# Introduction to ORM

* Stands for Object Relational Mapper
* They are used to translate data from a table structure in SQL into something C# language can understand.
  + C# understands objects really well.
  + SQL understands tables really well.
* The ORM that we will use is Entity Framework Core

# Introduction to Entity Framework Core

* One of the popular ORM for .Net 5.0
* It allows us to work with a database by using C# objects and almost completely remove the need for most of the data-access code you will usually have to write.
  + A framework makes life easier for developers and that is why EF is a framework.
* The purpose we created that RRModel that is a class library in our project so that we can use those models to create tables for us in the database or vice versa.

## Two approaches to EF

* Database first approach
  + This is when you create a database first and then translate the table data structure into object data structure for C#.
* Code first approach
  + This is what we were doing last week essentially.

## Some artifacts you’ll be working with

* DBContext
  + A class in EF core that represents a session with the database and can be used to query and save instances of your entities.
* Connection String
  + Will be used to connect to our database (in our case the azure database)
  + Basically, a long string that has information about our database and how to connect to it.
* Migration
  + They are a snapshot of the database schema/architecture given the current state of your models.
* Entities
  + It is a class in EF core that maps to a database table.
* Relationships
  + They are the same as multiplicity in SQL.
  + They basically tell the relationship between entities/models.

## Eager and Lazy Loading