INFO90002 Week 8 Tutorial

Objectives:

In this week's lab you will learn to:

- CREATE tables using SQL Data Definition Language (DDL)
- INSERT UPDATE and DELETE using SQL Data Manipulation Language (DML)
- CREATE foreign key constraints
- Test constraint behaviour when deleting and updating rows
- DROP tables using SQL DDL
- Test how REFERENTIAL INTEGRITY and FOREIGN KEYS work

1. PREREQUISITE:

Please ensure that you have disabled SAFE UPDATES in the Preferences

ON Windows MySQL Worksheet

EDIT→ PREFERENCES→SQL EDITOR->OTHER

Uncheck the safe update check box.

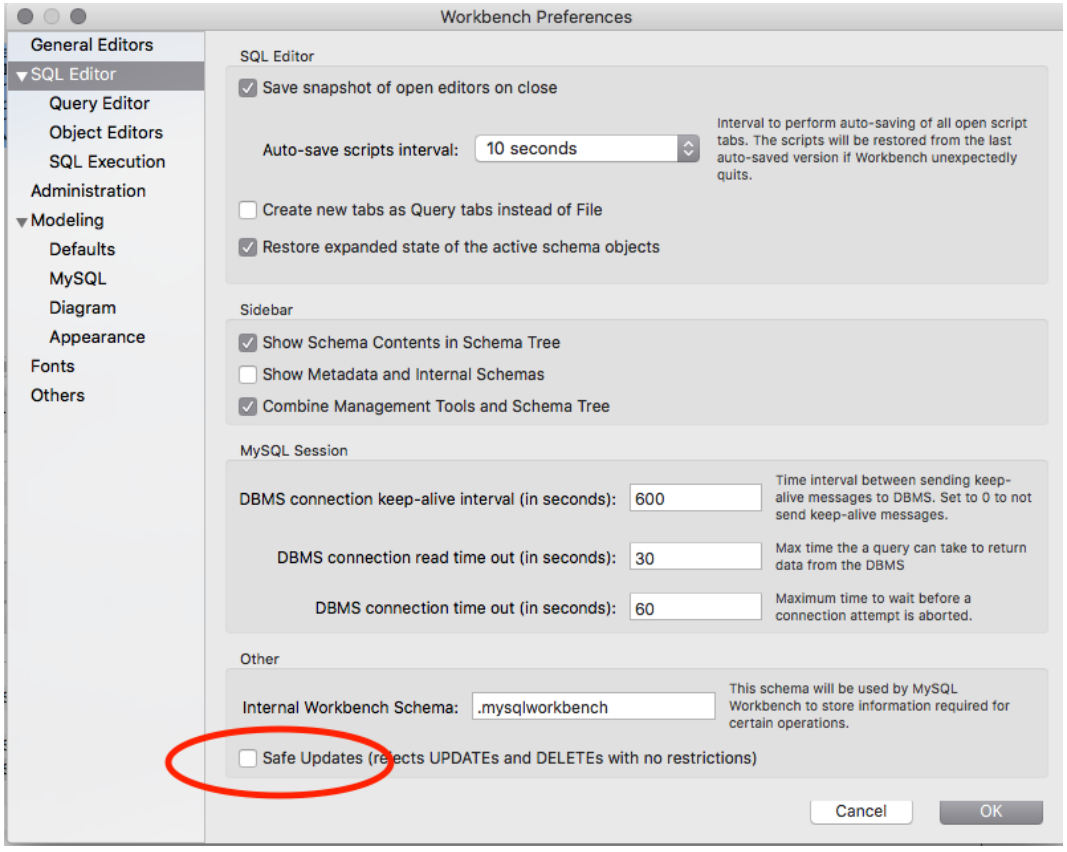

Figure 1 Screenshot of MySQL Workbench SQL Editor Preferences window

2: DDL & DML

In this week's lab we will be learning Data Definition Language (DDL) and Data Manipulation Language (DML).

DDL

The first statement is the CREATE TABLE statement. The CREATE TABLE statement is part of DDL that will create objects in the database. The syntax for the create table command is shown below:

```
CREATE TABLE tablename
(field1 datatype [modifier],
field2 datatype [modifier],
…,
PRIMARY KEY (identifier)
);
```

NOTE:

- tablename is the name of the table
- field1, field2 etc are the names of the fields in the table.
- datatype is the type of the field.
- modifier [ ] is an optional value, and changes how the field operates in the database.

2.1 Start MySQL Workbench and connect to your MySQL database

Use instructions from week 1 Lab if you have not connected to the MySQL database before.

2.2 Type the CREATE statement for a table called customer containing the following:

FieldName	DataType	Modifier
CustomerID	SMALLINT	AUTO_INCREMENT
FirstName	VARCHAR(25)	
LastName	VARCHAR(30)	NOT NULL
Phone	VARCHAR(16)	NOT NULL
BirthDate	DATE	NOT NULL

```
CREATE TABLE customer
(CustomerID smallint auto_increment,
 FirstName varchar(25),
 LastName varchar(30) not null,
 Phone varchar(16) not null,
 BirthDate date not null,
 Primary Key (CustomerID));
```

2.3 Execute the SQL DDL statement.

HINT: To run the statement press the first lightning symbol in the Toolbar

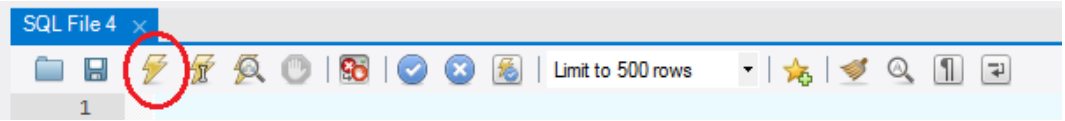

Figure 2 The Lightning button executes all statements in the query window

2.4 Did you see anything change?

HINT: You may have seen a "Query Completed" in the bottom left of the window.

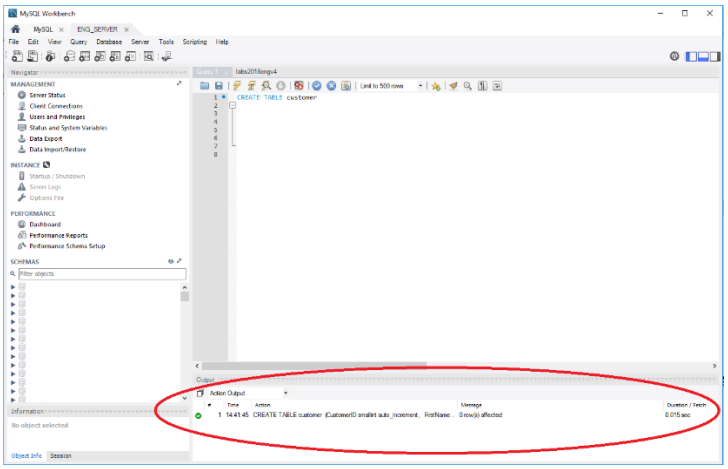

Figure 2: The Action Output (circled in red) will show the result of all statements. A green tick indicates the statement was successfully executed.

MySQL Workbench only displays output of SELECT statements.

Confirm the table has been created

```
DESC customer
```

	Field	Type	Null	Key	Default
▶	CustomerID	smallint(6)	NO	PRI	NULL
	FirstName	varchar(25)	YES		NULL
	LastName	varchar(30)	NO		NULL
	Phone	varchar(16)	NO		NULL
	BirthDate	date	NO		NULL

Figure 3 The meta data for customer table

SQL DML Insert, Update and Delete

Once the customer table is created in the database, the next step is to insert data into the table. To insert data into a database we use the DML INSERT statement.

INSERT

The format of the INSERT statement is:

```
INSERT INTO tablename VALUES
(val1, val2, val3, val4, …),
(val1, val2, val3, val4, …);
```

This INSERT syntax allows the insert of many rows at once.

2.5 Use the example of the INSERT statement to enter the following data into the customer table:

CustomerID	FirstName	LastName	Phone	BirthDate
DEFAULT	Jon	Smith	22522255	1985-12-5
DEFAULT	An	Hone	22454852	1956-1-22
DEFAULT	Jun	De	26854582	1984-6-6
DEFAULT	Kent	King	12556842	1971-5-9
DEFAULT	Barb	Smith	33665252	1969-6-12
DEFAULT	NULL	Senti	75584252	2000-2-25

Table 2: Information for each row to be inserted into the customer table

You will need to think carefully when you are crafting the SQL statement. The order of the data in this version of the INSERT statement must be the same as the table, and *no missing fields are allowed*. Think about the data types you are entering via this insert into the table. String and Date data types are usually enclosed in quotes (e.g. '1972-11-11' for a date data type).The data type chosen will affect how you craft your statement.

```
INSERT into customer values
(default, 'Jon', 'Smith', 22522255, '1985-12-05'),
(default, 'An','Hone', 22454852, '1956-01-22'),
(default, 'Jun', 'De',26854582,'1984-06-06'),
(default, 'Kent', 'King', 12556842, '1971-05-09'),
(default, 'Barb', 'Smith', 33665252, '1969-06-12'),
(default, NULL, 'Senti', 75584252, '2000-02-25');
```

2.6 Once you think you have written the SQL INSERT statement, run the SQL again using the lightning bolt.

> *NOTE: If you still have the previous create statement in the query window, you will get an error as clicking the lightning bolt runs all code in the window. To only run the current statement, make sure the cursor is on the INSERT statement and click on the 2nd lightning bolt.*

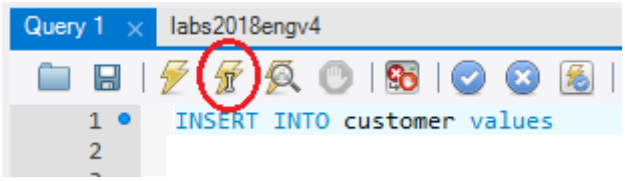

Figure 4: The second lightning bolt with I runs only the code where the cursor is located

2.7 Did your INSERT statement succeed or did you receive an error message?

HINT: Once again look at what happens – inspect the 'Action Output' window. You should see a message stating that there were 6 row(s) affected.

UPDATES & DELETES

We may need to update or delete rows from the table if the information changes.

A typical update statement format is shown below:

```
UPDATE tablename
SET columnname = <new value>
WHERE (condition clause);
```

2.8 Update the customer table by adding Senti's first name “Yoshi”.

```
UPDATE customer
SET firstname = 'Yoshi'
WHERE lastname = 'Senti';
```

2.9 Delete Kent King from the customer

```
DELETE FROM customer
WHERE firstname = 'Kent'
and lastname = 'King';
```

2.10 Confirm Kent King has been deleted and Yoshi Senti’s name is correct.

```
SELECT firstname, lastname
FROM customer
WHERE firstname = 'Yoshi' and lastname = 'Senti'
OR
firstname = 'Kent' and lastname = 'King';
```

3. Referential Integrity

3.1 On LMS in the LABS (practical) folder and download the SQL script *scott.sql* and install the Scott-Tiger tables by running the script scott.sql from MYSQL Workbench

FILE -> OPEN SQL SCRIPT

The "Scott-Tiger" script has installed 4 tables and populated three of them, EMP, DEPT and SALGRADE. However, the install script has not added the foreign key relationship between EMP & DEPT table using the deptno column.

3.2. Add the Foreign Key constraint in the EMP table which references the Primary Key in the DEPT table.

```
ALTER TABLE EMP
ADD CONSTRAINT FK_DEPT
FOREIGN KEY (deptno) REFERENCES DEPT(deptno)
ON DELETE CASCADE
ON UPDATE CASCADE
;
```

Now query the information_schema (the meta data also known as a data dictionary) to see the constraints

```
SELECT *
FROM information_schema.table_constraints
WHERE TABLE_NAME in ('emp', 'dept');
```

The organisation has decided to invest in the movie business and has hired the director Stephen Soderbergh

3.3. Write an insert statement to add a row to the EMP table.

Column	Value:
empno	8182
name	SODERBERGH
job	DIRECTOR
boss	7839
hiredate	2012-08-25
salary	8000
comm	NULL
deptno	NULL

HINT: Insert statements look something like this

```
INSERT INTO TABLE
VALUES (Col1, Col2, Col3, Col4, Col5, Col6, Col7, Col8);
```

Answer:

```
INSERT INTO EMP
VALUES (8182, 'SODERBERGH', 'DIRECTOR', 7839, '2012-08-25',8000,
NULL, NULL);
```

Note that currently we have not assigned a department number (deptno) to Steven Soderbergh.

A decision has been made to assign Steven Soderbergh to the new MOVIES department.

3.4. Update Steven Soderbergh's record in the EMP table so that he is in deptno 50.

What happened? Why did it happen? What is Steven Soderbergh's current department number in the EMP table?

```
UPDATE EMP
SET DEPTNO=50
WHERE NAME = 'SODERBERGH';
```

A referential integrity error occurred. The Foreign Key constraint did not allow the update as there is no department 50 in the DEPT table

Add the MOVIE department to the DEPT table:

3.5. Insert the following values into the DEPT table

Column Name:	Value:
deptno	50
name	MOVIE
location	HOLLYWOOD

```
INSERT INTO DEPT
VALUES (50,'MOVIE','HOLLYWOOD');
```

3.6. Now repeat question 5 and update Soderbergh's department to 50. Did this succeed? Why did it succeed?

It succeeded because there is a Primark Key row in DEPT table that has a value that matches the Foreign key value in the EMP table.

3.7. Add the Boss/Employee Foreign Key to the EMP table. Name the relationship FK_Boss

HINT: Refer to the syntax in Q2.3 to write the Boss Employee foreign key / primary key relationship.

```
ALTER TABLE EMP
ADD CONSTRAINT FK_BOSS
FOREIGN KEY (boss) REFERENCES EMP(emp)
ON DELETE CASCADE
ON UPDATE CASCADE
;
```

Query the information_schema meta data to see the table constraints on the EMP & DEPT tables

```
SELECT *
FROM information_schema.table_constraints
WHERE TABLE_NAME in ('emp', 'dept');
```

The business has recruited its first actor Jennifer Lawrence and Jennifer will be making a movie with Steven Soderbergh.

3.8 Write an INSERT statement to add Jennifer Lawrence's information to the EMP table

Column	Value:
empno	8385
name	LAWRENCE
job	ACTOR
boss	8182
hiredate	2012-08-28
salary	8500
comm	NULL
deptno	50

```
INSERT INTO EMP
VALUES (8385, 'LAWRENCE', 'ACTOR',8182, '2012-08-28',8500, NULL,
50):
```

Confirm that both Soderbergh and Lawrence are both in department 50 'Hollywood'.

3.9 Write a query.to retrieve the employee number, name, their manager's employee number, department number and their department name for all employees in department 50.

```
SELECT empno, emp.name, boss, deptno, dept.name
FROM EMP inner join DEPT
ON Emp.Deptno = Dept.Deptno
WHERE Emp.Deptno = 50;
```

The President Mr King has decided to that number 50 is bad luck and want's the movie department to be department number 60.

3.10 Update the MOVIE department in the DEPT table so it is now department number (deptno) 60.

```
UPDATE DEPT
SET DEPTNO = 60
WHERE DEPTNO=50;
```

3.11 Rerun the query you wrote in 2.10. Is there any differences to your result set? Why?

HINT: Because we had cascade on the foreign key constraint the change to the primary key has been applied to all foreign keys referencing the primary key.

```
SELECT empno, emp.name, boss, deptno, dept.name
FROM EMP inner join DEPT
ON Emp.Deptno = Dept.Deptno
WHERE Emp.Deptno = 50;
```

There are no rows returned. This is because the constraint has updated the foreign key records in the EMP table

3.12 Change your query so that you return all employees in department (deptno) 60.

```
SELECT empno, emp.name, boss, deptno, dept.name
FROM EMP inner join DEPT
ON Emp.Deptno = Dept.Deptno
WHERE Emp.Deptno = 60;
```

Stephen Soderbergh has been fired over creative differences with the president of the company Mr King. Mr King has demanded that Soderbergh be deleted from the EMP table.

3.13 Write the delete statement to remove Soderbergh from the EMP table

What happened? Why did it happen?

```
DELETE FROM EMP
WHERE Name = 'SODERBERGH';
```

An error occurred because there are foreign key records referencing the Soderbergh record.

3.14. Update the record or records that need to be changed before we can delete Soderbergh from the Emp table, then delete Soderbergh from the EMP table.

```
UPDATE EMP
SET BOSS=7839
WHERE NAME = 'LAWRENCE'
COMMIT;
DELETE FROM EMP
WHERE Name = 'SODERBERGH';
```

3.15 USING DDL Drop the DEPT, EMP, SALGRADE and BONUS tables in that order

```
DROP TABLE DEPT;
DROP TABLE EMP;
DROP TABLE SALGRADE;
DROP TABLE BONUS;
```

END OF TUTORIAL