Download the collegeApplicationDB.sql file from Canvas and import into MySQL. 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 |

1. Create the table : CollegeStats (cName: VARCHAR(20), appCount: int)

This table will contain for each college in the College table the number of applications to the school. Insert the cNames of the colleges from the College table into CollegeStat .

2. Write a stored procedure called updateCollegeStats(name) that takes as a parameter a college name and then updates the count of all applications to that school. Give the code and resulting table.

***Change all counts back to zero before starting problem 4.***

3. Write a stored procedure called updateAllCollegeStats that uses a cursor to update the count of total applications of the colleges in the CollegeStats table. Give the code and resulting table.

4. Write a trigger called incrApplyStats for the Apply table. This trigger will increment the application count for the appropriate college in the CollegeStats table after a new record is inserted into the Apply table.

Test by inserting the record (543, OSU, CS, Y) into Apply. Give the resulting tables for Apply and CollegeStats and code for the trigger.

5. Write a trigger called decrCollegeStats for the Apply table. This trigger will decrease the application count for the appriate college in the CollegeStats table a record is deleted from the Apply table.

Test by inserting the record (543, OSU, CS, Y) into Apply. Give the resulting tables for Apply and CollegeStats and code for the trigger.

**EXTRA CREDIT**

A. Create a trigger called AutoAccept that sets decision to “Y” if an application tuple is inserted into the Apply table and the corresponding student has a GPA greater than or equal to the minimum GPA required for that major and college.

B. Create a procedure that AutoAccepts applications in the Apply table that meet the criteria in part A.