# EF知识

1. Code First在第一次查询、更新或者删除时创建，或者使用context.Database.CreateIfNotExists()显式创建。
2. DbContext跟踪所有保存在DbSet中的对象的状态，这些状态可以是 Deleted Added Modified Unchanged
3. DbContext.SaveChanges() 运行在一个事务中，所有等待的提交要么全部执行，要么全部不执行或者回滚。SaveChanges保存在上下文中修改的数据到数据库中，但是没有关闭底部的连接。
4. Entity Framework 使用延迟执行，查询命令只有在遍历或者ToList等时才真正到数据库查询。
5. 数据库初始化策略有四种：A: Database.SetInitializer<XXXXXContext>(new DropCreateDatabaseAlways<XXXXXContext>()) 每次运行时创建新的数据库 B:  Database.SetInitializer<XXXXXContext>(new CreateDatabaseIfNotExists<XXXXXContext>()); 只有第一次运行时创建数据库，这是默认的方式 C:  Database.SetInitializer<XXXXXContext>(new DropCreateDatabaseIfModelChanges<XXXXXContext>()); 修改模型后创建新的数据库 D:   Database.SetInitializer<XXXContext>(null); 从不创建数据库
6. Sql Server和.Net数据类型之间的映射。 Seed方法用来初始化数据，但手工调用了CreateIfNotExists则Seed不会被调用。

|  |  |
| --- | --- |
| **SQL Server Database type** | **.NET Framework type** |
| Bigint | Int64 |
| binary, varbinary | Byte[] |
| Bit | Boolean |
| date, datetime, datetime2, smalldatetime | DateTime |
| Datetimeoffset | DateTimeOffset |
| decimal, money, smallmoney, numeric | Decimal |
| float | Double |
| int | Int32 |
| nchar, nvarchar,, char, varchar | String |
| real | Single |
| rowversion, timestamp | Byte[] |
| smallint | Int16 |
| time | TimeSpan |
| tinyint | Byte |
| uniqueidentifier | Guid |
|  |  |

1. EF默认是Unicode的
2. 三种关系： One-To-Many One-To-One Many-To-Many

One-To-Many 在One的一方加入 Virtual ICollection<Many> Property，Virtual是用来Lazy Loading的，Lazy Loading指的是真正访问数据时才去数据库访问数据。在Many中可以加也可以不加OneId外键标志和virtual One对象。可选的在OneMap中加入HasMany映射。

*Many-To-Many通过Junction连接表实现，只需在其中一个Map中添加*

HasMany(p => p.Companies).WithMany(c => c.Persons).Map(m =>

{

m.MapLeftKey("PersonId");

m.MapRightKey("CompanyId");

});

系统自动生成连接表。连接表可以添加额外的字段。

One-To-One One A其中一个用另一个B的主键做主键和外键 Virtual A A;

HasKey(s=>s.PersonId).HasRequired(s => s.Person).WithOptional(p => p.Student);

1. String.Contains()转换为SQL的LIKE.
2. First FirstOrDefault Single SinglrOrDefault Any All
3. EF默认是Lazy Loading. 可以通过 context.Configuration.LazyLoadingEnable = false;来激活Eager Loading或者使用Include方法.
4. Ignore(p => p.FullName);可以忽略一些Property.
5. EF支持枚举。
6. 使用Map实现Entity Split
7. Projection使用匿名类或者定义新的类。
8. DbFunctions可以使用的SQL函数
9. JOIN指的是Inner Join

from person in context.People

join personType in context.PersonTypes

on person.PersonTypeId equals personType.PersonTypeId

select new{}；

1. Distinct Union Intersect Except
2. SqlQuery ExecuteSqlCommand
3. Concurrency是多个用户在同一时间修改同一行记录时，产生冲突。冲突有：丢失更新和脏读。悲观锁：**假定会发生并发冲突，**屏蔽一切可能违反数据完整性的操作。 乐冠锁：**：假设不会发生并发冲突，**只在提交操作时检查是否违反数据完整性。
4. EF内建支持乐观锁。在SQL Server中添加一列byte[] RowVersion.

Property(p=>p.RowVersion).IsFixedLength().HasMaxLength(8).HasDatabaseGeneratedOption(DatabaseGeneratedOption.Computed).IsRowVersion();

1. 共享锁（S锁），又称读锁，若事务T对数据对象A加上S锁，则事务T可以读A但不能修改A，其他事务只能再对A加S锁，而不能加X锁，直到T释放A上的S锁。这保证了其他事务可以读A，但在T释放A上的S锁之前不能对A做任何修改。排它锁（X锁）又称写锁。若事务T对数据对象A加上X锁，事务T可以读A也可以修改A，其他事务不能再对A加任何锁，直到T释放A上的锁。这保证了其他事务在T释放A上的锁之前不能再读取和修改A。