

# SQL Server 2008 Tutorial

---

EGCI 321: LECTURE 07 (WEEK04)

# Outline

---

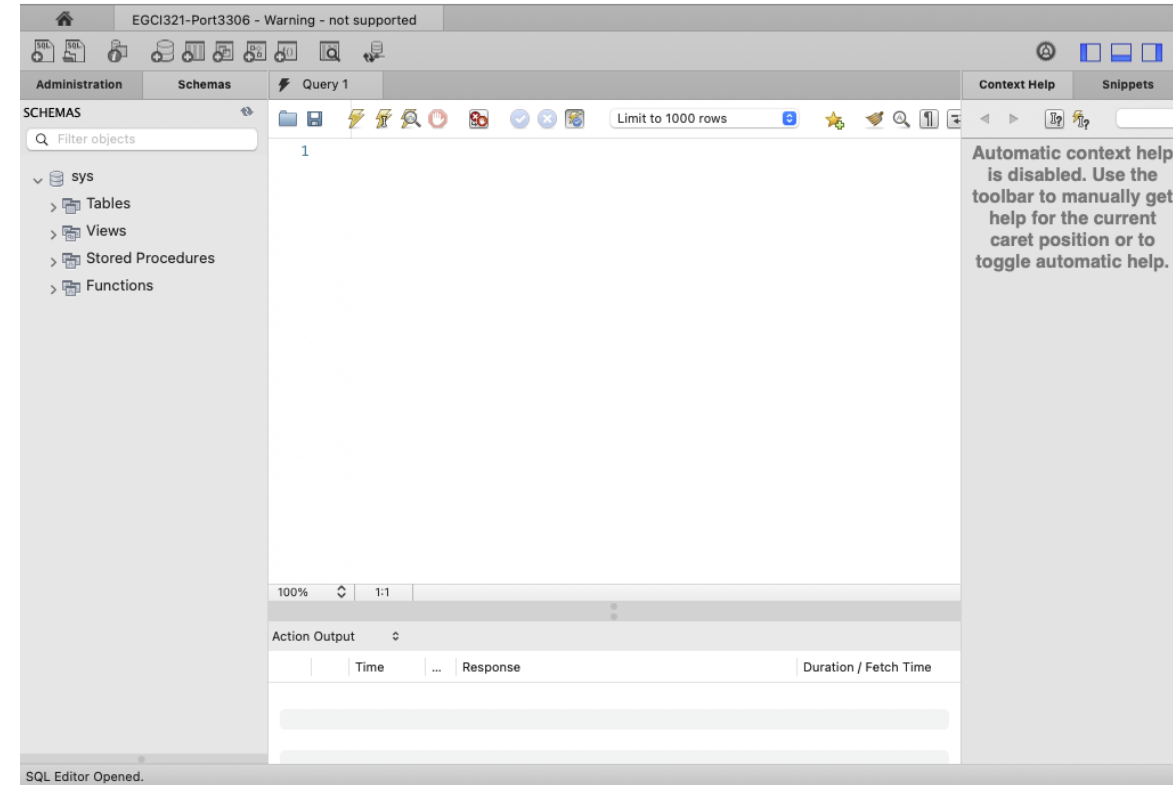
This tutorial consists of the following lessons:

- SQL Server Editions
- SQL Server - Management Studio
- SQL Server - Create a Database
- SQL Server - Create a Table
- SQL Server - Adding Data
- SQL Server - Query Designer
- SQL Server Views
- SQL Server Stored Procedures
- User Logins
- Server Roles
- Database Schemas
- Summary

# MySQL Workbench

MySQL Workbench enables you to create database objects

- Such as databases, tables, views
- View the data within your database
- Configure user accounts
- Transfer data between databases



# System Databases

---

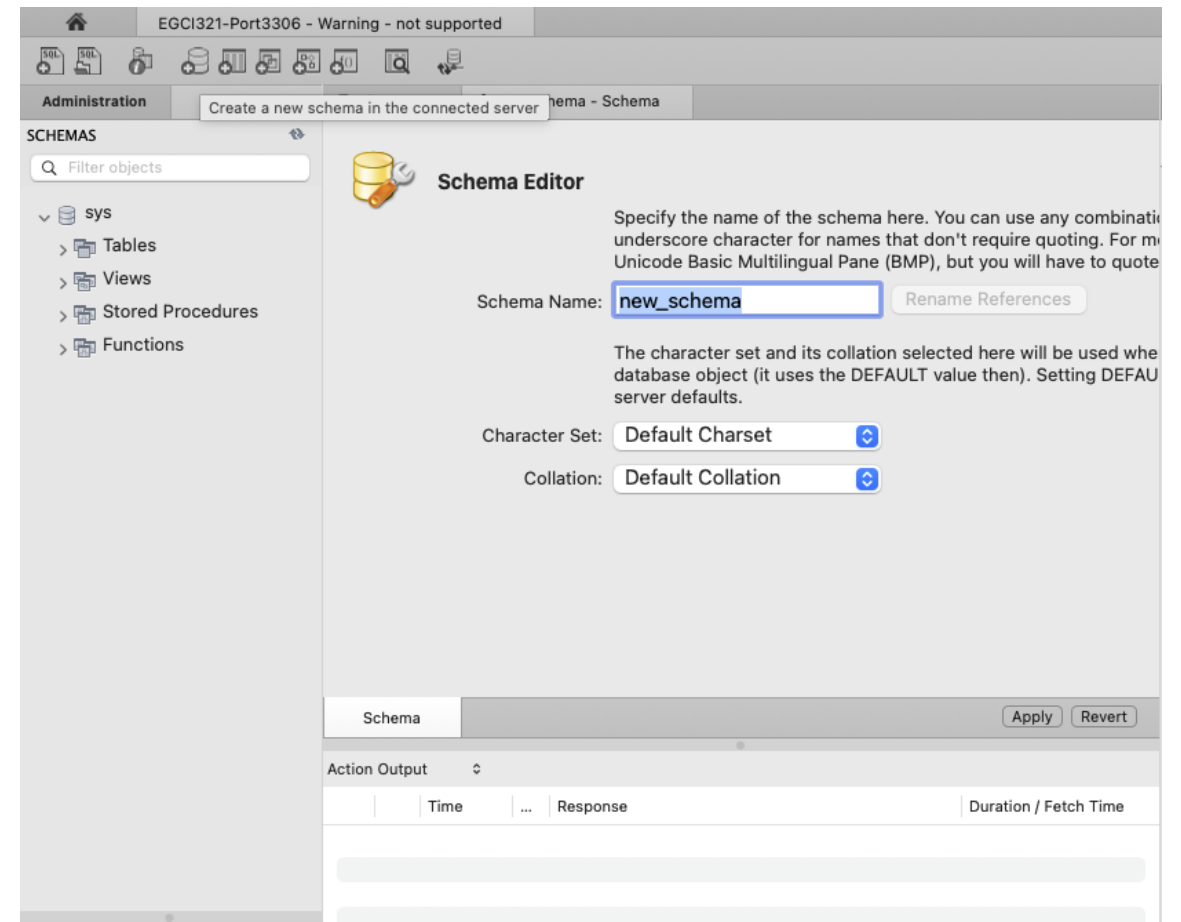
## Sys

This database is a set of objects, such as views and stored procedures, that help developers and database administrators (DBAs) interpret the data collected by the performance\_schema.

- It simplifies the complex and low-level data from the performance\_schema into a more human-readable format.
- The sys schema is installed by default in modern MySQL versions and provides reports for common tuning and diagnostic tasks.

# Creating a New Database

1. Right click on the "Create New Schema" icon
2. Name your database



# Database Schemas

---

A database schema is a way to logically group objects

- Such as tables, views, stored procedures
- Think of a schema as a container of objects

You can assign a user login permissions to a single schema

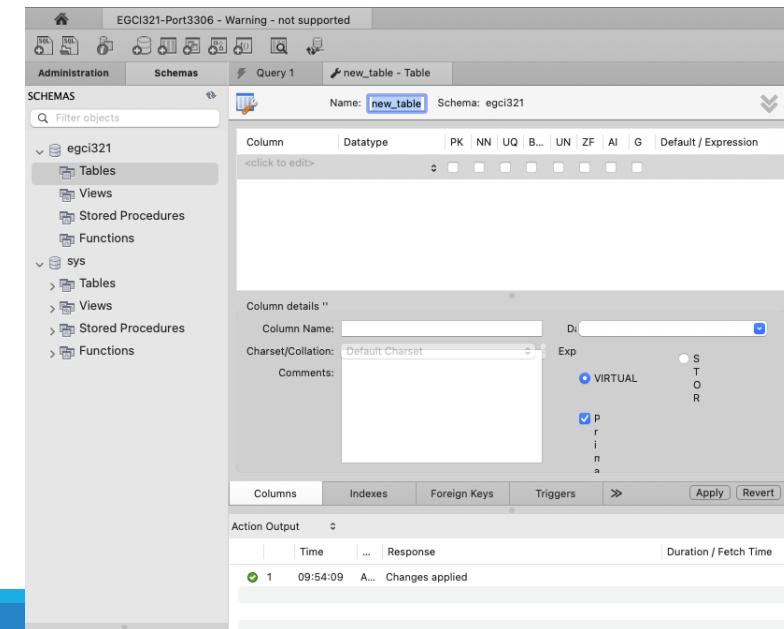
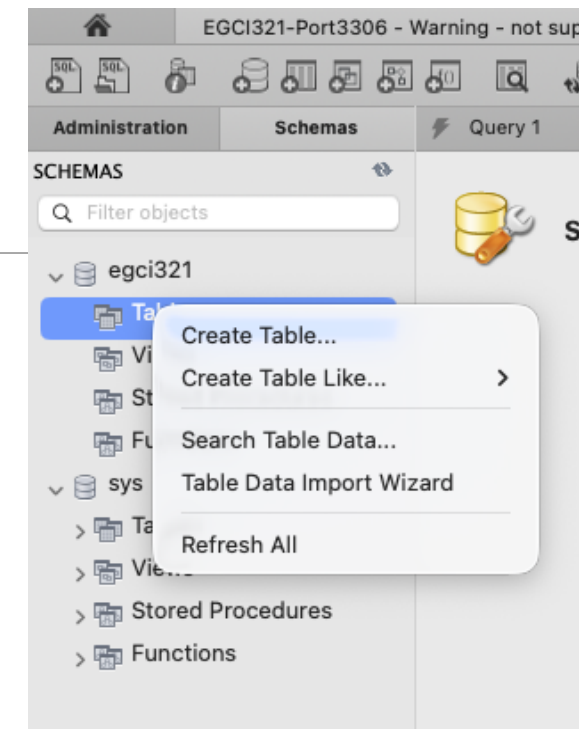
- User can only access the objects they are authorized to access

Schemas can be created and altered in a database, and users can be granted access to a schema

- A schema can be owned by any user, and schema ownership is transferable

# SQL Server - Create a Table

1. Ensuring you have the right database expanded, right click on the "Tables" icon and right click "Create Table..."
2. While you have this screen open, do the following:
  - Define table name
  - Define table schema: column name and column type
3. Click Apply



# Editing Table Rows

1. To use this option, right click on the table you wish to open, and select "Select Rows – Limit 1000":
2. You can now start entering the data directly into your table.

The screenshot displays a database management interface with two main windows. The left window, titled 'EGCI321-Port3306 - Warning - not supported', shows a 'SCHEMAS' panel on the left with a tree view containing 'egci321' and 'sys'. The 'egci321' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'students' table is selected, and a context menu is open with the option 'Select Rows - Limit 1000' highlighted. The main panel shows a query editor with the SQL statement: `1 SELECT * FROM egci321.students;`. Below the query editor is a 'Result Grid' showing the data for the 'students' table:

sid	sname	major
s01	Bob...	CI
s02	Tim	Fox
s03	Mike	Mike

The right window, titled 'Apply SQL Script to Database', shows a 'Review the SQL Script to be Applied on the Database' dialog. It contains the following SQL script:

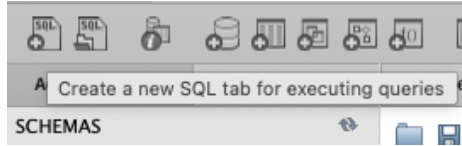
```
1 INSERT INTO 'egci321'.students ('sid', 'sname', 'major') VALUES ('s01', 'Bob Cat', 'CI');
2 INSERT INTO 'egci321'.students ('sid', 'sname', 'major') VALUES ('s02', 'Tim', 'Fox');
3
```

The dialog also includes a 'Go Back' button and an 'Apply' button.



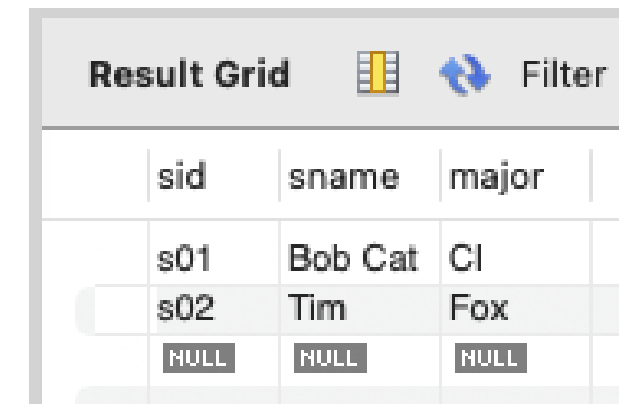
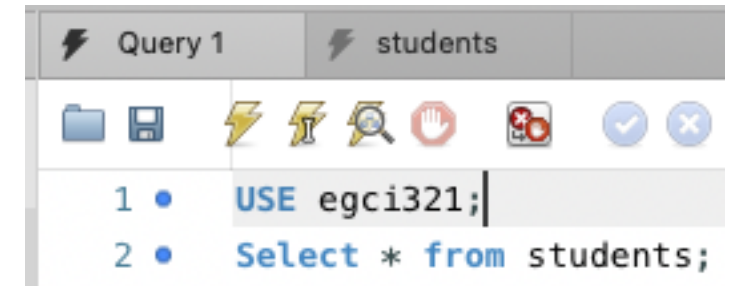
# Create a New Query

Just click the "New Query" button:



## Write/Run Your SQL Script

1. Type your query into the workspace on the right pane
2. Click "Execute" (you can also press. ⚡ )

A screenshot of a "Result Grid" interface. It has a title bar with "Result Grid" and a "Filter" button. Below the title bar is a table with three columns: "sid", "sname", and "major". The table contains three rows of data.

sid	sname	major
s01	Bob Cat	CI
s02	Tim	Fox
NULL	NULL	NULL

# Views Table

---

*view* is a pre-written query that is stored on the database.

- *view* consists of a **SELECT** statement, and when you run the view
  - ▶ you see the results of it like you would when opening a table
- Some people like to think of a view as a *virtual table*

A view can be useful when there are multiple users with different levels of access, who all need to see portions of the data in the database

Restrict access to specific *rows* in a table

Restrict access to specific *columns* in a table

*Join columns* from multiple tables and present them as though they are part of a single table

Present *aggregate information* (such as the results of the **COUNT** function)

# Explanation of Server Roles

---

Server Role	Description
sysadmin	Can perform any task in SQL Server.
serveradmin	Can set server-wide configuration options, can shut down the server.
setupadmin	Can manage linked servers and startup procedures.
securityadmin	Can manage logins and database permissions, read logs, change passwords.
processadmin	Can manage processes running in SQL Server.
dbcreator	Can create, alter, and drop databases.
diskadmin	Can manage disk files.
bulkadmin	Can execute <u>BULK INSERT</u> statements.
public	Every SQL Server user account belongs to this server role. When a server principal has not been granted or denied specific permissions on a securable object, the user inherits the permissions granted to public on that object. Only assign public permissions on an object when you want the object to be available to all users.