Assignment 9 (100 points) Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

For the first two questions, you will be using the **RecipesExample** table from the book. Data Diagram on page 732. Each question worth 13 points unless specified at the question.

1. Using the tables in RecipesExample,

a. Run a query to show every field in the Recipe\_Classes table. Paste your query here.

select \*

from Recipe\_Classes

b. How many rows are in your result set? 7 This shows how many recipe classes.

c. Run a query to show the unique RecipeClassID from the Recipes table. Paste your query here.

select distinct Recipe\_Classes.RecipeClassID

from Recipes

left join Recipe\_Classes

on Recipe\_Classes.RecipeClassID = Recipes.RecipeClassID

d. How many rows are in your result set? 6 This show how many recipe classes are being used on recipes.

e. How many recipe\_classes have no recipes?

1

f. Write a query to show how many recipes are in the Recipes tables. Paste your query here.

select Recipes.RecipeTitle

from Recipes

g. How many rows in your result set?

15

2. Start with the SELECT statement of: select recipe\_classes.recipeclassid, recipeid

In your FROM statement list the Recipe table first (your left table) and the Recipe\_classes table second (your right table) in your query.

a. use a left join. Paste your query here.

select recipe\_classes.recipeclassid, recipeid

from Recipes

left join Recipe\_Classes

on Recipes.RecipeClassID = Recipe\_Classes.RecipeClassID

b. how many recipes appear?   
 15

c. change the left join to a right join. Paste your query here

select recipe\_classes.recipeclassid, recipeid

from Recipes

right join Recipe\_Classes

on Recipes.RecipeClassID = Recipe\_Classes.RecipeClassID

d. . how many recipes appear?

16

e. scroll down to the bottom of your result set to see the 16th record. What are the values you see? NULL

This is because with the right join, you are pointing to the Recipe\_classes table which has all 7 classes. So it has to list the class even though there is no recipe assigned to it.

f. Now only show the recipeclassid that has no recipes. Add a WHERE Clause that looks for the RecipeID to be NULL. Paste your Query here.

select recipe\_classes.recipeclassid

from Recipes

right join Recipe\_Classes

on Recipes.RecipeClassID = Recipe\_Classes.RecipeClassID

where Recipes.RecipeID = NULL

g. Copy your query from step c. Change your select statement from select recipe\_classes.recipeclassid to select recipes.recipeclassid. Now the field you return is coming from the recipes table and not recipe\_classes. Paste your query here.

select Recipes.recipeclassid, recipeid

from Recipes

right join Recipe\_Classes

on Recipe\_Classes.RecipeClassID = Recipes.RecipeClassID

h. Scroll down to the bottom of your result set to see the 16th record. What are the values in the last row?

NULL

This is because the recipe table doesn’t have that record recipeclassid. Remember the recipes table does not have the recipeclassid = 7

3. Using the **AdventureWorks2012 database** for the rest of the questions. This is similar to the query on page 326.

Using the Production.product table and the Sales.salesorderdetail table.

Write two queries to find the unqiue productid in each table. Then calculate the number of product that have never been ordered. Don’t need to Paste your queries, just fill the numbers in the box below.

|  |  |  |
| --- | --- | --- |
| Table | How many unique productid? The number of rows in your result set. | Represents |
| Production.product | 266 | Number of products in inventory |
| Sales.SalesOrderDetail | 504 | Number of products sold |
| How many products have never been ordered? | 238 | This is the number in your result set when you do an outer join pointing to the table with the larger number of rows. |

In your select statement, show the productid, product.name and salesorderid

Use the two tables in your FROM statement. Use a left join if you list the Production.product table first. Or use a right join if you list the Sales.SalesOrderDetail first.

Use the ON statement to show how you are joining the two tables

Use a WHERE statement, to only show the Salesorderid that are NULL.

Paste your **final** query here. It should have the 238 rows.

select distinct pro.ProductID, pro.name, sod.SalesOrderID

from Sales.SalesOrderDetail as sod

right join Production.Product as pro

on sod.ProductID = pro.ProductID

where sod.SalesOrderID is NULL

4. Display all customers and any orders that contains the word LOCK in their product name. This query is similar to the **Embedded SQL** at the bottom of page 326 and top of page 327.

You can use your query from the previous step and remove the WHERE Statement and change the fields in your select statement.

In the select statement, I want to show the product.name and CustomerID

select distinct cus.CustomerID, pro.ProductID, pro.Name

from sales.Customer cus

right join Production.Product as pro

on cus.ModifiedDate = pro.ModifiedDate

where name like '%lock%'

Using the inner join for Production.Product table and the Salesorderdetail and the Salesorderheader. You need to add a JOIN and ON statement, to JOIN The SalesOrderHeader table.

You should have a

SELECT

FROM

JOIN

ON

JOIN

ON

Then add a WHERE Statement for product name containing LOCK in their name. Hint using a LIKE statement.

Should get 260 records in your result set. Show the product.name and customerid.

1. Paste Query here.

select \*

from production.product pro

join Sales.SalesOrderDetail sod

on pro.ProductID = sod.ProductID

join Sales.SalesOrderHeader soh

on sod.SalesOrderID = soh.SalesOrderID

where pro.name like '%lock%'

order by pro.ProductID

b. Now use that as the embedded Sql statement. Call that sql results SP. So after you put the parathesis around your query type ‘as SP’. Paste your query here. Don’t run it yet.

(select \*

from production.product pro

join Sales.SalesOrderDetail sod

on pro.ProductID = sod.ProductID

join Sales.SalesOrderHeader soh

on sod.SalesOrderID = soh.SalesOrderID

where pro.name like '%lock%');

c. From the sales.customer table we want to display the customerid and sp.name (field from the query in step a). A customer may have purchased a product more than once, so eliminate the duplicates. To see all customers that have purchased the LOCK item and have not purchased the LOCK item, you will need to use an LEFT JOIN with the customer table being the first table.

Select Customer.CustomerID, sp.name……….

FROM Sales.Customer

Left JOIN (your query from step b)

ON sp.customerid = customer.cusotmerid

Paste your query here. You should get 19,820 records.

d. Order your results by sp.name descending. Paste query here.

e. Since we wanted all customers, run a New query to just show how many customers are in the sales.customer table. Paste Query Here.

f. How many rows are in your result set? Same number as your embedded query.

5. With an Outer Join quey, show what products have never been purchased. Using the production.product with the purchasing.purchaseorderdetail. You should have 239 rows in your result set.

a. Paste your query here.

6. Use the Sales Orders Example Database diagram on page 724 in our book, or in the module Reference Material.

a. Is there any linking table or tables? If so, what are the names of the linking table(s). Remember when you describe the relationship of the Linking table, you don’t have a statement for each direction of the relationship. You state the relationship of the two tables that have a many to many relationship resolved because of the linking table.

b. Put the relationship of the tables into words. Remember (for non linking tables) you want two phrase for each line connecting the tables. One statement going in each direction of the relationship. It’s important to understand the relationship, so we know when the number of rows in our result set is going to increase.

7. Using the same Sales Orders Example Database diagram.

a. If I run a query for s specific customer number to see their orders. Using the Customers and Orders tables I get 18 rows in my results set. Now I want to add the Order\_Details table because I want to see the ProductNumber on those orders. Do I expect to get more, less or the same number of rows in my result set?

1. Now I want to see the Product name, so I add the Products Table. Do I expect to get more, less or the same number of rows in my result set?
2. Now I want to see the CategoriesDescription for each Product. Do I expect to get more, less or the same number of rows in my result set?

8. I create a query using the Vendors Table and The Product\_Vendors table and get 220 rows. I want to add the ProductName to my query so I add the Product table. Do I expect to get more, less or the same number of rows in my result set? ( 5 pts)

9. I create a query using the Orders and Customer table for a specific date and get 27 rows in my result set. I add the Employees table so I can show the EmpFirstName. Do I expect to get more, less or the same number of rows in my result set? (4 pts)