**EX16\_AC\_CH06\_GRADER\_CAP\_HW - Northwind Product Analysis**

**Project Description:**

*You have been asked to modify a database that tracks driver data and insurance details. You will update, add to, and delete records from tables. You will also create queries that aggregate data, find unmatched, and repeating values.*

**Instructions:**

For the purpose of grading the project you are required to perform the following tasks:

| **Step** | **Instructions** | **Points Possible** |
| --- | --- | --- |
| **1** | Start Access. Open the downloaded Access database named *exploring\_a06\_Grader\_h1*. Save the database as **exploring\_a06\_Grader\_h1\_LastFirst**. | 0 |
| **2** | *One of the most important aspects of a database is that the actual data is not static. It is dynamic in nature. Consequently, there are times when you may need to update, edit, or delete some data. You can create specialized queries within Access to easily deal with these changes.*  Create a select query that includes the CategoryID and CategoryName from the Categories table and the UnitPrice and ProductName fields from the Products table (in that order). Run the query and note the CategoryIDs for Beverages and Dairy. | 8 |
| **3** | *You have been notified that the Unit Price for beverages has increased by 5%. You need to update the prices for just the beverages in the database table. This can be done by creating an “update query”. This kind of query will look at your criteria and then update the table appropriately. In order to create an update query, you must already have an existing query created.*  Add the appropriate CategoryID criterion to limit the query output to only beverages. Convert the query to an update query. Update the UnitPrice for beverages only by increasing it by 5 percent. | 10 |
| **4** | Update the UnitPrice for dairy products only by increasing it by 4 percent. **Run the query**. It important to monitor how many times an update query is executed as Update queries are capable of changing values (such as prices) more than one time. **Run the query again**. (**Note**: Your final UnitPrice for dairy products should now show a total increase of 8%.) Save the query as **Update Prices**. Close the query. | 10 |
| **5** | Create a select query that identifies all of the discontinued products. Include all fields from the Products table. | 8 |
| **6** | *The next type of specialized query is a Table Query. A “table query” will create a new table based on the query results. There are a number of products that get discontinued. It would be nice to have a table with only these products, which is what you are doing when you make the query into a table query.*  Convert the select query to a make table query. Name the new table **Discontinued Products**. Run the query. Save the query as **Make Discontinued Products Table**. Close the query. | 6 |
| **7** | *An “append query” will add or append data to an existing table. In this case, the Boston Crab Meat product is one of the products that will be discontinued, so you will want to add this to the discounted products table that was created using a table query.*  Make a copy of the Make Discontinued Products Table query and save it as **Append Discontinued Products Table**. Open the Append Discontinued Products Table query in Design view. Convert the make table query to an append query. The query will append to the Discontinued Products table. | 8 |
| **8** | Modify the criteria to add the Boston Crab Meat product to the Discontinued Products table, using the SupplierID from the Products table. Remove the criteria related to the Discontinued field. Run the query, save it, and then close it. | 4 |
| **9** | *A “delete query” will use the criteria in your query and delete the records that match. Then if there are many records that match the criteria, they will all get deleted. This is a great tool, because you won’t need to delete them one at a time.*   Make a copy of the Append Discontinued Products Table query and save it as **Delete Discontinued Products**. Open the Delete Discontinued Products query in Design view. Convert the append query to a delete query. | 8 |
| **10** | Modify the criteria to delete the discontinued products, as well as the record for Boston Crab Meat using its SupplierID (Hint: There will be two criteria). Run, save, and close the query. | 8 |
| **11** | Open the Profit query in Design view and add the LastName field from the Employees table to the last column of the design grid. Run, save, and close the query. | 4 |
| **12** | *“Crosstab” or “Cross Tabulation”, is a process or function that combines and/or summarizes data from one or more sources into a concise spreadsheet-like format. One part of your analysis is to look at each employee and how much profit they have made for each product category. Using the Crosstab query is the best tool to use to get this information.*   Use the query wizard to create a crosstab query based on the Profit query that shows total profit by LastName (row heading) and CategoryName (column heading). Name the query **Profit\_Crosstab**. | 8 |
| **13** | Modify the query to display CategoryName as a row heading field and LastName as a column heading field. Run, save, and close the query. | 6 |
| **14** | *This type of query uses something called “Null” for the criteria for the query. Null is used in databases to represent a missing value. When you look in a database table and a field appears blank, it really is not. It has a value, but the value is Null. You can use this concept to determine records that have no orders. Keep in mind that Null is not the same as a zero value or a field that contains spaces.*  Create a query to find out if any of the products have no current order details. Add all of the fields from the Products table to the results. Save the query as **Products With No Orders**. Run the query and close it. | 12 |
| **15** | Save the database. Close the database, and then exit Access. Submit the database as directed. | 0 |

| **Total Score** | **100** |
| --- | --- |