Using Explain evaluate the following query:

**Select \***

**From Student join enrollment on student.ID = enrollment.Student\_ID**

**join section on section.ID = enrollment.section\_ID**

**join department on major = department.name**

**join faculty on faculty.id = section.faculty\_ID**

**join address on address.id = student.address\_ID**

**join Course on section.course\_Number = course.num and section.dept\_id = course.dept\_ID**

**Where**

**--we want to make sure we have name information for students if we want to reach out to them**

**Student.Name\_Last Not Like ('')**

**-- the theater department has asked to be out of this study**

**and Student.Major <> 'Theater'**

**--no students who have failed as we're looking for passing grades**

**and Grade > '1.33'**

**--we want to make sure we only have instructors, and the theater department is not part of this study**

**and Faculty.job in**

**(Select job**

**From Faculty**

**Where Job not in ('Administrative','General Services','Human Resources')**

**and Dept <> 'THT')**

**and Section\_ID >=1**

**--summer courses don't always reflect accurately given their tight schedule and rapid fire delivery of materials**

**and Semester <> 'Summer'**

**--we don't want bias of an adivosr giving better grades**

**and Student.Advisor\_ID <> Section.Faculty\_ID**

**-- we don't want bias if a student is possibly a faculty members child**

**and Student.Address\_ID <> Faculty.Address\_ID**

**Order by Student.Name\_Last, Grade desc, Faculty.Name\_Last, Major**

1. Based on the output from the explain command, determine the steps you would take (building indexes on certain columns, modifying the query, both, etc.) to get the same output, but improve overall performance.
2. Execute three steps that you think would make for improvement in the performance of this query. Report their run time in the table below.

* **Not using \* in Select \***

If we use \* then it gives all columns of all these tables and it takes time so that’s why it’s better to use the column name which is required for the result. Therefore, give column names exactly under select section hence the time taken to get all the columns will be less.

Select student.ID, section.ID, student.Major, department.name, faculty.id, address.id, section.course\_Number, course.dept\_ID, course.num, section.dept\_id, enrollment.Student\_ID, enrollment.section\_ID, Student.Name\_Last, Student.Major, Faculty.job, Student.Advisor\_ID, Section\_ID, Section.Faculty\_ID, Student.Address\_ID

Graphical user interface, text, application

Description automatically generated

* **Use more indexes**

An index creates a unique data column without overlapping each other. It improves the speed of data retrieval. Therefore below indexes created.

CREATE INDEX ix\_id

ON faculty(id);

CREATE INDEX ix\_fac\_id

ON section(faculty\_id);

CREATE INDEX ix\_dept\_id

ON section(dept\_id);

CREATE INDEX ix\_student\_id

ON student(id);

CREATE INDEX ix\_major

ON student(major);

A picture containing text

Description automatically generated

* **Avoid Negative search**

As we see we are using negative search in where conditions like not equals (<>), not like, etc. but we need to avoid that type of search because if we give the exact search criteria on that time while query executing if that type of data is got then immediately query returns the result. Negative searches decrease the query execution speed that’s why we need to avoid that.