Cheatography

Entity Framework Cheat Sheet

by blinkdata via cheatography.com/7996/cs/3215/

Main Classes (1 per entity)

```
public class Student
{
        public Student() { }
        public int StudentID { get; set; }
        public string Studen tName { get; set; }
        public DateTime? DateOf Birth { 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 Standa rdName { get; set; }

        public IColle cti on< Stu den t> 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
{
       // School DbC onn ect ion String is the connection string from the config file
        public School Con text(): base("n ame =Sc hoo lDb Con nec tio nSt rin g")
       {
                        //D isable initia lizer - we dont want to lose data EVER
                        Dat aba se.S et Ini tia liz er< Sch ool DBC ont ext >(n ull);
         }

        public DbSet< Stu den t> Students { get; set; }
        public DbSet< Sta nda rd> 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/

Not published yet.
Last updated 13th May, 2016.

Cheatography

Entity Framework Cheat Sheet

by blinkdata via cheatography.com/7996/cs/3215/

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_proper-tyname)
[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.Entit-yState.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()) { }

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
newStu den t.S tud entName = " Bil l";
//create DBContext object
using (var dbCtx = new School DBE nti ties()) {
        //Add Student object into Students DBset
        dbC tx.S tu den ts.A dd (ne wSt udent);

       // call SaveCh anges method to save student into database
        dbC tx.S av eCh ang es();
}
```

Update Entity (in disconnected state)

```
//1. Get student from DB
using (var ctx = new School DBE nti ties())
{
       stud = ctx.St ude nts.Wh ere(s => s.Stud entName == "New Studen t1").Fi rst OrD efa ult <St ude nt>
();
}
//2. change student name in discon nected mode (out of ctx scope)
if (stud != null)
{
        stu d.S tud entName = " Updated Studen t1";
}
//save modified entity using new Context
using (var dbCtx = new School DBE nti ties())
{
       //3. Mark entity as modified
        dbC tx.E nt ry( stu d).S tate = System.Da ta.E nt ity.En tit ySt ate.Mo dified;

       //4. call SaveCh anges
        dbC tx.S av eCh ang es();
}
```

Delete Entity (in disconnected state)

```
//1. Get student from DB
using (var ctx = new School DBE nti ties())
{
        stu den tTo Delete = ctx.St ude nts.Wh ere(s => s.Stud entName == " Stu den t1").Fi rst OrD efa -
ult <St ude nt>();
}
//Create new context for discon nected scenario
```

Delete Entity (in disconnected state) (cont)

```
> using (var newContext = new SchoolDBEntities())
{
   newContext.Entry(studentToDelete).State = System.Data.Entity.EntityState.Deleted;
   newContext.SaveChanges();
}
```

Update Entity Graph using DbContext

TBA !!!

Complex - needs more research at this stage

Raw SQL

```
using (var ctx = new SchoolDBEntities())
{
        //U pdate command
       int noOfRo wUp dated = ctx.Da tab ase.Ex ecu teS qlC omm and ("Update student
                       set studen tname ='changed student by command' where studen tid =1");
        //I nsert command
       int noOfRo wIn serted = ctx.Da tab ase.Ex ecu teS qlC omm and ("insert into studen t(s tud ent -
name)
                        val ues ('New Studen t') ");
        //D elete command
       int noOfRo wDe leted = ctx.Da tab ase.Ex ecu teS qlC omm and ("delete from student
                       where studen tid =1");
}
```

Convert DbContext to ObjectContext

```
using (var ctx = new SchoolDBEntities()) {
       var object Context = (ctx as System.Da ta.E nt ity.In fra str uct ure.IO bje ctC ont ext Ada pte -
r).O bj ect Con text;
        //use object Context here..
}
```

Queries

```
// Get a record by its Primary key value - return null if no record found
using (var ctx = new School DBE nti ties())
{
       var student = ctx.St ude nts.Fi nd( _id);
}
// Get the first (TOP 1) record - return null if no record found
using (var ctx = new School DBE nti ties())
{
       var student = (from s in ctx.St udents
```

Queries (cont)

```
>                 where s.StudentName == "Student1"
                select s).FirstOrDefault<Student>();
}
// Get a List of records that match the criteria
using (var ctx = new SchoolDBEntities())
{
    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/

Not published yet.
Last updated 13th May, 2016.