Code First Approch: 1) It Creates automatic tables based on model class.

We define Dbset property in DbContext class , on the basis of that it creates tables.

0) Define a Model class

1)[ Key] this annotation is used for primery key

2) Define StudentContext which extends DbContext

public class ShopDbContext : DbContext

{

    // DbSet<Product> this entity  property name is Products . so entity framework will go in database it will find Products table will mapped

    // to Product Entity. by default table name is considerd as Property name

    // but the problem is we dont have Products table we have ProductInfo table so solution of this is

     // 1) public DbSet<Product> ProductInfo {get;set;}  write property name ProductInfo

     // 2) use annotation or fluent Api way

     // if we dont have table inside database and we let to entity framework to create table so EF will create table as Property name

    public DbSet<Product> products {get;set;}

    // if u dont have have property of DbSet<Order> so compiler write public  DbSet<Order>  Order {get;set;}

    // here automatically table name is considered in database as entity name

  /// u need to tell how to connect with data base for that override OnConfiguring method

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

    {

       // base.OnConfiguring(optionsBuilder);

        optionsBuilder.UseSqlite("Data Source=Shop.db");

    }

}

DataBase First approach:-1) In this approach the entity framework creates model classes and properties

Corresponding to existing database object , such as tables and columns.