/\*A. Find employee having highest rate or highest pay frequency\*/

use adventureworks2022;

select\*from [HumanResources].[Employee]

select\*from person.person

select \*from [HumanResources].[EmployeePayHistory]

select max(PayFrequency),

(select concat\_ws(' ',firstname,LastName)

from person.person pp

where pp.businessEntityID=ep.businessentityId)Emp\_name

from [HumanResources].[EmployeePayHistory] ep

group by BusinessEntityID

--B. Analyze the inventory based on the shelf wise count of the product and their quantity.

select\*from Production.Product

select\*from [Production].[ProductInventory]

select distinct(pp.NAme),pi.Shelf,

count(pi.Shelf) over (partition by pp.Name )Count\_of\_shelf,

count(pi.Quantity ) over(partition by pp.Name)count\_per\_quantity

from [Production].[ProductInventory] pi,

Production.Product pp

--C. Find the personal details with address and address type

select \*from person.person

select\*from [Person].[Address]

select\*from [Person].[BusinessEntityAddress]

select pp.firstNAme,

pp.LastName,

pa.addressline1,

pa.AddressLine2,

be.addressTypeID

from person.person pp,

[Person].[Address] pa,

[Person].[BusinessEntityAddress] be

where pp.BusinessEntityID=be.BusinessEntityID and

pa.AddressID=be.addressID

--D. Find the job title having more revised payments

select\*from [HumanResources].[EmployeePayHistory]

select\*from [HumanResources].[Employee]

select e.jobtitle

from [HumanResources].[Employee] e,

[HumanResources].[EmployeePayHistory] ep

where

e.businessEntityID=ep.businessEntityID and

ep.payfrequency>0;

--E. Display special offer description, category and avg(discount pct) per the year

select\*from [Sales].[SpecialOffer]

select Description,

Category,

avg(discountPct) over (partition by year(StartDate)) Avg\_dct\_pct\_per\_year

from [Sales].[SpecialOffer]

--F. Display special offer description, category and avg(discount pct) per the month

select Description,

Category,

avg(discountPct) over (partition by month(StartDate)) Avg\_dct\_pct\_per\_month

from [Sales].[SpecialOffer]

--G. Using rank and dense rand find territory wise top sales person

select\*from [Sales].[SalesPerson]

select\*from [Sales].[SalesPersonQuotaHistory]

select\*from sales.specialOffer

select\*from sales.salesTerritory

select\*from person.person

select firstname from person.person

where BusinessEntityID in (select BusinessEntityID,

rank()over (partition by st.Name) Rank\_Name,

dense\_rank() over(partition by st.Name)DRank\_NAme

from[Sales].[SalesPerson] sp,

sales.salesTerritory st

where sp.TerritoryID=st.TerritoryID)

/\*H. Calculate total years of experience of the employee and

find out employees those who server for more than 20 years\*/

select\*from [HumanResources].[JobCandidate]

select\*from [HumanResources].[EmployeePayHistory]

select\*from [HumanResources].[Employee]

select\*from [HumanResources].[EmployeeDepartmentHistory]

select datediff(year,StartDate,Enddate) Total\_year\_ofExp

from [HumanResources].[EmployeeDepartmentHistory]

/\*I. Find the employee who is having more vacations than

the average vacation taken by all employees\*/

select\*from HumanResources.Employee

select businessEntityID ,

max(VacationHours)from HumanResources.Employee max\_vact\_hr

group by BusinessEntityID

/\*L. Is there any person having more than one credit card (hint: PersonCreditCard)\*/

--Required Tables

select\*from sales.PersonCreditCard

select\*From Person.Person

--Query

select distinct(BusinessEntityID)from

sales.PersonCreditCard

/\*#Conclusions:So, here we can see that all the businessEntityID are unique

beacuse rows before applying distinct and after are sames so there is

no person who is having more than one credit card \*/

--M. Find how many subcategories are available per product . (product sub category

select\*from Production.Product

select\*from Production.ProductCategory

select ProductID,

Name ProductNAme,

count(productsubcategoryID)over (partition by Name) Count\_ofSubcategory\_per\_name

from Production.Product

/\*N. Find total standard cost for the active Product where end date is not updated. (Product cost history)\*/

select\*from [Production].[ProductCostHistory]

select sum(standardcost) over (partition by ProductId)

from [Production].[ProductCostHistory]

where EndDate is null

/\*O. Which territory is having more customers (hint: customer)

(Non-anonymous question

)\*/

select\*from sales.SalesTerritory

select\*from sales.customer

select Name,

max(c.customerId)over (partition by t.territoryID)

from sales.customer c,

sales.SalesTerritory t

--Conclusions: Northwest is having more customers