**COP4710 – Theory and Structure of Databases**

**Summer 2016**

**Homework 3a**

Due Sunday Night, May 29, 2016

***►Eric Adams◄***

**All Cut-And-Paste – Watch how the process flows.**

You’ll need these Read-Me's:

* Read-Me 2010 Logging into your CS Department Account.
* Read-Me 2020 Logging into your MySQL account.
* Read-Me 0340 SQL's Data Definition Language (DDL)

How to cut-and-paste in a terminal window (I think):

1. **Select** text: Drag over it with your mouse (hold the left mouse button while moving the mouse).
2. PuTTY  
   **Copy**: Release the left mouse button after dragging. Don't press Ctl-C.  
   **Paste**: Right-click the mouse.
3. Tectia Client:  
   **Copy**: **Ctrl-Insert  
   Paste**: Use the menu, or set Preferences to paste-on-right-click.

***SQL is not case sensitive, but MySQL is set to honor case-sensitive names. Therefore:***

* ***SQL keywords are non-case sensitive***
* ***Table names are case sensitive***

Instructions:

* Log into your CS Department account, and from there log into your MySQL account.
* Run these SQL statements (***just cut-and-paste them***), and ***cut-and-paste*** the ***results*** into your homework. (***100% cut and paste***, but experiment along the way).

**The current database.**

1. Use this statement to determine your current database.

**SELECT DATABASE();**

*+---------------+*

*| DATABASE() |*

*+---------------+*

*| cgs4710t01\_db |*

*+---------------+*

*1 row in set (0.01 sec)*

In MySQL, you're “in” a database just like you're “in” a directory. SQL statements work on the tables of the database you're “in” (called the *currently logged database* or the *default database*).

The **DATABASE()** function returns the name of the current database, If you run the function by itself, you'll get an error. The **SELECT** statement displays the output of the DATABASE() function. (**DATABASE()** is specific to MySQL).

2. Switch to your database. For this exercise, do this even if you're already in your own database:

**USE** *mynamedb***;** --your login name with “db” appended

*Database changed*

**Data Definition Language (DDL).**

3. Create the table from the Read-Me:.

**CREATE TABLE Employee (**

**employeeNum INT,**

**lastName CHAR(20) NOT NULL,**

**firstName CHAR(20) NOT NULL,**

**deptNum INT,**

**gender ENUM('m','f')**

**);**

*Query OK, 0 rows affected (0.00 sec)*

**ALTER TABLE Employee**

**ADD PRIMARY KEY (employeeNum);**

*Query OK, 0 rows affected (0.03 sec)*

*Records: 0 Duplicates: 0 Warnings: 0*

4. See a list of your tables:

**SHOW TABLES;**

*+-------------------------+*

*| Tables\_in\_cgs4710t01\_db |*

*+-------------------------+*

*| Employee |*

*+-------------------------+*

*1 row in set (0.00 sec)*

5. See how your table was created:

**SHOW CREATE TABLE Employee;**

*+----------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+*

*| Table | Create Table |*

*+----------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+*

*| Employee | CREATE TABLE `Employee` (*

*`employeeNum` int(11) NOT NULL default '0',*

*`lastName` char(20) NOT NULL,*

*`firstName` char(20) NOT NULL,*

*`deptNum` int(11) default NULL,*

*`gender` enum('m','f') default NULL,*

*PRIMARY KEY (`employeeNum`)*

*) ENGINE=MyISAM DEFAULT CHARSET=latin1 |*

*+----------+-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------+*

*1 row in set (0.00 sec)*

6. List the columns

**SHOW COLUMNS IN Employee;**

*+-------------+---------------+------+-----+---------+-------+*

*| Field | Type | Null | Key | Default | Extra |*

*+-------------+---------------+------+-----+---------+-------+*

*| employeeNum | int(11) | NO | PRI | 0 | |*

*| lastName | char(20) | NO | | NULL | |*

*| firstName | char(20) | NO | | NULL | |*

*| deptNum | int(11) | YES | | NULL | |*

*| gender | enum('m','f') | YES | | NULL | |*

*+-------------+---------------+------+-----+---------+-------+*

*5 rows in set (0.00 sec)*

7. Get rid of the gender column

**ALTER TABLE Employee**

**DROP COLUMN gender;**

**SHOW COLUMNS IN Employee;**

*+-------------+----------+------+-----+---------+-------+*

*| Field | Type | Null | Key | Default | Extra |*

*+-------------+----------+------+-----+---------+-------+*

*| employeeNum | int(11) | NO | PRI | 0 | |*

*| lastName | char(20) | NO | | NULL | |*

*| firstName | char(20) | NO | | NULL | |*

*| deptNum | int(11) | YES | | NULL | |*

*+-------------+----------+------+-----+---------+-------+*

*4 rows in set (0.00 sec)*

**Data Manipulation Language (DML).**

8. Add a row of data:

**INSERT INTO Employee (EmployeeNum, LastName, FirstName, DeptNum)**

**VALUES (217, 'Watson', 'Bill', 200);**

*Query OK, 1 row affected (0.00 sec)*

9. Add multiple rows in a single statement (MySQL specific):

**INSERT INTO Employee (EmployeeNum, LastName, FirstName, DeptNum)**

**VALUES (014, 'Smith', 'Bob', 100),**

**(086, 'Jones', 'Bill', 200),**

**(127, 'Doe', 'John', 100),**

**(192, 'Doe', 'Jane', 300);**

*Query OK, 4 rows affected (0.00 sec)*

*Records: 4 Duplicates: 0 Warnings: 0*

10. List all the data in your table:

**SELECT \***

**FROM Employee;**

*+-------------+----------+-----------+---------+*

*| employeeNum | lastName | firstName | deptNum |*

*+-------------+----------+-----------+---------+*

*| 217 | Watson | Bill | 200 |*

*| 14 | Smith | Bob | 100 |*

*| 86 | Jones | Bill | 200 |*

*| 127 | Doe | John | 100 |*

*| 192 | Doe | Jane | 300 |*

*+-------------+----------+-----------+---------+*

*5 rows in set (0.00 sec)*

11. Change the department of an employee. Note that we find the row to be changed by its Primary Key, which is guaranteed to be unique, so only a single row will be changed. (SELECT first to see what will be changed):

**SELECT \***

**FROM Employee**

**WHERE employeeNum = 217;**

**UPDATE Employee**

**SET FirstName = 'William'**

**WHERE employeeNum = 217;**

*Query OK, 1 row affected (0.00 sec)*

*Rows matched: 1 Changed: 1 Warnings: 0*

12. Delete an Employee row.

**DELETE FROM Employee**

**WHERE employeeNum = 217;**

**SELECT \***

**FROM Employee;**

*+-------------+----------+-----------+---------+*

*| employeeNum | lastName | firstName | deptNum |*

*+-------------+----------+-----------+---------+*

*| 14 | Smith | Bob | 100 |*

*| 86 | Jones | Bill | 200 |*

*| 127 | Doe | John | 100 |*

*| 192 | Doe | Jane | 300 |*

*+-------------+----------+-----------+---------+*

*4 rows in set (0.00 sec)*

13. Create a FamilyMember table:

**CREATE TABLE FamilyMember (**

**employeeNum INT,**

**famMbrId INT,**

**lastName CHAR(20),**

**firstName CHAR(20),**

**relationship CHAR(20)**

**);**

**ALTER TABLE FamilyMember**

**ADD PRIMARY KEY (employeeNum, famMbrId);**

14. Add data to FamilyMember table:

**INSERT INTO FamilyMember**

**VALUES (014, 01, 'Smith', 'Susan', 'Wife'),**

**(014, 02, 'Smith', 'Billy', 'Son'),**

**(014, 03, 'Smith', 'John', 'Son'),**

**(192, 01, 'Doe', 'Jane', 'Wife'),**

**(192, 02, 'Doe', 'Mary', 'Daughter'),**

**(428, 01, 'Smith', 'Betty', 'Wife');**

15. Query using a join over two tables:

**SELECT e.firstName, e.lastName, f.relationship, f.firstName**

**FROM Employee AS e LEFT JOIN FamilyMember AS f**

**ON e.employeeNum = f.employeeNum;**

*+-----------+----------+--------------+-----------+*

*| firstName | lastName | relationship | firstName |*

*+-----------+----------+--------------+-----------+*

*| Bob | Smith | NULL | NULL |*

*| Bill | Jones | NULL | NULL |*

*| John | Doe | NULL | NULL |*

*| Jane | Doe | NULL | NULL |*

*+-----------+----------+--------------+-----------+*

*4 rows in set (0.00 sec)*