LAB 2:

1. Write a query that displays all the rows from the Person.Person table where the rows were modified after December 29, 2000. Display the business entity ID number, the name columns, and the modified date.
2. Rewrite the query from question 1 so that it displays the rows that were not modified during December 2000.
3. Write a query that displays the product ID and name for each product from the Production.Product table with the name starting with Chain.
4. Write a query that displays the business entity ID number, first name, middle name, and last name from the Person.Person table for only those rows that have E or B stored in the middle name column.
5. Write a query displaying the order ID, order date, and total due from the Sales.SalesOrderHeader table. Retrieve only those rows where the order was placed during the month of September 2001 and the total due exceeded $1,000.
6. Write a query displaying the sales orders where the total due exceeds $1,000. Retrieve only those rows where the salesperson ID is 279 or the territory ID is 6.
7. Write a query displaying the ProductID, Name, and Color columns from rows in the Production.Product table. Display only those rows in which the color is not blue.
8. Write a query that returns the business entity ID and name columns from the Person.Person table. Sort the results by LastName, FirstName, and MiddleName.
9. Write a query that displays in the “AddressLine1 (City PostalCode)” format from the Person.Address table.
10. Write a query using the Production.Product table displaying the product ID, color, and name columns. If the color column contains a NULL value, replace the color with No Color.
11. Modify the query written in question 2 so that the description of the product is displayed in the “Name: Color” format. Make sure that all rows display a value even if the Color value is missing.
12. Write a query using the Sales.SpecialOffer table. Display the difference between the MinQty and MaxQty columns along with the SpecialOfferID and Description columns.
13. Write a query using the Sales.SpecialOffer table that multiplies the MaxQty column by the DiscountPCT column. If the MaxQty value is null, replace it with the value 10. Include the SpecialOfferID and Description columns in the results.
14. Write a query that displays the first 10 characters of the AddressLine1 column in the Person.Address table.
15. Write a query that calculates the number of days between the date an order was placed and the date that it was shipped using the Sales.SalesOrderHeader table. Include the SalesOrderID, OrderDate, and ShipDate columns.
16. Write a query that displays only the date, not the time, for the order date and ship date in the Sales.SalesOrderHeader table. (Use any of the styles that return only date)
17. Write a query that adds six months to each order date in the Sales.SalesOrderHeader table. Include the SalesOrderID and OrderDate columns.
18. Write a query that displays the year of each order date and the numeric month of each order date in separate columns in the results. Include the SalesOrderID and OrderDate columns.
19. Write a statement that generates a random number between 1 and 10 each time it is run.
20. Write a query using the Sales.SalesOrderHeader table to display the orders placed during 2001 by using a function. Include the SalesOrderID and OrderDate columns in the results.
21. Write a query using the Sales.SalesOrderHeader table listing the sales in order of the month the order was placed and then the year the order was placed. Include the SalesOrderID and OrderDate columns in the results.
22. The HumanResources.Employee table does not contain the employee names. Join that table to the Person.Person table on the BusinessEntityID column. Display the job title, birth date, first name, and last name.
23. The customer names also appear in the Person.Person table. Join the Sales.Customer table to the Person.Person table. The BusinessEntityID column in the Person.Person table matches the PersonID column in the Sales.Customer table. Display the CustomerID, StoreID, and TerritoryID columns along with the name columns.
24. Write a query that joins the Sales.SalesOrderHeader table to the Sales. SalesPerson table. Join the BusinessEntityID column from the Sales.SalesPerson table to the SalesPersonID column in the Sales.SalesOrderHeader table. Display the SalesOrderID along with the SalesQuota and Bonus.
25. The catalog description for each product is stored in the Production.ProductModel table. Display the columns that describe the product from the Production.Product table, such as the color and size along with the catalog description for each product.
26. Write a query that displays the names of the customers along with the product names that they have purchased. Hint: Five tables will be required to write this query!
27. Write a query that displays all the products along with the SalesOrderID even if an order has never been placed for that product. Join to the Sales.SalesOrderDetail table using the ProductID column.
28. The Sales.SalesOrderHeader table contains foreign keys to the Sales.CurrencyRate and Purchasing.ShipMethod tables. Write a query joining all three tables, making sure it contains all rows from Sales.SalesOrderHeader. Include the CurrencyRateID, AverageRate, SalesOrderID, and ShipBase columns.
29. Write a query to determine the number of customers in the Sales.Customer table.
30. Write a query using the Production.Product table that displays the minimum, maximum, and average ListPrice.
31. Write a query that shows the total number of items ordered for each product. Use the Sales.SalesOrderDetail table to write the query.
32. Write a query using the Sales.SalesOrderDetail table that displays a count of the detail lines for each SalesOrderID.
33. Write a query using the Production.Product table that lists a count of the products in each product line.
34. Write a query that displays the count of orders placed by year for each customer using the Sales.SalesOrderHeader table.
35. Write a query that creates a sum of the LineTotal in the Sales.SalesOrderDetail table grouped by the SalesOrderID. Include only those rows where the sum exceeds 1,000.
36. Write a query that groups the products by ProductModelID along with a count. Display the rows that have a count that equals 1.
37. Write a query using the Sales.SalesOrderHeader, Sales.SalesOrderDetail, and Production.Product tables to display the total sum of products by ProductID and OrderDate.
38. Display the 3rd joined employee.
39. Display the customer who has placed 2nd highest orders