Practical 2

Working with tables-1

Learning objectives

1. Insert values to tables
2. Retrieve data from rows and columns from a single table
3. Update values of a table row
4. Delete rows of a table

Tasks

In this practical, you will use SQL commands and add data to database tables you have created earlier. Then you will retrieve, update and delete data using SQL commands. You will see the content of tables as well.

Make sure you are in I: drive, and the DBS/Prac02 directory when doing this practical if you are using Curtin labs or your computer's DBS/Prac02 directory if you use your own computer.

1. Creating tables using the text file

Use terminal and copy your 'Pra01tables' file (text file) from Prac01 directory to Prac02 directory. Rename the 'Pra01tables' file in Prac02 directory to 'Prac02tables'.

Open the 'Prac02tables' file in vim.

Three names of the tables shown in the CREATE TABLE statements are 'TestStudents' , ' TestUnits' and 'TestEnrolments'.Change these table names to 'Students', 'Units' and 'Enrolments' , save and close the file.

Now, while in Prac02 directory, connect to the MySQL server using the commands (refer to Prac01 Task 2 if you wish to look at the command).

Open a file in Vim (> vim Prac02Commands) or any text editor you wish to use and first create your commands in this file from this point onward.

Use comment line before starting each task using any comment styles below

# this is a comment

-- this is also a comment

/* this is a comment too

***/**

Look at the available databases in the MySQL server using a SQL command

We are going to use ‘dswork’ database. Type the correct command to use the ‘dswork’ database. Create the three tables ‘Students’, ‘Units’ and ‘Enrolments’ using ‘Prac02tables’ file.

Use `SHOW COLUMNS` command to see the structure of each table. (Refer the Prac01 for the command.)

2. Adding tuples

The tables that you have created are currently empty. You will now add data to the tables and then remove some of it. After every step you should use the appropriate SELECT statement to check the changes to the table.

1. Add student James Bond (with student number 12345007 and phone 0408007007) to the Students table.

   Note1: use of quotes

   - You should use single quotes around the numbers as the columns are of character type. Be careful to use the proper single quote **'** instead of smart quote such as **‘**or the angled quote **`**. Using the wrong sort of quote can cause errors that are difficult to understand.

   Add a few more tuples (at least five) to Students table including one for you.

   View all the data in Student table with a "SELECT *" query.

2. Add the unit with the following values to the units table:

   Name: 'Database Systems'
   Index: 'ISYS1001'
   Dept: 'COMPUTING'

   Add a few more tuples to Units table including the units you are currently enrolled in. You don’t need to enter the correct unit codes if you don’t want to look them up.

3. James Bond has enrolled in Database Systems this year. Add an entry to the enrollments table to reflect this change. He has no mark or grade for the unit yet, so you will have to use NULL for the fields whose value is unknown.

   Add a few more tuples (at least eight) to Enrolments table including some of your own enrolments – feel free to give yourself good grades. The correct way to do this is given in the lecture slides. Add some enrolment data which have

   View all the data in Enrolment table with a "SELECT *" query.

The Enrolments table should have one line (called a “tuple”) for every student-unit pair.

3. Retrieving data from some columns

1. Retrieve only the first name and the telephone number of students from the Students table.
2. Find only the unit name and the department the unit is offered.
3. Retrieve codes of all the units students have enrolled into.
4. **Additional task:** You may see duplicate units as Enrolment table. **DISTINCT** keyword can be combined with SELECT statement to avoid duplicates. Revise your previous command to retrieve only unique unit codes from the Enrolment table.

4. Retrieving tuples using a condition

1. Retrieve Units offered by 'COMPUTING' department.

   (Hint: You have to use a SELECT statement with WHERE clause)
2. Retrieve student number, mark for the unit and grade for the unit for where the mark for the unit is greater than 60. (Hint: Use Enrolment table)
3. **Additional task:** Retrieve student number, mark for the unit and grade for the unit, where the mark for the unit is between 66 and 85 (including 65 and 85 both).

5. Updating data

1. Update the unit code of the database systems unit to ‘ISYS1020’ in Units table.

   Reflection : Now the unit code of Database Systems is changed. Does this change reflected in the unit code of Enrolment table? Does this cause any problems for the Enrolments table?

6. Deleting tuples

1. Delete James Bond's entry from the students table (using a condition on sno in the where clause) and check the result with a "SELECT *" query.

   Reflection: Does this cause any problems for the Enrolments table? Should it?

2. Delete the Database Systems unit from the units table (using a condition on unitindex).

   Reflection: Note any effects on the Enrolments table.

Make sure you have saved all the commands you have created today in the DBS/Prac02/Prac02Commands file. We will be using the same commands in next week as well.

Check whether you have achieved learning objectives:

I am confident that I can write SQL statements to,

add data to tables using (INSERT)	✓
retrieve all data from a table (SELECT)	
retrieve some columns of data from a table (SELECT)	
retrieve rows of data fulfilling some conditions (SELECT)	
update data in a table (UPDATE)	
deleting all data or selected data in a table (DELETE)	

Please refer lecture slides, reading materials, and online resources and attempt again, if all the objectives were not achieved. Ask your tutor and get help if you need any clarification.

It's always a good practise to try to finish the practical of a particular week, before attempting the next practical worksheet as your work will be building upon the previous week's tasks.