Fundamentals of Data Management

Credit Tasks 10.2: Performance

Overview

You'll learn how DBMSs execute queries, how to find out about database statistics, when to create indexes and how to investigate whether an index is being used.

Purpose

Gain practical experience of some performance-related matters in databases.

Task

Download the Ubuntu virtual machine from Canvas and open it in the VMWare Player. Follow the instructions to open two connections to the MySQL server from the MySQL Workbench. Work through the tasks below.

Time

This task should be completed in your lab class and submitted for feedback in lab 10 or at the beginning of lab 11.

Resources

- Online module (from Canvas)

- MySQL (on FDM virtual machine) and MySQL Workbench (or other RDBMS and suitable client).

Feedback

Discuss your solutions with the tutorial instructor.

Next

Get started on module 12.

Credit Tasks 10.2 — Submission Details and Assessment Criteria

Document your solutions to the tasks using a word processor. Upload your solutions to Doubtfire, then discuss it with your tutor.

Getting Started

Open the VMWare Player and start the virtual machine. The password for fdm is admin. Click on the application icon (the uppermost icon on the left. Find MySQL Workbench (type the name into the search field if it isn't among the icons).

Open the MySQL Workbench. Click on the SalesOrdersConnection that shows in the connections list. The password is root. In the right upper corner, there are three buttons that let you minimise the left, lower and right subwindow. Use them to get more space for the SQL editor in the middle.
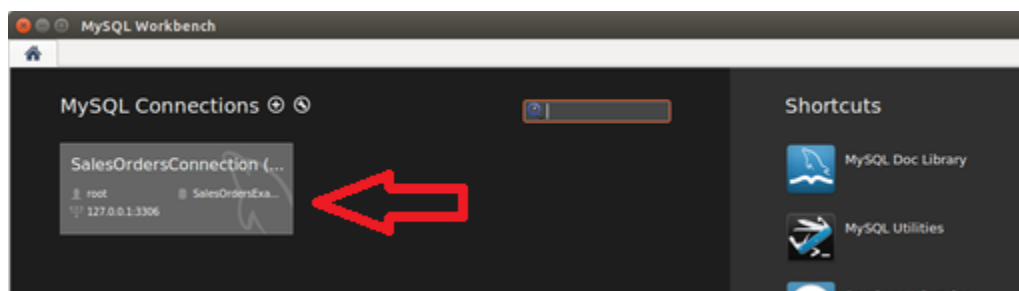
This is the schema of the SalesOrdersExample database:
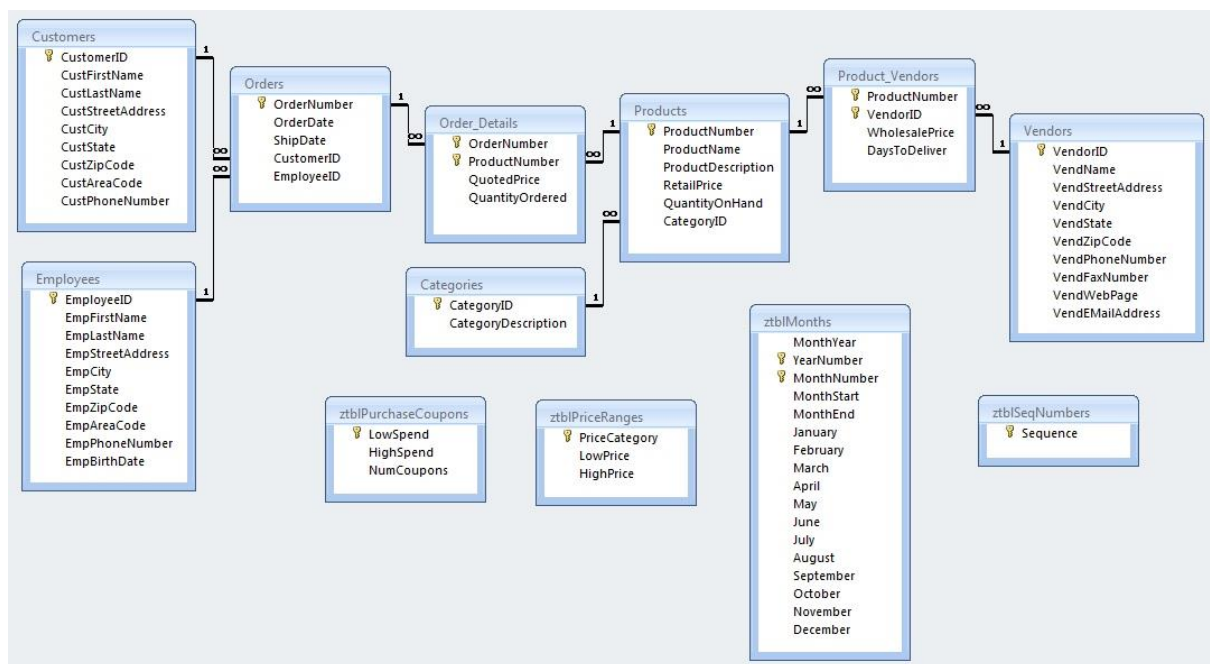
To run a single query in the editor, put the cursor in the line of the query and press the second lightening-shaped button (shown in the picture).
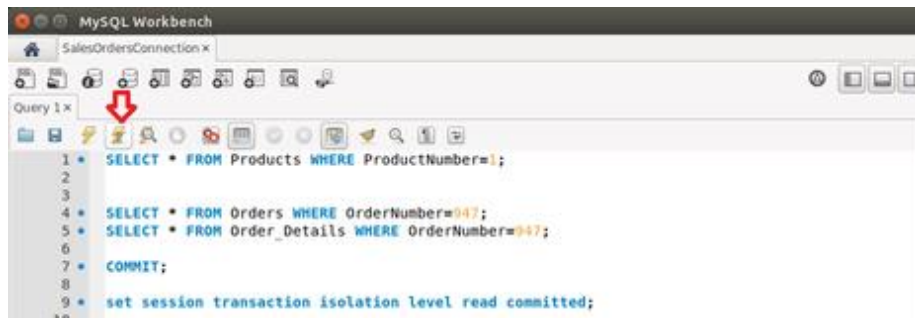
> **Note**: If you have made many changes to the database, you may want to recreate it by running the scripts SalesOrdersStructureMy.SQL and SalesOrdersDataMy.SQL. They are in the Documents/week10 folder.

Use either window and run a simple query:

EXPLAIN EXTENDED SELECT * FROM Products;

You will receive a listing that includes the following columns:

Column	What the values mean
ID	ID field for the step
select_type	The type of query, such as SIMPLE (no joins) or UNION. See listing: https://dev.mysql.com/doc/refman/5.7/en/explain-output.html#explain_select_type
table	The table accessed to get the rows.
type	The type of join used. Most common options: **all** – This usually means that the table has been searched sequentially, which is expensive (making an index might help). **ref** – The DBMS uses an index on the key column to find the matching rows. Listing of all options: https://dev.mysql.com/doc/refman/5.7/en/explain-output.html#jointype_const
possible_keys	The names of indexes that could be used. To find out which columns the indexes are made on, use SHOW indexes in <table_name>;
key	The index that was actually used for the search.
key_len	The length of the key that was used.

ref	Shows which columns or values were compared when the index mentioned under 'key' was used.
rows	Shows how many rows the DBMS believes it has to examine. This is an estimate based on statistics, not actual values.
filtered	The percentage of the table rows that will be examined based on the condition in the WHERE clause.
Extra	Any additional information.

Subtask 10.2.1

Run the following JOIN of two tables which has two restrictions. Examine the output – is this what you expect? Prefix the query with "EXPLAIN EXTENDED" and review the output.

SQL
| ```
EXPLAIN EXTENDED
SELECT * FROM Orders NATURAL JOIN Order_Details WHERE
QuotedPrice > 1000 AND OrderDate BETWEEN '2012-10-01'
AND '2012-10-31';
``` |

Examine the query plan output with the help of the column explanations given above.

Describe in your own words how the DBMS is fetching the rows. For each table, does it look through the data rows or access them through an index? How does this influence the number of rows examined? According to the output, which table was accessed first? Why do you think the DBMS decided to access this table first instead of the other one (both have restrictions)?

If you need to find out which columns are accessed through an index, run

SHOW indexes in Order_Details;

Document your findings and upload them.