```
use NORTHWND select * from dbo.Shippers; select * from dbo.email; insert into dbo.email(id,email) values(3,'ali@gmail.com');
---------------------------------------------------------- /* show duplicate email*/ select email from email group by email having(count(*))>1; Alter table email
CHANGE column email.email to email_address VARCHAR(255); /*remove duplicate*/ delete from email WHERE email.id not in ( select min_id
from( select min(email.id) as min_id,email from email group by email ) as query ); select * from email; truncate table email
-----------------------2----------------------------------- select Categories.CategoryName,Categories.Description from dbo.Categories; select * from
dbo.Employees; ----------------------3---------------------------------- /* 3. Sales Representatives We'd like to see just the FirstName, LastName, and
HireDate of all the employees with the Title of Sales Representative. Write a SQL statement that returns only those employees. */ select
Employees.FirstName,Employees.LastName,Employees.HireDate from dbo.Employees where Employees.Title = 'Sales Representative';
----------------------4------------------------- /*Now we'd like to see the same columns as above, but only for those employees that both have the title of
Sales Representative, and also are in the United States.*/ Select Employees.FirstName, Employees.LastName, Employees.HireDate from
Employees where Employees.Title='Sales Representative' AND Employees.Country = 'USA'; /*Show all the orders placed by a specific employee.
The EmployeeID for this Employee (Steven Buchanan) is 5.*/ select * from Employees; select * from Orders; select * from [Order Details]; select
emp.FirstName as employee_Fname,emp.LastName as employee_Lname, O.EmployeeID,O.OrderDate,OD.* from Orders as O inner join
Employees as emp ON O.EmployeeID=emp.EmployeeID inner join [Order Details] as OD ON OD.OrderID=O.OrderID where O.EmployeeID=5;
----------------------------------------- /* total unit price sale by employee steven*/ select emp.FirstName,sum(OD.Quantity) as total_quantity from
Orders as O inner join Employees as emp ON O.EmployeeID=emp.EmployeeID inner join [Order Details] as OD ON OD.OrderID=O.OrderID where
emp.FirstName='steven' group by emp.FirstName; /* max unit price sale by employee steven*/ select MAX(total_quantity) as sum_of_max
from( select emp.FirstName,sum(OD.Quantity) as total_quantity from Orders as O inner join Employees as emp ON
O.EmployeeID=emp.EmployeeID inner join [Order Details] as OD ON OD.OrderID=O.OrderID group by emp.FirstName ) as maxx
------------------------------------------ /* In the Suppliers table, show the SupplierID, ContactName, and ContactTitle for those Suppliers whose
ContactTitle is not Marketing Manager. */ select * from Suppliers; select S.SupplierID,S.ContactName,S.ContactTitle from Suppliers as S where
S.ContactTitle != 'Marketing Manager'; /* In the products table, we'd like to see the ProductID and ProductName for those products where the
ProductName includes the string "queso". */ select P.ProductID,P.ProductName from Products as P where P.ProductName like '%queso%'; /*
Looking at the Orders table, there's a field called ShipCountry. Write a query that shows the OrderID, CustomerID, and ShipCountry for the orders
where the ShipCountry is either France or Belgium. */ select O.OrderID,O.CustomerID,O.ShipCountry from Orders as O where
O.ShipCountry='France' or O.ShipCountry= 'Belgium'; /* Now, instead of just wanting to return all the orders from France of Belgium, we want to
show all the orders from any Latin American country. But we don't have a list of Latin American countries in a table in the Northwind database.
So, we're going to just use this list of Latin American countries that happen to be in the Orders table: Brazil Mexico Argentina Venezuela It
doesn't make sense to use multiple Or statements anymore, it would get too convoluted. Use the In statement. */ select *from Orders as O
where O.ShipCountry in ('Brazil','Mexico','Argentina','Venezuela') ; /* ----------------------10------------------------- 10 For all the employees in the
Employees table, show the FirstName, LastName, Title, and BirthDate. Order the results by BirthDate, so we have the oldest employees first. */
select * from Employees; select emp.FirstName,emp.LastName,emp.Title,emp.BirthDate from Employees as emp order by emp.BirthDate
ASC; /* age between 1948 to 1960 */ select emp.FirstName,emp.LastName,emp.Title,emp.BirthDate from Employees as emp where
```

```
YEAR(emp.BirthDate) BETWEEN 1948 AND 1960; /* print employee where age is same */ select
emp.FirstName,emp.LastName,emp.Title,emp.BirthDate from Employees as emp where YEAR(emp.BirthDate) = '1948'; /* update date of birth.
*/ update Employees SET BirthDate ='1948' where Employees.EmployeeID = 2; select Employees.EmployeeID from Employees where
Employees.BirthDate='1948'; /* =========================11========================== In the output of the query above, showing
the Employees in order of BirthDate, we see the time of the BirthDate field, which we don’t want. Show only the date portion of the BirthDate
field. */ use NORTHWND; SELECT emp.FirstName,emp.LastName, CONVERT(DATE, emp.BirthDate) AS Date_only FROM Employees AS emp
ORDER BY CONVERT(DATE, emp.BirthDate); /* e FirstName and LastName columns from the Employees table, and then create a new column
called FullName, showing FirstName and LastName joined together in one column, with a space in-between */ select emp.FirstName,
emp.LastName, CONCAT(emp.FirstName,' ',emp.LastName) as Full_Name from Employees as emp; /*-----------------View -------------*/ Create View
Fullname AS select emp.FirstName, emp.LastName, CONCAT(emp.FirstName,' ',emp.LastName) as Full_Name from Employees as emp; select
* from Fullname; /*-----------Start with letter A-----*/ select * from Fullname as FN where FN.Full_Name like 'A%'; select * from Fullname as FN
where FN.Full_Name like '%A'; select * from Fullname as FN where FN.Full_Name like '%Y%'; /* -------------count those employee which name start
with A*/ select COUNT(Fullname.Full_Name) as cfn from Fullname where Fullname.Full_Name like 'A%' ; /*In the OrderDetails table, we have the
fields UnitPrice and Quantity. Create a new field, TotalPrice, that multiplies these two together. We’ll ignore the Discount field for now. In
addition, show the OrderID, ProductID, UnitPrice, and Quantity. Order by OrderID and ProductID.*/ select * from [Order Details]; select *,
(OD.Quantity*OD.UnitPrice) as Total_price from [Order Details] as OD order by OD.OrderID ASC, OD.ProductID ASC; use NORTHWND; select
count(C.CustomerID) as total_customer from Customers as C; /* Show the date of the first order ever made in the Orders table. */ SELECT
min(Orders.OrderDate) AS first_order from Orders; select Top 1 * from Orders order by Orders.OrderDate DESC ; select min(Orders.OrderID) as
minorder ,Orders.* from Orders; /* Show a list of countries where the Northwind company has customers. */ select DISTINCT(C.Country) from
Customers as C ; -- also do with groupby---------------- /* Show a list of all the different values in the Customers table for ContactTitles. Also
include a count for each ContactTitle. This is similar in concept to the previous question “Countries where there are customers” , except we now
want a count for each ContactTitle. */ select C.ContactTitle,count(C.ContactTitle) as total_contact from Customers as C group by
C.ContactTitle; ---------------MAX(Count) not used in sql select MAX(count_cus) as max_count from ( select COUNT(C.CustomerID) as count_cus
from Customers as C group by C.ContactTitle ) As CountTable; ----------------------------------- select top 1 C.ContactTitle,Count(C.ContactTitle) as ct
from Customers as C group by C.ContactTitle order by ct DESc; /* ----------------------------18---------------------------- e’d like to show, for each product,
the associated Supplier. Show the ProductID, ProductName, and the CompanyName of the Supplier. Sort by ProductID. */ SELECT
P.ProductID,P.ProductName,S.CompanyName from Suppliers as S join Products as P ON S.SupplierID=P.SupplierID order by P.ProductID ASc;
----------------groupby company with products--------------- SELECT S.CompanyName,COUNT(P.ProductName) as product_Number from Suppliers
as S join Products as P ON S.SupplierID=P.SupplierID group by S.CompanyName; ------------------------print company who have maximum
product------------- USE NORTHWND; select Top 2 s.CompanyName,count(p.ProductName) as total_product from Suppliers as S join Products as
P ON S.SupplierID=P.SupplierID group by s.CompanyName order by total_product DESC ; -----------------------------------------------------------------------
SELECT MM.CompanyName, MM.max_prod FROM ( SELECT S.CompanyName, COUNT(P.ProductName) AS total_product,
MAX(COUNT(P.ProductName)) over () AS max_prod FROM Suppliers AS S JOIN Products AS P ON S.SupplierID = P.SupplierID GROUP BY
S.CompanyName ) AS MM WHERE MM.total_product = MM.max_prod; /* We’d like to show a list of the Orders that were made, including the
Shipper that was used. Show the OrderID, OrderDate (date only), and CompanyName of the Shipper, and sort by OrderID. */ select
S.ShipperID,o.OrderID,CONVERT(Date,O.OrderDate) as Dates, from Orders as O join Shippers as S ON S.ShipperID=O.ShipVia ORDER BY
O.OrderID ASC; -------------made 3 group with shipper id 1,2,3--------------- --------------view with ID=1------------------- create View group1 as select
S.ShipperID,o.OrderID,CONVERT(Date,O.OrderDate) as Dates,S.CompanyName from Orders as O join Shippers as S ON S.ShipperID=O.ShipVia
where S.ShipperID=1; select * from group1; ----------------------------------------------------------------------- create View group2 as select
S.ShipperID,o.OrderID,CONVERT(Date,O.OrderDate) as Dates,S.CompanyName from Orders as O join Shippers as S ON S.ShipperID=O.ShipVia
where S.ShipperID=2; select * from group2; /********************************************************************************
********************************************************************************* we’d like to see the total number of products in each category.
Sort the results by the total number of products, in descending order */ select * from Products; select P.CategoryID, COUNT(P.ProductName) as
Total_prod from Products as P group by P.CategoryID ; ----------aslo display category Name---------------- select P.CategoryID,C.CategoryName,
COUNT(P.ProductName) as Total_prod from Products as P JOIN Categories as C ON P.CategoryID=C.CategoryID group by
P.CategoryID,C.CategoryName order by Total_prod DESC ; /* In the Customers table, show the total number of customers per Country and City.
*/ select * from Customers; select C.Country,C.City, COUNT(C.CustomerID) as Total_Customer from Customers as C group by C.City,C.Country
order BY Total_Customer DESC; /* What products do we have in our inventory that should be reordered? For now, just use the fields
UnitsInStock and ReorderLevel, where UnitsInStock is less than the ReorderLevel, ignoring the fields UnitsOnOrder and Discontinued. Order the
results by ProductID. */ Select * from Products select P.* from Products as P where P.UnitsInStock < P.ReorderLevel order by ProductID ASC ; /*
Now we need to incorporate these fields—UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued—into our calculation. We’ll define “products
that need reordering” with the following: UnitsInStock plus UnitsOnOrder are less than or equal to ReorderLevel The Discontinued flag is false
(0) */ Select * from Products as P where (P.UnitsInStock + P.UnitsOnOrder)<=P.ReorderLevel AND P.Discontinued != 1; /* A salesperson for
Northwind is going on a business trip to visit customers, and would like to see a list of all customers, sorted by region, alphabetically. However,
he wants the customers with no region (null in the Region field) to be at the end, instead of at the top, where you’d normally find the null values.
Within the same region, companies should be sorted by CustomerID */ select C.CustomerID,C.CompanyName, C.Region , Case when C.Region
IS NULL THEN '1' ELSE 0 END as RegionFlag from Customers as C ; /* Some of the countries we ship to have very high freight charges. We'd
like to investigate some more shipping options for our customers, to be able to offer them lower freight charges. Return the three ship countries
with the highest average freight overall, in descending order by average freight */ USE NORTHWND; select * from Suppliers; select Top 3
O.ShipCountry, AVG(O.Freight) as avg_heigh from Orders as O group by O.ShipCountry order by avg_heigh DESC ; ------------------where
orderdate=2015 select Top 3 O.ShipCountry, avg(O.Freight) as avg_heigh from Orders as O where Year(O.OrderDate)=2015 group by
O.ShipCountry order by avg_heigh DESC; /* We're continuing to work on high freight charges. We now want to get the three ship countries with
the highest average freight charges. But instead of filtering for a particular year, we want to use the last 12 months of order data, using as the
end date the last OrderDate in Orders. */ select * from Orders select * from [Order Details] select TOP 3 O.ShipCountry,avg(O.Freight) as
AVG_FREI from Orders as O group by O.ShipCountry Having O.OrderDate= Dateadd(yy,-1,Max(O.OrderDate)) order by AVG_FREI DESC Select
Dateadd(yy, -1, (Select Max(OrderDate) from Orders)) -------------Select Customers who dont order select * from Customers select O.CustomerID
```

```
from Orders as O where O.CustomerID is NULL /* ----------------------------------------------------
============================================================ ----------------------------------------------------- Advance problems */ use
NORTHWND select * from orders; select * from [Order Details] select * from Customers select * from Products select
O.OrderID,C.ContactName,Total_amount= SUM(OD.Quantity*OD.UnitPrice) from Orders as O join Customers as C ON
C.CustomerID=O.CustomerID JOIN [Order Details] as OD ON O.OrderID=OD.OrderID where Year(O.OrderDate)=1905 Group by
O.OrderID,C.ContactName Having Sum(OD.Quantity*OD.UnitPrice)>2000; ----------------------------------------------------- Select
EmployeeID ,OrderID ,OrderDate From Orders Where OrderDate = EOMONTH(OrderDate ) Order by EmployeeID ,OrderID ------------Window
Function--------------------- /* Select window function(SUM,AVG,Rank,Dense Rank)() over (pertion by..... order by....... ) as window name from table
*/ ----------------------Rank Unit price low to heigh---------------------- select * from [Order Details]; select dense_rank() over(order by OD.UnitPrice ASC)
rank from [Order Details] as OD Select Row_Number() over(Order by O.OrderDate) AS RowNumber, O.OrderID,O.CustomerID ,O.OrderDate from
Orders as O -----Rank------- Select Rank() over(PARTITION BY O.CustomerID Order by O.OrderDate) AS Ranktotalammount,
O.OrderID,O.CustomerID ,O.OrderDate from Orders as O Use NORTHWND create PROCEDURE rank_fun as Select avg(O.Freight) over(PARTITION
BY O.CustomerID Order by O.OrderDate) AS avg_Freight, O.OrderID,O.CustomerID ,O.OrderDate from Orders as O ; exec rank_fun;
------------------------------------------------------------ SELECT CustomerID OrderDate, SUM(Orders.Freight) OVER (PARTITION BY CustomerID ORDER
BY OrderDate) AS RunningTotal FROM Orders; ---------------------------------000000000000000000000000000000--- create table
mynumber( numbers int ) insert into mynumber(numbers) values (1),(2),(3),(4),(4); select * from mynumber ----query (select distinct max
number ) select top 1 mn.numbers from mynumber as mn group by mn.numbers having count(*)=1 order by mn.numbers DESC ;
-------------------------------------------------------------- ---------------------------------------------------------------- create table email( id int, email varchar(50) )
insert into email(id,email) values (1,'ali@gmail.com'),(3,'ali@gmail.com'); select * from email -- i found where duplicate select email from email
group by email having count(*)!=1; -------------delete both duplicates----------------- DELETE FROM email WHERE email IN ( SELECT email FROM
email GROUP BY email HAVING COUNT(*) > 1 ); -- problem: write without duplicate email, delete max id email. DELETE FROM email WHERE id
IN ( SELECT max_id from( select MAX(id) as max_id FROM email GROUP BY email HAVING COUNT(*) > 1 ) as max_id);
```