Make sure your tables are as shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Apply** | | | |
| sID | cName | major | decision |
| 123 | Cornell | EE | Y |
| 123 | OSU | CS | Y |
| 123 | OSU | EE | N |
| 123 | U of O | CS | Y |
| 123 | MIT | CS | N |
| 234 | U of O | biology | N |
| 345 | Cornell | bioengineering | N |
| 345 | Cornell | CS | Y |
| 345 | Cornell | EE | N |
| 345 | MIT | bioengineering | Y |
| 543 | MIT | CS | N |
| 678 | Cornell | history | N |
| 678 | Cornell | psychology | Y |
| 678 | OSU | history | Y |
| 765 | OSU | history | Y |
| 876 | MIT | biology | Y |
| 876 | MIT | marine biology | N |
| 876 | OSU | CS | N |
| 987 | OSU | CS | Y |
| 987 | U of O | CS | Y |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student** | | | |
| sID | sName | GPA | sizeHS |
| 123 | Amy | 3.90 | 1000 |
| 234 | Bob | 3.60 | 1500 |
| 345 | Craig | 3.50 | 500 |
| 456 | Doris | 3.90 | 1000 |
| 543 | Craig | 3.40 | 2000 |
| 567 | Edward | 2.90 | 2000 |
| 654 | Amy | 3.90 | 1000 |
| 678 | Fay | 3.80 | 200 |
| 765 | Jay | 2.90 | 1500 |
| 789 | Gary | 3.40 | 800 |
| 876 | Irene | 3.90 | 400 |
| 987 | Helen | 4.00 | 800 |

|  |  |  |
| --- | --- | --- |
| **College** | | |
| cNmae | State | enrollment |
| Cornell | NY | 21000 |
| MIT | MA | 10000 |
| OSU | OR | 28000 |
| U of O | OR | 25000 |

|  |  |  |
| --- | --- | --- |
| **MinimumGPA** | | |
| cName | major | minGPA |
| OSU | CS | 3.75 |
| OSU | EE | 3.5 |
| OSU | history | 2.8 |
| U of O | CS | 3.6 |
| U of O | biology | 3.75 |
| Cornell | bioengineering | 3.8 |
| Cornell | CS | 3.4 |
| Cornell | EE | 3.6 |
| Cornell | history | 3.6 |
| Cornell | psychology | 2.8 |
| MIT | biology | 3.5 |
| MIT | bioengineering | 3.5 |
| MIT | CS | 3.9 |
| MIT | marine biology | 3.5 |

**Answer the following questions about the College Application Database that you imported into PHPMyAdmin. Note: you will need to add the minimum GPA table.**

Problem: 1 (**24 pts, 3pts each)** Give the SQL query statement to answer the following questions along with the resulting table.

1. List the names and majors of all students applying to a major related to biology. *(Hint: use % and bio)*

SELECT Student.sName, Apply.major FROM Student INNER JOIN Apply on Student.sID = Apply.sID where Apply.major like 'bio%'

1. Find the number of students applying to Cornell?

SELECT count(DISTINCT sID)

FROM Apply where cName like 'cornell'

1. List the names and ids of students applying to a CS major and from a small HS (less than 600 students).

SELECT Student.sName, Student.sID FROM Student INNER JOIN Apply on Student.sID = Apply.sID where Apply.major like 'cs' and Student.sizeHS <= 600

1. Determine the average minimum GPA requirement for each college.

select avg(minGPA), cName from MinimumGPA GROUP BY cName

1. Determine the name of the college with the greatest average minimum GPA requirement.

select cName, MAX(avg\_gpa)

FROM (SELECT cName, avg(minGPA) as avg\_gpa from MinimumGPA GROUP BY cName) as subquery

1. List the sID and names of students that applied to both OSU and Uof O.

SELECT subquery.sID, Student.sName from (SELECT Apply.sID, Apply.cName from Apply where Apply.cName like 'OSU'

UNION

SELECT Apply.sID, Apply.cName from Apply where Apply.cName like 'u of o') as subquery

inner JOIN Student on subquery.sID = Student.sID

GROUP by subquery.sID

HAVING COUNT(subquery.sID) > 1

1. List the names and ids of the students that applied to OSU but did not apply to U of O.

Ok, this one was really dumb, but I’ll try to explain what I did. I used a union to find the number of occurrences, and wanted a value of 1 for anyone that has applied to osu, I then wanted to increment it by one more for people who have applied to u of o as well, but I realized that the bob student became an issue for this because he only applied to u of o once, so I just made the query in the union select a separate row to increment his value again.

SELECT subquery.sID, Student.sName from (SELECT Apply.sID, Apply.cName from Apply where Apply.cName like 'OSU'

UNION

SELECT Apply.sID, Apply.cName from Apply where Apply.cName like 'u of o'

UNION

SELECT Apply.sID, Apply.major from Apply where Apply.cName like 'u of o') as subquery

inner JOIN Student on subquery.sID = Student.sID

GROUP by subquery.sID

HAVING COUNT(subquery.sID) <= 1

1. For students with GPA’s higher than the minimum requirements for the major they applied for, give the student’s name, GPA, college name and major.

SELECT Student.sName, Student.sID, Student.GPA, Apply.cName, Apply.major FROM Student

INNER JOIN Apply on Student.sID = Apply.sID

INNER JOIN MinimumGPA on Apply.major = MinimumGPA.major AND Apply.cName = MinimumGPA.cName

WHERE Student.GPA >= MinimumGPA.minGPA

Problem 2: **(6 pts, 2 pts each)** Give the SQL command for the following**. *Note: Do not execute the commands and change your database***

1. Write an SQL statement to add a student with sID of 888, sName of Pat, GPA of 3.45 and sizeHS of 800 to the Student table.

INSERT INTO `Student` (`sID`, `sName`, `GPA`, `sizeHS`, `Advisor`) VALUES ('888', 'Pat', '3.45', '800', NULL);

1. Write an SQL statement to remove Pat from the Student table.

Delete from student where sID = 888;

1. Write an SQL statement to change the GPA of Jay from 2.90 to 3.10.

Update student set GPA = 3.10 where sID =765;

**EXTRA CREDIT:** List the names and ids of the students that applied to all schools. Do not use the names of the schools in your query.