

SQL Query Joins

ASP.Net 2.0 – Visual Studio 2005

CSE686 – Internet Programming

Instructor: James W. Fawcett

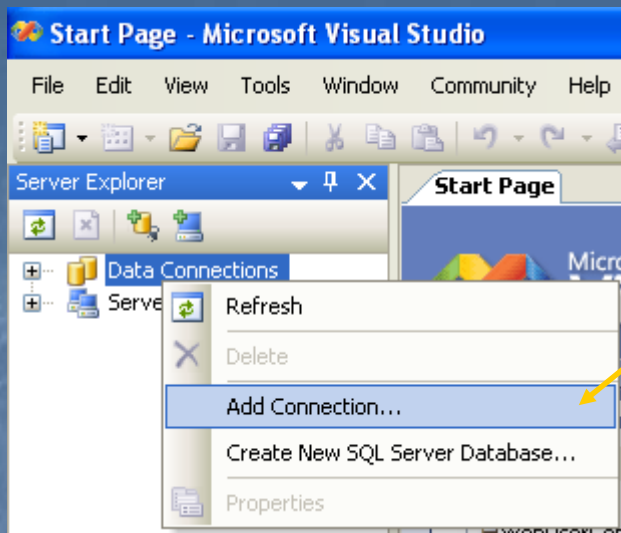
TA: Murat K. Gungor

Summer 2006

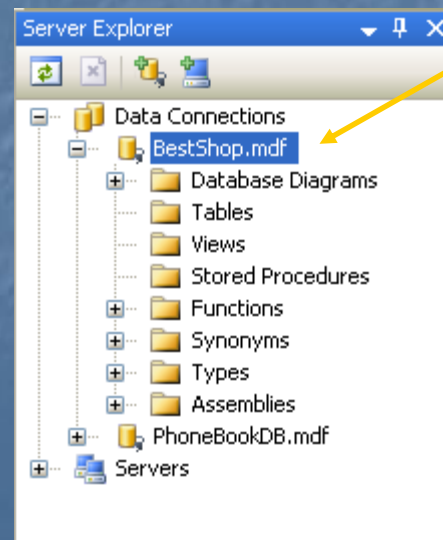
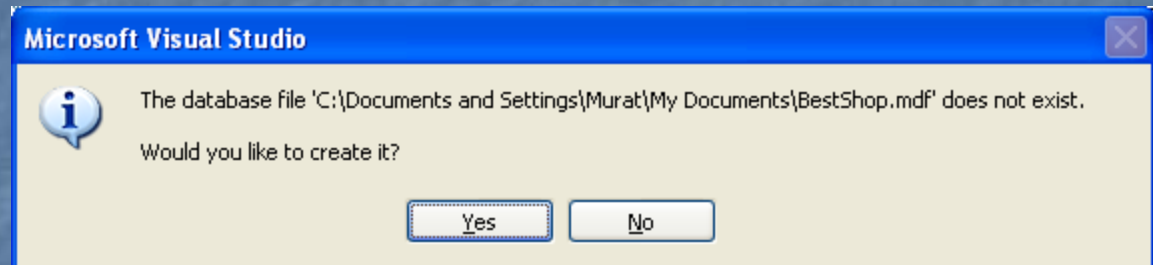
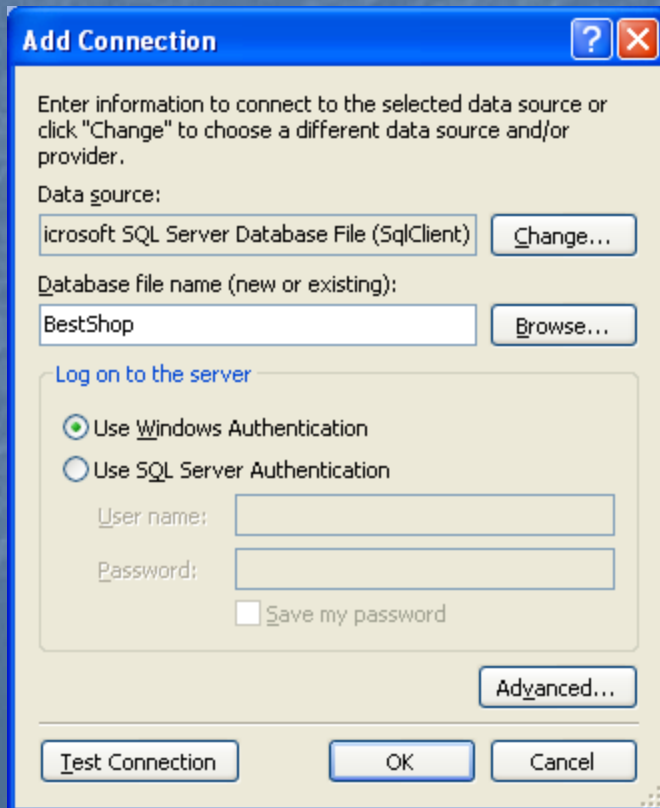
Query Joins

- Why do we need query joins?
 - View information from two or more separate database tables
- Type of query joins
 - Inner Join
 - Outer Join
 - Left Outer Join
 - Right Outer Join

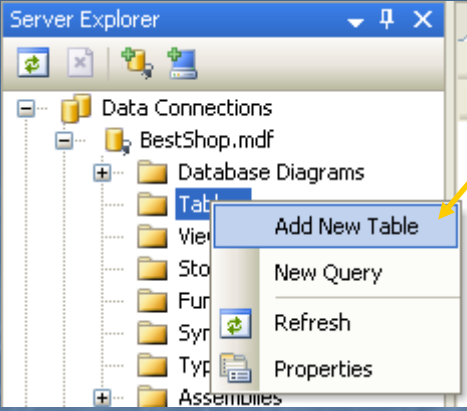
Add new SQL Database File



Let's Open Visual Studio .NET 2005 Using Sever Explorer Add Connection...



We just created BestShop.mdf (mdf = Master Database File)



Lets add two tables
Customers and Orders

dbo.Customer...BESTSHOP.MDF OrderTable: Q...BESTSHOP

Column Name	Data Type	Allow Nulls
CustomerID	numeric(18, 0)	<input type="checkbox"/>
Name	char(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Choose Name

Enter a name for the table:

OK Cancel

Column Properties

Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

CustomerID is Unique
and it is an Identity

dbo.OrderTabl...BESTSHOP.MDF dbo.Customer...BESTSHOP

Column Name	Data Type	Allow Nulls
ProductID	numeric(18, 0)	<input type="checkbox"/>
ProductName	char(50)	<input checked="" type="checkbox"/>
CustomerID	numeric(18, 0)	<input type="checkbox"/>

Choose Name

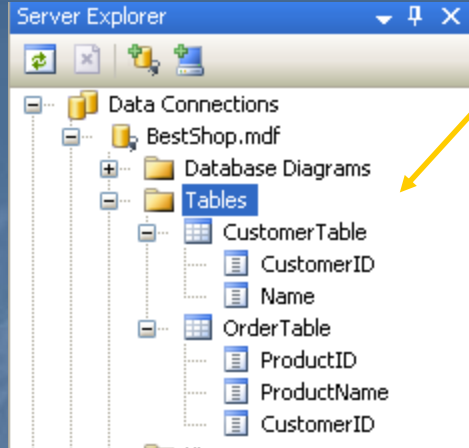
Enter a name for the table:

OK Cancel

Column Properties

Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1

ProducID is Unique and
it is an Identity



We have
CustomerTable and
OrderTable

OrderTable: Q...BESTSHOP.MDF) CustomerTable: Q...BESTSHOP.MDF)

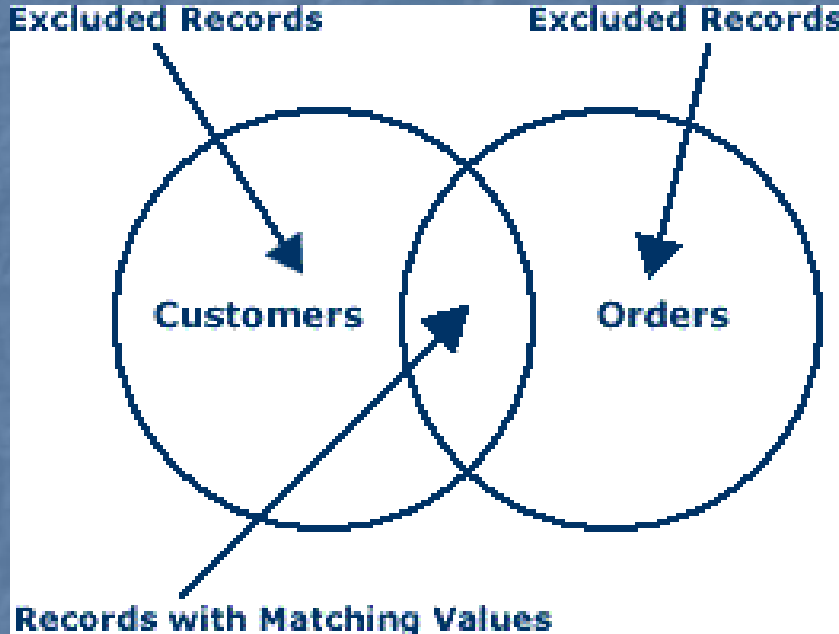
	CustomerID	Name	
	1	Dr Fawcett	...
	2	Murat	...
	3	George	...
	4	Jennifer	...
	5	Liza	...
▶*	NULL	NULL	

Lets add some data to
our tables

OrderTable: Q...BESTSHOP.MDF) CustomerTable: Q...BESTSHOP.MDF)

	ProductID	ProductName	CustomerID
	1	VCR	1
	2	Dell 700m	3
	3	Canon Camera	3
	4	Sony Vaio	5
▶*	NULL	NULL	NULL

Inner Join



- An **inner join** is a join that selects only those records from both database tables that have matching values.
- Records with values in the joined field that do not appear in both of the database tables will be ***excluded*** from the query.

```
SELECT CustomerTable.Name, OrderTable.ProductName  
FROM CustomerTable INNER JOIN OrderTable  
ON CustomerTable.CustomerID = OrderTable.CustomerID
```

Inner Join Query Result

Query Builder

CustomerTable

- * (All Columns)
- CustomerID
- Name

OrderTable

- * (All Columns)
- ProductID
- ProductName
- CustomerID

Column	Alias	Table	Output	Sort Type	Sc
Name		CustomerT...	<input checked="" type="checkbox"/>		
ProductName		OrderTable	<input checked="" type="checkbox"/>		

```
SELECT CustomerTable.Name, OrderTable.ProductName
FROM CustomerTable INNER JOIN
OrderTable ON CustomerTable.CustomerID = OrderTable.CustomerID
```

Name	ProductName
Dr Fawcett	VCR
George	Dell 700m
George	Canon Camera
Liza	Sony Vaio

1 of 4 | Cell is Read Only.

Execute Query | Validate Query | OK | Cancel

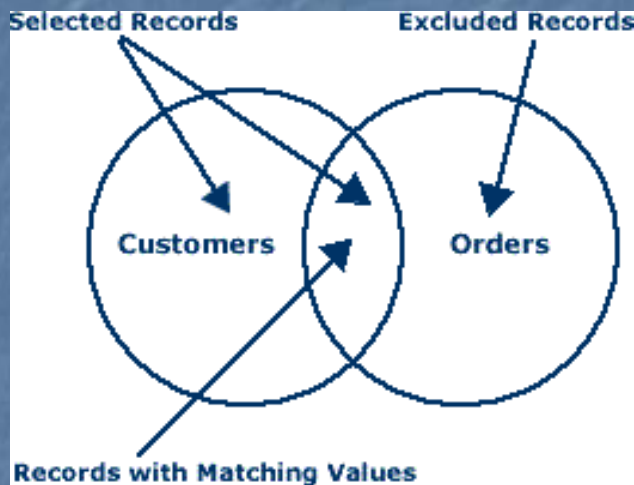
- The **INNER JOIN** returns all rows from both tables where there is a match.
- If there are rows in CustomerTable that do not have matches in OrderTable, those rows will **not** be listed.

We see ONLY matching values

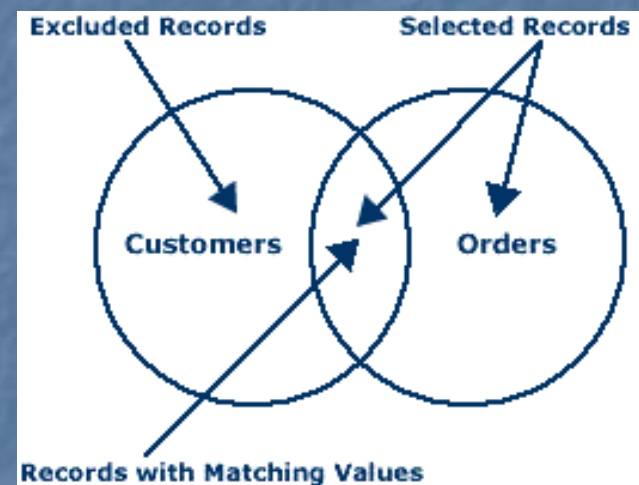
```
SELECT CustomerTable.Name, OrderTable.ProductName
FROM CustomerTable INNER JOIN OrderTable
ON CustomerTable.CustomerID = OrderTable.CustomerID
```

Outer Join

- An **outer join** selects all of the records from one database table and only those records in the second table that have matching values in the joined field.



In a **left outer join**, the selected records will include all of the records in the first database table.



In a **right outer join**, the selected records will include all records of the second database table.

SYNTAX

```
SELECT field1, field2, field3  
FROM first_table LEFT JOIN second_table  
ON first_table.keyfield = second_table.foreign_keyfield
```


SELECT CustomerTable.Name, OrderTable.ProductName
FROM CustomerTable **LEFT OUTER JOIN** OrderTable
ON CustomerTable.CustomerID = OrderTable.CustomerID

Query Builder

CustomerTable: * (All Columns), CustomerID, Name

OrderTable: * (All Columns), ProductID, ProductName, CustomerID

Column	Alias	Table	Output	Sort Type	Sc
Name		CustomerT...	<input checked="" type="checkbox"/>		
ProductName		OrderTable	<input checked="" type="checkbox"/>		

```

SELECT CustomerTable.Name, OrderTable.ProductName
FROM CustomerTable LEFT OUTER JOIN
OrderTable ON CustomerTable.CustomerID = OrderTable.CustomerID
    
```

Name	ProductName
Dr Fawcett ...	VCR ...
Murat ...	NULL
George ...	Dell 700m ...
George ...	Canon Camera ...
Jennifer ...	NULL
Liza ...	Sony Vaio ...

1 of 6 | Cell is Read Only.

Execute Query | Validate Query | OK | Cancel

NULL values also appears

- The **LEFT OUTER JOIN** returns all the rows from the first table (CustomerTable), even if there are no matches in the second table (OrderTable).
- If there are rows in CustomerTable that do not have matches in OrderTable, those rows **also** will be listed.

OrderTable: Q...BESTSHOP.MDF) Customer

CustomerID	Name
1	Dr Fawcett ...
2	Murat ...
3	George ...
4	Jennifer ...
5	Liza ...
**	NULL

SELECT CustomerTable.Name, OrderTable.ProductName
FROM CustomerTable **RIGHT OUTER JOIN** OrderTable
ON CustomerTable.CustomerID = OrderTable.CustomerID

The Query Builder window displays the following configuration:

- CustomerTable:** * (All Columns), CustomerID, Name
- OrderTable:** * (All Columns), ProductID, ProductName, CustomerID

The resulting SQL query is:

```
SELECT CustomerTable.Name, OrderTable.ProductName
FROM CustomerTable RIGHT OUTER JOIN
OrderTable ON CustomerTable.CustomerID = OrderTable.CustomerID
```

The results table shows the following data:

Name	ProductName
Dr Fawcett	VCR
George	Dell 700m
George	Canon Camera
Liza	Sony Vaio

- The **RIGHT OUTER JOIN** returns all the rows from the second table (OrderTable), even if there are no matches in the first table (CustomerTable).
- If there had been any rows in OrderTable that did not have matches in CustomerTable, those rows **also** would have been listed.

End of Presentation

■ Resources

- http://www.databasedev.co.uk/query_joins.html
- http://www.w3schools.com/sql/sql_join.asp
- <http://en.wikipedia.org/wiki/JOIN>
- http://www.databasejournal.com/sql/etc/article.php/26861_1402351_1