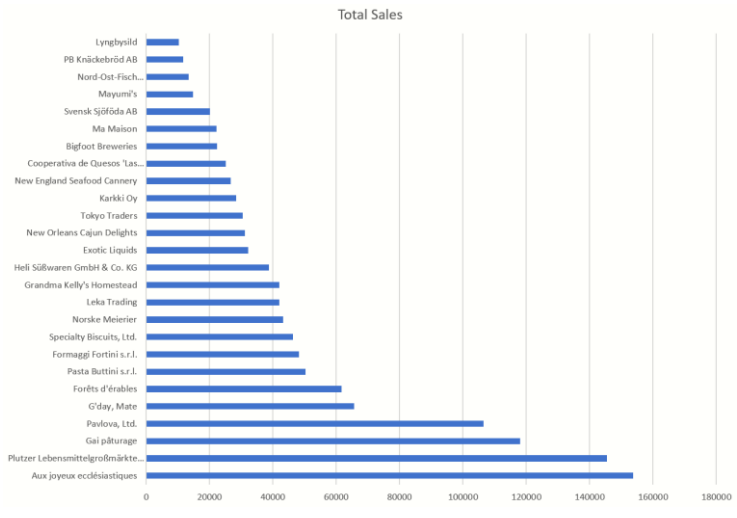
SQL Mini Project

```
USE Northwind

--1.1
SELECT CustomerID, CompanyName, Address, City, Region, PostalCode, Country
FROM Customers
--WHERE IN Limits the results in City to just Paris OR London
WHERE City IN ('Paris','London')

--1.2
SELECT *
FROM Products
--Wildcard searches for specific characters anywhere in QuantityPerUnit and limits the result to just them
WHERE QuantityPerUnit LIKE '%bottle%'

--1.3
--INNER JOIN Suppliers so that the suppliers name and country can be added in the select
SELECT p.ProductName, s.CompanyName, s.Country FROM Products p
INNER JOIN Suppliers s ON p.SupplierID = s.SupplierID
WHERE QuantityPerUnit LIKE '%bottle%'

--1.4
--COUNT adds up all values grouped together by GROUP BY
SELECT c.CategoryID, c.CategoryName, count(p.CategoryID) AS "Number Of Products"
FROM Categories c
INNER JOIN Products p ON p.CategoryID = c.CategoryID
--Group columns together that have the same CategoryID and CategoryName
GROUP BY c.CategoryID, c.CategoryName
--Order the table so that greatest Number Of Products is at the top
ORDER BY "Number Of Products" DESC

--1.5
--Concatination to add multiple columns into one column
SELECT CONCAT(Title,', ',FirstName,' ',LastName,' From ',City) AS "Employees From UK"
FROM Employees
WHERE Country = 'UK'

--1.6
--Calculate the sales totals for each region
SELECT ROUND(SUM((od.UnitPrice*od.Quantity*(1-od.Discount))),2) AS "Sales Total", r.RegionID, r.RegionDescription
FROM [Order Details] od
--4 Inner Joins to connect Order Details to Region
INNER JOIN Orders o ON od.OrderID=o.OrderID
INNER JOIN EmployeeTerritories e ON o.EmployeeID=e.EmployeeID
INNER JOIN Territories t ON e.TerritoryID=t.TerritoryID
INNER JOIN Region r ON t.RegionID=r.RegionID
GROUP BY r.RegionID, r.RegionDescription
HAVING ROUND(SUM((od.UnitPrice*od.Quantity*(1-od.Discount))),2) > 1000000
ORDER BY "Sales Total" DESC

--1.7
SELECT count(*) AS "Total Orders"
FROM Orders
--Limiting results to have over 100 Orders and Order to either USA or UK
WHERE (Freight > 100) AND ShipCountry IN ('USA','UK')

--1.8
--DISTINCT removes any duplicates
--TOP 2 limits the rows to just the top 2 rows
--The top 2 discounts both have the same value
SELECT DISTINCT TOP 2 od.OrderID, od.UnitPrice*od.Quantity*od.Discount AS "Discount Applied"
FROM [Order Details] od
INNER JOIN Orders o ON od.OrderID=o.OrderID
ORDER BY "Discount Applied" DESC
```

```sql
--2.1
DROP DATABASE IF EXISTS ALynch_db
CREATE DATABASE  ALynch_db;
USE ALynch_db;

--Delete table if it already exists and then creates it
DROP TABLE IF EXISTS sparta_table
CREATE TABLE spartan_table
(
    --Creating columns for all the details
    title char(10),
    first_name char(20),
    last_name char(20),
    university_attended char(40),
    years_at_university char(2),
    course_taken char(40),
    mark_achieved char(5)
);

--2.2
--Inserting values for five people into the pre-created table
INSERT INTO spartan_table (
    title, first_name, last_name, university_attended, years_at_university, course_taken, mark_achieved
)
VALUES (
    'Mr', 'Henry', 'Bath', 'University of cloud', '4', 'Art and some other stuff', 'First'
),(
    'Miss', 'Charlotte', 'Kimble', 'University of Fish', '3', 'Fish Studies', '2:1'
),(
    'Ms', 'Chloe', 'Womble', 'Womble Universty', '4', 'Cleaning and tidyness', 'First'
),(
    'Mr', 'Robert', 'Richard', 'Knowhere University', '10', 'Trying to find the University', 'Fail'
),(
    'Miss', 'Kim', 'Kimble', 'Magic University', '3', 'Potions, propulsion and potatoes', '2:2'
)

SELECT * FROM spartan_table

--3.1
USE Northwind

--Join Employees onto itself so that ReportsTo can be keyed to a name.
SELECT CONCAT(em.FirstName,' ',em.LastName) AS "Employee Name", CONCAT(e.FirstName,' ',e.LastName) AS "Reports To"
FROM Employees e
RIGHT JOIN Employees em ON e.EmployeeID=em.ReportsTo

--3.2
--Calculate total sales and then use HAVING to filter the results
--Two Joins to connect Suppliers and Order Details
SELECT s.CompanyName, ROUND(SUM((od.UnitPrice*od.Quantity*(1-od.Discount))),2) AS "Total Sales"
FROM Suppliers s
INNER JOIN Products p ON s.SupplierID=p.SupplierID
INNER JOIN [Order Details] od ON p.ProductID=od.ProductID
GROUP BY s.CompanyName
HAVING ROUND(SUM((od.UnitPrice*od.Quantity*(1-od.Discount))),2) > 10000
ORDER BY "Total Sales" DESC
```



```sql
--3.3
--Subquery to gain value for latest year in table
SELECT TOP 10 c.CompanyName AS "Company Name", ROUND(SUM((od.UnitPrice*od.Quantity*(1-od.Discount))),2) AS "YTD"
FROM Customers c
INNER JOIN Orders o ON c.CustomerID=o.CustomerID
INNER JOIN [Order Details] od ON o.OrderID=od.OrderID
WHERE YEAR(ShippedDate)=(SELECT MAX(YEAR(o.ShippedDate)) FROM Orders o)
GROUP BY c.CompanyName
ORDER BY ROUND(SUM((od.UnitPrice*od.Quantity*(1-od.Discount))),2) DESC

--3.4
--Calculate Difference between OrderDate and ShippedDate to calculate how long that ship time was.
--Cast the reult as a float so that it can be averaged and rounded
SELECT CONCAT(MONTH(OrderDate),'-', YEAR(OrderDate)) AS "Month",
ROUND(AVG(CAST(DATEDIFF(d,OrderDate,ShippedDate) AS FLOAT )),2) AS "Average Ship Time By Month"
FROM Orders
GROUP BY YEAR(OrderDate),MONTH(OrderDate)
ORDER BY YEAR(OrderDate),MONTH(OrderDate)
```