# *Database Management II (420-D20-HR)*

# *Lab 10 –* Optimization – Database Indices

Date assigned: Tuesday, April 18, 2017

Date due: Tuesday, April 18, 2017 17:50

**Objectives:**

At the end of this lab you will be able to:

1. Understand what a database index is and how it works.
2. Create and drop indicies
3. Analyze the effect of indexing on a database

**To Start:**

1. Rename this document to ***username*\_D20\_L10\_OptimizationII.docx**.

**To Be Handed In:**

1. Your ***username*\_D20\_L10\_OptimizationII.docx** should be uploaded to Moodle.

# Research

Learn about what a database index is, the costs and benefits. Research, starting [here](https://en.wikipedia.org/wiki/Database_index) and answer the following questions.

1. What is the definition of a database index (in your own words)?

A data structure to help improve selecting data from a database, however it means an additional cost in create/updates/deletes

1. What is the performance of a linear (non-indexed) search versus an indexed search? (Describe in big ‘O’ notation). Give an example with a table of 1 million entries what the worst case efforts would be.

Using a linear search through the table would yield a big Oh notation of O(N), because at max you’ll need to check every row in the table. Using an indexed search, the indexes are stored in a binary tree, which has a search of O(logn).

1. Does an index help with wildcard searches? i.e. WHERE last\_name like ‘%ardasian%’. Why or why not?

No it doesn’t, because context sensitive from left to right

1. What are two key uses/benefits of an index?

They make searching through a database much faster because you can search through a non-key column as though it were a primary key. They give searching a speed of O(N)

Police the DB constraints

1. What are the costs/disadvantages of an index?

It means that every time you create/update/delete a row, you need to rebalance your binary tree. And storage space

1. For compound (multiple-column) indexes, what relevance does the column order of the index make?

Left-right sensitivity

# Hands-On

Open the D20\_L10\_OptimizationII.sql file included with this lab. There you will find the code snippets for this lab.

## Create and populate the emp\_opt table as described in the sql file. Fill in the following based on your guess as to whether cost will be low or high. Do not fill in the shaded cells.

Note: Assume:

* emp\_id is always the primary key
* you only have one experimental index at a time. For example, you first add an index for first name, do your costing, then remove that index, move on, and add the index on last name.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lookup # | Description | No additional indicies | Index on first name | Index on last name | Index on last,first name | Index on Salary |
| 1 | Select random employee by emp\_id | Low |  |  |  |  |
| 2 | Get person by first name | High | Low | High | Low |  |
| 3 | Get person by last name | High | High | Low | Low |  |
| 4 | Get person by last, first name | High | High | High | Low |  |
| 5 | Get person by first, last name | High | Low | Low | Low |  |
| 6 | Find all people with salary > 200,000 | High |  |  |  | Low |
| 7 | Calculate average salary | High |  |  |  | High |

## Fill in the following based on the ‘explain plan’ feature in Oracle SQL Developer. Do not fill in the shaded cells.

Note: Assume:

* emp\_id is always the primary key
* you only have one experimental index at a time. For example, you first add an index for first name, do your costing, then remove that index, move on, and add the index on last name.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lookup # | Description | Actual Cost  (Explain Plan) | Index on first name | Index on last name | Index on last,first name | Index on Salary |
| 1 | Select random employee by emp\_id | 2 |  |  |  |  |
| 2 | Get person by first name | 272 | 3 | 272 | 272 |  |
| 3 | Get person by last name | 272 | 272 | 3 | 5 |  |
| 4 | Get person by last, first name | 272 | 14 | 14 | 5 |  |
| 5 | Get person by first, last name | 272 | 14 | 14 | 5 |  |
| 6 | Find all people with salary > 200,000 | 273 |  |  |  | 273 |
| 7 | Calculate average salary | 272 |  |  |  | 272 |

## In what cases does the compound index (lastname, firstname) speed up name searches? Why?

It will always speed up the query for a last name, or a first name and a last name

Anything with a last name

## Why is the emp\_id lookup so quick despite you not having explicitly added an index for it?

Because it’s the primary key

## If database indicies are so great at improving performance, why not index everything that we might need to lookup by?

Because they make creates/updates/deletes a lot slower. If you need to read a lot from the table, they might be worth it, but otherwise not. And space

## Clean up before you leave. Before you finish this lab, please remember to drop the emp\_opt table. They take up a lot of space. Have you dropped emp\_opt?

Yes

# Assessment

1. Complete the CES (Course Evaluation System) feedback for D20 and E21. You were sent an email invitation for this today. Have you completed both D20 and E21 assessment?

Yes

1. What did you learn in completing this lab?

Indexes

1. What did you have difficulty with?

Nothing was super hard, I just had a hard time focusing on not code

1. What did you do well?

The evaluation of the execution plan

1. How many hours did you spend in completing this lab?

2

1. What took you the most time?

Part A because I couldn’t focus on it