# Assignment 8 – Indices

Good indexing can make or break your database. Without indices, a database management system must cycle through every record for every query (on the relevant tables).

For this assignment, you must consider every table in your database. For each table, you should consider:

1. How many records do I expect? Is this a tiny, data reference table (like “Freshman, Sophomore, Junior, Senior”) or is this an enormous table, like one for every grade every student has ever received from this University?
2. How often will this table be updated?
3. How often will this table be searched/queried?

These questions indicate how important it is to make an index and what the performance implications are.

Consider the columns that you think will make a good index (you can have many indices):

1. How selective can I make the index (how much of the table can I limit the search to with a particular index)?
2. How large would a particular index be? Large fields take more index space than smaller fields.
3. How often would this particular column or collection of columns be updated? Even if the table is updated often, perhaps a particular column would not be.

After considering these things, please write the SQL to create any indices that you determine appropriate.

Example:

We have a table:

StudentFirstName, StudentLastName, StudentGPA, Major, EnrollmentStatus

At this University, we might expect 15,000 students PER YEAR. This table could have a few million rows.

This table will be inserted into every Fall, with huge updates at the end of the semester and minor updates daily.

This table would probably be queried often for reports.

Considering columns –

FirstName is not a column that makes sense for reports. It will be ON the reports, but no one will likely search for all of the “Michael”s.

LastName is a column that you MIGHT search for, but not all that often. Names are kind of big (making a big index). On the other hand, a last name index would be very selective (if I search for “Phipps”, I will likely find only a few records). This index would very rarely be updated. With high selectivity and low updatability arguing **for** an index, but usage and size arguing against, I probably would not index this. Future usage may change my mind.

StudentGPA is probably not very selective. There will be a small number of scores that most people have (there are only 400 possible – 0.00 – 4.00). The index would be small, since a NUMBER(3,2) is tiny. The value in GPA changes twice a year. Typically with GPA, I am not looking for a particular value (except maybe 0.00 or 4.00), so an index wouldn’t be helpful. I would not index this.

Major is not very selective – it would split the million(s) of rows by the number of majors (let’s call that 100). It is very useful because I would probably want to search or group by major. The index would be fairly small, since there are not a lot of names for majors. This column is not updated very often. I would index on major:

CREATE INDEX IX\_STUDENT\_MAJOR ON student(major)

EnrollmentStatus is probably very selective. If we have statuses: Enrolled, Graduated, Abandoned (dropped out), most students will NOT be enrolled. Selecting 15,000 out of 1.5 million is very selective. The index would be fairly small based on having only three categories. This column would not be updated very frequently (one big update at the end of the Spring, with little updates throughout). I would index on EnrollmentStatus:

CREATE INDEX IX\_STUDENT\_ENROLLMENTSTATUS on student(enrollmentStatus)

Please put your indices in the description, but also make a separate SQL file for them.

**Your script must have a .sql extension and must be uploaded to blackboard. A script that doesn’t run is an automatic 0.**

**You must run each of your queries to test that it works.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rubric | Poor | OK | Good | Great |
| Table Sizing | Missing(0) | Some are ok (6) | Most are ok (13) | Nearly all are OK (20) |
| Table Updating | Missing(0) | Some are ok (6) | Most are ok (13) | Nearly all are OK (20) |
| Table searched/queried | Missing(0) | Some are ok (6) | Most are ok (13) | Nearly all are OK (20) |
| Column selectivity/size/update | Missing(0) | Some are ok (6) | Most are ok (13) | Nearly all are OK (20) |
| Indices are correct and reasonable | Missing(0) | Some are ok (6) | Most are ok (13) | Nearly all are OK (20) |