**PROG8080**

**PROGRAMMING: DATABASE MANAGEMENT**

**Assignment: 4**

**Name:** Shravani Gandla

**Student Number:** 8697289

1. If you takealook at the Product table in the Production schema, so (Production.Product) you will see a large selection of products that the company sells. Now look at the Production.ProductReview table. This table lists all the products that customers have written reviews for. (Amazon will ask you to review each product once you purchase it). I would like you to come up with a SQL query that will list the Name, ProductReviewID and Color of each product that has a review written for it and is not of the color Multi or Yellow.

Hint: Now there are only 4 products that have ratings associated with them, so as a hint, you result set should be just 3 rows or records.

**Comment:**

The below query shows that the Name from the Production.Product table, Color from the Production.product table and ProductID from the Production.ProductReview table are selected By using “SELECT” From Production.Product table and Production.Productreview tables.These two tables are joined by using “INNER JOIN” which extracts the matched information from two tables using “ON” clause. There are 3 conditions like Comments is not null, then only it shows the comments whose color is not equal to Multi or Yellow.

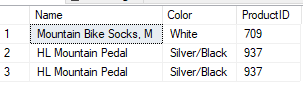
**Query:**

SELECT Production.Product.[Name],Production.Product.[Color], Production.ProductReview.[ProductID] FROM Production.Product

INNER JOIN Production.ProductReview ON Production.Product.ProductID = Production.ProductReview.ProductID

WHERE Production.ProductReview.Comments IS NOT NULL AND Color <> 'Multi' AND Color <>'Yellow';

**Output:**



2. Consider a join between the SalesTerritory table and the SalesPerson table on their TerritoryIDcolumns. A basic join should show the any territory that has been assigned to a sales person.However I would like you to produce SQL that would display 3 columns, the Territory name, the BusinessEntityID and the Year to Date sales numbers. Add in filters to show only rows where the YearToDate numbers are over $1 Million and are in the US region and the percentage of commission given to the BusinessEntityID is less than 10%.

Hint: you should end up with 8 rows.

**Comment:**

The below query shows that the Name,SalesYTD columns from Sales.Sales.Territory table and BusinessEntityID from Sales.SalesPerson table are selected by using “SELECT” and joined the two tables by using “INNER JOIN” where it gives the matched data from two tables. Here, the “ON” clause compares the TerritoryID from both tables and extracts the data based on the conditions like SalesYTD>1000000 and CountryRegionCode = ‘US’ and percentage of Commission < 10%.

Here I have converted the percentage into a number.

Given 10%

10\*1/100 = 0.1

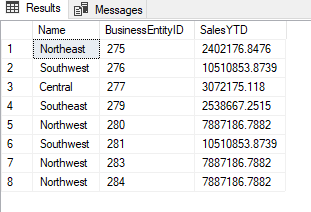
So, I used this 0.1 in the condition.

**Query:**

SELECT Sales.SalesTerritory.[Name],Sales.SalesPerson.[BusinessEntityID],Sales.SalesTerritory.[SalesYTD] FROM Sales.SalesTerritory INNER JOIN Sales.SalesPerson ON Sales.SalesPerson.TerritoryID = Sales.SalesTerritory.TerritoryID WHERE Sales.SalesTerritory.SalesYTD>1000000

AND Sales.SalesTerritory.CountryRegionCode = 'US' AND Sales.Salesperson.CommissionPct < 0.1;

**Output:**



3. Expand the above query to also include the sales person’s first and last names.

**Comment:**

The below query shows that the Name,SalesYTD columns from Sales.Sales.Territory table, BusinessEntityID from Sales.SalesPerson table and first name and last name from Person.Person table are selected by using “SELECT” and joined the three tables by using “INNER JOIN” where it gives the matched data from two tables. I have used two INNER JOINS as I have 3 tables and 2 ON clauses. Here, the first “ON” clause compares the TerritoryID from Sales.SalesTerritory , Sales.SalesPerson tables and second ON clause compares BusinessEntityID from Person.Person, Sales.SalesPerson extracts the data based on the conditions like SalesYTD>1000000 and CountryRegionCode = ‘US’ and percentage of Commission < 10%.

**Query:**

SELECT Person.Person.[FirstName],Person.Person.[LastName], Sales.SalesTerritory.[Name],Sales.SalesPerson.[BusinessEntityID],

Sales.SalesTerritory.[SalesYTD] FROM(( Sales.SalesTerritory INNER JOIN Sales.SalesPerson

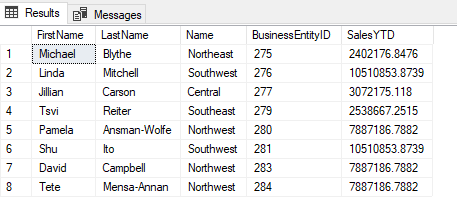
ON Sales.SalesPerson.TerritoryID = Sales.SalesTerritory.TerritoryID) INNER JOIN Person.Person

ON Person.Person.BusinessEntityID = Sales.SalesPerson.BusinessEntityID)

WHERE Sales.SalesTerritory.SalesYTD>1000000 AND Sales.SalesTerritory.CountryRegionCode = 'US'

AND Sales.Salesperson.CommissionPct < 0.1;

**Output**:



4. Build a query that will return the FirstName,LastName,BusinessEntityID,Title and NationalIDNumber of all the employees in the company.Your query should return 290 rows.

**Comment:**

The below query shows that the firstname column from Person.person table, lastname column from Person.Person table, BusinessEntityID column from HumanResources.Employee table,Title column from Person.Person table and NationalIDNumber column from HumanResources.Employee table are selected By using “SELECT”.The two tables are joined by using the “INNER JOIN” which extracts the matched data from the two tables.The “ON” clause is used to give a condition where it matches both BusinessEntityID’s from two tables.

**Query:**

SELECT Person.Person.[FirstName],Person.Person.[LastName],HumanResources.Employee.[BusinessEntityID],

Person.Person.[Title],HumanResources.Employee.[NationalIDNumber]

FROM HumanResources.Employee INNER JOIN

Person.Person ON HumanResources.Employee.[BusinessEntityID] = Person.Person.BusinessEntityID ;

**Output:**

