--1.List of Persons’ full name, all their fax and phone numbers, as well as the phone number and fax of the company they are working for (if any).

SELECT Application.people.FullName, Application.people.FaxNumber, Application.People.PhoneNumber, Purchasing.Suppliers.PhoneNumber, Purchasing.Suppliers.FaxNumber

from Application.People

inner join Purchasing.Suppliers

on application.People.PersonID = Purchasing.Suppliers.PrimaryContactPersonID

--2. If the customer's primary contact person has the same phone number as the customer’s phone number, list the customer companies.

SELECT distinct Purchasing.Suppliers.SupplierName

from Purchasing.Suppliers

inner join Sales.Customers

on Purchasing.Suppliers.PhoneNumber = sales.Customers.PhoneNumber

--3. List of customers to whom we made a sale prior to 2016 but no sale since 2016-01-01.

--SELECT distinct CustomerID from sales.orders where orderdate < '01/01/2016'

--4. List of Stock Items and total quantity for each stock item in Purchase Orders in Year 2013.

--select stockitemid, quantityperouter, LeadTimeDays from warehouse.StockItems

--SELECT distinct stockitemID, sum(Quantity) as totalquantity from warehouse.stockitems group by stockitemID

--5. List of stock items that have at least 10 characters in description.

--SELECT stockitemid, stockitemname, searchdetails from warehouse.StockItems

--WHERE LEN(StockItemNAME) > 10

--SELECT PURCHASEORDERID, COMMENTS, INTERNALCOMMENTS FROM PURCHASING.PurchaseOrders

--SELECT DISTINCT PurchaseOrderID FROM PURCHASING.PurchaseOrderLines WHERE LEN([Description]) > 10

--6. List of stock items that are not sold to the state of Alabama and Georgia in 2014.

--SELECT \* FROM Application.StateProvinces

--SELECT \* FROM Warehouse.StockItems

--7. List of States and Avg dates for processing (confirmed delivery date – order date).

SELECT customerid,buyinggroupid,paymentdays,deliveryaddressline2, deliverypostalcode,postalpostalcode, ValidFrom from sales.customers

SELECT stateprovincecode, StateProvinceid from Application.StateProvinces

--8. List of States and Avg dates for processing (confirmed delivery date – order date) by month.

select Application.StateProvinces.StateProvinceName, MONTH(Sales.orders.OrderDate) as diff\_by\_month, AVG(DATEDIFF(DAY, Sales.Orders.OrderDate, sales.Invoices.ConfirmedDeliveryTime)) as avg\_different\_day

from sales.orders

inner join sales.Invoices on sales.Orders.CustomerID= sales.Invoices.CustomerID

inner join sales.Customers on sales.Customers.CustomerID = sales.Orders.CustomerID

inner join Application.Cities on Application.Cities.CityID = Sales.Customers.DeliveryCityID

inner join Application.StateProvinces on Application.StateProvinces.StateProvinceID = Application.Cities.StateProvinceID

group by Application.StateProvinces.StateProvinceName, MONTH(Sales.orders.OrderDate)

--9. List of StockItems that the company purchased more than sold in the year of 2015.

SELECT distinct warehouse.StockItems.StockItemName

from warehouse.StockItems

inner join sales.orderLines

on abs(warehouse.StockItems.QuantityPerOuter - sales.orderLines.PickedQuantity) > 0 and year(Sales.OrderLines.PickingCompletedWhen) = 2015

--10. List of Customers and their phone number, together with the primary contact person’s name, to whom we did not sell more than 10 mugs (search by name) in the year 2016.

SELECT sales.Customers.CustomerName, sales.Customers.PhoneNumber, Application.People.FullName

from Sales.Customers

INNER join Application.People on sales.Customers.PrimaryContactPersonID = Application.People.PersonID

INNER join Sales.Invoices on sales.Customers.CustomerID = sales.Invoices.CustomerID and YEAR(Sales.Invoices.InvoiceDate) = 2016

--11. List all the cities that were updated after 2015-01-01.

SELECT CityName FROM Application.cities

Where ValidFrom > '2015-01-01'

--12. List all the Order Detail (Stock Item name, delivery address, delivery state, city, country, customer name, customer contact person name, customer phone, quantity) for the date of 2014-07-01. Info should be relevant to that date.

SELECT warehouse.StockItems.StockItemName

from warehouse.StockItems

INNER join Sales.Orders on Warehouse.StockItems.StockItemID = sales.Orders.OrderID and sales.orders.OrderDate = '2014-07-01'

--13. List of stock item groups and total quantity purchased, total quantity sold, and the remaining stock quantity (quantity purchased – quantity sold)

select distinct warehouse.stockitemstockgroups.stockgroupid, Purchasing.PurchaseOrderLines.ReceivedOuters, sales.InvoiceLines.Quantity, (Purchasing.PurchaseOrderLines.ReceivedOuters-sales.InvoiceLines.Quantity) as remaining\_q

from Purchasing.PurchaseOrderLines

INNER JOIN sales.invoiceLines on Purchasing.PurchaseOrderLines.StockItemID = Sales.InvoiceLines.StockItemID

right join warehouse.stockitemstockgroups on Purchasing.PurchaseOrderLines.StockItemID = warehouse.stockitemstockgroups.stockitemid

ORDER BY Warehouse.StockItemStockGroups.StockGroupID

--14. List of Cities in the US and the stock item that the city got the most deliveries in 2016. If the city did not purchase any stock items in 2016, print “No Sales”.

SELECT top 1 application.cities.cityname , count(\*) as cities\_get\_delivered\_times

from sales.customers

inner join Application.Cities on Sales.Customers.DeliveryCityID = Application.Cities.CityID

inner join Sales.CustomerTransactions on Sales.Customers.CustomerID = sales.CustomerTransactions.CustomerID and YEAR(sales.CustomerTransactions.TransactionDate) = 2016

group by Application.Cities.CityName

order by cities\_get\_delivered\_times DESC

--15. List any orders that had more than one delivery attempt (located in invoice table).

SELECT OrderID from sales.Invoices

where JSON\_VALUE(Sales.Invoices.ReturnedDeliveryData, '$.events[1].comments') = 'Receiver not present'

--18. Create a view that shows the total quantity of stock items of each stock group sold (in orders) by year 2013-2017. [Stock Group Name, 2013, 2014, 2015, 2016, 2017]

create view tqn as

SELECT \* FROM

(

SELECT

OrderID,

orderdate

FROM

sales.Orders

) t

PIVOT(

count(OrderID)

FOR orderdate IN (

[2013],

[2014],

[2015],

[2016],

[2017] )

) AS pivot\_table;

GO

select \* from tqn

--20. Create a function, input: order id; return: total of that order. List invoices and use that function to attach the order total to the other fields of invoices.

CREATE FUNCTION func\_countordernum (@orderid int)

RETURNS int AS

BEGIN

DECLARE @invoiceid INT, @num INT;

SELECT @num = SUM(si.InvoiceID)

from sales.Invoices si

where si.orderID = @orderid

RETURN @num

END;