Exercises 10

The purpose of these exercises is to get some experience in applying your knowledge of query execution to improve database design and SQL.

IMDB revisited

Consider again the queries from the project:
1. Which movies have John Travolta and Uma Thurman starred in together?
2. Which actors and directors have a name starting with Q?
3. What is the name and birth year for all actors in Pulp Fiction? Your query should list the actors in increasing order of birth year.
4. What are the titles and years of all movies from the 90s where John Travolta starred?
5. How many Oscars has Quentin Tarantino won?
6. In 1994, what was the average length of a movie for each language?
7. Which actors in Pulp Fiction have never, before or after, starred in the same movie as one of the other actors in Pulp Fiction.
8. Which movie starring John Travolta has the highest user rating?

For each of the above, consider how you would imagine that the DBMS executes the query, given the indexes you have made available. Use the EXPLAIN feature in Query Browser (or type EXPLAIN in front of a query) to see the query plan. Information on query plans can be found at: http://dev.mysql.com/doc/refman/5.0/en/using-explain.html

For each query, consider what change in the database schema (denormalization, partitioning, or indexing) could speed up its execution.

5 Database efficiency (15%)

Consider the relation \( T(id, name) \) with 100,000 tuples. Values of \( id \) are positive integers, and values of \( name \) are strings of length at most 30. The following queries are of interest:

1. \( \text{SELECT } * \text{ FROM } t \text{ WHERE } id = 100 \)
2. \( \text{SELECT } * \text{ FROM } t \text{ WHERE } id > 10 \)
3. \( \text{SELECT } * \text{ FROM } T \text{ WHERE } id > 100 \text{ and } id < 9000 \text{ and } name = 'Mads' \)

a) State for each of the above queries whether an index would speed it up. In the cases where the answer is “yes” you also have to specify:
- Would a Hash-index or a B-tree index be the fastest index?
- Which attribute(s) should be indexed?
- Would a clustered index make the query significantly faster than an unclustered index?