

1DV013 – Database Theory

Practical Assignment 2 (30 p)

Deadline: October 12, 2014

Read in the movie data from the previous assignment into your database (use only the means of the database). It can be done in 2 steps:

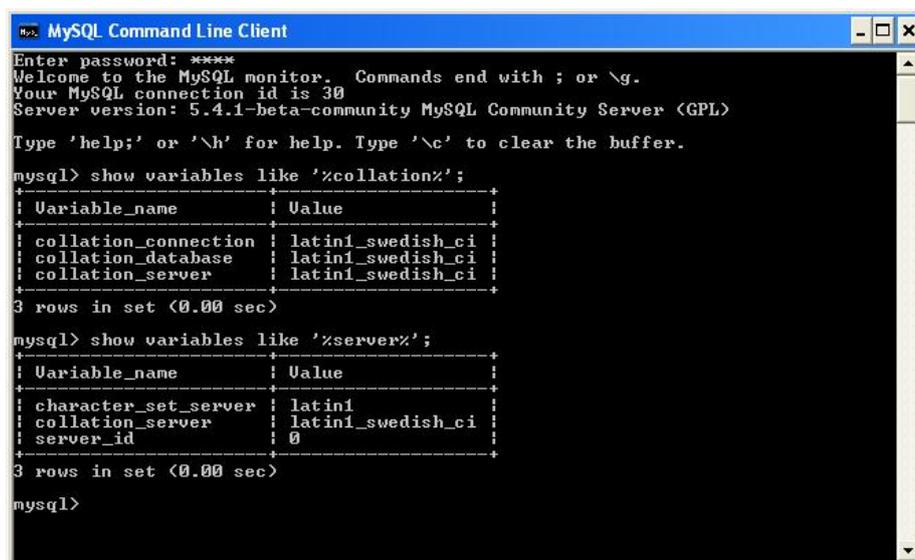
1. Load data into database using temporary created tables. For this, use the [Load Data](#) command;
`mysql> load data infile file_name ignore into table table_name
fields terminated by '\t';`
Read what "ignore" and "replace" commands do.
2. Fit the data into your database schema created before. Use Data Manipulation Language (i.e. *insert*, *delete*, *update*).

If you run into performance problems, e.g. your data is too big to fit into a table by simple insert query, try to insert data in the process of creating table, e.g.

```
create table table_name select * from table_name2;
```

Don't forget to assign primary keys for tables, it speeds up *join* operation between tables.

If you run into encoding problems in Windows while loading the data, make sure that your language for non-Unicode programs (Control Panel -> Regional and Language options -> Advanced) corresponds to your database settings. For example, if your regional language is Swedish, then your database setting should look like this:



```
MySQL Command Line Client
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 30
Server version: 5.4.1-beta-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show variables like '%collation%';
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| collation_connection | latin1_swedish_ci |
| collation_database  | latin1_swedish_ci |
| collation_server    | latin1_swedish_ci |
+-----+-----+
3 rows in set (0.00 sec)

mysql> show variables like '%server%';
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| character_set_server | latin1                |
| collation_server    | latin1_swedish_ci   |
| server_id         | 0                     |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

Figure 1: Encoding settings

To change them read [MySQL documentation](#).

If you run into other problems – e.g., timeouts – you need to experiment a little, just like in real life.

When reading in the data, follow these rules:

- A movie is defined uniquely by its *name* and *year*. This allows for, e.g., remakes.
- If you find a movie which is not contained in *movies.txt*, then ignore this movie, i.e., ignore entries from various files that reference movies not contained in *movies.txt*.
- In *times.txt*, ignore *episodes* and other non-integer information (e.g., "fps") and set the time of the corresponding movies to 0 (for unknown runtime). If a movie has more than one runtime specified, take the first one that you find in the file. For example, "Boy Who Turned Yellow, The 1972" has two values specified in *times.txt*, so take "17" here (you can do it by using *ignore* in the *load data* command).

Describe the number of rows in every table in your DB (after loading the data) in your report.

Create SQL statements for the following queries:

1. What year did *Kasiping* air?
2. In how many movies has *Abel, Alfred* starred?
3. In which movies that *Aaronson, Evan* had produced did *Beers, Thom* star in?
4. In how many horror movies have people with last name *Clark* acted?
5. Which movies have a play time of 3 1/2 hours or longer?
6. How many movies have been produced, directed, and written by one and the same person?
7. What year was the maximum number of movies released?
8. How many movies did only one actor star in?

You are expected to work with the design described in your PA1. If you have decided to make any changes to your design, explain them and attach an updated SQL script that creates the database. Hand in **all of your** SQL statements that construct and populate the database, and the SQL queries **together with** their results (include the results into your report). Note: if you are asked about quantities, you should answer with numbers and not lists of rows. If you provide SQL output as screenshots, make sure they contain sufficient data.

Your submission must include a PDF report and SQL script files. Assignment must be submitted as a single ZIP archive via the Moodle submission system. Email submission attempts will be ignored.