**6G4Z0016 Databases**

**Introduction to MariaDB Databases and SQL**

**Creating Tables and Manipulating Data Labsheet 1**

**Aim**

Students should save their work as they go along. This can be done by saving their scripts from MySQL Workbench or by copying and pasting the SQL code into one or several .txt files and saving these.

The aim of this session is to be able to:

* manipulate data in a table using INSERT, UPDATE and DELETE commands.
* create a simple table using the CREATE command
* use Primary and Foreign Key Constraints
* apply CHECK constraints where appropriate.

**Resources**

**Activities**

1. Ensure you have downloaded the createHR.sql script from Moodle.
2. Start MySQL Workbench and connect to your account on the University’s mudfoot server (see the document “Getting Started with MySQL Workbench” on Moodle for help). Your account details will have been emailed to you already.
3. Open the createHR.sql script you have downloaded with File->”Open SQL Script” or pressing Ctrl+Shift+O
4. Run the script by clicking the lightning button:



1. Open a new SQL file with File->”New Query Tab” or pressing Ctrl+T
   1. This is where you can write your own SQL commands and save the script.

**PART A: Data Manipulation**

a) To insert a new row containing values for each column, list the values in the default order of the columns in the table. Type in and execute the following SQL INSERT statement.

INSERT INTO departments(department\_id, department\_name,   
 manager\_id, location\_id)  
VALUES (280, 'Public Relations B', 100, 1700);

Type in and run the following SQL query to ensure that this row of data has been inserted.

SELECT \*   
FROM departments  
WHERE department\_id = 280;

b) Sometimes not all the data is available when you want to insert data into a table. Supposing you want to create a new department with department ID of 290 and name ‘Finance B’ but you do not yet have the manager’s ID or the location ID. Write an SQL statement to insert this data. Record your answer below.

For your answer, did you use the implicit or the explicit method? Make sure you know the difference between the two!

c) Insert a new Department with ID 300 and name ‘Sports Science’ with Location ID of 2500 into the Departments table.

d) Write an SQL statement to check that the two new departments 290 and 300 have been inserted. (see SELECT Statements Pt1 lecture: slide 13)

e) The UPDATE clause allows the modification of one to ALL rows in a table. The following example updates department 280 by changing its name to ‘Public Engagement’

UPDATE departments

SET department\_name = 'Public Engagement'

WHERE department\_id = 280;

Copy and paste and execute this SQL statement and use a query to ensure that the name has been changed.

f) Existing rows are removed from a table using the DELETE statement. To delete a specific row use a WHERE clause. Consider the following query

DELETE FROM departments  
WHERE department\_name = 'Public Engagement';

Copy and paste this statement in and execute it. What is the message returned by mariaDB?

Instead try

DELETE FROM departments  
WHERE department\_id = 280;

If the statement gives you errors, then check your spelling and the make sure you are using the correct case.

**NOTE**: The DELETE statement must be used with care especially if the delete is unconditional i.e. has no WHERE clause.

g) The HR department wants you to create statements to insert, update and delete employee data. As a prototype, you will use the MY\_EMPLOYEE table to test the SQL statements before giving them to the HR department. Copy and Paste and execute the following SQL script to create the MY\_EMPLOYEE table.

CREATE TABLE my\_employee

(id DECIMAL(4) PRIMARY KEY,

last\_name VARCHAR(25),

first\_name VARCHAR(25),

userid VARCHAR(8),

salary DECIMAL(9,2));

**NOTE: The CREATE statement will be described and explored fully in Part B of the tutorial.**

If your table creates successfully you will see the message Table created.

h) To describe the structure of the table you have created use the DESCRIBE command. Type in the following SQL statement and execute it. Observe the output.

DESCRIBE my\_employee;

i) Create an INSERT statement to add the row of data below into the my\_employee table;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **LAST\_NAME** | **FIRST\_NAME** | **USERID** | **SALARY** |
| 1 | Patel | Ralph | rpatel | 895 |

j) Write insert statements to load in the rows of data shown in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **LAST\_NAME** | **FIRST\_NAME** | **USERID** | **SALARY** |
| 2 | Dancs | Betty | Bdancs | 860 |
| 3 | Biri | Ben | Bbiri | 1100 |
| 4 | Newman | Chad | Cnewman | 750 |

Record your answer in the box below.

k) Write a select statement to display all the contents of the my\_employee table.

l) Use an UPDATE statement to change the last name of Betty Dancs to Drexler (ID = 2). Record your SQL statement below.

m) Delete Betty Drexler from the my\_employee table and verify the changes using a query.

**PART B Creating and Managing Tables**

3.

a) The CREATE TABLE statement is used to create tables within the database. This statement has an immediate effect on the database. You must have privileges in the database in order to create tables. Consider the statement below.

CREATE TABLE dept

(deptno DECIMAL(2),

dname VARCHAR(14),

loc VARCHAR(13),

create\_date DATE DEFAULT CURDATE());

The table DEPT contains 4 fields, one of which is a DECIMAL, two are strings represented by the type VARCHAR and one is a DATE. The create\_date field has a DEFAULT column (optional) which prevents NULL values being entered. In this example the system date is automatically inserted when a row of data is inserted into the table.

Copy and paste in the above statement and create the dept table. Use the DESCRIBE command (“DESCRIBE dept”) to verify the table has been created.

b) It is important when creating the table that the data types of the fields are created in accordance with the data dictionary. MariaDB has a number of data types that are described in the documentation https://mariadb.com/kb/en/data-types/. For the purposes of this unit you will be expected know and use VARCHAR, CHAR, DECIMAL, INT, DATE and DATETIME when creating tables. Make your own notes about these four types below.

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| VARCHAR(n) | Character string. Variable length. Maximum length n. |
| CHAR(n) | Character string. Fixed-length n |
| INT | Integer values. |
| DECIMAL(p,s) | Exact numerical, precision p, scale s. Example: DECIMAL(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal |
| DATE | Stores year, month, and day values (can also store time) |
| DATETIME | Stores year, month, day, hour, minute and second values (can also store microseconds) |

c) Constraints can be applied to a table when it is created to enforce rules. MariaDB has the following constraint types: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY and CHECK. For the purpose of this unit you will need to know at least how to apply PRIMARY and FOREIGN key constraints. Study the example code below and try to understand each of the constraints that have been applied.

CREATE TABLE employees\_teach

(

employee\_id DECIMAL(6) PRIMARY KEY,

first\_name VARCHAR(20),

last\_name VARCHAR(25) NOT NULL,

email VARCHAR(25) NOT NULL UNIQUE,

phone\_number VARCHAR(20),

hire\_date DATE NOT NULL,

job\_id VARCHAR(10) NOT NULL,

salary DECIMAL(8,2) CHECK (salary>0),

commission\_pct DECIMAL(2,2),

manager\_id INT(6),

department\_id INT(4),

CONSTRAINT empl\_dept\_fk FOREIGN KEY (department\_id) REFERENCES departments (department\_id),

CONSTRAINT empl\_mgr\_fk FOREIGN KEY (manager\_id) REFERENCES employees (employee\_id)

);

In the above code example, notice that the foreign keys constraints are named.

If you need to make changes to the table structure after creation you can use the ALTER TABLE statement.

d) To remove a table from the database use the DROP TABLE command. Type in and execute the following command to remove the my\_employee TABLE you created in PART A of the tutorial.

DROP TABLE my\_employee;

e) Write a statement to remove the dept table you created in a).

f) Create the SECTION table based on the following table instance chart (this is the same as a data dictionary but turned the other way, so rows become columns and vice versa).

|  |  |  |
| --- | --- | --- |
| Column Name | ID | NAME |
| Key Type | Primary Key |  |
| Nulls/Unique |  |  |
| FK Table |  |  |
| FK Column |  |  |
| Data Type | DECIMAL | VARCHAR |
| Length | 7 | 25 |

g) Create the EMP table based upon the following table instance chart.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column Name | EMP\_NO | LAST\_NAME | FIRST\_NAME | SECTION\_ID |
| Key Type | Primary Key |  |  | Foreign Key |
| Nulls/Unique |  |  |  |  |
| FK Table |  |  |  | SECTION |
| FK Column |  |  |  | ID |
| Data Type | DECIMAL | VARCHAR | VARCHAR | DECIMAL |
| Length | 8 | 25 | 25 | 7 |

h) Interrogate the structure of each of the tables you have created using the DESCRIBE command. If you have time, try inserting one row of data into each table.