Лабораторная работа 11

ПСКП

**Задание 01**

1. Ознакомьтесь с диаграммой базы данных (БД).



1. Параметры SQL-сервера: ***172.16.193.223, student/fitfit*.**
2. Скрипт для создания и заполнения таблиц БД - ***11.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 приложение **11-01**, представляющее собой HTTP-сервер, прослушивающий порт ***3000***.
2. Для хранения данных приложение **11-01** должно использовать БД **XYZ.**
3. Сервер должен обрабатывать запросы, описанные в следующих таблицах. Для взаимодействия с БД используйте **пакет mssql.**

|  |  |  |  |
| --- | --- | --- | --- |
| Метод  запроса | URI | | Назначение |
| GET | / | | получить статический HTML-файл, назначение файла описывается в задании 03 |
| GET | api/faculties | | получить список всех факультетов в json-формате |
| GET | api/pulpits | | получить список всех кафедр в json-формате |
| GET | api/subjects | | получить список всех учебных дисциплин в json-формате |
| GET | api/auditoriumstypes | | получить список всех типов учебных аудиторий в json-формате |
| GET | api/auditoriums | | получить список всех учебных аудиторий в json-формате |
| GET | /api/faculty/xyz/pulpits | | xyz - код факультета, вернуть список кафедр, относящихся к факультету с кодом xyz |
| GET | /api/auditoriumtypes/xyz/ auditoriums | | xyz - код типа аудитории, вернуть список аудиторий указанного типа с кодом xyz |
| POST | api/pulpits | | добавить новую кафедру, данные в json-формате |
| POST | api/subjects | | добавить новую учебную дисциплину, данные в json-формате |
| POST | api/auditoriumstypes | | добавить новый тип учебной аудитории, данные в json-формате |
| POST | api/auditoriums | | добавить новую учебную аудиторию, данные в json-формате |
| PUT | api/faculties | | корректировать информацию о факультете, данные в json-формате |
| PUT | api/pulpits | | корректировать информацию о кафедре, данные в json-формате |
| PUT | api/subjects | | корректировать информацию об учебной дисциплине, данные в json-формате |
| PUT | api/auditoriumstypes | | корректировать информацию о типе учебной аудитории, данные в json-формате |
| PUT | /auditoriums | | корректировать информацию об учебной аудитории, данные в json-формате |
| DELETE | api/faculties/xyz | | xyz - код факультета, удалить факультет с кодом xyz |
| DELETE | api/pulpits/xyz | | xyz - код кафедры, удалить кафедру с кодом xyz |
| DELETE | api/subjects/xyz | | xyz - код дисциплины, удалить дисциплину с кодом xyz |
| DELETE | api/auditoriumtypes/xyz | | xyz - код типа аудитории, удалить тип аудитории с кодом xyz |
| DELETE | api/auditoriums/xyz | | xyz - код аудитории, удалить аудиторию с кодом xyz |
| Метод  запроса | | Назначение | |
| GET | | Возвращает HTML-файл, найденные в БД данные или сообщение об ошибке в json-формате | |
| POST | | принимает данные в json-формате,  возвращает добавленные в БД данные или сообщение об ошибке в json-формате | |
| PUT | | принимает данные в json-формате,  возвращает измененные в БД данные или сообщение об ошибке в json-формате | |
| DELETE | | возвращает удаленные в БД данные или сообщение об ошибке в json-формате | |

1. Для соединения с БД сервердолжен применять пул соединений (максимальное кол-во соединений в пуле – 10, минимальное – 4).
2. Для проверки работоспособности приложения, используйте **POSTMAN**.

**Задание 03**

1. HTML-файл, пересылаемый при запросе **GET /**, должен содержать разметку и JS-код, позволяющий отправлять следующие запросы, получать, обрабатывать и отображать ответы:

**GET api/pulpits**

**POST api/pulpits**

**PUT api/pulpits**

**DELETE api/pulpits/xyz**

1. В случае получения от сервера сообщения об ошибке, это сообщение должно быть отображено на странице.

**Задание 04** Ответьте на следующие вопросы

1. Перечислите параметры соединения с сервером БД.
2. Перечислите группы SQL-операторов и операторы, входящие в эти группы.
3. Поясните понятие «результирующий набор».
4. Поясните понятия «транзакция», «фиксация транзакции», «откат транзакции». Как создать транзакцию с помощью пакета **mssql**.
5. Поясните понятие «пул соединений» и его назначение.
6. Перечислите параметры соединения с сервером БД.

Host, user, password, database

1. Перечислите группы SQL-операторов и операторы, входящие в эти группы.
2. **DML (Data Manipulation Language):**
   1. **SELECT**: Получение данных.
   2. **INSERT**: Вставка данных.
   3. **UPDATE**: Обновление данных.
   4. **DELETE**: Удаление данных.
3. **DDL (Data Definition Language):**
   1. **CREATE**: Создание объектов базы данных (таблиц, индексов).
   2. **ALTER**: Изменение структуры базы данных.
   3. **DROP**: Удаление объектов базы данных.
4. **DCL (Data Control Language):**
   1. **GRANT**: Предоставление прав доступа.
   2. **REVOKE**: Отзыв прав доступа.
5. Поясните понятие «результирующий набор».

Результирующий набор (Result Set) - это набор данных, возвращаемых запросом к базе данных. Он представляет собой таблицу с рядами и столбцами, содержащую результат выполнения SQL-запроса.

1. Поясните понятия «транзакция», «фиксация транзакции», «откат транзакции». Как создать транзакцию с помощью пакета **mssql**.

Логическая единица работы, состоящая из одного или нескольких SQL-запросов, выполняемых как единое целое.

1. Поясните понятие «пул соединений» и его назначение.

Механизм управления и повторного использования соединений с базой данных. Вместо создания и закрытия соединений каждый раз при запросе, пул соединений поддерживает заранее созданный набор соединений, который может быть повторно использован для обработки запросов.