Chapter 5 Assignment Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**After each result set, paste your query.** (30 points)

Human Resources Database

1. To see all the data in the table run a query to show all the fields in the EmployeePayHistory. Notice the numeric format of the RateChangeDate. Change the query to only show the RateChangeDate as 11 characters using the CAST function shown on page 124. Order by the ratechangedate with the most recent date first.

Notice it now appears with Jan, Feb instead of numbers. Your result set should look like. (316 rows)



Paste your query here.

select cast(RateChangeDate as char(11)), BusinessEntityID, RateChangeDate, Rate, PayFrequency, ModifiedDate

from HumanResources.EmployeePayHistory

order by RateChangeDate desc

2. Adding to the query you wrote for question number 1. Add the Employee table to your query and add the fields JobTitle and Birthdate to your result set. (316 rows

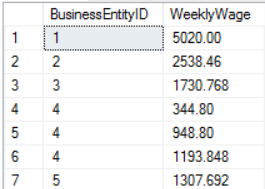
Paste your query here.

select cast(RateChangeDate as char(11)), eph.BusinessEntityID, RateChangeDate, Rate, PayFrequency, eph.ModifiedDate, JobTitle, BirthDate

from HumanResources.EmployeePayHistory eph, HumanResources.Employee emp

order by RateChangeDate desc

3. From the EmployeePayHistory table, list the BusinessEntityID and calculate the Rate times 40 (representing a 40 hour work week) and call it WeeklyWage. (316 rows

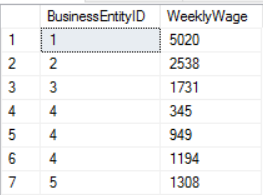


Paste your query here.

select BusinessEntityID, Rate, Rate \* 40 as WeeklyWage

from HumanResources.EmployeePayHistory

4. Revising the query from the previous step, the WeeklyWage would look better if there was rounded to the dollar amount. Use the As INTEGER character function to make this correction. Shown on page 138.



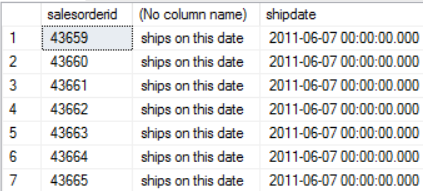
Paste your query here.

select BusinessEntityID, Rate, cast(Rate \* 40 as Integer) as WeeklyWage

from HumanResources.EmployeePayHistory

Sales Database

5. Using the SalesOrderHeader table, display the SalesOrderID and then the character literal **Ships on this date** and then the ShipDate. Shown on page 118. (31465 rows)



Paste your query here.

select SalesOrderID, ' ships on this date ', ShipDate

from Sales.SalesOrderHeader

6. With your query from the previous question, add the Sales.Customer table. Add the character literal **in this territory** and then the territoryid.

Notice, when you add TerritoryId to your select statement, it comes back an ambiguous field because it is in both tables as a Foreign key. We didn’t use it to link the two tables because you never link a FK to a FK.

Paste your query here.

select SalesOrderID, ' ships on this date ', ShipDate, 'in this territory', cus.TerritoryID

from Sales.SalesOrderHeader soh, Sales.Customer cus

7. With your query from the previous question, add the Sales.SalesTerritory table and link it to the Customer table only. Add the Name field to your result set. (31465 rows)

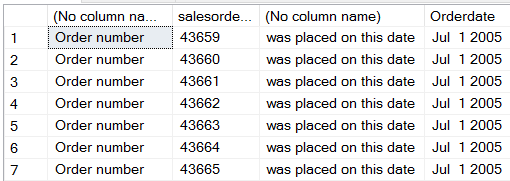
Paste your query here.

select SalesOrderID, ' ships on this date ', ShipDate, 'in this territory', cus.TerritoryID, Name

from Sales.SalesOrderHeader soh, Sales.Customer cus, Sales.SalesTerritory ter

where cus.TerritoryID = ter.TerritoryID

8. Using the SalesOrderHeader table, use the character literal **Order Number** followed by the column SalesOrderID followed by the character literal **was placed on this date** followed by the OrderDate. With the OrderDate as 11 characters. Shown on page 124. Make sure your Orderdate has a column title by using an Alias. (31465 rows)



Paste your query here.

select 'Order Number', SalesOrderID, 'was placed on this date', cast(OrderDate as char(11)) as OrderDate

from Sales.SalesOrderHeader

9. With your query from the previous question, add the SalesOrderDetail table and the productid. Notice you get more rows in your results set because Each SalesOrderHeader has many Order details lines. People order more than one item. Notice when you add this table, the SalesOrderId in your select statement is now ambigous because it’s in both tables. (121,317 rows)

Paste your query here.

select 'Order Number', soh.SalesOrderID, 'was placed on this date', cast(OrderDate as char(11)) as OrderDate, sod.ProductID

from Sales.SalesOrderHeader soh, Sales.SalesOrderDetail sod

10. With your query from the previous question, add the Production.Product table and the product name. Notice when you add this table, the ProductId in your select statement is now ambigous because it’s in both tables. In your result set only include product names that begin with Sport. The Where statement goes at the bottom of your query. (9180 rows)

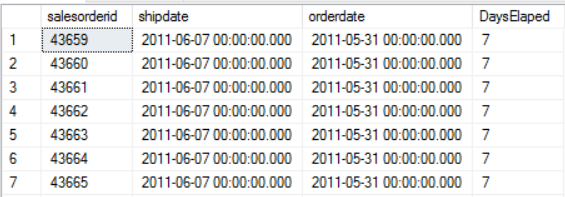
Paste your query here.

select 'Order Number', soh.SalesOrderID, 'was placed on this date', cast(OrderDate as char(11)) as OrderDate, sod.ProductID, pro.Name

from Sales.SalesOrderHeader soh, Sales.SalesOrderDetail sod, Production.Product pro

where Name like 'Sport%'

11. Using the SalesOrderHeader table, list the SalesOrderID, Shipdate, OrderDate and how many days between when it’s ordered and when it ships. . And call the last field DaysElaped. To subtract the dates you will need to CAST them as INTEGER. Shown on page 146.

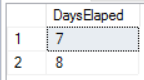


Paste your query here.

select SalesOrderID, ShipDate, OrderDate, cast(ShipDate - OrderDate as int) as DaysElapsed

from Sales.SalesOrderHeader

12. From looking at the result set of the previous query, I see that there are a lot of the orders that take 7 days. Write a query that would return only the unique number of days it takes to ship an order. (Hint result set should have one field and only two rows. You only want to display the unique DaysElasped).



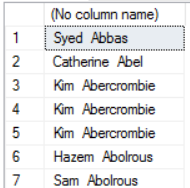
Paste your query here.

select distinct cast(ShipDate - OrderDate as int) as DaysElapsed

from Sales.SalesOrderHeader

Person Database

13. From the person table in the person database, concatenate the First Name and Last name with a space in between. Shown on page 135. (19972 rows)

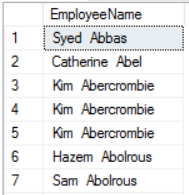


Paste your query here.

select FirstName + ' ' + LastName

from Person.Person

14. Revising your query from the lasts step, give the concatenated expression the title of **EmployeeName**. Similar query on page 137. (19972 rows)

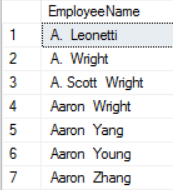


Paste your query here.

select FirstName + ' ' + LastName as EmployeeName

from Person.Person

15. Revising your query from the last step, order the result set by FirstName. (19972 rows)



Paste your query here.

select FirstName + ' ' + LastName as EmployeeName

from Person.Person

Order By FirstName