## Разработка на Софтуер

### Контролна работа № 2 Тема: Приложения и База от Данни

### 1-ва част - Теория

Всички въпроси (без 1ви) са за програмния език **C#**.

1. Какво е **JSON**? За какво служи и къде най-често се употребява?

* JSON, или JavaScript Object Notation, е текстово базиран отворен стандарт създаден за човешки четим обмен на данни. Произлиза от скриптовия език JavaScript, за да представя прости структури от данни и асоциативни масиви, наречени обекти. Въпреки своята връзка с JavaScript, това е езиково независима спецификация, с анализатори, които могат да преобразуват много други езици в JSON.

1. Кои **добрите практики** познавате при употреба на чист **SQL**? (мин. 1)

* Не използвайте SELECT \* във вашите заявки.
* Винаги използвайте псевдоними на таблици, когато вашият SQL израз включва повече от един източник.
* Използвайте по-четливите клаузи ANSI-Standard Join вместо старите съединения.
* Не използвайте номера на колони в клаузата ORDER BY.
* Винаги използвайте списък с колони във вашите INSERT изрази.
* Никога не използвайте двойни кавички във вашия T-SQL код.
* Не добавяйте имената на съхранените процедури с префикс „sp\_“.
* Винаги използвайте SQL форматиращ инструмент, за да форматирате вашия sql като Instant SQL Formatter (безплатен и онлайн)

1. Какво е **ORM (Обектно Релационен Модел)**? За какво служи?

* Обектно-релационен модел (ORM, O / RM и O / R) в компютърните науки е техника за програмиране за преобразуване на данни между несъвместими типови системи с помощта на обектно-ориентирани програмни езици. Това създава на практика „база данни за виртуален обект“, която може да се използва от езика за програмиране. Налични са както безплатни, така и търговски пакети, които извършват обектно-релационно картографиране, въпреки че някои програмисти избират да конструират свои собствени ORM инструменти.

1. Какво разбирате от термина **“Database First”**?

* Database First ви позволява да извършите обратно проектиране на модел от съществуваща база данни. Моделът се съхранява в EDMX файл (разширение .edmx) и може да се преглежда и редактира в Entity Framework Designer. Класовете, с които взаимодействате във вашето приложение, се генерират автоматично от EDMX файла.

1. Какво разбирате от термина **“Code First”**?

* Entity Framework представи подходът Code-First с Entity Framework 4.1. Code-First е полезен главно в Domain Driven Design. В подхода Code-First се фокусирате върху домейна на вашето приложение и започвате да създавате класове за обекта на вашия домейн, вместо първо да проектирате вашата база данни и след това да създавате класовете, които съответстват на дизайна на вашата база данни. първо код в рамката на обекта, Както можете да видите на горната фигура, EF API ще създаде базата данни въз основа на класовете и конфигурацията на вашия домейн. Това означава, че първо трябва да започнете да кодирате в C # или VB.NET и след това EF ще създаде базата данни от вашия код.

1. Какво е  **Миграция**?

* В проектите от реалния свят моделите на данни се променят с внедряването на функциите: добавят се и премахват нови обекти или свойства и схемите на базата данни трябва да се променят съответно, за да се синхронизира с приложението. Функцията за миграция в EF Core осигурява начин за постепенно актуализиране на схемата на базата данни, за да я поддържа в синхрон с модела на данни на приложението, като същевременно запазва съществуващите данни в базата данни.

На високо ниво миграциите функционират по следния начин:

* Когато се въведе промяна на модела на данни, разработчикът използва инструменти на EF Core, за да добави съответна миграция, описваща актуализациите, необходими за синхронизиране на схемата на базата данни. EF Core сравнява текущия модел със моментна снимка на стария модел, за да определи разликите и генерира миграционни файлове с източник; файловете могат да бъдат проследявани в контрола на източника на вашия проект, както всеки друг файл източник.
* След като бъде генерирана нова миграция, тя може да бъде приложена към база данни по различни начини. EF Core записва всички приложени миграции в специална таблица за история, като му позволява да знае кои миграции са приложени и кои не.

### 2-ра част - Практика

1. Направете минмална програма с CRUD Операции (**SQL** или **ORM)**.