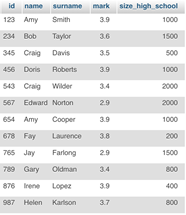
1.- Create the following database (with name ApplyingToCollege) with the following schema and data:

STUDENTS (id, name, surname, mark, size\_high\_school)

COLLEGES (name, state, enrollment)

APPLIES (sid\*, college\*, major, decision)

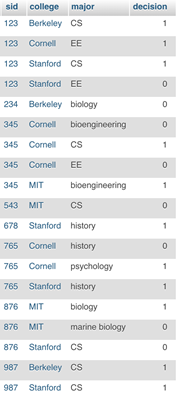
STUDENTS:



COLLEGES:



APPLIES:



create database ApplyingToCollege;

use ApplyingToCollege;

create table COLLEGES(

name VARCHAR(10) primary key,

state VARCHAR(2),

enrollment int) Engine = InnoDB;

create table STUDENTS(

id int primary key,

name VARCHAR(20),

surname VARCHAR(20),

mark real,

size\_high\_school int) Engine = InnoDB;

create table APPLIES(

sid int,

    college VARCHAR(10),

    major VARCHAR(20),

    decision boolean,

    primary key (sid, college, major),

    foreign key (sid) references STUDENTS(id),

    foreign key (college) references COLLEGES(name)) Engine = InnoDB;

insert into STUDENTS values (123, 'Amy', 'Smith',  3.9, 1000);

insert into STUDENTS values (234, 'Bob', 'Taylor', 3.6, 1500);

insert into STUDENTS values (345, 'Craig', 'Davis', 3.5, 500);

insert into STUDENTS values (456, 'Doris', 'Roberts', 3.9, 1000);

insert into STUDENTS values (567, 'Edward', 'Norton', 2.9, 2000);

insert into STUDENTS values (678, 'Fay', 'Laurence', 3.8, 200);

insert into STUDENTS values (789, 'Gary', 'Oldman', 3.4, 800);

insert into STUDENTS values (987, 'Helen', 'Karlson', 3.7, 800);

insert into STUDENTS values (876, 'Irene', 'Lopez', 3.9, 400);

insert into STUDENTS values (765, 'Jay', 'Farlong', 2.9, 1500);

insert into STUDENTS values (654, 'Amy', 'Cooper', 3.9, 1000);

insert into STUDENTS values (543, 'Craig', 'Wilder', 3.4, 2000);

insert into COLLEGES values ('Stanford', 'CA', 15000);

insert into COLLEGES values ('Berkeley', 'CA', 36000);

insert into COLLEGES values ('MIT', 'MA', 10000);

insert into COLLEGES values ('Cornell', 'NY', 21000);

insert into APPLIES values (123, 'Stanford', 'CS', true);

insert into APPLIES values (123, 'Stanford', 'EE', false);

insert into APPLIES values (123, 'Berkeley', 'CS', true);

insert into APPLIES values (123, 'Cornell', 'EE', true);

insert into APPLIES values (234, 'Berkeley', 'biology', false);

insert into APPLIES values (345, 'MIT', 'bioengineering', true);

insert into APPLIES values (345, 'Cornell', 'bioengineering', false);

insert into APPLIES values (345, 'Cornell', 'CS', true);

insert into APPLIES values (345, 'Cornell', 'EE', false);

insert into APPLIES values (678, 'Stanford', 'history', true);

insert into APPLIES values (987, 'Stanford', 'CS', true);

insert into APPLIES values (987, 'Berkeley', 'CS', true);

insert into APPLIES values (876, 'Stanford', 'CS', false);

insert into APPLIES values (876, 'MIT', 'biology', true);

insert into APPLIES values (876, 'MIT', 'marine biology', false);

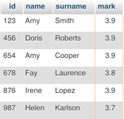
insert into APPLIES values (765, 'Stanford', 'history', true);

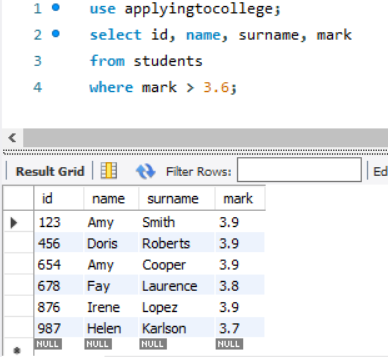
insert into APPLIES values (765, 'Cornell', 'history', false);

insert into APPLIES values (765, 'Cornell', 'psychology', true);

insert into APPLIES values (543, 'MIT', 'CS', false);

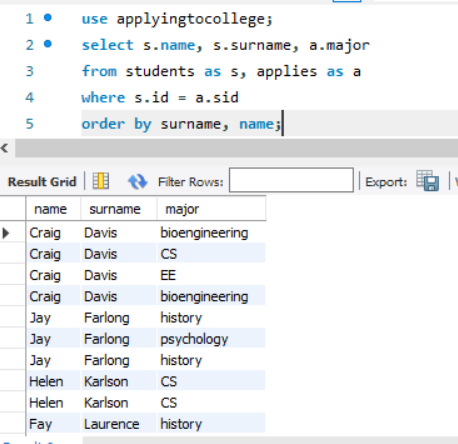
2.- Show me ids, names, surnames, and marks of the students with mark > 3.6.





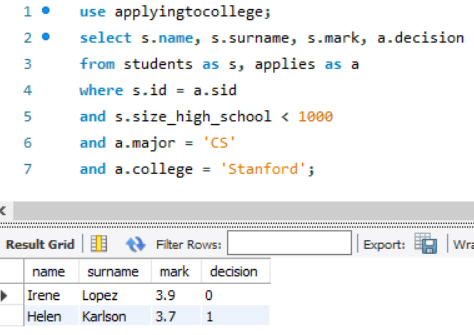
3.- Show me student names, surnames, and majors that they have applied to. Sort the results by name and surname.





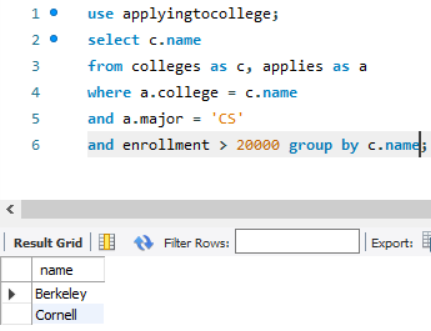
4. Show me names, surnames, marks and application decision of students with size\_high\_school < 1000 applying to  CS at Stanford.



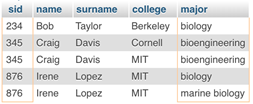


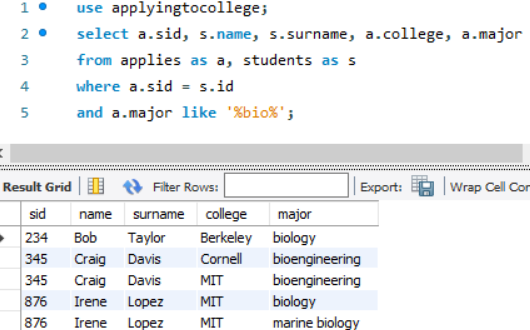
5. Show me all large campuses (enrollment>20000) with CS applicants.



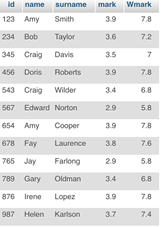


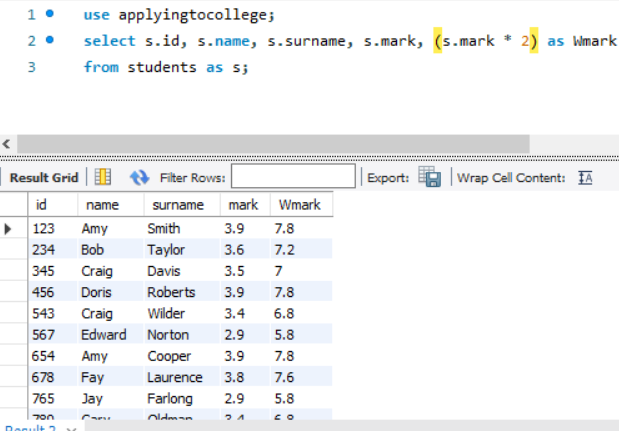
6.-  Show me applicants to bio majors.



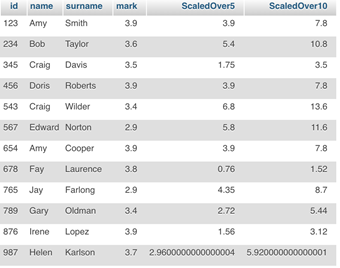


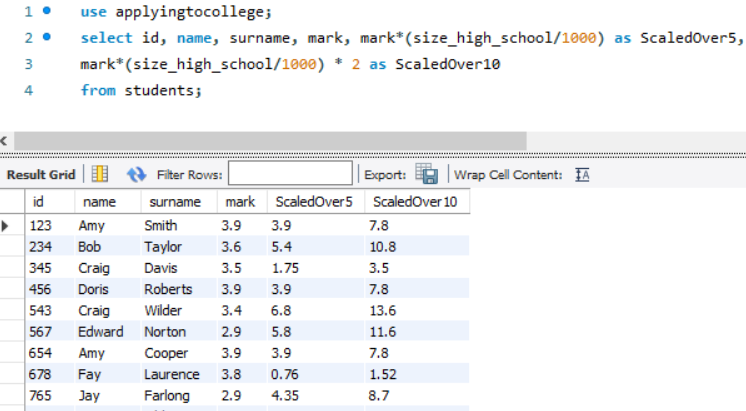
7. Show the weighted scores on 10 points (now weighted on 5 points).





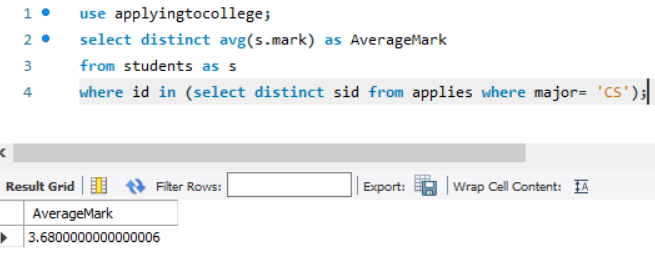
8. Scale the marks considering the size of the high school (mark\*(high\_school\_size/1000)). Show the weighted scores on 5 and 10 points.





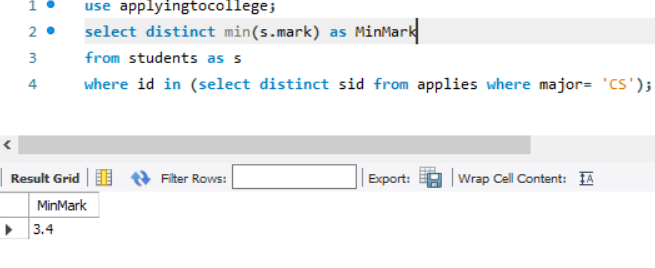
9. Show me the average mark of all students applying to CS (you must think that the same student could apply for CS in many Universities). Clue: wrong average 3.714285714285714, right average 3.6800000000000006.





10. Show me the lowest mark of students applying to CS.





use applyingtocollege;

select distinct min(s.mark) as MinMark

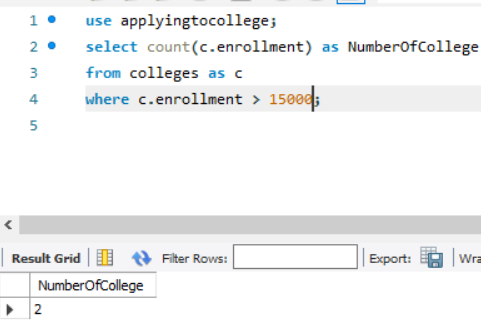
from students as s, applies as a

where a.sid = s.id

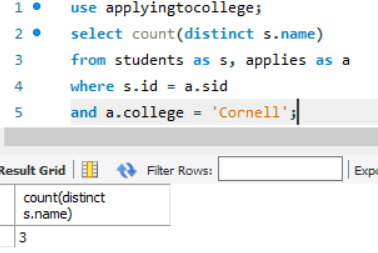
and a.major = 'CS';

11. Show me the number of colleges bigger than 15,000.



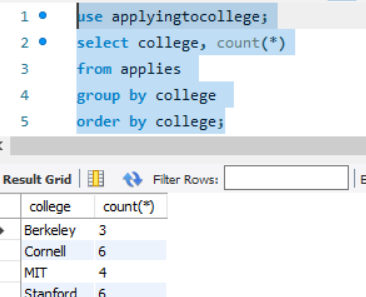


12. Show me the number of students applying to Cornell (you must thing that students can apply to many majors in the same University). Clue: The right number should be 3.

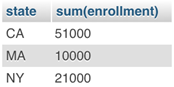


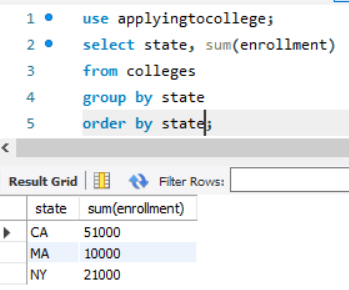
13. Show me the number of applications to each college (sorted by college name).





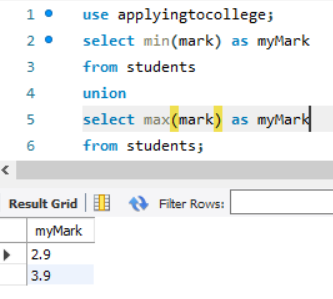
15. Show me the number of college enrollments by state.





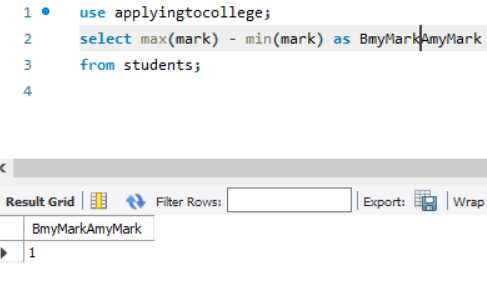
16. Show me the minimum and maximum marks of all the students.



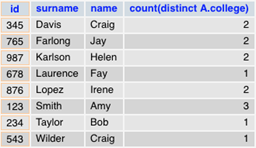


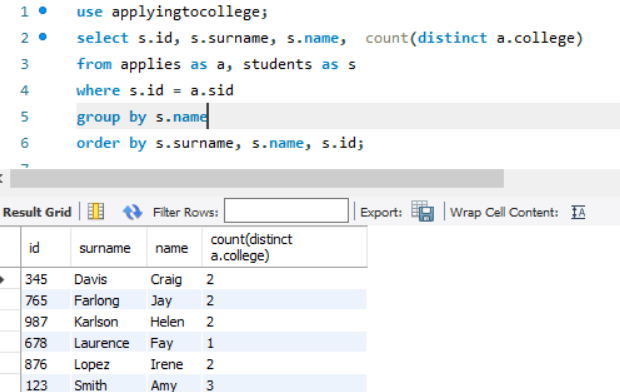
17. Show me the difference between the maximum and the minimum marks of all the students.



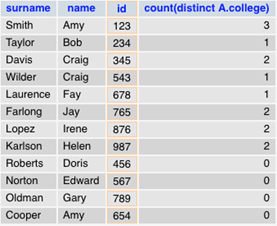


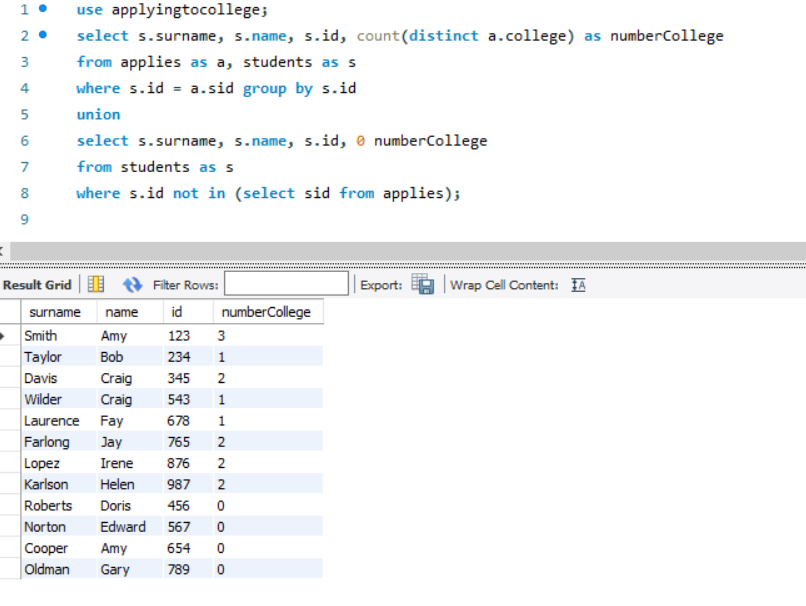
18. Show me the number of colleges applied to by each student. Sort the results by student surname, student name and student id.



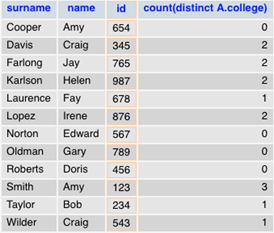


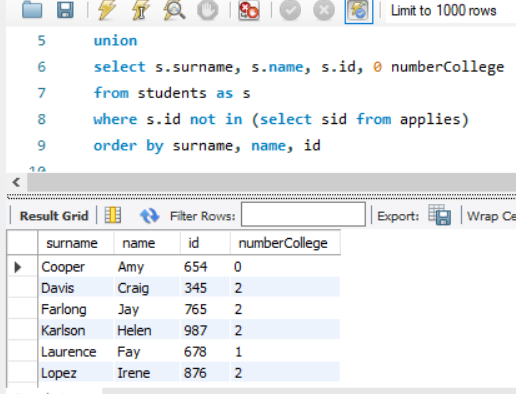
19. Show me the number of colleges applied to by each student, including  0 for those who applied nowhere. CLUE: Do the union between to queries.





20. Do the last query sorting by student surname, student name and student id.

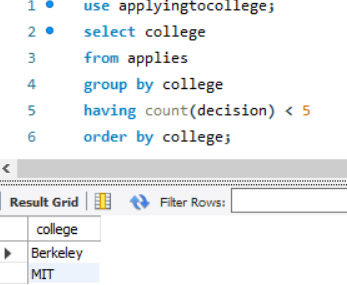




21. Show me the colleges with fewer than 5 applications. Show only the college name and sort the results by name. Do the exercise with a simple query.

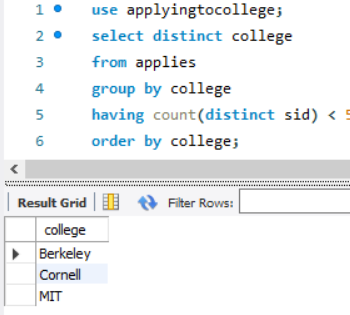
In this result duplicate applications are considered:





In this result duplicate applications are not considered:



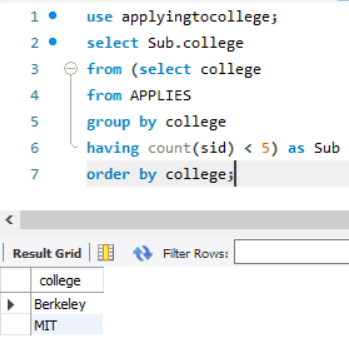


Do both queries!

22. Show me colleges with fewer than 5 applications. Show only the college name and sort the results by name. Do the exercise using a subquery.

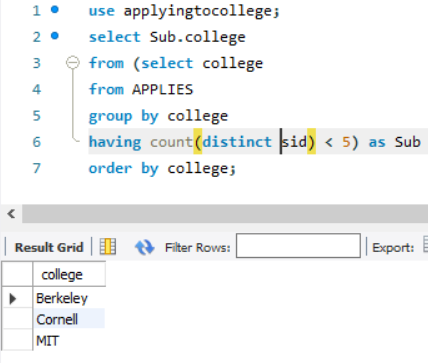
In this result duplicate applications are considered:





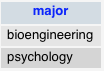
In this result duplicate applications are not considered:

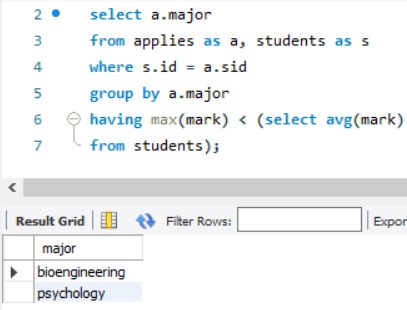




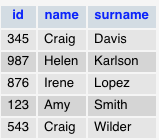
Do both queries!

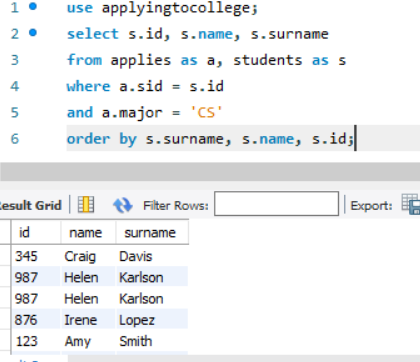
23. Show me the majors whose applicant's maximum mark is below the average.



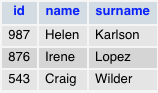


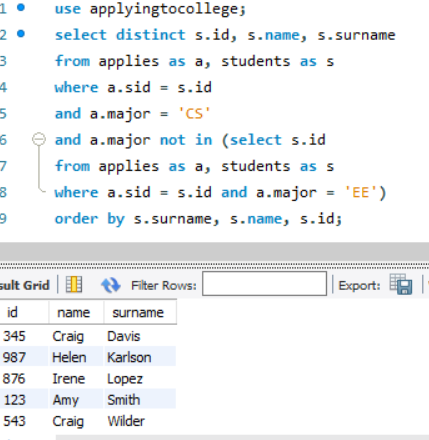
24. Show me IDs, names, and surnames of students applying to CS.  Sort the results by student surname, student name and student id.





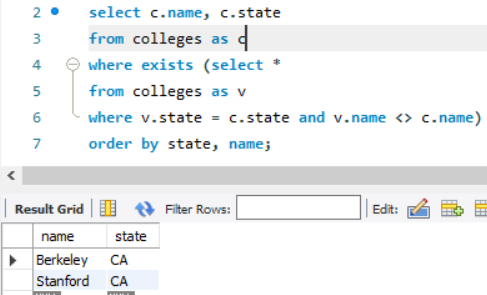
25. Show me the students who applied to CS but not EE. Sort the results by student surname, student name and student id.





26.  Show me the colleges such that some other college is in the same state. Sort the results by state and college. It’s mandatory to use EXISTS.

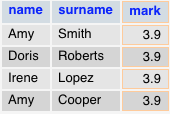




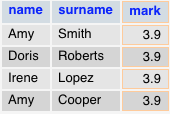
27. Show me the name of the biggest college.



28. Show me the student (or students) with highest mark (using EXISTS).



29. Show me the student with highest mark (using "where mark >= all")



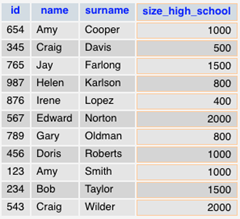
30. Show me the name of the college with the higher enrollment than all other colleges (using 'where enrollment > all').



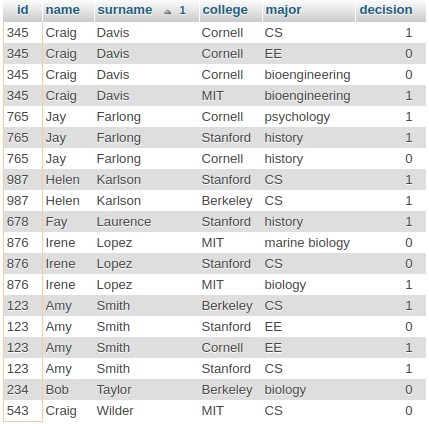
31. Show me the name of the college with the higher enrollment than all other colleges (using 'not enrollment <= Any')



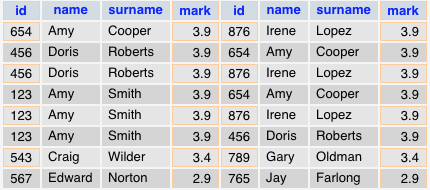
32. Show me the students not from the smallest high school.



33. Show me the application information order by student surname and name.



34. Show me the pairs of students with same mark (ordered by its marks descendent and surnames/names ascendent).



35.  Show me a list of college names and student names together ordered by name.



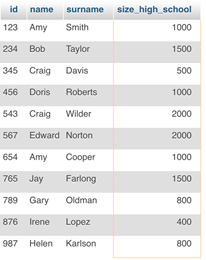
36. Show me IDs of students who applied to both CS and EE.



37. Show me IDs of students who applied to CS but not EE.



38.  Show me the students that don’t belong to the smallest high school (use exists).



39. Show me id of the students who applied to CS but not EE (use ‘any’ and two subqueries).



40.  Insert a new college with name 'UIB', in the state 'IB', and with size 11500.

41. Insert into APPLIES with college 'UIB', major 'IB', AND DECISION NULL all students who didn't apply anywhere.  (Clue: insert with subselect).

42. Admit to UIB EE all students who were refused (decision=false) in EE elsewhere.