) not null,

telNo varchar2(20) not null,

department varchar2(20) not null,

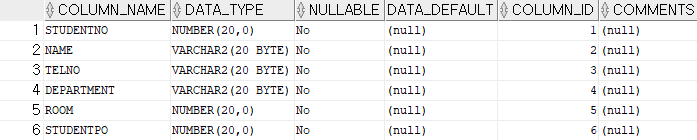
room number(20) not null,

studentPo number(20) not null

);

alter table student

add constraint pk\_student primary key(studentNo);



create table details(

pointNo number(20) not null,

indate date not null,

point number(20) not null,

content varchar2(20) not null,

studentNo number(20) not null

);

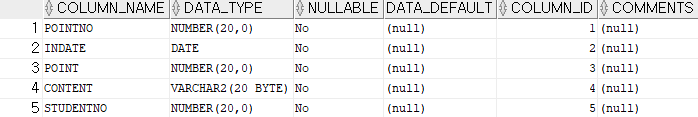
alter table details

add constraint pk\_details primary key(pointNo);

alter table details

add constraint fk\_details foreign key(studentNo)

references student(studentNo);



create table manager(

managerNo number(20) not null,

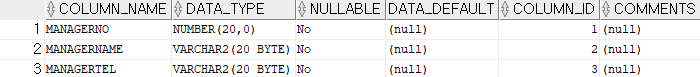
managerName varchar2(20) not null,

managerTel varchar(20) not null

);

alter table manager

add constraint pk\_manager primary key(managerNo);



**< 테이블 데이터 추가하기 >**

insert into student

values('201823886', '김산', '010-2434-9170', '관광일본어과', '421', '0');

insert into student

values('201886030', '김태선', '010-2434-9170', '컴퓨터정보과', '420', '0');

insert into student

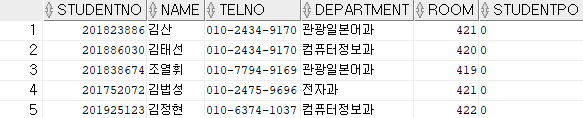
values('201838674', '조열휘', '010-7794-9169', '관광일본어과', '419', '0');

insert into student

values('201752072', '김법성', '010-2475-9696', '전자과', '421', '0');

insert into student

values('201925123', '김정현', '010-6374-1037', '컴퓨터정보과', '422', '0');



insert into details

values(1, sysdate, 2, '설문조사참여', 201823886);

insert into details

values(2, sysdate, 2, '불량학생고발', 201886030);

insert into details

values(3, sysdate, -2, '무단외박', 201838674);

insert into details

values(4, sysdate, -2, '불청결', 201752072);

insert into details

values(5, sysdate, 2, '청결', 201823886);



insert into manager

values('1', '김종현', '051-860-3217');

insert into manager

values('2', '김진숙', '051-860-3214');

insert into manager

values('3', '백건효', '051-860-3215');

insert into manager

values('4', '허봉식', '051-860-3213');

insert into manager

values('5', '김경민', '051-860-3210');

