```sql
--1. Select FirstName, LastName, and HireDate of all the employees with the
Title of
--Sales Representative. Write a SQL statement that returns only those
employees
SELECT FirstName,LastName,HireDate
FROM Employees
WHERE Title='Sales Representative'
--2. Select 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 FirstName,LastName,HireDate
FROM Employees
WHERE Title='Sales Representative' AND Country='USA'
--3. Show all the orders placed by a specific employee. The EmployeeID for
this
--Employee (Steven Buchanan) is 5.
SELECT * FROM Orders WHERE EmployeeID=5
--4. Show all the orders placed by a specific employee in 03.1997. The
EmployeeID
--for this Employee (Steven Buchanan) is 5.
SELECT * FROM Orders WHERE EmployeeID=5 AND year(OrderDate)=1997 AND
month(OrderDate)=3
--5. Show all the orders placed by a specific employee on Mondays in 1997.
The
--EmployeeID for this Employee (Steven Buchanan) is 5.
SELECT * FROM Orders WHERE EmployeeID=5 AND year(OrderDate)=1997 AND
datename(weekday,Orderdate)='Monday'
--6. In the Suppliers table, show the SupplierID, ContactName, and
ContactTitle for
--those Suppliers whose ContactTitle is not Marketing Manager.
SELECT SupplierID,ContactName,ContactTitle
FROM Suppliers
WHERE NOT ContactTitle='Marketing Manager'
--7. In the products table, we'd like to see the ProductID and ProductName
for those
--products where the ProductName includes the string 'queso'.
SELECT ProductID,ProductName
FROM Products
WHERE ProductName LIKE '%queso%'
--8. 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 OrderID, CustomerID,ShipCountry
FROM Orders
WHERE ShipCountry IN ('France','Belgium')

--9. 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 FirstName,LastName,Title,BirthDate
FROM Employees
ORDER BY BirthDate
--10. 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.
SELECT FirstName,LastName,Title,convert(date,BirthDate)
FROM Employees
```

```sql
ORDER BY BirthDate

--11. Show the 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 FirstName,LastName,FirstName+' '+LastName AS 'FullName'
FROM Employees

--12. In the OrderDetails table, we have the fields UnitPrice and Quantity.
Create a
--new field, TotalPrice, that multiplies these two together. Ignore the
Discount field for now.
SELECT UnitPrice,Quantity,UnitPrice*Quantity AS 'TotalPrice'
FROM [Order Details]
--13. Do the same, but include discount value
SELECT UnitPrice,Quantity,UnitPrice*Quantity*(1-Discount) AS 'TotalPrice'
FROM [Order Details]
--14. How many customers do we have in the Customers table? Show one value
only.
SELECT COUNT(CustomerID)
FROM Customers
--15. Show the date of the first order ever made in the Orders table.
SELECT  TOP 1 convert(date,OrderDate)
FROM Orders
ORDER BY OrderDate
--16. Show a list of countries where the Northwind company has customers.
SELECT DISTINCT Country
FROM Customers
WHERE Country IS NOT NULL

--17. For each product, show the ProductID, ProductName, and the
CompanyName of
--the Supplier. Sort by ProductID
SELECT ProductID,ProductName,CompanyName
FROM Products
LEFT OUTER JOIN Suppliers ON Products.SupplierID=Suppliers.SupplierID
ORDER BY ProductID

--18. Show all the products, with the associated CategoryName
SELECT ProductName,CategoryName
FROM Products
LEFT OUTER JOIN Categories ON Products.CategoryID=Categories.CategoryID

--19. For all orders ordered in 1997, show the OrderID, OrderDate (date
only), and
--CompanyName of the Shipper, and sort by OrderID.
SELECT OrderID, convert(date,OrderDate),CompanyName
FROM Orders
LEFT OUTER JOIN Shippers ON Orders.ShipVia=Shippers.ShipperID
WHERE year(OrderDate)=1997
ORDER BY OrderID
--20. Show number of products in each category. Sort the results by the
total number
--of products, in descending order.
SELECT CategoryName, COUNT(ProductID)
FROM Categories
LEFT OUTER JOIN Products ON Categories.CategoryID=Products.CategoryID
GROUP BY Categories.CategoryID,Categories.CategoryName
ORDER BY COUNT(ProductID) DESC

--21. Show the total number of customers per Country.
```

```sql
SELECT Country,COUNT(CustomerID)
FROM Customers
WHERE Country IS NOT NULL
GROUP BY Country


--22. Show the total number of customers per Country and City.
SELECT Country,City,COUNT(CustomerID)
FROM Customers
WHERE Country IS NOT NULL
GROUP BY Country,City
WITH ROLLUP
--23. What products do we have in our inventory that should be reordered.
SELECT ProductID,ProductName
FROM Products
WHERE ISNULL(UnitsInStock,0)+ISNULL(UnitsOnOrder,0)<=ISNULL(ReorderLevel,0)
--24. We'll define "products that need reordering" with the following:
SELECT ProductID,ProductName AS 'products that need reordering'
FROM Products
WHERE ISNULL(UnitsInStock,0)+ISNULL(UnitsOnOrder,0)<=ISNULL(ReorderLevel,0)
--25. Do the same but select products for which UnitsInStock plus
UnitsOnOrder are
--less than or equal to ReorderLevel and The Discontinued flag is false
(0).
SELECT ProductID,ProductName AS 'products that need reordering'
FROM Products
WHERE ISNULL(UnitsInStock,0)+ISNULL(UnitsOnOrder,0)<=ISNULL(ReorderLevel,0)
AND Discontinued=0
--26. 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 CustomerID,Region
FROM Customers
ORDER BY case when Region is null then 1 else 0 end, CustomerID
--27. 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.
SELECT TOP 3 ShipCountry,AVG(Freight)
FROM Orders
GROUP BY ShipCountry
ORDER BY AVG(Freight) DESC
--28. Do the same but now, instead of using all the orders we have, we only
want to
--see orders from the year 1997
SELECT TOP 3 ShipCountry,AVG(Freight)
FROM Orders
WHERE year(OrderDate)=1997
GROUP BY ShipCountry
ORDER BY AVG(Freight) DESC
```

```sql
--29. Do the same but now, 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 TOP 3 ShipCountry,AVG(Freight)
FROM Orders
WHERE DATEDIFF(month,OrderDate,(SELECT TOP 1 OrderDate FROM Orders ORDER BY
OrderDate DESC))<=12
GROUP BY ShipCountry
ORDER BY AVG(Freight) DESC
--30. There are some customers who have never actually placed an order.
Show these customers.
SELECT CustomerID,CompanyName
FROM Customers
WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders)
--31. There are some customers who have placed no orders in 1997. Show
these customers.
SELECT CustomerID,CompanyName
FROM Customers
WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders WHERE
year(OrderDate)=1997)
--32. One employee (Margaret Peacock, EmployeeID 4) has placed the most
orders.
--However, there are some customers who've never placed an order with her.
--Show only those customers who have never placed an order with her.
SELECT CustomerID,CompanyName
FROM Customers
WHERE CustomerID NOT IN (SELECT CustomerID FROM Orders WHERE EmployeeID=4)

--33. We want to send all of our high-value customers a special VIP gift.
We're
--defining high-value customers as those who've made at least 1 order with
a
--total value (not including the discount) equal to 10,000 or more. We only
want
--to consider orders made in the year 1996.
SELECT DISTINCT Customers.CustomerID,CompanyName
FROM Customers
INNER JOIN Orders ON Orders.CustomerID=Customers.CustomerID
LEFT OUTER JOIN [Order Details] ON [Order Details].OrderID=Orders.OrderID
AND year(OrderDate)=1996
GROUP BY Customers.CustomerID,Customers.CompanyName,Orders.OrderID
HAVING SUM(Quantity*UnitPrice)>=10000


--34. Do the same, but instead of requiring that customers have at least
one individual
--orders totaling $10,000 or more, he wants to define high-value customers
as
--those who have orders totaling $15,000 or more in 1996
SELECT DISTINCT Customers.CustomerID,CompanyName
FROM Customers
INNER JOIN Orders ON Orders.CustomerID=Customers.CustomerID
LEFT OUTER JOIN [Order Details] ON [Order Details].OrderID=Orders.OrderID
AND year(OrderDate)=1996
GROUP BY Customers.CustomerID,Customers.CompanyName
HAVING SUM(Quantity*UnitPrice)>=15000

--35. Do the same, but use the discount when calculating high-value
customers
SELECT DISTINCT Customers.CustomerID,CompanyName
```

```
FROM Customers
INNER JOIN Orders ON Orders.CustomerID=Customers.CustomerID
LEFT OUTER JOIN [Order Details] ON [Order Details].OrderID=Orders.OrderID
AND year(OrderDate)=1996
GROUP BY Customers.CustomerID,Customers.CompanyName
HAVING SUM(Quantity*UnitPrice*(1-Discount))>=15000
--36. Show all orders made on the last day of the month. Order by
EmployeeID and OrderID
SELECT OrderID,EmployeeID,OrderDate
FROM Orders
WHERE day(OrderDate)=(SELECT DAY(DATEADD(DD,-1,DATEADD(MM,DATEDIFF(MM,-
1,OrderDate),0))))
ORDER BY EmployeeID,OrderID
--37. Show the 10 orders with the most line items, in order of total line
items.
SELECT TOP 10 Orders.OrderID, SUM(Quantity)
FROM Orders
INNER JOIN [Order Details] ON Orders.OrderID=[Order Details].OrderID
GROUP BY Orders.OrderID
ORDER BY SUM(Quantity) DESC
--38. Show a random set of 2% of all orders
SELECT TOP (SELECT CONVERT(int,0.02*COUNT(OrderID)) FROM Orders) *
FROM Orders
ORDER BY NEWID()

--39. Some customers are complaining about their orders arriving late.
Which orders are late?
SELECT OrderID,RequiredDate,ShippedDate
FROM Orders
WHERE DATEDIFF(day,RequiredDate,ShippedDate)>0

--40. Which salespeople have the most orders arriving late
SELECT TOP 1 Employees.EmployeeID,FirstName,LastName,COUNT(OrderID)
FROM Employees
INNER JOIN Orders ON Orders.EmployeeID=Employees.EmployeeID AND
DATEDIFF(day,RequiredDate,ShippedDate)>0
GROUP BY Employees.EmployeeID,FirstName,LastName
ORDER BY COUNT(OrderID) DESC
--41. Which salespeople have the most orders arriving late, related to the
total orders per salesperson.
SELECT Employees.EmployeeID,FirstName,LastName,CAST(COUNT(o1.OrderID) AS
DECIMAL(3,2))/CAST(COUNT(o2.orderid) AS DECIMAL (3,2))
FROM Employees
INNER JOIN Orders o2 ON o2.EmployeeID=Employees.EmployeeID
INNER JOIN Orders o1 ON o1.OrderID=o2.OrderID AND
DATEDIFF(day,o1.RequiredDate,o1.ShippedDate)>0
GROUP BY Employees.EmployeeID,FirstName,LastName
ORDER BY CAST(COUNT(o1.OrderID) AS DECIMAL(3,2))/CAST(COUNT(o2.orderid) AS
DECIMAL (3,2)) DESC
--42. Show a list of all countries where suppliers and/or customers are
based
SELECT DISTINCT s.Country
FROM Suppliers AS s
UNION
SELECT DISTINCT c.Country
FROM Customers AS c
WHERE c.Country NOT IN (SELECT DISTINCT s.Country FROM Suppliers AS s)
--43. There are some customers for whom freight is a major expense when
ordering
--from Northwind. However, by batching up their orders, and making one
larger
```

```sql
--order instead of multiple smaller orders in a short period of time, they
could
--reduce their freight costs significantly. Show those customers who have
made
--more than 1 order in a 5 day period. The sales people will use this to
help
--customers reduce their costs.
SELECT DISTINCT
Customers.CustomerID,CompanyName,DATEDIFF(day,o.OrderDate,o1.OrderDate)
FROM Customers
INNER JOIN Orders o ON o.CustomerID=Customers.CustomerID
INNER JOIN Orders o1 ON o1.CustomerID=Customers.CustomerID
GROUP BY
Customers.CustomerID,CompanyName,o.OrderDate,o1.OrderDate,o.OrderID,o1.Orde
rID
HAVING DATEDIFF(day,o.OrderDate,o1.OrderDate)>0 AND
DATEDIFF(day,o.OrderDate,o1.OrderDate)<=5 AND o.OrderID!=o1.OrderID
```