leonardo suarez Group 4 Assigment

Learning how to create diagrams in a database as a navigation tool. Creating diagram views as subject areas that isolates various sub-systems for querying information. Using the subject areas to solve problems for the business and document the necessary information needed to provide the appropriate query resolution. Writing 30 queries by each group member across the five databases identified.

Developing the soft skills needed in the business work environment such as teamwork, documentation and creating workflows.

Learning the structure of a newly acquired database systems by your company without any documentation.

Each Problem should follow the format in Problem 01.

Contents

[Problem 01: 10](#_Toc529118462)

[-- PROVIDE A LIST OF JOB CANDIDATES WHO GOT HIRED ALONG WITH THEIR RESPECTIVE JOB TITLE AND HIRE DATE 10](#_Toc529118463)

[Explanation: 10](#_Toc529118464)

[-- OUTPUT SHOULD SHOW ANYONE WHO GOT HIRED AND THE TITLE THEY RECEIVED ALONG WITH THE HIRE DATE 10](#_Toc529118465)

[Database 10](#_Toc529118466)

[Diagram(s) of tables 10](#_Toc529118467)

[Columns from Standard view 11](#_Toc529118468)

[Query 11](#_Toc529118469)

[Sample Output with total number of rows returned (2) 12](#_Toc529118470)

[Problem 02: 13](#_Toc529118471)

[-- PROVIDE A LIST OF CITIES WITHT THE POSSIBLE DELIVERY METHODS 13](#_Toc529118472)

[Explanation: 13](#_Toc529118473)

[Database 13](#_Toc529118474)

[Diagram(s) of tables 13](#_Toc529118475)

[Columns from Standard view 14](#_Toc529118476)

[Query 14](#_Toc529118477)

[Sample Output with total number of rows returned (28) 15](#_Toc529118478)

[Problem 03: 16](#_Toc529118479)

[-- PROVIDE LIST OF LAST NAMES THAT INCLUDE AT LEAST ONE 1 "a" 16](#_Toc529118480)

[Explanation: 16](#_Toc529118481)

[Database 16](#_Toc529118482)

[Diagram(s) of tables 16](#_Toc529118483)

[Columns from Standard view 17](#_Toc529118484)

[17](#_Toc529118485)

[Query 17](#_Toc529118486)

[Sample Output with total number of rows returned (2) 18](#_Toc529118487)

[Problem 04: 19](#_Toc529118488)

[-- PROVIDE A LIST OF PRODUCTS, THEIR PRICE AND THE CATEGORY THEY BELONG TO 19](#_Toc529118489)

[Explanation: 19](#_Toc529118490)

[Database 19](#_Toc529118491)

[Diagram(s) of tables 19](#_Toc529118492)

[Columns from Standard view 20](#_Toc529118493)

[Query 21](#_Toc529118494)

[Sample Output with total number of rows returned (77) 21](#_Toc529118495)

[Problem 05: 22](#_Toc529118496)

[-- PROVIDE A LIST OF AVAILABLE CURRENCY FORMS 22](#_Toc529118497)

[Explanation: 22](#_Toc529118498)

[Database 22](#_Toc529118499)

[Diagram(s) of tables 22](#_Toc529118500)

[Columns from Standard view 23](#_Toc529118501)

[Query 23](#_Toc529118502)

[Sample Output with total number of rows returned (105) 23](#_Toc529118503)

[Problem 06: 24](#_Toc529118504)

[-- provide me a list of all orders with a reason type that is not "other" 24](#_Toc529118505)

[Explanation: 24](#_Toc529118506)

[Database 24](#_Toc529118507)

[Diagram(s) of tables 24](#_Toc529118508)

[Columns from Standard view 25](#_Toc529118509)

[Query 25](#_Toc529118510)

[Sample Output with total number of rows returned (4237) 26](#_Toc529118511)

[Problem 07: 27](#_Toc529118512)

[-- make sure that every employee in the engineering department has a login ID 27](#_Toc529118513)

[Explanation: 27](#_Toc529118514)

[Database 27](#_Toc529118515)

[Diagram(s) of tables 27](#_Toc529118516)

[Columns from Standard view 28](#_Toc529118517)

[Query 28](#_Toc529118518)

[Sample Output with total number of rows returned (7) 29](#_Toc529118519)

[Problem 08: 30](#_Toc529118520)

[-- PROVIDE A LIST OF ALL TRANSACTIONS WITH UNKNOWN PAYMENT METHOD THAT OCCURED AT THE END OF THE MONTH IN 2016 30](#_Toc529118521)

[Explanation: 30](#_Toc529118522)

[Database 30](#_Toc529118523)

[Diagram(s) of tables 30](#_Toc529118524)

[Columns from Standard view 31](#_Toc529118525)

[Query 31](#_Toc529118526)

[Sample Output with total number of rows returned (319) 32](#_Toc529118527)

[Problem 09: 33](#_Toc529118528)

[-- PROVIDE A LIST OF STOCK ITEMS SOLD 33](#_Toc529118529)

[Explanation: 33](#_Toc529118530)

[Database 33](#_Toc529118531)

[Diagram(s) of tables 33](#_Toc529118532)

[Columns from Standard view 34](#_Toc529118533)

[Query 34](#_Toc529118534)

[Sample Output with total number of rows returned (227) 35](#_Toc529118535)

[Problem 10: 36](#_Toc529118536)

[-- PROVIDE A LIST THAT SHOWS IF ANY EMPLOYEE RATE CHANGED AT THE BEGINNING OF 2015 36](#_Toc529118537)

[Explanation: 36](#_Toc529118538)

[Database 36](#_Toc529118539)

[Diagram(s) of tables 36](#_Toc529118540)

[Columns from Standard view 37](#_Toc529118541)

[Query 37](#_Toc529118542)

[Sample Output with total number of rows returned (67) 38](#_Toc529118543)

[Problem 11: 39](#_Toc529118544)

[-- COUNT THE NUMBER OF PEOPLE WITH THE SAME SCORE 39](#_Toc529118545)

[Explanation: 39](#_Toc529118546)

[Database 39](#_Toc529118547)

[Diagram(s) of tables 39](#_Toc529118548)

[Columns from Standard view 40](#_Toc529118549)

[Query 40](#_Toc529118550)

[Sample Output with total number of rows returned (6) 41](#_Toc529118551)

[Problem 12: 42](#_Toc529118552)

[-- FIND HOW MANY CUSTOMERS WE HAVE IN EACH GEOGRAPHY KEY 42](#_Toc529118553)

[Explanation: 42](#_Toc529118554)

[Database 42](#_Toc529118555)

[Diagram(s) of tables 42](#_Toc529118556)

[Columns from Standard view 43](#_Toc529118557)

[Query 43](#_Toc529118558)

[Sample Output with total number of rows returned (336) 44](#_Toc529118559)

[Problem 13: 45](#_Toc529118560)

[-- #OF ORDERS PER SHIPPING COMPANY 45](#_Toc529118561)

[Explanation: 45](#_Toc529118562)

[Database 45](#_Toc529118563)

[Diagram(s) of tables 45](#_Toc529118564)

[Columns from Standard view 46](#_Toc529118565)

[Query 46](#_Toc529118566)

[Sample Output with total number of rows returned (3) 47](#_Toc529118567)

[Problem 14: 48](#_Toc529118568)

[Explanation: 48](#_Toc529118569)

[Database 48](#_Toc529118570)

[Diagram(s) of tables 48](#_Toc529118571)

[Columns from Standard view 49](#_Toc529118572)

[Query 49](#_Toc529118573)

[Sample Output with total number of rows returned (2155) 50](#_Toc529118574)

[Problem 15: 51](#_Toc529118575)

[Explanation: 51](#_Toc529118576)

[Database 51](#_Toc529118577)

[Diagram(s) of tables 51](#_Toc529118578)

[Columns from Standard view 52](#_Toc529118579)

[Query 52](#_Toc529118580)

[Sample Output with total number of rows returned (14) 53](#_Toc529118581)

[Problem 16: 54](#_Toc529118582)

[Explanation: 54](#_Toc529118583)

[Database 54](#_Toc529118584)

[Diagram(s) of tables 54](#_Toc529118585)

[Columns from Standard view 55](#_Toc529118586)

[Query 55](#_Toc529118587)

[Sample Output with total number of rows returned (2155) 55](#_Toc529118588)

[Problem 17: 56](#_Toc529118589)

[Explanation: 56](#_Toc529118590)

[Database 56](#_Toc529118591)

[Diagram(s) of tables 56](#_Toc529118592)

[Columns from Standard view 57](#_Toc529118593)

[Query 57](#_Toc529118594)

[Sample Output with total number of rows returned (2155) 57](#_Toc529118595)

[Problem 18: 58](#_Toc529118596)

[Explanation: 58](#_Toc529118597)

[Database 58](#_Toc529118598)

[Diagram(s) of tables 58](#_Toc529118599)

[Columns from Standard view 59](#_Toc529118600)

[Query 59](#_Toc529118601)

[Sample Output with total number of rows returned (2155) 59](#_Toc529118602)

[Problem 19: 60](#_Toc529118603)

[Explanation: 60](#_Toc529118604)

[Database 60](#_Toc529118605)

[Diagram(s) of tables 60](#_Toc529118606)

[Columns from Standard view 61](#_Toc529118607)

[Query 61](#_Toc529118608)

[Sample Output with total number of rows returned (2155) 61](#_Toc529118609)

[Problem 20: 62](#_Toc529118610)

[Explanation: 62](#_Toc529118611)

[Database 62](#_Toc529118612)

[Diagram(s) of tables 62](#_Toc529118613)

[Columns from Standard view 63](#_Toc529118614)

[Query 63](#_Toc529118615)

[Sample Output with total number of rows returned (2155) 63](#_Toc529118616)

[Problem 21: 64](#_Toc529118617)

[Explanation: 64](#_Toc529118618)

[Database 64](#_Toc529118619)

[Diagram(s) of tables 64](#_Toc529118620)

[64](#_Toc529118621)

[Columns from Standard view 64](#_Toc529118622)

[64](#_Toc529118623)

[Query 65](#_Toc529118624)

[Sample Output with total number of rows returned (3) 65](#_Toc529118625)

[Problem 22: 66](#_Toc529118626)

[Explanation: 66](#_Toc529118627)

[Database 66](#_Toc529118628)

[Diagram(s) of tables 66](#_Toc529118629)

[Columns from Standard view 68](#_Toc529118630)

[Query 68](#_Toc529118631)

[Sample Output with total number of rows returned (2155) 68](#_Toc529118632)

[Problem 23: 69](#_Toc529118633)

[Explanation: 69](#_Toc529118634)

[Database 69](#_Toc529118635)

[Diagram(s) of tables 69](#_Toc529118636)

[Columns from Standard view 70](#_Toc529118637)

[Query 70](#_Toc529118638)

[Sample Output with total number of rows returned (2155) 70](#_Toc529118639)

[Problem 24: 71](#_Toc529118640)

[Explanation: 71](#_Toc529118641)

[Database 71](#_Toc529118642)

[Diagram(s) of tables 71](#_Toc529118643)

[Columns from Standard view 72](#_Toc529118644)

[Query 72](#_Toc529118645)

[Sample Output with total number of rows returned (2155) 72](#_Toc529118646)

[Problem 25: 73](#_Toc529118647)

[Explanation: 73](#_Toc529118648)

[Database 73](#_Toc529118649)

[Diagram(s) of tables 73](#_Toc529118650)

[Columns from Standard view 74](#_Toc529118651)

[Query 74](#_Toc529118652)

[Sample Output with total number of rows returned (2155) 74](#_Toc529118653)

[Problem 26: 75](#_Toc529118654)

[Explanation: 75](#_Toc529118655)

[Database 75](#_Toc529118656)

[Diagram(s) of tables 75](#_Toc529118657)

[Columns from Standard view 76](#_Toc529118658)

[Query 76](#_Toc529118659)

[Sample Output with total number of rows returned (2155) 76](#_Toc529118660)

[Problem 27: 77](#_Toc529118661)

[Explanation: 77](#_Toc529118662)

[Database 77](#_Toc529118663)

[Diagram(s) of tables 77](#_Toc529118664)

[Columns from Standard view 78](#_Toc529118665)

[Query 78](#_Toc529118666)

[Sample Output with total number of rows returned (2155) 78](#_Toc529118667)

[Problem 28: 79](#_Toc529118668)

[Explanation: 79](#_Toc529118669)

[Database 79](#_Toc529118670)

[Diagram(s) of tables 79](#_Toc529118671)

[Columns from Standard view 80](#_Toc529118672)

[Query 80](#_Toc529118673)

[Sample Output with total number of rows returned (2155) 80](#_Toc529118674)

[Problem 29: 81](#_Toc529118675)

[Explanation: 81](#_Toc529118676)

[Database 81](#_Toc529118677)

[Diagram(s) of tables 81](#_Toc529118678)

[Columns from Standard view 82](#_Toc529118679)

[Query 82](#_Toc529118680)

[Sample Output with total number of rows returned (2155) 82](#_Toc529118681)

[Problem 30: 83](#_Toc529118682)

[Explanation: 83](#_Toc529118683)

[Database 83](#_Toc529118684)

[Diagram(s) of tables 83](#_Toc529118685)

[Columns from Standard view 84](#_Toc529118686)

[Query 84](#_Toc529118687)

[Sample Output with total number of rows returned (2155) 84](#_Toc529118688)

# Problem 01:

## -- PROVIDE A LIST OF JOB CANDIDATES WHO GOT HIRED ALONG WITH THEIR RESPECTIVE JOB TITLE AND HIRE DATE

## Explanation:

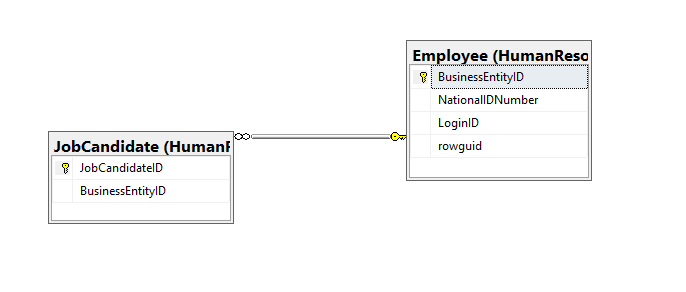
## -- OUTPUT SHOULD SHOW ANYONE WHO GOT HIRED AND THE TITLE THEY RECEIVED ALONG WITH THE HIRE DATE

## Database

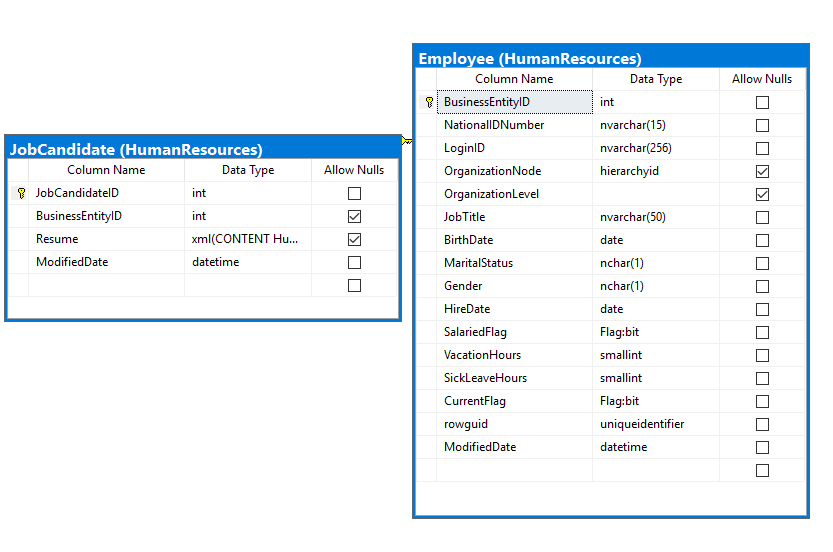
USE AdventureWorks2014;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE AdventureWorks2014;

SELECT JC.JobCandidateID

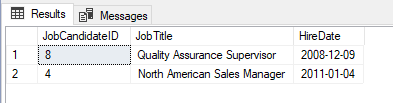
,E.JobTitle

,E.HireDate

FROM HumanResources.Employee AS E

INNER JOIN HumanResources.JobCandidate AS JC ON E.BusinessEntityID = JC.BusinessEntityID;

## Sample Output with total number of rows returned (2)



# Problem 02:

## -- PROVIDE A LIST OF CITIES WITHT THE POSSIBLE DELIVERY METHODS

## Explanation:

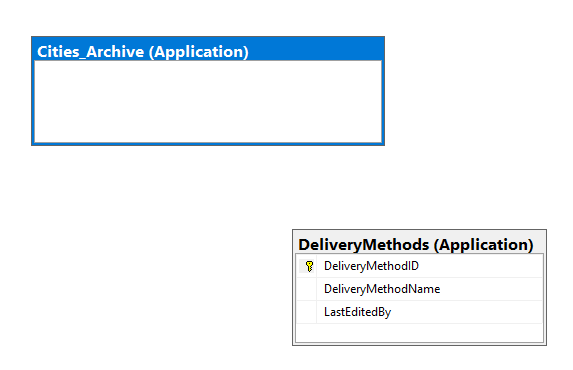
-- CROSS JOIN ALL THE POSSIBILITIES

## Database

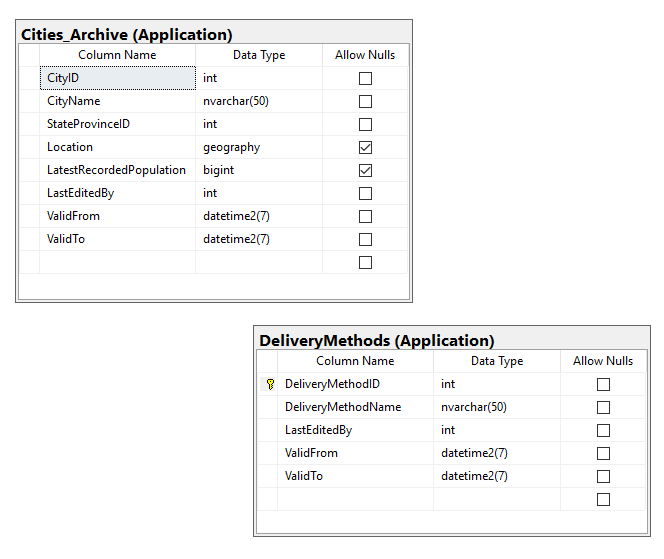
USE WideWorldImporters;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE WideWorldImporters;

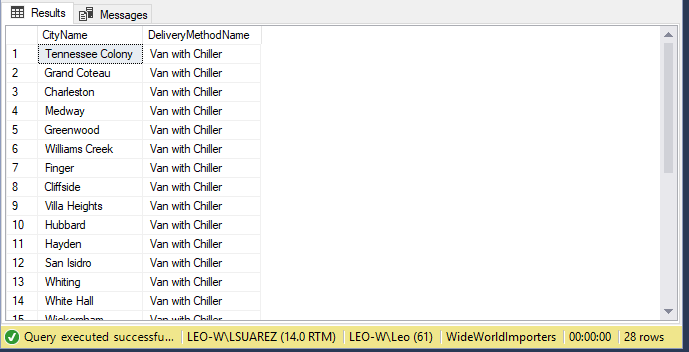
SELECT CA.CityName

,DM.DeliveryMethodName

FROM Application.Cities\_Archive AS CA

CROSS JOIN Application.DeliveryMethods\_Archive AS DM

## Sample Output with total number of rows returned (28)



# Problem 03:

## -- PROVIDE LIST OF LAST NAMES THAT INCLUDE AT LEAST ONE "a"

## Explanation:

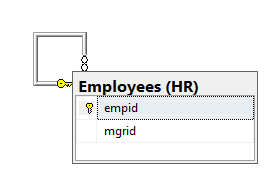
-- RUN THROUGH ALL THE NAMES AND ONLY RETURN THOSE WHICH HAVE 1 ONE MORE LETTERS “a”

## Database

USE TSQLV4;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view

## 

## Query

USE TSQLV4;

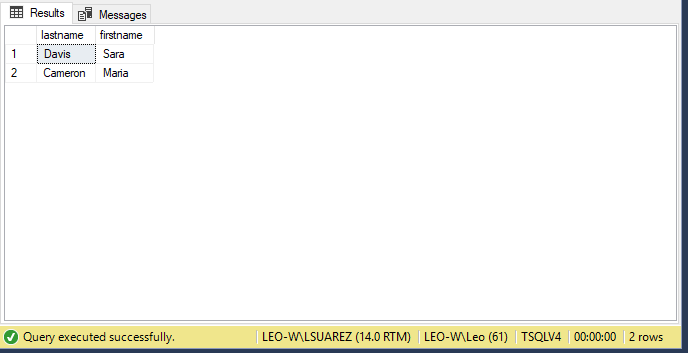
SELECT lastname

,firstname

FROM HR.Employees

WHERE LastName LIKE '%a%';

## Sample Output with total number of rows returned (2)



# Problem 04:

## -- PROVIDE A LIST OF PRODUCTS, THEIR PRICE AND THE CATEGORY THEY BELONG TO

## Explanation:

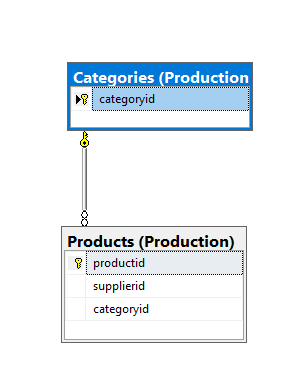
-- EACH PRODUCT SHOULD LIST THEIR PRICE AND THE CATEGORY THEY CORRESPOND TO

## Database

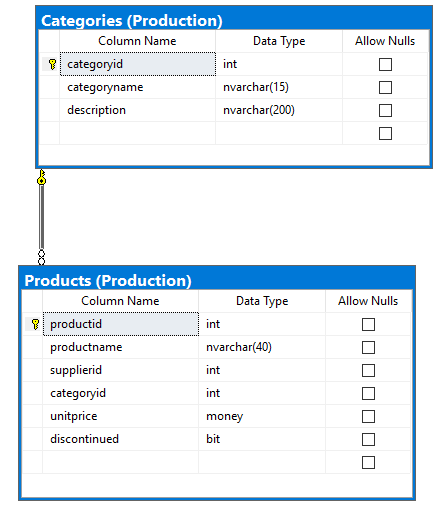
USE TSQLV4

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE TSQLV4

SELECT P.productname AS Product

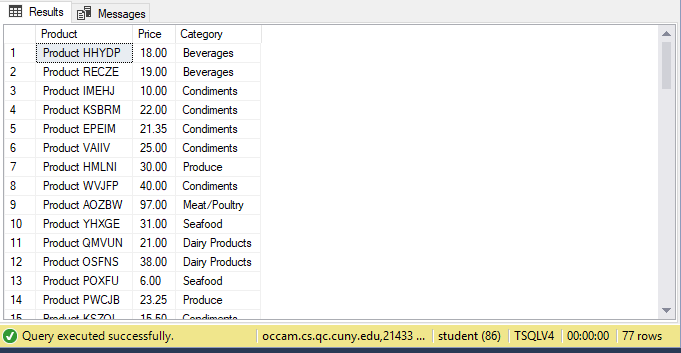
, P.unitprice AS Price

, C.categoryname AS Category

FROM Production.Products as P

INNER JOIN Production.Categories AS C ON P.categoryid = C.categoryid

## Sample Output with total number of rows returned (77)



# Problem 05:

## -- PROVIDE A LIST OF AVAILABLE CURRENCY FORMS

## Explanation:

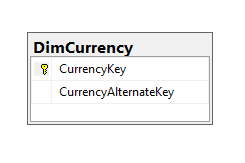
-- LIST ALL AVAILABLE CURRENCY FORMS

## Database

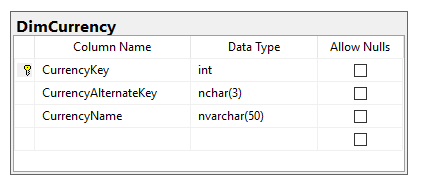
USE AdventureWorksDW2014;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

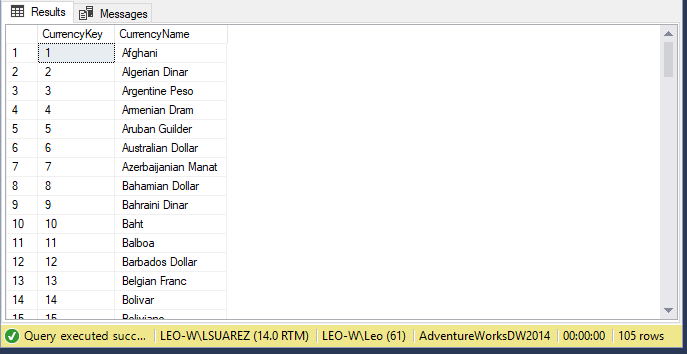
USE AdventureWorksDW2014;

SELECT CurrencyKey

,CurrencyName

FROM dbo.DimCurrency

## Sample Output with total number of rows returned (105)



# Problem 06:

## -- PROVIDE A LIST OF ALL ORDERS WITH A REASON TYPE THAT IS NOT “OTHER”

## Explanation:

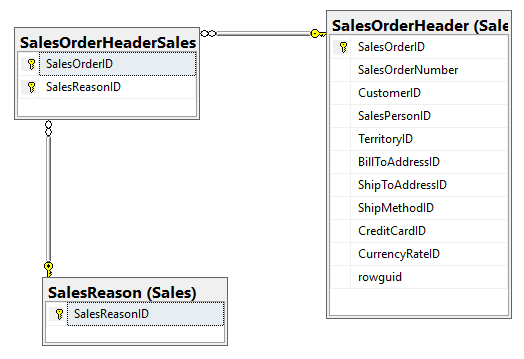
-- EVERY ORDER THAT IS RETURNED SHOULD NOT INDICATE “OTHER” AS A REASON TYPE

## Database

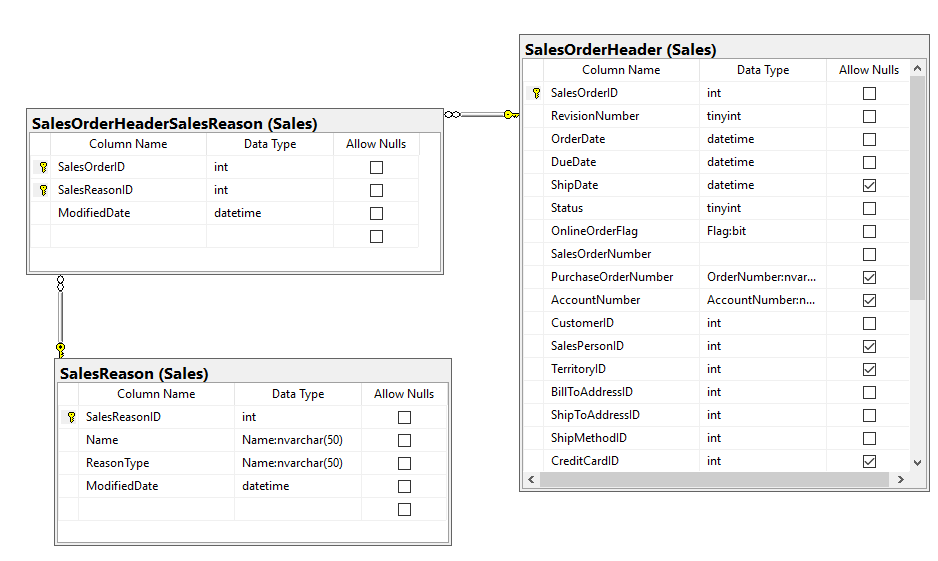
use AdventureWorks2014;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

use AdventureWorks2014;

select OH.CustomerID, OH.SalesOrderNumber,SR.ReasonType

from sales.SalesOrderHeader as OH

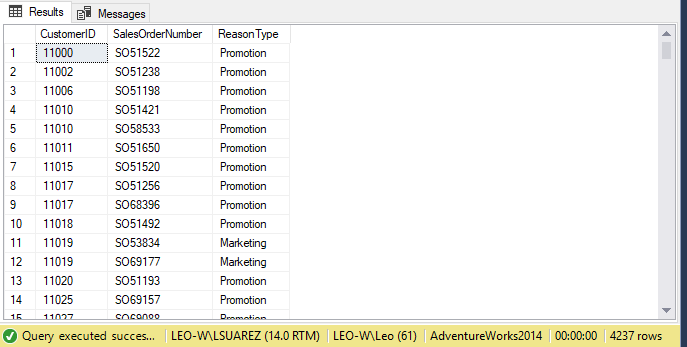
join sales.SalesOrderHeaderSalesReason as OHR

on OH.SalesOrderID = OHR.SalesOrderID

join Sales.SalesReason as SR

on (SR.SalesReasonID = OHR.SalesReasonID and not SR.ReasonType = 'other')

## Sample Output with total number of rows returned (4237)



# Problem 07:

## -- make sure that every employee in the engineering department has a login ID

## Explanation:

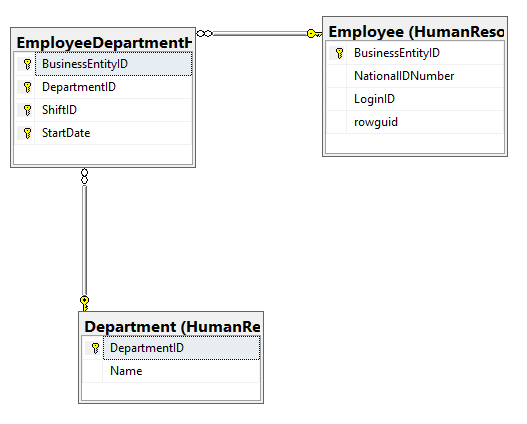
-- make sure that every employee in the engineering department has a login ID

## Database

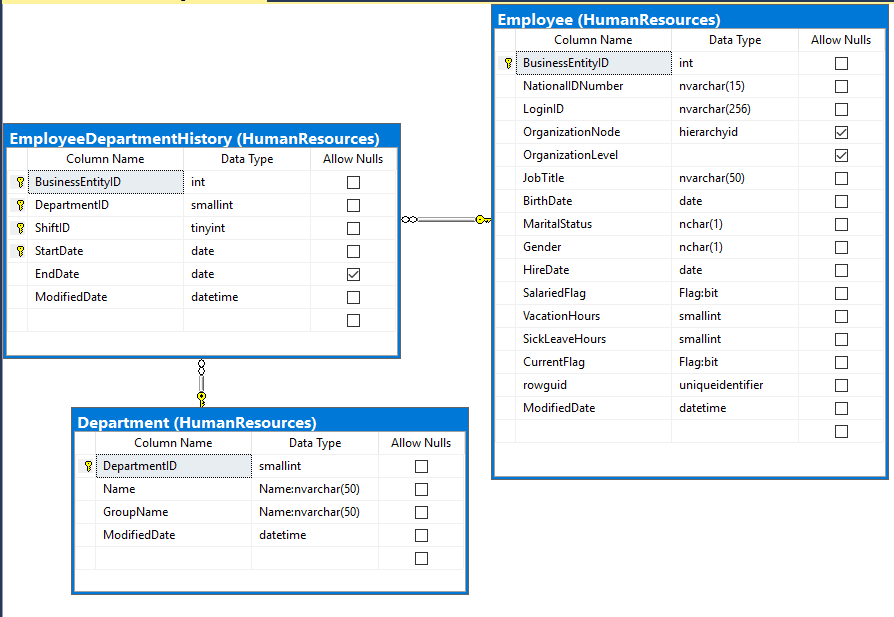
use AdventureWorks2014;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

use AdventureWorks2014;

select E.LoginID, D.DepartmentID, D.[Name]

from HumanResources.Employee as E

join HumanResources.EmployeeDepartmentHistory as DH

on e.BusinessEntityID = DH.BusinessEntityID

join HumanResources.Department as D

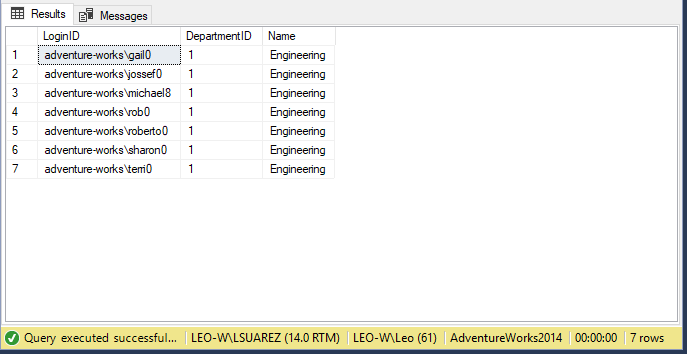
on (d.DepartmentID = 1 and DH.DepartmentID = 1)

GROUP BY D.DepartmentID

,LoginID

,D.[Name]

## Sample Output with total number of rows returned (7)



# Problem 08:

## -- PROVIDE A LIST OF ALL TRANSACTIONS WITH UNKNOWN PAYMENT METHOD THAT OCCURED AT THE END OF THE MONTH IN 2016

## Explanation:

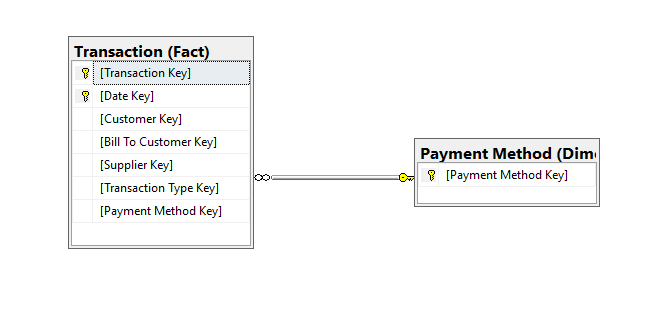
-- PROVIDE A LIST OF ALL TRANSACTIONS WITH UNKNOWN PAYMENT METHOD THAT OCCURED AT THE END OF THE MONTH IN 2016

## Database

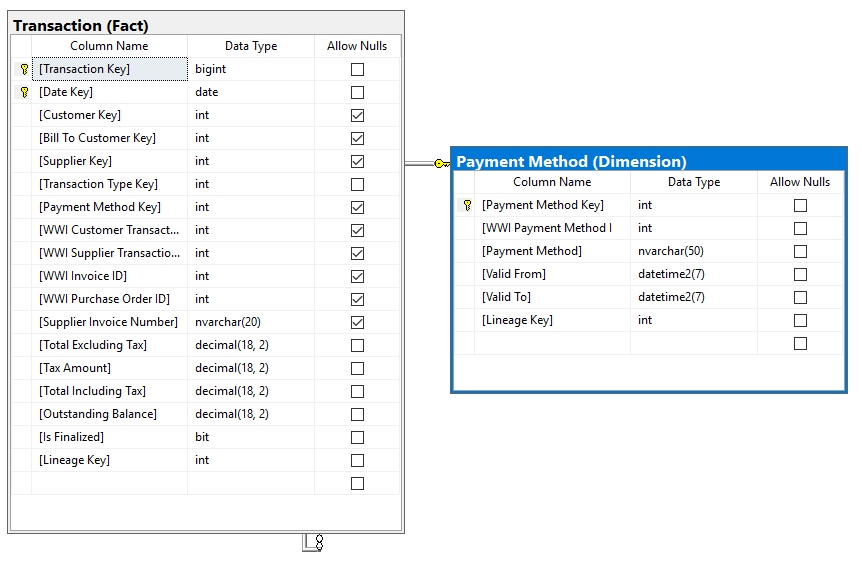
USE WideWorldImportersDW;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE WideWorldImportersDW;

SELECT T.[Transaction Key]

,PM.[Payment Method]

,T.[Date Key] AS DATE

FROM Fact.[Transaction] AS T

INNER JOIN Dimension.[Payment Method] AS PM

ON T.[Payment Method Key] = PM.[Payment Method Key] AND PM.[Payment Method] = 'Unknown'

WHERE T.[Date Key] = EOMONTH(T.[Date Key])

AND YEAR(T.[Date Key]) = 2016

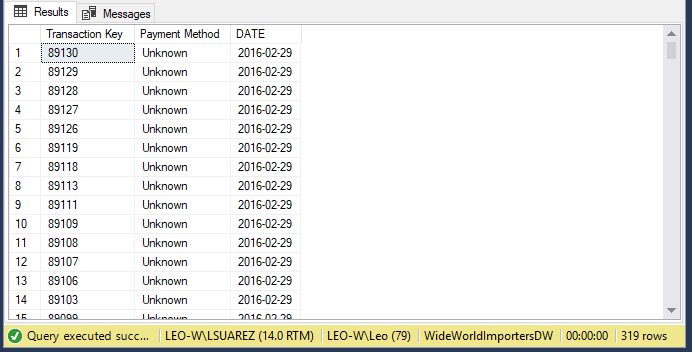
GROUP BY T.[Transaction Key]

,PM.[Payment Method]

,T.[Date Key]

ORDER BY T.[Date Key]

## Sample Output with total number of rows returned (319)



# Problem 09:

## -- PROVIDE A LIST OF STOCK ITEMS SOLD

## Explanation:

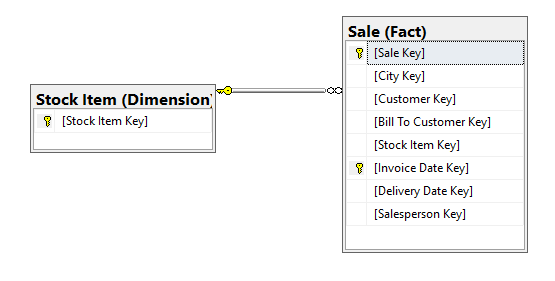
-- PROVIDE A LIST OF STOCK ITEMS SOLD

## Database

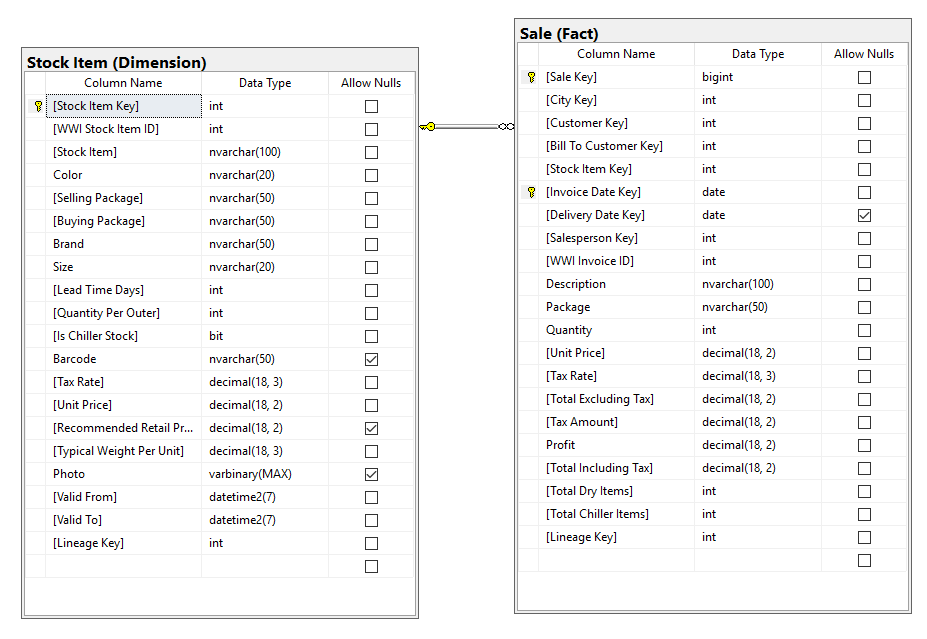
USE WideWorldImportersDW

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE WideWorldImportersDW

SELECT DSI.[Stock Item Key]

,DSI.[Stock Item]

,SUM(FS.Quantity) AS QuantitySold

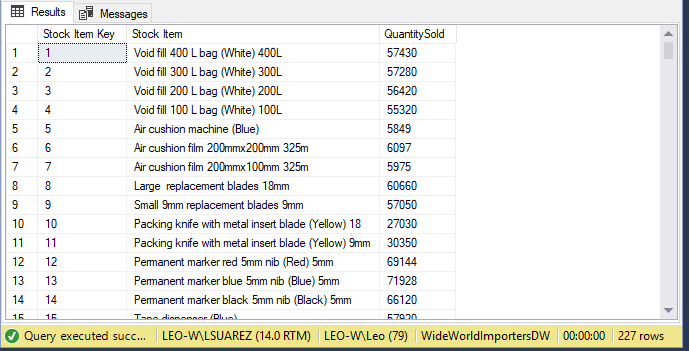
FROM Dimension.[Stock Item] AS DSI

INNER JOIN Fact.Sale AS FS ON FS.[Stock Item Key] = DSI.[Stock Item Key]

GROUP BY DSI.[Stock Item]

,DSI.[Stock Item Key]

## Sample Output with total number of rows returned (227)



# Problem 10:

## -- PROVIDE A LIST THAT SHOWS IF ANY EMPLOYEE RATE CHANGED AT THE BEGINNING OF 2015

## Explanation:

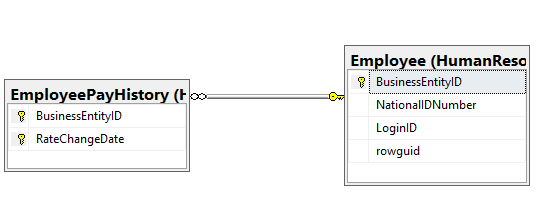
-- PROVIDE A LIST THAT SHOWS IF ANY EMPLOYEE RATE CHANGED AT THE BEGINNING OF 2015

## Database

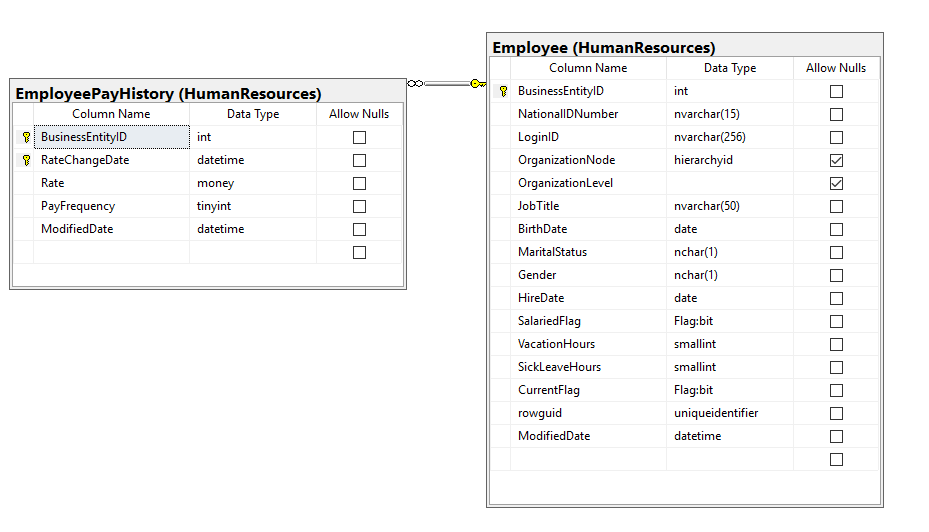
USE AdventureWorks2014

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE AdventureWorks2014

SELECT E.JobTitle

,PH.RateChangeDate

FROM HumanResources.Employee AS E

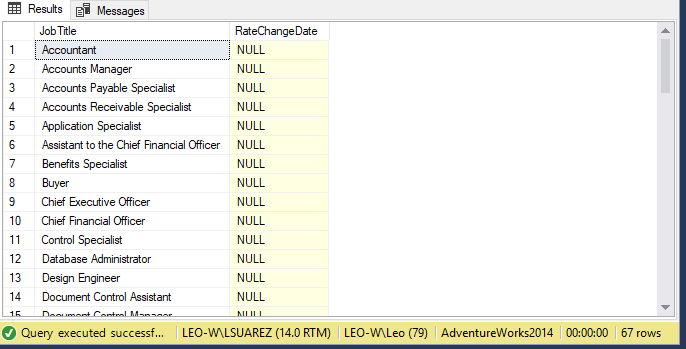
LEFT JOIN HumanResources.EmployeePayHistory AS PH

ON PH.RateChangeDate = CAST('20150101'AS DATE)

GROUP BY E.JobTitle

,PH.RateChangeDate

## Sample Output with total number of rows returned (67)



# Problem 11:

## -- COUNT THE NUMBER OF PEOPLE WITH THE SAME SCORE

## Explanation:

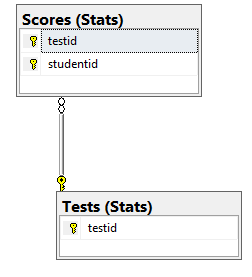
-- COUNT THE NUMBER OF PEOPLE WITH THE SAME SCORE

## Database

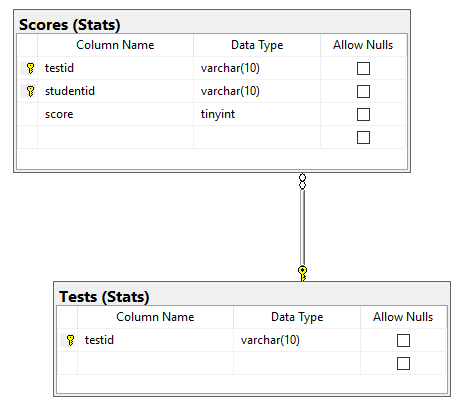
USE TSQLV4;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE TSQLV4;

SELECT

COUNT(\*) AS PeopleWithTheSameScore,

S.score AS Score

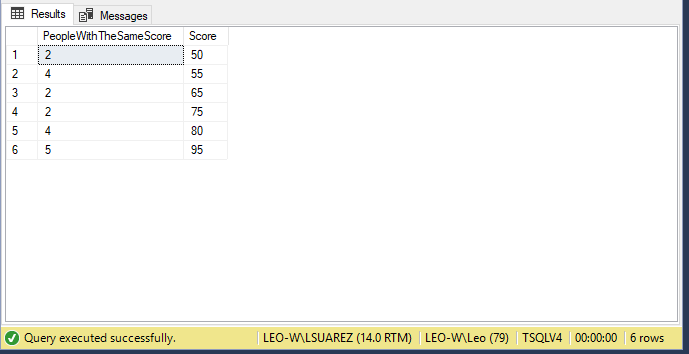
FROM

Stats.Tests AS T INNER JOIN Stats.Scores AS S

ON t.testid = s.testid

GROUP BY s.score

## Sample Output with total number of rows returned (6)



# Problem 12:

## -- FIND HOW MANY CUSTOMERS WE HAVE IN EACH GEOGRAPHY KEY

## Explanation:

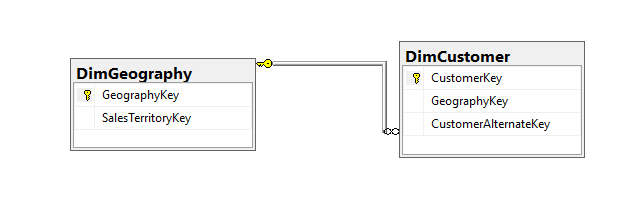
-- FIND HOW MANY CUSTOMERS WE HAVE IN EACH GEOGRAPHY KEY

## Database

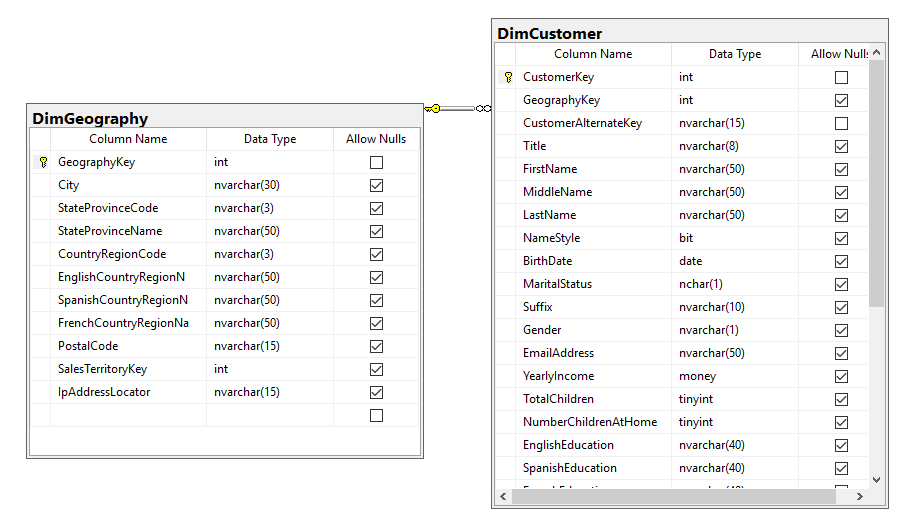
USE AdventureWorksDW2014;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE AdventureWorksDW2014;

SELECT

COUNT(DISTINCT C.CustomerKey) AS #CUSTOMERS,

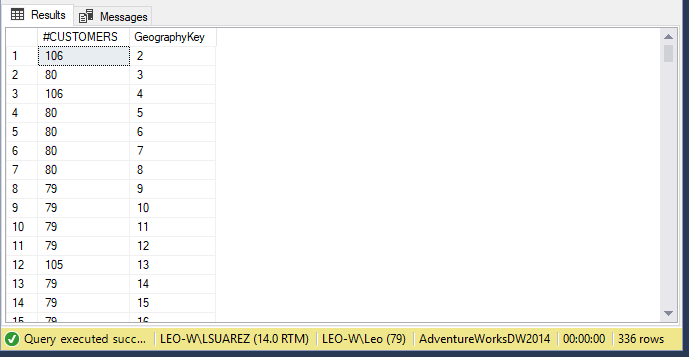
g.GeographyKey

FROM

DimCustomer AS C INNER JOIN DimGeography AS G ON C.GeographyKey = G.GeographyKey

GROUP BY g.GeographyKey

## Sample Output with total number of rows returned (336)



# Problem 13:

## -- #OF ORDERS PER SHIPPING COMPANY

## Explanation:

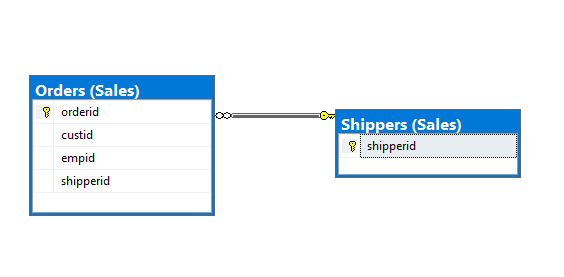
-- #OF ORDERS PER SHIPPING COMPANY

## Database

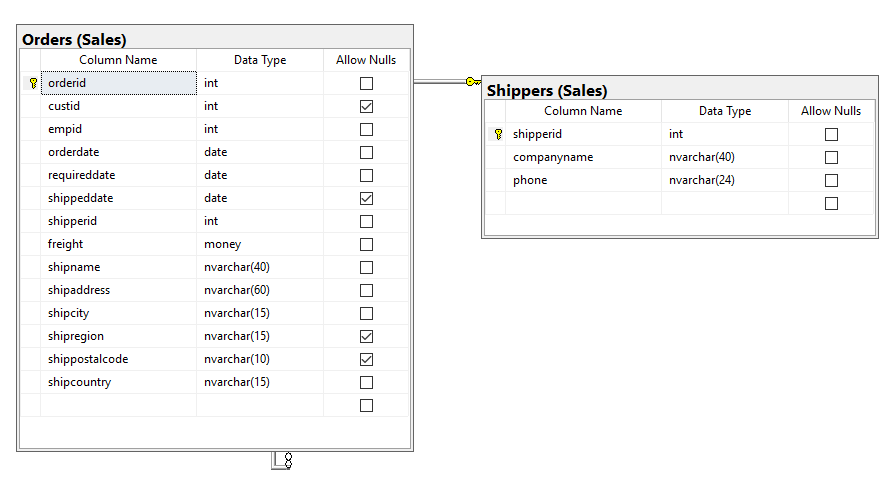
USE TSQLV4;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE TSQLV4;

SELECT

COUNT( DISTINCT O.orderid) AS orders,

companyname

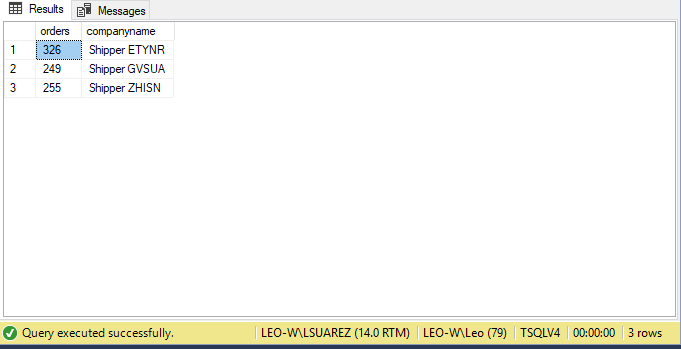
FROM Sales.Shippers AS SP

INNER JOIN Sales.Orders AS O

ON SP.shipperid = O.shipperid

GROUP BY companyname

## Sample Output with total number of rows returned (3)



# Problem 14:

## --- PROVIDE A LIST OF THE TOP 200 STOCK ITEMS WITH THE HIGHEST LINE PROFIT

## Explanation:

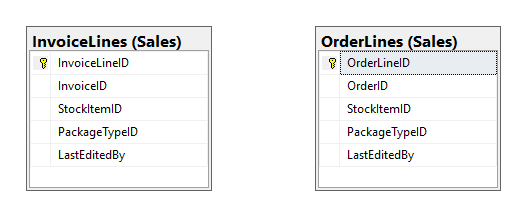
--- THE LIST SHOULD BE ORDERED BY LINE PROFIT IN DESCENDING ORDER, PROVIDE DESCRIPTION AND ORDER LINE ID AS WELL AS WHEN THE PICKING WAS COMPLETED

## Database

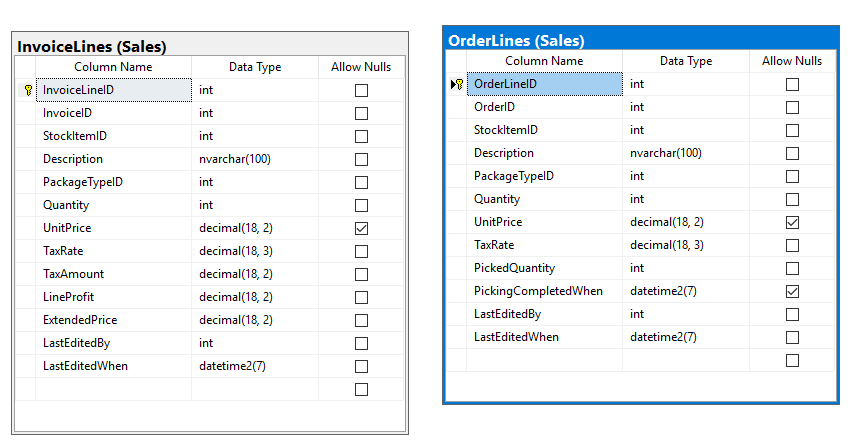
use WideWorldImporters

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

use WideWorldImporters

select top 200 OL.StockItemID,

OL.[Description],

OL.OrderLineID,

OL.PickingCompletedWhen,

IL.LineProfit

from Sales.InvoiceLines as IL

inner join (select StockItemID,

[Description],

OrderLineID,

PickingCompletedWhen

from Sales.OrderLines

WHERE PickingCompletedWhen Between CAST('20160201' as DATE) and CAST('20160301' as DATE)

group by StockItemID,

[Description],

OrderLineID,

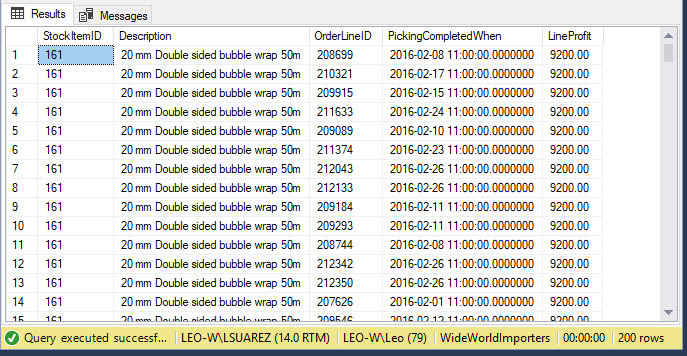
PickingCompletedWhen

)as OL

on IL.StockItemID = OL.StockItemID

order by IL.LineProfit desc

## Sample Output with total number of rows returned (2155)



# Problem 15:

## --- PROVIDE A LIST OF CUSTUMERS AND SUPPLIERS WHERE THE CITY AND COUNTRY MATCH

## Explanation:

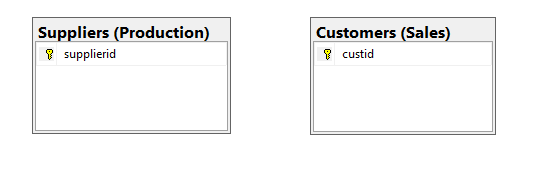
--- LIST SHOULD JOIN CUSTID AND SUPPLIERID BY CITY AND COUNTRY AS CUSTOMER REGION, SHOW COMPANY NAME AND CONTACT TITLE

## Database

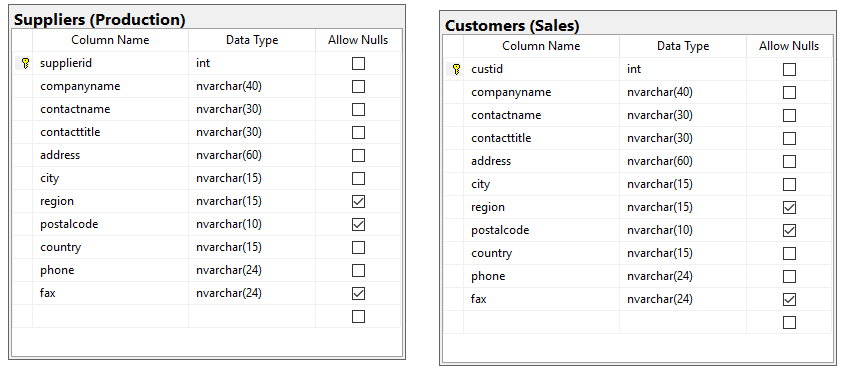
use TSQLV4

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

use TSQLV4

select cu.custid,

CU.companyname,

CU.contacttitle,

CONCAT(CU.city ,', ', CU.country) as CustomerRegion,

SU.supplierid,

SU.companyname,

SU.contacttitle,

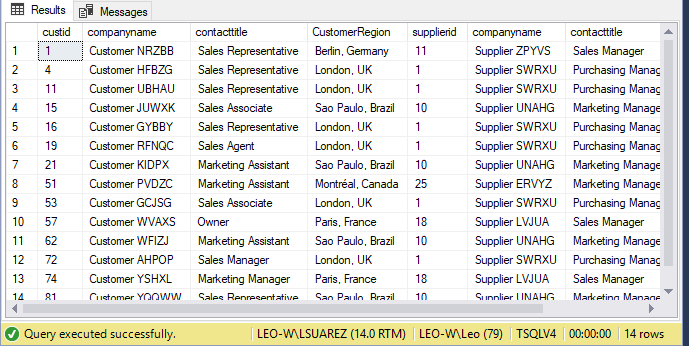
CONCAT(SU.city ,', ', SU.country) as SupplierRegion

from sales.customers as CU

inner join Production.Suppliers as SU

on cu.country = SU.country and cu.city = su.city

## Sample Output with total number of rows returned (14)



# Problem 16:

## --PROVIDE A LIST OF EMPLOYEE WHOSE PAY RATE CHANGED BETWEEEN 2009 AND 2010

## Explanation:

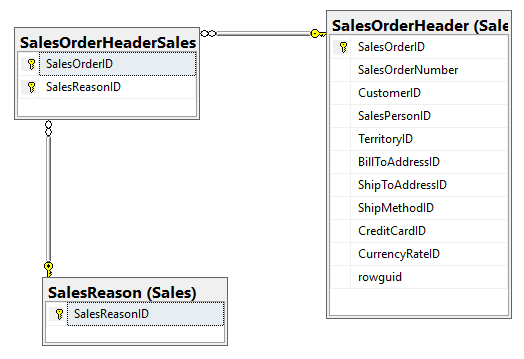
--JOIN TWO HUMAN RESOURCES AND EMPLOYEE PAY HISTORY MATCHED BY BUSINESS ENTITY ID AND SHOW ONLY VALUES BETWEEN 2009 AND 2010

## Database

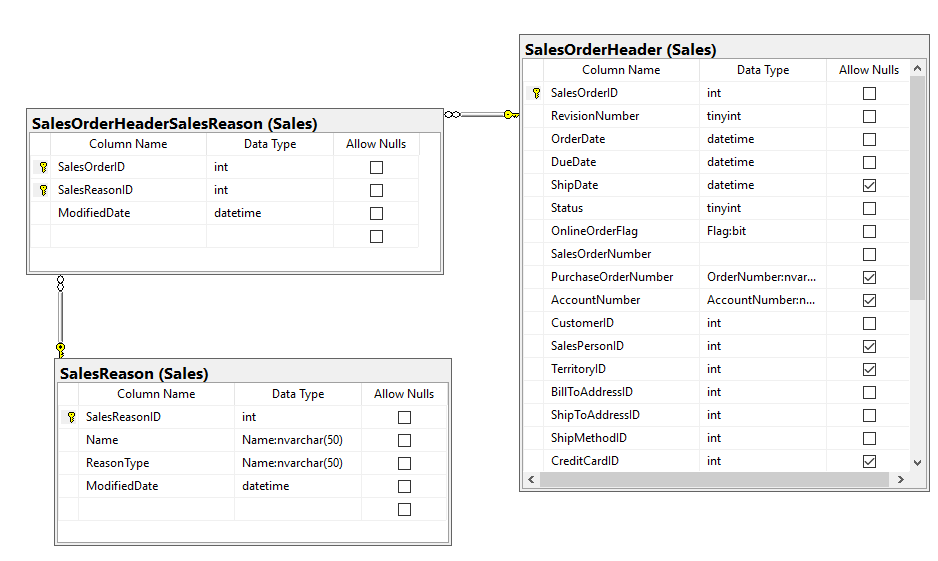
USE AdventureWorks2014

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE AdventureWorks2014

SELECT E.JobTitle

,PH.RateChangeDate

FROM HumanResources.Employee AS E

INNER JOIN HumanResources.EmployeePayHistory AS PH

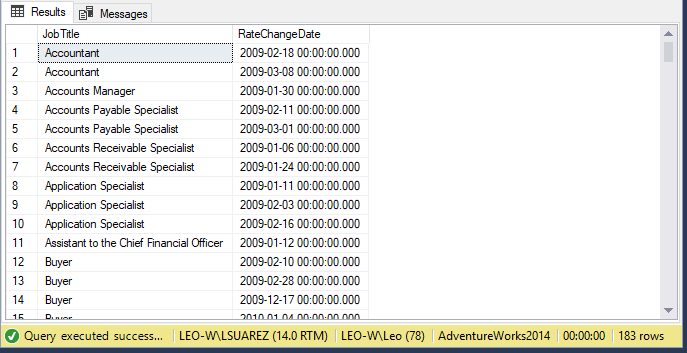
ON e.BusinessEntityID = ph.BusinessEntityID

WHERE YEAR(PH.RateChangeDate) BETWEEN 2009 AND 2010

GROUP BY E.JobTitle

,PH.RateChangeDate

## Sample Output with total number of rows returned (183)



# Problem 17:

## --PROVIDE ALL ETF PURCHASES BETWEEN 2013 AND 2016

## Explanation:

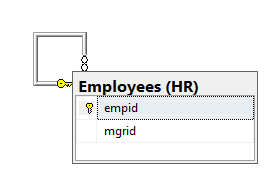
--PROVIDE ALL ETF PURCHASES BETWEEN 2013 AND 2016

## Database

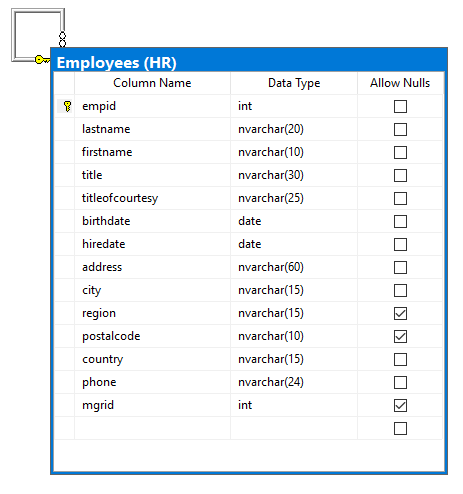
USE WideWorldImportersDW;

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join



## Columns from Standard view



## Query

USE WideWorldImportersDW;

SELECT T.[Transaction Key]

,PM.[Payment Method]

,T.[Date Key] AS DATE

FROM Fact.[Transaction] AS T

INNER JOIN Dimension.[Payment Method] AS PM

ON T.[Payment Method Key] = PM.[Payment Method Key]

WHERE YEAR(T.[Date Key]) BETWEEN 2013 AND 2016

AND PM.[Payment Method] = 'EFT'

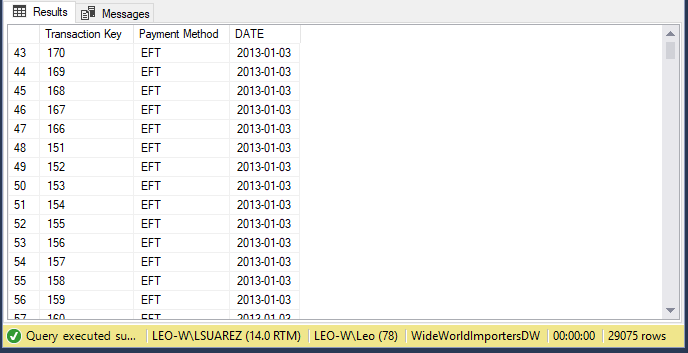
GROUP BY T.[Transaction Key]

,PM.[Payment Method]

,T.[Date Key]

ORDER BY T.[Date Key]

## Sample Output with total number of rows returned (29075)



# Problem 18:

## Explanation:

## Database

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join

## Columns from Standard view

## Query

## Sample Output with total number of rows returned (2155)

# Problem 19:

## Explanation:

## Database

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join

## Columns from Standard view

## Query

## Sample Output with total number of rows returned (2155)

# Problem 20:

## Explanation:

## Database

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join

## Columns from Standard view

## Query

## Sample Output with total number of rows returned (2155)

# Problem 21:

## Explanation:

## Database

USE AdventureWorks2014

## Diagram(s) of tables

Foreign Key(s) or column(s) used for the join

## 

## Columns from Standard view

## 

## Query

USE AdventureWorks2014

DROP FUNCTION

IF EXISTS dbo.LastFourEquals;

GO

CREATE FUNCTION dbo.LastFourEquals

(

@LastFour int

)

RETURNS TABLE

AS

RETURN

SELECT

CAST(CardNumber AS bigint) AS number

FROM Sales.CreditCard

WHERE (CAST(CardNumber AS bigint) % 10000) = @LastFour

USE AdventureWorks2014

SELECT number as CreditCardNumber

-- ,LF.ExpMonth

--,LF.CardType

FROM dbo.LastFourEquals(1254) as number

INNER JOIN Sales.CreditCard AS CR ON CAST(CR.CardNumber as bigint) = number

GROUP BY number

## Sample Output with total number of rows returned (3)

