1. Sales Representatives in the United States

Now we'd like to see only for those employees that both have the title of Sales Representative, and also are in the United States

```
SELECT * FROM Sales WHERE title = 'Sales Representative' AND Country = 'United States';
```

2. Orders placed by specific EmployeeID

Show all the orders placed by a specific employee. The EmployeeID for this Employee (Steven Buchanan) is 5.

```
SELECT * FROM Orders WHERE EmployeeID = 5;
```

3. Suppliers and ContactTitles

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 ContactTitle <> 'Marketing Manager';
```

4. Products with "queso" in ProductName

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%';
```

5. Orders shipping to France or Belgium

Looking at the Orders table, there's afield called ShipCountry. Write a query that shows the OrderID, CustomerID, and ShipCountry for the orders where the ShipCountryis either France or Belgium.

```
SELECT OrderID, CustomerID, ShipCountry
FROM Orders
WHERE ShipCountry = 'France' OR ShipCountry = 'Belgium';
```

```
SELECT OrderID, CustomerID, ShipCountry
FROM Orders WHERE ShipCountry IN ('France', 'Belgium');
```

6. Orders shipping to any country in Latin America

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 OrderID, CustomerID, ShipCountry FROM Orders
WHERE ShipCountry IN ('Brazil', 'Mexico', 'Argentina', 'Venezuela');
```

7. Employees, in order of age

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 ASC;
```

8. Showing only the Date with a DateTime field

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, DATE(BirthDate) AS BirthDate
FROM Employees ORDER BY BirthDate ASC;
```

9. Employees full name

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, CONCAT(FirstName, ' ', LastName) AS FullName
FROM Employees;
```

10.OrderDetails amount per line item

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 OrderID, ProductID, UnitPrice, Quantity, UnitPrice * Quantity AS TotalPrice
FROM OrderDetails ORDER BY OrderID, ProductID;
```