CS50 for MBAs

SQL

Pedro Farias

Built on slides by Will Claybaugh

23-Feb-2018

Overview of the Discussion:

1. Why Databases Matter
2. Relations vs Non-relation
3. Basic SQL Commands

What’s wrong with Old Faithful?

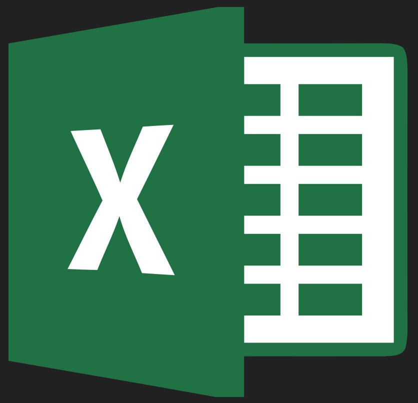
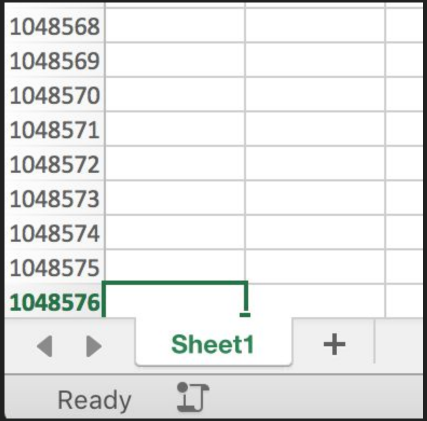


Image source: Wikimedia

1,048,576 rows, 15 digit precision



What is a Database?

- A file on a hard drive
- Databases are not the physical computers
- Key questions:
    - How are they structured?
    - What happens when the file becomes too big?

CAP Theorem (You Can’t Have It All)

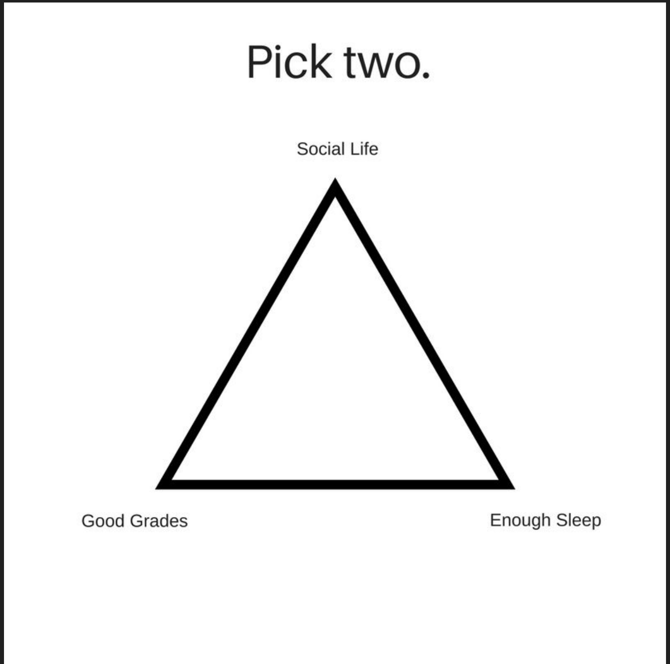


Image source: Legion

CAP Theorem (You Can’t Have It All)

- Pick two:
- Consistency: Users always get the most recent data
- Availability: Always get a response (even if outdated)
- Partition Tolerance: Keeps working if the network goes to heck

CAP Theorem (You Can’t Have It All)

- Pick two:
- Consistency: Users always get the most recent data
- Availability: Always get a response (even if outdated)
- **Partition Tolerance**: Keeps working if the network goes to heck

- But the network will eventually go to heck…
- Consistency vs. Availability will help determine whether you want a SQL or No SQL database

SQL vs. No SQL

Trade-offs abound!

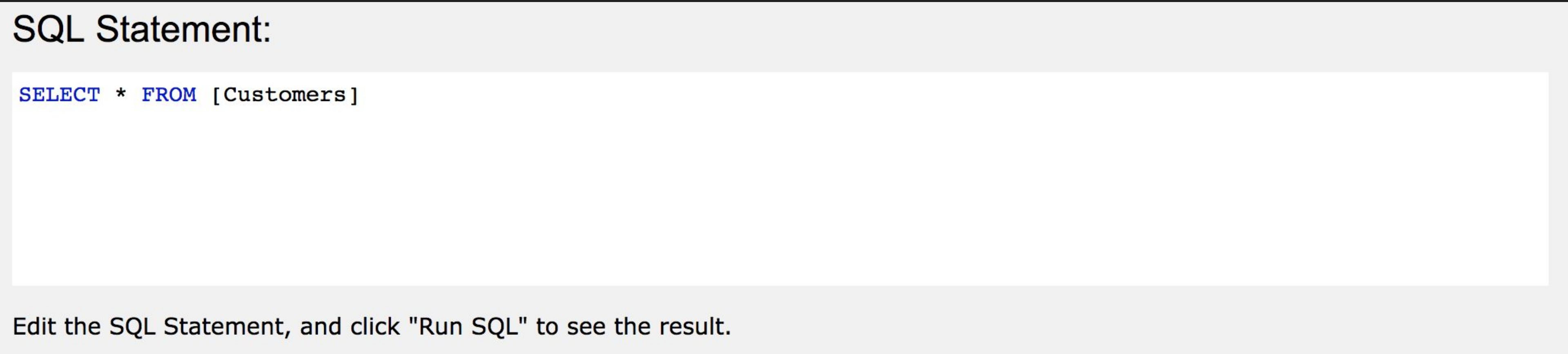
SQL	No SQL
• Consistency- You’ll always see up-to-date data	• Availability- You’ll get something reasonable
• ACID- The database will properly handle [+10, -10], even if those commands are interrupted	• ACID is negotiable- Maybe it’s fine that edits to a Facebook post get lost
• Requires cross-referencing many tables to understand an entry	• Duplicates data, but once you find an entry you have it all
• Hard to scale to multiple machines	• Designed to scale to multiple machines [easy sharding]

SQL	No SQL
• Always uses the row/column model	• Lots of different flavors, each with pros and cons
• All entries have same properties, settled at database creation	• New features can be added to single records, on the fly
• Built for never losing data, and always being right	• Can be very reliable, or can be optimized for size/speed
• **Think: banking, shipping records**	• **Think: Facebook posts, YouTube videos**

Summary: SQL vs. No SQL

- Row/Column vs. Flexible
- Maturity vs. Scalability
- Banking is different from Twitter

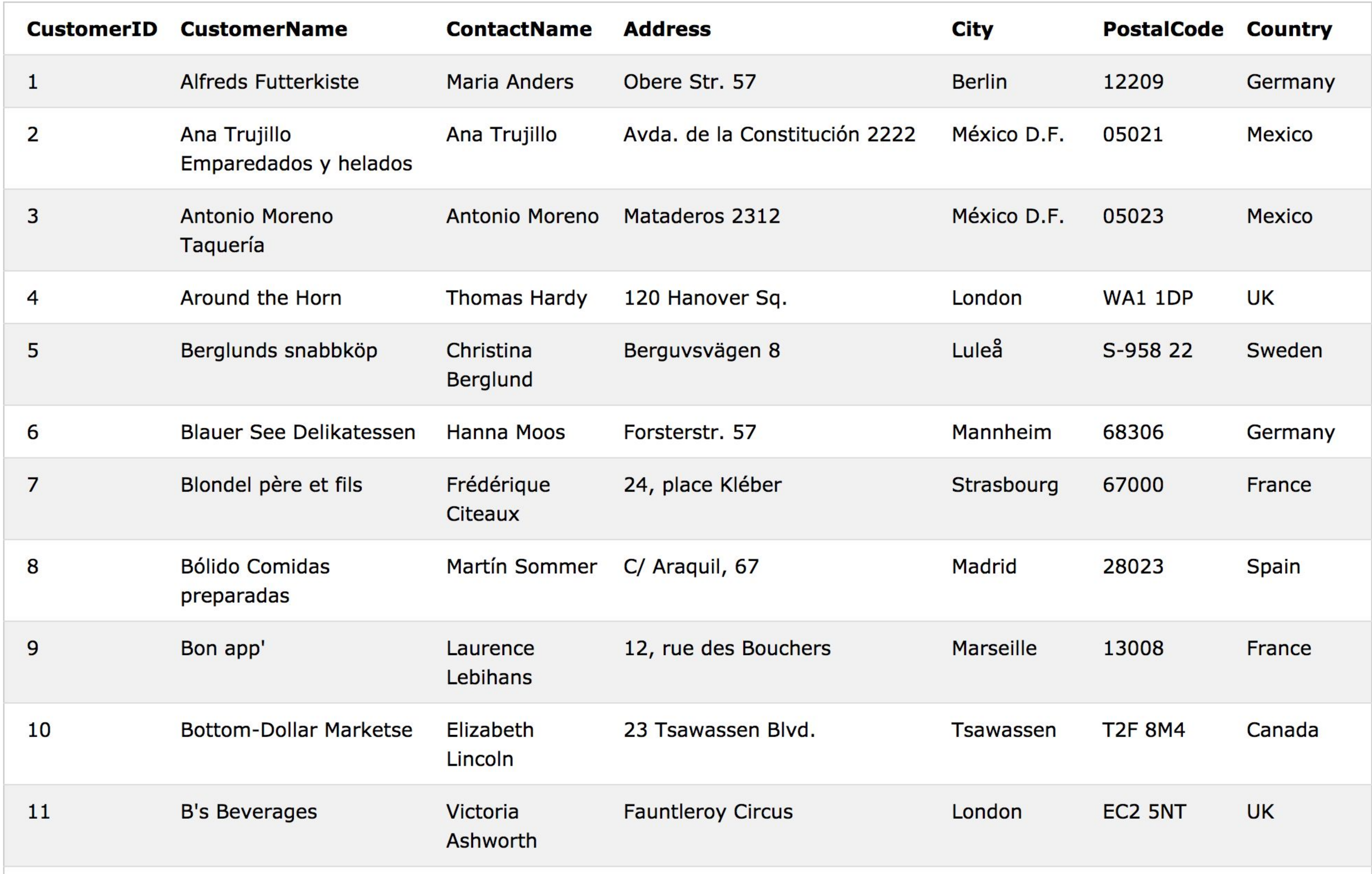
European Grocery (1996/7) Demo:
https://www.w3schools.com/sql/trysql.asp?filename=trysql_create_table



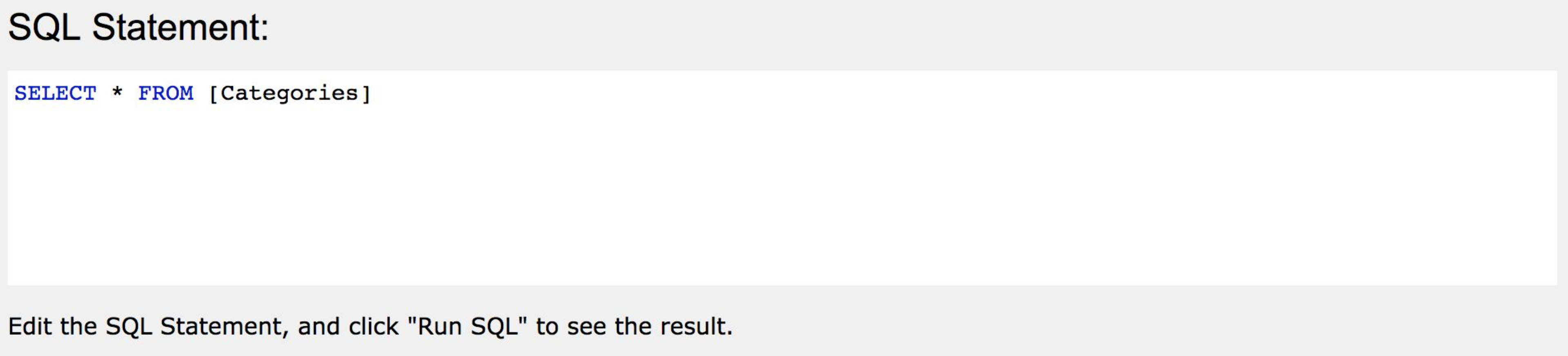
SQL Statement:

```sql
SELECT * FROM [Customers]
```

Edit the SQL Statement, and click "Run SQL" to see the result.



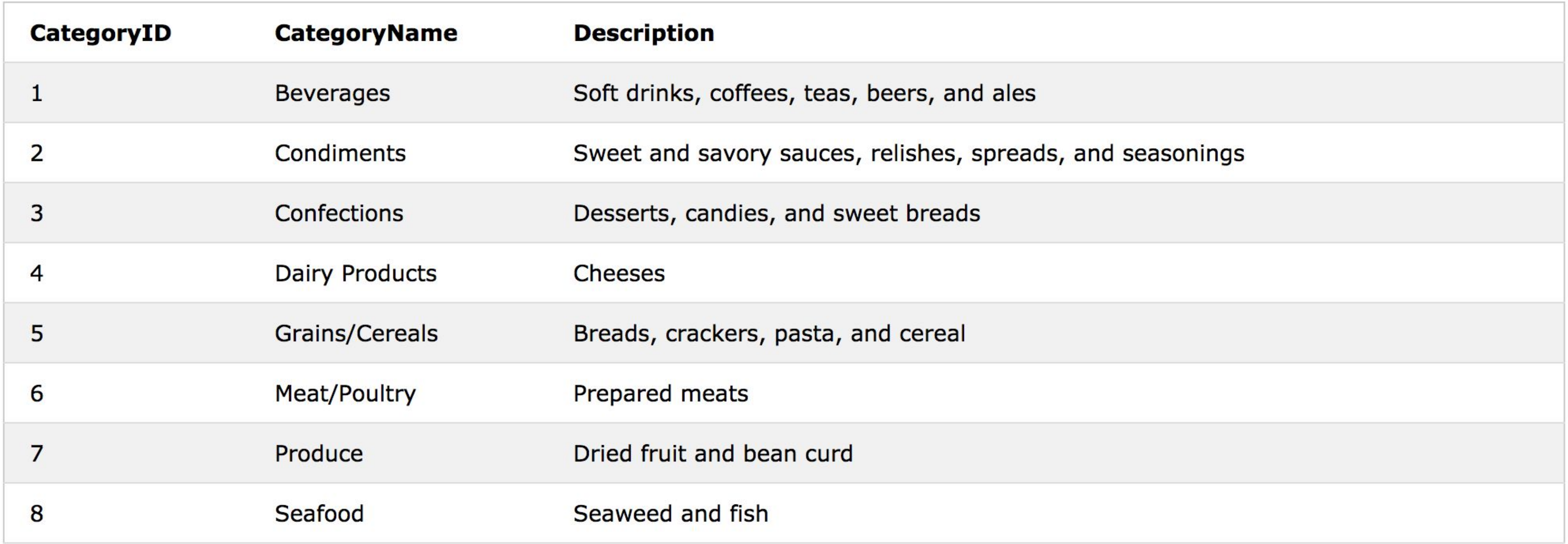
CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
6	Blauer See Delikatessen	Hanna Moos	Forsterstr. 57	Mannheim	68306	Germany
7	Blondel père et fils	Frédérique Citeaux	24, place Kléber	Strasbourg	67000	France
8	Bólido Comidas preparadas	Martín Sommer	C/ Araquil, 67	Madrid	28023	Spain
9	Bon app'	Laurence Lebihans	12, rue des Bouchers	Marseille	13008	France
10	Bottom-Dollar Marketse	Elizabeth Lincoln	23 Tsawassen Blvd.	Tsawassen	T2F 8M4	Canada
11	B's Beverages	Victoria Ashworth	Fauntleroy Circus	London	EC2 5NT	UK



SQL Statement:

```
SELECT * FROM [Categories]
```

Edit the SQL Statement, and click "Run SQL" to see the result.



CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
2	Condiments	Sweet and savory sauces, relishes, spreads, and seasonings
3	Confections	Desserts, candies, and sweet breads
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta, and cereal
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean curd
8	Seafood	Seaweed and fish

Insert a “Utensils” category to the “Categories” table

```
INSERT INTO Categories (colname, colname)
VALUES (v1, v2)
```

“Orders” Table

So much normalization! What is normalization?

Select only the orders filled by Employee #4

```
SELECT * FROM [Orders] WHERE EmployeeID == 4
```

Select only the orders by Customer #10 that were filled by Employee #4

```
SELECT * FROM [Orders] WHERE EmployeeID ==
4 AND CustomerID=10
```

Select the orders by Customer #10 or those that were filled by Employee #4

```sql
SELECT * FROM [Orders] WHERE EmployeeID == 4 OR CustomerID == 10
```

Update the database: Employee #5 actually only uses Shipper #3!

✗

```
UPDATE Orders SET ShipperID == 3
UPDATE Orders SET ShipperID == 3 WHERE
EmployeeID == 5
```

How do I see the change?

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

The same customer should have access to different shippers to optimize the system. Let’s check this is the case

```
SELECT ShipperID,CustomerID FROM Orders
ORDER BY CustomerID
```

Summary: SQL Commands

- SELECT: Get specific columns
- INSERT: Add new rows
- UPDATE: Change an existing row
- WHERE: Specify which particular rows
- ORDER BY: How to sort the results

“OrderDetails” Table

Which product is the most popular?

```
SELECT ProductID,SUM(Quantity) FROM
[OrderDetails] GROUP BY ProductID
```

```
SELECT ProductID,SUM(Quantity) FROM [OrderDetails] GROUP BY ProductID ORDER BY SUM(Quantity)
```

SUM, COUNT, MIN, MAX, or AVG
DESC

But what even are these products? How can we join databases?

```
SELECT * FROM OrderDetails JOIN Products ON
OrderDetails.ProductID == Products.ProductID
```

TableName.ColumnName

Summary: SQL Commands

- SELECT: Get specific columns
- INSERT: Add new rows
- DELETE: Removes rows
- UPDATE: Change an existing row
- WHERE: Specify which particular rows
- ORDER BY: How to sort the results
- COUNT: Counts the number of occurrences
- GROUP BY: Combines to one response per unique entry
- JOIN: Ties two normalized tables back together
- SUM/MAX/MIN/AVG

Practice Questions

- Q1: How many customers in Germany?
    - A: 11
- Q2: How many units of product 29 did we ship?
    - A: 168
- Q3: How many orders in October?
    - A: 26. Research the BETWEEN clause
- Q4: Which country has the most customers?
    - USA. But you should be able to tell me who is tied for 2nd
- Q5: Which country has the most orders?
    - USA with 29. But should be able to tell me and verify the number in Belgium
- Q6: Which customer placed the most orders?
    - Ernst Handel with 10

Practice Answers

- Q1: How many customers in Germany?
  - SELECT Count(*) FROM Customers WHERE Country='Germany'
- Q2: How many units of product 29 did we ship?
  - SELECT Sum(Quantity) FROM [OrderDetails] where ProductID=29
- Q3: How many orders in October?
  - SELECT Count(*) FROM [Orders] WHERE OrderDate BETWEEN '1996-10-01' AND '1996-10-31‘
- Q4: Which country has the most customers?
  - SELECT Count(*),Country FROM [Customers] GROUP BY Country
- Q5: Which country has the most orders?
  - SELECT count(*),Customers.Country FROM [Orders] JOIN Customers on Customers.CustomerID=Orders.CustomerID GROUP BY Customers.Country
- Q6: Which customer placed the most orders
  - SELECT Count(*),CustomerName FROM Orders JOIN Customers ON Customers.CustomerID=Orders.CustomerID GROUP BY Orders.CustomerID ORDER BY Count(*) DESC