## Instructions:

* The assignment aims to reinforce SQL query skills using the **WideWorldImporters** database. You will write and execute SQL queries to retrieve and manipulate data using various SQL constructs covered in Chapter 13 of our SQL text.
* Write SQL queries that fulfill the requirements listed below.
* Use appropriate column names, functions, and sorting techniques where necessary.
* Ensure queries return meaningful results based on the dataset.
* Save your work as a .sql file and upload it to D2L Dropbox folder for the assignment. Use the following file naming convention: *yourFirstName\_yourLastName\_SQL\_HW\_08.sql*
* Include DROP VIEW IF EXISTS statements before each CREATE VIEW to avoid errors on re-runs.
* Do not modify any original records in the database.
* For update/insert/delete exercises, use only test data that you insert yourself.

Problem 1: Create a Simple View

**Objective:** Understand how to define a basic view.  
**Task:** Create a view named vw\_CustomerContacts that includes the CustomerID, CustomerName, and PhoneNumber from the Sales.Customers table.

Problem 2: Restrict Access via View

**Objective:** Learn how to use views to restrict sensitive columns.  
**Task:** Create a view named vw\_PublicSuppliers that returns only the SupplierID, SupplierName, and PhoneNumber from the Purchasing.Suppliers table, excluding credit and bank-related information.

Problem 3: Create View with a Join

**Objective:** Demonstrate joining tables inside a view.  
**Task:** Create a view named vw\_InvoiceDetails that joins Sales.InvoiceLines and Sales.Invoices, returning InvoiceID, InvoiceDate, StockItemID, and Quantity.

Problem 4: Create a View with TOP and ORDER BY

**Objective:** Create views that use ordering and row limits.  
**Task:** Create a view vw\_Top10ExpensiveItems that shows the top 10 most expensive stock items by UnitPrice from Sales.InvoiceLines.

Problem 5: View with UNION ALL and Column Aliases

**Objective:** Combine multiple data sources into one view.  
**Context:** You want to create a unified view of people who are either suppliers or customers.  
**Task:** Create a view vw\_Contacts that combines customer and supplier contact info (name and phone), with a column indicating their role.

Problem 6: Create Summary View with SCHEMABINDING

**Objective:** Build a summary view that can’t be broken by schema changes.  
**Task:** Create a view vw\_SalesSummary that summarizes total invoice amounts per customer using Sales.Invoices, with the WITH SCHEMABINDING option.

Problem 7: Create an Updatable View

**Objective:** Understand conditions for view updateability.  
**Task:** Create a view vw\_MyTestCustomers that shows CustomerID, CustomerName, and PhoneNumber from Sales.Customers. Then insert a new row into the view.

Problem 8: Create a Read-Only View

**Objective:** Prevent data modifications via a view.  
**Task:** Create a view vw\_ReadOnlyCustomers based on Sales.Customers that includes a computed column CustomerInfo = CustomerName + PhoneNumber.

Problem 9: Modify and Delete Views

**Objective:** Manage view lifecycle.  
**Task A:** Create a view vw\_StaffEmails listing PersonID, FullName, and EmailAddress from Application.People.  
**Task B:** Modify the view to include only staff with non-null email.  
**Task C:** Drop the view.

Problem 10: Use Views and Catalog View

**Objective:** Use metadata views to explore schemas and update through a view.  
**Task A:** Select name and schema of all views using sys.views and sys.schemas.  
**Task B:** Update a row using a view and enforce integrity with WITH CHECK OPTION.

**Grading Rubric**

Each query is worth \*\*10 points\*\*, evaluated based on the following criteria:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Excellent (10 - 9 pts)** | **Good (8 - 7 pts)** | **Satisfactory (6 - 5 pts)** | **Needs Improvement (4 - 0 pts)** |
| **Correctness of Query (50%)** | Query executes without errors and returns the expected result set. | Query executes with minor logical errors but mostly correct. | Query contains errors affecting correctness. | Query does not execute properly or is incorrect. |
| **Use of Required SQL Concepts (30%)** | Successfully applies the required SQL concepts. | Uses most required SQL concepts correctly. | Uses some SQL concepts but omits key elements. | SQL concepts are misused or omitted. |
| **Query Formatting & Readability (10%)** | Query is well-structured and properly formatted. | Query is readable but could use better formatting. | Query is difficult to read with inconsistent formatting. | Query lacks readability and proper structure. |
| **Use of Proper Column Naming and Aliasing (10%)** | All columns are named meaningfully. | Most columns are appropriately named. | Some columns are meaningfully named. | Column naming and aliasing are unclear or missing. |