Activity – Working with Relational Databases

Activity One – Considering Relational Database Tables

In this activity, you will be working with several database tables that contain data for a business.

Go to https://www.w3schools.com/sql/

And click on the green "Try it Yourself" button (or go directly to this page using https://www.w3schools.com/sql/trysql.asp?filename=trysql_select_all)

On the right-hand side of this screen, you will see a list of eight database tables:

Your Database:

Tablename	Records
Customers	91
Categories	8
Employees	10
OrderDetails	518
Orders	196
Products	77
Shippers	3
Suppliers	29

One at a time, click on the links in that list (in the order given below rather than the alphabetical order in which they are listed) and consider the data in the overall database.

1) What data is stored in the **Customers** table?

2) What data is stored in the **Categories** table?

3) What data is stored in the **Employees** table?

4) What data is stored in the **Suppliers** table?

5) What data is stored in the **Products** table?

6) What data is stored in the **Orders** table? What attributes (columns) in this table are unique to this table? What attributes are linked to data in other tables?

7) What data is stored in the **OrderDetails** table? What attributes (columns) in this table are unique to this table? What attributes are linked to data in other tables?

Activity Two – Using SELECT queries.

The SELECT command is arguably the most basic of the commands. It allows you to view all or a subset of the columns in a database. When you were clicking on the links in the previous activity you were actually submitting very basic queries to the database. In this activity we will look at how to use the SELECT command.

Click on the Customers link on the right-hand list. Observe that the text box under **SQL Statement:** changes to read

SELECT * FROM [Customers]

There are two parts to this command that you can modify to change the results of the query:

1. The label in the square brackets can refer to any table in the database. [And the square brackets are actually optional]
    - Modify the command above by replacing Orders with Categories.
        - Press the Green **Run SQL** button to issue the command and see the results.
    - Modify the command above by replacing Orders with Employees.
        - Again, don't forget to run the command.

2. The * symbol is what is known as a "wild card" and it basically asks the computer to select every column from the Orders table. But you can replace this symbol with specific column name(s) to select only a subset of the table.
    - Modify the original command above by replacing the * with ContactName
        - This will show you only the list of all of the contacts in the Customers table.

3. If you want to see more than one column you can list multiple column headers, separated by a comma:
    - Type in the following and press "Run SQL"
        - SELECT ContactName,Address FROM Customers

Some data is very repetitive and what we get from a base SELECT query isn't quite what we want.

4. Type in the following query:
    - SELECT Country FROM Customers
    - What do you notice about the results?

5. In many situations we only want to see an individual country listed once. To accomplish this, we can submit the following query:
    - SELECT DISTINCT Country FROM Customers
    - How is this result different from the previous result?

Each of the SELECT queries we have looked at previously have provided results for every row in the table. But sometimes we want to look at only certain rows that meet certain criteria.

Type in the following queries. Describe the results for each.

a. SELECT ContactName, Address, City FROM Customers where Country="Germany"

b. SELECT OrderID FROM OrderDetails where ProductID=51 and Quantity>=20

c. SELECT * FROM Employees where Notes LIKE "%BA%"

d. SELECT * FROM Employees where Notes LIKE "% BA %"

Activity Three – Using JOIN queries.

The JOIN operator is a way to combine data from multiple tables into one, aggregated, result.

Type in the following queries. Describe the results for each.

- SELECT * FROM Orders INNER JOIN Employees ON Orders.EmployeeID=Employees.EmployeeID

- SELECT Orders.OrderID, Orders.CustomerID, Employees.LastName, Employees.FirstName FROM Orders INNER JOIN Employees ON Orders.EmployeeID=Employees.EmployeeID

- SELECT Products.ProductName,Categories.CategoryName FROM Products INNER JOIN Categories ON Products.CategoryID=Categories.CategoryID