**2022 데이터베이스 실습 레포트**

**데스파**

B02

2104637 박희선

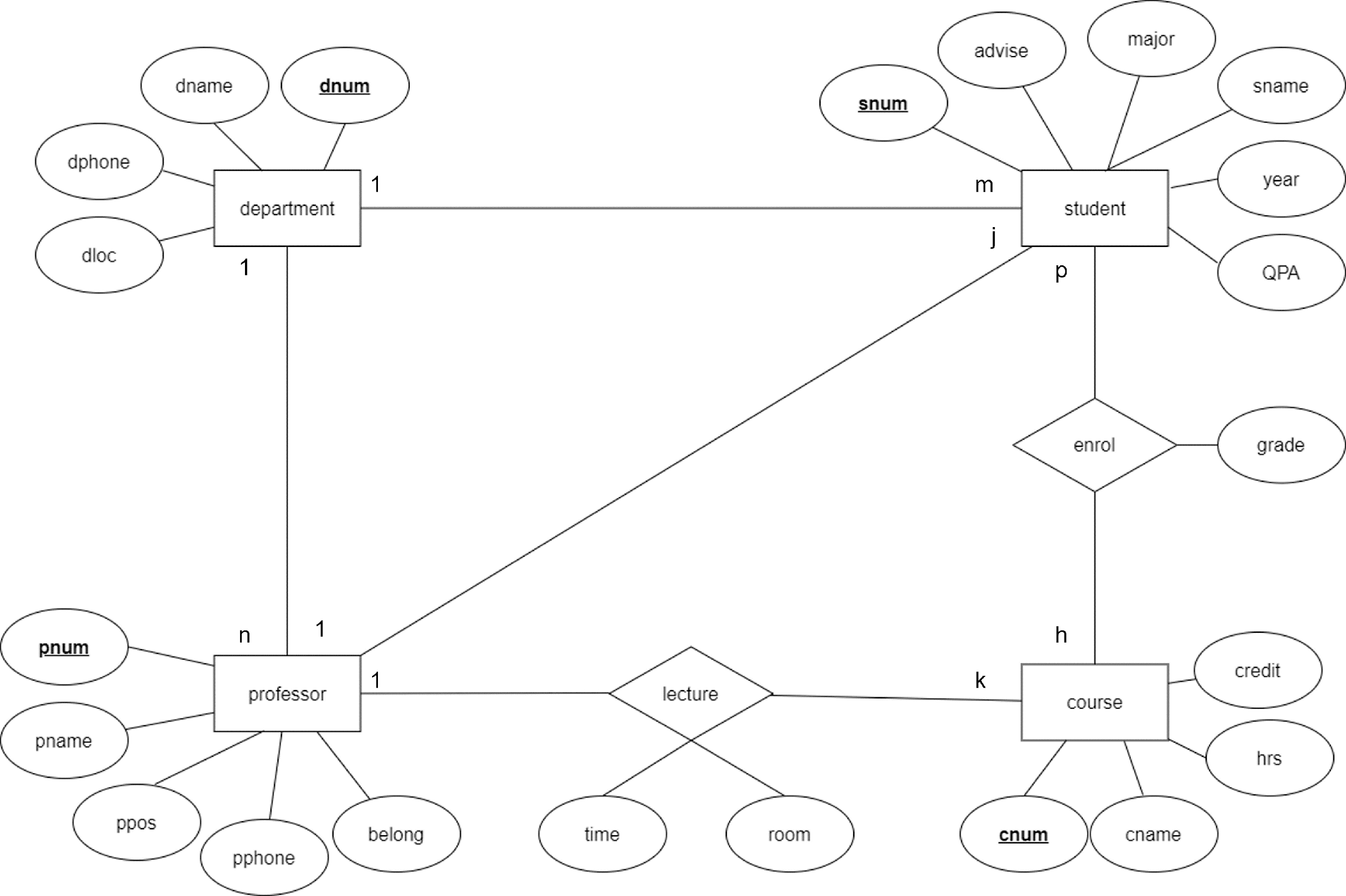
2104649 양지연

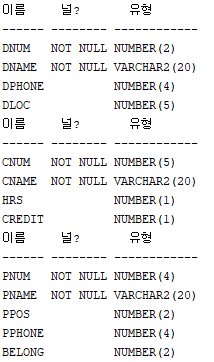
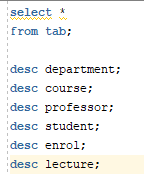
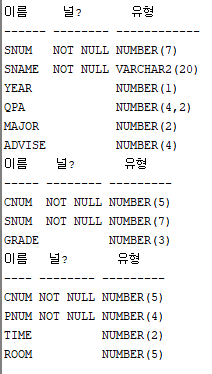
2104689 전수진

2104714 황다은

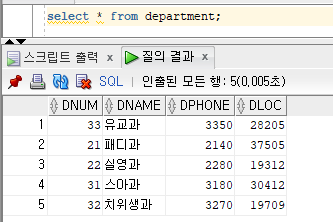
**목차**

1. 한양여대 schema
2. DB 내용
3. 교수님 쿼리
4. 데스파 쿼리
5. 실습 일지
6. 실습 후기
7. 한양여대 schema

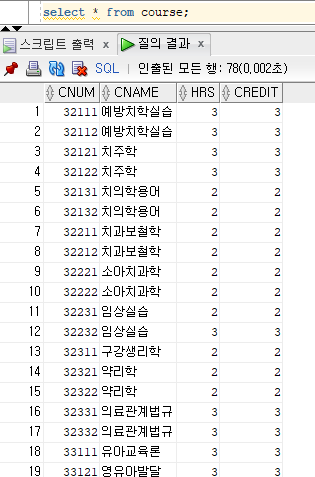


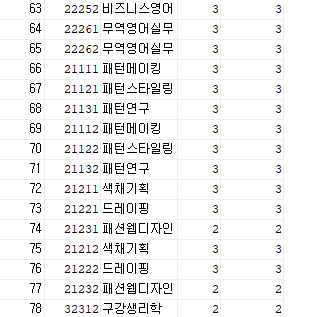
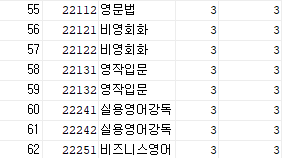
 

1. DB 내용
2. department 테이블 (학과)

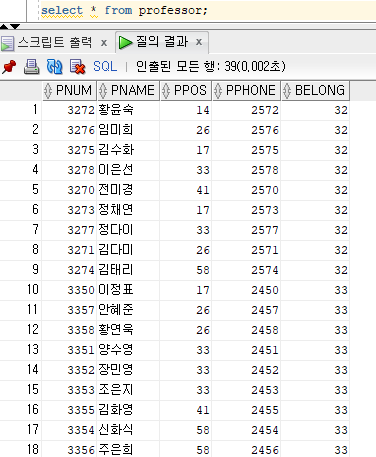


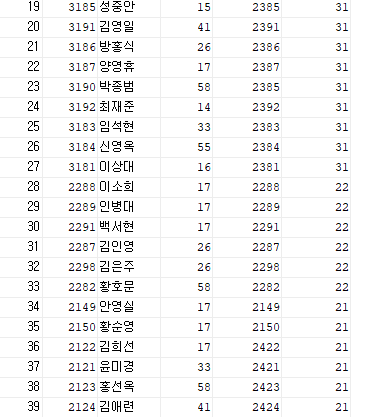
1. course 테이블 (강의) - 3 x 6 x 3 + 2 x 6 x 2 = 78개

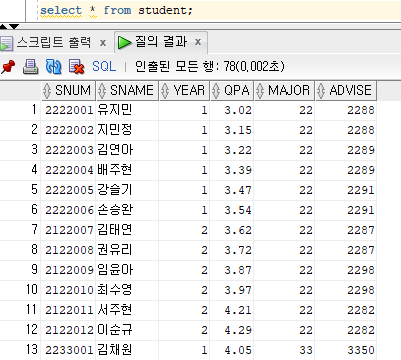


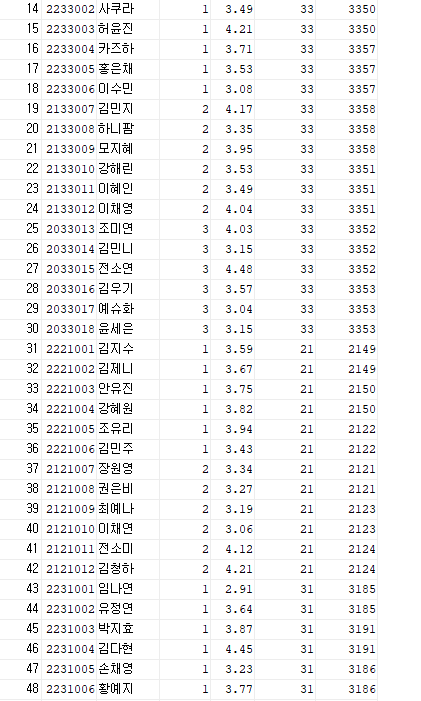


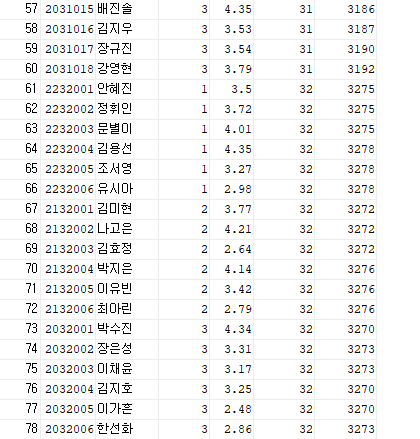
1. professor 테이블 (교수) - 3 x 9 + 2 x 6 = 39명



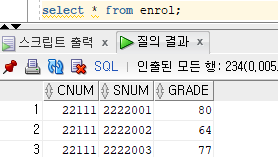


1. student 테이블 (학생) – 78명





1. enroll – 234개

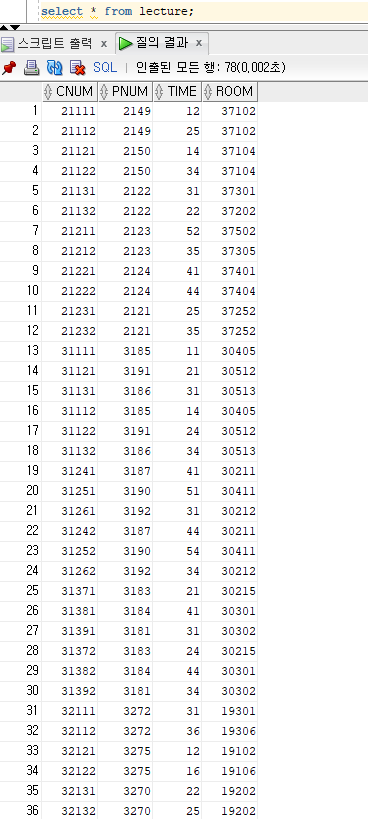


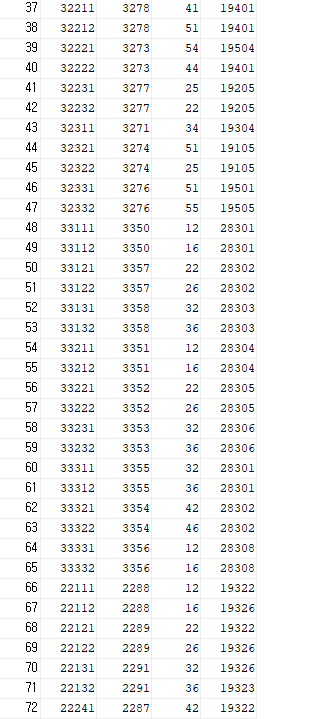
.

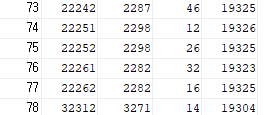
.

.



1. lecture – 78개





1. 교수님 쿼리

1) 각 학과의 이름과 전공 학생 수를 보이시오 (학과명순)

SELECT DISTINCT dname as 학과명, COUNT(student.sname) as 학생수

FROM department, Student

WHERE department.dnum = student.major

GROUP BY department.dname

ORDER BY department.dname;



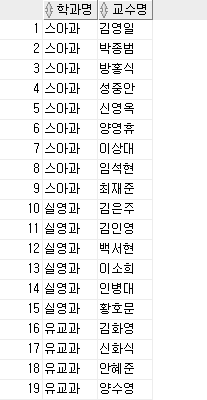
2) 각 학과 명과 소속된 교수 이름을 보이시오 (학과명순)

SELECT DISTINCT dname as 학과명, pname as 교수명

FROM Department, Professor

WHERE department.dnum = professor.belong

ORDER BY department.dname;



3) 각 교수의 이름과 지도 학생 수를 보이시오

select pname as 교수, count(snum) as 지도학생

from professor, student

where professor.pnum = student.advise

group by professor.pname;





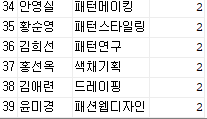
4) 각 교수 명과 담당 과목명, 그리고 과목 수를 보이시오

select professor.pname as 교수명, course.cname as 과목명, count(lecture.cnum) as 과목수

from professor, lecture, course

where professor.pnum = lecture.pnum and course.cnum = lecture.cnum

group by professor.pname, cname;



5) 각 학과 명과 지도 반을 맡지 않은 교수 명을 보이시오

select DISTINCT pname as 교수명, dname as 학과명

from professor, department

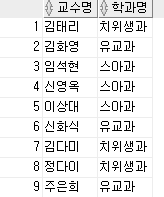
where professor.belong = department.dnum

MINUS

SELECT DISTINCT pname, dname

FROM department, professor, student

WHERE professor.pnum IN (SELECT student.advise FROM Student);



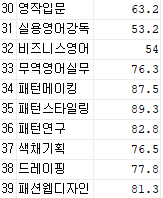
6) 각 과목명과 평균 점수를 보이시오

select distinct cname as 과목명, round(avg(grade),1) as 평균점수

from course, enrol

where course.cnum = enrol.cnum

group by cname;

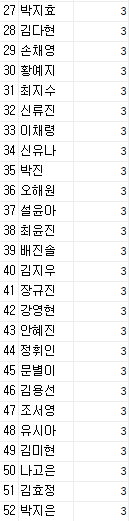
7) 각 학생의 이름과 수강 과목 수를 보이시오

select student.sname as 학생명, count(cnum) as 과목수

from student, enrol

where student.snum = enrol.snum

group by student.sname;

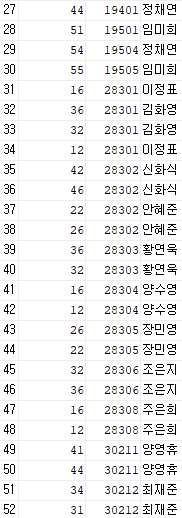
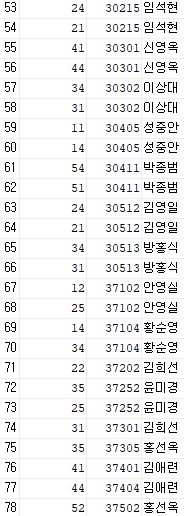
8) 강의실 번호와 시간, 강의 담당 교수 명을 보이시오 (강의실 번호 순)

select time as 강의시간, room as 강의실, pname as 담당교수

from professor, lecture

where professor.pnum = lecture.pnum

order by lecture.room asc;

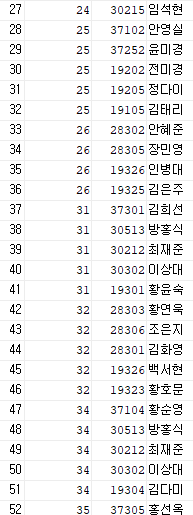
9) 강의 시간과 강의실 번호, 강의 담당 교수 명을 보이시오 (강의시간 순)

select time as 강의시간, room as 강의실, pname as 담당교수

from professor, lecture

where professor.pnum = lecture.pnum

order by lecture.time asc;

1. 데스파 쿼리

1) 장학금 받을 수 있는 학생명과 학과명, 성적

SELECT dname AS 학과명, sname AS 학생명, QPA AS 성적

FROM Student, Department

WHERE Student.major = Department.dnum AND Student.QPA >= 4.25

ORDER BY dname, QPA DESC;



2) 패디과 성적이 평균보다 낮은 학생의 상담을 잡기 위한, 해당 학생의 시간표를 보이시오

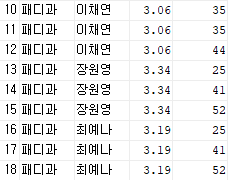
select dname as 학과명, sname as 학생명, qpa as 성적, time as 시간표

from student, department, lecture, enrol

where student.major = department.dnum and lecture.cnum = enrol.cnum and enrol.snum = student.snum

and dname = '패디과' and qpa < (select avg(qpa) from student)

order by sname, time;

3) 실영과 1학년 평균 점수보다 낮은 점수를 받은 학생을 보이시오.

select dname as 학과명, year as 학년, sname as 학생명, qpa as 학점

from student, department

where student.major = department.dnum and dname = '실영과' and

qpa < (select round(avg(qpa),2)

from student, department

where student.major = department.dnum

group by dname, year

having year = 1 and dname ='실영과');



4) 모든 과목 중 평균 점수가 가장 낮은 과목과 이를 담당하는 교수명을 보이시오

create view average

as select cname AS 과목명, round(avg(grade), 2) AS 평균, pname As 교수명, belong

from course, enrol, student, professor, lecture

where course.cnum = enrol.cnum AND enrol.snum = student.snum AND course.cnum = lecture.cnum AND lecture.pnum = professor.pnum

group by cname, pname, belong;

select dname as 학과명, 과목명, 교수명, 평균 as 평균점수

from department, average

where department.dnum = average.belong AND 평균 in (select min(평균) from average);



5) 치위생과 학생의 성적들 중 제일 높은 학생의 지도 교수가 누구인지 보이시오.

CREATE VIEW sqpa

AS SELECT major as 학과, max(qpa)as 점수

FROM student

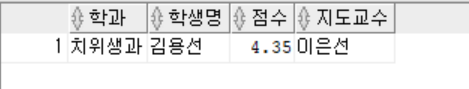
GROUP BY major;

select dname as 학과, sname as 학생명, 점수, pname as 지도교수

from department, professor, sqpa, student

where department.dnum = sqpa.학과 and student.advise = professor.pnum and sqpa.점수 = student.qpa and dname = '치위생과'

group by dname, sname, 점수, pname;



6) 패디과에서 수요일에 새로운 강의를 하나 넣을 예정이다. 이때 패디과 교수들 중 수요일에 수업이 없는 교수를 보이시오

select dname as 학과명, pname as 교수명

from department, professor, lecture

where department.dnum = professor.belong and professor.pnum = lecture.pnum and dname = '패디과'

MINUS

select dname, pname

from department, professor, lecture

where department.dnum = professor.belong and professor.pnum = lecture.pnum and dname = '패디과' and substr(lecture.time, 1, 1) = 3;



1. 스마트it학과의 3학점 강의를 하는 교수의 정보를 보이시오.

select distinct cname as 과목명, credit as 학점, pname as 교수명

from course, lecture, professor, department

where course.cnum = lecture.cnum and lecture.pnum = professor.pnum and professor.belong = department.dnum and dname = '스아과' and credit = '3';



8) 실영과 이소희교수의 강의 중 70점 이상을 맞은 학생을 구하시오.

create view pposc2

as select dname, pname, lecture.cnum

from department, professor, lecture

where department.dnum = professor.belong and lecture.pnum = professor.pnum and dname = '실영과';

select dname as 학과명, pname as 교수명, cname as 과목명, sname as 학생명, grade as 점수

from pposc2, student, enrol, course

where pposc2.cnum = enrol.cnum and enrol.snum = student.snum and course.cnum = enrol.cnum and pname = '이소희' and grade >=70;



9) 학년별로 성적이 가장 높은 학생명을 보이시오.

create view yearqpa

as select year as 학년, max(qpa) as 점수

from student

group by year;

select 학년, dname as 학과명, sname as 학생명, 점수

from yearqpa, student, department

where yearqpa.점수 = student.qpa and student.major = department.dnum;



10) 스아과를 제외한 학과별 평균 점수중 3.5 이하인 학과명과 점수를 보이시오

select dname as 학과명, round(avg(qpa), 2) as 평균점수

from department, student

where department.dnum = student.major and dname != '스아과'

group by dname

having 3.5 >= round(avg(qpa), 2);



1. 실습 일지

- 11주차(11/9)

마당 서점 테이블 삭제 후 한양여자대학교 테이블 입력

각 테이블에 들어갈 insert문 분량을 나누어 작성

3년제 학과

1) 스마트it과 (공학) - 지연 (31)

2) 치위생과 (자연과학) - 다은 (32)

3) 유아교육과 (인문사회) - 희선 (33)

2년제 학과

1) 패션디자인과 (예체능) - 희선 (21)

2) 실무영어과 (인문사회) - 수진 (22)

각 테이블에 들어갈 튜플에 대한 규칙과 설명을 정함

1) department 테이블 (학과)

dnum(2): 학과 번호

dname(20): 학과 이름

dphone(4): 학과 번호 2자리 + 과사 번호2자리

dloc(5): 과건물 번호 2자리 + 강의실 번호 3자리

- 학과 이름

패디과: 패션디자인과 - 2년제

실영과: 실무영어과 - 2년제

스아과: 스마트it과 - 3년제

유교과: 유아교육과 - 3년제

치위생과 - 3년제

- 건물 번호 정한 숫자

본관 (19), 도서관 (34), 정보문화관 (30), 디자인관 (37), 교수회관 (40), 유아교육관 (28),

식품영양관 (69), 행원스퀘어 (10), 행원파크 (52)

2) course 테이블 (강의)

cnum(5): 학과 번호 2자리 + 수강 학년 번호 1자리 + 과목 번호 1 자리 + 반 번호 1자리

cname(20): 과목명

hrs(1): 몇 시간짜리 수업인지

credit(1): 학점

- 각 학과에 각 학년마다 세과목(과목명이 세개) 개설

- 각 학년마다 두반

3) professor 테이블 (교수)

pnum(4): 학과 번호 2 자리 + 교수의 연구실 전화번호 뒤 2자리

pname(20): 교수 이름

ppos(2): 교수 직급 1자리 + 교수 호봉 1자리

phone(4): 교수 연구실 전화번호

belong(2): department의 dnum값 2자리

- 교수 직급당 호봉

정교수-(1-7), 부교수-(2-6), 조교수-(3-3), 전임강사-(4-1), 명예교수-(5-8)

- 교수 1명당 2과목 강의

- 3년제 학과: 교수 9명, 지도교수 6명

4) student 테이블 (학생)

snum(7): 입학 년도 2자리 + 학과 번호 2자리 + 001부터 차례대로 3자리

sname(20): 학생 이름

year(1): 학년

QPA(4,2): 학점 (ex: 3.40)

major(2): department의 dnum값 2자리 - 소속 학과

advise(4): proffesor의 pnum값 4자리 - 지도 교수

insert 문을 메모장에 작성하여

구글 드라이브에 올려 팀원들과 공유할 수 있도록 함

- 12주차(11/16)

department, course, professor, student 테이블 오라클에 작성

오류 수정하고, 개수가 맞게 들어갔는지 확인

오타 수정하고, 콤마나 세미클론이 빠진 부분 채워 넣음

course 테이블에서 valuse 값이 20이 넘어간 부분들을 수정함

예방치학실습: 예방치학 및 실습

파이썬: 파이썬 프로그래밍

자바스크립트: 자바스크립트 서버플랫폼

안드로이드: 안드로이드 프로그래밍

시분설: 시스템분석설계

영작입문: 영문서작성입문

비영회화: 비즈니스영어회화

student 테이블에서 오류가 발생하여 기존 튜플을 삭제

<튜플 삭제>

delete from student

where sname = ‘학생이름’;

모든 오류를 다 고친 후, 오라클에 입력

Enorl, Lecture 테이블 작성

팀원들끼리 맡은 학과의 enorl, lecture을 작성

이를 구글 드라이브에 공유하여 발생하는 오류를 잡고, 오타자를찾음

부모 속성 키가 없다는 오류가 발생하여 값을 잘못 입력한 부분을

찾아 부모 속성 키와 동일한 값을 만들어줌

오라클에 모든 값을 입력한 뒤 9개의 쿼리를 확인

- 13주차(11/23)

교수님의 테스트 쿼리 9개를 풀음

- 14주차(11/30)

뷰에 대해 배움

팀에서 10개의 쿼리를 작성함

- 15주차(12/7)

조별 10개 쿼리 발표

1. 실습 후기
2. 2104637 박희선

2학년 1학기 때 배웠던 이론들을 토대로 마당서점 데이터베이스를 실습했을 때는 처음부터 DB를 구축한 것이 아니라 이미 구축된 DB에서 원하는 결과값을 나타내기 위한 질의 작성법을 배운 것이라 완벽히 이해했다고 볼 수는 없었습니다. 하지만 이번 DB 실습 때 팀플레이를 하면서 저희가 직접 값을 설정하고 입력하고 테이블을 만들어서 잘못 입력한 튜플은 없는지 확인하고 수정하는 여러 과정들을 통해 DB에 대하여 완벽하게는 아니더라도 DB가 무엇인지에 대해서는 알게 됐습니다. 제일 먼저 오라클에 테이블을 만들기 전 만들어야 하는 ER 다이어그램을 만들 때도 객체와 객체의 관계 설정하는 부분에 대해서도 여러 방면으로 생각해야 한다는 것을 알게됐고, 그 관계를 어떻게 받아들이느냐에 따라 결과가 달라진다는것도 알게 됐습니다. 이후에 오라클에 테이블을 만들 때도 이러한 관계에 따라 먼저 만들어야 하는 테이블이 있고 나중에 만들어야 하는 테이블이 있다는 점을 통해 DB는 서로 연결되어 있다는 것을 몸소 느끼게 되었고, 이번 실습 때보다 더 많은 정보를 입력하게 됐다면 훨씬 더 복잡할 거란 생각을 했습니다. 테이블을 만든 후에는 팀원들과 정보를 어떻게 입력할 것인지에 대해 상의를 하고, 각자 맡은 분량의 정보를 구한 뒤 값을 입력했습니다. 이 부분을 실습할때는 정말 DBA가 된 느낌이 들어 흥미가 생기기도 했으며, 조금 헷갈리는 부분이 있거나 모르는 부분이 있을 때 팀원들과 소통하며 알아가는 부분에 대해서도 혼자보다는 여러명이서 함께 하는 것이 더 도움이 된다는 것을 알게 됐습니다. 그래서 완벽하게 입력했다고 믿은 DB에서 오류가 나타났을 때는 당황스럽기도 하였지만, 이러한 오류가 났기 때문에 INSERT 문을 수정하는 방법에 대해서도 알게 됐고 삭제하는 방법에 대해서도 알게 됐습니다. 그러면서 뼈저리게 느끼게 된 것은 DB를 구축할 때는 애매모호하게 구축하면 안 된다는 점이었습니다. 처음부터 완전히 잘 설계된 것을 토대로 만들어야 많은 정보를 입력했을 때 오류도 나지 않고 수정할 것이 적어지는데, 애매모호하게 구축을 하게 된다면 수정하는 것만으로 오류를 고칠 수 없게 되기 때문입니다. 그래서 DB는 서로 연관되어 있고, 생각보다 훨씬 복잡하고, 그러기 때문에 처음부터 제대로 된 계획과 설정값이 있어야 한다는 걸 실습할 때마다 느끼게 됐습니다. 처음에는 이걸 우리가 어떻게 하지? 할 수 있을까? 반신반의했던 것들을 처음부터 차근차근 만들어 가고 실행하며 직접 확인하니 더 머리에 깊게 남게 됐고, 내년에 졸작을 하게 됐을 때 백지 상태에서의 시작이 아닌 DB에 어느 정도 아는 상태에서 시작하게 될 것 같아 저에겐 매우 뜻 깊었던 실습 시간이었습니다.

1. 2104649 양지연

1학기 때 이론으로 배울 때에는 유익한 학문이고 배우기 쉬워 보였습니다. 그러나 실제로 SQL 프로그램을 돌리면서 실습을 해보니 생각과는 전혀 다르게 흘러간다는 걸 알게 되었습니다. 교수님이 쉽게 쿼리를 만들어 보여주는 것이 그게 쉬워서가 아니라 그 분야의 교수님일 만큼 잘해서이기 때문이란 걸 이번 학기 내내 온몸으로 느낄 수이었습니다. 의미가 있는 쿼리를 만들어내는 것이 제일 어려웠던 것 같습니다. 쿼리 자체는 그동안 배웠던 것을 총동원해 쏟아부 우면 만들어낼 수 있습니다. 그러나 의미 그 자체를 생각해서 쿼리를 작성해 내는 것은 본인이 직접 문제를 만들어내서 푸는 것이니, 무의식적으로 본인이 풀기 쉬운 방향으로 기울어지는 경향도 있어서 어려웠습니다.ER-다이어그램을 작성하는 것도 보기와는 다르게 작성하는 것도 어려웠습니다. 이론도 어렵게 느껴졌고 흡수라는 개념도 다른 곳에서 찾기 어려운 이론이라 ER-다이어그램을 그리는 내내 “이게 맞을까?” 하는 의심이 떠나지 않았습니다. ER-다이어그램을 그리면서 다이어그램 넷이라는 구글에서 제공하는 다이어그램을 쉽게 그릴 수 있는 웹을 찾게 되어서 다른 과목의 과제를 수행할 때에도 유용하게 사용할 수 있어서 편리한 발견이었습니다.데이터베이스에 들어갈 인서트문을 작성할 때, 메모장으로 작성하였습니다. 지금에서야 생각해보니 엑셀을 썼으면 쉬웠을 것이란 생각이 듭니다. 메모장으로 작성하니 비슷한 문장들의 연속이니 오타나 잘 못 입력한 값을 찾기가 쉽지 않았기 때문입니다. 입력할 값들을 규칙성을 세워서 작성해서 쿼리를 작성하는 것이 어렵지 않았습니다. 교수님이 내주신 9개의 쿼리를 작성할 때부터 개인 노트북에 sql developer을 설치하려고 하였습니다. 그러나 이상하게 다운은 되었는데, 워크시트가 생성이 되지 않았습니다. 다른 팀원들도 설치가 어려웠는데 이런 방법을 썼다며 알려준 블로그 글을 보고 따라 해보았지만 이상하게 되는 게 하나도 없었습니다. sql plus를 이용하려 해도 그것이 없다고 나왔습니다. 강의실에 설치된 것을 가져오거나 다른 팀원들 것을 usb에 옮겨 가져와 설치하였지만 결과가 똑같았고, 오라클에서 다시 다운로드하려 하니 롤백 현상까지 일어났습니다. 하는 수없이 저는 쿼리를 작성한 후 다른 팀원들에게 실행해달라고 하거나 강의실에 갈 때마다 최대한 하는 방식으로 만들어야 했습니다. 팀으로 진행하는 프로젝트여서 다행이라고 여겼습니다. 쿼리를 작성할 때에도 다른 팀원들에게 도움을 많이 받았습니다. 문제를 잘 이해할지 못할 때에도 먼저 이해한 팀원들이 설명해 주었고 조언도 받아서 팀원들에게 큰 고마움을 느꼈습니다. 얼마 전 온라인 졸업작품 전시회를 열었을 때, 선배님들의 졸업 작품에는 공통점이 있었습니다. 모두 데이터베이스가 있다는 것입니다. 당연한 부분이지만, 이 과목을 수강하길 정말 잘했단 것입니다. 내년 졸업작품을 만들 때, 이 과목에서 배운 기술과 팀 프로젝트를 진행하였던 경험을 바탕으로 더 잘 만들 수 있을 거 같습니다.

1. 2104689 전수진

이해가 한번에 잘 안 되는 편인데, 쿼리문을 실행할 데베를 직접 만들면서 이해가 잘 되고 뇌에도 강하게 각인된 것 같다. 테이블문을 짜고, 작성한 테이블문의 조건에 맞게 data dictionary를 구성하고, 그에 맞는 인서트문을 쓰면서 정보 하나하나가 유기적으로 연결되어 있으며 무엇 하나 쓸모 없이 겉돌고 있지 않고 전부 필요하다는 걸 느꼈다. 데이터베이스 문법에 맞게 원하는 정보를 맞는 위치에 기입해두면 다 되니 쉬운 것 같으면서도 그만큼 치밀하기 때문에 오류 없이 한번에 됐던 적은 거의 없는 것 같다. sql developer 프로그램에 대해 잘 몰라서 생기는 오류들도 있었다. 그래도 그때마다 피드백을 확인하고 고치고 시행착오를 겪으며 그래도 모르겠을 때는 팀원들의 도움을 받으며 한양여대 데이터베이스를 완성할 수 있었다. 각자 파트를 나눠 작성한 테이블문이나 인서트문을 한데 모아 정리해 구글드라이브에 업데이트한 걸 보면 깔끔함에 뿌듯함이 느껴지기도 했다. 그 다음으로 쿼리를 만들었는데, 어려웠지만 팀원들이 여러모로 도와줬고 다들 끝까지 포기하지 않았다. 큰 고마움을 느꼈다.팀플할때 원활한 의사소통과 참여하려는 의지가 중요하다는 건 알고 있었지만 그 이상으로 서로 이끌고 격려하며 북돋는 자세도 매우 중요하다는 걸 느꼈다. 이것 덕분에 끝까지 잘 진행될 수 있었다고 생각하기 때문이다. 다음에는 팀원으로써의 역량도 키우고 이런 마음가짐을 가져야겠다고 다짐했다. 쿼리를 만들면서 데이터베이스 자체를 계속 업데이트하기도 했는데 이때 쿼리를 실행하는 것은 데이터베이스가 잘 구축됐는지 확인하는 역할도 한다는 걸 알게 되었다. 그리고 빈틈 없는 데베를 잘 구축한다는 건 정말 중요하고 또 어려운 일이란 것도 다시금 깨닫게 됐다. 조별 차별화 쿼리를 만들 때에는 의미 있는 쿼리가 뭔지에 대해 많이 고민했다. 내가 생각하는 의미 있는 쿼리는 정말로 현실에서 쓰일 것 같은 현실성 있고 유용성이 있는 쿼리이다. 결국 그렇게 쓰기 위해 데이터베이스를 구축하는 것이기 때문이다. 그래서 학생 성적 열람 등 한양여대 데이터베이스 사용자 입장에서 질의를 고민하며 이 프로젝트를 하면서 다양한 데이터베이스 사용자의 입장들을 다 겪어보는 것 같단 생각을 했다. 또 나름 규모가 있는 프로젝트를 하고 있다고 느꼈는데, 막상 진행할 때에는 교수님이 매주 내준 진도까지 각자 시간을 조금씩 내서 하고 있는게 다라고 생각했다. 가능했던 이유는 기한이 세세히 정해져 있고, 그 순서가 올바르기 때문이라고 생각한다. 먼저 다이얼그램으로 만드려는 데베의 관계를 이해하고, 이를 바탕으로 테이블문과 인서트문을 짜고, 기본 쿼리와 심화 쿼리를 만든다. 지나온 과정이 자연스럽게 데이터베이스에 대한 이해를 도우면서 팀 프로젝트도 잘 되게 만든 것이다. 이번 팀플 수업은 나름 졸업작품의 예행연습 역할을 한다고 생각하는데, 데베에 관한 지식이나 일정 계획을 짜는 것, 팀플을 할 때의 태도 등에 대한 감각을 잘 가지고 졸작을 할 수 있을 것 같아 큰 도움이 될 것 같다.

1. 2104714 황다은

이론을 배울 때에는 관계대수, 릴레이션, 다이어그램 같은 것들이 어디에 쓰이는지 궁금했는데 실습을 시작하면서 더 열심히 공부하지 않은 것을 후회했다. sql developer를 설치할 때 자리에서 키보드가 고장 나, 가상키보드로 마우스로 하나씩 클릭하면서 겨우 설치했고, 다음 수업때는 컴퓨터가 켜지지 않아서 팀원이 하는 걸 보며 같이 실습했다. 집에서 오라클, SE 버전 등 두 시간 정도 검색해서 겨우 developer를 설치했다. 이론보다 실습이 훨씬 재밌었다. 처음 수업에서는 교수님이 말씀하시는 것을 따라 입력해 테이블을 만들고 인서트로 값을 넣으면 어떻게 결과를 가져오는지 잘 몰랐었는데 한양여대 DB를 만들기 시작하면서 왜 데이터베이스를 배워야 하는지 느꼈다. 우리가 회원가입을 할 때 쓰는 이름, 생년월일, 휴대전화번호, 아이디, 비밀번호 등 개인정보들이 입력되면 어디에 어떻게 저장되는지 궁금했는데 그에 대해 약간의 지식을 얻은 것 같아서 흥미로웠다. 인서트값을 직접 지정해주고 쿼리를 작성해 나오는 결과를 보면서 팀원들과 인서트문을 보면서 결과 테이블이 맞게 나오는지 확인했었는데, 틀릴 때마다 같이 하나씩 찾으면서 update를 사용해 값을 고칠 때, 질의에 맞게 쿼리를 작성해야 하는데 이해가 잘되지 않고 조인 연산을 어떻게 써야 하는지 어려울 때, 쿼리를 실행했는데 원하는 결과가 나오지 않아서 어떻게 수정해야 할지 막막했을 때 팀원들에게 정말 고마웠다. 처음에는 질의를 두 개씩 나눠서 해결하려 했는데 쿼리를 작성하는게 쉽지 않았다. 디스코드로 통화하고 화면을 공유하면서 같이 노력했는데 질의에 맞는 결과 테이블이 나올 때 정말 뿌듯했다. 3학년이 되면 졸업작품을 해야하고 팀을 만들어 프로젝트를 수행해야 하는데 먼저 겪어볼 수 있어서 좋았다. 다들 각자의 사정으로 시간을 맞추기 어려웠지만, 프로젝트를 위해 최우선으로 시간을 내고, 문제가 풀리지 않을 때는 직접 검색해서 찾아오거나 같이 고민을 해주고 잘 못 하는 팀원이 있으면 잘하는 팀원이 도와주기도 하고 아이디어도 내고 하면서 더욱 돈독해진 부분도 있는 것 같고 팀 프로젝트를 진행할 때 꼭 가져야 할 태도인 것 같다. 나는 팀원으로써 내가 아는 부분을 모르는 팀원에게는 최대한 이해할 수 있게끔 설명해주고, 시간이 되지 않거나 오류로 어려워하는 팀원이 있으면 같이 해결할 수 있도록 검색하고, 책을 찾아보면서 팀에 조금이라도 도움이 되기 위해 노력했다. 인서트값을 하나씩 비교해서 결과 테이블이 다르게 나와 튜플을 수정해야 했을 때 update문이라는 것을 알게 되었고 수정할 수 있다는 안도감과 동시에 이것을 찾은 팀원에게 고마웠다. 우리가 직접 만든 질의에서는 아무리 검색하고 찾아보고 같이 고민해봐도 오류가 나서 결국 교수님께 질문을 드렸었는데 친절하게 답변해주셔서 정말 감사했다. 팀 프로젝트를 진행할 때에는 각자 맡은 일에 책임감을 느끼고 해결하려는 의지가 중요하다고 느꼈다. 팀원들과 소통하면서 문제점이나 오류를 바로 수정할 수 있었고, 프로젝트를 위해 할애하는 시간이 많아 더 높은 퀄리티의 프로젝트를 진행할 수 있었다.