ICE – Database I – Introduction to SQL

ICE Purpose:

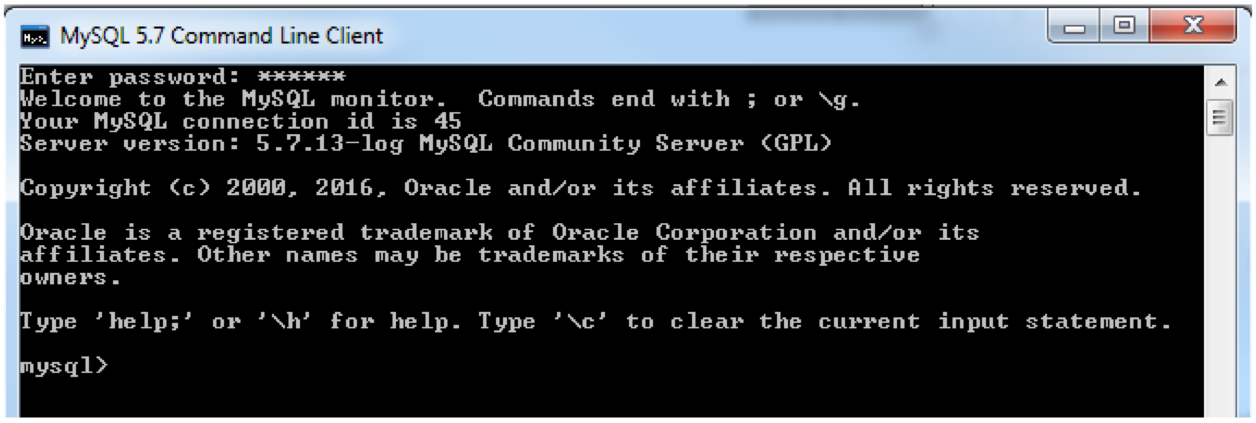
The purpose of this ICE is to introduce students to the syntax and functions of Structured Query Language (SQL). The students will specifically learn MySQL running on a Windows platform. The students will learn the basic commands of SQL to query a table, modify the structure of the table, insert data into a table, and delete data from a table.

ICE Assumption:

This ICE assumes that students have already installed MySQL, to include MySQL Workbench, on their laptops. A second assumption is that a database named “mydb” has been created on their laptops (localhost).

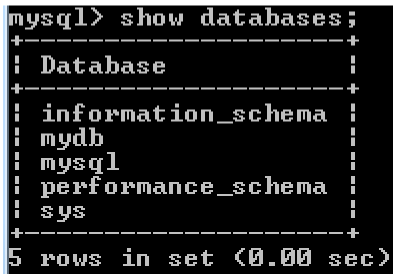
Launch MySQL Command Line:

Click on the Start Button and select MySQL > MySQL Server 5.7 > MySQL Server 5.7 Command Line Client. Enter the password, **sqldba**, to connect to your local MySQL server. You should see the following command window:



```
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.7.13-log MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

To see what databases are available, type in **SHOW DATABASES;** NOTE: a SQL statement can be split among multiple lines, but a SQL statement must end with a semicolon.



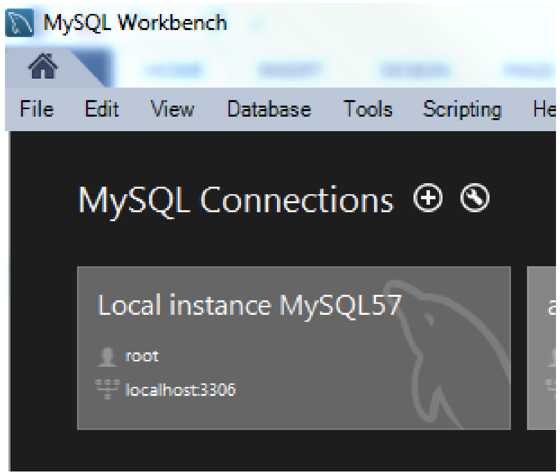
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

We will create a new database named **it355_ice**. Type in the command **CREATE DATABASE it355_ice;** Once again type in **SHOW DATABASES;** to see the new database. Type in **USE**

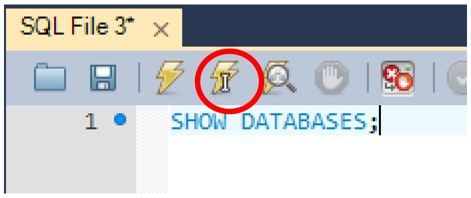
it355_ice; We are now using the mydb database. To see what tables are in the database type in **show tables;** You should get an empty set because there are currently no tables in the database.

<u>Launch MySQL Workbench:</u>

Click on the Start Button and select MySQL > MySQL Workbench. Once MySQL Workbench launches, click on "Local instance" under the MySQL Connections.



You will then see the default layout of MySQL Workbench with a blank query frame. In the blank, type in **SHOW DATABASES;** just like you did in the command line interface. Then either click on the lightning bolt with superimposed cursor or press Control-Enter together to execute just the SQL statement that your cursor is located at. We will use this icon or Control-Enter as we assemble SQL statements.



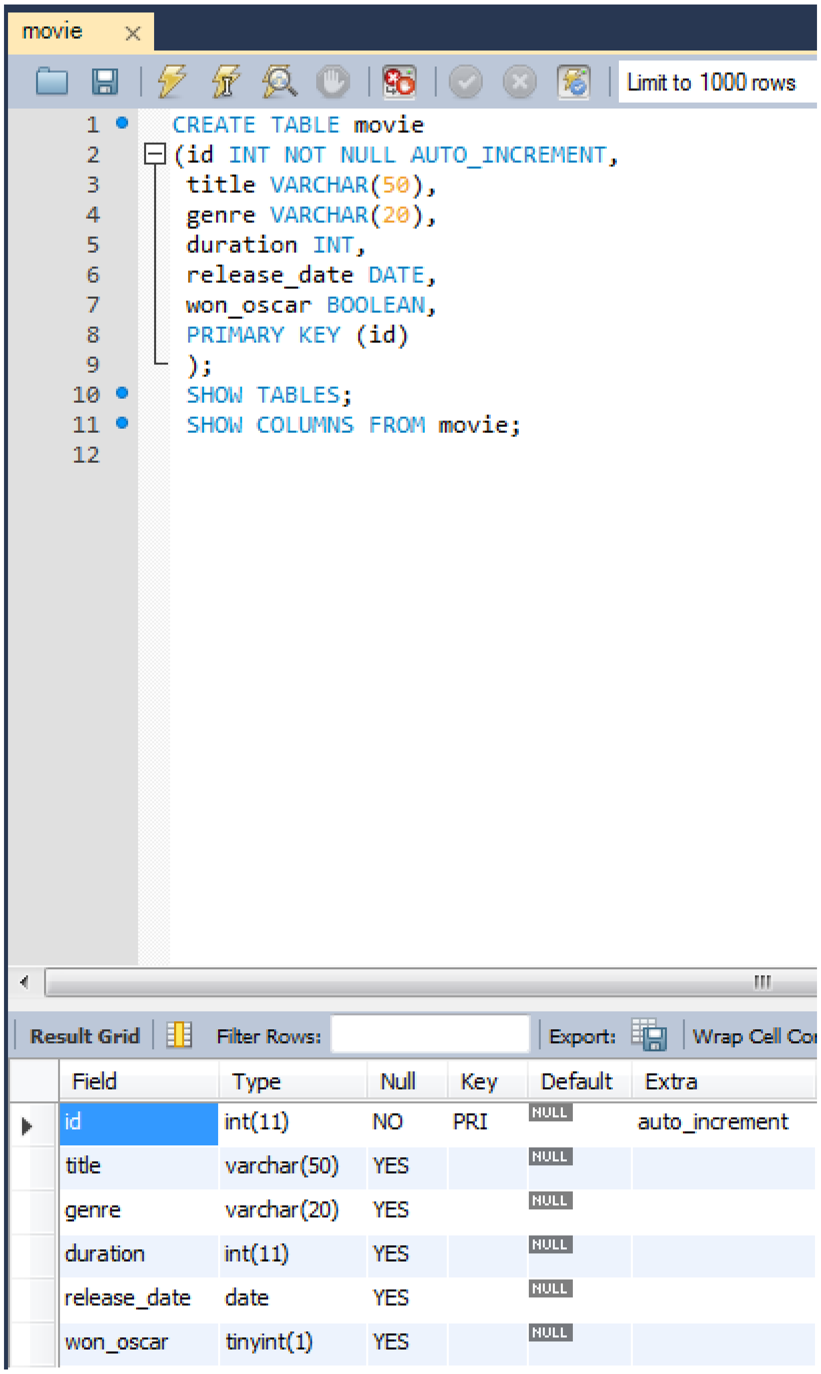
```
SHOW DATABASES;
```

Notice that you get the same table of databases like the command line. So we can use either the MySQL Command Line Interface or MySQL Workbench interchangeably. For the remainder of this ICE, we will use the MySQL Workbench. On the next line, type in **USE it355_ice;**

<u>Importing a Table:</u>

We will first create an empty table named movie that we want to import data into. Click the Open Script (Folder) icon and navigate to movie.sql. Execute the query. Now, execute **SHOW TABLES;** in both the MySQL Workbench and MySQL Command Line. In the MySQL

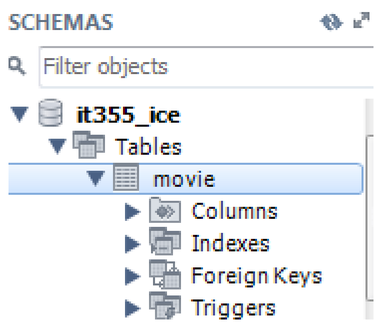
Workbench, execute **SHOW COLUMNS FROM movie;** to inspect the columns and associated data types for each column.



```
CREATE TABLE movie
(id INT NOT NULL AUTO_INCREMENT,
 title VARCHAR(50),
 genre VARCHAR(20),
 duration INT,
 release_date DATE,
 won_oscar BOOLEAN,
 PRIMARY KEY (id)
 );
 SHOW TABLES;
 SHOW COLUMNS FROM movie;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
title	varchar(50)	YES		NULL	
genre	varchar(20)	YES		NULL	
duration	int(11)	YES		NULL	
release_date	date	YES		NULL	
won_oscar	tinyint(1)	YES		NULL	

In the left hand column, right click on **movie** under **Tables** under **it355_ice** (you may have to refresh the folders to see the new table). Then select **Table Data Import Wizard**. Navigate to your turn-in folder and select **movie.csv** and click **Next**. Select the option to use existing table named movie and click **Next** until the import is executed.



The Basics of a SQL Query (SELECT and FROM)

Now, let's inspect imported data with our first SELECT command. Execute the command

SELECT *
FROM movie;

to return all the data in the table. The * is a wildcard that means all. In this case, we are selecting to return all data in all columns in movie.

We can specify which columns that we want to return such as movie and genre with the command

SELECT title, genre
FROM movie;

Creating an Alias using AS

We can also rename the column headings in the resulting table using the AS keyword such as

SELECT title AS 'Movie Title', genre AS Genre
FROM movie;

This is called aliasing. Notice that you have to use quotation marks set a column heading with multiple words. You can also omit the AS to shorten the command such as

SELECT movie 'Movie Title', genre Genre
FROM movie;

Returning DISTINCT Values

Execute
SELECT genre
FROM movie;

You will see in the resulting table there are multiple duplicate records. In order to return only the distinct (or different) records, you must use the DISTINCT keyword. Now execute

```
SELECT DISTINCT genre
FROM movie;
```

Filtering Results Using WHERE

Often when we are running queries, we only want to return records that meet a certain condition. The other records can just be filtered out. We do this by adding the WHERE keyword to our SQL query. For example, if we only want to see the list of movies that won an Oscar then we would execute

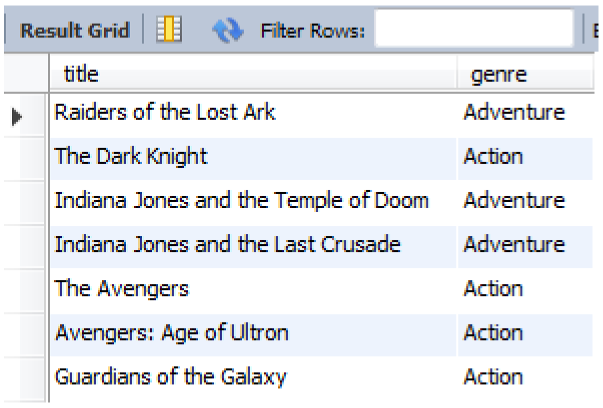
SELECT title
FROM movie
WHERE won_oscar = TRUE;

We can also filter using multiple conditions by using AND and OR keywords. For example, to return a list of animated movies that won an Oscar, we would execute

SELECT title
FROM movie
WHERE won_oscar = true
AND genre = 'animation';

Notice that the value 'animation' is in quotes because it is a string of characters. Likewise, if we wanted to watch an action or adventure movie we could execute the following query:

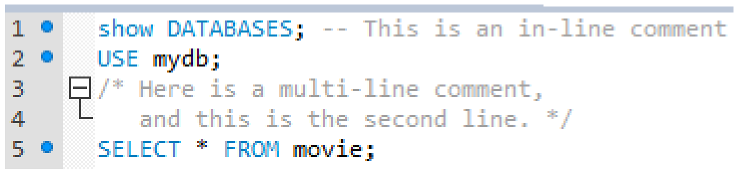
SELECT title, genre
FROM movie
WHERE genre = 'action'
OR genre = 'adventure';



title	genre
Raiders of the Lost Ark	Adventure
The Dark Knight	Action
Indiana Jones and the Temple of Doom	Adventure
Indiana Jones and the Last Crusade	Adventure
The Avengers	Action
Avengers: Age of Ultron	Action
Guardians of the Galaxy	Action

Comments in SQL

Use -- for an in-line comment; use /* */ for a multi-line comment.



```
show DATABASES; -- This is an in-line comment
USE mydb;
/* Here is a multi-line comment,
   and this is the second line. */
SELECT * FROM movie;
```