

Main tables (Entities with information):

- Customers (CustomerID, plus info about customers)
 - o 1 to Many relation to OrderDetails via the Orders table (ie a customer may have many orders but each order only has 1 customer – In Orders orderID is unique but CustomerID is not)
- Categories (CategoryID, plus info about categories)
 - o Many to 1 relation to Products via a CategoryID column in Products. Each product has 1 category but a category can describe many products.
- Employees (EmployeeID, plus info about employees)
 - o 1 to Many relation to OrderDetails via the Orders table (ie an Employee may have many orders but each order only has 1 employee – In Orders orderID is unique but EmployeeID is not)
- OrderDetails (OrderDetailID, OrderID, ProductID, quantity)
- Products (ProductID, plus info about products)
 - o 1 To Many relation to OrderDetails via Orders table (ie a product can be part of many orderDetails but each orderDetails has only 1 product and 1 quantity.
 - o 1 to Many relation to Categories via categoryID column in Products
- Shippers (ShipperID, plus info about shippers)
 - o 1 to Many relation to OrderDetails via Orders table (ie a shipper can ship many orderDetails but each orderDetails has only 1 shipper.
- Suppliers (SupplierID, plus info about suppliers)
 - o Suppliers has a 1 to many relation with Products via a supplierID field in Products. productID is unique but supplierID is not. (note this might no be a good design because if a product could have many suppliers, the company might get a better price. But as this Database is designed, all products can only have 1 supplier)

Some SQL statement to try (cut and paste edit if you like)

SQL 1) Add your company as a customer

```
insert into customers
```

```
(CustomerName, ContactName,Address,City,PostalCode, Country)
```

```
values('BB LLC','Bill Byrne','400 Cedar Avenue','West Long Branch','07740','USA')
```

SQL 1a) Check that the row was added correctly

```
select * from customers
```

note: the field customerID is an auto increment field, which means when you add a new record, you don't need to supply the customerID, the database will assign the next number to it.

SQL 2) List all orders that include the product Boston Crab Meat

```
SELECT *  
FROM orderDetails, products  
WHERE orderDetails.productid = products.productid  
AND products.productName = 'Boston Crab Meat'
```

SQL 3) List all products that are in category of Seafood in order of price

```
SELECT *  
FROM products, categories  
WHERE products.categoryID = categories.categoryID  
AND categories.categoryName = 'Seafood'  
ORDER by price
```