```
use AdventureWorks2022;

create database customerDB;
use customerDB;

/* 1.	Create a customer table having following column with suitable data type
Cust_id  (automatically incremented primary key)
Customer name (only characters must be there)
Aadhar card (unique per customer)
Mobile number (unique per customer)
Date of birth (check if the customer is having age more than15)
Address
Address type code (B- business, H- HOME, O-office and should not accept any other)
State code ( MH – Maharashtra, KA for Karnataka)
*/


create table customer(
	cust_id int identity(1,1) primary key,
	customer_name varchar(30) not null check (customer_name not like '%[^A-Za-z]%'),
	adhar_no char(12) not null unique,
	mobile_no char(10) not null unique,
	dob date not null check (datediff(YY, dob, getdate())>15),
	adrs nvarchar(200) not null,
	adrs_typ_cod char(1) not null check (adrs_typ_cod in ('B','H','O')),
	state_code char(2) not null check (state_code in ('MH','KA'))
	);

--Create another table for Address type which is having
--Address type code must accept only (B,H,O)
--Address type  having the information as  (B- business, H- HOME, O-office)
```

```
create table address_type(
            adrs_typ_cod CHAR(1) PRIMARY KEY CHECK (adrs_typ_cod IN ('B', 'H', 'O')),
   adrs_typ_desc VARCHAR(50) not null
            );
/*
Create table state_info having columns as
State_id  primary unique
State name
Country_code char(2)
*/

create table state_info (
   state_id int primary key,
   state_name varchar(100) NOT NULL,
   country_code char(2) NOT NULL
            );
--Alter tables to link all tables based on suitable columns and foreign keys.
alter table customer
add constraint FK_custmr_adrs_typ
foreign key (adrs_typ_cod) references address_type(adrs_typ_cod);

alter table customer
add constraint FK_customer_state_info
foreign key (state_code) references state_info(state_id);

-- Change the column name from customer table customer name as c_name
alter table customer drop constraint CK__customer__custom__398D8EEE;
exec sp_rename 'customer.customer_name', 'c_name', 'COLUMN';
alter table customer
```

```
add constraint CK_customer_c_name
check (c_name NOT LIKE '%[^A-Za-z ]%');

--Insert the suitable records into the respective tables
select * from address_type;

insert into address_type (adrs_typ_cod, adrs_typ_desc) values
('B', 'Business'),
('H', 'Home'),
('O', 'Office');

insert into customer (c_name, adhar_no, mobile_no, dob, adrs, adrs_typ_cod, state_code) values
('sanket wade', '123456789012', '9876543210', '2000-05-15', '123 Street, Pune', 'H', 'MH'),
('KL Rahul', '987654321098', '9123456789', '1995-08-22', '456 Road, Bangalore', 'O', 'KA'),
('MS Dhoni', '567890123456', '9234567890', '1998-12-10', '789 Avenue, Mumbai', 'B', 'MH');

select * from customer;

--Change the data type of  country_code to varchar(3)
alter table state_info
alter column country_code varchar(3) not null;



-- Based on adventurework solve the following questions

use AdventureWorks2022;

--1. find the average currency rate conversion from USD to Algerian Dinar and Australian Doller
select * from Sales.CurrencyRate
select * from sales.Currency where Name ='Algerian Dinar'
select cr.ToCurrencyCode, cr.FromCurrencyCode, avg(cr.AverageRate) as avg_rate

from Sales.CurrencyRate cr

where cr.ToCurrencyCode in ('DZD', 'AUD')

group by cr.ToCurrencyCode, cr.FromCurrencyCode;




/* 2.Find the products having offer on it and display product name ,

safety Stock Level, Listprice,  and product model id, type of discount,

percentage of discount,  offer start date and offer end date*/


select * from Sales.SpecialOffer

select * from Production.Product

use adventureworks2022;


select

   (select name from production.product p where p.productid = sop.productid) as product_name,

   (select safetystocklevel from production.product p where p.productid = sop.productid) as
safetystocklevel,

   (select listprice from production.product p where p.productid = sop.productid) as listprice,

   (select productmodelid from production.product p where p.productid = sop.productid) as
productmodelid,

   (select type from sales.specialoffer so where so.specialofferid = sop.specialofferid) as
discount_type,

   (select discountpct * 100 from sales.specialoffer so where so.specialofferid = sop.specialofferid) as
percentage_discount,

   (select startdate from sales.specialoffer so where so.specialofferid = sop.specialofferid) as
offer_start_date,

   (select enddate from sales.specialoffer so where so.specialofferid = sop.specialofferid) as
offer_end_date

from sales.specialofferproduct sop

order by (select startdate from sales.specialoffer so where so.specialofferid = sop.specialofferid)
desc;
```

```
--3.create view to display Product name and Product review


select top 5 * from production.productreview;
use AdventureWorks2022;
create view vw_prodrvws as
select
   p.name as product_name,
   pr.comments as product_review
from production.product p,
    production.productreview pr
where p.productid = pr.productid;


select * from vw_prodrvws;
------------------------------------------------------------------------
-- 4.find out the vendor for product  paint, Adjustable Race and blade
select * from Purchasing.ProductVendor
select * from Purchasing.Vendor
select name from production.product where name in ('Paint', 'Adjustable Race', 'Blade');


select
   p.name as product_name,
   v.name as vendor_name
from purchasing.productvendor pv
join production.product p on pv.productid = p.productid
join purchasing.vendor v on pv.businessentityid = v.businessentityid
where p.name in ('Paint', 'Adjustable Race', 'Blade');


-- 5.find product details shipped through ZY - EXPRESS
select * from Purchasing.ShipMethod
select * from Purchasing.PurchaseOrderDetail
select * from Purchasing.PurchaseOrderHeader
select * from Production.Product


select distinct pp.productid, pp.name, pp.productnumber,
   pp.safetystocklevel, pp.reorderpoint, pp.standardcost,
   pp.listprice, pp.size, pp.weight, pp.color
from production.product pp,
    purchasing.purchaseorderdetail pod,
    purchasing.purchaseorderheader poh,
    purchasing.shipmethod sm
where sm.shipmethodid = poh.shipmethodid
  and pp.productid = pod.productid
  and poh.purchaseorderid = pod.purchaseorderid
  and sm.name = 'ZY - EXPRESS';


--6.find the tax amt for products where order date and ship date are on the same day
select
   poh.shipdate,
   soh.orderdate,
   poh.taxamt
from sales.salesorderheader soh,
    purchasing.purchaseorderheader poh
where poh.shipdate = soh.orderdate
and soh.shipmethodid = poh.shipmethodid;



-- 7.find the average days required to ship the product based on shipment type.


select  sm.name as shipment_type,
   avg(datediff(day, h.orderdate, h.shipdate)) as avg_shipping_days
from purchasing.purchaseorderheader h,  purchasing.shipmethod sm
where h.shipmethodid = sm.shipmethodid
```

```
group by sm.name;

-- 8.find the name of employees currently working in day shift

select p.firstname, p.lastname  from person.person p
where p.businessentityid in (
   select edh.businessentityid from humanresources.employeedepartmenthistory edh
   where edh.shiftid in ( select s.shiftid  from humanresources.shift s
   where s.name = 'Day')  -- Ensure correct case)
           );

--9. based on product and product cost history find the name ,
--  service provider time and average Standardcost

select
   prod.name as product_name,
   cost_hist.modifieddate as service_time,
   avg(cost_hist.standardcost) avg_cost
from production.product prod, production.productcosthistory cost_hist
where prod.productid = cost_hist.productid
group by prod.name, cost_hist.modifieddate
order by prod.name;

-- 10. find products with average cost more than 500



select
   prod.name as product_name,
   avg(cost_hist.standardcost) as avg_cost
from production.product prod, production.productcosthistory cost_hist
where prod.productid = cost_hist.productid
```

```
group by prod.name
having avg(cost_hist.standardcost) > 500
order by avg_cost ;

--11.find the employee who worked in multiple territory

select pp.firstname + ' ' + pp.lastname  employee_name,
      count(distinct tih.territoryid) territory_count
from sales.salesterritory ti, sales.salesterritoryhistory tih, person.person pp
where tih.businessentityid = pp.businessentityid
and ti.territoryid = tih.territoryid
group by pp.firstname, pp.lastname
having count(distinct tih.territoryid) > 1;

--12.find out the Product model name,  product description for culture as Arabic

select * from Production.ProductModel
select * from Production.ProductDescription
select * from Production.ProductModelProductDescriptionCulture
select * from Production.Culture

select pm.name as product_model,
      pd.description as product_description
from production.productmodel pm,  production.productdescription pd,
    production.productmodelproductdescriptionculture pmpdc, production.culture c
where pm.productmodelid = pmpdc.productmodelid
and c.cultureid = pmpdc.cultureid
and pmpdc.productdescriptionid = pd.productdescriptionid
and c.name like '%arabic%';
```

```
-- 13.Find first 20 employees who joined very early in the company
select top 20
   p.firstname, p.lastname, e.businessentityid, e.hiredate
from humanresources.employee e,  person.person p
where e.businessentityid = p.businessentityid
order by e.hiredate asc;

-- 14.     Find most trending product based on sales and purchase.
select top 1 p.name
from production.product p,
    sales.salesorderdetail sod, purchasing.purchaseorderdetail pod
where p.productid = sod.productid
and p.productid = pod.productid
group by p.name
order by count(sod.salesorderid) + count(pod.purchaseorderid) desc;

-- 15.      display EMP name, territory name, saleslastyear salesquota and bonus

select *  from sales.salesperson
select * from person.person
select *from sales.salesterritory



select
   (select firstname + ' ' + lastname from person.person where businessentityid = sp.businessentityid)
as emp_name,
   (select name from sales.salesterritory where territoryid = sp.territoryid) as territory_name,
   saleslastyear,
   salesquota,
   bonus
from sales.salesperson sp;
```

```
-- 16.      display EMP name, territory name,
--                        saleslastyear salesquota and bonus from Germany and United Kingdom
select * from sales.salesperson


select
   (select firstname + ' ' + lastname from person.person where businessentityid = sp.businessentityid)
as emp_name,
   (select name from sales.salesterritory where territoryid = sp.territoryid) as territory_name,
   saleslastyear,
   salesquota,
   bonus
from sales.salesperson sp
where sp.territoryid in (
   select territoryid from sales.salesterritory where countryregioncode in ('DE', 'GB')
);



--17.      Find all employees who worked in all North America territory
--find all employees who worked in all North America territory



Select(select CONCAT_ws(' ',firstname,lastname) from Person.Person p
             where p.BusinessEntityID=ss.BusinessEntityID) empname,
              (select  [Group] from Sales.SalesTerritory st
              where st.TerritoryID=ss.TerritoryID) grp,
              (select Name from Sales.SalesTerritory st
              where st.TerritoryID=ss.TerritoryID) cname,
              (select SalesLastYear from Sales.SalesTerritory st
              where st.TerritoryID=ss.TerritoryID) slast,
            (select SalesQuota from Sales.SalesTerritory st
            where st.TerritoryID=ss.TerritoryID) squota,
            (select Bonus from Sales.SalesTerritory st
            where st.TerritoryID=ss.TerritoryID) bonus
from Sales.SalesPerson ss
where ss.TerritoryID IN
(select TerritoryID from Sales.SalesTerritory where [Group] = 'North America');

--18.    find all products in the cart

select * from sales.shoppingcartitem
select *from production.product

select * from production.product
where productid in
(select productid
from sales.shoppingcartitem);
-------------------------------------
select name as product_name
from production.product
where productid in (select productid from sales.shoppingcartitem);



--19.    find all the products with special offer
select * from sales.specialofferproduct
select * from production.product

select ProductNumber,name as product_name
from production.product pp
where productid in (select productid from sales.specialofferproduct sof);
```

```
--20.find all employees name , job title,
--  card details whose credit card expired in the month 11 and year as 2008

select * from person.person;
select * from sales.creditcard
select * from humanresources.employee

select
   (select firstname + ' ' + lastname from person.person
    where businessentityid = (select businessentityid
                                                                       from sales.personcreditcard
                                                                       where creditcardid = c.creditcardid)) as
emp_name,

   (select jobtitle
    from humanresources.employee
           where businessentityid = (select businessentityid
    from sales.personcreditcard
    where creditcardid = c.creditcardid)) as job_title,
cardnumber
from sales.creditcard c
where expyear = 2008 and expmonth = 11;



-- 21.     Find the employee whose payment might be revised  (Hint : Employee payment history)

select
   (select firstname + ' ' + lastname from person.person where businessentityid =
eph.businessentityid) as emp_name
from humanresources.employeepayhistory eph
where ratechangeDate = (
   select max(ratechangedate)
   from humanresources.employeepayhistory eph2
   where eph2.businessentityid = eph.businessentityid
);

--22.      Find total standard cost for the active Product. (Product cost history)

select sum(standardcost) as total_standard_cost
from production.productcosthistory
where productid in (
   select productid from production.product where sellenddate is null
);



use AdventureWorks2022;

select * from person.person
select * from person.AddressType

-- 23.     Find the personal details with address and address type
--(hint: Business Entiry Address , Address, Address type)
SELECT
   p.firstname + ' ' + p.lastname  emp_name,
   a.addressline1 + ' ' + ISNULL(a.addressline2, '') + ', ' + a.city AS full_address,
   at.name AS address_type
FROM Person.Person p
JOIN Person.businessentityaddress ba ON p.BusinessEntityID = ba.          BusinessEntityID
JOIN Person.Address a ON ba.AddressID = a.AddressID
JOIN Person.AddressType at ON ba.AddressTypeID = at.AddressTypeID;

-- 24.      Find the name of employees working in group of North America territory
```

```
select * from Sales.SalesPerson

select * from Person.Person

select * from Sales.SalesTerritory


select

P.FirstName + ' ' + P.LastName Emp_Name

from Sales.SalesPerson SP

join Person.Person P on SP.BusinessEntityID = P.BusinessEntityID

join Sales.SalesTerritory ST on SP.TerritoryID = ST.TerritoryID

where ST.[Group] = 'North America' and ST.TerritoryID is not null;


-- 25.     Find the employee whose payment is revised for more than once


select E.BusinessEntityID, P.FirstName + ' ' + P.LastName as Emp_Name,

count(*) as Revision_Count

from HumanResources.EmployeePayHistory EPH

join HumanResources.Employee E on EPH.BusinessEntityID = E.BusinessEntityID

join Person.Person P on E.BusinessEntityID = P.BusinessEntityID

group by E.BusinessEntityID, P.FirstName, P.LastName

having count(*) > 1;



-- 26.display the personal details of  employee whose payment is revised for more than once.



select * from HumanResources.Employee

select * from Person.Person

select * from HumanResources.EmployeePayHistory


select p.BusinessEntityID, p.FirstName + ' ' + p.LastName as emp_name, p.PersonType,
p.EmailPromotion
```

```
from HumanResources.EmployeePayHistory eph
join HumanResources.Employee e on eph.BusinessEntityID = e.BusinessEntityID
join Person.Person p on e.BusinessEntityID = p.BusinessEntityID
group by p.BusinessEntityID, p.FirstName, p.LastName, p.PersonType, p.EmailPromotion
having count(*) > 1;



-- 27.     Which shelf is having maximum quantity (product inventory)
select top 10 Shelf, sum(Quantity) as TotalQuantity
from Production.ProductInventory
group by Shelf
order by TotalQuantity desc;



--28.     Which shelf is using maximum bin(product inventory)
select top 1 Shelf, count(distinct Bin) as TotalBins
from Production.ProductInventory
group by Shelf
order by TotalBins desc;

--29.     Which location is having minimum bin (product inventory)
select * from Production.ProductInventory

select top 10 LocationID,
    count(distinct Bin) as TotalBins,
    sum(Quantity) as TotalQuantity
from Production.ProductInventory
group by LocationID
order by TotalBins asc;

-- 30.    Find out the product available in most of the locations (product inventory)
```

```
select top 1
   (select Name from Production.Product where Product.ProductID = pi.ProductID) as ProductName,
   pi.ProductID,
   count(distinct pi.LocationID) as TotalLocations
from Production.ProductInventory pi
group by pi.ProductID
order by TotalLocations desc;

--31.     Which sales order is having most order qualtity.

select * from Sales.SalesOrderDetail;

select top 1 SalesOrderID,
sum(OrderQty) as TotalOrderQuantity
from Sales.SalesOrderDetail
group by SalesOrderID
order by TotalOrderQuantity desc;


/* 32.    find the duration of payment revision on every interval
(inline view) Output must be as given format
## revised time – count of revised salries
## duration – last duration of revision e.g
there are two revision date 01-01-2022 and revised in 01-01-2024
so duration here is 2years */



select
```

```
(select FirstName from Person.Person p where p.BusinessEntityID = e.BusinessEntityID) as
First_Name,

(select LastName from Person.Person p where p.BusinessEntityID = e.BusinessEntityID) as
Last_Name,

count(e.RateChangeDate) as Revised_Times,

datediff(year, min(e.RateChangeDate), max(e.RateChangeDate)) as Duration_Years

from

(select BusinessEntityID, RateChangeDate from HumanResources.EmployeePayHistory) e

group by e.BusinessEntityID

order by Revised_Times desc;




-- 33.check if any employee from jobcandidate table is having any payment revisions

select

p.FirstName + ' ' + p.LastName as EmployeeName,

jc.BusinessEntityID,

count(eph.RateChangeDate) as RevisionCount from HumanResources.JobCandidate jc

join HumanResources.EmployeePayHistory eph on jc.BusinessEntityID = eph.BusinessEntityID

join Person.Person p on jc.BusinessEntityID = p.BusinessEntityID

group by p.FirstName, p.LastName, jc.BusinessEntityID

having count(eph.RateChangeDate) > 0;


-- 34.     check the department having more salary revision

select

(select name from HumanResources.Department

where departmentid = (select departmentid

    from HumanResources.EmployeeDepartmentHistory

    where businessentityid = eph.businessentityid)) as department_name,

count(*) as revision_count

from HumanResources.EmployeePayHistory eph

where businessentityid in
```

```
   (select businessentityid from HumanResources.EmployeeDepartmentHistory)
group by businessentityid
order by revision_count desc;



-- 35.	check the employee whose payment is not yet revised
select
(select firstname + ' ' + lastname
from Person.Person p
where p.BusinessEntityID = eph.BusinessEntityID) as employee_name
from HumanResources.EmployeePayHistory eph
group by eph.BusinessEntityID
having count(*) = 1;

--36.	find the job title having more revised payments
select
(select jobtitle  from HumanResources.Employee e
where e.BusinessEntityID = eph.BusinessEntityID) as job_title,
count(*) as revision_count
from HumanResources.EmployeePayHistory eph
group by eph.BusinessEntityID
order by revision_count desc;

--37.	find the employee whose payment is revised in shortest duration (inline view)
select
emp.FirstName + ' ' + emp.LastName as Employee_Name,
min(datediff(day, eph1.RateChangeDate, eph2.RateChangeDate)) as Shortest_Duration
from HumanResources.EmployeePayHistory eph1, HumanResources.EmployeePayHistory eph2,
Person.Person emp
where eph1.BusinessEntityID = eph2.BusinessEntityID
and eph1.BusinessEntityID = emp.BusinessEntityID
and eph1.RateChangeDate < eph2.RateChangeDate
group by emp.FirstName, emp.LastName
order by Shortest_Duration asc;

-- 38.     find the colour wise count of the product (tbl: product)

select Color, count(*)  Prdct_Cnt  from Production.Product
where Color is not null
group by Color
order by Prdct_Cnt desc;



--39.     find out the product who are not in position to sell
--(hint: check the sell start and end date

select * from Production.Product



select Name, ProductNumber, SellStartDate, SellEndDate
from Production.Product
where SellEndDate is not null
and SellEndDate < getdate();



-- 40.      find the class wise, style wise average standard cost

select * from Production.Product



select Class, Style, avg(StandardCost) as AvgStandardCost
from Production.Product
```

```
where Class is not null and Style is not null
group by Class, Style;

--41.      check colour wise standard cost

select * from Production.Product

select Color, avg(StandardCost) as AvgStandardCost
from Production.Product
where Color is not null
group by Color;



-- 42.      find the product line wise standard cost

select ProductLine, avg(StandardCost) as AvgStandardCost
from Production.Product
where ProductLine is not null
group by ProductLine;

-- 43. Find the state wise tax rate (hint: Sales.SalesTaxRate, Person.StateProvince)
use AdventureWorks2022

select * from Sales.SalesTaxRate
select sp.Name as state_name, str.TaxRate
from Sales.SalesTaxRate str, Person.StateProvince sp
where str.StateProvinceID = sp.StateProvinceID;



--44.      Find the department wise count of employees
select d.Name as department_name, count(e.BusinessEntityID)  employee_count
```

```
from HumanResources.Employee e, HumanResources.EmployeeDepartmentHistory edh,
HumanResources.Department d

where e.BusinessEntityID = edh.BusinessEntityID

and edh.DepartmentID = d.DepartmentID

group by d.Name;


-- 45.    Find the department which is having more employees

select * from HumanResources.Employee

select * from HumanResources.EmployeeDepartmentHistory

select * from HumanResources.Department


select top 1 d.Name as department_name, count(e.BusinessEntityID) as employee_count

from HumanResources.Employee e, HumanResources.EmployeeDepartmentHistory edh,

HumanResources.Department d

where e.BusinessEntityID = edh.BusinessEntityID

and edh.DepartmentID = d.DepartmentID

group by d.Name

order by employee_count desc;


--46.    Find the job title having more employees


select top 1 e.JobTitle, count(e.BusinessEntityID) as employee_count

from HumanResources.Employee e

group by e.JobTitle

order by employee_count desc;



-- 47.    Check if there is mass hiring of employees on single day

select * from HumanResources.Employee

select HireDate, count(BusinessEntityID) as employee_count

from HumanResources.Employee
```

```
group by HireDate
having count(BusinessEntityID) > 1
order by employee_count desc;

-- 48.    Which product is purchased more? (purchase order details)


select top 1 ProductID, sum(OrderQty) as total_purchased
from Purchasing.PurchaseOrderDetail
group by ProductID
order by total_purchased desc;

--49.    Find the territory wise customers count   (hint: customer)
select * from Sales.Customer;
select TerritoryID, count(CustomerID) as customer_count
from Sales.Customer
group by TerritoryID
order by customer_count desc;



--50.    Which territory is having more customers (hint: customer)
select top 1 TerritoryID, count(CustomerID) as customer_count
from Sales.Customer
group by TerritoryID
order by customer_count desc;



-- 51.    Which territory is having more stores (hint: customer)
use AdventureWorks2022;
select * from Sales.Customer
select top 1 TerritoryID, count(*) as StoreCount
from Sales.Customer
```

```
where StoreID is not null
group by TerritoryID
order by StoreCount desc;

-- C:\Users\sanke\Documents\SQL Server Management Studio

--52.     Is there any person having more than one credit card (hint: PersonCreditCard)
select * from Sales.PersonCreditCard

select BusinessEntityID, count(*) as CreditCardCount
from Sales.PersonCreditCard
group by BusinessEntityID
having count(*) > 1;

--53.    Find the product wise sale price (sales order details)

select p.Name as ProductName, sod.ProductID, avg(sod.UnitPrice) as AverageSalePrice
from Sales.SalesOrderDetail sod
join Production.Product p on sod.ProductID = p.ProductID
group by p.Name, sod.ProductID
order by AverageSalePrice desc;

--54.    Find the total values for line total product having maximum order
select * from Sales.SalesOrderDetail

select top 1 p.Name as ProductName, sod.ProductID, sum(sod.LineTotal) as total_line_value
from Sales.SalesOrderDetail sod
join Production.Product p on sod.ProductID = p.ProductID
group by p.Name, sod.ProductID
order by sum(sod.OrderQty) desc;
```

```
-- 55.    Calculate the age of employees

select
   p.FirstName + ' ' + p.LastName as EmployeeName,
   e.BirthDate,
   datediff(year, e.BirthDate, getdate()) as Age
from HumanResources.Employee e
join Person.Person p on e.BusinessEntityID = p.BusinessEntityID;

--56.    Calculate the year of experience of the employee based on hire date
select
p.FirstName + ' ' + p.LastName as EmployeeName,
e.HireDate,
datediff(year, e.HireDate, getdate()) as ExperienceYears
from HumanResources.Employee e
join Person.Person p on e.BusinessEntityID = p.BusinessEntityID
order by ExperienceYears desc;

--57.    Find the age of employee at the time of joining
select * from HumanResources.Employee
select
   p.FirstName + ' ' + p.LastName as EmployeeName,
   e.BirthDate,
   e.HireDate,
   datediff(year, e.BirthDate, e.HireDate) as age_at_joining
from HumanResources.Employee e
join Person.Person p on e.BusinessEntityID = p.BusinessEntityID;

--58.    Find the average age of male and female

select gender, avg(datediff(year, birthdate, getdate())) as avg_age
```

```
from HumanResources.Employee
group by gender;

-- 59.     Which product is the oldest product as on the date
--(refer  the product sell start date)
select * from Production.Product

select top 1 Name, SellStartDate
from Production.Product
order by SellStartDate asc;

--60.      Display the product name, standard cost,
--and time duration for the same cost. (Product cost history)

select * from Production.ProductCostHistory

select
   p.Name as ProductName,
   pch.StandardCost,
   datediff(day, pch.StartDate, pch.EndDate)  DurationInDays
from Production.ProductCostHistory pch
join Production.Product p
   on pch.ProductID = p.ProductID
where pch.EndDate is not null
order by DurationInDays desc;

-- 61.     Find the purchase id where shipment is done 1 month later of order date
use AdventureWorks2022;
select PurchaseOrderID
from Purchasing.PurchaseOrderHeader
where datediff(month, OrderDate, ShipDate) = 1;
```

```sql
-- 62.     Find the sum of total due where shipment is done 1 month later of order date
--( purchase order header)
select sum(TotalDue) as total_due_sum
from Purchasing.PurchaseOrderHeader
where datediff(month, OrderDate, ShipDate) = 1;

-- 63.Find the average difference in due date and ship date based on  online order flag


select
   OnlineOrderFlag,
   avg(datediff(day, DueDate, ShipDate)) as AvgDaysDifference
from Sales.SalesOrderHeader
group by OnlineOrderFlag;

--64.      Display business entity id, marital status, gender, vacationhr,
-- average vacation based on marital status


select
   BusinessEntityID, MaritalStatus,Gender,VacationHours,
   (select avg(VacationHours) from HumanResources.Employee e2
   where e2.MaritalStatus = e1.MaritalStatus) as AvgVacationHours
from HumanResources.Employee e1;



--65.      Display business entity id, marital status, gender, vacationhr,
-- average vacation based on gender


select
BusinessEntityID, MaritalStatus, Gender, VacationHours,
(select avg(VacationHours) from HumanResources.Employee e2
where e2.Gender = e1.Gender) as AvgVacationHours

from HumanResources.Employee e1;



-- 66.    Display business entity id, marital status, gender, vacationhr,

-- average vacation based on organizational level


select

BusinessEntityID, MaritalStatus, Gender, VacationHours, OrganizationLevel,

(select avg(VacationHours) from HumanResources.Employee e2

where e2.OrganizationLevel = e1.OrganizationLevel) as AvgVacationHours

from HumanResources.Employee e1;


-- 67.    Display entity id, hire date,

-- department name and department wise count of employee and

--count based on organizational level in each dept


select

e.BusinessEntityID, e.HireDate, d.Name as DepartmentName,

(select count(*) from HumanResources.EmployeeDepartmentHistory edh

where edh.DepartmentID = d.DepartmentID) as DeptEmployeeCount,

(select count(*) from HumanResources.EmployeeDepartmentHistory edh2

join HumanResources.Employee e2 on edh2.BusinessEntityID = e2.BusinessEntityID

where edh2.DepartmentID = d.DepartmentID

and e2.OrganizationLevel = e.OrganizationLevel) as OrgLevelEmployeeCount

from HumanResources.Employee e

join HumanResources.EmployeeDepartmentHistory edh on e.BusinessEntityID =
edh.BusinessEntityID

join HumanResources.Department d on edh.DepartmentID = d.DepartmentID;


--68.     Display department name, average sick leave and sick leave per department
```

```
select
   d.Name as DepartmentName,
   avg(e.SickLeaveHours) as AvgSickLeave,
   sum(e.SickLeaveHours) as TotalSickLeave
from HumanResources.Employee e
join HumanResources.EmployeeDepartmentHistory edh on e.BusinessEntityID =
edh.BusinessEntityID
join HumanResources.Department d on edh.DepartmentID = d.DepartmentID
group by d.Name;


--69.	Display the employee details first name, last name,  with total count of various shift
-- done by the person and shifts count per department

select
   p.FirstName,
   p.LastName,
   count(eh.ShiftID) as Total_Shifts,
   d.Name as Dp_Name,
   count(eh.ShiftID) as Shift_Count_Per_Dept
from HumanResources.EmployeeDepartmentHistory eh
join Person.Person p on eh.BusinessEntityID = p.BusinessEntityID
join HumanResources.Department d on eh.DepartmentID = d.DepartmentID
group by p.FirstName, p.LastName, d.Name;


--70.Display country region code, group average sales quota based on territory id
select * from Sales.SpecialOffer
select * from Sales.SalesTerritory
select * from Person.StateProvince
select * from Person.CountryRegion

select st.CountryRegionCode,[group],
avg(sp.SalesQuota)over(partition by sp.territoryid) avg_sale_quota
from Sales.SalesTerritory st, Sales.SalesPerson sp
where sp.TerritoryID = st.TerritoryID



---71.    Display special offer description, category and avg(discount pct) per the category

select SpecialOfferID,Description,Category,
avg(DiscountPct)over(partition by category)avg_discount
from Sales.SpecialOffer

--72.     Display special offer description, category and avg(discount pct) per the month

select Description,Category,
avg(DiscountPct)over(partition by month(Enddate) ) avg_per_month
from Sales.SpecialOffer

--73.     Display special offer description, category and avg(discount pct) per the year

select Description,Category,avg(DiscountPct)over(partition by year(Enddate) ) avg_disc_year
from Sales.SpecialOffer
order by Category

---74.  display special offer desc,category and avg disc pct as per the type
select Description,Category,avg(DiscountPct)over(partition by [Type] ) discount_on_typ
from Sales.SpecialOffer
select * from Sales.SpecialOffer
```

```
--75.	Using rank and dense rand find territory wise top sales person
use AdventureWorks2022;
select
St.TerritoryID,Sp.BusinessEntityID,Sp.SalesYTD,
rank() over (partition by St.TerritoryID order by Sp.SalesYTD desc) as rank,
dense_rank() over (partition by St.TerritoryID order by Sp.SalesYTD desc) as denserank
from Sales.SalesPerson Sp
join Sales.SalesTerritory St on Sp.TerritoryID = St.TerritoryID
order by St.TerritoryID, rank;
```