**Database Case Study – Letty’s Costume Rental**

**Assumptions**

1. **Scenario**

“When a customer checks inhis rented costume, the Checkedln field of the Rental record is then checked. At the end ofeach day, reservations whose Checkedln field is yes should be moved to the RentalHistory table and deleted from the Rental table. Moving these reservations requires a query that identifies the reservations to move, as well as an update query to append the reservations to

the RentalHistory table”

**Assumption**

The CheckedIn fields will be moved from Rental table to Rental History Table every time the database is opened. We can choose to accept the append and delete option. The append and delete action queries are assigned to the Macro AutoExec.

1. **Scenario**

“What percentage of our customers return their costumes late?”

**Assumption**

Followed the criteria mentioned in assignment folder iink **“ For Query 6 - I will accept a list of customers who are late (uniquely identified)”**

1. **Scenario**

“What costume size is most popular? Provide a count by size.”

**Assumption :**

Provided the count by size and the maximum among count as the popular one.

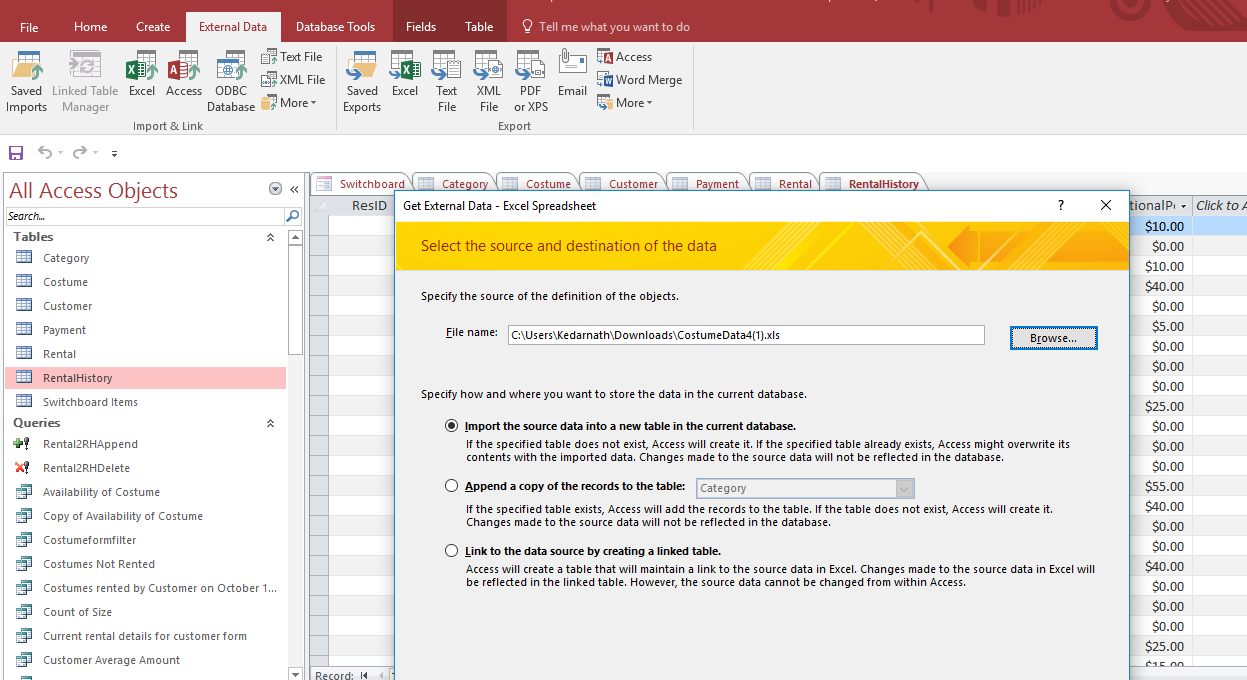
1. Executed the macro for moving the tables from Rental to RentalHistory and for the question 14 used the another version of the access.

**Features Added to the Database**

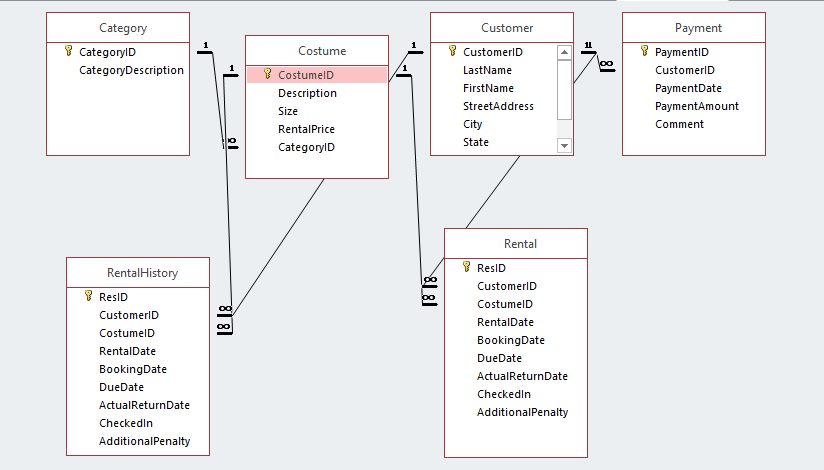
1. Developed the test the design forms to navigate to other form by clicking on the button and closing the current form (using macro builder)
2. Applied the combo boxes where ever possible to retrieve the data like filters
3. Placed print and closed button at the end of each report to enable the customer to take printout or close the report

**Screenshots**

**Import from Excel**

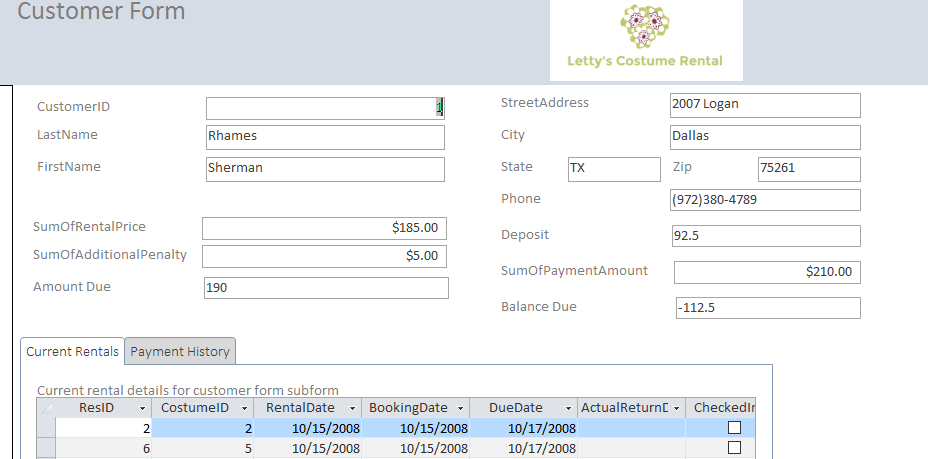


**Access Relationship**

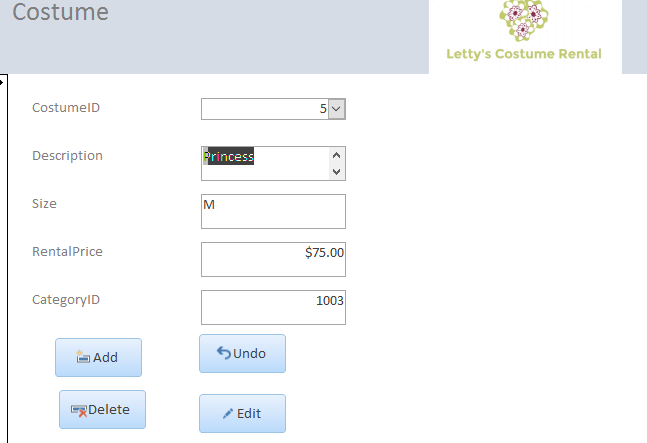


**Forms**

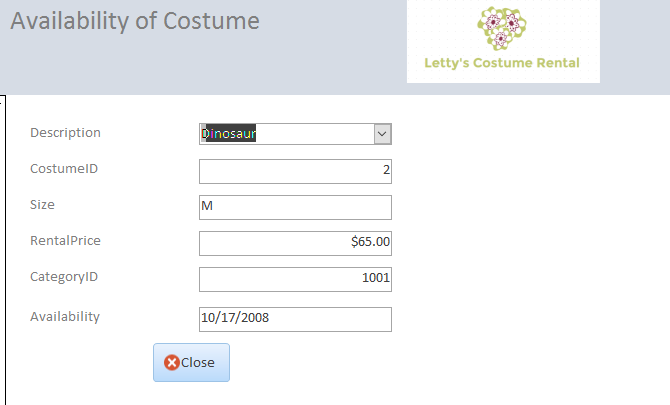
1. Customer Form



1. Costume Form



1. Costume Availability



**Auto Look up**

Query :

SELECT Rental.CostumeID, Costume.Description, Costume.Size, Costume.RentalPrice, Costume.CategoryID, IIf([Rental]![CheckedIn]=Yes,"Available",[Rental].[DueDate]) AS Availability

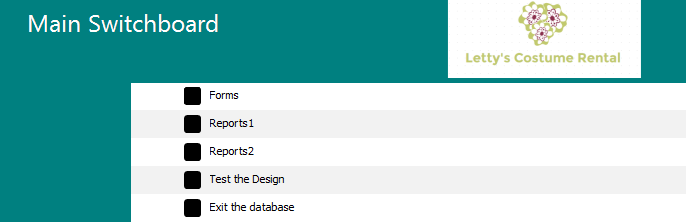
FROM Costume INNER JOIN Rental ON Costume.CostumeID = Rental.CostumeID

WHERE (((Costume.Description)=[Forms]![Availability of Costume]![AvailCombo]));

Screenshot:

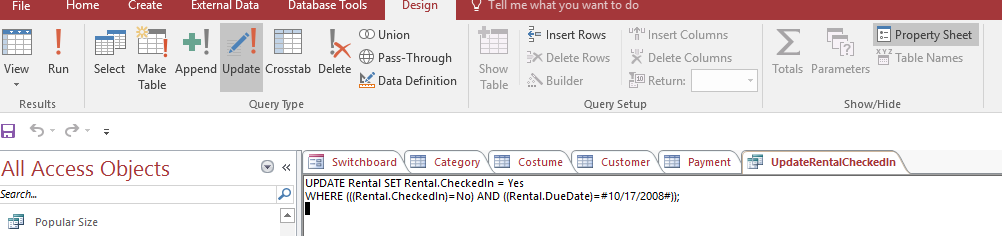


**Switchboard**



**Update Query**

Used Update query for testing the design for “Update the Rental table's records to show that all rentals due on October 17, 2008, have been returned.”

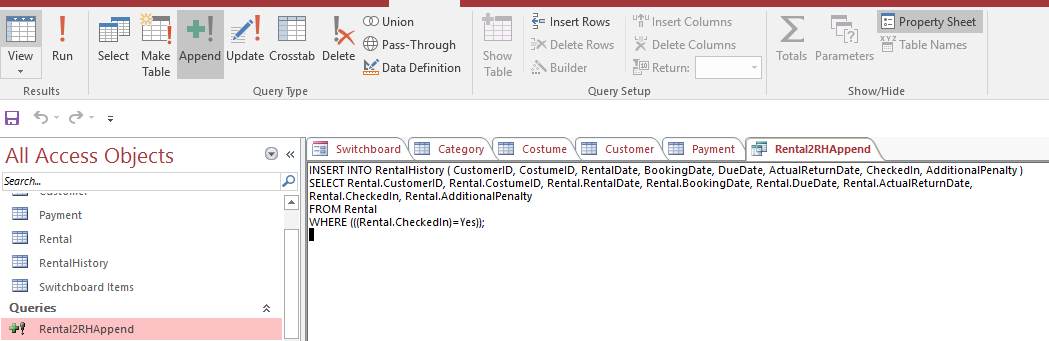


**Update Query :**

UPDATE Rental SET Rental.CheckedIn = Yes

WHERE (((Rental.CheckedIn)=No) AND ((Rental.DueDate)=#10/17/2008#));

**Used Append and Delete for Moving data from Rental to Rental History Table**



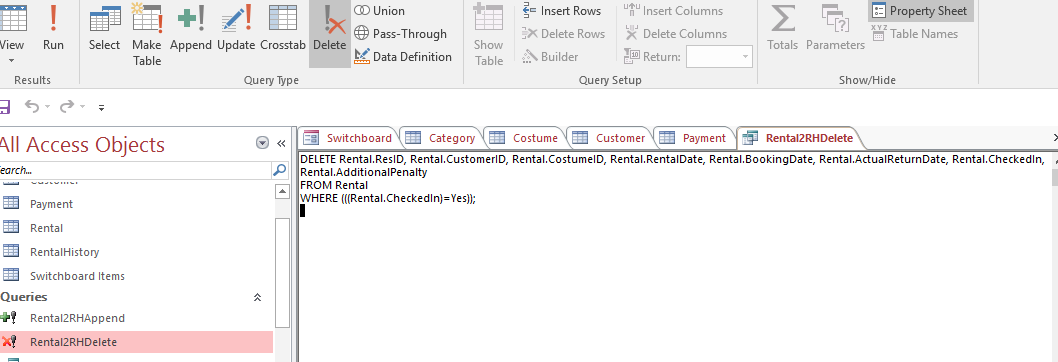
**Append query :**

INSERT INTO RentalHistory ( CustomerID, CostumeID, RentalDate, BookingDate, DueDate, ActualReturnDate, CheckedIn, AdditionalPenalty )

SELECT Rental.CustomerID, Rental.CostumeID, Rental.RentalDate, Rental.BookingDate, Rental.DueDate, Rental.ActualReturnDate, Rental.CheckedIn, Rental.AdditionalPenalty

FROM Rental

WHERE (((Rental.CheckedIn)=Yes));



**Delete Query**

DELETE Rental.ResID, Rental.CustomerID, Rental.CostumeID, Rental.RentalDate, Rental.BookingDate, Rental.ActualReturnDate, Rental.CheckedIn, Rental.AdditionalPenalty

FROM Rental

WHERE (((Rental.CheckedIn)=Yes));

**Select Query**

1. Which costumes were rented today? Which customers rented these costumes? (Use

October 15, 2008, as the date. At a minimum, your results should show the customer's

first and last name, costume description, reservation date, costume size, and rental price.)

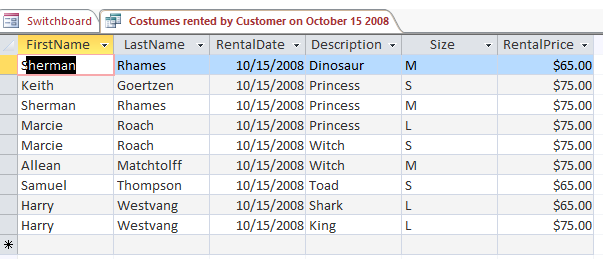
**Solution :**

SELECT Customer.FirstName, Customer.LastName, Rental.RentalDate, Costume.Description, Costume.Size, Costume.RentalPrice

FROM Customer INNER JOIN (Costume INNER JOIN Rental ON Costume.CostumeID = Rental.CostumeID) ON Customer.CustomerID = Rental.CustomerID

WHERE (((Rental.RentalDate)=#10/15/2008#));

**Screenshot:**



1. Based on rental history data, which costumes have never been rented? (Base your

answer on rental history table data.)

**Solution:**

SELECT Costume.CostumeID, Costume.Description, Costume.Size, Costume.RentalPrice

FROM Costume

WHERE (((Costume.CostumeID) Not In (SELECT RentalHistory.CostumeID

FROM RentalHistory;)));

**Screenshot**



1. How many dinosaur costumes are currently checked out? How many princess

costumes are currently checked out?

**Solution :**

SELECT Costume.Description,Count(Costume.Description) AS [Count of Costume]

FROM Costume INNER JOIN Rental ON Costume.CostumeID = Rental.CostumeID

GROUP BY Costume.Description, Rental.CheckedIn

HAVING (((Costume.Description)="Princess") AND ((Rental.CheckedIn)=No))

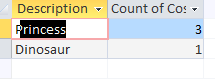
UNION ALL SELECT Costume.Description, Count(Costume.Description) AS [Count of Costume]

FROM Costume INNER JOIN Rental ON Costume.CostumeID = Rental.CostumeID

GROUP BY Costume.Description, Rental.CheckedIn

HAVING (((Costume.Description)="Dinosaur") AND ((Rental.CheckedIn)=No));

**Screenshot**



4.Based on rental history data, what is the average rental amount spent by each

customer? (Base your answer on rental history table data. Do not include penalties or

deposits.)

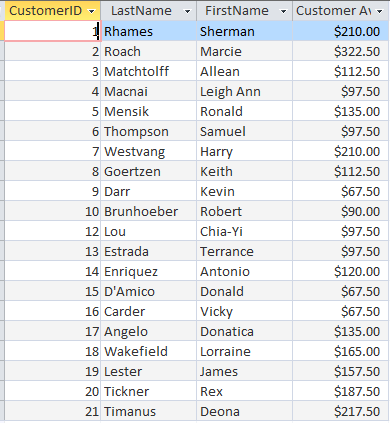
Solution:

SELECT RentalHistory.CustomerID, Customer.LastName, Customer.FirstName, Avg(Payment.PaymentAmount) AS [Customer Avg Payment]

FROM (Customer INNER JOIN Payment ON Customer.CustomerID = Payment.CustomerID) INNER JOIN RentalHistory ON Customer.CustomerID = RentalHistory.CustomerID

GROUP BY RentalHistory.CustomerID, Customer.LastName, Customer.FirstName;

Screenshot:



5. Based on rental history data, which customers have paid penalties? For each customer,

provide a total. (Base your answer on rental history table data.)

Solution :

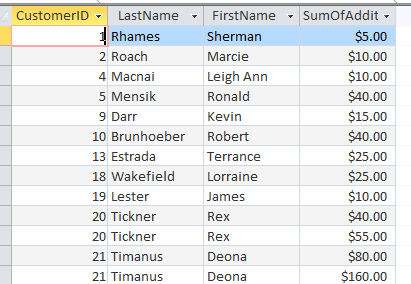
SELECT RentalHistory.CustomerID, Customer.LastName, Customer.FirstName, Sum(RentalHistory.AdditionalPenalty) AS SumOfAdditionalPenalty

FROM Customer INNER JOIN RentalHistory ON Customer.CustomerID = RentalHistory.CustomerID

GROUP BY RentalHistory.CustomerID, Customer.LastName, Customer.FirstName, RentalHistory.AdditionalPenalty

HAVING (((RentalHistory.AdditionalPenalty)>0));

Screenshot :



6. What percentage of our customers return their costumes late? / For Query 6 - I will accept a list of customers who are late (uniquely identified) – **Used Final Case Study access file for this answer**

Solution :

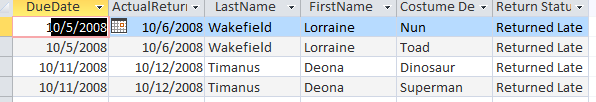
SELECT Rental.DueDate, Rental.ActualReturnDate, Customer.LastName, Customer.FirstName, Costume.Description AS [Costume Description], IIf([Rental].[ActualReturnDate] Is Not Null And [Rental].[ActualReturnDate]<=[Rental].[DueDate],"Returned On Time",IIf([Rental].[ActualReturnDate] Is Not Null And [Rental].[ActualReturnDate]>[Rental].[DueDate],"Returned Late","Not Returned")) AS [Return Status]

FROM Customer INNER JOIN (Costume INNER JOIN Rental ON Costume.CostumeID = Rental.CostumeID) ON Customer.CustomerID = Rental.CustomerID

WHERE (((IIf([Rental].[ActualReturnDate] Is Not Null And [Rental].[ActualReturnDate]<=[Rental].[DueDate],"Returned On Time",IIf([Rental].[ActualReturnDate] Is Not Null And [Rental].[ActualReturnDate]>[Rental].[DueDate],"Returned Late","Not Returned")))='Returned Late'))

ORDER BY Rental.DueDate, Customer.LastName, Costume.Description;

Screenshot :



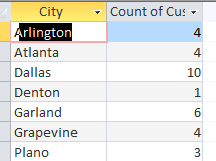
7.Where do the store's customers live? Provide a count by city.

Solution : SELECT Customer.City, Count(Customer.City) AS [Count of Customer Location]

FROM Customer

GROUP BY Customer.City;

Screenshot :



**8.** Ms. Scott wants to identify the store's repeat customers. Provide Ms. Scott with the

names and addresses of all repeat customers.

Solution :

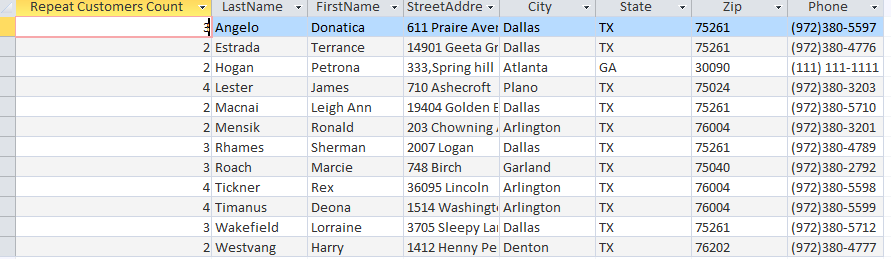
**SELECT Count(RentalHistory.CustomerID) AS [Repeat Customers Count], Customer.LastName, Customer.FirstName, Customer.StreetAddress, Customer.City, Customer.State, Customer.Zip, Customer.Phone**

**FROM Customer INNER JOIN RentalHistory ON Customer.CustomerID = RentalHistory.CustomerID**

**GROUP BY Customer.LastName, Customer.FirstName, Customer.StreetAddress, Customer.City, Customer.State, Customer.Zip, Customer.Phone**

**HAVING (((Count(RentalHistory.CustomerID))>1));**

**Screenshot:**



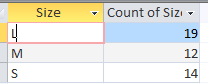
**9.** What costume size is most popular? Provide a count by size. ( Provided the max of Count of Size in the report)

Solution :SELECT Costume.Size, Count(Costume.Size) AS [Count of Size]

FROM Costume INNER JOIN RentalHistory ON Costume.CostumeID = RentalHistory.CostumeID

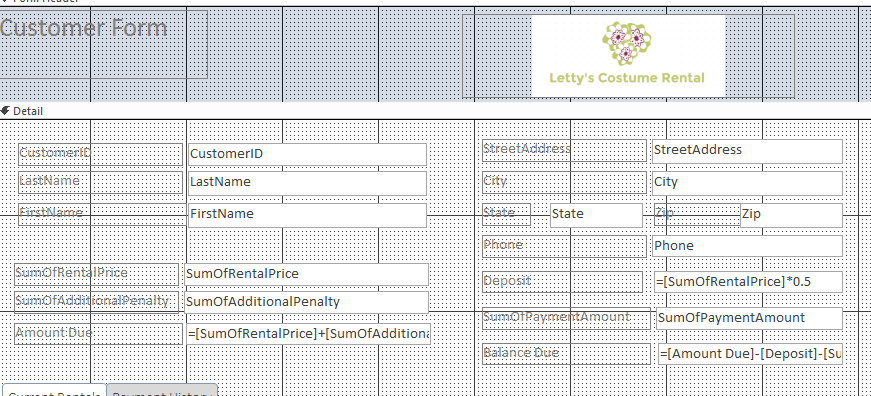
GROUP BY Costume.Size;

Screenshot:

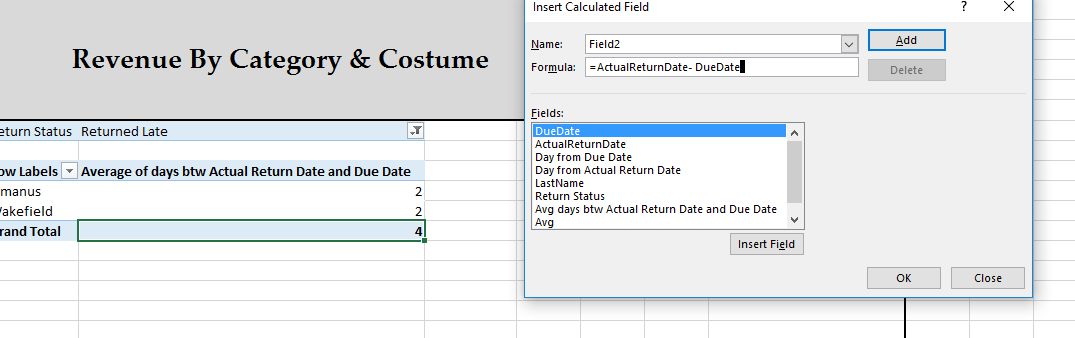




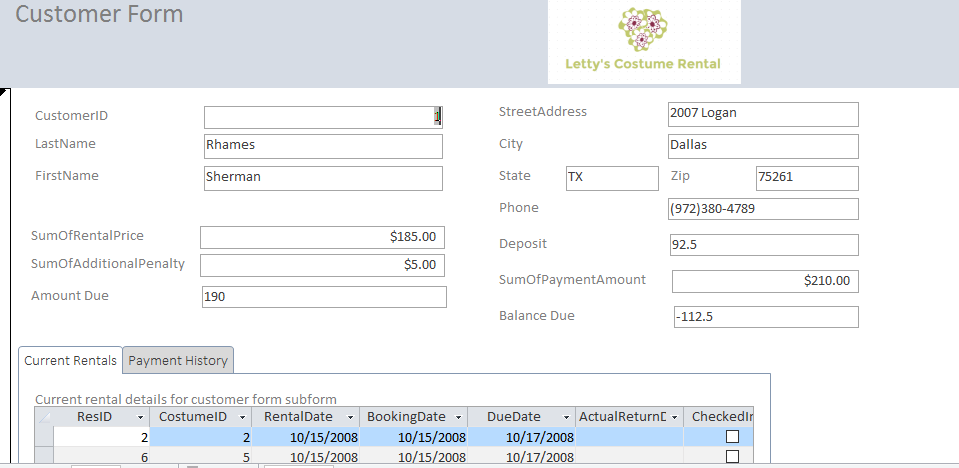
**Calculated Fields (Access)**



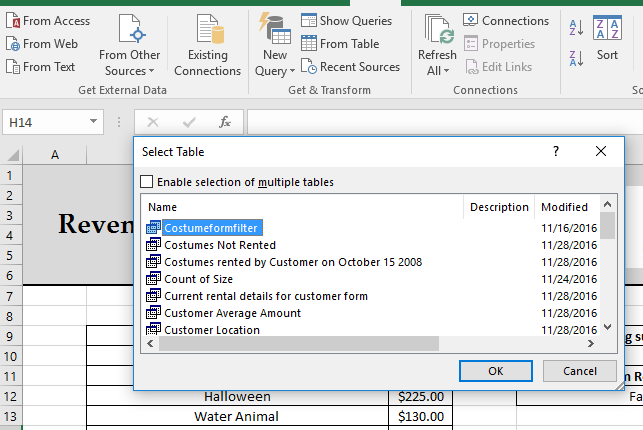
**Calculated Field (Excel)**



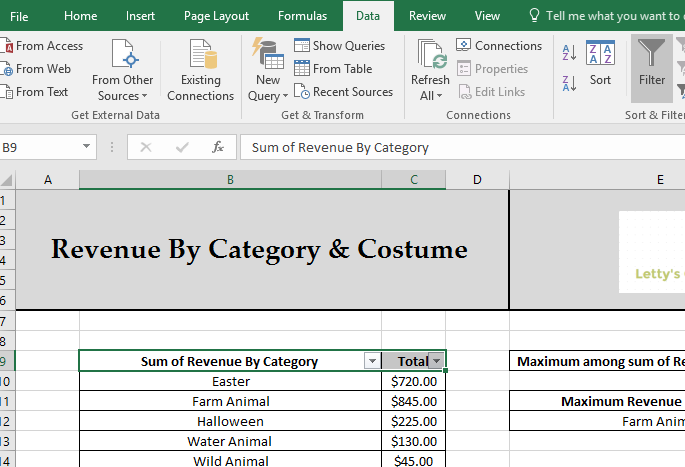
**Sub forms**



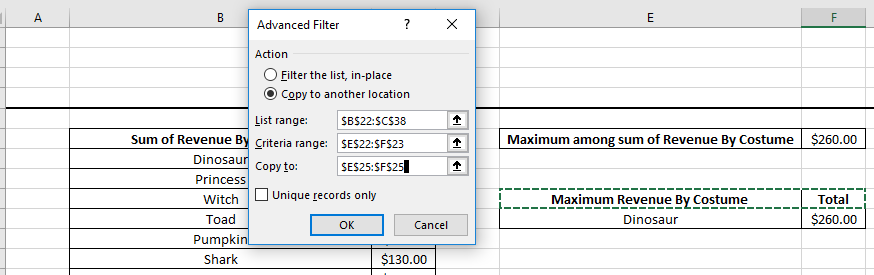
**Link to Access**



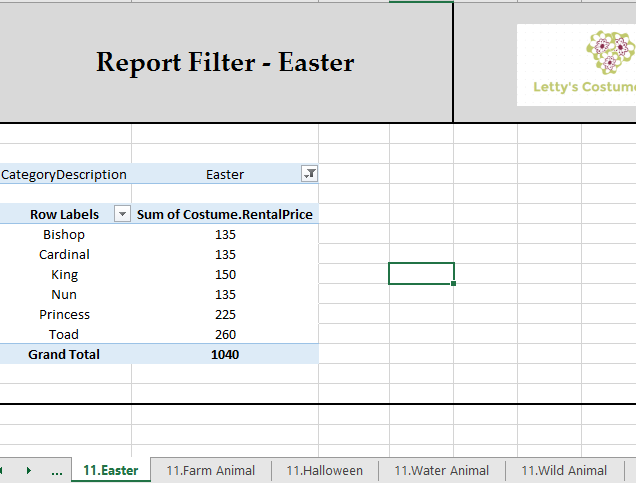
**Filter**



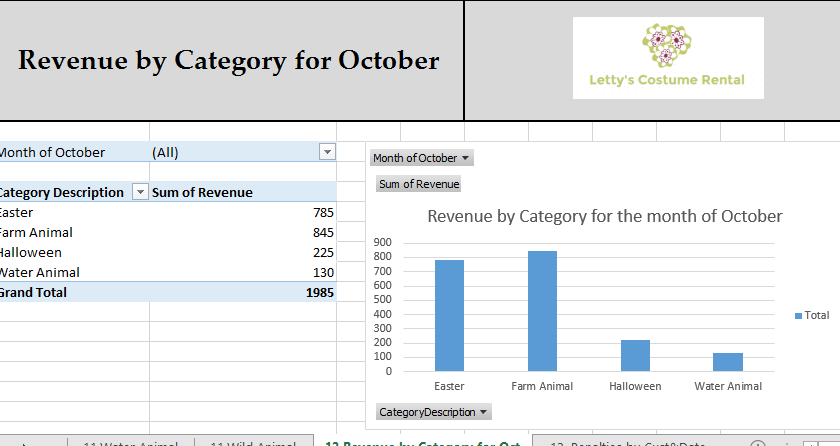
**Advance Filter**



**Report Filter**

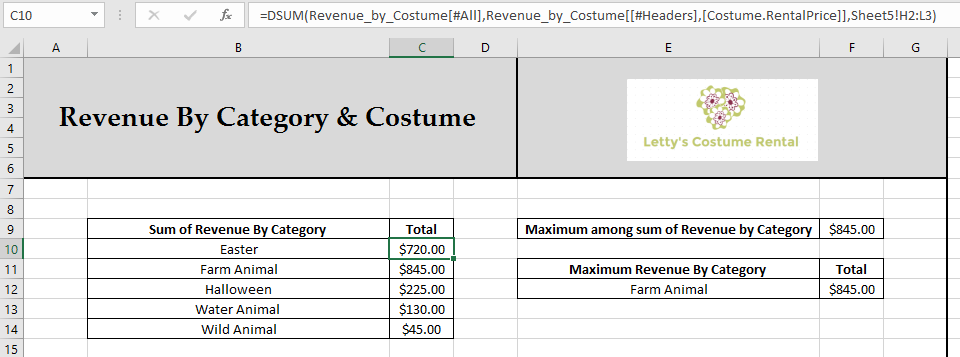


**Chart**

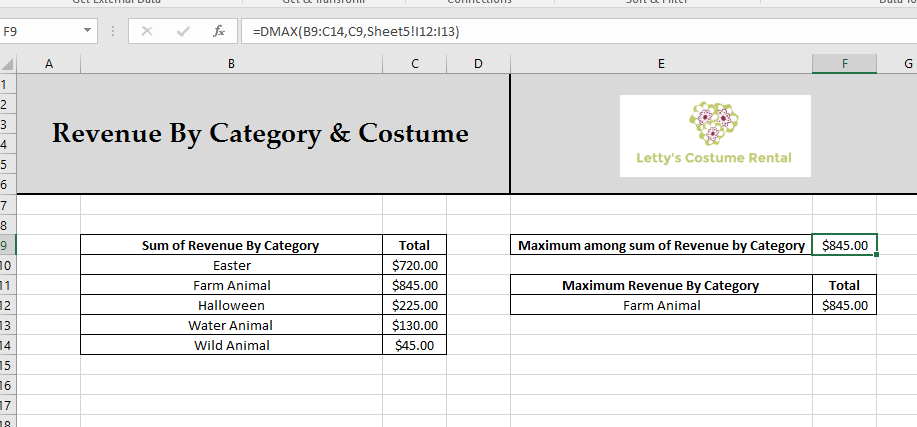


**Database function**

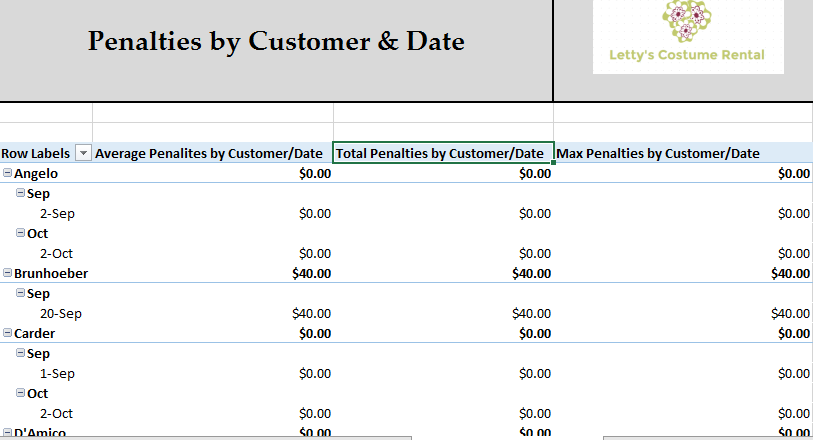
**DSUM**

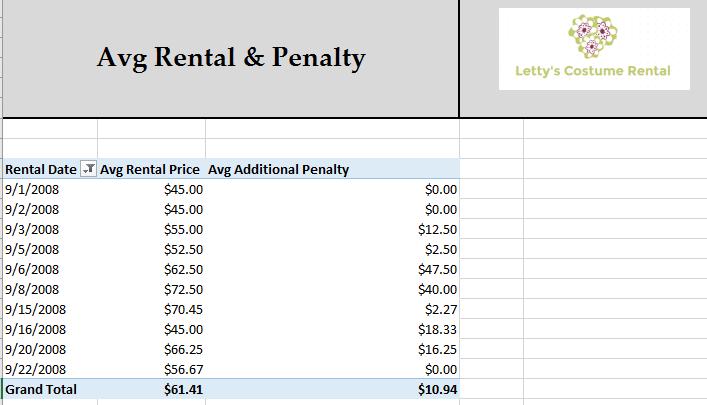


**DMAX**



**Pivot Tables**





KEY FEATURES

1. Provided an option to test the design of the database by adding it in the switchboard.
2. Aligned the switchboard to navigate through various form and reports of the database.
3. Developed the test the design forms to navigate to other form by clicking on the button and closing the current form (using macro builder)
4. Applied the combo boxes where ever possible to retrieve the data like filters
5. Placed print and closed button at the end of each report to enable the customer to take printout or close the report
6. Included all the details on the excel in a single workbook (Except for question 14)
7. Created a macro for appending records from Rental table to Rental History table and deleting the record from Rental table for those items that are checkedIn. Once Ms.Scott open the database prompt will be requested to append and delete actions.
8. The option of sending a mail once the costume is booked or returned could be added to the database to make it effective.
9. **PLEASE ADD POINTS IF YOU FIND ANYTHING**