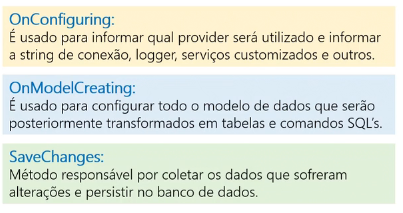
* ORM (object relational mapping): mapeia as classes do projeto com as tabelas do banco de dados
* Composição de classes é quando uma classe depende da outra gerando um relacionamento
* As queries sql geradas do EF são melhores que a do EF (Querys mais otimizadas)
* Comandos
  + Cria uma solution: dotnet new sln -n <NOME\_PROJETO>
  + Cria uma console (-o past -f qual o framework): dotnet new console -n <NOME\_PROJETO> -o Curso -f netcoreapp3.1
  + Adiciona um projeto a solution: dotnet sln <NOME\_SOLUTION\_COM\_EXTENSÃO> add <CAMINHO\_CSPROJ>
* Code First: cria as classes e depois o banco de dados é gerado
* Database First: cria o banco de dados e depois temos duas alternativas:
  + Escrever as classes manualmente de acordo com o Banco de Dados
  + Fazer a engenharia reversa do Banco de Dados sem a necessidade de escrever as classes. Esse processo se chama Scaffold
* DbContext: Combinação dos padrões Unit of Work e Repository, que contém um conjunto de métodos responsáveis por gravar e ler informações do banco de dados



* Migrações: versiona o modelo de dados de acordo com as últimas alterações
* Instalações
  + Instalar o Microsoft.EntityFrameworkCore.SqlServer no projeto
  + Microsoft.EntityFrameworkCore.Design é o pacote que gera as migrações
  + Microsoft.EntityFrameworkCore.Tools é o pacote que gera os comandos para a migração
  + Instalação do CLI EntityFramework para executar os comandos via linha de comando: dotnet tool install --global dotnet-ef --version <NUMERO\_VERSAO>
  + Mostra a versão do EF CLI: dotnet ef
* Adiciona uma migration: dotnet ef migrations add <NOME\_MIGRATION> -p <PATH\_CSPROJ> -o <NOME\_PASTA\_ARQUIVO\_MIGRATION> -c <NOME\_DO\_CONTEXTO>
* Arquivo Migration
  + Método Up: mudanças que estão acontecendo na migração. Pode colocar scripts SQL que são executados na criação do Banco de Dados.
  + Método Down: reverte as alterações do método Up. Pode colocar scripts SQL que são executados na exclusão do Banco de Dados.
* Gera um script das alterações do banco de dados: dotnet ef migrations script -p <PATH\_CSPROJ> -o <NOME\_PASTA\_SCRIPT.sql>
* Em alguma situação gerar o script é necessário pelo fato de não ter acesso ao Banco de Dados de Produção
* Atualiza o banco de dados (-v = verbose): dotnet ef database update -p <PATH\_CSPROJ> -v
* Gera script SQL com validação de objetos (-i = idempotente validação): dotnet ef migrations script -p <PATH\_CSPROJ> -o <NOME\_PASTA\_SCRIPT.sql> -i
  + A validação dos objetos é com base na migração criada (IF NOT EXISTS)
* Remove uma migração: dotnet ef migrations remove -p <PATH\_CSPROJ>

