**IS 456 Database Systems Management**

**HOP08 – Creating your Database and CRUD operations**

1/5/2019 Developed by Clark Ngo

Center for Information Assurance (CIAE) @City University of Seattle (CityU)



**Before You Start**

* Version numbers may not match with the most current version at the time of writing. If given the option to choose between stable release (long-term support) or most recent, please choose the stable release rather than beta-testing version.
* This tutorial targets Windows users and MacOS users.
* There might be subtle discrepancies along the steps. Please use your best judgement while going through this cookbook style tutorial to complete each step.
* For your working directory, use your course number. This tutorial may use a different course number as an example.
* The directory path shown in screenshots may be different from yours.
* If you are not sure what to do or confused with any steps:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Build a Database from Scratch
* Create a Schema
* Delete a Schema
* Create a Table
* Delete a Table
* Create, Read, Update, Delete (CRUD) Operations

**Resources**

* SQL Tutorial – <https://www.w3schools.com/sql/default.asp>

**Preparation**

**Run your Docker Application**

Find the Docker App and double-click

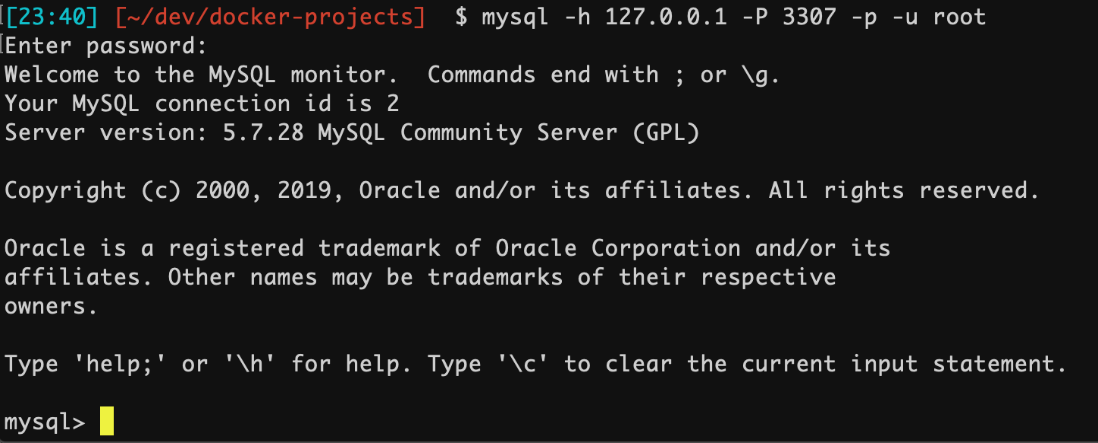
**Run an MySQL interactive shell**

Open your terminal / command prompt and type the following:

mysql -h 127.0.0.1 -P 3307 -p -u root

When prompted for password: *passwd*

Example output in MacOS:



**Use a Database**

**Syntax: USE *database\_name*;**

USE DATABASE classicmodels;

**Creating and Deleting a Database**

**Syntax: CREATE SCHEMA *database\_name*;**

CREATE SCHEMA test;

mysql> CREATE SCHEMA test;

Query OK, 1 row affected (0.01 sec)

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| classicmodels |

| mysql |

| performance\_schema |

| sys |

| test |

| world |

+--------------------+

7 rows in set (0.00 sec)

**Syntax: DROP SCHEMA *database\_name*;**

DROP SCHEMA test;

mysql> DROP SCHEMA test;

Query OK, 0 rows affected (0.01 sec)

mysql> SHOW DATABASES;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| classicmodels |

| mysql |

| performance\_schema |

| sys |

| world |

+--------------------+

6 rows in set (0.00 sec)

**Create a schema named *zoo***

**Creating and Deleting a Table**

**Syntax: CREATE TABLE *table\_name* (*attribute\_name1 DATA\_TYPE*, *attribute\_name2 DATA\_TYPE)*;**

CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);

mysql> CREATE TABLE pet (name VARCHAR(20), owner VARCHAR(20), species VARCHAR(20), sex CHAR(1), birth DATE, death DATE);

Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;

+---------------+

| Tables\_in\_zoo |

+---------------+

| pet |

+---------------+

1 row in set (0.01 sec)

mysql> DESCRIBE pet;

+---------+-------------+------+-----+---------+-------+

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

+---------+-------------+------+-----+---------+-------+

| name | varchar(20) | YES | | NULL | |

| owner | varchar(20) | YES | | NULL | |

| species | varchar(20) | YES | | NULL | |

| sex | char(1) | YES | | NULL | |

| birth | date | YES | | NULL | |

| death | date | YES | | NULL | |

+---------+-------------+------+-----+---------+-------+

6 rows in set (0.01 sec)

**Syntax: DROP TABLE *table\_name*;**

DROP TABLE pet;

**Questions you can answer for submission:**

Knowledge: What does the number in varchar do? Why choose number 20?

Knowledge: What does Null YES and Default NULL do?

**Inserting Data into a Table**

**Syntax: INSERT INTO *table\_name* VALUES(*value\_attribute1, value\_attribute2 …)*;**

INSERT INTO pet VALUES ('Papito', 'Clark', 'Chihuahua', 'm', '2007-09-30', NULL);

Query OK, 1 row affected (0.00 sec)

**Verify the Inserted Data using SELECT command**

mysql> SELECT \* FROM pet;

+--------+-------+-----------+------+------------+-------+

| name | owner | species | sex | birth | death |

+--------+-------+-----------+------+------------+-------+

| Papito | Clark | Chihuahua | m | 2007-09-30 | NULL |

+--------+-------+-----------+------+------------+-------+

1 row in set (0.01 sec)

**Verify the specific Inserted Data using WHERE command**

mysql> SELECT \* FROM pet WHERE name = 'Papito';

+--------+-------+-----------+------+------------+-------+

| name | owner | species | sex | birth | death |

+--------+-------+-----------+------+------------+-------+

| Papito | Clark | Chihuahua | m | 2007-09-30 | NULL |

+--------+-------+-----------+------+------------+-------+

1 row in set (0.00 sec)

**Questions you can answer for submission:**

Technical: What is the command for? Why would you use the command?

Knowledge: Can you insert blank values? Why?

Best practice: Why is SELECT \* not a good practice for queries?

**Updating Data**

**Syntax: UPDATE *table\_name* SET *column\_name1* = *value* WHERE *condition*;**

UPDATE pet SET species = 'dog' WHERE name = 'Papito';

mysql> UPDATE pet SET species = 'dog' WHERE name = 'Papito';

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

**Verify the Inserted Data using SELECT command**

mysql> SELECT \* FROM pet;

+--------+-------+---------+------+------------+-------+

| name | owner | species | sex | birth | death |

+--------+-------+---------+------+------------+-------+

| Papito | Clark | dog | m | 2007-09-30 | NULL |

+--------+-------+---------+------+------------+-------+

1 row in set (0.00 sec)

**Questions you can answer for submission:**

Technical: What is the command for? Why would you use the command?

Knowledge: What happens when you use UPDATE while using a WHERE that satisfies multiple rows?

Knowledge: What happens when you use UPDATE without a WHERE clause?

**Deleting Data**

**Syntax: DELETE FROM *table\_name* WHERE *condition*;**

DELETE FROM pet WHERE name = ‘Meowser’;

**Before deleting**

mysql> SELECT \* FROM pet;

+---------+-------+---------+------+------------+-------+

| name | owner | species | sex | birth | death |

+---------+-------+---------+------+------------+-------+

| Papito | Clark | dog | m | 2007-09-30 | NULL |

| Meowser | Jason | cat | f | 2001-01-22 | NULL |

| Dolly | Jake | sheep | f | 1990-04-15 | NULL |

+---------+-------+---------+------+------------+-------+

3 rows in set (0.01 sec)

**Executing the delete operations**

mysql> DELETE FROM pet WHERE sex = 'm';

Query OK, 1 row affected (0.01 sec)

mysql> SELECT \* FROM pet;

+---------+-------+---------+------+------------+-------+

| name | owner | species | sex | birth | death |

+---------+-------+---------+------+------------+-------+

| Meowser | Jason | cat | f | 2001-01-22 | NULL |

| Dolly | Jake | sheep | f | 1990-04-15 | NULL |

+---------+-------+---------+------+------------+-------+

2 rows in set (0.00 sec)

**Questions you can answer for submission:**

Technical: What is the command for? Why would you use the command?

Knowledge: What happens when you use DELETE while using a WHERE that satisfies multiple rows?

Knowledge: What happens when you use DELETE without a WHERE clause?

**Build a new Database and Tables with Primary and Foreign Keys**

**Create a schema called *donut\_shop***

**Use the *donut\_shop* database**

CREATE TABLE Customer

(

CustomerID INT NOT NULL AUTO\_INCREMENT,

FirstName varchar(20) NOT NULL,

LastName varchar(20) NOT NULL,

StreetAddress varchar(20) NOT NULL,

Apartment varchar(20) NOT NULL,

City varchar(20) NOT NULL,

State varchar(20) NOT NULL,

Zip varchar(7) NOT NULL,

HomePhone varchar(14) NOT NULL,

MobilePhone varchar(14),

OtherPhone varchar(14),

PRIMARY KEY (CustomerID)

);

CREATE TABLE Invoice

(

DonutOrderID INT NOT NULL AUTO\_INCREMENT,

CustomerID INT NOT NULL,

OrderDate Date NOT NULL,

Notes varchar(255),

PRIMARY KEY (DonutOrderID),

FOREIGN KEY (CustomerID) REFERENCES Customer (CustomerID )

);

CREATE TABLE Donut

(

DonutID INT NOT NULL AUTO\_INCREMENT,

DonutName varchar(20) NOT NULL,

Description varchar(30) NOT NULL,

UnitPrice decimal(5,2) NOT NULL,

PRIMARY KEY (DonutID)

);

CREATE TABLE InvoiceLineItem

(

DonutOrderID INT NOT NULL,

DonutID INT NOT NULL,

Qty INT NOT NULL,

PRIMARY KEY (DonutOrderID,DonutID),

FOREIGN KEY (DonutOrderID) REFERENCES Invoice (DonutOrderID) ,

FOREIGN KEY (DonutID) REFERENCES Donut (DonutID )

);

**Show the tables of the database**

**Questions you can answer for submission:**

Technical: What is the command for? Why would you use the command?

Knowledge: What’s the use of REFERENCES? Why do you need it?