Introduction to Database Systems
Exercise Week 8: Indexing and Performance

Björn Þór Jónsson

Readings

PDBM: Chapter 12

The Exercise

The goal of the exercise is to experience the (positive and negative) impacts of indices. In the process, you will get a first taste of database administration, which can be a very slow process.

You are given two different Sports databases, with 20K and 2M rows in the Results relation, respectively. Note that even the larger database is still quite small, but large enough that execution of the various scripts may take a while. You should preferably work in pairs or small groups (e.g., via screen sharing) so that you can discuss the results.

For each database size, you will work also with a version of the database that has PRIMARY KEY and FOREIGN KEY definitions, and the corresponding indexes, and one that does not have these definitions and therefore no indexes.

You may thus immediately create four databases: E8KS (keys, small), E8NS (no keys, small), E8KL (keys, large) and E8NL (no keys, large). In the appendix, you can find a list of all commands used in the exercise, in rough time order.

Note: This assignment is created with `psql` ***(command prompt) in mind. If you haven't managed to run*** `psql` ***on your computer, then a) do this exercise with someone who can run*** `psql`***, and b) talk to the TAs to get*** `psql` ***running.***

Part I: Filling the Database

Using the E8NS database, run the commands in 1_CREATE_NOKEY.sql, which create a database schema very similar to Homework 1 (two large text fields have been added) but without any PRIMARY KEY and FOREIGN KEY constraints.

Note: When running scripts, use the command prompt and specify the database to run them in.

Note: The SQL scripts include commands to measure time. You need to use a `-q` ***parameter in your*** `psql` ***command, to avoid printing INSERT 0 1 for every insertion—when you do two million insertions later, that would take a while!***

Then populate the schema with the commands in 2_FILL_20K.sql, again running inside the E8NS database. Write down how long it takes, so that you can compare the insertions with and without the overhead of index maintenance.

Note: In the 2_FILL_X.sql scripts, automatic commits have been turned off to remove transaction processing overheads.

Turning to the E8KS database, which has primary and foreign keys, run the commands in 1_CREATE_KEY.sql, which again create a database schema very similar to Homework 1 (two large text fields have been added), but with all the PRIMARY KEY and FOREIGN KEY constraints.

Then populate the schema with the commands in 2_FILL_20K.sql, as before, but running inside the E8KS database. Write down how long it takes, so that you can compare the insertions with and without the overhead of index maintenance.

Note: It is likely that if you run the queries twice, the second run will be faster than the first run. If you then run them again, the query time should not change much. Why is this?

Part II: Small Database

Run 3_QUERY_I.sql and 3_QUERY_II.sql in the E8NS database. Write down how long it takes to execute each of the two queries.

Then run 3_QUERY_I.sql and 3_QUERY_II.sql in the E8KS database and compare the execution time of Query I and Query II with that of the E8NS database.

Note: The 20K database is so small that you may not see significant differences in query execution time in this part, measured with the SELECT NOW command. If that happens, try adding the EXPLAIN ANALYZE keywords to the queries (before SELECT) and run them again. This method gives a more accurate measurement of running time. Why do you think that is?

- What could explain the difference in execution time between Query I and Query II? Think about the number of records surviving the join condition.
- Why do you think the queries run faster in the E8KS database? If they do!
- Is the relative speed of query I and II the same with and without indexes? If not, then why?

Try opening pgAdmin and running the queries with the EXPLAIN ANALYZE keywords added to the front, in both databases.

- Do you see indices used in the query plans resulting from the EXPLAIN ANALYZE command?

Part III: Query Tuning

Consider Query 14 from Exercise 2. In the scripts 4_QUERY_I.sql and 4_QUERY_II.sql are two different versions of this query.

Run the two versions of Query 14 in the E8NS database. Write down how long the execution of each takes. Abort queries that takes more than a minute.

Now run the two versions of Query 14 in the E8KS database. Write down how long the execution of each takes. Again, if the slower version (or both) takes more than a minute, abort the execution.

- Which query/database combinations can complete within a reasonable time?
- Can you explain why execution would be faster in the E8KS database? Again, using EXPLAIN ANALYZE may help…
- Can you explain why version II is so much faster? Does it have anything to do with indexes?

Part IV: Large Database

Repeat Parts I, II and III using the 2_FILL_2M.sql script to fill the E8NS and E8NL databases. Write down all the fill/query times, as you did in parts I, II and III, and try to answer all the questions posed in those parts. In many cases, they will be easier to answer, as the performance differences will be more clear.

Note: The fill script may take a while to run (on my laptop, it took about 4 minutes). If your computer is slower, then you may share results. The discussions are the most important part of your learning from this exercise.

Note: The query in 4_QUERY_I.sql will very likely take a long time, especially for the E8NL database. If it runs more than a minute you may simply cancel execution, as it is already unacceptably long. Other queries should complete within (tens of) seconds on reasonable hardware.

Part V: Index Tuning

The command to create an index is:

```
create index <name> on Results(<columns>);
```

Based on your knowledge, now try to create an index to speed up the execution of the query version in 4_QUERY_II.sql in this large database.

Hint: There is a single best index for this query. Focus on the attributes used in the sub-query. See if you can make a covering index for the (two) attributes used from the Results relation.

Hint: You can start with the E8NS database, if you wish to make fast experiments. You should see some impact of the correct index even there.

- Which attribute(s) should be included in the index and in which order, for best results?
- Does this index have an impact on the query version in 4_QUERY_I.sql as well? Why?

Part VI: Summary

Now look at all your numbers, and see whether you can spot both benefits and drawbacks of indices.

- If filling the database is so much slower with indices, then why do we (and why must we!) use them?

Appendix: Commands

The following are the commands that I have used to solve the exercise, roughly in this order. Depending on which user you are using, you may need to modify the commands to fit your situation.

// Create the databases

```
createdb -U postgres E8NS
createdb -U postgres E8KS
createdb -U postgres E8NL
createdb -U postgres E8KL
```

// Create the tables (with or without keys/foreign keys)

```
psql -q E8NS postgres < 1_CREATE_NOKEY.sql
psql -q E8KS postgres < 1_CREATE_KEY.sql
psql -q E8NL postgres < 1_CREATE_NOKEY.sql
psql -q E8KL postgres < 1_CREATE_KEY.sql
```

// Fill the databases **(from now on, the time should be measured)**

```
psql -q E8NS postgres < 2_FILL_20K.sql
psql -q E8KS postgres < 2_FILL_20K.sql
psql -q E8NL postgres < 2_FILL_2M.sql
psql -q E8KL postgres < 2_FILL_2M.sql
```

// Run the queries to filter sports/people on the **small** database

```
psql E8NS postgres < 3_QUERY_I.sql
psql E8NS postgres < 3_QUERY_II.sql
psql E8KS postgres < 3_QUERY_I.sql
psql E8KS postgres < 3_QUERY_II.sql
```

// Run Query 14 from Exercise 2 on the **small** database

```
psql E8NS postgres < 4_QUERY_I.sql
psql E8NS postgres < 4_QUERY_II.sql
psql E8KS postgres < 4_QUERY_I.sql
psql E8KS postgres < 4_QUERY_II.sql
```

// Run the queries to filter sports/people on the **large** database

```
psql E8NL postgres < 3_QUERY_I.sql
psql E8NL postgres < 3_QUERY_II.sql
psql E8KL postgres < 3_QUERY_I.sql
psql E8KL postgres < 3_QUERY_II.sql
```

// Run Query 14 from Exercise 2 on the **large** database

```
psql E8NL postgres < 4_QUERY_I.sql
psql E8NL postgres < 4_QUERY_II.sql
psql E8KL postgres < 4_QUERY_I.sql
psql E8KL postgres < 4_QUERY_II.sql
```