CP5804 Database Systems

Week 3: Lab activities

Using MySQL Workbench for Creating a Database Model

- **Learning outcomes and objectives**
  Student will be able to:
  - build a conceptual model into logical model in MySQL Workbench,
  - implement the logical model into physical schema using the "Forward Engineering" process in MySQL Workbench, and
  - apply the "Reverse Engineering" process for documenting a database and to share the data model with others.

- **Pre-requisites**
  You are assumed to have completed previous Lab activities (Week 1 Lab and Week 2 Lab) in order to be familiar with necessary features of MySQL Workbench for creating full version of ERDs. You are also assumed the detailed knowledge of ER notation and ERD. Chapter 3 &4 from Coronel-Morris textbook, which explains relational database models and ER modelling are also required reading.

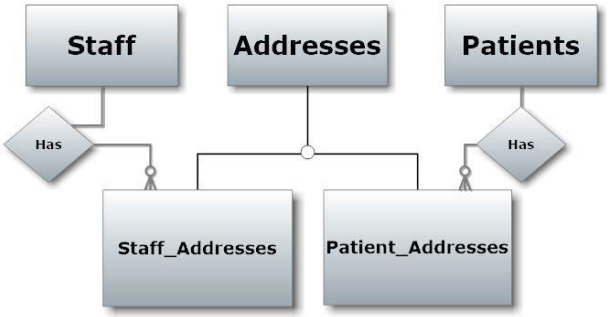
- **Connecting MySQL Workbench to MySQL Server**
  Through the lab activity in Week 1 and Week 2, you are assumed to have MySQL Workbench installed in your computer. For previous lab activities, you were not essentially required to have your MySQL Workbench fully connected to MySQL server. However, to proceed the lab in this week, you need to complete the connection to the MySQL server from MySQL Workbench. Please refer to Week 1 Lab document to complete the connection process.

- **Task Overview**
  Our task is to build the following conceptual model into logical model. We will then implement the logical model into physical schema using the "forward engineer" process in MySQL Workbench. Additionally, we will learn how to apply "reverse engineer" process on MySQL Workbench to get an ER model from an established physical schema.

  The overall task is summarised through the following diagram. As you can see in the diagram, this week lab consists of three tasks:

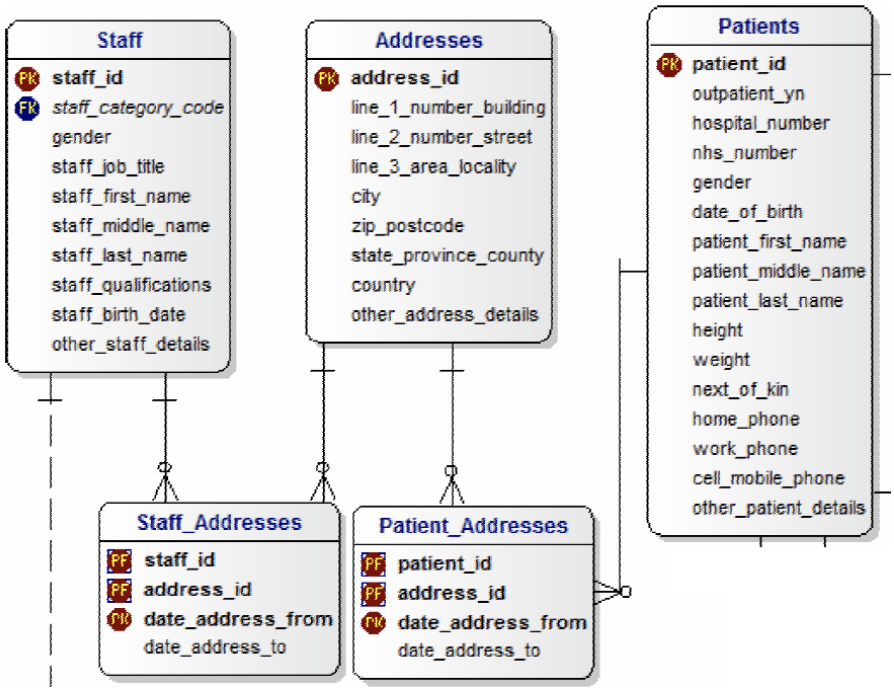
  - Task 1 - Create an ER diagram (logical model) on MySQL Workbench. The ER diagram created in this task will be used for the next task (Task 2 – processing forward engineer)
  - Task 2 - Apply forward engineering process to construct the physical schema based on the logical model created in Task 1.
  - Task 3 - Apply reverse engineering process to create the corresponding ERD using an established database.

Conceptual model



[Task 1]

Logical Model



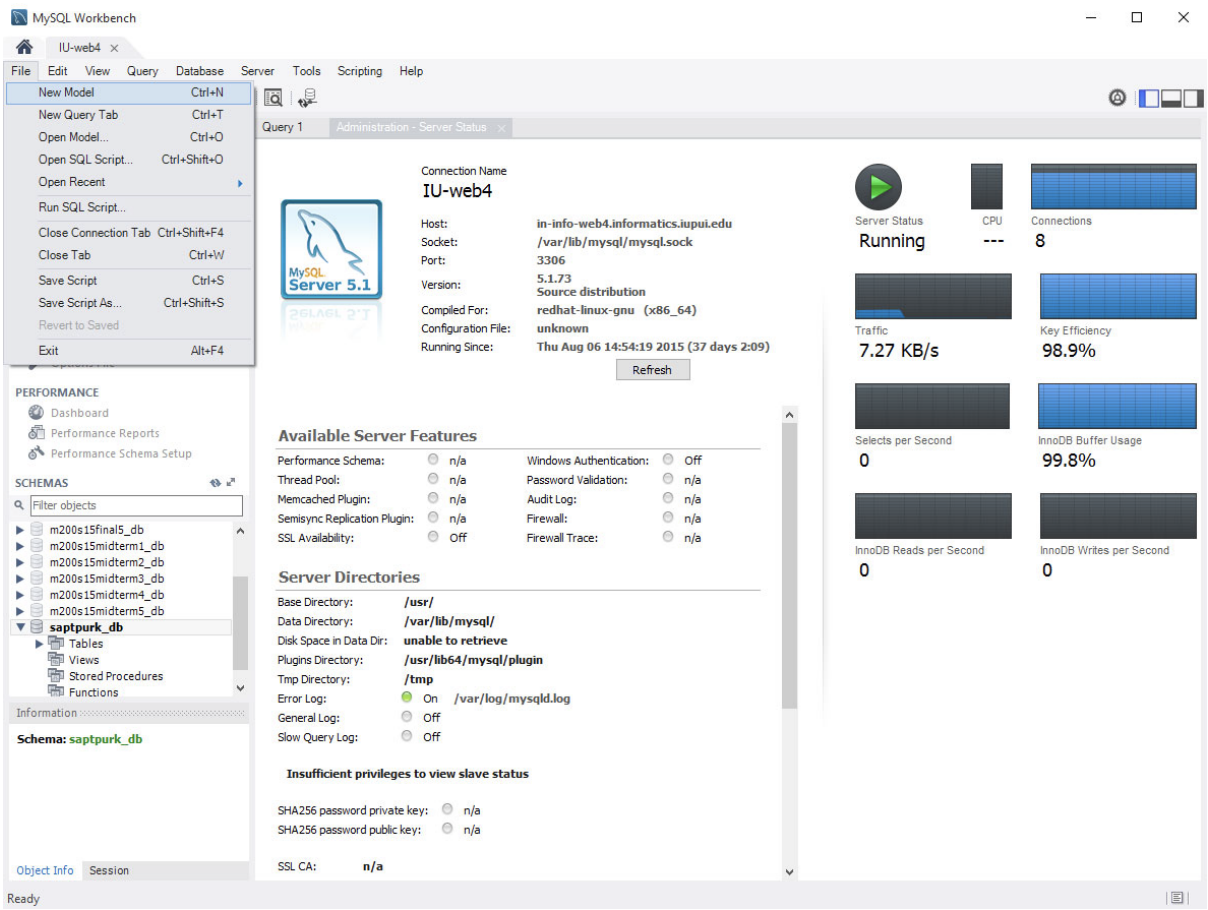
[Task 2] [Task 3]

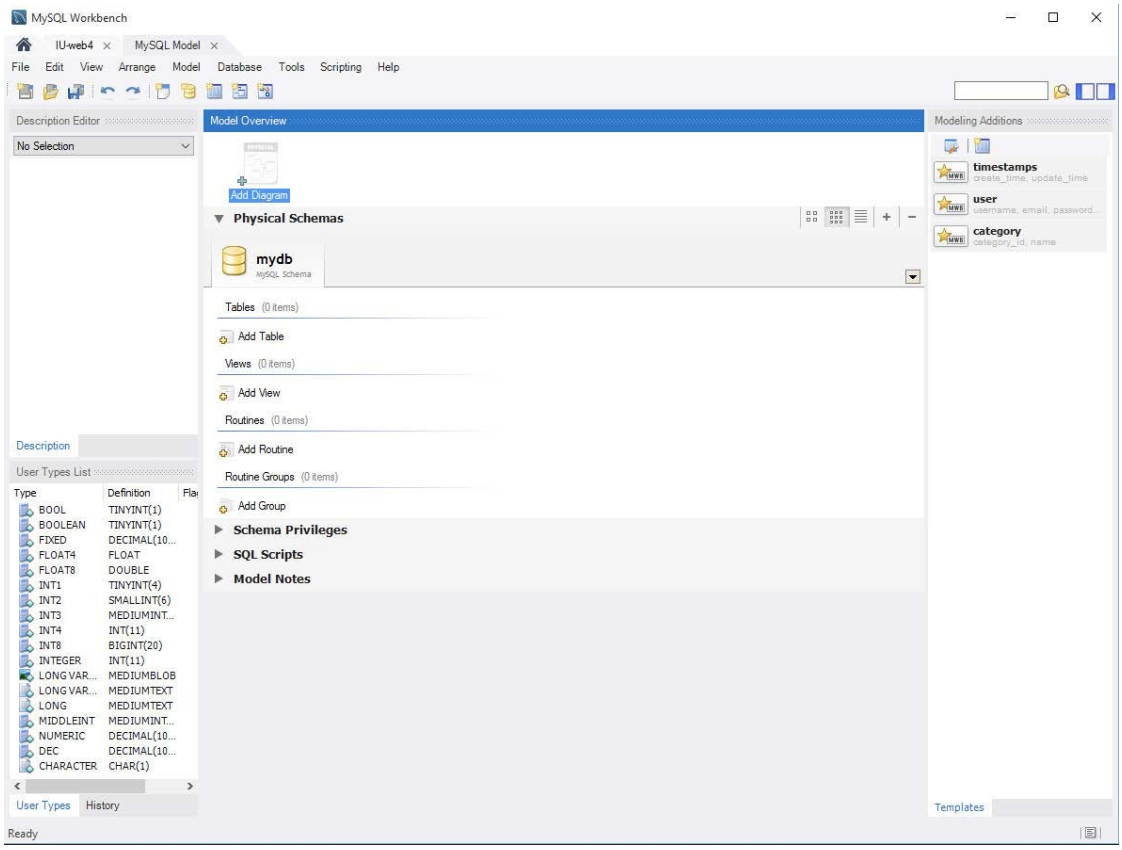
Physical schema
(Physical database model)

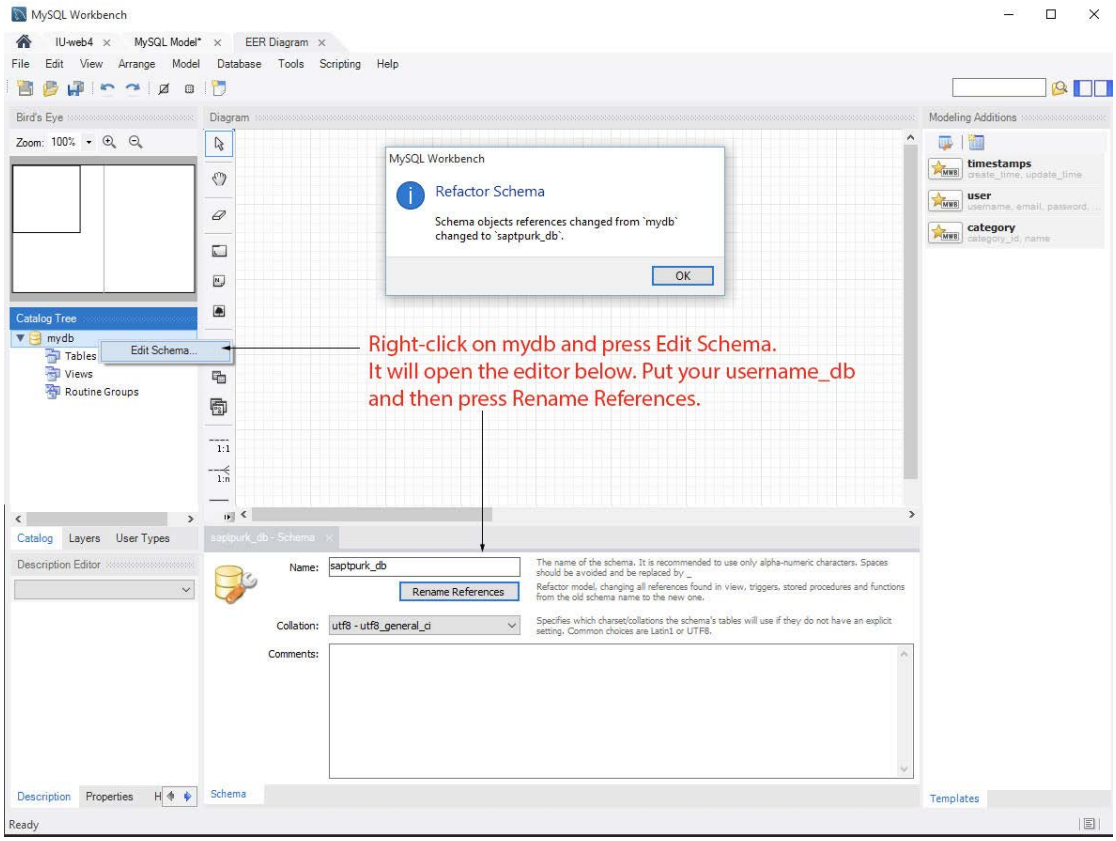
[Task 1]

Creating ER diagram on MySQL Workbench (building the conceptual model into logical model)

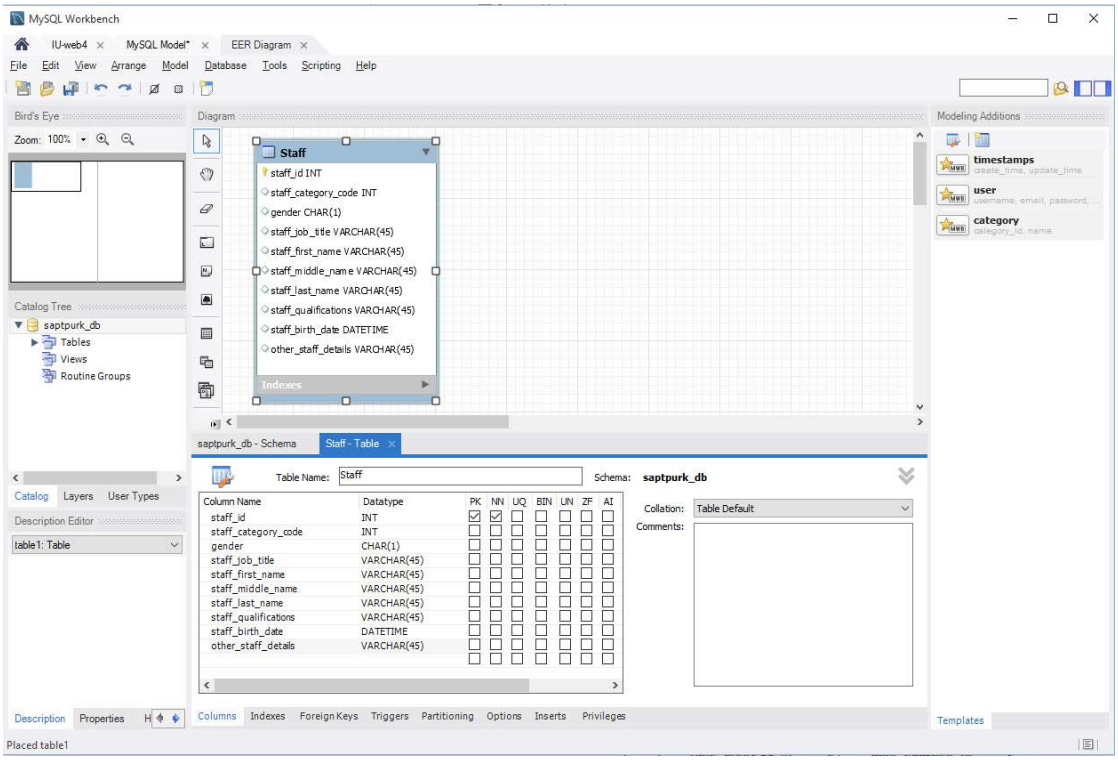
1. Establish a database connection to your local server (most of you may have named the server connection as “MyFirstConnection”. You can click MyFirstConnection generated in your MySQL Workbench.
2. Create an ER diagram. You already learned how to create an ER diagram through the previous lab activities (Week 2 Lab), but here the process is summarised again for your information.
    - Go to File -> New Model or Press Ctrl + N
    - Double-click on Add Diagram and you will be taken to the ER canvas from where you can start creating an ER diagram
    - The drawing canvas for the EER that opened in MySQL workbench is used to draw the table and add the relationships between the tables
    - Rename your schema to match your username_db. For example, if your personal username is joanne, your schema is called joanne_db.




MySQL Workbench
IU-web4
MySQL Model
File Edit View Arrange Model Database Tools Scripting Help
Description Editor
No Selection
Model Overview
Add Diagram
Physical Schemas
mydb
MySQL Schema
Tables (0 items)
Add Table
Views (0 items)
Add View
Routines (0 items)
Add Routine
Routine Groups (0 items)
Add Group
Schema Privileges
SQL Scripts
Model Notes
Modeling Additions
timestamps
user
category
Description
User Types List
Type
Definition
BOOL TINYINT(1)
BOOLEAN TINYINT(1)
FIXED DECIMAL(10...
FLOAT4 FLOAT
FLOAT8 DOUBLE
INT1 TINYINT(4)
INT2 SMALLINT(6)
INT3 MEDIUMINT...
INT4 INT(11)
INT8 BIGINT(20)
INTEGER INT(11)
LONG VAR... MEDIUMBLOB
LONG VAR... MEDIUMTEXT
LONG MEDIUMTEXT
MIDDLEINT MEDIUMINT...
NUMERIC DECIMAL(10...
DEC DECIMAL(10...
CHARACTER CHAR(1)
User Types
History
Templates
Ready


MySQL Workbench
IU-web4
MySQL Model*
EER Diagram
File Edit View Arrange Model Database Tools Scripting Help
Bird's Eye
Zoom: 100%
Diagram
MySQL Workbench
Refactor Schema
Schema objects references changed from `mydb` changed to `saptpurk_db`.
OK
Modeling Additions
timestamps
user
category
Catalog Tree
mydb
Tables
Views
Routine Groups
Edit Schema...
Right-click on mydb and press Edit Schema.
It will open the editor below. Put your username_db
and then press Rename References.
Catalog
Layers
User Types
Description Editor
saptpurk_db - Schema
Name: saptpurk_db
Rename References
The name of the schema. It is recommended to use only alpha-numeric characters. Spaces should be avoided and be replaced by _
Refactor model, changing all references found in view, triggers, stored procedures and functions from the old schema name to the new one.
Collation: utf8 - utf8_general_ci
Specifies which charset/collations the schema's tables will use if they do not have an explicit setting. Common choices are Latin1 or UTF8.
Comments:
Description
Properties
Schema
Templates
Ready

- Add tables to the drawing panel
- Create a new table and change the table name to **Staff** and then add the column names as shown below.
- **staff_id** is the primary key hence the **PK** is checked and **NN** is for NOT NULL
- Also check the **AI** checkmark for AUTO INCREMENT. This is only applicable for primary keys that are INT type.
- The other column names just like the logical model should be added.
- Similarly, add other tables first and then add relationships towards the end.

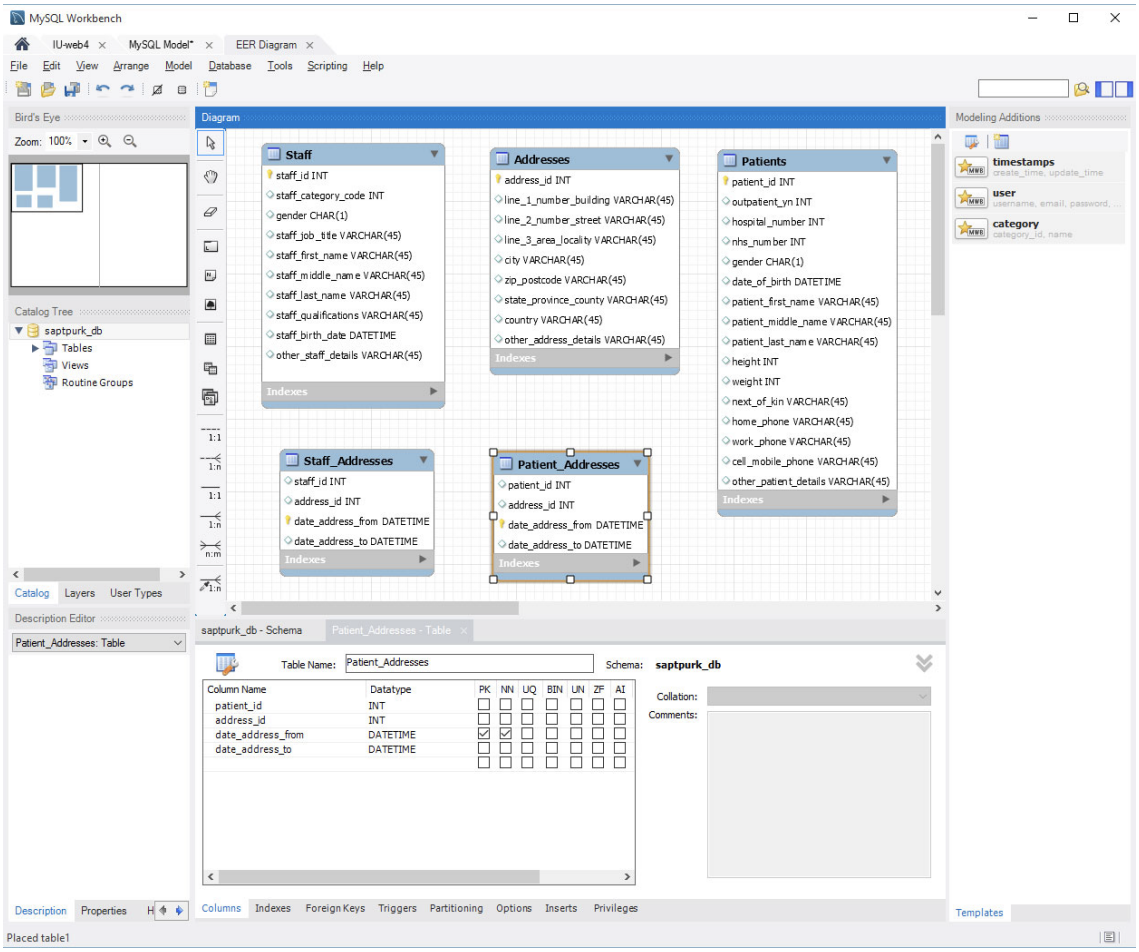


!!! Warning !!!

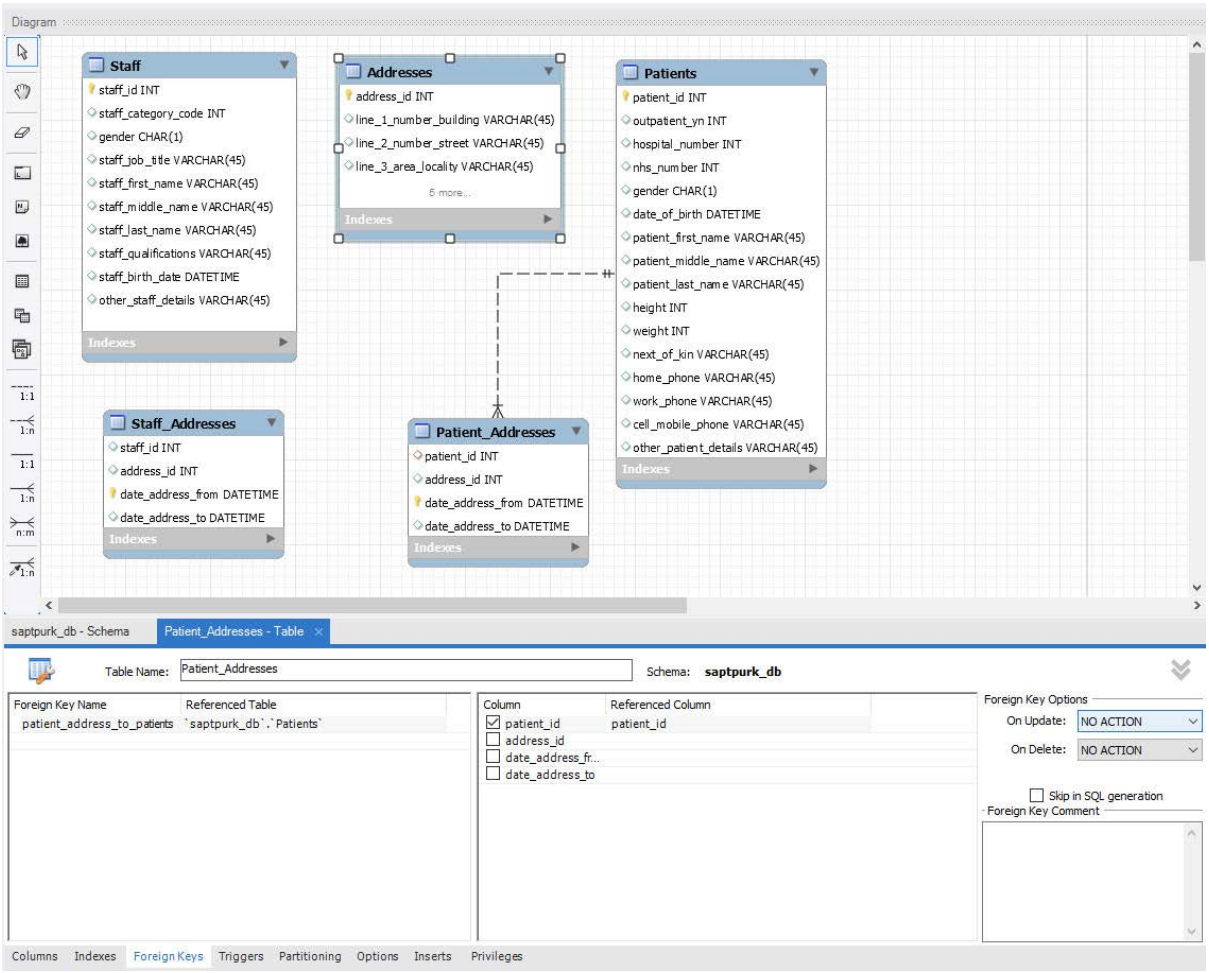
At any stage, frequently save your model in case the system is unexpectedly turned off or corrupted.

(Click File -> Save model and save the mwb file)

- The tables would like this once they have been created in the canvas grid.



- Now it is time to add relationships



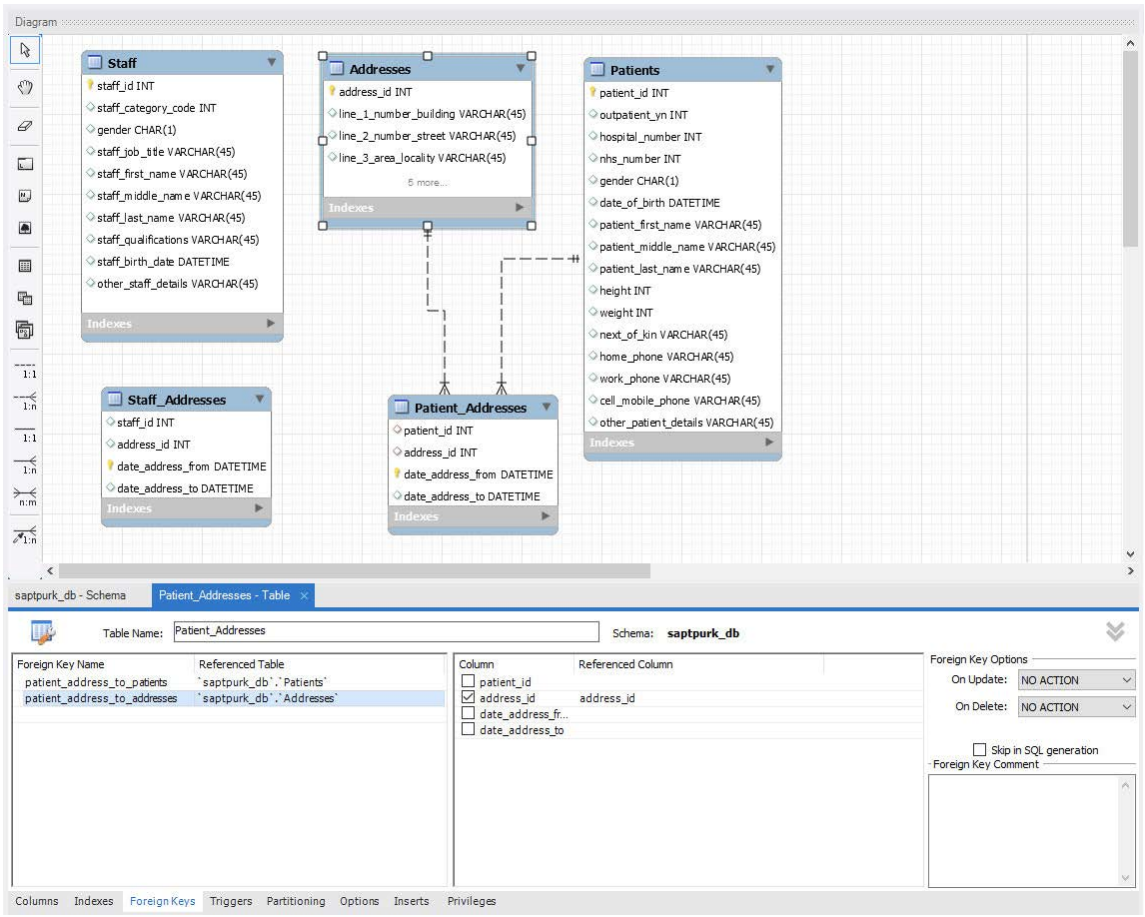
Let's create a 1:M relationship between two tables: Patients and Patient_Addresses.

If the Patient_Addresses does not contain the foreign key attribute, an easy way is to click the 1:n relationship icon from the left panel, the Patient_Addresses table and then the Patients table consecutively. This action will trigger MySQL Workbench to add a foreign key attribute automatically to the Patient_Addresses table by referencing (automatically) to the primary key of the Patients table. (as you exercised this way in Week 2 Lab activity).

If you have already created the foreign key attribute (patient_id) in the Patient_Addresses table manually (as you have done in this prac), you may apply another way of creating a relationship as in the following instructions.

- Double click on the Patient_Addresses table and at the bottom of the screen press the Foreign Keys tab.
- Give it any name – for example as shown in the screenshot above you can name it patient_addresses_to_patients
- The referenced table is the one which has the Primary key column – in this case it's the Patients table. Double click on referenced table and select from the dropdown.
- Check mark the patient_id column and then in the referenced column select patient_id (this is from the Patients table).

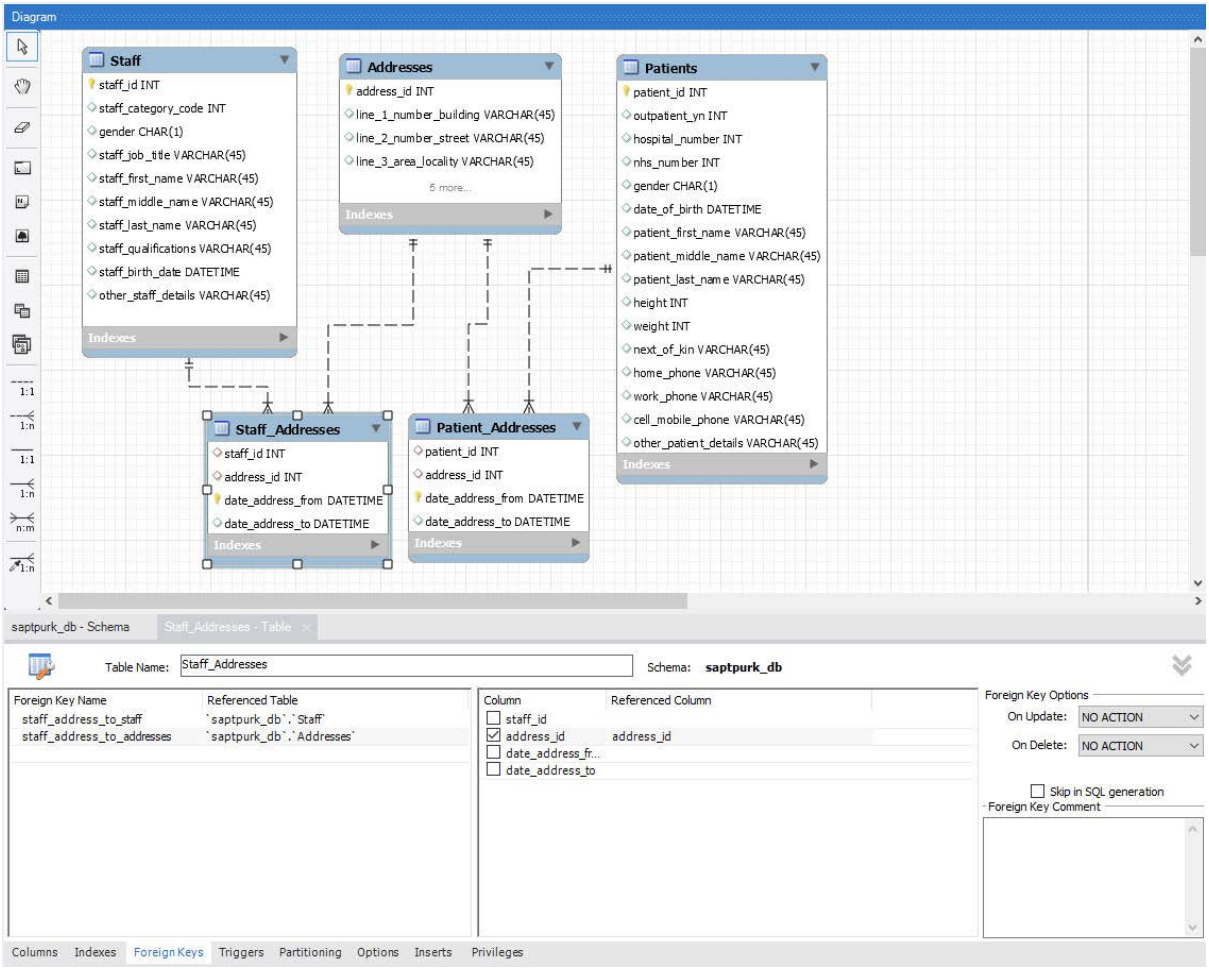
- Look at the On Update: NO ACTION. I won't explain cascading at the moment, but it is important to define foreign keys behaviour. When data is updated or deleted in foreign key tables, how should the values get updated in the primary key tables.



There is second foreign key (address_id) in the Patient_Addresses table. Thus, we need to create another relationship to the Addresses table.

- Give it a name – patient_address_to_addresses
- Select the referenced table as Addresses. Then checkmark theaddress_id and in the referenced column select the address_id
- This will create the 1-to-many relationships between Addresses and Patient_Addresses table

Do the same with the Staff_Addresses table like shown below.



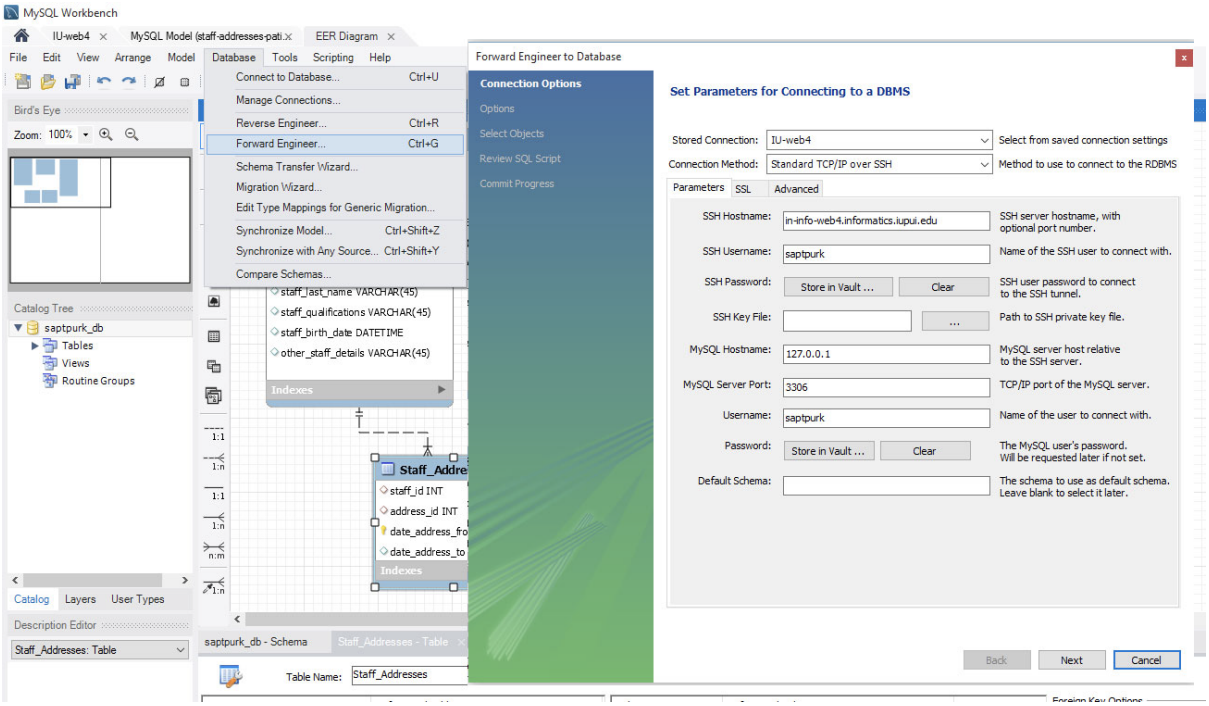
- Click File -> Save model by saving it as a .mwb file.

[Task 2]

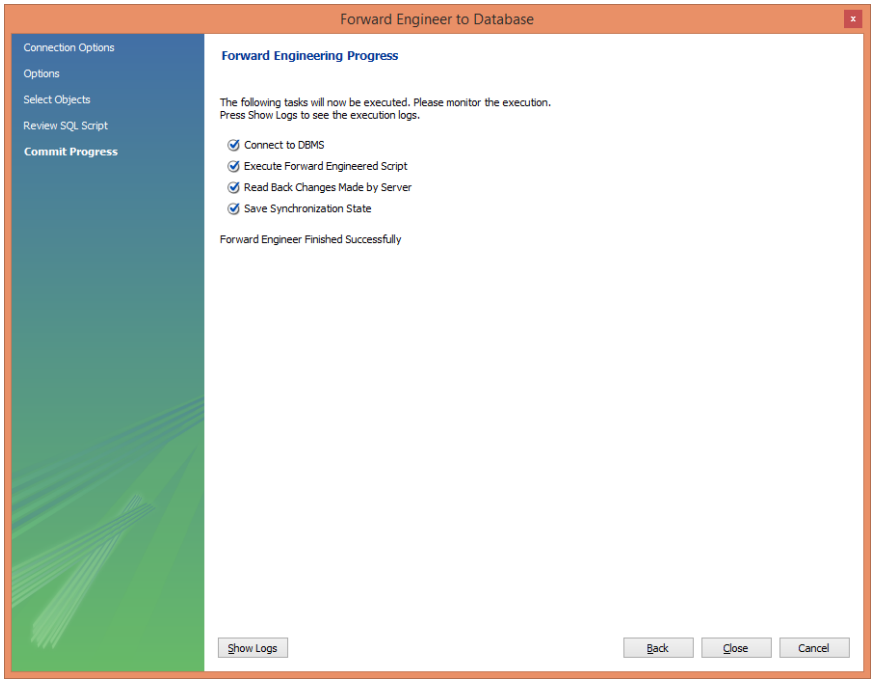
Forward Engineering (implementing the logical model into physical database schema)

Now, we want to implement this logical model into a physical schema using “Forward Engineer”

- Go to Database menu and then Forward Engineer
- Select the Stored Connection to the local server that you created earlier (e.g. MyFirstConnection).
- Press Next, Keep the defaults on the next screen and Press Next again, till you reach the message “Forward Engineer Finished Successfully”.



You'll see the screenshot as below.



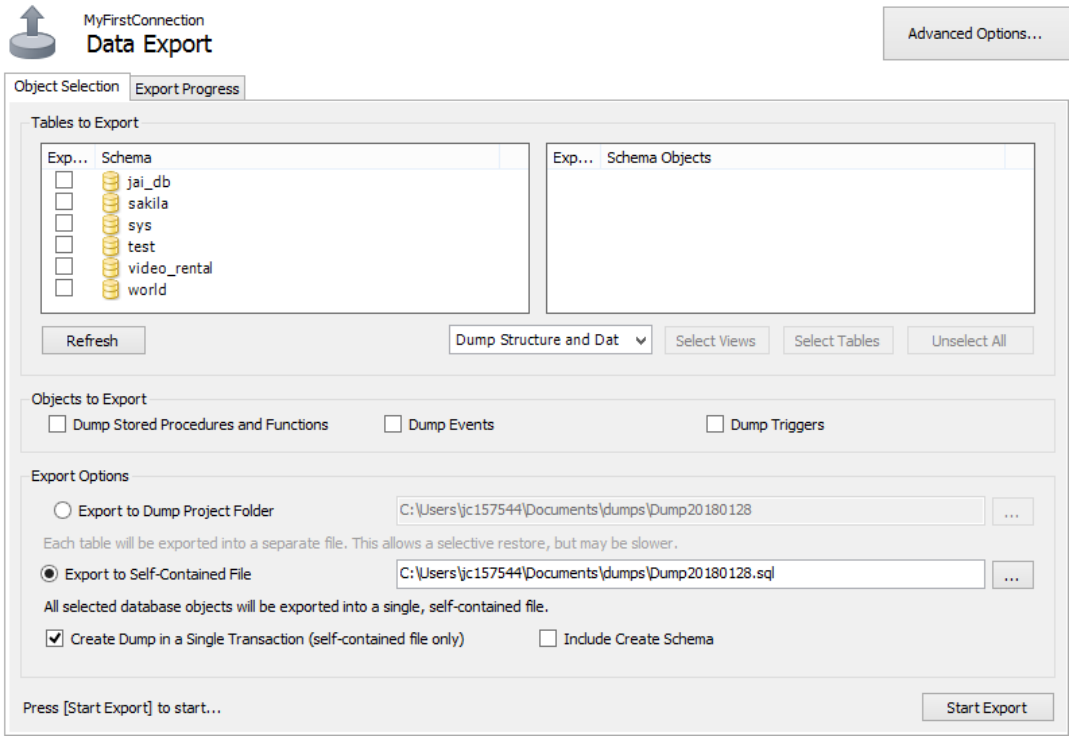
Press Close and continue to the next task (Task 3).

- **Backup your database**

  MySQL Workbench can export a backup of your database to a file on your local computer. This is also sometimes referred to as "data dump"

  We strongly recommend that you create regular backups of your database so it can be easily and quickly recovered in the unlikely event that your data is lost or becomes corrupted.

- Click ‘Data Export’ under ‘Server’ on the top menu bar.
  You will see “Data Export” administration setting as shown in the following screenshot.



- Select your database to export
- Select “Export to Self-Contained File” option and set the location and file name to save the file. This selection will create one single SQL file named XXX.sql
- Then, click Start Export
  You may come up with a warning window saying “mysqldump version mismatch” and you can simply ignore it and click “continue anyway” button.
- This SQL file can be imported anytime or anywhere (someone else’s computer) via MySQL Workbench (‘Data Import’ menu option under the Server menu)

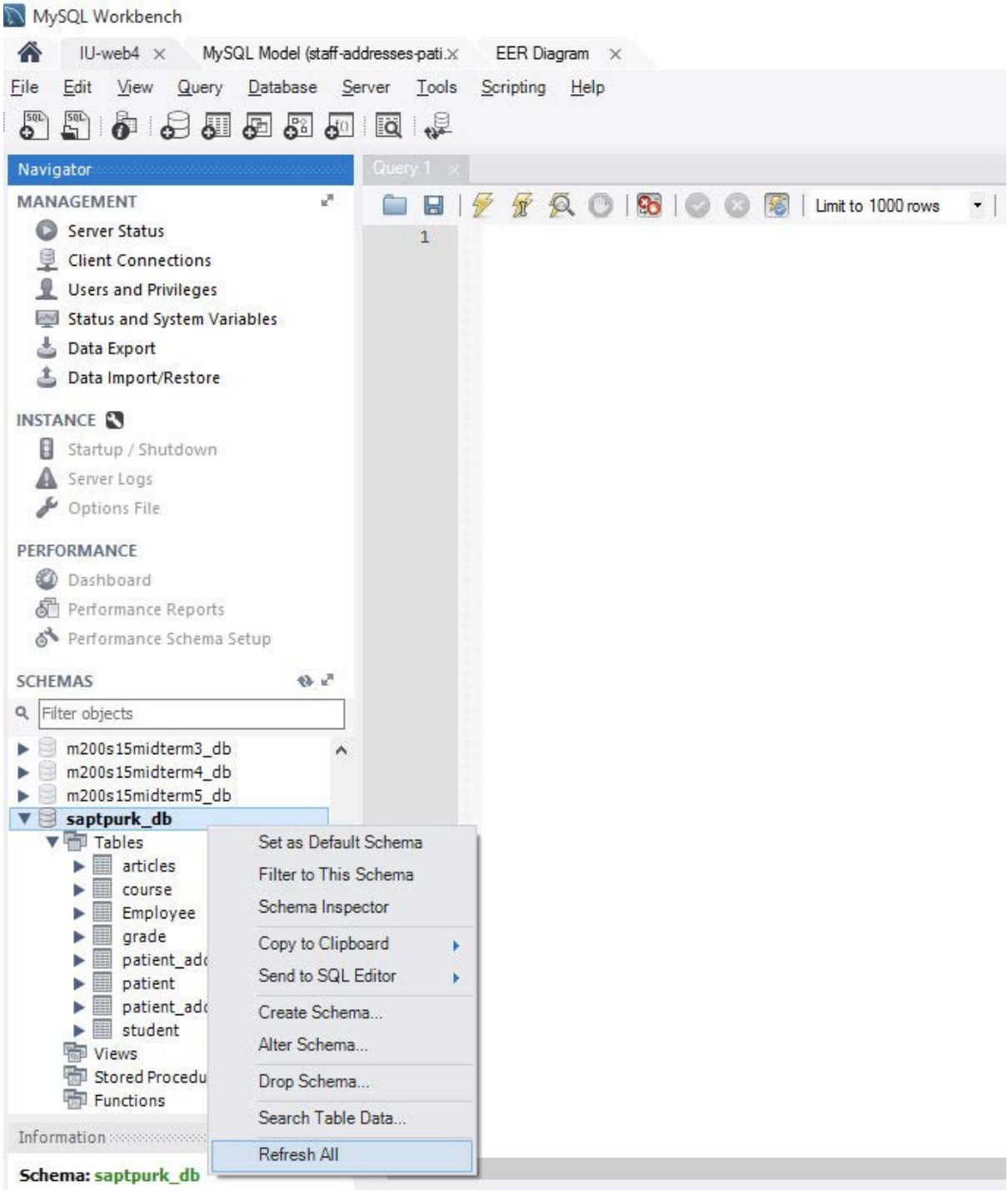
!!! Important Note !!!

In particular if you use MySQL Workbench in the university lab computers, please note that your connection to the server is removed once you logged-out, so you have to back up your database and save it as .sql file in your own storage device (e.g. USB) not to risk losing your work.

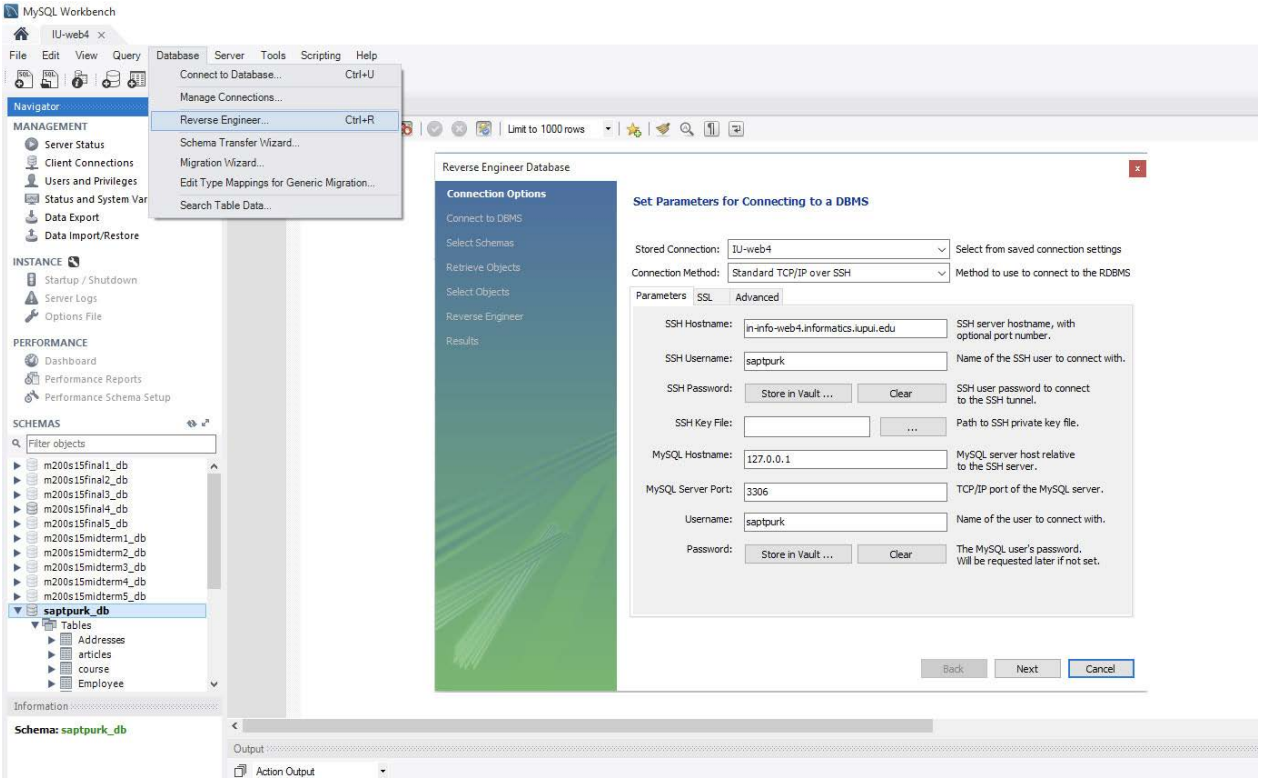
[Task 3]

Reverse Engineering (creating the ERD from the existing physical database)

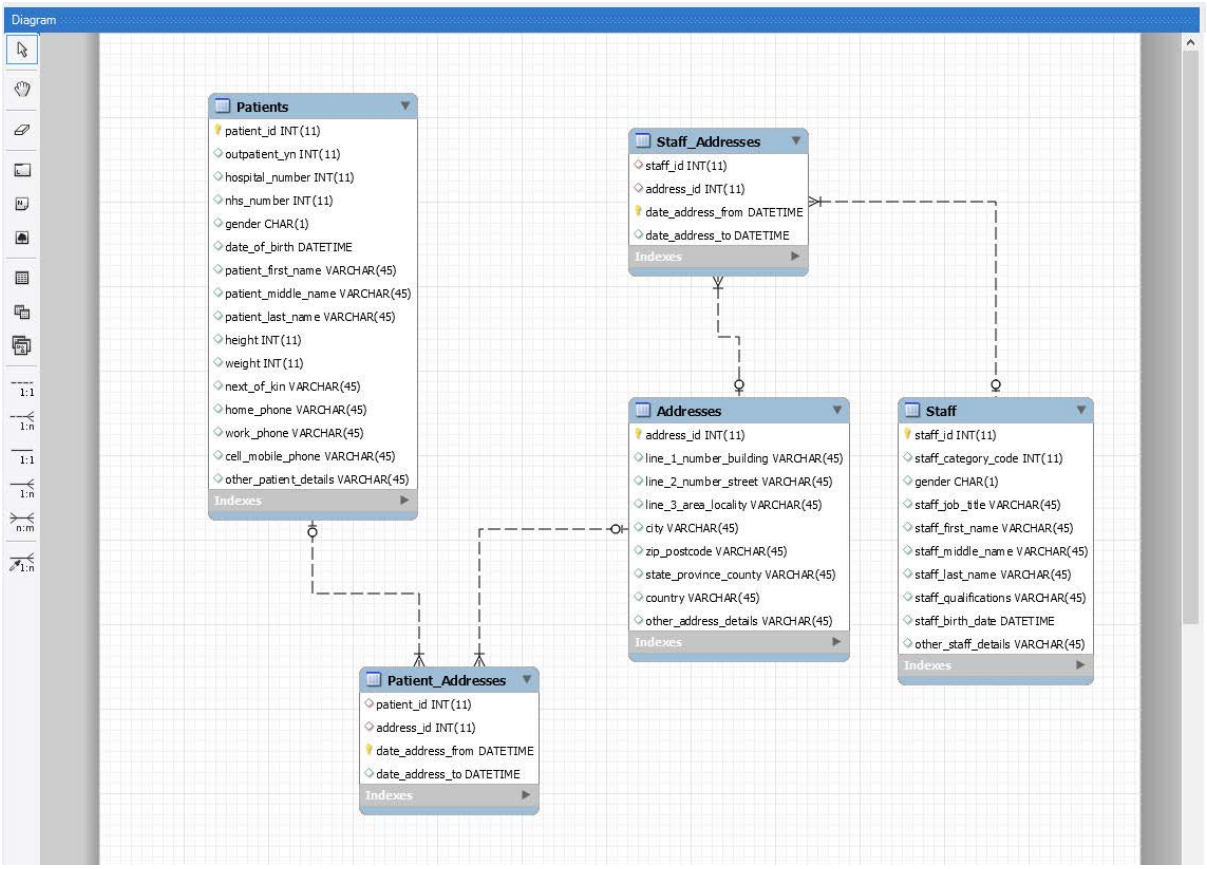
- Go back to the MyFirstConnection at the very top.
- If you don't see all the new tables, right click and go to the Refresh All button.
- This is the physical schema with the tables that have been created through the process of Forward Engineering you have done in Task 2.
- Close the MySQL Model and EER diagram tab from here. We will now move to the Reverse Engineer process. This process is to get an ER model from a physical schema. This process is used for documenting a database and to share the data model with others.



- Double click on your schema – xxx.db as you named such that it becomes bold
- Go to Database menu and click on Reverse Engineer
- Select your Stored Connection to the local server and press Next
- All Connections and schema retrieval should work as seen below. Press Next on the following screen as well.



- On the next screen check mark the database schema (xxx.db) and press NEXT
- Press next on the following screen and then execute on the following screen.
- It should deploy the tables on the canvas grid, something like the next screen.



- This ER model might need arrangement of the tables to give it the same kind of look that you see here.
- Save this ER model as a file (xxx-reversed.mwb).

This ER model produced by the 'reverse engineer' process may look nearly same as the original ER mode you created in [Task 1] in this prac. Thus in general, you may feel that this reverse engineering process would not be necessary.

However, in most practices, the original ER model you had designed at the early stage may have been modified during your further working on physical database schema (after processed forward-engineering) for various reasons (e.g. modifying PKs or FKs, adding additional attributes, deleting existing attributes etc.). In this case, applying reverse-engineering process helps you produce the updated ERD easily (by automatic processing)

This is the end of Week 3 Lab.

You are required to submit all result files from each task to be marked off:

Task 1: xxx.mwb

Task 2: xxx.sql

Task3: xxx-reversed.mwb
