

### Тестовое задание:

Вы работаете над проектом справочной системы, в которой пользователь имеет возможность сохранять объекты произвольной структуры (динамическая схема данных). Структура подобных объектов задается пользователем и может изменяться с течением времени. Жизненный цикл работы справочной системы начинается с создания метаданных - структуры будущих объектов. В состав метаданных входят:

1. Класс - именованный контейнер для описания структуры будущих объектов этого класса. Может иметь в своем составе атрибуты. Каждому объекту справочной системы однозначно ставится в соответствие определенный класс.
2. Атрибут - именованный контейнер для описания значения какой-либо характеристики объекта. Каждому значению какой-либо характеристики объекта однозначно ставится в соответствие определенный атрибут.
3. Тип данных. Каждый атрибут в системе характеризуется типом данных, который ограничивает область значений определенной характеристики объекта. На начальном этапе в системе реализовано 3 типа данных: "Строка" и "Число с плавающей точкой", "Дата и время". Поддержка новых типов данных может появиться в системе в будущем.

Пример: Класс "Насос" имеет атрибуты: "Марка", "Напор", "Масса", "Дата установки".

После создания метаданных (схемы данных) пользователь может начать вносить объекты (данные) в справочную систему. Объект - именованная сущность, характеризующуюся классом и набором конкретных значений его атрибутов.

Пример: "Насос1": { "Марка": "ЭЦВ 4-2.5-65", "Напор": 25, "Масса": 18 }

Объекты в справочной системе хранятся в виде дерева - каждый объект либо связан отношением "потомок-родитель" с другим объектом, либо является объектом первого уровня (без родителя).

Спроектируйте реляционную базу данных для вышеописанного проекта с использованием СУБД Microsoft SQL Server 2012.

С помощью Microsoft Visual Studio 2013 (2015) на языке программирования C# с использованием ASP.NET Web API 2 реализуйте набор JSON REST сервисов для реализации следующего функционала:

1. Создание объекта определенного класса в определенном месте дерева объектов с возможностью задания значений атрибутов.
2. Получение поддерева объектов по заданному объекту (только наименования и иерархия).
3. Получение заданного объекта (полностью, со значениями).
4. Среди потомков заданного объекта (могут быть объекты разных классов) выбрать список насосов, установленных до сегодняшнего дня, с массой более 10 кг (для демонстрации работы этого сервиса необходимо создать класс "Насос" с атрибутами "Масса", "Дата установки" и заполнить БД соответствующими данными).

Все операции с данными необходимо осуществить с использованием хранимых процедур на языке Transact SQL. Использование систем ORM исключается.

Поддержка иерархии объектов должна быть реализована на уровне структуры базы данных, при этом, необходимо сделать упор на производительность запросов выборки поддеревьев.

При JSON сериализации/десериализации значения атрибутов объектов должны быть по возможности выражены соответствующими типами данных JSON.

БД наполните тестовыми данными согласно следующим требованиям:

1. Не менее 3 классов со структурой на усмотрение соискателя (не менее 3 атрибутов всех типов данных в каждом классе).
2. Дерево объектов общим числом 1000000, не менее 10 уровней вложенности, содержащее объекты всех 3 классов.

Проект Web API разместите в публичном Git репозитории и предоставьте к нему доступ работодателю. Резервную копию созданной базы данных поместите в тот же репозиторий.