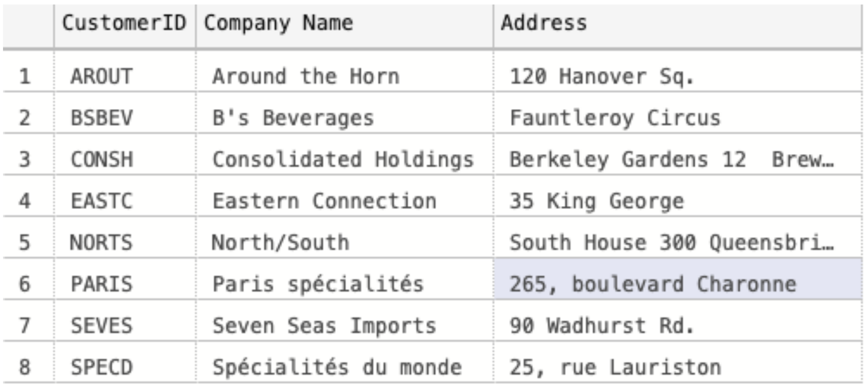
Jakub Matyjewicz
Java SDET

1.1 Write a query that lists all Customers in either Paris or London. Include Customer ID, Company Name and all address fields.

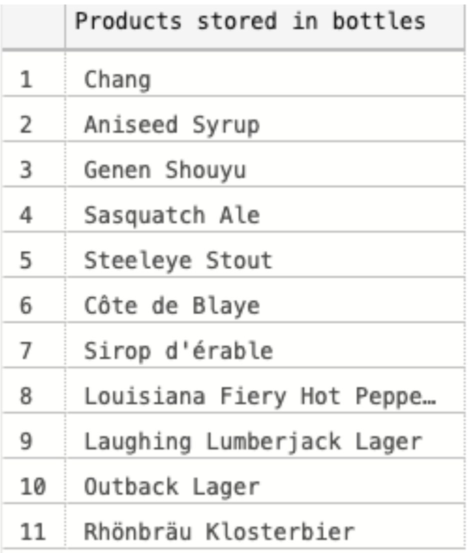
```
SELECT c.CustomerID, c.CompanyName AS "Company Name" , c.Address FROM Customers c WHERE c.City
IN('Paris','London');
```



	CustomerID	Company Name	Address
1	AROUT	Around the Horn	120 Hanover Sq.
2	BSBEV	B's Beverages	Fauntleroy Circus
3	CONSH	Consolidated Holdings	Berkeley Gardens 12 Brew…
4	EASTC	Eastern Connection	35 King George
5	NORTS	North/South	South House 300 Queensbri…
6	PARIS	Paris spécialités	265, boulevard Charonne
7	SEVES	Seven Seas Imports	90 Wadhurst Rd.
8	SPECD	Spécialités du monde	25, rue Lauriston

1.2 List all products stored in bottles.

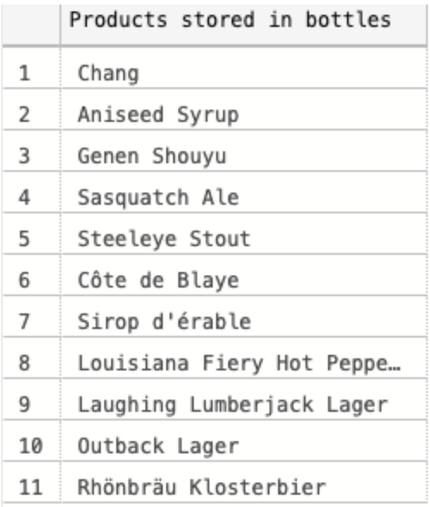
```
SELECT p.ProductName AS "Products stored in bottles" FROM Products p WHERE p.QuantityPerUnit
LIKE '%bottles%';
```



	Products stored in bottles
1	Chang
2	Aniseed Syrup
3	Genen Shouyu
4	Sasquatch Ale
5	Steeleye Stout
6	Côte de Blaye
7	Sirop d'érable
8	Louisiana Fiery Hot Peppe…
9	Laughing Lumberjack Lager
10	Outback Lager
11	Rhönbräu Klosterbier

1.3 Repeat question above but add in the Supplier Name and Country.

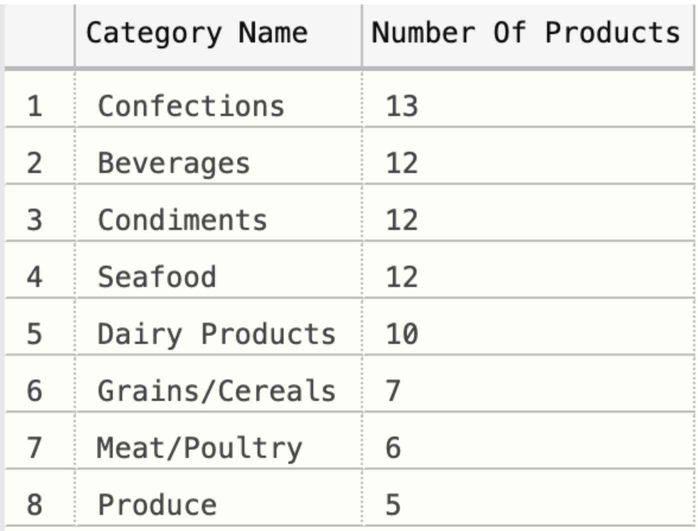
```
SELECT p.ProductName AS "Product Name", s.CompanyName AS "Supplier Name ", s.Country
FROM Products p LEFT JOIN Suppliers s ON s.SupplierID = p.SupplierID WHERE  p.QuantityPerUnit
LIKE '%bottles%';
```



	Products stored in bottles
1	Chang
2	Aniseed Syrup
3	Genen Shouyu
4	Sasquatch Ale
5	Steeleye Stout
6	Côte de Blaye
7	Sirop d'érable
8	Louisiana Fiery Hot Peppe…
9	Laughing Lumberjack Lager
10	Outback Lager
11	Rhönbräu Klosterbier

1.4 Write an SQL Statement that shows how many products there are in each category. Include Category Name in result set and list the highest number first.

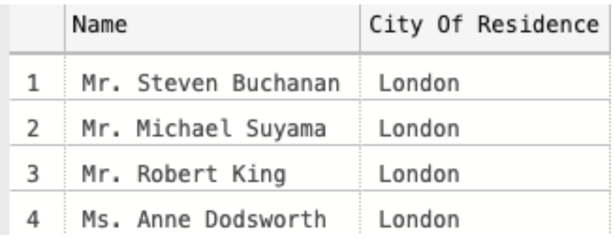
```
SELECT  c.CategoryName AS "Category Name", COUNT(p.ProductID) AS "Number Of Products"
FROM Products p JOIN Categories c ON p.CategoryID=c.CategoryID
GROUP BY c.CategoryName ORDER BY 2 DESC;
```



	Category Name	Number Of Products
1	Confections	13
2	Beverages	12
3	Condiments	12
4	Seafood	12
5	Dairy Products	10
6	Grains/Cereals	7
7	Meat/Poultry	6
8	Produce	5

1.5 List all UK employees using concatenation to join their title of courtesy, first name and last name together. Also include their city of residence.

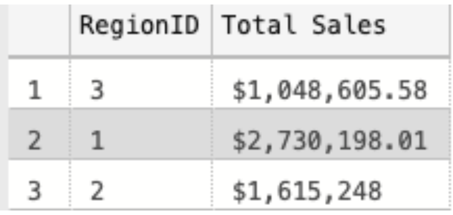
```
SELECT CONCAT(e.TitleOfCourtesy,' ', e.FirstName,' ',e.LastName) AS "Name", e.City AS "City Of
Residence" FROM Employees e WHERE e.Country='UK';
```



	Name	City Of Residence
1	Mr. Steven Buchanan	London
2	Mr. Michael Suyama	London
3	Mr. Robert King	London
4	Ms. Anne Dodsworth	London

1.6 List Sales Totals for all Sales Regions (via the Territories table using 4 joins) with a Sales Total greater than 1,000,000. Use rounding or FORMAT to present the numbers.

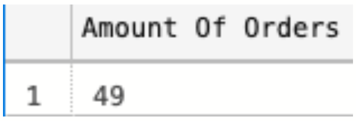
```
SELECT t.RegionID,
FORMAT(SUM(od.UnitPrice*od.Quantity*(1-od.Discount)),'$#,###,###.##') AS "Total Sales"
FROM Territories t INNER JOIN EmployeeTerritories et ON t.TerritoryID=et.TerritoryID
INNER JOIN Employees e ON et.EmployeeID = e.EmployeeID
INNER JOIN Orders o ON e.EmployeeID = o.EmployeeID
INNER JOIN [Order Details] od ON o.OrderID = od.OrderID
GROUP BY t.RegionID
HAVING SUM(od.UnitPrice*od.Quantity*(1-od.Discount)) >= 1000000;
```



	RegionID	Total Sales
1	3	$1,048,605.58
2	1	$2,730,198.01
3	2	$1,615,248

1.7 Count how many Orders have a Freight amount greater than 100.00 and either USA or UK as Ship Country.

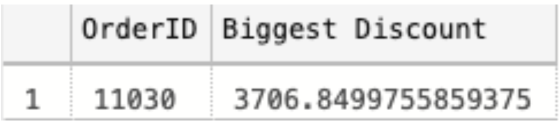
```
SELECT COUNT(o.OrderID) AS "Amount Of Orders"
FROM Orders o WHERE o.Freight >100  AND o.ShipCountry IN('USA','UK');
```



	Amount Of Orders
1	49

1.8 Write an SQL Statement to identify the Order Number of the Order with the highest amount(value) of discount applied to that order.

```
SELECT TOP 1 o.OrderID, SUM(o.Quantity*o.UnitPrice*o.Discount) AS "Biggest Discount"
FROM [Order Details] o GROUP BY o.OrderID ORDER BY 2 DESC;
-- select TOP 1 of  the biggest discount, group by id of order  and sort by second column which
is "Biggest Discount"
```



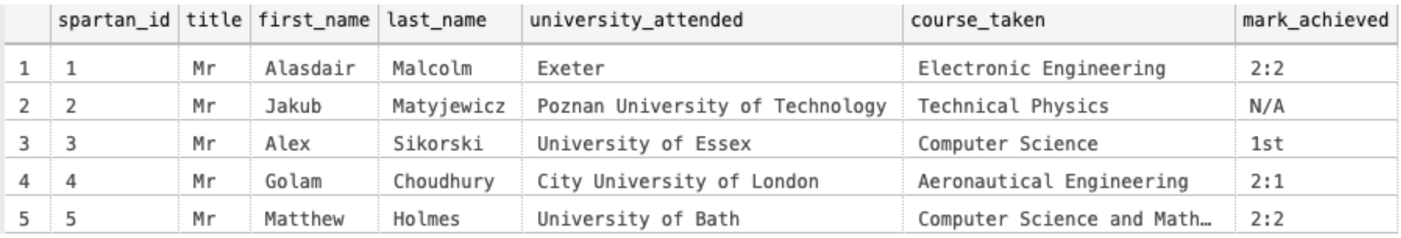
	OrderID	Biggest Discount
1	11030	3706.8499755859375

2.1 Write the correct SQL statement to create the following table:
Spartans Table – include details about all the Spartans on this course. Separate Title, First Name and Last Name into separate columns, and include University attended, course taken and mark achieved. Add any other columns you feel would be appropriate.
IMPORTANT NOTE: For data protection reasons do NOT include date of birth in this exercise.

```
CREATE TABLE spartan_table (
    spartan_id int IDENTITY(1,1) PRIMARY KEY,
 -- primary key identity start with 1 and auto increment by 1
    title VARCHAR(5),
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    university_attended VARCHAR(50),
    course_taken VARCHAR(50),
    mark_achieved CHAR(3)
-- CHAR(3) as the input can have only 3 characters (2:3, 2:2, 2:1 or 1st)
);
```

2.2 Write SQL statements to add the details of the Spartans in your course to the table you have created.

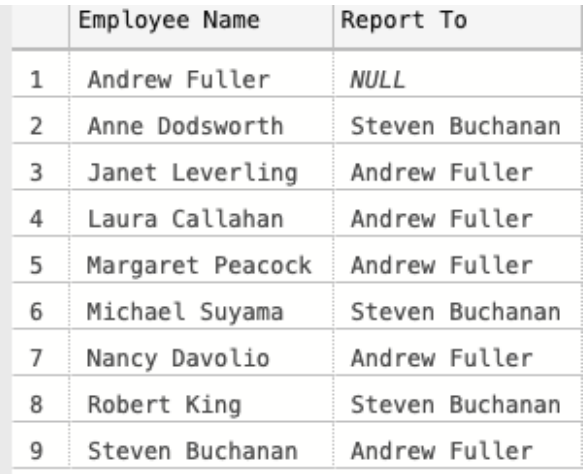
```
INSERT INTO spartan_table VALUES
('Mr', 'Alasdair', 'Malcolm', 'Exeter', 'Electronic Engineering','2:2'),
('Mr', 'Jakub', 'Matyjewicz', 'Poznan University of Technology', 'Technical Physics', 'N/A'),
('Mr', 'Alex', 'Sikorski', 'University of Essex', 'Computer Science', '1st'),
('Mr', 'Golam', 'Choudhury', 'City University of London', 'Aeronautical Engineering', '2:1'),
('Mr', 'Matthew', 'Holmes', 'University of Bath', 'Computer Science and Mathematics', '2:2');
-- no insert into column name as we are inserting in order
```



	spartan_id	title	first_name	last_name	university_attended	course_taken	mark_achieved
1	1	Mr	Alasdair	Malcolm	Exeter	Electronic Engineering	2:2
2	2	Mr	Jakub	Matyjewicz	Poznan University of Technology	Technical Physics	N/A
3	3	Mr	Alex	Sikorski	University of Essex	Computer Science	1st
4	4	Mr	Golam	Choudhury	City University of London	Aeronautical Engineering	2:1
5	5	Mr	Matthew	Holmes	University of Bath	Computer Science and Math…	2:2

Write SQL statements to extract the data required for the following charts:
3.1 List all Employees from the Employees table and who they report to. Please mention the Employee Names and the ReportTo names.

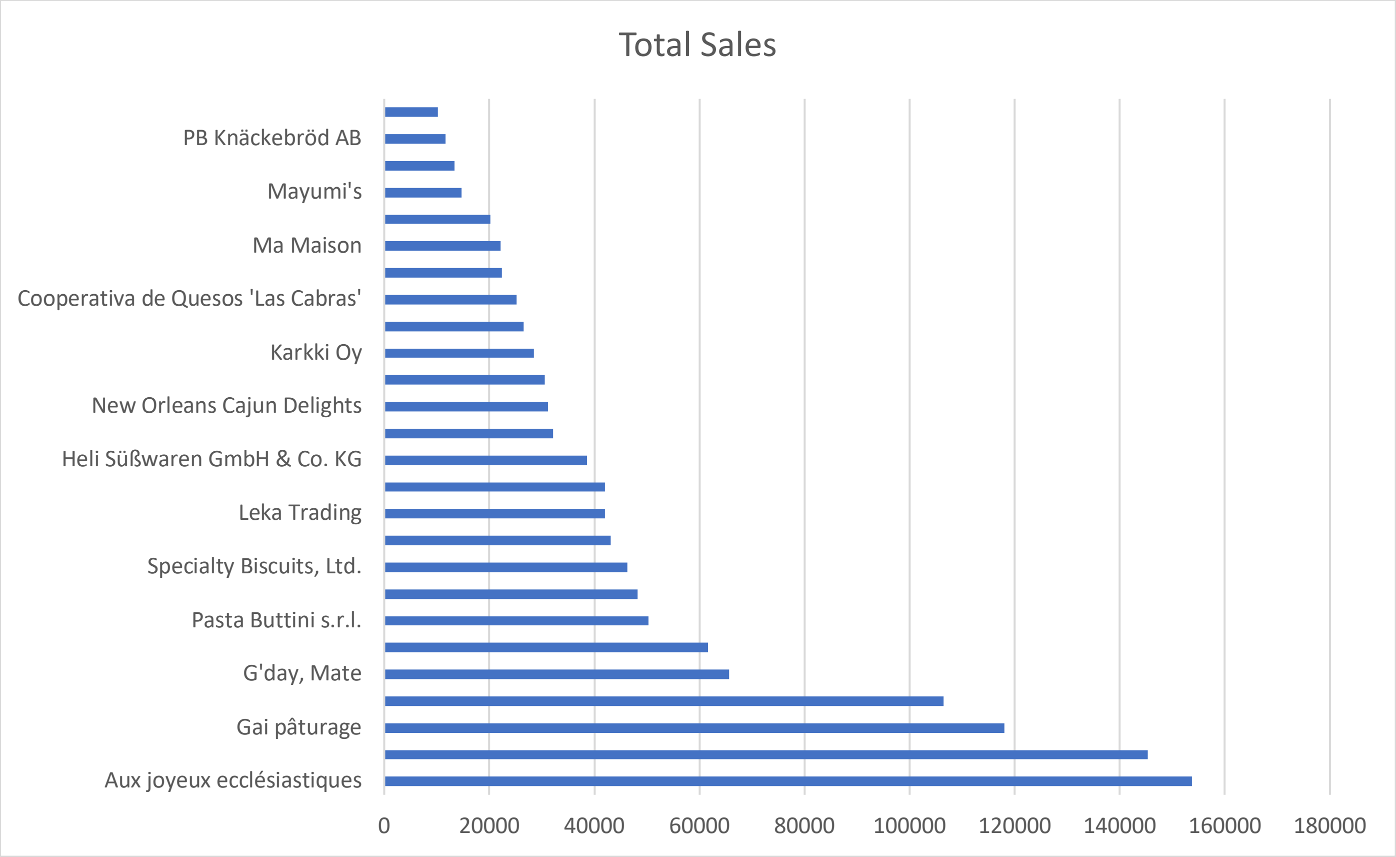
```
SELECT DISTINCT CONCAT(e.FirstName,' ',e.LastName) AS "Employee Name",
em.FirstName+' '+em.LastName AS "Report To"
FROM Employees e LEFT JOIN Employees em ON e.ReportsTo=em.EmployeeID;
-- LEFT JOIN as we don't want to miss any of person whose reportTo is NULL
```



	Employee Name	Report To
1	Andrew Fuller	*NULL*
2	Anne Dodsworth	Steven Buchanan
3	Janet Leverling	Andrew Fuller
4	Laura Callahan	Andrew Fuller
5	Margaret Peacock	Andrew Fuller
6	Michael Suyama	Steven Buchanan
7	Nancy Davolio	Andrew Fuller
8	Robert King	Steven Buchanan
9	Steven Buchanan	Andrew Fuller

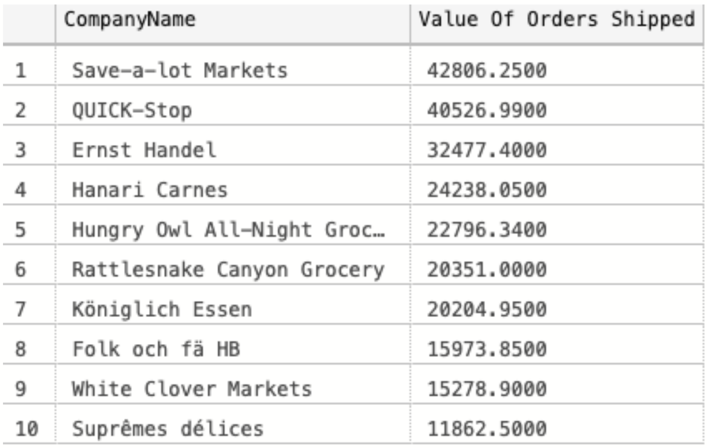
3.2 List all Suppliers with total sales over $10,000 in the Order Details table. Include the Company Name from the Suppliers Table and present as SEa bar chart

```
SELECT s.CompanyName, SUM((1-od.Discount)*od.UnitPrice*od.Quantity) AS "Total Sales"
FROM [Order Details] od
INNER JOIN Products p ON od.ProductID = p.ProductID
INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
GROUP BY s.CompanyName
HAVING SUM((1-od.Discount)*od.UnitPrice*od.Quantity) > 10000
ORDER BY 2 DESC;
```



3.3 List the Top 10 Customers YTD for the latest year in the Orders file. Based on total value of orders shipped.

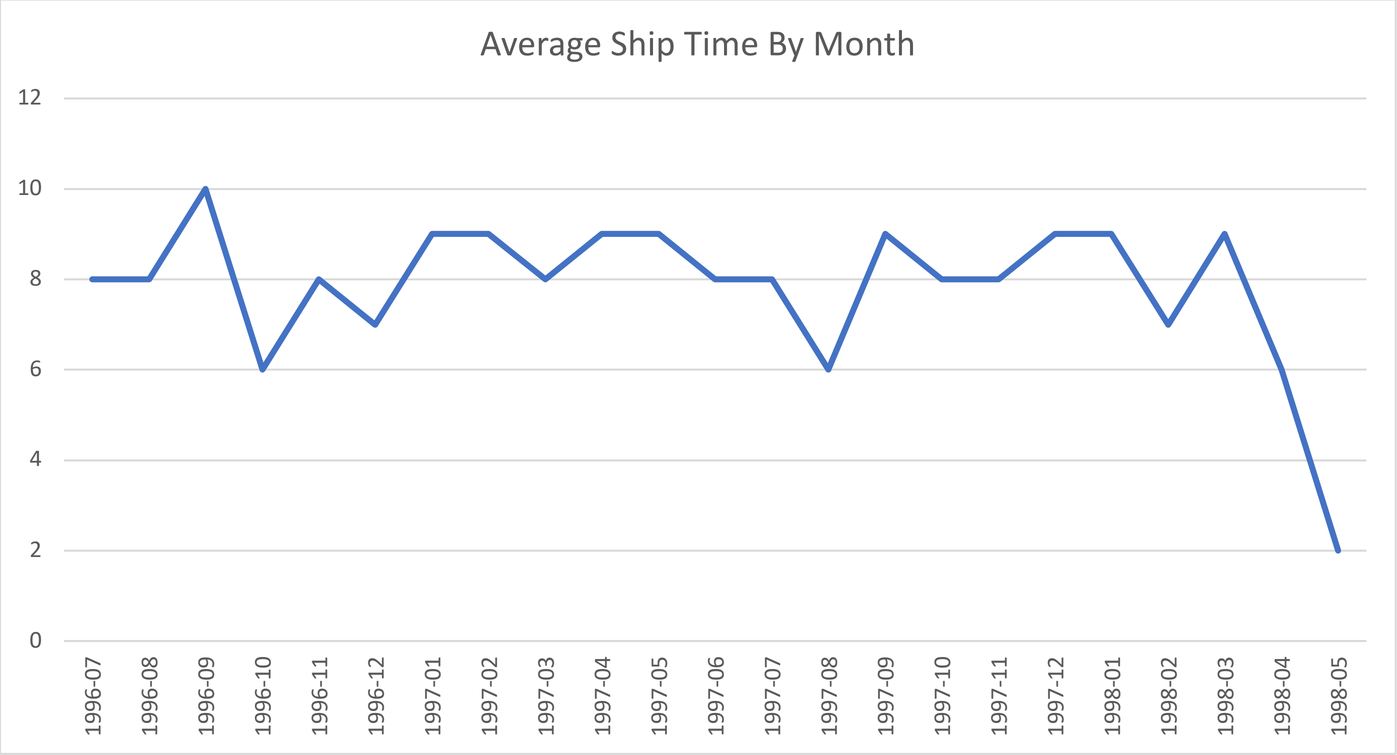
```
SELECT DISTINCT TOP 10 c.CompanyName,
SUM(od.UnitPrice*od.Quantity) AS "Value Of Orders Shipped"
FROM Orders o INNER JOIN [Order Details] od ON o.OrderID=od.OrderID
INNER JOIN Customers c ON o.CustomerID=c.CustomerID
WHERE FORMAT(o.OrderDate, 'year')
LIKE (SELECT TOP 1 FORMAT(ord.OrderDate, 'year') AS "Years" FROM Orders ord ORDER BY 1 DESC )
AND o.ShippedDate IS NOT NULL
GROUP BY c.CompanyName
ORDER BY 2 DESC;
```



	CompanyName	Value Of Orders Shipped
1	Save-a-lot Markets	42806.2500
2	QUICK-Stop	40526.9900
3	Ernst Handel	32477.4000
4	Hanari Carnes	24238.0500
5	Hungry Owl All-Night Groc…	22796.3400
6	Rattlesnake Canyon Grocery	20351.0000
7	Königlich Essen	20204.9500
8	Folk och fä HB	15973.8500
9	White Clover Markets	15278.9000
10	Suprêmes délices	11862.5000

3.4 Plot the Average Ship Time by month for all data in the Orders Table.

```sql
SELECT FORMAT(o.OrderDate, 'yyyy-MM') AS "Month",
AVG(DATEDIFF(dd,o.OrderDate,o.ShippedDate)) AS "Average Ship Time By Month"
FROM Orders o GROUP BY FORMAT(o.OrderDate, 'yyyy-MM') ORDER BY 1;
-- Calculating average difference between two dates and group them by month
```


Average Ship Time By Month