

# **Computer Engineering Department**

King Mongkut's University of Technology Thonburi

## **CPE 241 Database**

### **Problem Session #01 Getting started**

**Date:** Check PS01 on LEB2

#### **Prepare your data structure and Environment**

Welcome to the first installment of our problem session series, designed to help you develop essential database-related skills. In this course, you will explore various aspects of database systems, including design, normalization, management, and integration with web applications. The lab session will guide you through setting up your first working environment and designing a structured framework for storing and managing your data effectively. This foundational knowledge will prepare you for more advanced database operations and integrations in future sessions.

#### **Task List**

##### **1. Identify Data Requirements**

- Determine the type of data you need to store.
- Outline key attributes and relationships between data elements.

##### **2. Set Up the First Development Environment**

- Install and configure a database management system (e.g., MySQL, PostgreSQL, or SQLite).
- Set up additional tools as needed such as DBMS GUI (Graphical User Interface)

## Task 1: Identify Data Requirements

From the lecture, you will begin thinking about the data you want to store and how to organize it into tables. At this stage, you don't need to worry about being perfectly correct; the goal is to start conceptualizing data structures.

### 1. Understand the Goal

- You will be tasked with storing specific types of data. For example, if the goal is to track student grades in courses, you need to design tables to store this information effectively.

### 2. Design Initial Data Structures

- For each type of data, create a table and define the data it will keep. For example:
  - **students** needs **studentID**, **firstname**, and **lastname**
  - **courses** only contains **courseName**
  - **grades** needs to have **studentID**, **courseName**, and **grade**

### 3. Understand Keys

- A key is an attribute (or a combination of attributes) that uniquely identifies a data record. For instance:
  - **studentID** uniquely identifies a student.
  - **courseName** uniquely identifies a course.
  - For **grades**, a combination of **studentID** and **courseName** is needed to uniquely identify each record.

### 4. Express Your Data Structures

- Use the format `data_structure(data_to_keep)` to represent each table.
- For example:
  - **students(studentID, firstname, lastname)**
  - **courses(courseName)**
  - **grades(studentID, courseName, grade)**

By completing this step, you will have a basic framework for organizing your data into tables, which will serve as the foundation for future tasks in this session.

You are going to think of how to store data for the following situation:

1. a law firm wants to keep employees and their offices.
2. Thai customs want to keep tourists and their passports that they used to enter Thailand.
3. an airline wants to keep customers and their booking flights.
4. a bank wants to keep customers and their accounts.
5. a car dealer that wants to store cars, customers, and orders.
6. a film archive that wants to store movies, actors, and casts.
7. a hospital that wants to store patients, doctor, and appointment.
8. a company that wants to store departments, employees, employee phone numbers.
9. a supermarket that wants to store products, receipts, and receipt items.
10. an online store that wants to store customers, products, and reviews.
11. a bookstore wants to keep books and authors of each book.
12. a restaurant wants to keep tables, menus, and order of each table.

You will have to create a pdf file containing your answer and submit it before the due date.

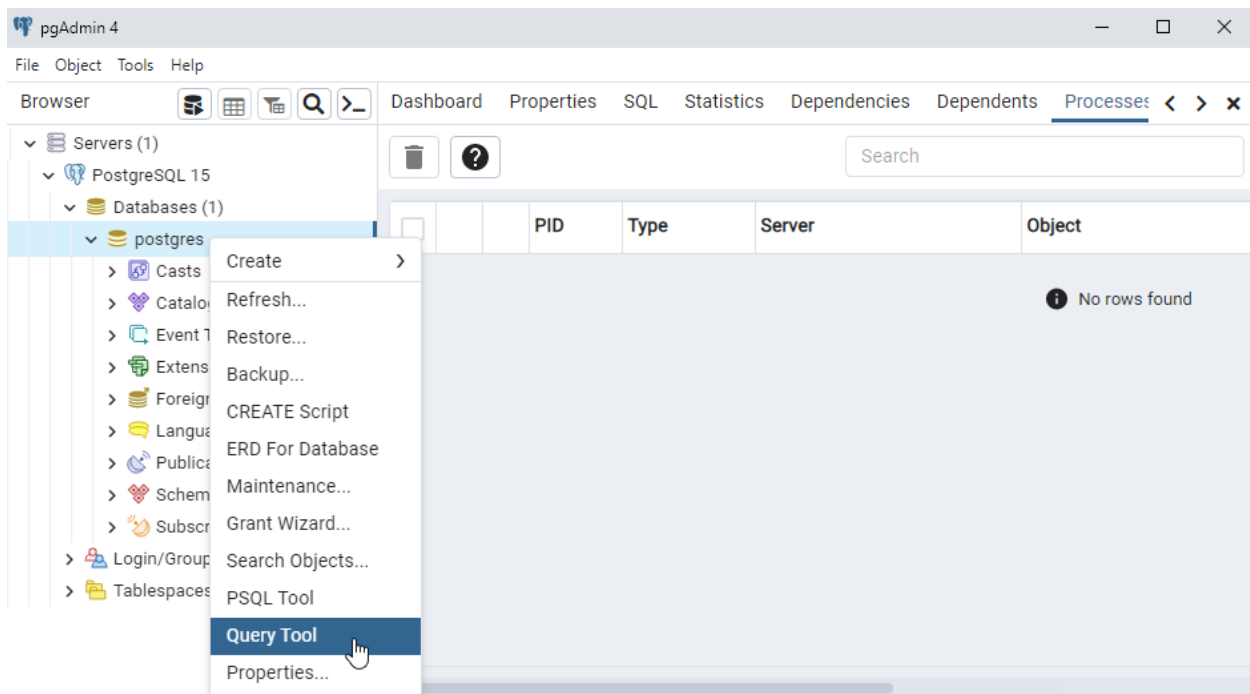
## Task 2: Set Up Development Environment

To practice and complete your term project, you will need to install a development environment. In this course, we will focus on using a Linux-based environment. The goal is to install and configure this stack, preparing it for database and web application development. However, in this lab you will only start with any DBMS.

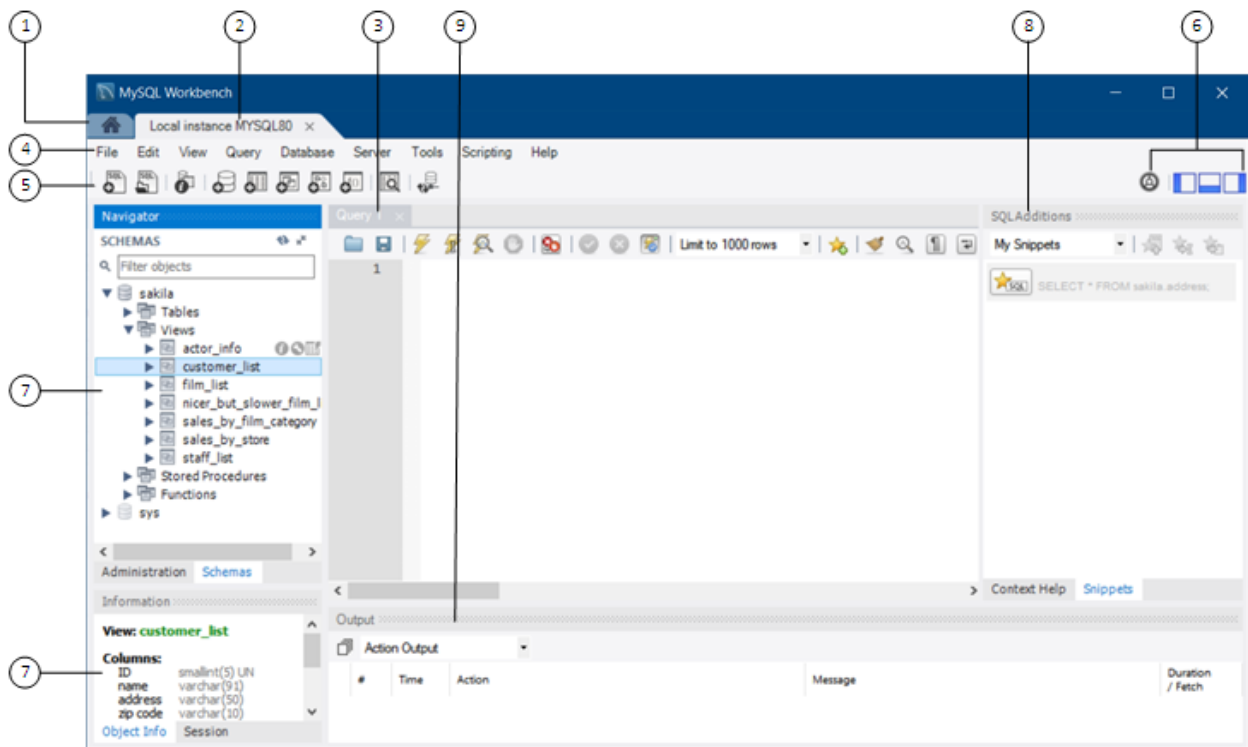
### Steps to Complete

1. Install any DBMS, even Microsoft Access is fine!
2. Install DBMS GUI if needed such as pgAdmin (for PostgreSQL), MySQL Workbench (for MySQL), etc.

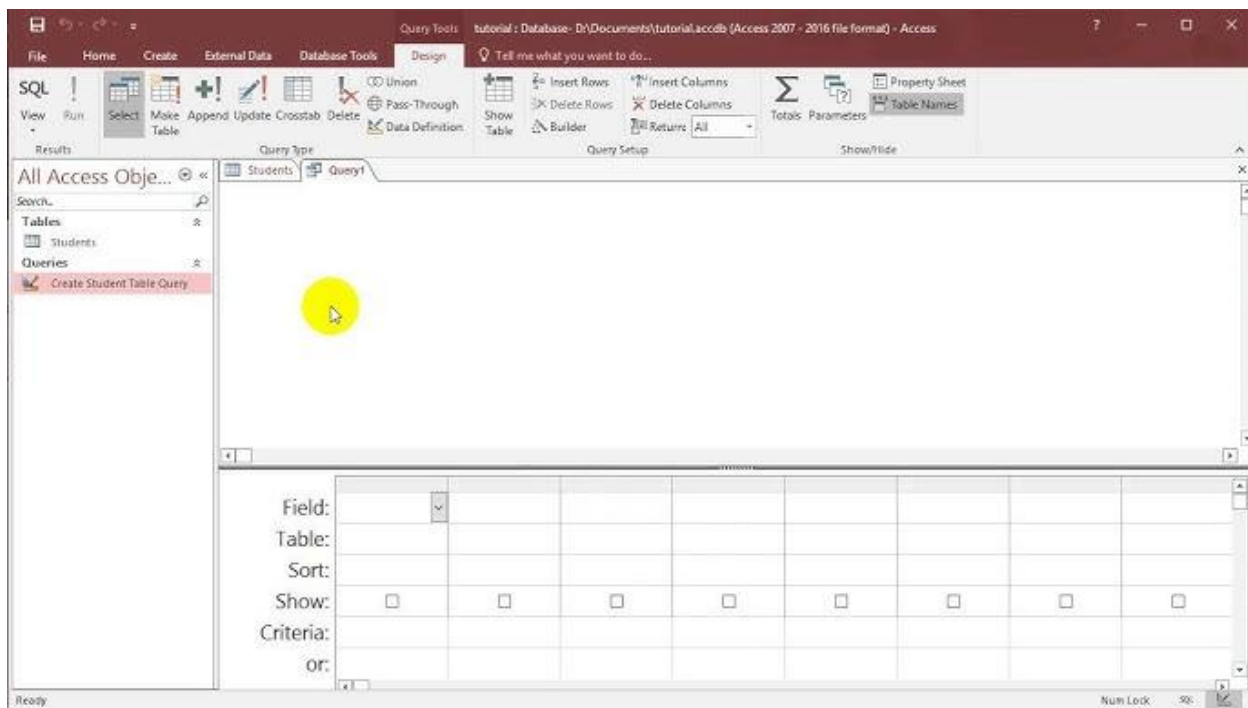
#### *pgAdmin*



## MySQL Workbench



## Microsoft Access



### 3. Test Your Environment

- Find a way to run SQL commands. It could be run through an SQL Editor.
- Try to create a database name 'students'. You can run the following commands:

```
CREATE DATABASE students;
```

```
SHOW DATABASES;
```

- Let's explore the tool by trying to create a table in the database.

---

#### Outcome

If all steps are completed successfully and no errors arise, the system will show a list of databases including 'students' you created. Unless you choose to use MS Access, the program will force you to save your database as the name you want at the first moment. This environment can be used for all practical work in the next few labs. We will need to set up more computer environments in order to create a fully functional development platform ready for developing your databases and web applications later.

---

#### Submission

Submit your Task 1 design on LEB2 before the due date. There is nothing to submit for the second task, but make sure that you can connect to your DBMS. There should be an assessment activity under the name of PS01 listed.