Access 2016 capstone project AC-3

Working with a Sales Database

In this project, you will work with a sales database from Top't Corn, a popcorn company with a multiple food trucks and two retail stores. Previously, Top't Corn kept their data in multiple Excel workbooks. Recently, they decided to expand their product offerings at different price points, and they realized they needed a more robust database to track sales. You will help them create new database tables and clean up data imported from Excel.

You will begin by creating new tables to track sales and sale details. You will use the Form Wizard to create a form based on the new tables. Next, you will modify the existing Items table and create forms based on that table. You will create a new form from scratch in Layout view to display records from the Locations table. Next, you will clean up the imported data in the Sales_Archive table, import additional records from an Excel worksheet, and create a relationship between the data in the Sales_Archive and the Items tables. You will create a series of queries using text, numeric, and date criteria. Finally, you will create a report using the Report Wizard and another report from scratch in Layout view.

Skills needed to complete this project:

- Create and save a new table
- Add a new field to a table
- Create a lookup field using values from another table
- Apply an input mask to a field in a table
- Apply date formatting to a field by modifying the field Format property
- Create a lookup field using list values
- Create a new record in a table
- Adjust table column widths
- Set a default value for a field in a table
- Use the Form Wizard to create a new form
- Change the data type of a field
- Create a Single Record form based on a table
- Create a Split form based on a table
- Create a new blank form in Layout view
- Add fields to a blank form from Layout view
- Resize controls in a form
- Move controls in a form
- Add a logo to a form header
- Delete a field from a table
- Delete a record from a table
- Find and replace data in a table
- Rename a field in a table
- Import records from an Excel worksheet
- Create a one-to-many relationship between two tables
- Enforce referential integrity in a one-to-many relationship
- Create a simple select query to combine fields from multiple tables
- Add text criteria to a query
- Hide a field in a query
- Use OR in a query
- Add numeric criteria to a query
- Specify the sort order in a query
- Use AND in a query
- Add date criteria to a query
- Add a calculated field to a query
- Create a parameter query
- Use the Report Wizard to create a new report
- Group records in a report
- Add totals to a report
- Create a new blank report
- Add fields to a blank report from Layout view
- Resize controls in a report
- Arrange controls in a report
- Add the date to a report header
- Add page numbers to a report footer

IMPORTANT: Download the resource file needed for this project from the **Resources** link. Be sure to extract the file after downloading the resources zipped folder. Please visit **SIMnet Instant Help** for step-by-step instructions.

Step 1 Download start file

1. Open the start file **AC2016-Capstone-Level3**. **NOTE:** If necessary, enable active content by clicking the **Enable Content** button in the Message Bar.

2. The file will be renamed automatically to include your name. Change the **project file** name if directed to do so by your instructor, and **save** it.

3. Create a new table from scratch to track sales.

    a. The first field should be an **AutoNumber** field named: `SaleID`
    b. The second field should be a **Date & Time** field named: `SaleDate`
    c. The third field should be a lookup field. (Hint: Use the Lookup Wizard to create the new field.) It should display the **LocationDescription** field from the **Locations** table. Values in the lookup should be sorted by values in the **LocationDescription** field. Include the **LocationID** field in the lookup, but do not display it. (Hint: Hide the key column.) Enable data integrity by restricting deletions. Name this field: `SaleLocation`

    d. Save the table with the name: `Sales`

4. Switch to Design view and modify field properties.

    a. Add an input mask to the **SaleDate** field. Use the **Short Date** input mask. Do not change any other input mask options.

    b. Apply the **Long Date** format to the **SaleDate** field.

5. Add a new lookup field as the last field in the **Sales** table to track payment type. (Hint: Use the Lookup Wizard to create the new field.)

    a. Name the field: `PaymentType`
    b. The lookup field should display these values in this order:
        `Cash`
        `Credit Card`
        `Gift Card`
        `Store Credit`
    c. Limit data entry to values in the list. Do not allow multiple values.
    d. Save the table.

6. Switch back to Datasheet view to add sample records to the **Sales** table.

    a. Add three records to the table with the following data. (Hint: Remember, the first field in the table, **SaleID**, is an AutoNumber field, so there is nothing to enter for each record.)

SaleDate	SaleLocation	PaymentType
10/01/2016	Georgetown	Credit Card
10/01/2016	George Washington University	Cash
10/01/2016	George Washington University	Cash

    b. Adjust the width of the **SaleDate** field so the entire long date is visible.

    c. Save and close the table.

7. Create a new table to capture the details for each sale.

     a. The first field should be an AutoNumber field named: **SaleDetailID**

     b. The second field should be a lookup field named: **SaleID** The lookup field should be limited to values in the **SaleID** field of the **Sales** table. Include only the **SaleID** in the lookup field. Enable data integrity by restricting deletions.

     c. Save the table as: **SaleDetails**

     d. Add a third field to the far right of the table. Name this field: **Item** This is another lookup field. Include all the fields from the **Items** table. Sort the lookup items by values in the **ItemName** field. Hide the primary key field. Enable data integrity by restricting deletions.

     e. Add a **Number** field to the right of the **Item** field. Name the field: **Quantity**

     f. Set the default value for the **Quantity** field to: **1**

     g. Add three records to the table with the following data. (Hint: Remember, the first field in the table, **SaleDetailID**, is an AutoNumber field, so there is nothing to enter for each record.)

SaleID	Item	Quantity
1	Chocolate	4
1	Sea Salt and Caramel	2
2	Sea Salt and Caramel	4

     h. Close the table.

8. Use the Form Wizard to create a new form for inputting sales data.

     a. Include all the fields from the **Sales** table.

     b. Include the **Item** and **Quantity** fields from the **SaleDetails** table.

     c. View the form data by records in the **Sales** table with related records in the **SaleDetails** table displayed in a subform.

     d. The subform should be displayed as a Datasheet.

     e. Name the main form: **SalesForm** and name the subform: **SaleDetailsSubform** (Hint: Be sure to remove the space between SaleDetails and Subform in the subform name suggested by Access.)

     f. Open the form in Form view to review your work.

     g. Navigate to the record in the main form for SaleID **3** and enter sale details in the subform as follows:

       Item: **Original Blend**, Quantity: **3**

       Item: **Old Bay**, Quantity: **2**

     h. Close the form.

9. Open the **Items** table and modify the table fields as follows:

     a. Set the **Default Value** property for the **Price** field to: **6**

     b. Change the data type for the **Price** field to: **Currency**

     c. Autofit the width of the **ItemName** field.

　　　　d. Save the changes and close the **Items** table.

10. Create a Single Record form using the **Items** table as the record source. Save the form with the name: **SingleRecordForm**

11. Create a form to display records from the **Items** table in two formats with the Single Record form at the top and a Datasheet form at the bottom.

　　　　a. Create a **Split Form** based on the **Items** table.

　　　　b. Name the form: **SplitForm**

12. Begin a new blank form in Layout view.

　　　　a. From the **Locations** table, add the **LocationID, LocationDescription,** and **Comments** fields in that order, at the left side of the form.

　　　　b. Widen the labels so that they are just wide enough for **LocationDescription** to be completely visible.

　　　　c. From the **Locations** table, add the **OpenTime** field to the right of the **LocationID** controls.

　　　　d. Reduce the width of the **OpenTime** bound control so the control is just wide enough to display the time data.

　　　　e. From the **Locations** table, add the **CloseTime** field to the right of the **OpenTime** controls.

　　　　f. Reduce the width of the **CloseTime** bound control so the control is just wide enough to display the time data.

　　　　g. Move the **OpenTime** and **CloseTime** controls so they are next to the **LocationDescription** controls instead.

　　　　h. From the **Locations** table, add the **Days** field to the form layout in the empty space to the right of the **LocationID** controls, above the **OpenTime** controls.

Download Resources　　**i.** Add a logo to the form header. Use this file, located with the resources for this project: **toptCornLogo-small.png**

　　　　j. Save the form with the name: **LocationsForm**

13. Close the forms.

14. Open the **Sales_Archive** table.

　　　　a. Delete the **Total** field.

　　　　b. Find the record with the ID **500** and delete it. (Hint: It is the last record in the table.)

　　　　c. Find and replace each **ItemID** value **OLDB** with **OLDB005**.

　　　　d. Rename the **TotalSal** field to: **TotalSale**

　　　　e. Save and close the table.

Download Resources　　**15.** Import records from the Excel file **NewSalesData** (downloaded from the Resources link) and append a copy of the records to the **Sales_Archive** table.

16. Use the Relationships window to create a relationship between the *Items* and *Sales_Archive* tables.

　　　　a. Show the **Sales_Archive** table in the Relationships window.

    b. Create a one-to-many relationship between the **ItemID** field in the **Items** table and the **ItemID** field in the **Sales_Archive** table. You may rearrange the tables in the Relationships window if you want.

    c. Enforce referential integrity so a record cannot be deleted or altered in the *Items* table if it would cause a conflict with the data in the *Sales_Archive* table.

    d. Close the Relationships window and save the changes.

17. Create a query to display sales of Truffle flavored popcorn from the *Sales_Archive* table.

    a. Include the following fields in this order: the **SaleDate**, **Quantity**, and **TotalSale** fields from the **Sales_Archive** table and the **ItemName** field from the **Items** table.

    b. Add the criteria **Truffle** to the **ItemName** field.

    c. Hide the **ItemName** field in the query results. Run the query to check your work. (Hint: There should be 46 records in the query results.)

    d. Save the query as **TruffleQry** and then close the query.

18. Create a query to display sales of Old Bay *or* Truffle flavored popcorn from the *Sales_Archive* table

    a. Include the following fields in this order: the **SaleDate**, **Quantity**, and **TotalSale** fields from the **Sales_Archive** table and the **ItemName** field from the **Items** table.

    b. Add the criteria **Old Bay** or **Truffle** to the **ItemName** field. Run the query to check your work. (Hint: There should be 110 records in the query results.)

    c. Save the query as **NewFlavorsQry** and then close the query.

19. Create a query to display sales for more than $100.00 from the *Sales_Archive* table.

    a. Include the following fields in this order: the **SaleDate** from the **Sales_Archive** table, **ItemName** field from the **Items** table, and **TotalSale** from the **Sales_Archive** table.

    b. Add criteria to the **TotalSale** field to return only sales **greater than 100**.

    c. Specify the sort order in the query, so the results always display the records with the highest values in the **TotalSale** field first. Run the query to check your work. (Hint: There should be 17 records in the query results.)

    d. Save the query as **HighDollarSalesQry** and close it.

20. Create a query to display sales for more than $100 of Old Bay flavored popcorn from the *Sales_Archive* table

    a. Include the following fields in this order: the **ItemName** field from the **Items** table and the **SaleDate**, **Quantity**, and **TotalSale** fields from the **Sales_Archive** table.

    b. Add the criteria to the query to return only records where the **ItemName** is **Old Bay** and the **TotalSale** is **greater than 100**. Run the query to check your work. (Hint: There should be 3 records in the query results.)

    c. Save the query as **HighDollarOldBayQry** and then close the query.

21. Create a query to display sales of Original Blend popcorn on July 4, 2016 from the *Sales_Archive* table.

    a. Include the following fields in this order: the **ItemName** field from the **Items** table and the **SaleDate** and **Quantity** fields from the **Sales_Archive** table.

    b. Add the criteria to the query to return only records where the **ItemName** is **Original Blend** and the Date is **July 4, 2016**. Run the query to check your work. (Hint: There should be 3 records in the query results.)

    c. Save the query as `July4OriginalBlendQry` and then close the query.

22. Create a query to calculate the per unit price of the archived sales.

    a. Include the following fields in this order: the **ItemName** field from the **Items** table and the **Quantity** and **TotalSale** fields from the **Sales_Archive** table.

    b. Add a calculated field to the far right of the query to calculate the value of the **TotalSale divided by Quantity**. Name the field: `CostPerUnit` Run the query to check your work. (Hint: There should be 234 records in the query results.)

    c. Save the query as `CostPerUnitQry` and then close the query.

23. Create a parameter query to display sales from a specific date.

    a. Include these fields in this order: the **SaleDate** field from the **Sales_Archive** table, the **ItemName** field from the **Items** table, the **Quantity** and **TotalSale** fields from the **Sales_Archive** table.

    b. Use the prompt: `Enter the sale date:`

    c. Run the query to check your work. Enter the date `7/4/2016` when prompted. (Hint: There should be 9 records in the query results.)

    d. Save the query as `ByDateParameterQry` and close it.

24. Use the Report Wizard to create a report based on the **NewFlavorsQry** query.

    a. Include the fields from the **NewFlavorsQry** query in this order: **ItemName**, **SaleDate**, **Quantity**, and **TotalSale**.

    b. View the data by the **Items** table.

    c. Do not add any additional grouping.

    d. Sort the detail records by sale date.

    e. Use the Stepped layout in Portrait orientation.

    f. Name the report `NewFlavorRpt` and then view the report to check your work.

25. Switch to Layout view and add more grouping and totals to the **NewFlavorRpt** report.

    a. Display the *Group, Sort, and Total* pane and add new grouping by values in the **SaleDate** field by **Month**.

    b. Add totals to each group to calculate the **Sum** of values in the **TotalSale** field.

    c. Save and close the report.

26. Create a new report from scratch in Layout view.

    a. From the **Sales_Archive** table, add the **SaleDate** field to the report. Add these fields in order to the right of the **SaleDate** controls: **ItemName** from the **Items** table, and **Quantity** and **TotalSale** from the **Sales_Archive** table.

    b. Resize the **ItemName** controls so all the item names are visible. (Hint: Widen the **ItemName** column so the entire *Sea Salt and Caramel* name is visible.)

     c.   Add the **ItemID** field from the **Items** table. Move the **ItemID** controls so they appear to the left of the **ItemName** controls.

     d.   Add the current date to the report header. Use this date format: **Thursday, February 25, 2016** Do not include the time.

     e.   Add page numbers centered in the report footer. Use this page number format: **Page N of M**

     f.   Save the report as: `SalesArchiveRpt`

Step 2 Upload & Save

27. Save and close any open database objects and then close the database.

28. Upload and save your **project file**.

Step 3 Grade my Project

29. Submit project for grading.