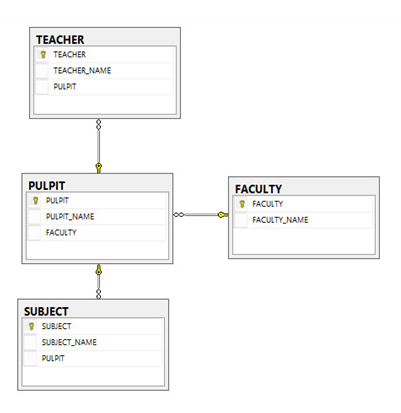
Лабораторная работа 16

ПСКП

ПОИТ-3

**Задание 01**

1. Ознакомьтесь с литературой <https://diskstation.belstu.by:5001> Для\_студентов\_ФИТ\_БГТУ\ЛИТЕРАТУРА\GRAPHQL
2. Ознакомьтесь с диаграммой базы данных(БД)



1. Параметры SQL-сервера: ***172.16.193.223, student/fitfit***
2. Скрипт для создания и заполнения таблиц базы данных ***C:/CommonSpace/10.sql***.
3. Создайте собственный экземпляр БД с именем: **XYZ**, где XYZ – инициалы студента. После сдачи лабораторной работы ***БД должна обязательно быть удалена***.

**Задание 02**

1. Разработайте GrpaphQL-схему, позволяющую выполнять следующие запросы к БД.

|  |  |  |
| --- | --- | --- |
| Наименование | Параметры | Запрос |
| getFaculties | faculty | Если faculty не задан, то формируется список всех факультетов.  Если faculty задан, то формируются данные по заданному факультету |
| getTeachers | teacher | Если teacher не задан, то формируется список всех преподавателей.  Если teacher задан, то формируются данные по заданному преподавателю |
| getPulpits | pulpit | Если pulpit не задан, то формируется список всех кафедр.  Если pulpit задан, то формируются данные по заданной кафедре |
| getSubjects | subject | Если subject не задан, то формируется список всех дисциплин.  Если subject задан, то формируются данные по заданной дисциплине |
| setFaculty | faculty | Если факультет с кодом faculty.faculty уже существует, то заменить.  Если факультет с кодом faculty.faculty не существует, то добавить новый факультет.  Сформировать данные о добавленном/измененном факультете |
| setTeacher | teacher | Если преподаватель с кодом teacher.teacher уже существует, то заменить.  Если преподаватель с кодом teacher.teacher не существует, то добавить нового преподавателя.  Сформировать данные о добавленном/измененном преподавателе |
| setPulpit | pulpit | Если кафедра с кодом pulpit.pulpit уже существует, то заменить.  Если кафедра с кодом pulpit.pulpit не существует, то добавить новую кафедру.  Сформировать данные о добавленной/измененной кафедре |
| setSubject | subject | Если дисциплина с кодом subject.subject уже существует, то заменить.  Если кафедра с кодом subject.subject не существует, то добавить новую дисциплину.  Сформировать данные о добавленной/измененной дисциплине |
| delFaculty | faculty | Если факультет с кодом faculty.faculty существует, то удалитm и сформировать true.  Если факультет с кодом faculty.faculty не существует, то сформировать false |
| delTeacher | teacher | Если преподаватель с кодом teacher.teacher существует, то удалить и сформировать true.  Если преподаватель с кодом teacher.teacher не существует, то сформировать false |
| delPulpit | pulpit | Если кафедра с кодом pulpit.pulpit существует, то удалить и сформировать true.  Если кафедра с кодом pulpit.pulpit не существует, то сформировать false. |
| delSubject | subject | Если дисциплина с кодом subject.subject существует, то удалить и сформировать true.  Если кафедра с кодом subject.subject не существует, то сформировать false. |
| getTeachersByFaculty | faculty | Сформировать список всех преподавателей заданного факультета faculty |
| getSubjectsByFaculties | faculty | Сформировать список всех кафедр и прикрепленных к ним дисциплин заданного факультета faculty |

1. Разработайте приложение **16-01**, представляющее собой HTTP-сервер, прослушивающий порт ***3000***. Сервер должен обрабатывать запросы, описанные в представленной выше таблице. Данные должны сохраняться в БД MSSQL.(п.3).

**Задание 02** Ответьте на следующие вопросы

1. Поясните понятие **GraphQL**.

\* формальный язык запросов

(используется клиентом для загрузки данных с сервера)

\*декларативный язык для извлечения д-х

**Характеристики:**

\*позвол дел.агрегированные запросы (можно в 1 запросе запрос.все необх.д-е)

\*исп.сист.типов для описания д-х

1. Поясните понятие схема **GraphQL**.

Набор определений типов (им.расширение .qraphsql)

Типы – объекты, которые определяют, как будет выглядеть API.

В типах есть поля, кот.возвращ.опред.разновидности д-х.

«!» - поле не дб пустым.

1. Расшифруйте аббревиатуру **SDL GraphQL**.

Schema Definition Language graphQL

1. Поясните понятие **resolver** **GraphQL**.

Функция, которая возвращает д-е для определенного поля

1. Поясните понятие **query GraphQL**.

С их помощью получают данные с сервера (запросы)

1. Поясните понятие **mutation** **GraphQL**.

Изменения – модификация данных на сервера и их обновление

1. Поясните понятие **subscription GraphQL**.

Подписки – методы поддержания постоянной связи с сервером

1. Поясните понятие **context GraphQL**.

Может быть объектом и функцией.

Функция – чтобы мы могли устанавливать контекст каждый раз, когда есть запрос. (включается в каждый GraphQL-запрос)

1. Поясните схему работы модуля **graphql**

Состоит из 3 основных блоков: схема, запросы и распознаватели.

1. Поясните следующие компоненты **subscription, interface, enum, fragment, union схемы GraphQL.**

Interface – абстрактные типы, которые мб реализованы как типы объекта. Определяет все поля, которые дб включены в любой объект, кот.их реализ.

Enum – скалярный тип, который позволяет полю возвращать ограниченный набор строковых значений.

Фрагмент – выборка опред.типа (нужно указ.тип, связанный с каждым фрагментом в его определении)

Union – для возврата 1 из нескольких разных типов.