

### Main Classes (1 per entity)

```
public class Student
{
    public Student() { }
    public int StudentID { get; set; }
    public string StudentName { get; set; }
    public DateTime? DateOfBirth { get; set; }
    public byte[] Photo { get; set; }
    public decimal Height { get; set; }
    public float Weight { get; set; }

    public int StandardId { get; set; } // foreign key for line below (paired)
    public Standard Standard { get; set; } // 1 per student
}

public class Standard {
    public Standard() { }
    public int StandardId { get; set; }
    public string StandardName { get; set; }

    public ICollection<Student> Students { get; set; } // many per standard
}
```

If you dont define the foreign key field name <class>Id in the Student class, it will be created automatically and called Standard\_StandardId

### DbContext Class (1 required)

```
public class Context: DbContext
{
    // SchoolDbConnectionString is the connection string from the config file
    public SchoolContext(): base("name=SchoolDbConnectionString")
    {
        //Disable initializer - we dont want to lose data EVER
        Database.SetInitializer<SchoolDbContext>(null);
    }

    public DbSet<Student> Students { get; set; }
    public DbSet<Standard> Standards { get; set; }
}
```

### Conventions

Primary key is Id or <class name>Id (or use Data Annotation)

Every Entity MUST have a primary key

Foreign keys are created as IList<T> or ICollection<T>



By **blinkdata**  
[cheatography.com/blinkdata/](http://cheatography.com/blinkdata/)

Not published yet.  
Last updated 13th May, 2016.  
Page 1 of 5.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

### Data Annotations

[Key]	Make this into a primary key
[Key] [Column(Order=1)]	First part of composite primary key
[Key] [Column(Order=2)]	Second part of composite primary key
[TimeStamp] public byte[] RowVersion { get; set; }	Used for concurrency checking. Only works for byte[]. Autofills
[ConcurrencyCheck]	Use as a concurrency check. Any type. No autofill
[Required]	Required value. Forces NOT NULL
[MaxLength(50)]	Maximum of 50 characters
[MinLength(2)]	Minimum of 2 characters
[MaxLength(50),MinLength(2)]	Min and Max length combined
[StringLength(50)]	Make nvarchar(50) instead of nvarchar(max)
[Column("Name")]	Use this as field name in the DB instead of the property name
[Column("Name", TypeName="varchar")]	Set the fieldname and the data type
[NotMapped]	Dont create a field in the database (unbound data)
[ForeignKey("SpecificIdField")]	Use the specified id field to hold the foreign key value
[Index]	Create a non clustered index on thsi field
[Index( "INDEX_REGNUM", IsClustered=true, IsUnique=true )]	Create a clustered, unique index with the given name (instead of IX_propertyname)

[Table("StudentMaster")] Use this as the table name instead of the class name

The "Table" annotation goes just before the public class line. All other annotations go before the properties themselves

### DbEntityEntry

var entry = context.Entry(student)	Get a DbEntityEntry for the current student
entry.State	Return Modified, Deleted, Added, Unchanged or Detached
entry.OriginalValues["age"]	The original (unchanged) value
entry.CurrentValues["age"]	The current value
context.Entry(student).State = System.Data.Entity.EntityState.Modified;	Force to a modified state (even if it hasnt been) Needed for disconnected entities
entry.Reload();	Forces the data to be reloaded from the database (state will become UnChanged) All changes will be lost

Note : The context will have been created with:  
using (var context = new SchoolDBEntities()) { }



By **blinkdata**  
[cheatography.com/blinkdata/](http://cheatography.com/blinkdata/)

Not published yet.  
Last updated 13th May, 2016.  
Page 2 of 5.

Sponsored by **CrosswordCheats.com**  
Learn to solve cryptic crosswords!  
<http://crosswordcheats.com>

### Add Entity (in disconnected state)

```
// create new Student entity object in disconnected scenario (out of the scope of DbContext)
var newStudent = new Student();
//set student name
newStudent.StudentName = "Bill";
//create DbContext object
using (var dbCtx = new SchoolDBEntities()) {
    //Add Student object into Students DBset
    dbCtx.Students.Add(newStudent);

    // call SaveChanges method to save student into database
    dbCtx.SaveChanges();
}
```

### Update Entity (in disconnected state)

```
//1. Get student from DB
using (var ctx = new SchoolDBEntities())
{
    stud = ctx.Students.Where(s => s.StudentName == "New Student1").FirstOrDefault<Student>();
}
//2. change student name in disconnected mode (out of ctx scope)
if (stud != null)
{
    stud.StudentName = "Updated Student1";
}
//save modified entity using new Context
using (var dbCtx = new SchoolDBEntities())
{
    //3. Mark entity as modified
    dbCtx.Entry(stud).State = System.Data.Entity.EntityState.Modified;

    //4. call SaveChanges
    dbCtx.SaveChanges();
}
```

### Delete Entity (in disconnected state)

```
//1. Get student from DB
using (var ctx = new SchoolDBEntities())
{
    studentToDelete = ctx.Students.Where(s => s.StudentName == "Student1").FirstOrDefault<Student>();
}
//Create new context for disconnected scenario
using (var newContext = new SchoolDBEntities())
{
    newContext.Entry(studentToDelete).State = System.Data.Entity.EntityState.Deleted;
}
```



### Delete Entity (in disconnected state) (cont)

```
newContext.SaveChanges();
}
```

### Update Entity Graph using DbContext

TBA !!!

Complex - needs more research at this stage

### Raw SQL

```
using (var ctx = new SchoolDBEntities())
{
    //Update command
    int noOfRowUpdated = ctx.Database.ExecuteSqlCommand("Update student
        set studentname ='changed student by command' where studentid=1");
    //Insert command
    int noOfRowInserted = ctx.Database.ExecuteSqlCommand("insert into student(studentname)
        values('New Student')");
    //Delete command
    int noOfRowDeleted = ctx.Database.ExecuteSqlCommand("delete from student
        where studentid=1");
}
```

### Convert DbContext toObjectContext

```
using (var ctx = new SchoolDBEntities()) {
    var objectContext = (ctx as System.Data.Entity.Infrastructure.IObjectContextAdapter).ObjectContext;
    //use objectContext here..
}
```

### Queries

```
// Get a record by its Primary key value - return null if no record found
using (var ctx = new SchoolDBEntities())
{
    var student = ctx.Students.Find(_id);
}

// Get the first (TOP 1) record - return null if no record found
using (var ctx = new SchoolDBEntities())
{
    var student = (from s in ctx.Students
        where s.StudentName == "Student1"
        select s).FirstOrDefault<Student>();
}

// Get a List of records that match the criteria
using (var ctx = new SchoolDBEntities())
{
```



### Queries (cont)

```
var studentList = (from s in ctx.Students
    where s.StudentName == "Student1"
    orderby s.StudentName ascending
    select s).ToList<Student>();
}
```

### Other Notes

How to set default values

Create a new partial class (don't edit the autogenerated one) and set the default values in the constructor of the new class.



By **blinkdata**

[cheatography.com/blinkdata/](http://cheatography.com/blinkdata/)

Not published yet.

Last updated 13th May, 2016.

Page 5 of 5.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>