Entity framework is able to track the changes made to entities, so the correct updates are made on the database when the SaveChanges method of context is called. The Change Tracking tracks changes while adding new records to the entity collections, modifying or removing existing entities.

Every entity must have EntityKey (primary key) property in order to be tracked by the context.

Entity framework will not add any entity in the conceptual model which does not have an EntityKey property.

DbChangeTracker class gives you all the information about current entities being tracked by the context.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DBFirst1

{

    class Autotracking

    {

        static void Main(string[] args)

        {

            ProdMgmtEntities db = new ProdMgmtEntities();

            Product prod = new Product() { Pid = 18, Pname = "ccccc", Cid = 11, Price = 99 };

            db.Products.Add(prod);

            foreach(var t in db.ChangeTracker.Entries<Product>())

            {

                Console.WriteLine(t.State);

            }

        }

    }

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DBFirst1

{

    class Autotracking

    {

        static void Main(string[] args)

        {

            ProdMgmtEntities db = new ProdMgmtEntities();

            Product prod = new Product() { Pid = 18, Pname = "ccccc", Cid = 11, Price = 99 };

            db.Products.Add(prod);

            Console.WriteLine("ADDED>>>>>>>>>>>>>>");

            foreach(var t in db.ChangeTracker.Entries<Product>())

            {

                Console.WriteLine(t.State);

            }

            Product prodmod = db.Products.Find(13);

            if(prodmod!=null)

            {

                prodmod.Price = 999;

            }

            Console.WriteLine("MODIFIED >>>>>>>>>>>>>>");

            foreach (var t in db.ChangeTracker.Entries<Product>())

            {

                Console.WriteLine(t.State);

            }

            Product proddel = db.Products.Find(13);

            if (prodmod != null)

            {

                db.Products.Remove(proddel);

            }

            Console.WriteLine("DELETED >>>>>>>>>>>>>>");

            foreach (var t in db.ChangeTracker.Entries<Product>())

            {

                Console.WriteLine(t.State);

            }

        }

    }

}