ICT365

Software Development Frameworks

Dr Afaq Shah

Murdoch
UNIVERSITY

ADO.net

Murdoch
UNIVERSITY

Learning objectives

Murdoch UNIVERSITY

- To be able to program the key components of a C#/.NET application for
  - Establishing a database connection
  - Defining a data model for its tables
  - Querying the database

Topics

Murdoch UNIVERSITY

Important .NET Class libraries: Windows Forms, ADO.NET, ASP.NET, XML and Web Services

ADO.net

- Establishing a database connection
- Defining a data model for its tables
- Querying the database

Framework Class Libraries

Much of the power of .NET comes from its standard class libraries - Framework Class Libraries, eg:

- mscorlib.dll
- system.dll
- system.data.dll (System.Data namespace)
- system.drawing.dll (System.Drawing namespace)
- system.web.dll (System.Web namespace)
- system.web.services.dll (System.Web.Services namespace)
- system.windows.forms.dll (System.Windows.Forms namespace)
- system.xml.dll (System.Xml namespace)

E.g.

Murdoch UNIVERSITY

ADO.NET:

- a set of classes and tools for creating data base applications.
- can connect to databases -> SQL Server, Oracle and Microsoft Access

ASP.NET:

- a set of classes and tools for creating web applications.

Web Services

- are internet based applications that use XML messages (**SOAP** messages) for communications.

Introduction to ADO.NET

Murdoch UNIVERSITY

Active Data Objects

- A set of class definitions included in the .NET Framework
- Objects collaborate to provide .NET applications (relatively) easy access to databases.
- Designed for *scalability*.

Oracle

Murdoch UNIVERSITY

- Probably we will use this instead of MS SQLServer
- Later we may look at Azure, a cloud-based solution, but it’s a little complicated getting set up, so instead, We will use:
- Oracle.
- Following slides cover MySQL, but its more or less the same for Oracle

MySQL/Oracle and .NET

Murdoch UNIVERSITY

- Principles are the same regardless of the database solution
- Get the infrastructure (DBMS and its tools)
- Set up a database and get its server running
- Establish interoperability of the database and the .NET environment

.NET is proprietary to Microsoft

Add to your program a ‘connection’

This specifies:

the running DBMS service,

database (set of tables, views etc.)

your ‘login’ as an authorised user on that database

Specifics: setting up a database

Murdoch UNIVERSITY

- Get MySQL running

  Install XAMPP to get MYSQL, Apache, XAMPP Control Panel and PHPMyAdmin

  Then just start Apache* and MySQL in XAMPP Control Panel

  Use PHPMyAdmin (or MySQL console) to build a database (e.g., 'employee') with an employee table, and also to add a user ("Tim") and give that user permissions on the database

* Apache only needed for PHPMyAdmin - .NET will use Internet Information Server (IIS) Express to serve the applications Web pages

Specifics: interoperability

Murdoch UNIVERSITY

Install MySQL Connector Net on the machine

An ADO.NET driver for MySQL

Install 'Packages' on the .NET application

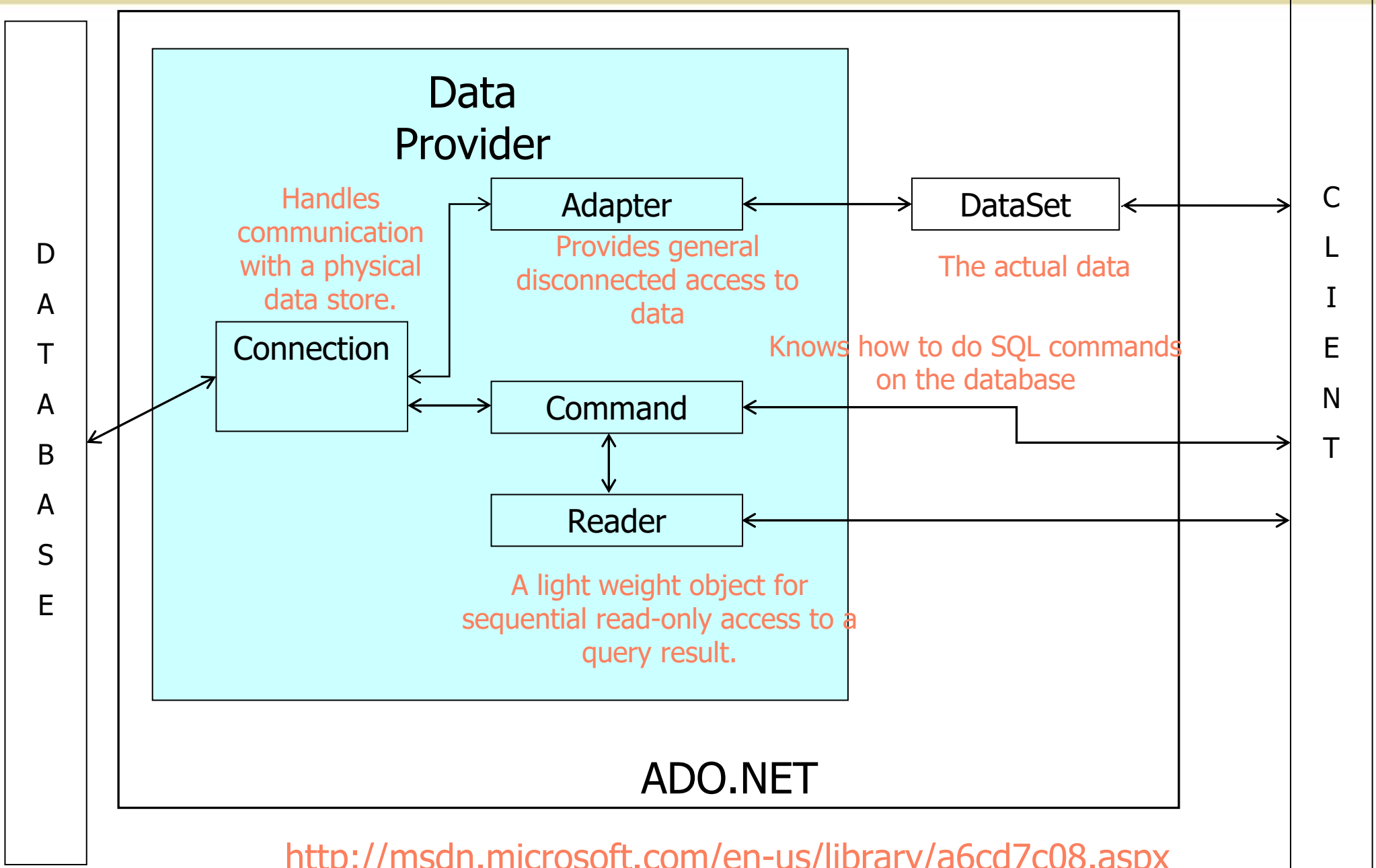
Entity Framework (EF)

MySQL

MySql.Data, MySql.Data.Entities and (for a Web project) MySql.Web

- Now your C#/.NET application is ready to talk to a MySQL database through ADO.NET (and extended with EF)

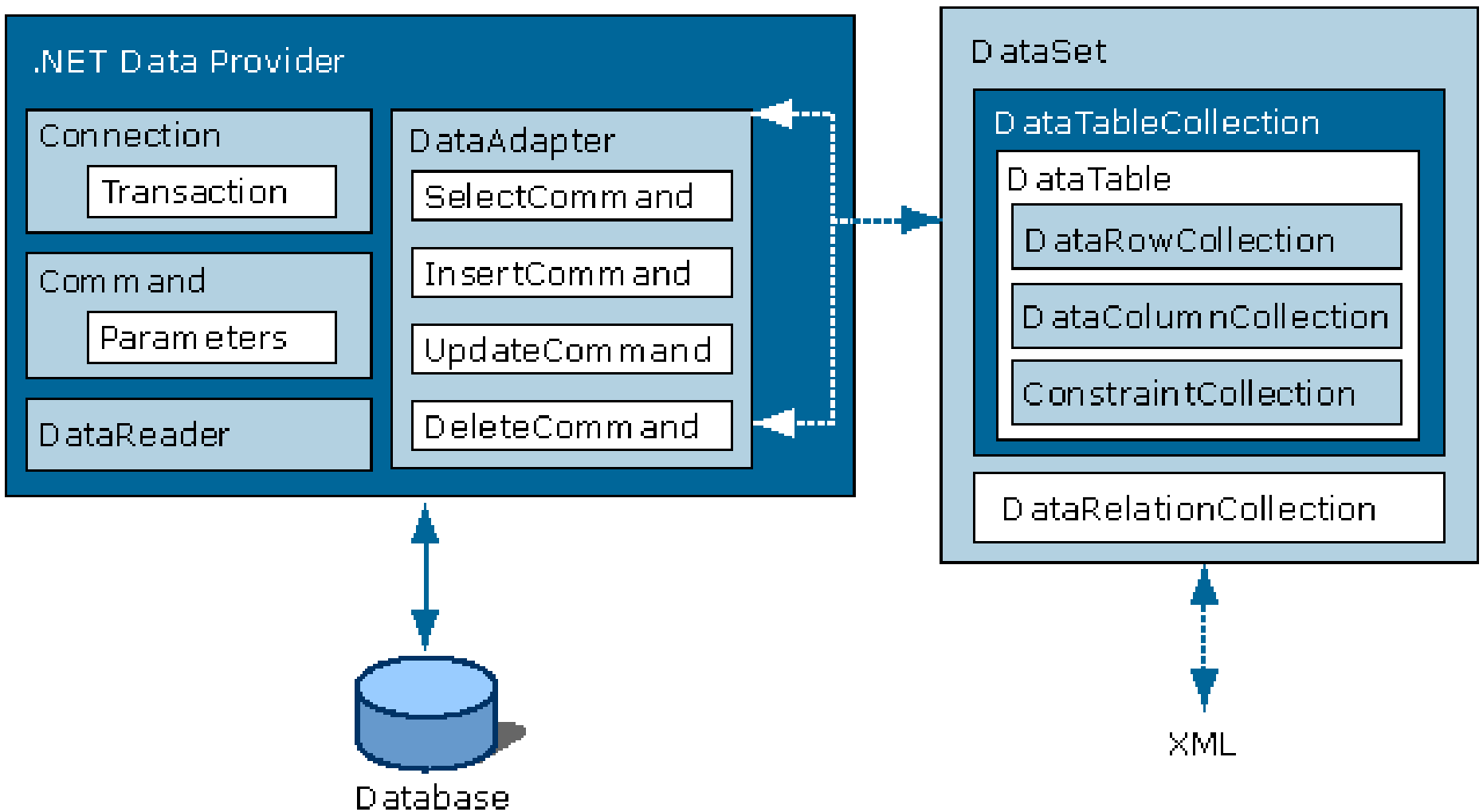
The ADO.NET Object Model



http://msdn.microsoft.com/en-us/library/a6cd7c08.aspx

ADO.NET Architecture Diagram

Murdoch UNIVERSITY



ADO.NET Namespaces

Murdoch UNIVERSITY

System.data	Core namespace, defines types that represent data
System.Data.Common	Types shared between managed providers
System.Data.OleDb	Types that allow connection to OLE DB compliant data sources
System.Data.SqlClient	Types that are optimized to connect to Microsoft® SQL Server
System.Data.SqlTypes	Native data types in Microsoft® SQL Server

Connection String

Murdoch UNIVERSITY

MS SQL Server Examples

```
connStr = "server=scorpio.edu.au; User ID=gnulu;
                Password=xxxxxxx"
```

```
connStr = "server=mssql99.blah.net; database=DB_999;
                User ID=kangaroo; Password=xxxxx"
```

```
connStr = "server=(local)\\VSDOTNET;
        database=Bulk_Mail_Addresses;
        Trusted_Connection=yes";
```

Data Provider Objects

Murdoch UNIVERSITY

- Command Object

Knows how to execute a SQL command on a server

- Properties:

CommandText

A SQL statement to be executed at the data source.

SqlCommand1.CommandText = "SELECT * FROM Address_List";

Can be changed by the program.

- Methods

ExecuteReader

ExecuteScalar

ExecuteNonquery

Data Provider Objects

Murdoch UNIVERSITY

- DataReader

- A fast, low-overhead object, similar to a StreamReader for file input.
- Provides forward-only, read-only stream of data from a data source
- Created by calling the *ExecuteReader* method of a Command object

  Never with “new”

- Connection must remain open while the DataReader is used.

Data Provider Objects

Murdoch UNIVERSITY

- There are two kinds of “Adapters”

Data Adapter

- Present in ADO 1.0
- Continued in ADO 2.1

Table Adapter

- New in ADO 2.0

DataAdapter

Murdoch UNIVERSITY

- Provides access to a collection of data from a data source

  Permits client to close connection while processing the data.

  Effectively provides a local cache

  Client accesses the cache rather than the actual database.

- Actual database can be updated when desired.

Data Provider Objects

Murdoch UNIVERSITY

- DataAdapter

Contains four Command Objects:

SelectCommand

UpdateCommand

InsertCommand

DeleteCommand

Uses SelectCommand to fill a DataSet

Uses other command objects to transmit changes back to the data source

Table Adapter

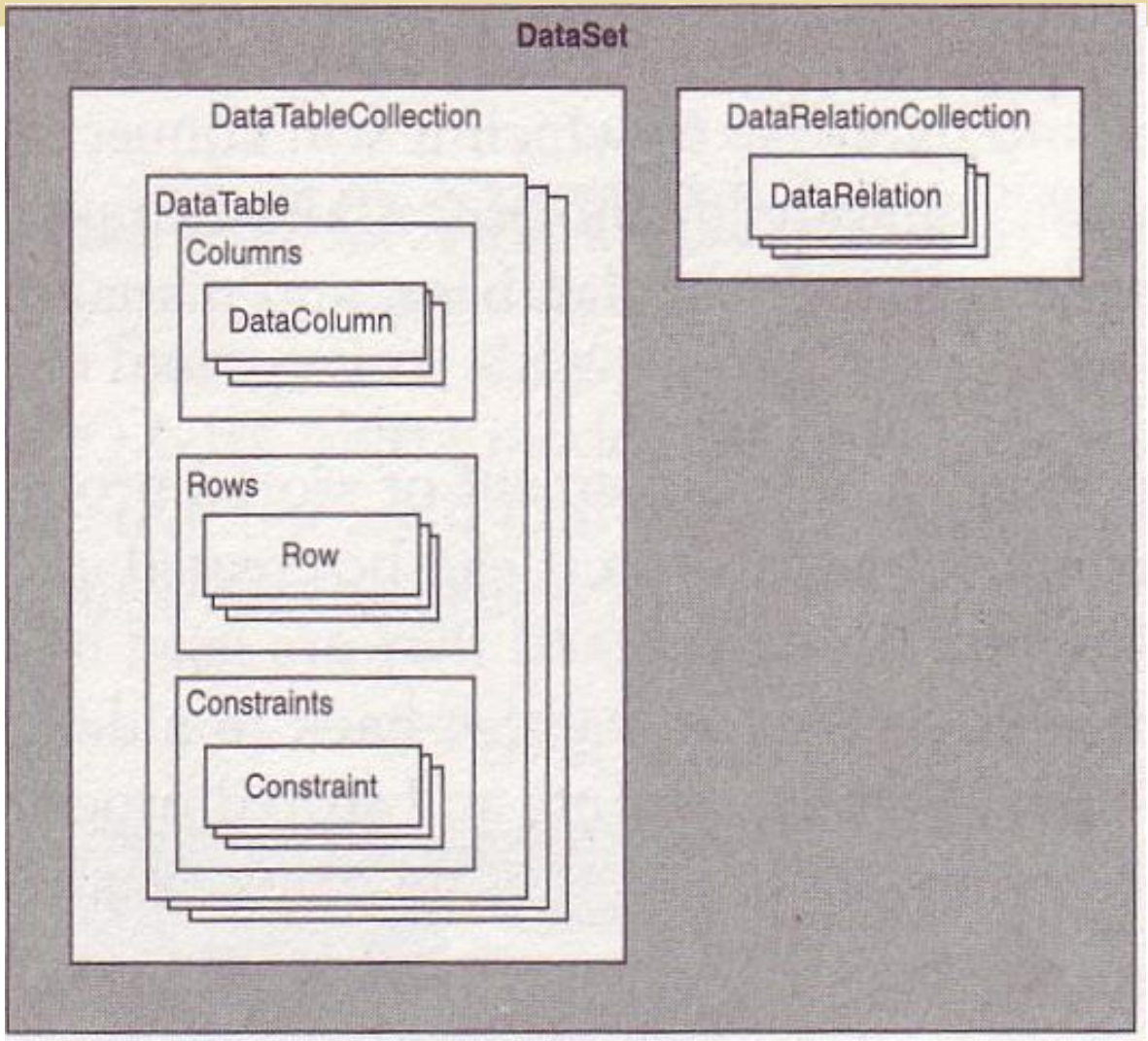
Murdoch UNIVERSITY

- A Table Adapter component fills a dataset with data from the database.
- Table Adapters can also perform adds, updates, and deletes on the database.
- Includes a DataAdapter and a Connection object.
- Improves functionality and ease of use of the original DataAdapter.

DataSet

Murdoch UNIVERSITY

- In-memory copy of data
- No connection to a database.
- Simple form of relational database
  - collection of *tables*
  - collection of *DataRelations*



From *ADO.NET 2.0 Step by Step*

DataSet

Murdoch UNIVERSITY

- The DataTable

  Columns

    Like the column defintions from a SQL CREATE TABLE statement

    Name, Data Type, Max Length, Allows Nulls

  Rows

    The data

  Constraints

    Foreign Key

    Unique

DataSet

Murdoch UNIVERSITY

- Data Relation

Programmatic interface for navigating from a master row in one table to related rows in another.

Does not enforce referential integrity

A *foreign key constraint* does that.

DataSet

Murdoch UNIVERSITY

- A DataSet object can exist independently of any database.
- We will only use DataSets to hold data retrieved from a database.

```
...
using MySql.Data.MySqlClient;
...
      static void Main(string[] args)
      {
string cs =
@"server=100.222.99.999;uid=gil;pwd=ICT365;database=Session3";
       MySqlConnection conn = null;
       try  {
          conn = new MySqlConnection(cs);
          conn.Open();
          Console.WriteLine("MySQL version : {0}", conn.ServerVersion);
      MySqlCommand cmd =
        new MySqlCommand("select * from employee",conn);
             MySqlDataReader reader=cmd.ExecuteReader();
          reader.Read();
          Console.WriteLine(reader.GetString(1));
             }
        catch (MySqlException ex) {

           Console.WriteLine("Error: {0}", ex.ToString());
    }
        finally {
          if (conn != null) conn.Close();
    }
...
```

A connection needs to specify server, user, password and database

Sending text of SQL to the DBMS, then reading one line and printing content of its 2nd field

Closing the database connection when we're done, managed in try… catch… finally

ADO.NET and Entity Framework

Murdoch UNIVERSITY

- A set of components to access data in the .NET framework

  Evolved from 'ActiveX Data Objects' (ADO) (at least that's the origin of its name)

- Entity Framework (EF) is an open-source object-relational mapping for ADO.NET

  - Allows us to have a 'model' in our application

  - The power of OO programming applied to our database logic

EF object-relational mapping

Murdoch UNIVERSITY

- A **database** is mapped to a class
- Each table is a distinct class that inherits from DbSet

A DbSet is a generic type (see https://msdn.microsoft.com/en-us/library/512aeb7t.aspx) that is parameterised with a class for the row instances

- Each **row** in the database is an instance of class

- One property per column

Our Model: Employee.cs

```
...
using System.Data.Entity;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace Console_Connect;
{
public class EmployeesContext : DbContext
    {
        public EmployeesContext() : base("MySqlConnection") { }

        public DbSet<Employee> Employees { get; set; }
    }

 [Table("employee")]
    public class Employee
    {
       [Key]
          public int id { get; set; }
   public string Surname { get; set; }
   public string GivenNames { get; set; }
   public DateTime DateOfBirth { get; set; }
    }
```

First 'using' directive is added for referencing the DbContext object; next two are for the 'key' and 'table' data annotations

namespace matches the application's Main

Use that connection we added to the App.config file

Our context's DbSet 'Employees' will be a collection of objects of type Employee

The table in MySQL is named 'employee'

Define properties of this class with names matched to table fields in MySQL ([Key] identifies 'id' as the primary key as set in MySQL)

All public so they'll be visible to the Main

Each column modelled as a property (rather than just a field) with default get and set methods

Some action logic using the model

We can place this in the Main method of a Console application

Or in an event handler or action listener in other application templates

Open the connection by making an instance of our specialised child of DbContext; using statement manages closing connection when we leave the statements curly braces

```
...
 using (EmployeesContext db = new EmployeesContext())
    {
    Console.WriteLine(String.Format("Connection string: {0}",
db.Database.Connection.ConnectionString));

    int j = db.Employees.Count();
    Console.WriteLine(String.Format("We have {0} records.", j));

    var x2 = db.Database.SqlQuery<Employee>("select * from employee");

    foreach (Employee emp in x2)
       {
       Console.WriteLine(String.Format("Employee: {0}, DoB: {1:d}",
emp.Surname,emp.DateOfBirth));
       }
...
```

db has an Employees class that inherits from DbSet; it has some methods, like count

Native SQL query that will return instances of Employee

From the model, the compiler knows the names and data types of the properties for a row from the table

Focus on some of language elements

Murdoch UNIVERSITY

- These aren't restricted to EF or working with databases

Just really handy C# stuff!

'Generics' – take a type as a parameter (marked with angle brackets); let's us have code that general in terms of the type of object it works with. In this case, we're saying we expect the method to return an ordered set of whatever is put in the type parameter

```
var x2 = db.Database.SqlQuery<Employee>("select * from employee");
```

Implicitly typed local variable (var) – we don't have to worry on exactly what it's called… in this case System.Data.Entity.Infrastructure.DbRawSqlQuery<TElement> (phew!)

foreach loop – we get to name the index variable, emp, which is of the type of the elements in the collection x2 (Employee, in this case); you can make a foreach on an array

```
foreach (Employee emp in x2) {...
```

By the way, the system doesn't actually attempt to run the query until we enumerate it (i.e. until it needs to provide the answer)

And from the model definition

Murdoch UNIVERSITY

- How C# does basic OO

Colon (:) operator in a class definition says what class to derive a new 'child' class from

```
public class EmployeesContext : DbContext
    {
        public EmployeesContext() : base("MySqlConnection") { } ...
```

'base' keyword says to access the derived (parent) class; and a public method with the same name as the class is a constructor (runs when we do 'new') – so this is saying that a 'new' EmployeeContext with no parameters should be implemented as a new DbContext passed the string "MySqlConnection"

That's all we wanted to do, so no code in the curly braces… could've added additional stuff to happen after running the parent's constructor

Query with more specific return value

Murdoch UNIVERSITY

- Let's say we don't want all the fields in the table, but just a specific subset

```
...
 var x3 = db.Database.SqlQuery<EmpName>("select Surname,GivenNames from
employee");
 foreach (EmpName e2 in x3) {
      Console.WriteLine("Given Names: " + e2.GivenNames + " Surname: "
+ e2.Surname);
      }...
```

- We need a class for the return values

x3 is implicitly typed for the SqlQuery return value, but we still need a parameter for the generic type

Thus we need to add to our model a class definition

```
 public class EmpName
    {
        public string Surname { get; set; }
        public string GivenNames { get; set; }
    }
```

We could do something similar if we wanted a subset of fields from a join

A query with a ‘lambda’

Murdoch UNIVERSITY

- Built-in methods derived from DbSet* can save us from making an explicit query for many needs

```
foreach (var x in db.Employees.Where(a => a.Surname == "Good"))
    {
    Console.WriteLine(String.Format("Employee {0}; date of birth:
{1:d}", x.GivenNames + " " + x.Surname, x.DateOfBirth));
    }
```

Here the Where method expects as its parameter a function that operates on an instance on the elements of Employees and returns a boolean

We use the ‘lambda’ syntax to specify this (=> operator, can read as ‘such that’)

So this tests each employee record to see if their surname is “Good”
(‘a’ is an arbitrary variable name like the iterator variable in a for loop)

Murdoch UNIVERSITY

Oracle ADO.Net Example

- https://www.oracle.com/webfolder/technetwork/tutorials/obe/db/dotnet/GettingStartedNETVersion/GettingStartedNETVersion.htm

Create a simple data application by using ADO.NET

- https://docs.microsoft.com/en-us/visualstudio/data-tools/create-a-simple-data-application-by-using-adonet?view=vs-2019