```sql

select* from [HumanResources].[Employee]

---retreive all employee in organizational level 1

select* from [HumanResources].[Employee]
where OrganizationLevel=1

---retreive all male employee in the production technical department

select *from [HumanResources].[Employee]
where Gender='M' and JobTitle like 'production technician%'

---Retreive all Jobs Hired in 2008

select year(HireDate)as Year_of_Employment,JobTitle
from [HumanResources].[Employee]
where year(HireDate)=2008

---Retreive all Production Technician hired in 2008

select*
from [HumanResources].[Employee]
where JobTitle like 'production technician%' and year(HireDate)=2008

---how many production Technician was hired in each year

select count(JobTitle) as Total_Job_Employed,year(HireDate) as
  Year_Employed
from [HumanResources].[Employee]
where JobTitle like 'production technician%'
group by year(HireDate)

---reveal the jobs hired in 2008
select* from [HumanResources].[Employee]
where year(HireDate)=2008

----MULTIPLE CASE AND WHEN STATEMENT FOR SALES ORDER DETAIL USING UNIT
  PRICE TO DETERMINE THE DIFFERENCE PRICE RANGE
select*from[Sales].[SalesOrderDetail]

select UnitPrice,
case
when UnitPrice between 1000 and 2000 then 'Tax1'
when UnitPrice>2000 then 'Tax2'
when UnitPrice<1000 then 'No tax'
else 'Error'
end as sales_difference
from[Sales].[SalesOrderDetail]
```

```sql
order by UnitPrice desc

select*from[Purchasing].[PurchaseOrderDetail]
select*From[Purchasing].[PurchaseOrderHeader]

---Joining two tables using inner join,left join and full outter join

select*from[Purchasing].[PurchaseOrderDetail]pd
inner join [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID

select*from[Purchasing].[PurchaseOrderDetail]pd
left join [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID

select*from[Purchasing].[PurchaseOrderDetail]pd
right join [Purchasing].[PurchaseOrderHeader]ph on
  ph.PurchaseOrderID=pd.PurchaseOrderID

select*from[Purchasing].[PurchaseOrderDetail]pd
full outer join [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID

----MULTIPLE CASE AND WHEN STATEMENT FOR SALES ORDER DETAIL USING UNIT
  PRICE TO DETERMINE THE DIFFERENCE PRICE RANGE

select UnitPrice,ProductID,
case
when UnitPrice between 30 and 50 then 'Increase_Price_by_30%'
when UnitPrice>50 then 'Increase_Price_by_10%'
when UnitPrice<30 then 'Maintain_Current_Price'
else 'error'
end as Price_Evalution
from[Purchasing].[PurchaseOrderDetail]pd
join [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID
order by UnitPrice desc

select*from[Purchasing].[PurchaseOrderDetail]
select*from [Purchasing].[PurchaseOrderHeader]
select*from [Production].[Product]

---WHICH YEAR HAS THE HIGHEST PURCHASE?

select year(OrderDate) as Year_Of_Order,sum(OrderQty) as
  Total_Quantity_Ordered
from[Purchasing].[PurchaseOrderDetail]pd
JOIN [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID
```

```
JOIN[Production].[Product]pp on pp.ProductID=pd.ProductID
group by  year(OrderDate)
order by Total_Quantity_Ordered desc

---CALCULATE THE TOTAL PURCHASE OF ALL PRODUCTS IN YEAR 2011

select Name,sum(OrderQty) as Total_Quantity_Ordered,year(OrderDate) as
  Year_Of_Order
from[Purchasing].[PurchaseOrderDetail]pd
join[Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID
join[Production].[Product]pp on pd.ProductID=pp.ProductID
where year(OrderDate)=2011
group by Name,year(OrderDate)
order by Total_Quantity_Ordered desc


---RETRIEVE ALL REJETED QUANTITY AND ORDER QUANTITY OF ALL PRODUCT NAME

select Name,sum(RejectedQty) as total_rejected_quantity,sum(OrderQty) as
  Total_Order_Quantity
from[Purchasing].[PurchaseOrderDetail]pd
JOIN [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID
JOIN[Production].[Product]pp on pp.ProductID=pd.ProductID
group by Name

---SELECT THE TOP 5 PRODUCT WITH THE HIGEST RECEIVED QUANTITY

select top 5 Name,sum(ReceivedQty) as Total_Recieved_Quantity
from[Purchasing].[PurchaseOrderDetail]pd
JOIN [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID
JOIN[Production].[Product]pp on pp.ProductID=pd.ProductID
group by Name
order by Total_Recieved_Quantity desc


---CALCULATE THE DIFFERENCE BETWEENTHE ORDERDATE AND SHIPDATE
---ALONG WITH THE TOTAL FRIEGHT FOR ALL YEARS AND MONTH
select datediff(day,ph.OrderDate,ph.ShipDate) as Day_Difference,
                sum(Freight) as Total_Frieght,
                datepart(year,OrderDate) as Year_Of_Order,
                datepart(month,OrderDate) as Month_Of_Order
from[Purchasing].[PurchaseOrderDetail]pd
JOIN [Purchasing].[PurchaseOrderHeader]ph on
  pd.PurchaseOrderID=ph.PurchaseOrderID
JOIN[Production].[Product]pp on pp.ProductID=pd.ProductID
group by datepart(year,OrderDate),
```

```
132          datepart(month,OrderDate),
133          datediff(day,ph.OrderDate,ph.ShipDate)
134  order by Year_Of_Order
135
136
137  ---CALCULATE THE DIFFERENCE BETWEENTHE ORDERDATE AND SHIPDATE
138  ---ALONG WITH THE TOTAL FRIEGHT FOR ALL YEARS AND MONTH
139  -- sum of freight > 22588.1815
140
141  select datediff(day,ph.OrderDate,ph.ShipDate) as Day_Difference,
142                 sum(Freight) as Total_Frieght,
143                 datepart(year,OrderDate) as Year_Of_Order,
144                 datepart(month,OrderDate) as Month_Of_Order
145  from[Purchasing].[PurchaseOrderDetail]pd
146  JOIN [Purchasing].[PurchaseOrderHeader]ph on                              ↵
       pd.PurchaseOrderID=ph.PurchaseOrderID
147  JOIN[Production].[Product]pp on pp.ProductID=pd.ProductID
148  group by datepart(year,OrderDate),
149          datepart(month,OrderDate),
150          datediff(day,ph.OrderDate,ph.ShipDate)
151  having sum(Freight)>22588.1815
152  order by Year_Of_Order
153
154
155
```