1. **Practice Queries of the Northwind Database**

1. Create a report that shows the CategoryName and Description from the categories table sorted by CategoryName.

select categoryname, description from Categories;

1. Create a report that shows the ContactName, CompanyName, ContactTitle and Phone number from the customers table sorted by Phone.

select ContactName, CompanyName, ContactTitle, Phone

from Customers

order by phone;

1. Create a report that shows the capitalized FirstName and capitalized LastName renamed as FirstName and Lastname respectively and HireDate from the employees table sorted from the newest to the oldest employee.

select UPPER(firstname) as FirstName, UPPER(lastname) as LastName,

hiredate

from Employees

order by HireDate desc;

1. Create a report that shows the top 10 OrderID, OrderDate, ShippedDate, CustomerID, Freight from the orders table sorted by Freight in descending order.

select top 10 orderid, orderdate, shippeddate, customerid, freight

from orders

order by freight desc;

1. Create a report that shows all the CustomerID in lowercase letter and renamed as ID from the customers table.

select LOWER(customerid) as ID

from customers;

1. Create a report that shows the CompanyName, Fax, Phone, Country, HomePage from the suppliers table sorted by the Country in descending order then by CompanyName in ascending order.

select companyname, fax, phone, country, homepage

from suppliers

order by country desc, companyname;

1. Create a report that shows CompanyName, ContactName of all customers from ‘Buenos Aires' only.

select companyname, contactname

from Customers

where city = 'Buenos Aires';

1. Create a report showing ProductName, UnitPrice, QuantityPerUnit of products that are out of stock.

select productname, unitprice, quantityperunit

from Products

where UnitsInStock = 0;

1. Create a report showing all the ContactName, Address, City of all customers not from Germany, Mexico, Spain.

select contactname, address, city

from customers

where city NOT IN ('Germany','Mexico','Spain');

1. Create a report showing OrderDate, ShippedDate, CustomerID, Freight of all orders placed on 21 May 1996.

select orderdate, shippeddate, customerid, freight

from orders

where orderdate = '1996-05-21 00:00:00.000';

1. Create a report showing FirstName, LastName, Country from the employees not from United States.

select firstname, lastname, country

from employees

where country NOT LIKE 'USA';

1. Create a report that shows the EmployeeID, OrderID, CustomerID, RequiredDate, ShippedDate from all orders shipped later than the required date.

select employeeid, orderid, customerid, requireddate, shippeddate

from Orders

where ShippedDate > RequiredDate;

1. Create a report that shows the City, CompanyName, ContactName of customers from cities starting with A or B.

select city, companyname, contactname

from customers

where city LIKE 'A%' or city LIKE 'B%'

order by city;

1. Create a report showing all the even numbers of OrderID from the orders table.

select orderid

from orders

where orderid % 2 = 0;

1. Create a report that shows all the orders where the freight cost more than $500.

select \* from Orders

where Freight > 500;

1. Create a report that shows the ProductName, UnitsInStock, UnitsOnOrder, ReorderLevel of all products that are up for reorder.

select productname, unitsinstock, reorderlevel

from Products

where reorderlevel = 0;

1. Create a report that shows the CompanyName, ContactName number of all customer that have no fax number.

select companyname, contactname

from customers

where Fax IS NULL;

1. Create a report that shows the FirstName, LastName of all employees that do not report to anybody.

select firstname, lastname

from Employees

where ReportsTo IS NULL;

1. Create a report showing all the odd numbers of OrderID from the orders table.

select orderid

from orders

where orderid % 2 = 1;

1. Create a report that shows the CompanyName, ContactName, Fax of all customers that do not have Fax number and sorted by ContactName.

select companyname, contactname, fax

from Customers

where Fax IS NULL

order by ContactName;

1. Create a report that shows the City, CompanyName, ContactName of customers from cities that has letter L in the name sorted by ContactName.

select city, companyname, contactname

from Customers

where City LIKE '%L%'

order by ContactName;

1. Create a report that shows the FirstName, LastName, BirthDate of employees born in the 1950s.

select firstname, lastname, birthdate

from Employees

where BirthDate >= '1950-01-01' and BirthDate < '1960-01-01';

1. Create a report that shows the FirstName, LastName, the year of Birthdate as birth year from the employees table.

select firstname, lastname, YEAR(birthdate) as BirthYear

from Employees;

1. Create a report showing OrderID, total number of Order ID as NumberofOrders from the orderdetails table grouped by OrderID and sorted by NumberofOrders in descending order. **HINT: you will need to use a Groupby statement.**

select orderid, COUNT(orderid) as NumberofOrders

from [Order Details]

group by OrderID

order by NumberofOrders desc;

1. Create a report that shows the SupplierID, ProductName, CompanyName from all product Supplied by Exotic Liquids, Specialty Biscuits, Ltd., Escargots Nouveaux sorted by the supplier ID

select s.supplierid, p.productname, s.companyname

from Suppliers s

JOIN Products p

ON s.SupplierID = p.SupplierID

where s.CompanyName IN ('Exotic Liquids', 'Specialty Biscuits',

'Ltd.', 'Escargots Nouveaux')

order by s.SupplierID;

1. Create a report that shows the ShipPostalCode, OrderID, OrderDate, RequiredDate, ShippedDate, ShipAddress of all orders with ShipPostalCode beginning with "98124".

select shippostalcode, orderid, orderdate, requireddate, shippeddate,

shipaddress

from orders

where shippostalcode LIKE '98124';

1. Create a report that shows the ContactName, ContactTitle, CompanyName of customers that the has no "Sales" in their ContactTitle.

select contactname, contacttitle, companyname

from Customers

where ContactTitle NOT LIKE '%Sales%';

1. Create a report that shows the LastName, FirstName, City of employees in cities other than "Seattle";

select lastname, firstname, city

from Employees

where city NOT LIKE 'Seattle';

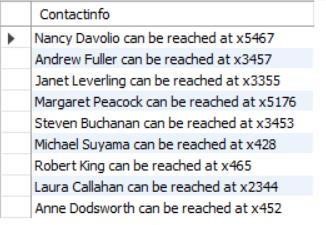
1. Create a report that shows the CompanyName, ContactTitle, City, Country of all customers in any city in Mexico or other cities in Spain other than Madrid.

select companyname, contacttitle, city, country

from Customers

where country IN ('Mexico','Spain') and city != 'Madrid';

Create a select statement that outputs the following:



select CONCAT(firstname,' ', lastname ,' ', 'can be reached at ', 'x',

extension) as ContactInfo

from Employees;

1. Create a report that shows the ContactName of all customers that do not have letter A as the second alphabet in their Contactname.

select contactname

from Customers

where ContactName NOT LIKE '\_A%';

1. Create a report that shows the average UnitPrice rounded to the next whole number, total price of UnitsInStock and maximum number of orders from the products table. All saved as AveragePrice, TotalStock and MaxOrder respectively.

select ROUND(AVG(unitprice), 0) as AvgUnitPrice, SUM(unitsinstock) as

TotalStock, MAX(unitsonorder) as MaxOrder

from Products;

1. Create a report that shows the SupplierID, CompanyName, CategoryName, ProductName and UnitPrice from the products, suppliers and categories table.

select s.supplierid, s.companyname, c.categoryname,

p.productname, p.unitprice

from Products p

JOIN Suppliers s

ON s.SupplierID = p.SupplierID

JOIN Categories c

ON c.CategoryID = p.CategoryID;

1. Create a report that shows the CustomerID, sum of Freight, from the orders table with sum of freight greater $200, grouped by CustomerID. **HINT: you will need to use a Groupby and a Having statement.**

select customerid, SUM(freight) as SFreight

from Orders

group by CustomerID

HAVING SUM(freight) > 200;

1. Create a report that shows the OrderID ContactName, UnitPrice, Quantity, Discount from the order details, orders and customers table with discount given on every purchase.

select od.orderid, c.contactname, od.unitprice,

od.quantity, od.discount

from [Order Details] od

JOIN Orders o

ON o.OrderID = od.OrderID

JOIN Customers c

ON c.CustomerID = o.CustomerID

where od.Discount != 0;

1. Create a report that shows the EmployeeID, the LastName and FirstName as employee, and the LastName and FirstName of who they report to as manager from the employees table sorted by Employee ID. **HINT: This is a SelfJoin.**

select a.EmployeeID, CONCAT(a.LastName,' ', a.FirstName) as employee,

CONCAT(b.LastName, ' ', b.FirstName) as manager

from Employees a

LEFT JOIN Employees b

ON b.EmployeeID = a.ReportsTo

order by a.EmployeeID;

1. Create a report that shows the average, minimum and maximum UnitPrice of all products as AveragePrice, MinimumPrice and MaximumPrice respectively.

select AVG(unitprice) as AvgUnitPrice, MIN(unitprice) as MinPrice,

MAX(unitprice) MaxPrice

from Products;

1. Create a view named CustomerInfo that shows the CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Country, Phone, OrderDate, RequiredDate, ShippedDate from the customers and orders table. **HINT: Create a View.**

CREATE VIEW CustomerInfo AS

select c.CustomerID, c.CompanyName, c.ContactName, c.ContactTitle,

c.Address, c.City, c.Country, c.Phone, o.OrderDate, o.RequiredDate,

o.ShippedDate

from Customers c

JOIN Orders o

ON c.CustomerID = o.CustomerID;

1. Change the name of the view you created from customerinfo to customer details.

EXEC sp\_rename 'dbo.CustomerInfo', 'CustomerDetails';

1. Create a view named ProductDetails that shows the ProductID, CompanyName, ProductName, CategoryName, Description, QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, ReorderLevel, Discontinued from the supplier, products and categories tables. **HINT: Create a View**
2. Drop the customer details view.
3. Create a report that fetch the first 5 character of categoryName from the category tables and renamed as ShortInfo

select SUBSTRING(CategoryName, 1, 5) as Short\_Info

from Categories;

1. Create a copy of the shipper table as shippers\_duplicate. Then insert a copy of shippers data into the new table **HINT: Create a Table, use the LIKE Statement and INSERT INTO statement.**

1. Create a select statement that outputs the following from the shippers\_duplicate Table:



1. Create a report that shows the CompanyName and ProductName from all product in the Seafood category.

select s.companyname, p.ProductName

from Categories c

JOIN Products p

ON c.CategoryID = p.CategoryID

JOIN Suppliers s

ON s.SupplierID = p.SupplierID

where CategoryName = 'Seafood';

1. Create a report that shows the CategoryID, CompanyName and ProductName from all product in the categoryID 5.

select c.categoryid, s.companyname, p.productname

from Categories c

JOIN Products p

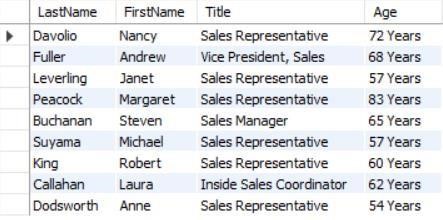
ON c.CategoryID = p.CategoryID

JOIN Suppliers s

ON s.SupplierID = p.SupplierID

where c.CategoryID = 5;

1. Delete the shippers\_duplicate table.
2. Create a select statement that ouputs the following from the employees table. **NB:** The age might differ depending on the year you are attempting this query.



1. Create a report that the CompanyName and total number of orders by customer renamed as number of orders since December 31, 1994. Show number of Orders greater than 10.

select c.companyname, COUNT(c.customerid) as NumberofOrders

from Customers c

JOIN Orders o

ON c.CustomerID = o.CustomerID

where o.OrderDate >= '1994-12-31'

group by c.CompanyName

having COUNT(c.customerid) > 10;

1. Create a select statement that ouputs the following from the product table **NB:** It should return 77rows.



select CONCAT(productname, ' ','weight/is',' ', quantityperunit, ' ',

'and cost ','$', ROUND(UnitPrice, 1)) as ProductInfo

from Products;