



# CPSC203 – Introduction to Problem Solving and Using Application Software

Winter 2010

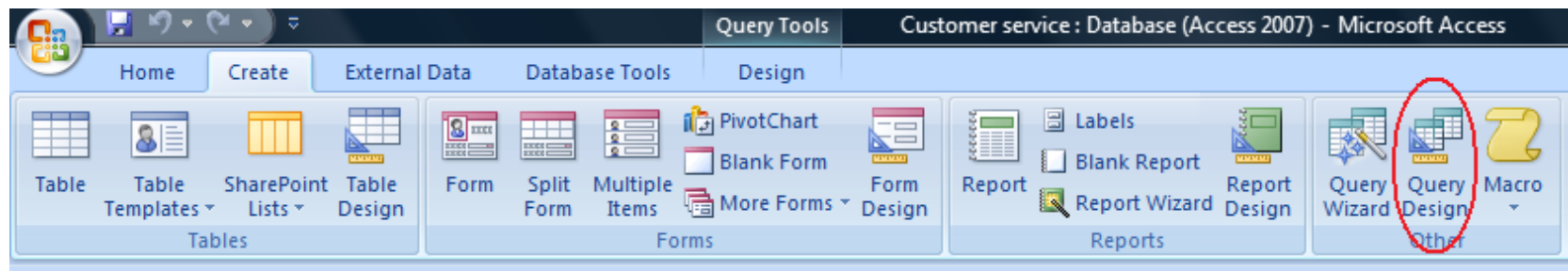
Tutorial 8: Mehrdad Nurolahzade

# Introduction

- Single-table queries
- Table relationships
- Multi-table queries
- Aggregate queries

# Single-Table Queries (1)

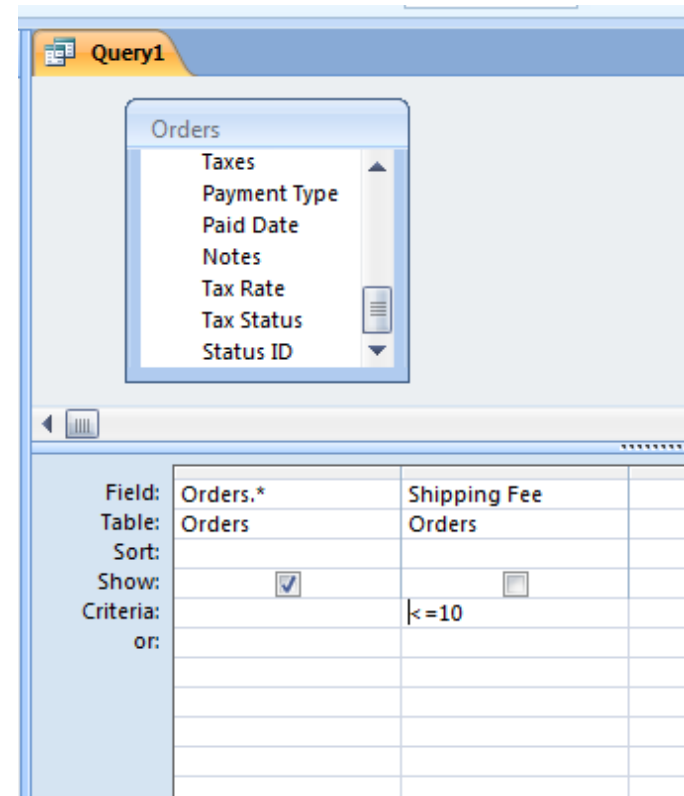
- Select Create > Query Design



- From the dialog box select a table/query and click on the **Add** button.
- Click **Close** to dismiss the dialog box.

# Single-Table Queries (2)

- Select fields
- Modify sort order
- Modify selection criteria
- Switch to **Datasheet View** to see query results
- Switch to **SQL View** to see query statement
- Switch to **Design View** to modify query



# Single-Table Queries (3)

Give me all the data in a table:

```
SELECT *  
FROM table_name
```

```
SELECT table_name.*  
FROM table_name
```

Give me all the data in some of the fields of a table:

```
SELECT field1, field2, ..., fieldN  
FROM table_name
```

```
SELECT table_name.field1, table_name.field2, ..., table_name.fieldN  
FROM table_name
```

# Single-Table Queries (4)

Give me all the data in a table that meets some condition(s):

```
SELECT *  
FROM table_name  
WHERE condition
```

```
SELECT *  
FROM Customer  
WHERE Age>20
```

```
SELECT *  
FROM Customer  
WHERE Age>=20 AND Age<30
```

```
SELECT *  
FROM Customer  
WHERE Age=20 OR Name='David'
```

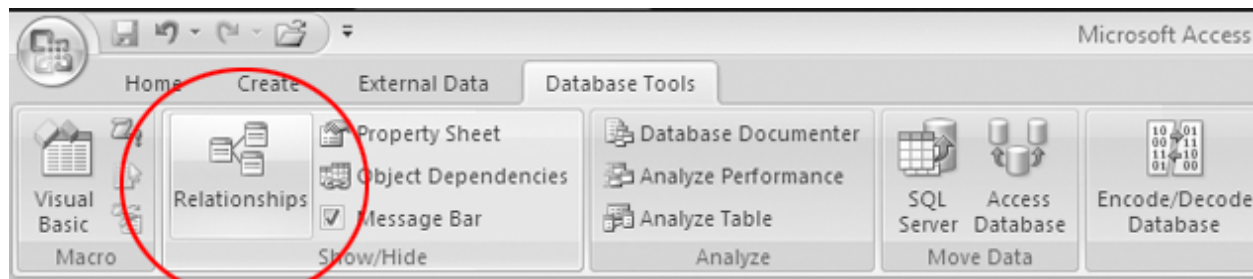
```
SELECT *  
FROM Customer  
WHERE City!='Calgary'
```

# Table Relationships

- Relationships are essentially links that associate a field in one table with a field in another table.
- Creating table relationships are important if we want to be able perform queries with multiple tables.

# Creating Relationships (1)

- Select Database Tools > Relationships

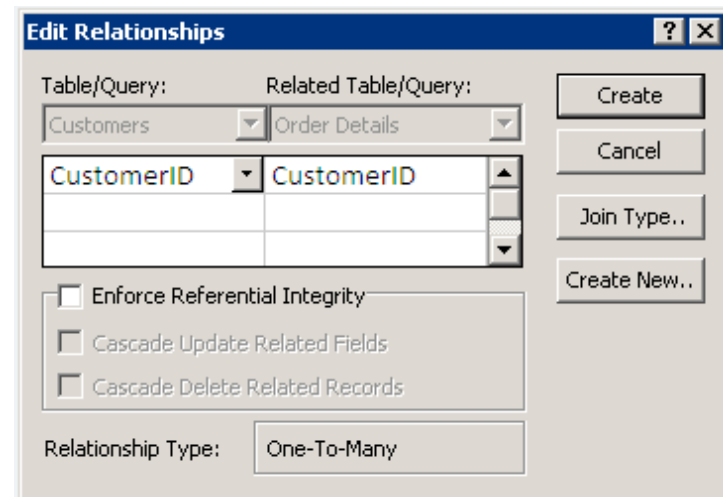


- From the dialog box select a tables/queries and click on the **Add** button.
- Click **Close** to dismiss the dialog box.



# Creating Relationships (2)

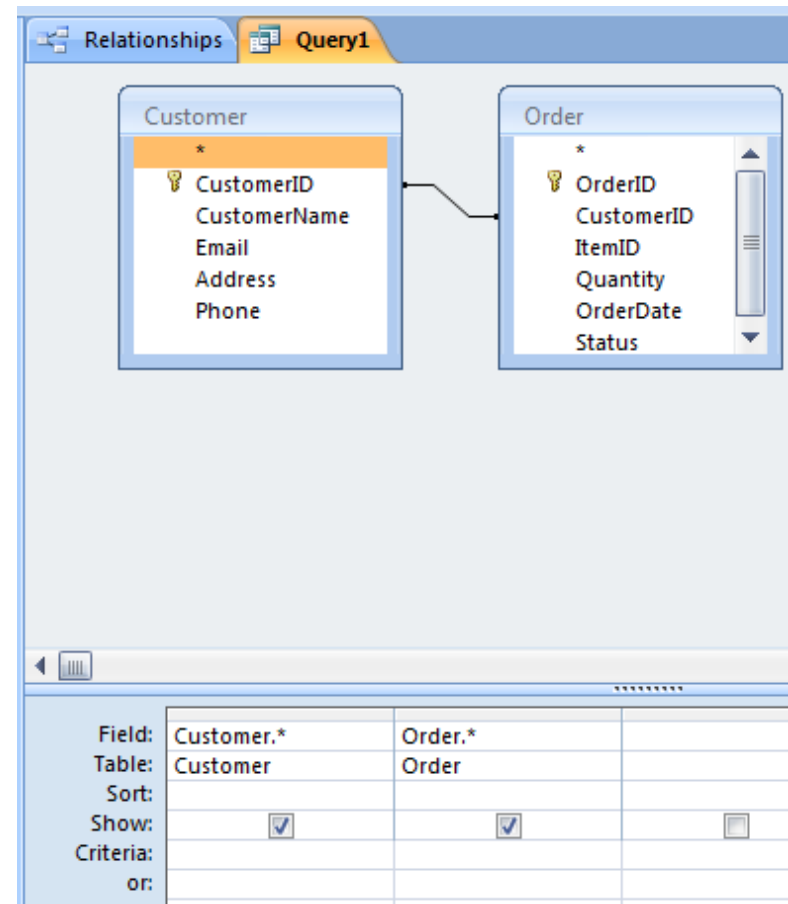
- Determines the fields in two tables that will participate in the relationship.
- Click the **Primary Key** and drag it into the **Foreign Key**.
- In **Edit Relationships** window press **Create**.



# Multi-Table Queries (1)

- Just like Single-Table Queries but this time add multiple tables/queries.

```
SELECT *  
FROM Customer, Order  
WHERE Customer.CustomerID=Order.CustomerID
```

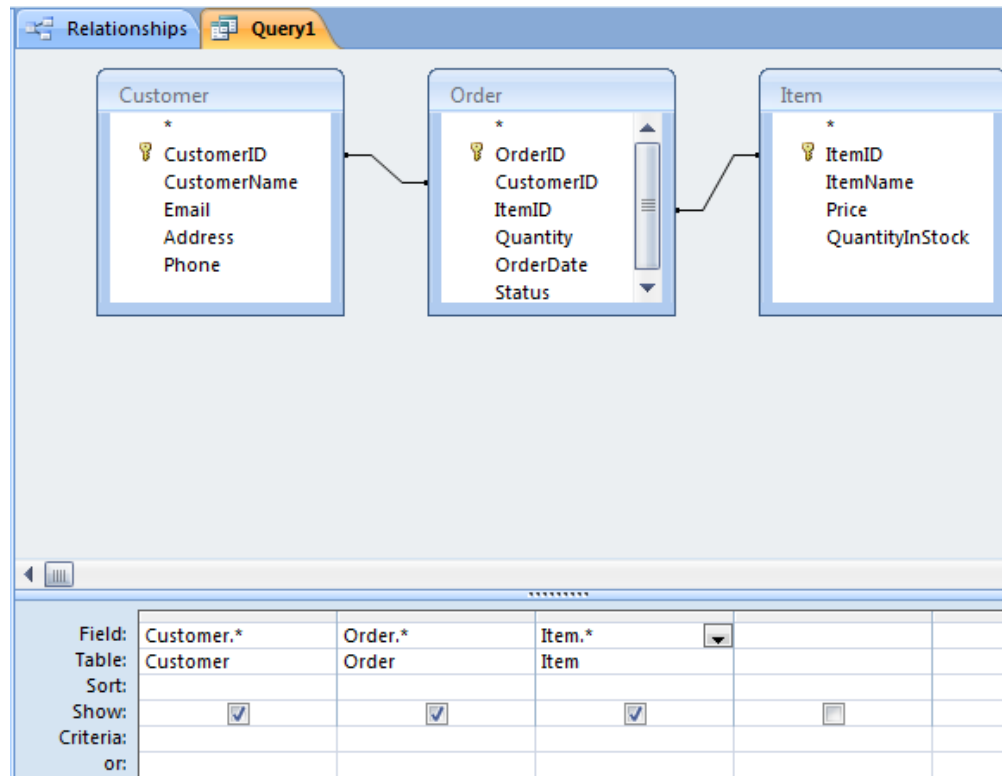


## Multi-Table Queries (2)

- Give me all the data across several tables:

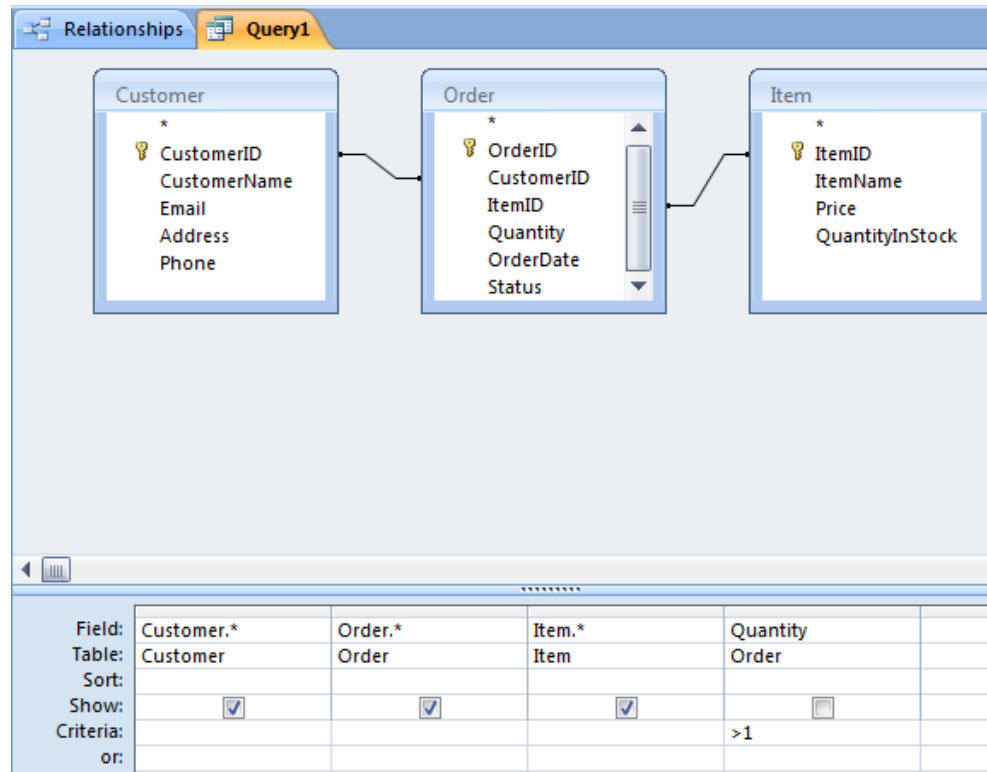
```
SELECT *  
FROM TableName1, TableName2 ... TableNameN  
WHERE (Table1.Pkey=Table2.Fkey) AND ... AND  
      (TableNameN-1.Pkey=TableNameN.Fkey)
```

# Multi-Table Queries (3)



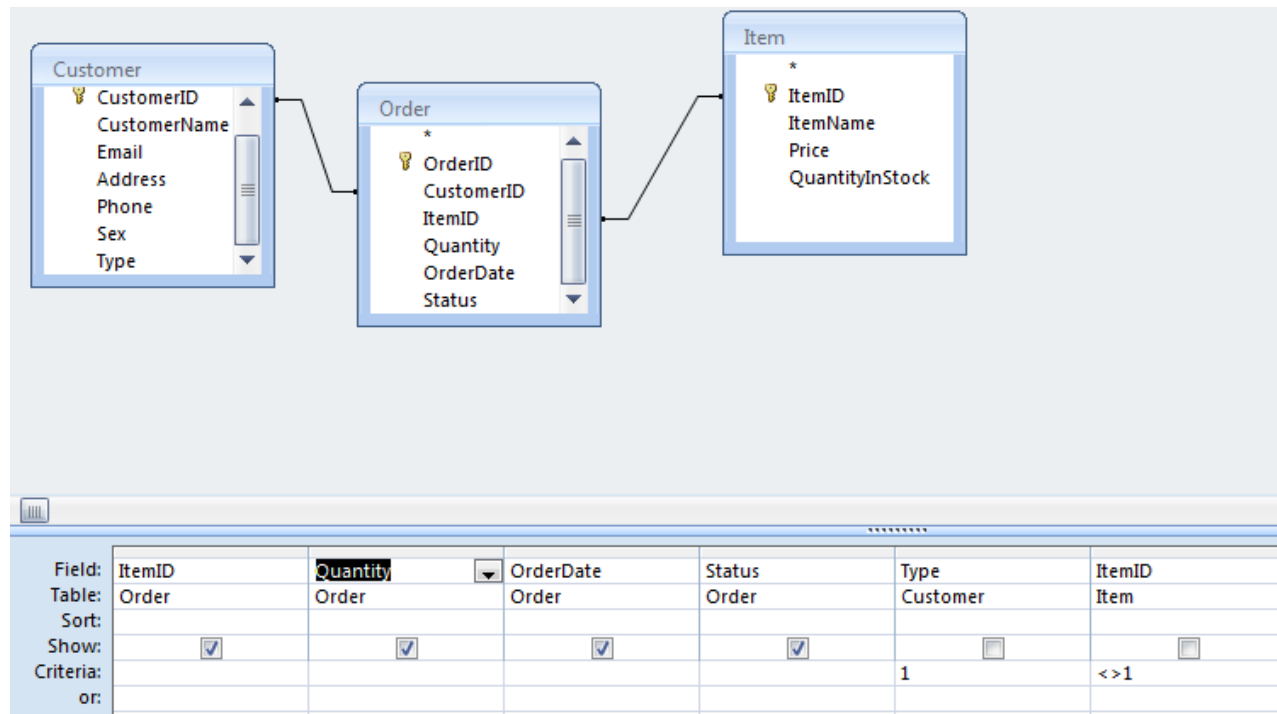
```
SELECT *  
FROM Customer, Order, Item  
WHERE Customer.CustomerID=Order.CustomerID AND Order.ItemID=Item.ItemID
```

# Multi-Table Queries (4)



```
SELECT *
FROM Customer, Order, Item
WHERE Customer.CustomerID=Order.CustomerID AND Order.ItemID=Item.ItemID AND
      Order.Quantity>1
```

# Multi-Table Queries (5)



```
SELECT Order.ItemID, Order.Quantity, Order.OrderDate, Order.Status
FROM Customer, Order, Item
WHERE Customer.CustomerID=Order.CustomerID AND Order.ItemID=Item.ItemID AND
Customer.Type=1 AND Item.ItemID<>1
```

# Custom Calculations

- A query can do a custom calculation on a set of fields and return the results:

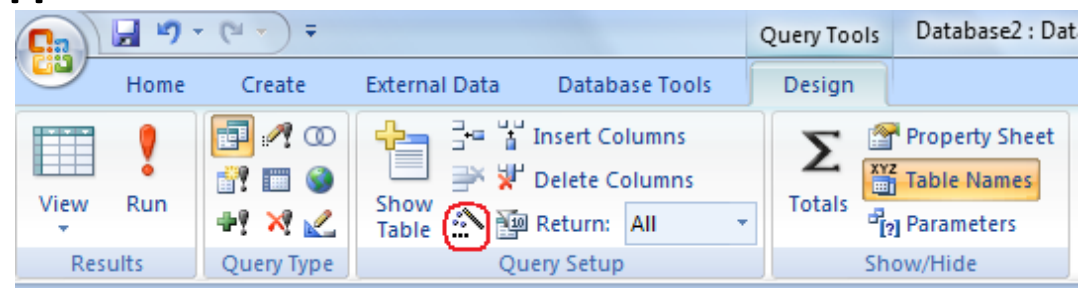
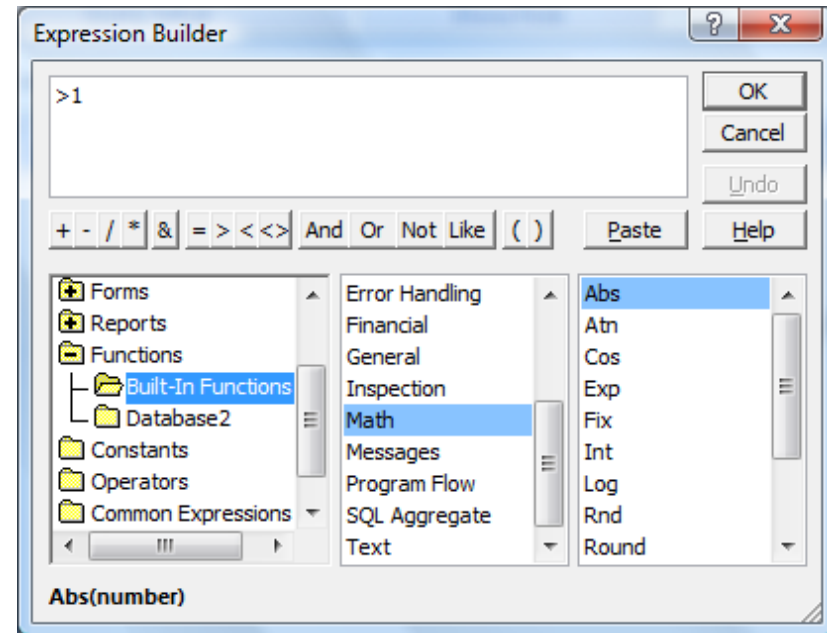
```
SELECT Order.OrderID, Order.Quantity*Item.Price AS  
    OrderTotal  
FROM Order, Item  
WHERE Order.ItemID=Item.ItemID
```

# Expression Builder

- Right click on a field and select **Build** from the context menu.

OR

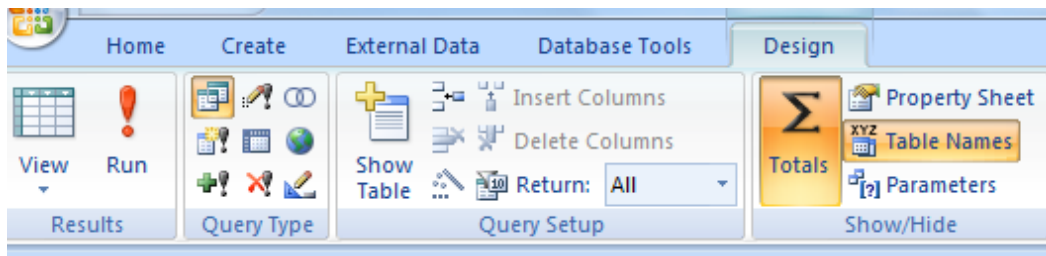
- Select Expression Builder icon from **Design** menu.



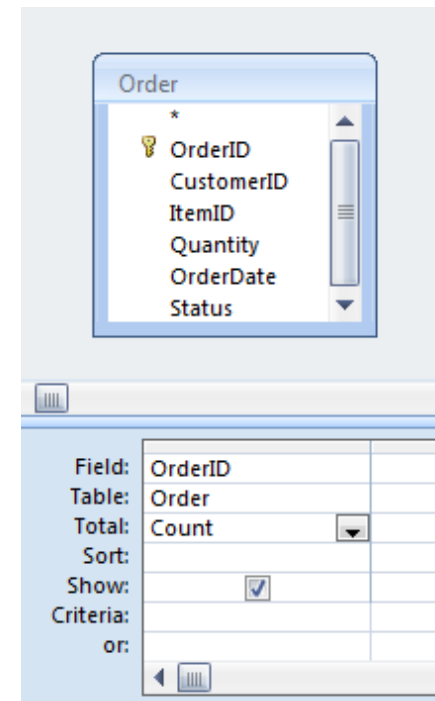


# Aggregate Queries

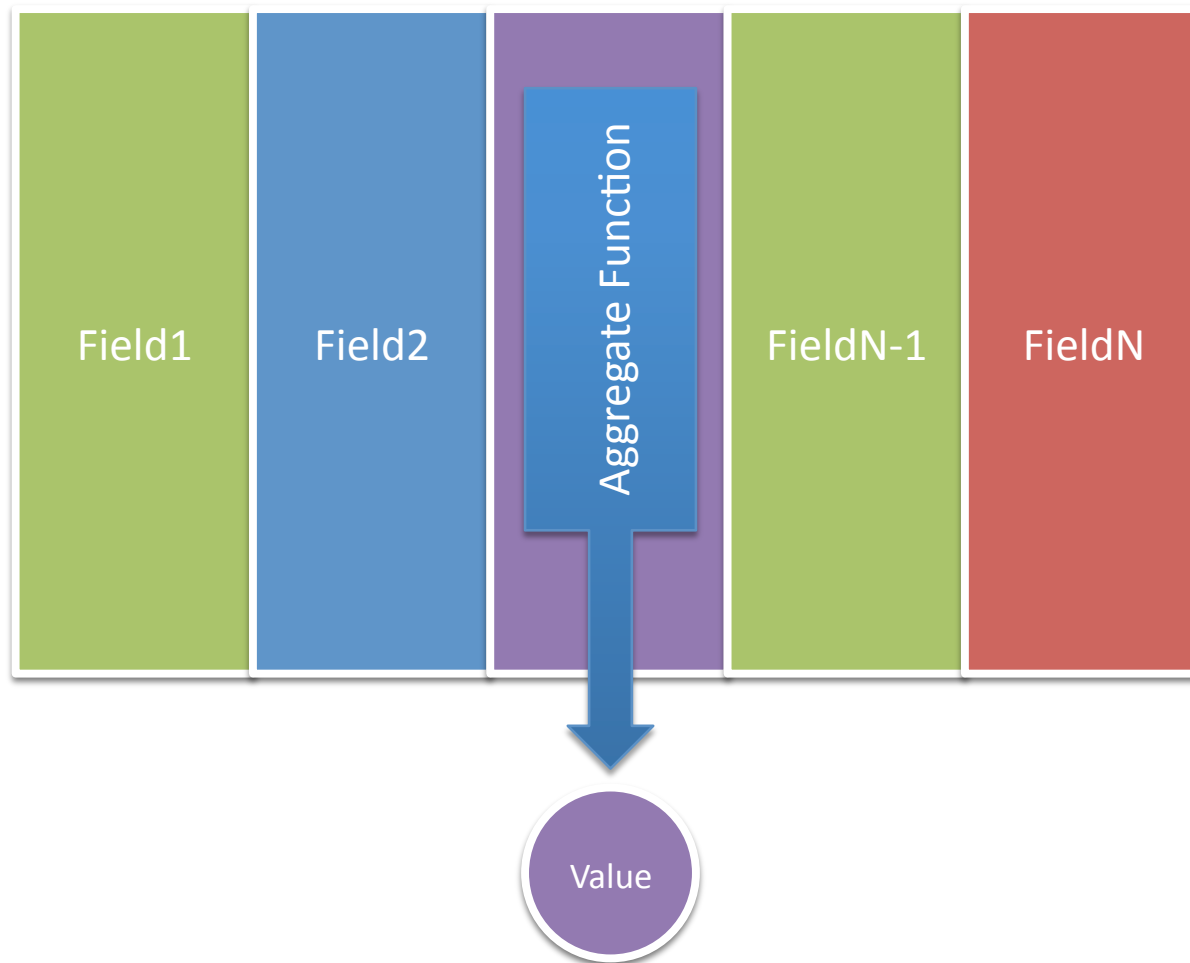
- Aggregate queries perform calculation on one or more fields.



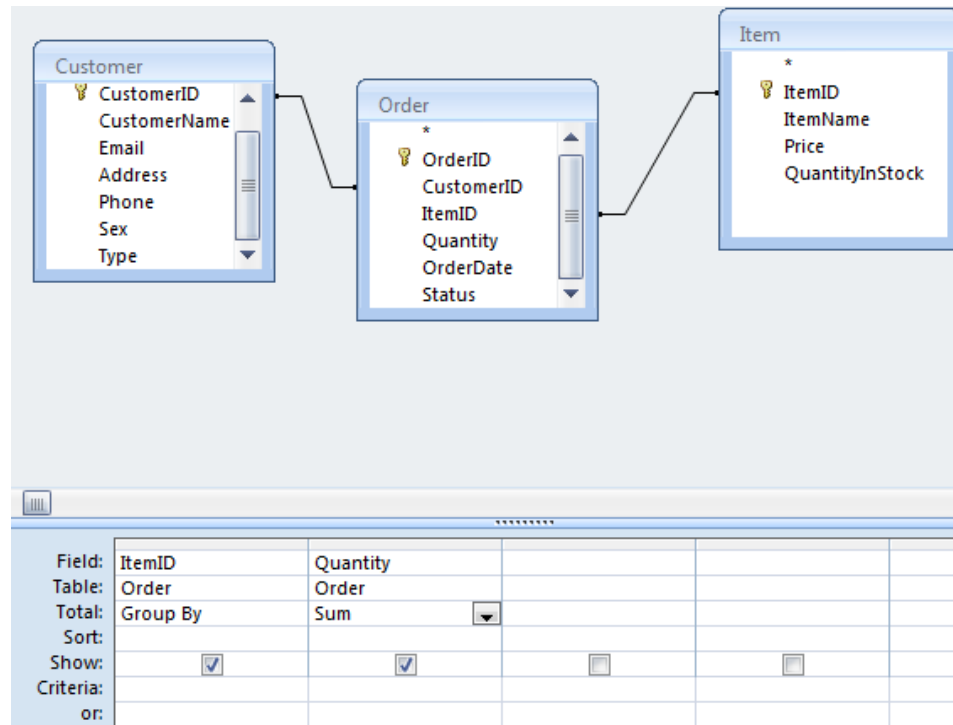
```
SELECT Count(Order.OrderID) AS CountOrderID  
FROM ORDER
```



# Aggregate Queries

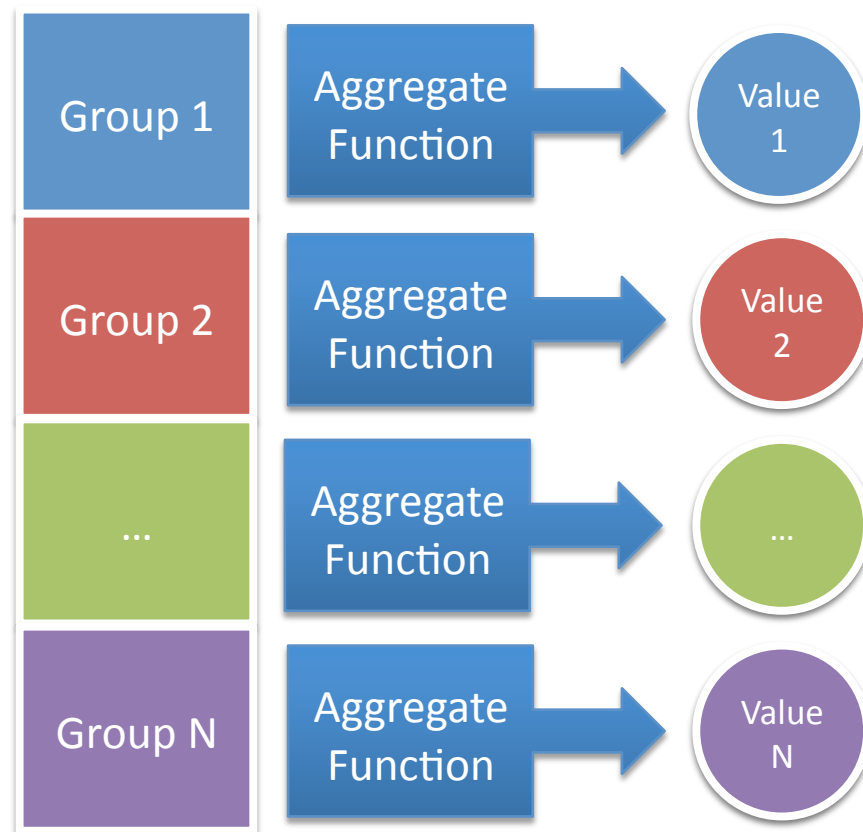


# Aggregate Queries with GROUP BY

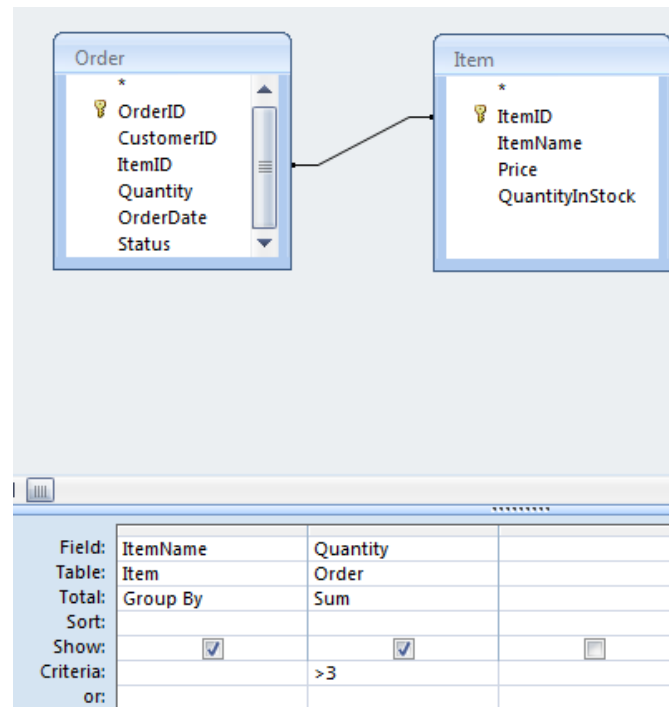


```
SELECT Order.ItemID, Sum(Order.Quantity) AS TotalQuantity
FROM Order, Item
WHERE Order.ItemID=Item.ItemID
GROUP BY Item.ItemID
```

# Aggregate Queries with GROUP BY



# Aggregate Queries with HAVING



```
SELECT Order.ItemName, Sum(Order.Quantity) AS TotalQuantity
FROM Order, Item
WHERE Order.ItemID=Item.ItemID
GROUP BY Item.ItemName
HAVING Sum(Order.Quantity)>3
```

# Aggregate Queries with HAVING

