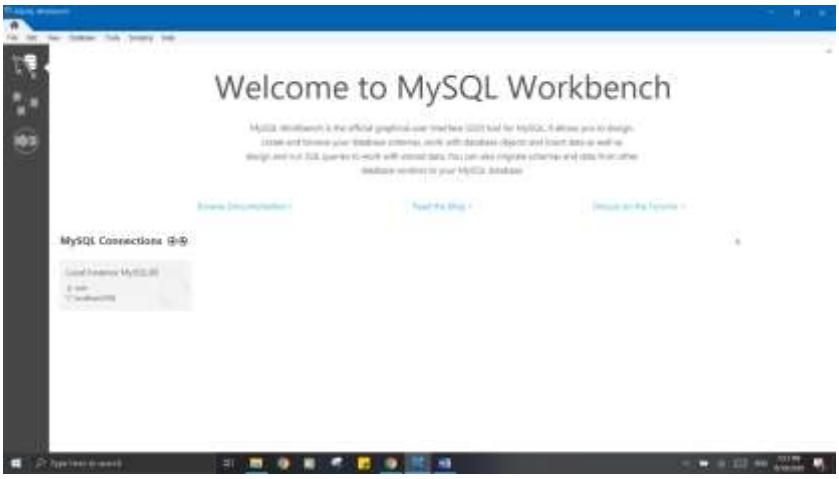
Name: Kunal Kamlesh Mehta
UBIT Name: kunalkam
UB Person Number: 50352960

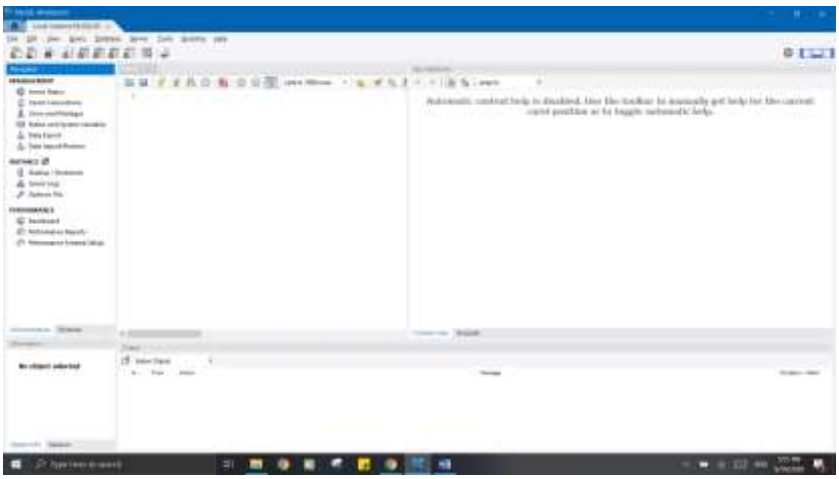
Programming Assignment 0

1. Setting up MySQL Server/Client
    a. Installation Complete. Server Running and visible in Workbench:

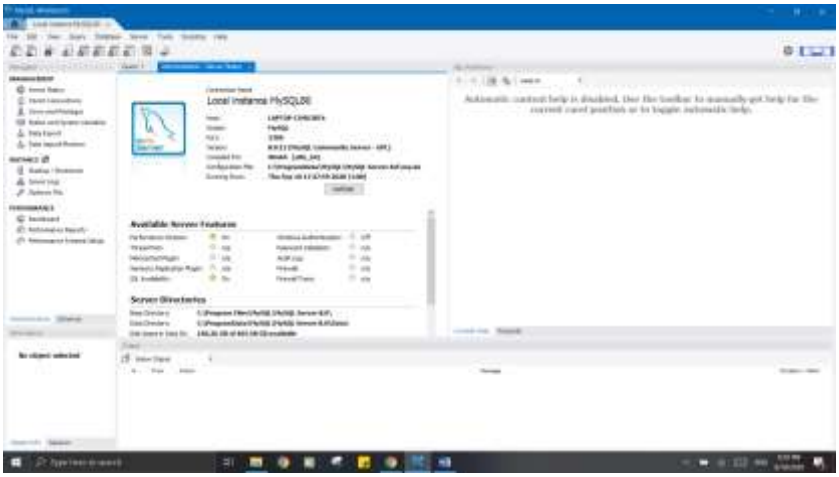


    b. Password entry:

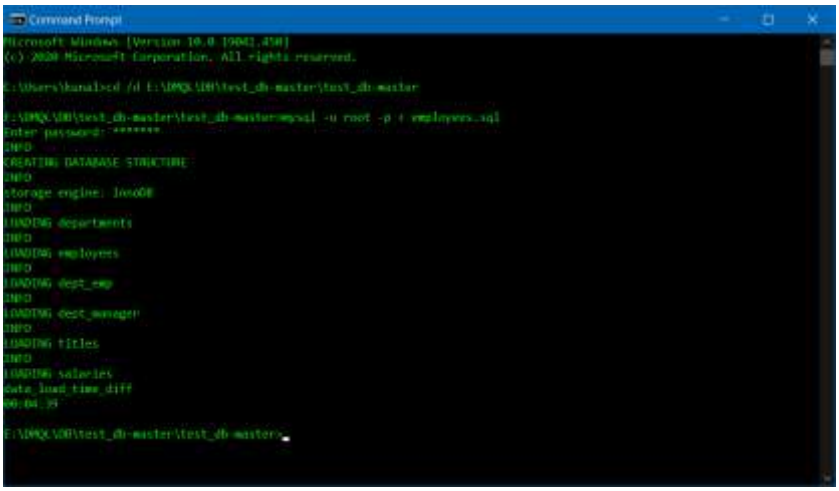
    c. Connection complete:



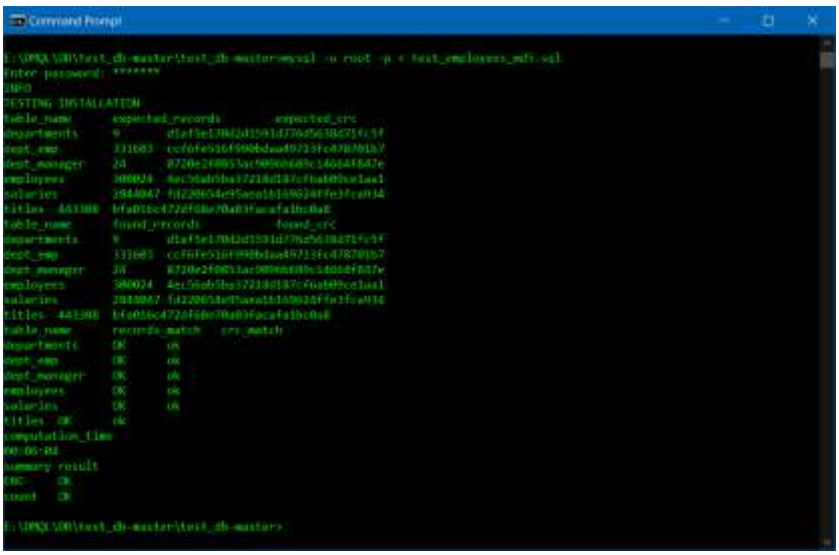
Name: Kunal Kamlesh Mehta
UBIT Name: kunalkam
UB Person Number: 50352960



2. Setting up a database:
    a. Step 3 result:

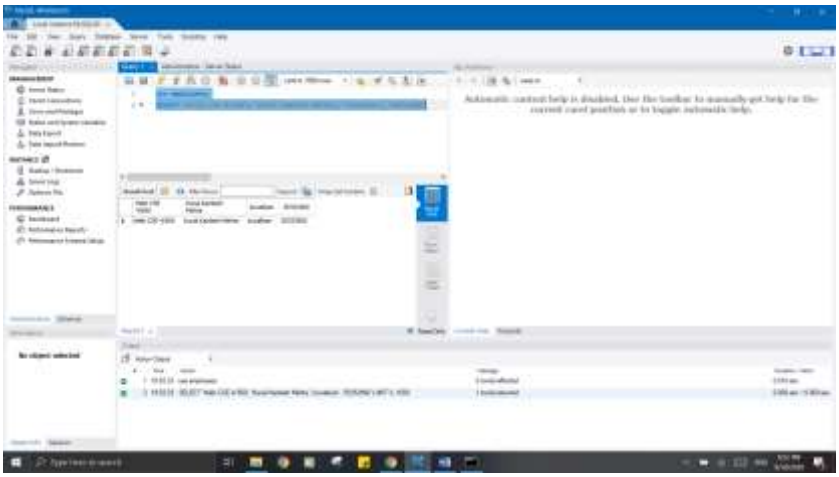


    b. Step 4 result:



Name: Kunal Kamlesh Mehta
UBIT Name: kunalkam
UB Person Number: 50352960

c. Step 5 result:



Explanation:

- The command in part 3 of step 2 invokes the ‘employees.sql’ script and executes the commands in it. These commands, upon inspection, create the database named ‘employees’, taking care to first check if any such named database exists and dropping if it does. Even within the database, it takes care to drop existing tables with the same names as that of the ones that are to be created, after which it proceeds to create the required tables like ‘employees’, ‘departments’ and so on with the specifications required (data type, not null, primary keys) and views like ‘dept_emp_latest_date’. The tail end of the script displays the messages seen on the command line prompt before importing the dump files.
  In essence, this command ran a script that created the database ‘employees’, and in that database the tables and views required were created, after which the dumps were imported using the source command. The dumps must be large in size since the creation of the database didn’t take much time yet the total time was more than 4 and a half minutes and the placement of the ‘LOADING xyz’ statements made it easy to observe what was taking more time.
  The last command simply shows the time elapsed since the command was run.
- The command in part 4 of step 2 invokes the ‘test_employees_md5.sql’ script and executes the commands therein. On viewing the script, it is clear that the script is a check on whether the values in the tables are as desired or not. To that end, there are two tables created as ‘expected_values’ and ‘found_values’ after making sure there aren’t any existing with the same name and discarding them if they do. The table names explain the intent of the script, and the expected values table has some values inserted into it, which upon observing seem to be each of the table names, the expected number of records in them and then the expected hash values which suggests that this is a cyclic redundancy check with the MD5 message-digest hash algorithm being used. There is also a table ‘tchecksum’ made to hold the checksum value. For each of the tables created by the previous script, the checksum value is first calculated and inserted into tchecksum, after which the found_values table is updated with the name of the table, number of records found in it and the crc check values. At the end, the ‘expected_values’ and ‘found_values’ tables are compared, as we also see in the output. If these are identical, the database passes the check and each table is displayed as whether or not it is ok. A summary is displayed at the end detailing whether or not the CRC check and the count of records check was passed by the database.
  In essence, this command ran a script that checked if the values in the database were as required using CRC.

Name: Kunal Kamlesh Mehta
UBIT Name: kunalkam
UB Person Number: 50352960

- The commands in part 5 of step 2 are to be executed in the SQL Client. The first command simply instructs MySQL to use the ‘employees’ database as the database for the following statements. The subsequent statement is a SELECT statement that doesn’t query any table, and therefore simply prints the ‘Hello CSE 4/560’, ‘Kunal Kamlesh Mehta’ and subsequent words, with the comma separator indicating each different value. The final display is in the form of a single-record table, with the values being column names as well as the observations.