CP5804 Database Systems

Week 4: Lab activities

Assignment Practice: Creating a Database Model using MySQL Workbench

Through the last weeks' lab activities you learned the general procedure working on MySQL Workbench - from creating a conceptual model in ERD to implement a database physical schema using the forward engineering process.

In this lab, you are going to develop a simple relational database for a small e-book library using MySQL Workbench. Through this lab activity, you will get experience using essential database facilities provided by your DBMS (MySQL Workbench) to create a relational database.

The experience you gain through this lab activity will directly help you to complete the final assignment of this subject.

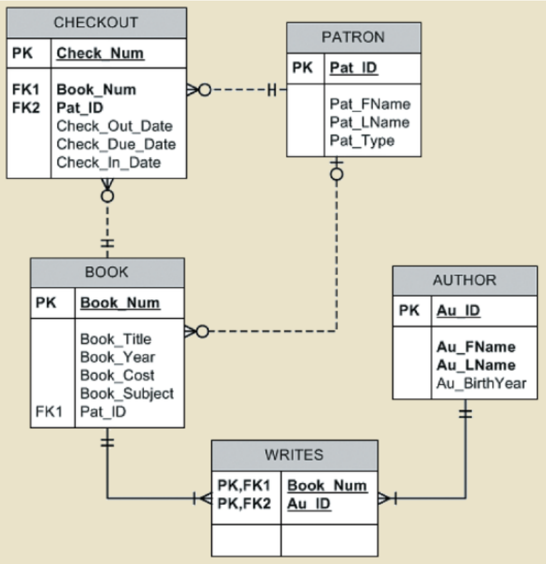
- **Learning outcomes and objectives**
  Student will be able to:
  - create a relational database for a given conceptual model (ERD) using MySQL Workbench
  - import raw data from the external file to a table using MySQL Workbench facility
  - use basic SQL for data administration (to create tables) and for data manipulation (to add, modify, delete, and retrieve data)

- **Files provided**
  - author.csv
  - book.csv
  - checkout.csv
  - patron.csv
  - writes.csv
  - fact_mysql.txt

- **Task Overview**

Use MySQL Workbench to create a relational database consisting of five tables as shown in the ERD provided.

This is a simple ERD for a small imaginary library which manages a collection of online books for use by a group of patrons. Patrons are allowed to check out a book and the library also keeps the data about authors of each book.
Note: To simplify determining which patron currently has a given book checked out, a redundant relationship between BOOK and PATRON is maintained.



You have two alternative ways to create this database.
One is using MySQL Workbench facilities to create database schema and tables and to fill the table with data by import facility, without writing/running SQL queries. The other way is using SQL queries in MySQL Workbench to create tables and insert all necessary data to the table created. For both experiences for you, this lab consists of two main tasks as summarised below.

- Task 1: Create a database using useful database facilities provided by MySQL Workbench: forward engineering, importing data from external files etc.
  - Step 1: Create an ER diagram (logical model) on MySQL Workbench.
  - Step 2: Apply forward engineering process to construct the physical schema based on the logical model created in Step 1.
  - Step 3: Import a raw data (provided as a 'comma-separated values' file or .csv) to each table.

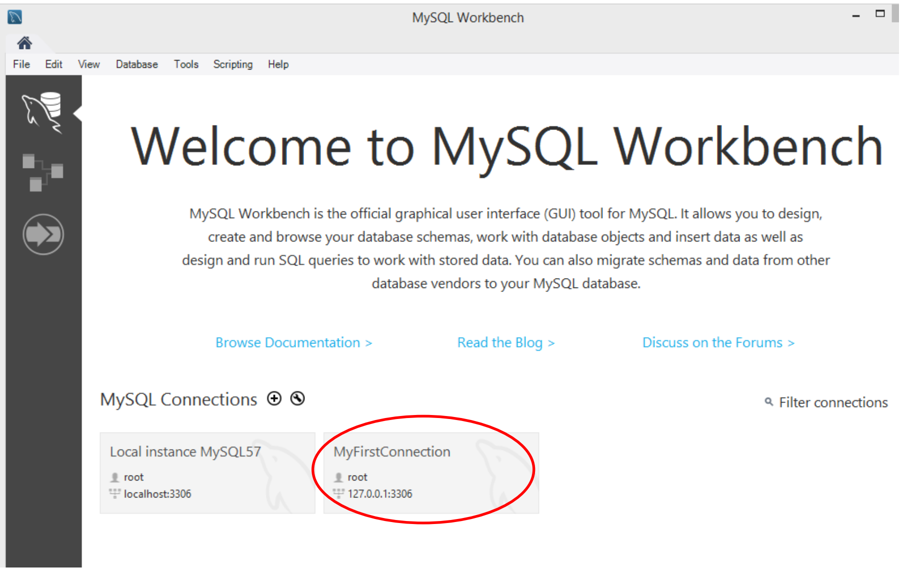
- Task 2: Create a database using SQL commands on MySQL Workbench
  - Step 1: Create an initial (empty) database schema having no tables contained.
  - Step 2: Write and Run SQL queries (via SQL query editor provided by MySQL Workbench) to create each table structure
  - Step 3: Write and Run SQL queries to insert data to each table
  - Step 4: Apply reverse engineering process to create the ERD for the database model created (You can do this step after Step 2)

[Task 1]

Create a database using useful database facilities provided by MySQL Workbench

Step 1: Create an ERD on MySQL Workbench

1. Run MySQL Workbench and make the connection appropriately.



Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

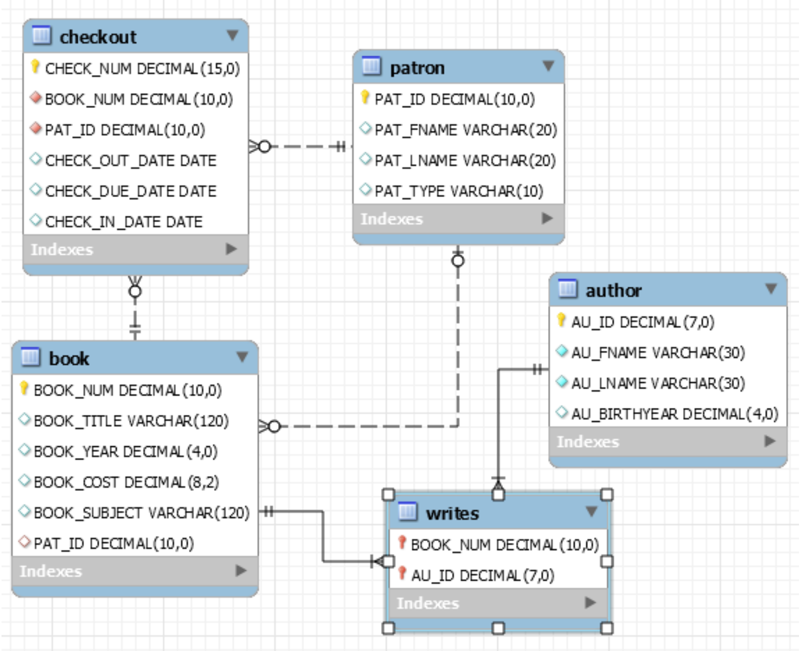
Browse Documentation > Read the Blog > Discuss on the Forums >

MySQL Connections ⊕ ⊘ Filter connections

Local instance MySQL57
root
localhost:3306

MyFirstConnection
root
127.0.0.1:3306

2. Select ‘New Model’ under the File menu (on the top menu bar). This will create a new tab called ‘MySQL Model*’
3. Double-click ‘Add Diagram’ on the ‘Model Overview’ panel. This will create a new tab called ‘EER Diagram’
4. On ‘EER Diagram’ tab, create an ERD having all necessary components required set properly. The final ERD should look like the image as shown below.



checkout
- CHECK_NUM DECIMAL(15,0)
- BOOK_NUM DECIMAL(10,0)
- PAT_ID DECIMAL(10,0)
- CHECK_OUT_DATE DATE
- CHECK_DUE_DATE DATE
- CHECK_IN_DATE DATE

patron
- PAT_ID DECIMAL(10,0)
- PAT_FNAME VARCHAR(20)
- PAT_LNAME VARCHAR(20)
- PAT_TYPE VARCHAR(10)

author
- AU_ID DECIMAL(7,0)
- AU_FNAME VARCHAR(30)
- AU_LNAME VARCHAR(30)
- AU_BIRTHYEAR DECIMAL(4,0)

book
- BOOK_NUM DECIMAL(10,0)
- BOOK_TITLE VARCHAR(120)
- BOOK_YEAR DECIMAL(4,0)
- BOOK_COST DECIMAL(8,2)
- BOOK_SUBJECT VARCHAR(120)
- PAT_ID DECIMAL(10,0)

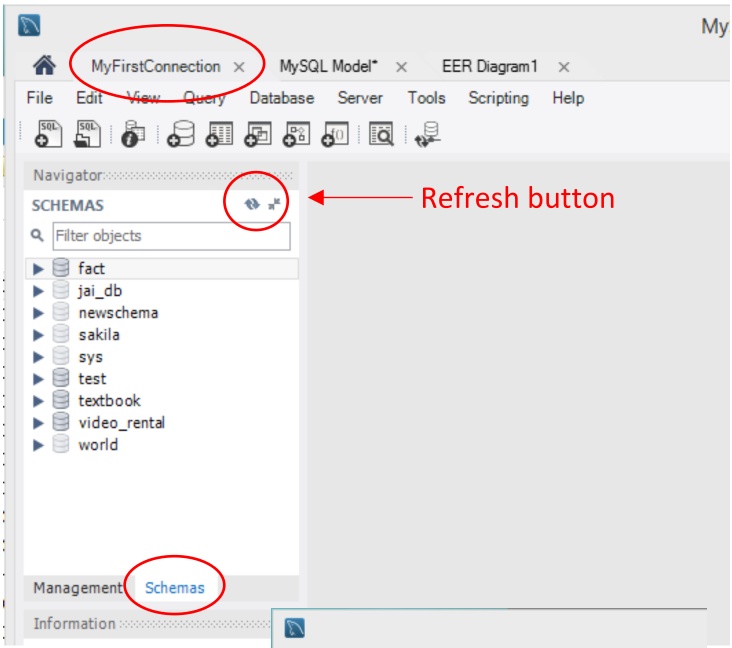
writes
- BOOK_NUM DECIMAL(10,0)
- AU_ID DECIMAL(7,0)

5. Rename your schema as you want.

6. Save this ERD model as a file named 'fact.mwb' and submit the file to be marked off.

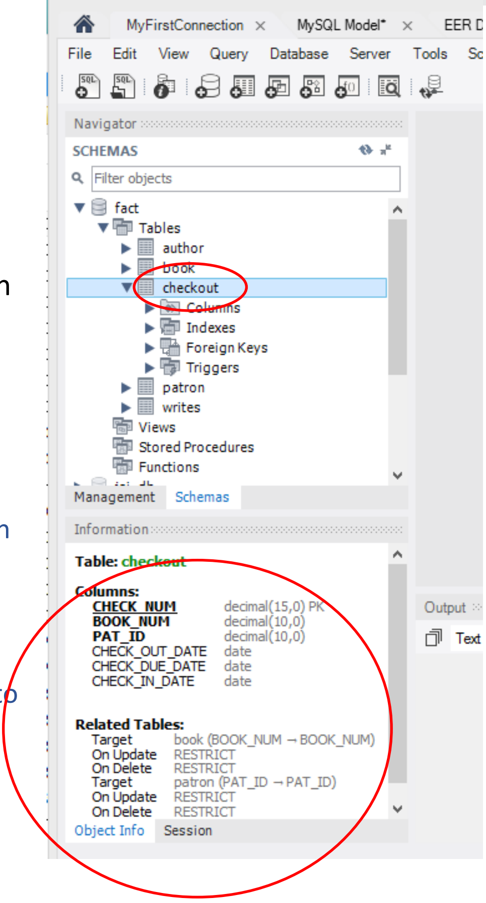
Step 2: Apply forward engineering process to construct the physical schema based on the logical model created in Step 1.

1. Go to Database menu and click 'Forward engineer …'
2. Proceed to complete the forward engineering process (refer to previous week's lab)
3. Go to the connection tab (usually 'MyFirstConnection') and look on the SCHEMAS navigator panel to see if there is a new database schema you created. (click the refresh button to update the list of SCHEMAS if needed)



4. Open the database you created (by clicking the database name on SCHEMAS panel) and check if all tables are created correctly. Currently the table has no actual data contents but has only columns (attributes) set in proper data types. You can check each table's structure in details through corresponding 'Information' panel.

Note: You are assumed to create and set all columns properly (setting data type and properties as requested) when you create the ERD in Step 1. However, if you find you need to modify some column properties (e.g. change column name or data type), you can change it in several different ways: 1)You can revisit EER Diagram tab and modify any property as wanted and then apply forward engineering process to create a new database schema or 2)You can create a SQL query to alter the table structure.

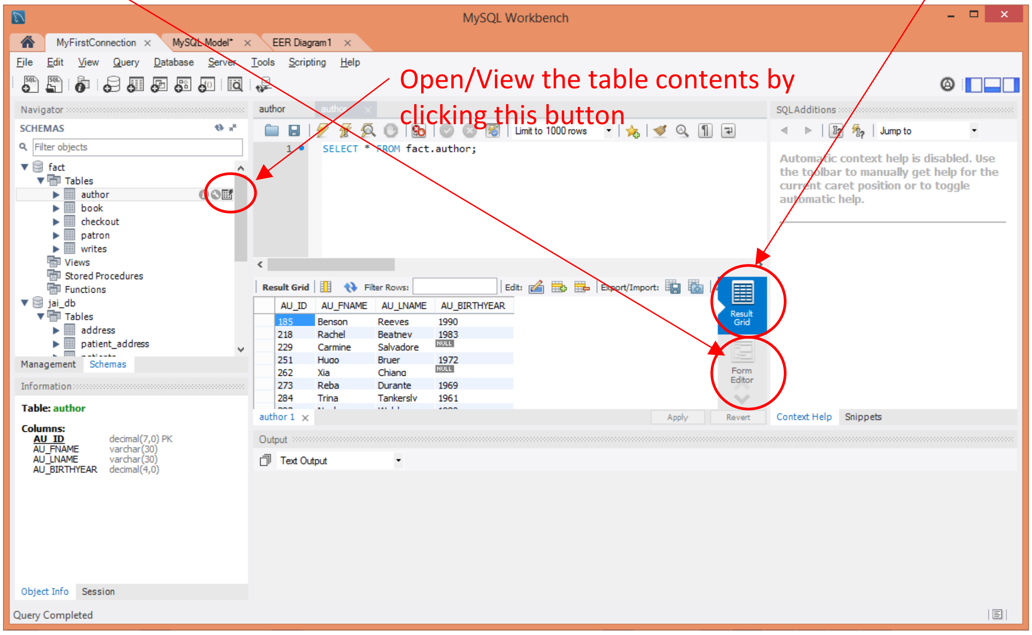


```
Table: checkout

Columns:
   CHECK_NUM         decimal(15,0) PK
   BOOK_NUM          decimal(10,0)
   PAT_ID            decimal(10,0)
   CHECK_OUT_DATE    date
   CHECK_DUE_DATE    date
   CHECK_IN_DATE     date


Related Tables:
   Target      book (BOOK_NUM → BOOK_NUM)
   On Update   RESTRICT
   On Delete   RESTRICT
   Target      patron (PAT_ID → PAT_ID)
   On Update   RESTRICT
   On Delete   RESTRICT
```

Step 3: Import a raw data (provided as a ‘comma-delimited’ text file) to each table

There are various ways to enter data to a table.

(1) Open/View the table contents and enter data records directly using ‘Result Grid’ or ‘Form Editor’ facilities.



Open/View the table contents by clicking this button

```
SELECT * FROM fact.author;
```

AU_ID	AU_FNAME	AU_LNAME	AU_BIRTHYEAR
185	Benson	Reeves	1990
218	Rachel	Beatnev	1983
229	Carmine	Salvadore	NULL
251	Huqo	Bruer	1972
262	Xia	Chiang	NULL
273	Reba	Durante	1969
284	Trina	Tankersly	1961

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Table: author

Columns:

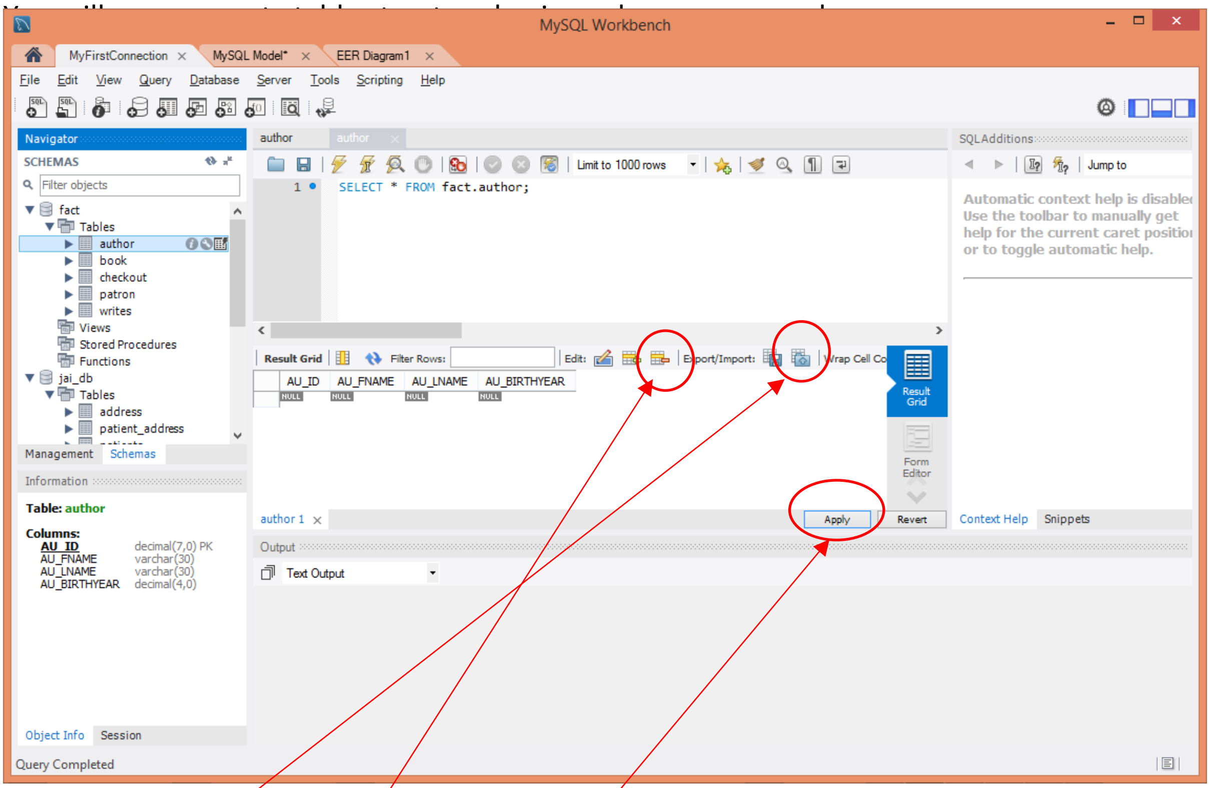
AU_ID	decimal(7,0) PK
AU_FNAME	varchar(30)
AU_LNAME	varchar(30)
AU_BIRTHYEAR	decimal(4,0)

(2) Write/Run a SQL query to insert data to a table

(3) Use ‘Import’ facility to import data from an external file

Among these three ways, let’s practice the third way here to import data from a given CSV file. MySQL Workbench allows to import a file (.csv) where each field data in a record is delimited by ‘comma’. For this lab, you are provided five CSV files for five tables respectively. Please download them and save in your computer before starting this task.

1. Select the author table from the Schemas navigator and click the table-viewer icon.



```
SELECT * FROM fact.author;
```

AU_ID	AU_FNAME	AU_LNAME	AU_BIRTHYEAR
NULL	NULL	NULL	NULL

Table: author

Columns:

AU_ID	decimal(7,0) PK
AU_FNAME	varchar(30)
AU_LNAME	varchar(30)
AU_BIRTHYEAR	decimal(4,0)

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

2. Click the button to import records from an external file.
3. Locate the file 'author.csv' in your computer and select the file to open and import. **In the latest version of Workbench, change the selection to use the currently existing table.**
4. You will see all records from author.csv are imported to the author table. You will find the first record is filled with the header row (having column names) of the original file. Delete this unnecessary record by selecting the row and clicking the 'delete selected rows' button.

   Note: Also, check if the current records imported contain any empty data cell. In order to make sure your DBMS treats the empty cell as a pure 'null', please select the empty cell and delete it (on MySQL Workbench). You will find the empty cell changes into null which indicates it contains now really nothing (null). **This will apply if you use the blanks dataset rather than the nulls dataset.**

5. Click 'Apply' to complete the process of inserting records automatically by MySQL Workbench SQL generator. Consequently, MySQL generates a series of SQL queries to insert the imported data and execute.
6. Apply the same process for other four tables, in this order: patron, book, checkout, writes to import data from corresponding external files (book.csv, checkout.csv, patron.csv, writes.csv)

[Task 2]

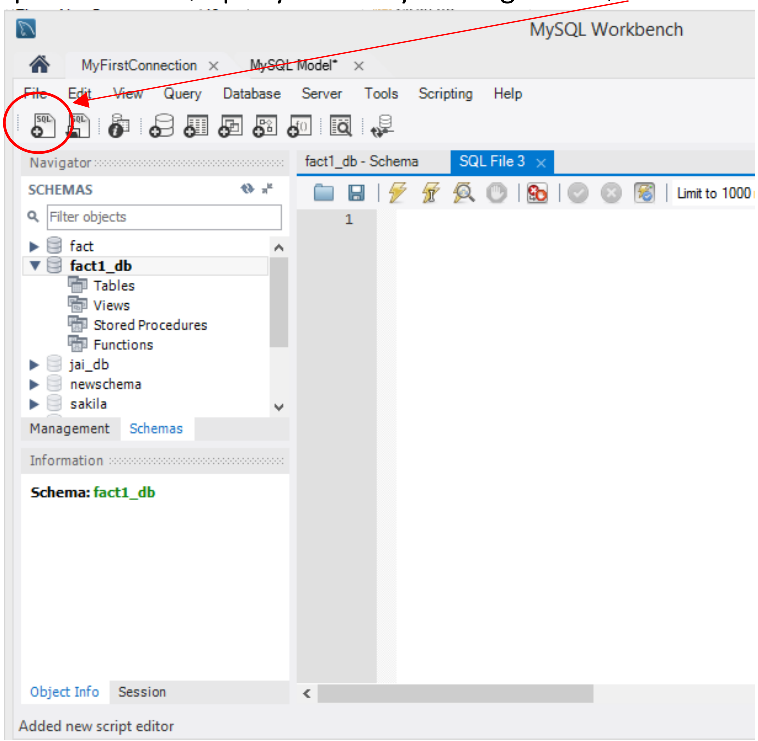
Create a database using SQL commands on MySQL Workbench

As an alternative way, we can use SQL to create database and table structures, perform basic data management chores (insert, update, and delete). Follow the process as instructed below to create the library database using SQL commands.

1. Start the new connection on MySQL Workbench
2. On the SCHEMAS Navigator panel, put the mouse over the list of schemas and right-click on the mouse, then select "Create Schema ...." menu.
3. Rename the schema into your own new library database title (e.g. 'fact1_db'), and click "Apply". This will process to create a new (empty) database.
4. Check through SCHEMAS Navigator panel to find that the current list of schemas includes your new database now. (If needed, refresh the list by clicking the refresh button)
5. Double click on the new database (fact1_db) and you will find that the database contains currently no components like tables

Now, it is time to create table structures. Each table structure will be created using one SQL query each.

6. Open a new SQL query editor by clicking the SQL+ icon on the overhead menu bar.

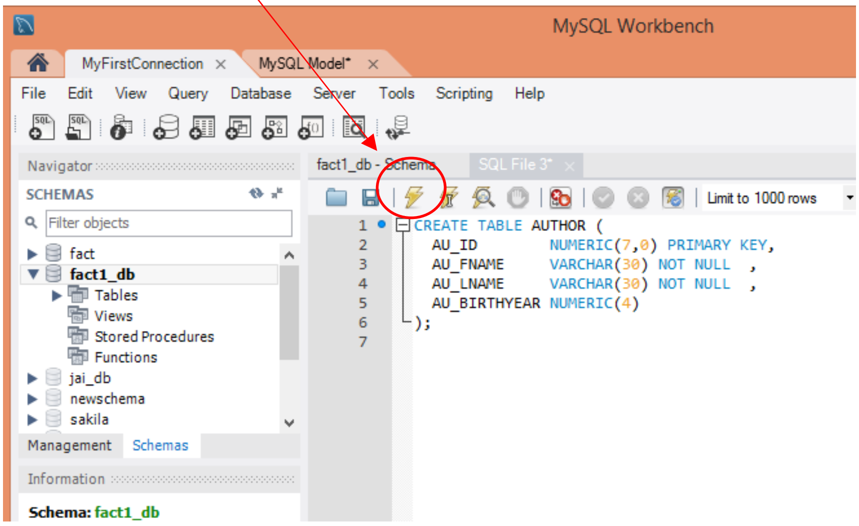


Schema: fact1_db

Added new script editor

7. Write the following SQL query to create the author table. After writing the query, click ‘execute’ button to execute the SQL code.

```
CREATE TABLE AUTHOR (
  AU_ID         NUMERIC(7,0) PRIMARY KEY,
  AU_FNAME      VARCHAR(30) NOT NULL  ,
  AU_LNAME      VARCHAR(30) NOT NULL  ,
  AU_BIRTHYEAR NUMERIC(4)
);
```



```
CREATE TABLE AUTHOR (
  AU_ID        NUMERIC(7,0) PRIMARY KEY,
  AU_FNAME     VARCHAR(30) NOT NULL  ,
  AU_LNAME     VARCHAR(30) NOT NULL  ,
  AU_BIRTHYEAR NUMERIC(4)
);
```

After running this query, you will find (through the navigator panel) that the fact1_db contains one new table called author. (refresh the list if needed)

(You can close the current query editor for your convenience, but it is optional to save the individual query or not. At this stage, you do not need to save, but you will need to save the query if needed for your assignment☺)

8. Create the second table patron by running the following SQL query.

```
CREATE TABLE PATRON (
  PAT_ID       NUMERIC(10,0) PRIMARY KEY,
  PAT_FNAME    VARCHAR(20) NOT NULL    ,
  PAT_LNAME    VARCHAR(20) NOT NULL    ,
  PAT_TYPE     VARCHAR(10) NOT NULL
);
```

9. You can apply the same way to create the other three tables by running each SQL query. However, you run a series of queries by running once. For this, you need to add BEGIN; command at the start and add COMMIT; at the end. The whole code to create three tables (book, patron, writes) at one time is shown below.

```
BEGIN;

CREATE TABLE PATRON (
  PAT_ID       NUMERIC(10,0) PRIMARY KEY,
  PAT_FNAME   VARCHAR(20) NOT NULL   ,
  PAT_LNAME   VARCHAR(20) NOT NULL   ,
  PAT_TYPE    VARCHAR(10) NOT NULL
);

CREATE TABLE BOOK (
  BOOK_NUM   NUMERIC(10,0) PRIMARY KEY,
  BOOK_TITLE VARCHAR(120) NOT NULL  ,
  BOOK_YEAR  NUMERIC(4)               ,
  BOOK_COST  NUMERIC(8,2)             ,
  BOOK_SUBJECT VARCHAR(120)         ,
  PAT_ID    NUMERIC(10),
  FOREIGN KEY(PAT_ID) REFERENCES PATRON(PAT_ID)
);

CREATE TABLE CHECKOUT (
  CHECK_NUM         NUMERIC(15) PRIMARY KEY,
  BOOK_NUM          NUMERIC(10),
  PAT_ID            NUMERIC(10),
  CHECK_OUT_DATE    DATE,
  CHECK_DUE_DATE    DATE,
  CHECK_IN_DATE     DATE,
  FOREIGN KEY (BOOK_NUM) REFERENCES BOOK(BOOK_NUM),
  FOREIGN KEY (PAT_ID) REFERENCES PATRON(PAT_ID)
);

CREATE TABLE WRITES (
  BOOK_NUM  NUMERIC(10),
  AU_ID     NUMERIC(7),
  CONSTRAINT WRITES_BOOK_AU_PK PRIMARY KEY (BOOK_NUM,
AU_ID),
  CONSTRAINT WRITES_BOOK_NUM_FK FOREIGN KEY(BOOK_NUM)
REFERENCES BOOK(BOOK_NUM),
  CONSTRAINT WRITES_AU_ID_FK FOREIGN KEY(AU_ID)
REFERENCES AUTHOR(AU_ID)
);

COMMIT;
```

Now your database (fact1_db) currently has table structures with no actual record (data). At this stage, you may import table data directly from external files as learned in the previous task. For this task, let’s try to use SQL code to insert each data into the existing table one by one (or by series of SQL codes as a whole).

10. Create a new query to insert the first record of author table.

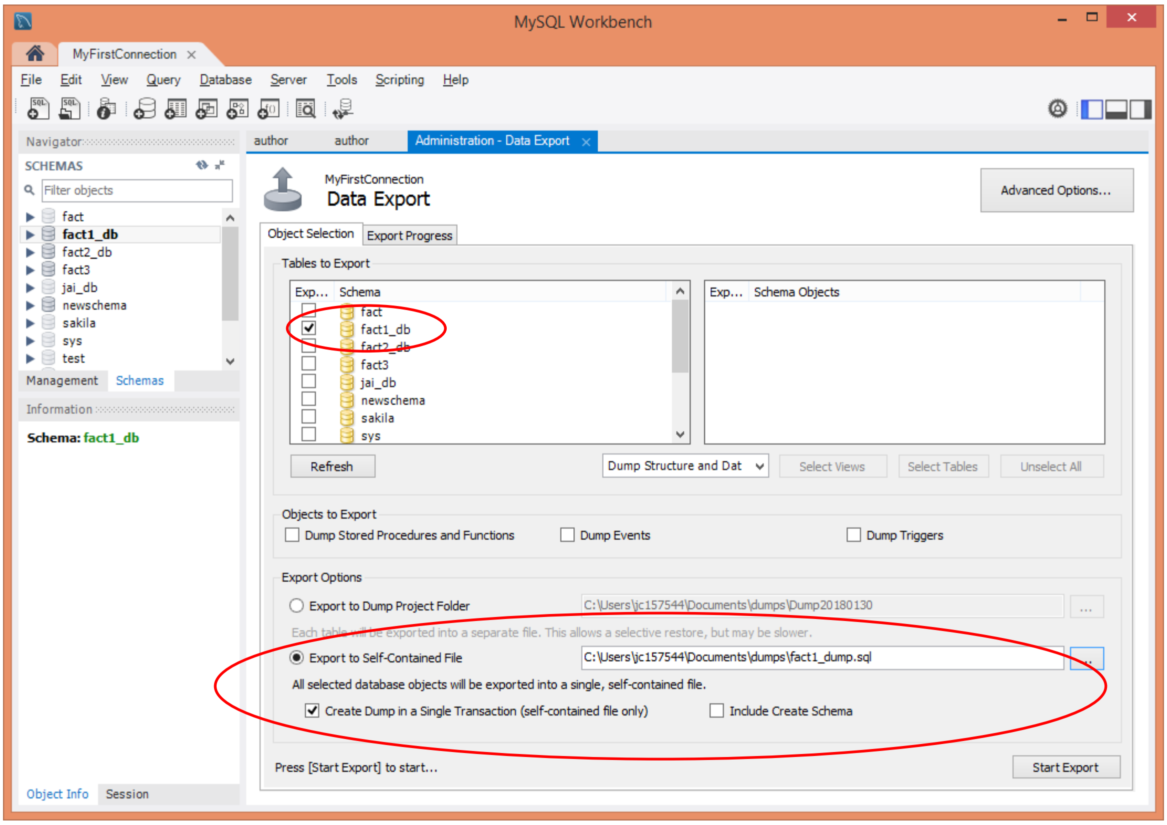
    ```
    INSERT INTO AUTHOR VALUES (185, 'Benson', 'Reeves', 1990);
    ```

11. To insert all other records, you need to create and run multiple number of SQL queries. For your convenience, MySQL codes for creating this database are provided. Please download the file ‘fact_mysql.txt’ and use the code appropriately for your

usage. (Hint: you can run the multiple queries at one time using ‘begin;’ and ‘commit;’ command)

12. Fully check (through SCHEMAS navigator) that all tables and data records are correctly imported.
13. Go to Server menu (on the top menu bar) and select ‘Data Export’ menu. This facility enables you to save the database as a self-contained file (.sql) so that you can open the database anywhere when needed in the future.
14. Select the database (fact1_db) to export. Select “Export to Self-Contained File” option under Export Options section. Also, click “create Dump in a Single Transaction” option. Do not forget to set the location and file name (e.g. fact1_dump.sql) where you want to save.

15. Click ‘Start Export’ button. (Click ‘Continue Anyway’ if you pop-up with a warning message like “version mismatch …”)



Schema: fact1_db

MyFirstConnection
Data Export

Tables to Export

Exp…	Schema
□	fact
☑	fact1_db
□	fact2_db
□	fact3
□	jai_db
□	newschema
□	sakila
□	sys

Exp…	Schema Objects

Objects to Export

□ Dump Stored Procedures and Functions □ Dump Events □ Dump Triggers

Export Options

○ Export to Dump Project Folder C:\Users\jc157544\Documents\dumps\Dump20180130

Each table will be exported into a separate file. This allows a selective restore, but may be slower.

● Export to Self-Contained File C:\Users\jc157544\Documents\dumps\fact1_dump.sql

All selected database objects will be exported into a single, self-contained file.

☑ Create Dump in a Single Transaction (self-contained file only) □ Include Create Schema

Press [Start Export] to start…

16. Submit this exported dump file (fact1_dump.sql) to be marked off this week’s lab activity

---

This is the end of Week 4 Lab.

You are required to submit the following two files to be marked off:

Task 1: fact.mwb

Task 2: fact1_dump.sql
