Project | Introduction to SQL

Project Overview

This project is focused on managing and manipulating relational databases using SQL.

This project is an excellent opportunity to showcase the skills you are learning in SQL. Therefore, you are encouraged to not only focus on getting the correct answer but also to focus on your submission appearance. Once you score 90% or higher on this project, you can submit your project to your professional portfolio in Wix, which you can later share throughout your professional journey as you apply for promotions or new jobs. Therefore, professional appearance should also be a priority in your submission.

Project Resources

To complete this assignment, log into the virtual machine and use the Adventure Works database within SQL Server Management Studio.

Format & Submission

Your submission should include two (2) files:

- The SQL script copied over into a Microsoft Word document. Name this file "Project4_SQLscript_LastName.docx."
- Screenshots of each SQL script showing the query and the results. Name this file "Project4_SQLscreenshots_LastName.docx."

Note the start of the SQL code for each question with the appropriate part and question number, such as:

/*Question A1*/

Also, not that you should use aliases for this project to keep the queries more straightforward.

Project Instructions

Part A

1. Create a SQL script that selects ALL the BusinessEntityID, LastName, and FirstName fields for the Person's table. Make sure the fields are in this order.
2. Add a sort to the query in A1 so that LastName is alphabetized A - Z.
3. Create a SQL script that adds the person's email address to query #A2.
4. Create a SQL script that adds the person's phone number to query #A3, including the PhoneNumberType Name field (e.g., cell, home, work).
5. Create a SQL script that adds the AddressLine1, City, StateProvinceID, and PostalCode fields to query #A4.
    - *Note 1*: two different people could have the same address. This database is designed to store each unique address individually, so while in this case, this detail for this address would only exist once in the database, it would be assigned to all individuals with that address.
    - *Note 2*: You must bring in two different Address tables to get this portion to work.
6. Modify the query from #A5 to include the StateProvinceCode, CountryRegionCode, and Name.

Part B

1. Create a query that calculates a count of Departments by Group Name from the HumanResources.Department table.
2. Create a completely different query that shows the distinct listing of Product Subcategories (Production.ProductSubcategory), including ProductSubcategoryID and Name in the query.
3. Modify the #B2 query to count the number of products from Production.Product by Subcategory, call the column "ProductCount".
4. Modify the #B3 query to include the Average StandardCost and the

Average ListPrice, and make sure to name the columns respectively. Order the results by the SubCategory Name (A-Z).

5. Add a column for the total product inventory quantity for each ProductSubcategory. Make sure to name it as well.

Part C

For this part of the project, you will design a new database and write queries, views, and functions for the database. Your database must have a minimum of 4 tables. All tables must be appropriately related and be in 3rd normal form. You must populate the tables with enough data so that you can write useful queries. Two of your four tables must have many to many relationships with a linking table between them to resolve this issue.

1. Select a topic for your database. You can Google databases to see what information is readily available. You can also select data that you have available from work or home. Examples include a database for your home library, inventory management, online retail, or payroll.
2. Create an Entity Relationship diagram to show your tables and relationships. Make sure all tables are in 3rd normal form, and you have one linking table between the two tables with many-to-many relationships. Create the preliminary diagram using the SQL Server Management Studio. Save this as a PDF or other suitable file format.
3. Submit the code/script that creates your database and all four tables (don't forget to add code that relates the tables via Foreign and Primary keys). Also, write scripts that add contents/data to the tables. Do not spend too much time adding large quantities of data. You will need just enough data in each table to show functionality when you run your queries. Be sure to identify each section of code so that the reader will know what it does. All scripts must be saved and uploaded in .sql format so that they can be tested in SQL Server.
4. Create one View per table and save the View script for each.
5. Using a join, you should also create one view that shows column information from two related tables.
6. You should create two (2) queries using data from two or more tables; each query should contain *calculated fields*. At least one of the queries must use GROUP BY and HAVING along with calculated fields. All

calculated fields must have Aliases for ease of readability by the users. You should also create at least one function.