Lesson 7-CRUD Functionality

Lesson 7 Concepts

1. SQL provides commands for creating and modifying databases, tables and data.
2. The DDL commands can create and modify *databases* and *tables*.
3. The DML commands can create and modify *data* in tables.
4. DML commands provide CRUD functionality.

SQL Provides Commands for Creating and Modifying Databases, Tables and Data

Beyond reading data from a database, SQL commands have the ability to create, alter and remove databases, tables, and the data within them. Although you will see that there is overlap in some of the commands, they are grouped according to their area of concern as Data Definition Language (DDL), or, Data Manipulation Language (DML).

The DDL Commands Can Create and Modify Databases and Tables

Data Definition Language (DDL) is the subset of SQL commands responsible for interacting with databases and their tables – as opposed to the *data* within the tables. With these commands you can create new databases or tables, alter existing ones, or remove them. These command statements are standardized and are (often) simpler than a **SELECT** statement. For example, to create a database:

```
CREATE DATABASE nameofnewdatabase;
```

Creating a new table requires more work as you need to specify the configurations for each column, that is, the names, data types, and modifiers for each column.

```
CREATE TABLE nameofnewtable (in parentheses specify the name and details of each column)   specify type of DB;
```

WARNING: Deleting tables is as easy as creating them!! Using the **DROP** command to delete…

```
DROP DATABASE nameofdatabase; DROP TABLE nameoftable;
```

The above commands are irreversible – no ctrl/cmd Z for your database! As such, you would not give permission for this command to all users (more on that later in the term).

Despite our best efforts to plan our database in advance, occasions will arise where we will need to modify the structure of a table after it has been created and had data added to it. The **ALTER** command allows us to add, remove, or modify columns and their structure.

The DML Commands Can Create and Modify Data in Tables

Data Manipulation Language (DML) is the subset of SQL commands responsible for querying the database and manipulating the data within the tables. With these commands you can add new rows, delete rows, modify rows, or create data sets from the tables – this is the group that **SELECT** belongs to. While **SELECT** retrieves data without modifying the actual data, the following commands change the data in the database, and can add or remove rows from tables.

DELETE

The simplest command is the **DELETE** command (should deleting be the easiest thing to do???):

```
DELETE FROM tablename WHERE criteria;
```

This command will remove the specified row(s) from the specified table(s) – (NB: the **DELETE** command also has the ability to delete from multiple tables in one command).

```
DELETE FROM employees WHERE id = 113;
```

UPDATE

To modify an existing row we use the **UPDATE** and **SET** commands, specifying which row(s) we want updated, and what the new value(s) should be:

```
UPDATE tablename SET columnname = newvalue WHERE
criteria;
```

To update the role property of our pet store employee…

```
UPDATE employees SET role = “Assistant” WHERE id = 112;
```

INSERT INTO

To add new rows we use the **INSERT INTO** command, specifying the columns and the values to insert:

```
INSERT INTO tablename (columns) VALUES (valuetoinsert);
```

NB: If you are adding data to all of the columns, you do not need to specify the columns, the values are assumed to be in the correct order. To add a new pet store sale…

```
INSERT INTO sales (date, item, employee)

VALUES (‘2021-06-30’, 1017, 112);
```

The reason we didn’t add the *id* column value is because we set this column to auto-increment when we built the table. This will automatically add the last value inserted in this column plus 1, and means we don’t have to provide the value in our command.

We can also add multiple values with one **INSERT INTO** command, with the values for each row in their own parentheses separated by commas. If you look at the file that I gave you to create the pet store tables, you will see a large **INSERT INTO** command in there:

```
INSERT INTO sales (id, date, item, employee) VALUES
(21000, '2021-06-09', 1002, 111),
(21001, '2021-06-09', 1003, 111),
(21002, '2021-06-09', 1004, 111),
(21003, '2021-06-10', 1005, 100),
(21004, '2021-06-10', 1011, 114);
```

Beyond the technical manipulation of data, these **INSERT**, **UPDATE**, and **DELETE** DML commands provide the tools that we need to create a dynamic database-driven web application.

DML Commands Provide CRUD Functionality

Websites, for the most part, have evolved to the point where "static" hand-coded HTML and content can be considered obsolete (in technology, not content). More often than not, websites receive content from a database and provide the ability to modify this content through a user interface. A web application that has this ability can be described as having CRUD functionality. This (unpleasant) acronym stands for **C**reate, **R**ead, **U**pdate, and **D**elete, and simply means that your application includes an interface that can modify the content stored in the database. All social media applications have this functionality, you can log in, add or edit content, and your "page" shows your new or updated content – without you typing a single line of HTML or CSS. In SQL, the DML commands discussed here line up with the CRUD terminology:

CRUD Terminology	SQL DML COMMAND
Create	INSERT INTO
Read	SELECT
Update	UPDATE
Delete	DELETE

Mastering the DML category of SQL commands means providing a robust and sophisticated CRUD functionality for your web applications.

References

Thomson, L., & Welling, L. (2003). *MySQL tutorial*. Sams.

Chapter 4: Creating Databases, Tables and Indexes (pp. 41-51, 55-56)
Chapter 5: Inserting, Deleting and Updating Data (pp. 65-76)

The above reference is available through the Humber Library (library.humber.ca).