

ORACLE®



**ORACLE®**

## **Entity Framework and ORDT**

Jim Murray

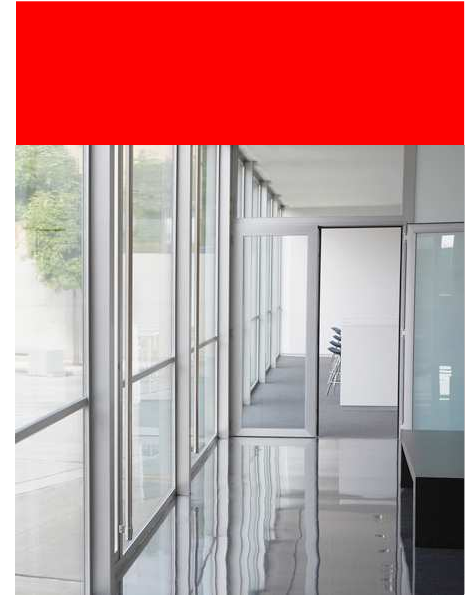
Oracle New England Development Centre



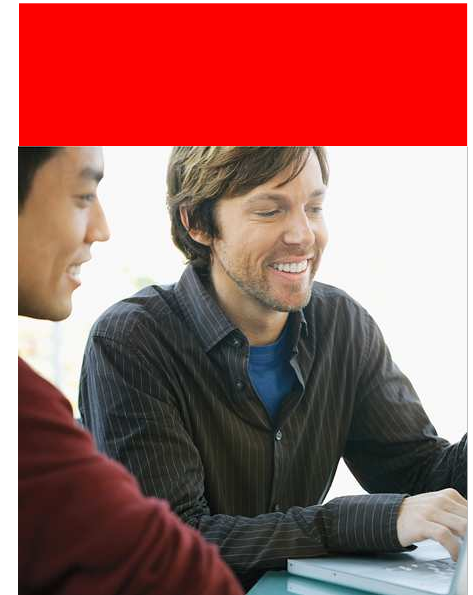


# Agenda

- Introduction to Entity Framework
- ORDT integration
- Some examples



# Introduction to Entity Framework





## Today's coding environment

- Object ?
- Relational ?
- New programmers
- New coding practices and styles
- New languages, compilers and IDEs
- New and demanding requirements from users
- More work less people to do it
- off-the-shelf versus roll-your-own



## Bridging the gap between OO and RELATIONAL

- OO been around for decades; relational databases even longer
  - Bridging the gap between has been time consuming and expensive:
    - Legacy ADO.NET
    - 3<sup>rd</sup> party frameworks



## What is wrong with Traditional coding?

- Writing good SQL queries is hard
- No help from compilers and/or IDE
- SQL is combination of STANDARD + “enhancements”
- Coding team may include more and more graduates trained in “new” coding styles rather than “traditional”



## Entity Framework

Entity Framework is Microsoft's attempt to address some of these issues.

- Decouple application from underlying data source
- Higher level of abstraction of the data
- Create and maintain data-oriented applications with less code than in traditional applications.
- More tools with greater “knowledge” of correct syntax, and semantics - “Intellisense”





## Entity Framework

**The ADO.NET Entity Framework provides an object-relational mapping solution and extends the capabilities of Microsoft's Language-Integrated Query (LINQ) technology by enabling developers to use LINQ against multiple vendors' databases.**



## *Introduction to LINQ*

- **L**anguage **I**Ntegrated **Q**uery
- Introduced in .NET Framework 3.5
- Aim: Unified syntax for data operations from
  - Collections of objects ( eg arrays, classes etc )
  - Databases (Relational and other data sources)
  - XML documents



## *Introduction to LINQ (cont)*

- Queries without LINQ

Object in code :

```
foreach (Employee e in employees)
    if ( e.State == "NH" )
```

*Relational:*

```
Select * from employees where state = 'NH'
```

*XPath/Xquery:*

```
//Employees/Employee[@State='NH']
```



## *Ado code without LINQ*

```
using (RdbConnection conn = new RdbConnection(conString))
{
    conn.Open();
    RdbCommand cmd = conn.CreateCommand();
    cmd.CommandText =
@"SELECT E.""Last_name"" , E.""First_name"" FROM
   ""Employees"" E WHERE E.""State"" = :state ";
    cmd.Parameters.AddWithValue(":state", "NH");
    using (RdbDataReader rdr = cmd.ExecuteReader()){
        while (rdr.Read()){
            .
            .
            .
        }
    }
}
```



## *Query with LINQ*

### **C#**

```
from e in employees  
where e.State == "NH"  
select e;
```

### **XLing**

```
from e in employees  
where e.State == "NH"  
select  
    . . . Some XML operations
```

## From a LINQ Perspective

Query Operators

C# 3.0

VB 9.0

### LINQ Providers

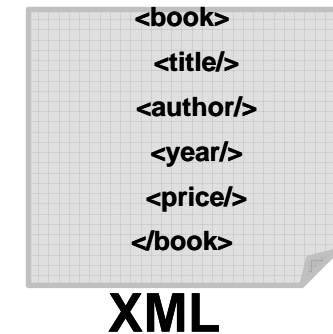
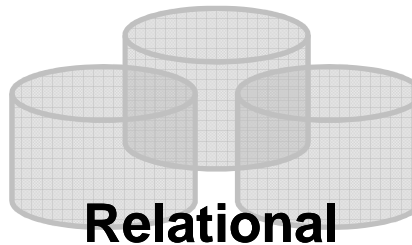
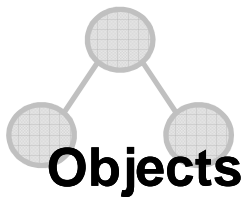
LINQ to  
Objects

LINQ to  
Datasets

LINQ to  
SQL

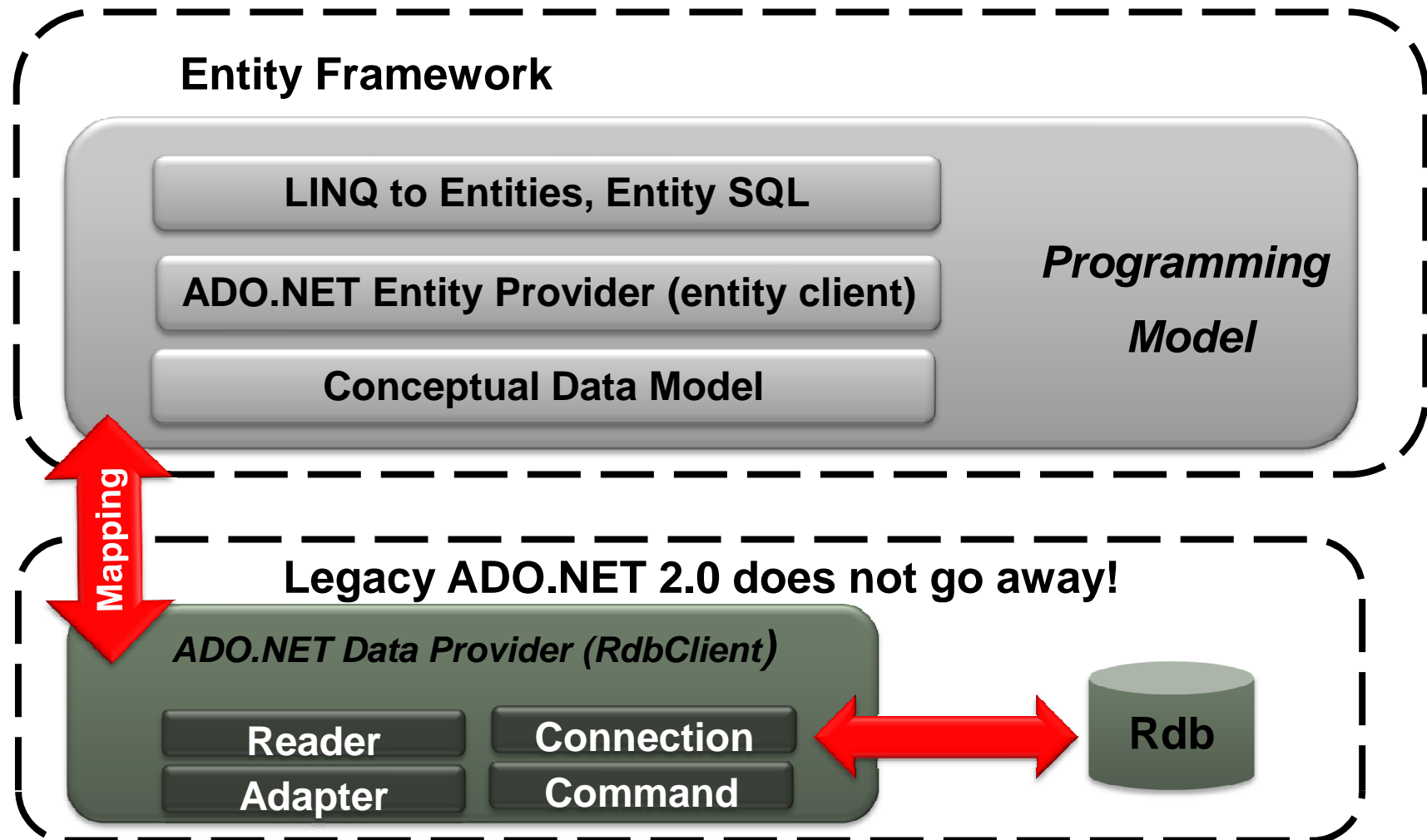
LINQ to  
Entities

LINQ to  
XML



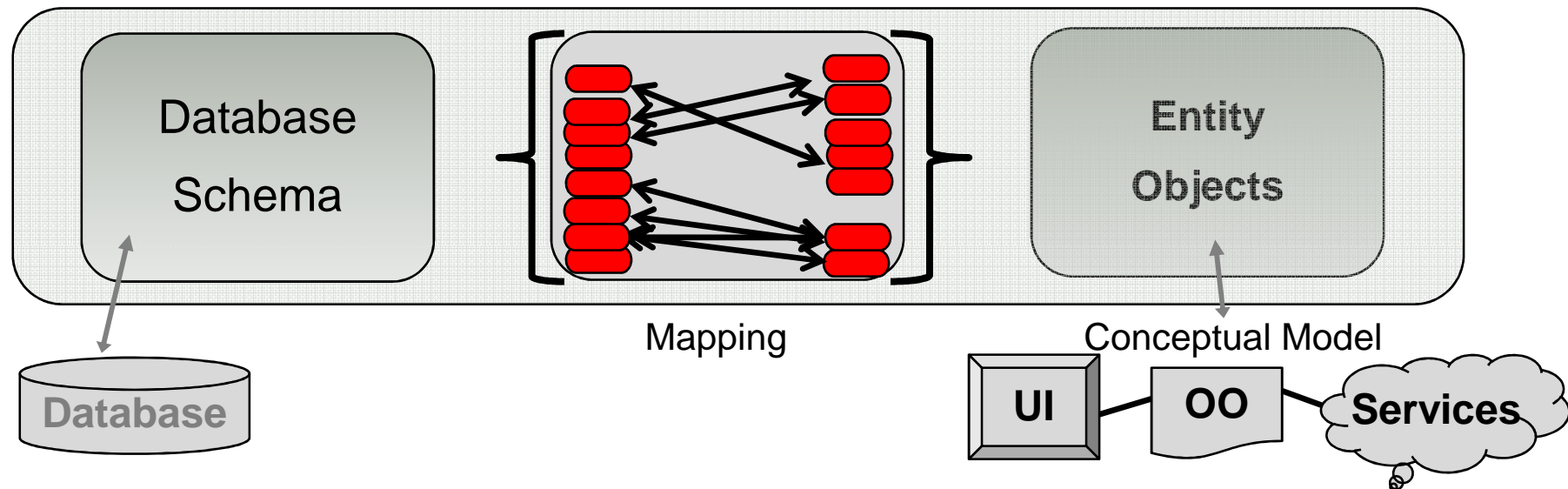
ORACLE

## From an ADO.NET Perspective



# Entity Data Model

- Set of objects that describe structure of your business data and map to your underlying data store



- Abstracts the developer from a *model* pleasing to a DBA (normalized, maintainable, efficient, secure), but *complicated* to program against



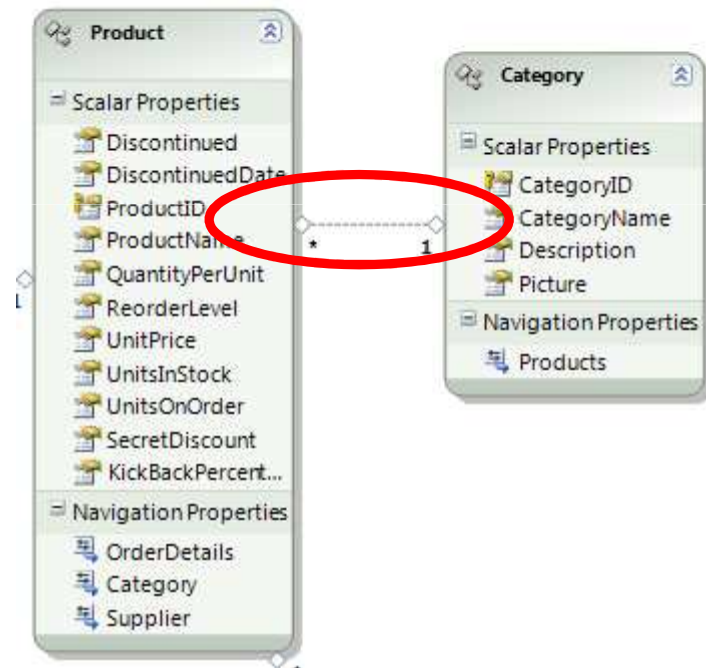


## ***Conceptual Model - mapping***

- ***Mapping***
  - With *mapping wizard*, create a new Entity Data Model
  - Examine entities, scalar and navigation properties in *EDM Designer*,
  - Look at the *Model Browser*
  - Look at the *Mapping Details*
  - Look at the *EDMX file*
  - Most importantly, look at *generated entity classes*
- End up with *object-representation* of our data store

## Associations/Navigation Properties

- When generating from database, EF infers <PK><FK> relationships and generates navigation properties
  - Multiplicity
  - Direction
- *Navigate* conceptual model without *explicit joins*
- Navigation properties, enable us to '*dot*' into related objects





## Approaches

- Database First
  - Build Model from tables and relationships existing already in database
- Model First
  - Build objects and relationship using wizard or entity editors
  - Use this model to Create DDL specifically for the underlying database system

### Code First ( EF 4.1 onwards)

- Create standard classes “plain old CLR objects” (POCO)
- Not supported by ORDT yet.



## ORDT Integration

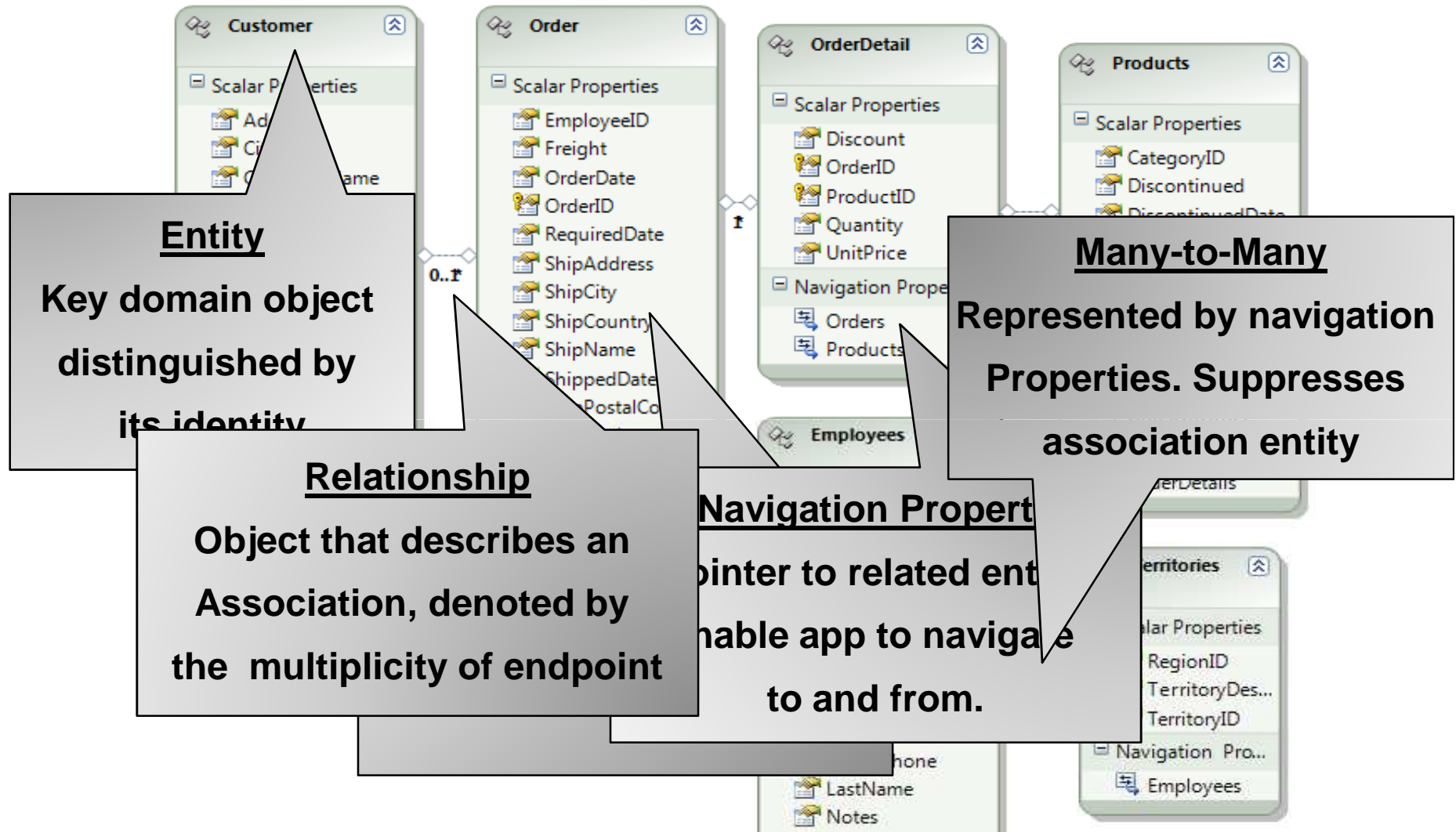
- Entity Framework Provider installed as part of ORDT
- Automatically installed when Visual Studio integration is chosen
- Visual Studio 2008 (SP1)
  - .NET Framework 3.5 Support
  - Data access and retrieval
  - Build Model from Database
- Visual Studio 2010
  - .NET Framework 4 Support
  - Additional DML support
  - Build Database from Model



## Some Examples

- Northwind database ( standard Microsoft demo database for Entity Framework)
- Compare code
  - standard ADO.NET code
  - LING code
- Visual Studio 2008 and Entity Framework

# Northwind Entities





## **Compare Code ADO.NET code :**

```
using (RdbConnection conn = new RdbConnection(conString))
{
    conn.Open();
    RdbCommand cmd = conn.CreateCommand();
    cmd.CommandText =
@"SELECT ""ShipCountry"" , O.""CustomerID"" , O.""ShipCity"" FROM
   ""Orders"" O, ""Customers"" C WHERE O.""CustomerID"" = C.""CustomerID"" AND
   :id = O.""CustomerID"" AND :country = C.""Country"" ";
    cmd.Parameters.AddWithValue(":id", "BLONP");
    cmd.Parameters.AddWithValue(":country", "France");
    using (RdbDataReader rdr = cmd.ExecuteReader()){
        while (rdr.Read()){
            string country = rdr.GetString(0);
            string city = rdr.GetString(1);
            System.Console.WriteLine("country " + country + " city " + city );
        }
    }
}
```



## ***Compare Code ADO.NET code :***

```
using (RdbConnection conn = new RdbConnection(conString))
{
    conn.Open();
    RdbCommand cmd = conn.CreateCommand();
    cmd.CommandText =
@"SELECT ""ShipCountry"" , O.""CustomerID"" , O.""ShipCity"" FROM
   ""Orders"" O, ""Customers"" C WHERE O.""CustomerID"" = C.""CustomerID"" AND
   :id = O.""CustomerID"" AND :country = C.""Country"" ";
    cmd.Parameters.AddWithValue(":id", "BLONP");
    cmd.Parameters.AddWithValue(":country", "France");
    using (RdbDataReader rdr = cmd.ExecuteReader()){
        while (rdr.Read()){
            string country = rdr.GetString(0);
            string city = rdr.GetString(1);
            System.Console.WriteLine("country " + country + " city " + city );
        }
    }
}
```





## ***Compare Code LING code :***

```
String id = "BLONP";  
String country = "France";
```

```
Entities1 context = new Entities1();  
var orders = from o in context.Orders  
              where o.Customers.CustomerID == id &&  
                  o.Customers.Country == country  
              select o;  
foreach (Orders o in orders)  
    System.Console.WriteLine("country " + o.ShipCountry + " city " + o.ShipCity);
```

**Demo1 - Microsoft Visual Studio**

File Edit View Refactor Project Build Debug Data Tools .NET Reflector Test Window Help

Debug Any CPU conso

Orders : Query(lo.....\northwind) Start Page RdbCommand [from metadata] Model1.Designer.cs

CustomerID	EmployeeID	Freight	OrderDate	OrderID	RequiredDate	ShipAddress	ShipCity	ShipCountry
BERGS	8	92.69	12/08/1996 12:...	10278	9/09/1996 12:0...	Berguvsvägen 8	Luleå	Sweden
BLOMP	2	55.28	25/07/1996 12:...	10265	22/08/1996 12:...	24, place Kléber	Strasbourg	France
CHOPB	5	22.98	11/07/1996 12:...	10254	8/08/1996 12:0...	Hauptstr. 31	Bern	Switzerland
ERNSH	9	146.06	23/07/1996 12:...	10263	20/08/1996 12:...	Kirchgasse 6	Graz	Austria
ERNSH	1	140.51	17/07/1996 12:...	10258	14/08/1996 12:...	Kirchgasse 6	Graz	Austria
FOLKO	6	3.67	24/07/1996 12:...	10264	21/08/1996 12:...	Östergatan 24	Brännå	Sweden

1 of 27

**Program.cs\***

```

Demo1.Program
Main(string[] args)
{
    String id = "BLOMP";
    String country = "France";

    Entities1 context = new Entities1();
    var orders = from o in context.Orders
        where o.Customers.CustomerID == id &&
              o.Customers.Country == country
        select o;
    foreach (Orders o in orders)
        System.Console.WriteLine("country " + o.ShipCountry + " city " + o.ShipCity);
}

```

**Model1.edmx\***

Customers Orders OrderDetails

Scalar Properties

- Customers: CustomerID, CompanyName, ContactName, ContactTitle, Address, City, Region, PostalCode, Country
- Orders: OrderID, EmployeeID, OrderDate, RequiredDate, ShippedDate, ShipVia, Freight, ShipName, ShipAddress, ShipCity, ShipRegion
- OrderDetails: OrderID, ProductID, UnitPrice, Quantity, Discount

Navigation Properties

- Customers: Orders (0..1 to \*)
- Orders: OrderDetails (1 to \*)

**Server Explorer**

localhost:1701E:\regtest\northwind

- Data Connections
- Tables
  - Categories
  - Customers
  - Employees
  - EmployeesTerritories
  - InternationalOrders
  - OrderDetails
  - Orders
    - CustomerID
    - EmployeeID
    - Freight
    - OrderDate
    - OrderID
    - RequiredDate
    - ShipAddress
    - ShipCity
    - ShipCountry
    - ShipName
    - ShippedDate
    - ShipPostalCode
    - ShipRegion
    - ShipVia
  - PreviousEmployees
  - Products
  - Regions
  - Shippers
  - Suppliers
  - Territories
- Views
- Domains
- Synonyms
- Sequences
- Procedures
- Functions

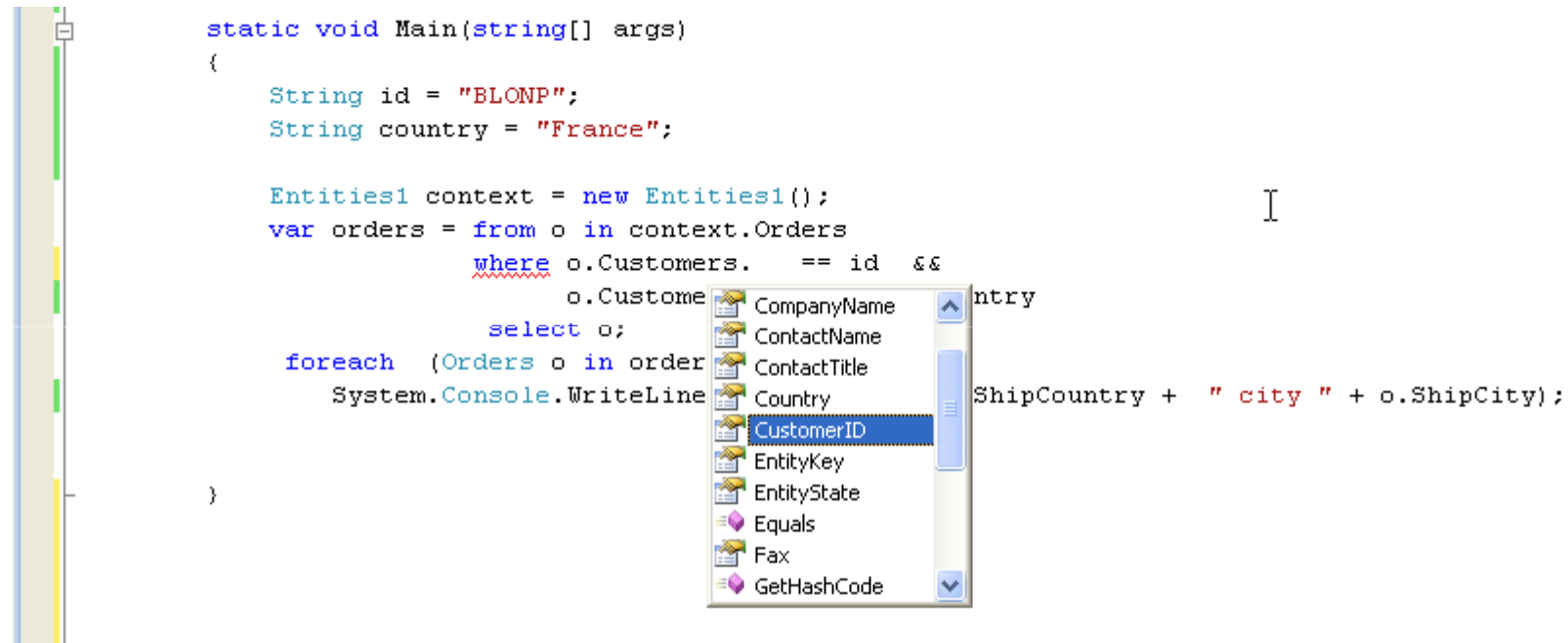
**Properties**

Output: Show output from: Constraint Validation

Error List Output Find Results 1 Find Symbol Results Test Results

Ready Ln 20 Col 49 Ch 49 INS


# Intellisense





## For More Information

search.oracle.com



OR

- [www.oracle.com/rdb](http://www.oracle.com/rdb)
- <http://msdn.microsoft.com/en-us/library/bb399567.aspx>
- [support.oracle.com](http://support.oracle.com)
- [Jim.murray@oracle.com](mailto:Jim.murray@oracle.com)



# Q U E S T I O N S & A N S W E R S

**SOFTWARE. HARDWARE. COMPLETE.**

ORACLE®