**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ**

**ИМ. Б.Л.РОЗИНГА (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным**

**и практическим работам**

**МДК.11.01**

| Студент | ИСПП-35 | |  |  | *А. И. Цырульник* |
| --- | --- | --- | --- | --- | --- |
|  | (Группа) | | (Подпись) | (Дата) | (И.О. Фамилия) |
| Преподаватель | | |  |  | *Ю. С. Маломан* |
|  | |  | (Подпись) | (Дата) | (И.О. Фамилия) |

**Лабораторная работа №1**

**Сбор и анализ требований методом use-case**

**1 Цель работы**

1.1 Изучить процесс описания требований к системе методом use-case.

1.2 Изучить процесс создания диаграммы вариантов использования.

**8 Контрольные вопросы**

8.1 Для чего используется диаграмма вариантов использования?

Ответ: Диаграмма вариантов использования — это графическое представление вариантов использования системы, которые описывают взаимодействие системы с ее пользователями или другими системами. Цель диаграммы вариантов использования — определить и описать все возможные сценарии использования системы, а также определить ключевых пользователей и системы, которые взаимодействуют с ней.

8.2 Что такое «актор» и как он обозначается на диаграмме вариантов использования?

Ответ: Актор — это любое существо, которое взаимодействует с системой, например, пользователь, другой система или устройство. Актор обозначается на диаграмме вариантов использования кругом или эллипсом, внутри которого указывается имя актора.

8.3 Что такое «прецедент» и как он обозначается на диаграмме вариантов использования?

Ответ: Прецедент — это конкретный сценарий использования системы, который описывает взаимодействие системы с актором. Прецедент обозначается на диаграмме вариантов использования прямоугольником или ромбом, внутри которого указывается краткое описание прецедента.

8.4 Что обозначает «отношение ассоциации»?

Ответ: Отношение ассоциации — это связь между актором и прецедентом, которая указывает на то, что актор участвует в этом прецеденте. Отношение ассоциации обозначается на диаграмме вариантов использования линией, которая соединяет актора и прецедента.

8.5 Что обозначает «отношение обобщения»?

Ответ: Отношение обобщения — это связь между двумя или более прецедентами, которые имеют общую основу или цель. Отношение обобщения обозначается на диаграмме вариантов использования линией с двумя стрелками, которая соединяет прецеденты.

8.6 Что обозначает «отношение включения»?

Ответ: Отношение включения — это связь между двумя или более прецедентами, где один прецедент является частью другого. Отношение включения обозначается на диаграмме вариантов использования линией с одной стрелкой, которая соединяет прецеденты.

8.7 Что обозначает «отношение расширения»?

Ответ: Отношение расширения — это связь между двумя или более прецедентами, где один прецедент является расширением другого. Отношение расширения обозначается на диаграмме вариантов использования линией с одной стрелкой, которая соединяет прецеденты.

**Вывод:** В ходе лабораторной работы был изучен процесс описания требований к системе методом use-case, а также процесс создания диаграммы вариантов использования.

**Лабораторная работа №2**

**Проектирование реляционной схемы базы данных в среде СУБД**

**1 Цель работы**

1.1 Научиться применять MySQL Workbench в процессе создания схем моделей БД;

1.2 Научиться представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**8 Контрольные вопросы**

8.1 Что такое «сущность»?

Ответ: Сущность - это объект, который имеет определенные характеристики и атрибуты.

8.2 Что такое «атрибут»?

Ответ: Атрибут - это характеристика или свойство, которое описывает сущность.

8.3 Что такое «ключевое поле»?

Ответ: Ключевое поле - это атрибут, который используется для идентификации уникальной записи в таблице базы данных. Ключевое поле может быть первичным ключом или внешним ключом.

8.4 Каково назначение первичных и внешних ключей?

Ответ: Первичный ключ - это ключевое поле, которое используется для идентификации уникальной записи в таблице базы данных. Внешний ключ - это ключевое поле, которое используется для связи с другой таблицей базы данных.

8.5 Что такое «связь»?

Ответ: Связь - это отношение между двумя или более сущностями.

8.6 Какие виды связей между сущностями существуют?

Ответ: Один к одному (1:1), Один ко многим (1:М), Многие ко многим (М:М).

8.7 Какие элементы входят в ER-диаграммы?

Ответ: Сущности, атрибуты, связи

8.8 Для чего применяются ER-диаграммы?

Ответ: ER-диаграммы используются для представления структуры базы данных и для определения отношений между сущностями.

**Вывод:** В ходе лабораторной работы научились применять MySQL Workbench в процессе создания схем моделей БД, а также представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**Лабораторная работа №3**

**Приведение БД к нормальной форме**

**1 Цель работы**

1.1 Изучить процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;

1.2 Изучить процесс декомпозиции отношений.

**8 Контрольные вопросы**

8.1 Что называется первичным ключом отношения?

Ответ: Первичным ключом отношения (PK) называется уникальный атрибут или набор атрибутов, который определяет каждую запись в отношении.

8.2 Что называется внешним ключом отношения?

Ответ: Внешним ключом отношения (FK) называется атрибут или набор атрибутов, который ссылается на первичный ключ другой таблицы.

8.3 В чем заключается процесс нормализации отношений?

Ответ: Процесс нормализации отношений заключается в удалении повторяющихся данных и обеспечении уникальности данных в каждой записи.

8.4 В каком случае атрибут А функционально зависит от атрибута В?

Ответ: Атрибут А функционально зависит от атрибута В, если значение А можно определить, зная значение В.

8.5 В каком случае атрибут А транзитивно зависит от атрибута В?

Ответ: Атрибут А транзитивно зависит от атрибута В, если значение А можно определить, зная значение В и значения других атрибутов.

8.6 Каким требованиям должно отвечать отношение, находящееся в 1НФ?

Ответ: Каждый атрибут сущности должен быть атомарным.

8.7 Каким требованиям должно отвечать отношение, находящееся во 2НФ?

Ответ: Находиться в 1НФ и каждый непервичный атрибут должен зависеть от PK.

8.8 Каким требованиям должно отвечать отношение, находящееся в 3НФ?

Ответ: Находиться в 2НФ и каждый непервичный атрибут должен транзитивно зависить от PK.

**Вывод:** В ходе лабораторной работы изучили процесс приведения отношений от ненормализованного вида к четвертой нормальной форме, а также процесс декомпозиции отношений.

**Лабораторная работа №4**

**Установка SQL сервера**

**1 Цель работы**

1.1 Научиться устанавливать разные СУБД, используя docker.

**8 Контрольные вопросы**

8.1 Какие стандартные порты для подключения к СУБД MSSQL, MySQL, Postgres?

Ответ: Для MSSQL: 1433, для MySQL: 3306, для Postgres: 5432

8.2 Какие стандартные папки для хранения данных СУБД MSSQL, MySQL, Postgres?

Ответ: Для MSSQL: /var/opt/mssql, для MySQL: /var/lib/mysql, для Postgres: /var/lib/postgresql/data

8.2 Для чего используется Docker?

Ответ: Docker позволяет создавать контейнеры, автоматизировать их запуск и развертывание, управляет жизненным циклом. Он позволяет запускать множество контейнеров на одной хост-машине.

**Вывод:** В ходе лабораторной работы была установлена СУБД MSSQL используя docker.

**Лабораторная работа №5**

**Создание базы данных в среде разработки**

**1 Цель работы**

1.1 Изучить способы обеспечения целостности данных в MS SQL Server;

1.2 Научиться работать в среде SQL Server Management Studio (SSMS).

**8 Контрольные вопросы**

8.1 Что такое SQL Server Management Studio?

Ответ: SQL Server Management Studio (SSMS) — это графическое пользовательское интерфейсное средство для управления и администрирования базами данных Microsoft SQL Server.

8.2 Какие виды авторизации поддерживаются в MS SQL Server?

Ответ: Windows Authentication, SQL Server Authentication.

8.3 Что такое первичный ключ?

Ответ: Первичный ключ - уникальный атрибут или набор атрибутов, который определяет каждую запись в отношении.

8.4 Как указать заполнение столбца автоинкрементными значениями?

Ответ: При создании таблицы указать в поле Identity Specification значение Yes

8.5 Как указать значение по умолчанию?

Ответ: При создании таблицы в поле Default Value or binding

8.6 Как задать проверочное ограничение?

Ответ: Поле, у которого надо указать ограничение -> ПКМ -> Check constraints -> Add

8.7 Как обеспечить уникальность значений в столбце или наборе столбцов?

Ответ: При создании таблицы с помощью SQL-запроса указать CONSTRAINT UNIQUE для нужного столбца

8.8 Что такое внешний ключ?

Ответ: Внешний ключ - атрибут или набор атрибутов, который ссылается на первичный ключ другой таблицы.

8.9 Какие значения может принимать внешний ключ?

Ответ: Любые.

**Вывод:** В ходе лабораторной работы был изучен способ обеспечения целостности данных в MS SQL Server, а также изучила как работать в среде SQL Server Management Studio (SSMS).

**Лабораторная работа №6**

**Создание представлений в СУБД**

**1 Цель работы**

1.1 Научиться создавать и использовать представления в MS SQL Server;

1.2 Закрепить навык создания запросов на выборку в MS SQL Server.

**8 Контрольные вопросы**

8.1 Что такое представления и чем они отличаются от таблиц?

Ответ: Это виртуальная таблица, которая отображает данные на основе запроса.

8.2 Для чего применяются представления?

Ответ: Используется для упрощения запросов, фильтрации данных, повышения безопасности и логики инкапсуляции

8.3 Возможно ли создание представления, включающего информацию из нескольких таблиц одновременно?

Ответ: Возможно с помощью JOIN

8.4 Какие требование предъявляются к обновляемым представлениям?

Ответ: Они должны основываться на одной таблице, без агрегатов, группировок и возвращаемых полей, с указанием соответствия обновляемых полей таблицы

8.5 Как создать обновляемое представление?

Ответ:

CREATE VIEW имя\_представления

WITH SCHEMABINDING

AS

SELECT столбец1, столбец2

FROM представление\_таблица;

**Вывод:** В ходе лабораторной работы были созданы представления и запросы на выборку в MSSQL.

**Лабораторная работа №7**

**Создание функций пользователя в СУБД**

**1 Цель работы**

1.1 Научиться создавать и использовать скалярные и табличные функции пользователя в MS SQL Server;

1.2 Закрепить навык создания запросов на выборку в MS SQL Server.

**8 Контрольные вопросы**

8.1 Что такое функции пользователя и для чего они применяются?

Ответ: Функции, которые вы используете для организации кода в основной части политики

8.2 Чем отличается скалярная функция от табличной?

Ответ: Скалярные функции - функции, которые возвращают число или текст, то есть одно или несколько значений; Табличные функции - функции, которые выводят результат в виде таблицы.

8.3 Как объявить переменную и присвоить ей значение?

Ответ: переменные объявляются с помощью DECLARE @имя\_переменной

8.4 Как указать в функции параметр по умолчанию?

Ответ: @param1 INT = 10

8.5 Как вызвать скалярную функцию?

Ответ: SELECT dbo.function\_name(param1, param2);

8.6 Как вызвать табличную функцию?

Ответ: SELECT \* FROM dbo.function\_name(param1, param2);

**Вывод:** В ходе лабораторной работы были созданы и использованы скалярные и табличные функции пользователя в MSSQL.

**Лабораторная работа №8**

**Создание хранимых процедур в СУБД**

**1 Цель работы**

1.1 Научиться создавать и использовать хранимые процедуры в MS SQL Server;

1.2 Закрепить навык создания запросов на выборку и модификацию данных в MS SQL Server.

**8 Контрольные вопросы**

8.1 Что такое хранимые процедуры и для чего они применяются?

Ответ: Это заранее написанные и защищенные в базе данных программы, которые выполняют одну или несколько операций

8.2 Чем отличается функция пользователя от хранимой процедуры?

Ответ: Пользовательские функции можно использовать как и любое другое выражение в SQL запросе, в то время как хранимые процедуры должны быть вызваны с помощью функции CALL

8.3 Как задать входные параметры хранимой процедуры?

Ответ: Объявляются параметры в скобках после имени процедуры: @имя\_параметра1 тип\_данных1, @имя\_параметра2 тип\_данных2 и т.д

8.4 Как задать выходные параметры хранимой процедуры?

Ответ: Объявляются параметры в скобках после имени процедуры, помечая их как OUTPUT.

8.5 Как вызвать выполнение хранимой процедуры?

Ответ: EXEC имя\_процедуры

**Вывод:** В ходе лабораторной работы были созданы и использованы хранимые процедуры в MSSQL.

**Лабораторная работа №9**

**Создание триггеров в СУБД**

**1 Цель работы**

1.1 Научиться создавать и использовать триггеры в MS SQL Server;

1.2 Закрепить навык создания запросов на выборку и модификацию данных в MS SQLServer.

**8 Контрольные вопросы**

8.1 Что такое триггер?

Ответ: Триггер - это хранимая процедура, которая автоматически выполняется при наступлении определенного события в базе данных (INSERT, UPDATE, DELETE).

8.2 Чем триггер отличается от хранимой процедуры?

Ответ: триггер срабатывает автоматически при возникновении событий (INSERT, UPDATE, DELETE), а хранимая процедура возникает в ручную и по запросу.

8.3 Как запустить триггер на выполнение?

Ответ: Триггер запускается автоматически при наступлении событий (INSERT, UPDATE, DELETE).

8.4 Каково назначение триггеров?

Ответ: Ограничение выполнения действий при наступлении определенного события(например, вставка, обновление, удаление) в базе данных.

8.5 Чем отличаются триггеры INSTEAD OF и AFTER?

Ответ: INSTEAD OF заменяет действие, выполняясь вместо него. AFTER выполняется после завершения действия.

**Вывод:** В ходе лабораторной работы были созданы и использованы триггеры в MSSQL.

**Лабораторная работа №10**

**Реализация доступа пользователей к базе данных**

**1 Цель работы**

1.1 Научиться использовать системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД;

1.2 Научиться назначать привилегии пользователю БД;

1.3 Закрепить навык создания объектов БД

**8 Контрольные вопросы**

8.1 В чем отличие между именами входа и пользователями БД?

Ответ: Имена входа используются для аутентификации на уровне сервера СУБД. Пользователи БД – для авторизации и управления доступом внутри конкретных баз данных.

8.2 Как идентифицируются пользователи в MS SQL Server?

Ответ: С помощью логинов.

8.3 На какие уровни разделяется система безопасности MS SQL Server?

Ответ: Уровень сервера, уровень базы данных.

8.4 Каково назначение ролей сервера?

Ответ: Используются для управления правами и предоставления доступа пользователей к административным функциям на уровне сервера.

8.5 Каково назначение ролей БД?

Ответ: Роли БД обеспечивают управление правами пользователей внутри конкретной БД, обеспечивая доступ к ее объектам и операциям

**Вывод:** В ходе лабораторной работы были использованы системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД, было изучено, назначать привилегии пользователю БД

**Лабораторная работа №11**

**Выполнение резервного копирования и восстановления БД**

**1 Цель работы**

1.1 Научиться выполнять резервное копирование БД;

1.2 Научиться сохранять во внешних файлах описание структуры и данные БД;

1.3 Научиться выполнять восстановление БД

**8** **Контрольные вопросы**

8.1 Для чего создаются резервные копии БД?

Ответ: Для восстановления данных при сбоях или повреждениях, защиты от потери данных, миграции данных.

8.2 В чем отличие между полным и разностным резервным копированием?

Ответ: Полное резервное копирование: Создает полную копию всех данных в БД,

8.3 Как часто должен выполняться каждый из видов резервного копирования БД (привести пример расписания)?

Ответ: Полное резервное копирование – еженедельно (например по субботам), разностное резервное копирование – Ежедневно (например, по будням).

8.4 Какие скрипты можно сформировать для объектов БД?

Ответ: Скрипт создания объекта (CREATE), Скрипт изменения объекта (ALTER), Скрипт удаления объекта (DROP), Скрипт резервного копирования объекта (BACKUP), Скрипт восстановления объекта из резервной копии (RESTORE)

8.5 Как выполнить восстановление БД из резервной копии?

Ответ: Выполнить команду RESTORE DATABASE из резервной копии.

8.6 Какая команда выполняет восстановление БД из резервной копии?

Ответ: RESTORE DATABASE

8.6 В каком порядке надо восстанавливать резервные копии?

Ответ: Полная резервная копия -> Разностные (инкрементные) резервные копии

8.7 Какой параметр у команды восстановления данных отключает/запускает восстановление БД?

Ответ: Параметр RECOVERY

8.8 Для чего используется мастер импорта и экспорта?

Ответ: Для перемещения данных между различными источниками и назначениями, экспорта данных из базы данных в файлы, Импорта данных из файлов в базу данных.

**Вывод:** В ходе лабораторной работы было выполнено резервное копирование БД, сохранено во внешних файлах описание структуры и данные БД, а также выполнено восстановление БД.

**Лабораторная работа №12**

**Экспорт данных базы в документы пользователя**

**1 Цель работы**

1.1 Научиться выполнять экспорт данных из БД.

**8 Контрольные вопросы**

8.1 Как выполнить экспорт csv-файла в Management Studio?

Ответ: Запустить Мастер импорта и экспорта>Выбрать "Экспорт в плоский файл">Указать CSV-файл в качестве назначения>Настроить параметры экспорта (разделители, кодировку и т.д.)>Выполнить экспорт.

8.2 Как выполнить экспорт xlsx-файла в Management Studio?

Ответ: Запустить Мастер импорта и экспорта>Выбрать "Экспорт в Microsoft Excel">Указать XLSX-файл в качестве назначения>Настроить параметры экспорта>Выполнить экспорт.

8.3 Как в Excel открыть файл формата txt/csv и xml?

Ответ:

TXT/CSV: В Excel выбрать "Данные" -> "Из текста/CSV" -> Указать файл и настроить параметры импорта>Открыть файл в Excel.

XML: В Excel выбрать "Данные" -> "Из XML" -> Указать файл XML -> Excel импортирует данные из XML-файла.

8.4 Данные из скольких таблиц могут храниться в файле csv?

Ответ: Данные из 1 таблицы

8.5 Как экспортировать данные в файл формата JSON?

Ответ: SELECT \*

FROM имя\_таблицы

FOR JSON PATH

8.6 Как экспортировать данные в файл формата XML?

Ответ: SELECT \*

FROM имя\_таблицы

FOR XML PATH('имя\_объекта'), ROOT('имя\_корня')

**Вывод:** В ходе лабораторной работы был выполнен экспорт данных из БД.

**Лабораторная работа №13**

**Импорт данных пользователя в базу данных**

**1 Цель работы**

1.1 Научиться выполнять импорт данных в БД.

**8 Контрольные вопросы**

8.1 Как выполнить импорт xlsx-файла в Management Studio?

Ответ: Использовать Мастер импорта/экспорта в MSSQL>Выбрать "Импорт из плоского файла">указать XLSX-файл в качестве источника>настроить параметры импорта>выполнить импорт.

8.2 Как выполнить импорт csv-файла в Management Studio?

Ответ: Использовать Мастер импорта/экспорта в MSSQL>Выбрать "Импорт из плоского файла">указать csv-файл в качестве источника>настроить параметры импорта>выполнить импорт.

8.3 Как выполнить импорт файла XML?

Ответ: Скопировать данные из xml-файла в переменную, затем: SELECT

имя\_таблицы.value('имя\_столбца', 'тип\_данныз') AS имя\_столбца,

…

FROM @xml.nodes('/имя\_корня/имя\_объекта') AS Products(Product);

8.4 Как выполнить импорт файла JSON?

Ответ: Скопировать данные из JSON-файла в переменную, затем: SELECT

JSON\_VALUE(value, '$.имя\_свойства') AS имя\_столбца,

…

FROM OPENJSON(@имя\_переменной)

**Вывод:** В ходе лабораторной работы был выполнен импорт данных в БД.

**Лабораторная работа №14**

**Создание слоя доступа к данным БД**

**1 Цель работы**

1.1 Научиться создавать приложение C# для организации доступа к БД.

1.2 Изучить свойства и методы компонентов SqlConnection, SqlCommand, SqlDataReader, научиться их применять и настраивать;

**8 Контрольные вопросы**

8.1 Какое назначение у элемента SqlConnection?

Ответ: Предназначен для установления соединения с БД SQL Server.

8.2 Какое назначение у элемента SqlCommand?

Ответ: Предназначен для выполнения SQL-запросов и хранимых процедур в БД SQL Server.

8.3 Какое назначение у элемента SqlDataReader?

Ответ: Предназначен для чтения результатов SQL-запросов из БД.

8.4 Какие пространства имен требуется подключить для реализации подключения к СУБД MS SQL Server, MySQL соответственно?

Ответ: Для подключения к MS SQL Server: System.Data.SqlClient, для подключения к MySQL: MySql.Data.MySqlClient

8.5 Какие методы класса SqlCommand позволяют выполнить SQL-запрос?

Ответ: ExecuteReader() - для выполнения запросов, возвращающих набор данных, ExecuteNonQuery() - для выполнения запросов, не возвращающих данные, ExecuteScalar() - для выполнения запросов, возвращающих одно значение

**Вывод:** В ходе лабораторной работы создали приложение C# для организации доступа к БД, а также изучили свойства и методы компонентов SqlConnection, SqlCommand, SqlDataReader, научиться их применять и настраивать;

**Лабораторная работа №15**

**Применение ORM для доступа к данным БД**

**1 Цель работы**

1.1 Научиться создавать приложение C# для организации доступа к БД.

1.2 Научиться проектировать приложение, использующее паттерн репозиторий и Dapper.

**8 Контрольные вопросы**

8.1 Что такое Dapper и для чего используется?

Ответ: Dapper - это микро-ORM (Object-Relational Mapping) библиотека для .NET, которая используется для упрощения работы с базами данных.

8.2 Какие методы Dapper позволяют извлечь данные из БД?

Ответ: 1) Query() - для выполнения SQL-запросов, возвращающих набор данных, QueryFirst() / QueryFirstOrDefault() - для получения первой строки результата, QuerySingle() / QuerySingleOrDefault() - для получения единственной строки результата

8.3 Какие методы Dapper позволяют изменить данные в БД?

Ответ: Execute() - для выполнения SQL-запросов, не возвращающих данные (INSERT, UPDATE, DELETE), ExecuteScalar() - для выполнения запросов, возвращающих одно значение

8.4 Зачем используется паттерн «репозиторий»?

Ответ: используется для: абстрагирования доступа к данным, Инкапсуляции логики работы с базой данных,Обеспечения единого интерфейса для доступа к данным.

**Вывод:** В ходе лабораторной работы создали приложение C# для организации доступа к БД, а также спроектировали приложение, использующее паттерн репозиторий и Dapper.

**Лабораторная работа №16**

**Применение ORM EF Core**

**1 Цель работы**

1.1 Научиться создавать приложение C# для организации доступа к БД.

1.2 Научиться разрабатывать приложение, использующее EF Core.

**8 Контрольные вопросы**

8.1 Что такое «ORM»?

Ответ: Технология, которая позволяет отображать объекты приложения на таблицы базы данных.

8.2 Что такое «EF Core» и для чего он предназначен?

Ответ: Объектно-реляционный маппер (ORM) для .NET, предназначенный для работы с базами данных.

8.3 Как получить данные из БД, используя EF Core?

Ответ: Определить DbContext и DbSet для нужных сущностей, Использовать методы DbSet, такие как Find(), ToList(), FirstOrDefault() и т.д.

8.4 Как выполнить редактирование (вставку, обновление, удаление данных), используя EF Core?

Ответ: Получить нужную сущность из контекста>Изменить данные сущности>Вызвать метод SaveChanges() для сохранения изменений в базе данных.

**Вывод:** В ходе лабораторной работы создали приложение C# для организации доступа к БД, а также разработали приложение, использующее EF Core.

**Лабораторная работа №17**

**Разработка приложения для фильтрации, поиска и сортировки данных**

**1 Цель работы**

1.1 Научиться создавать приложение для доступа к БД средствами EF Core.

1.2 Научиться выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.

**8 Контрольные вопросы**

8.1 Для чего используются метод OrderBy, OrderByDescending, ThenBy, ThenByDescending и в чем их отличие?

Ответ: Эти методы используются для сортировки данных в LINQ-запросах

8.2 Для чего используются методы Take и Skip и как они применяются при пагинации?

Ответ: Методы Take и Skip используются для реализации пагинации в LINQ-запросах

8.3 Для чего используется метод Where?

Ответ: Метод Where используется для фильтрации данных в LINQ-запросах.

8.4 Какие логические операторы могут применяться при составлении условий?

Ответ: При составлении условий в методе Where могут использоваться следующие логические операторы:

&& (and)

|| (or)

! (not)

8.5 Как проверить, что значение есть в списке?

Ответ: можно использовать оператор IN

8.6 Как проверить, что строка начинается с определенного текста?

Ответ: можно использовать оператор LIKE

8.7 Как проверить, что строка содержит определенный текст?

Ответ: можно использовать оператор LIKE

8.8 Как указать в Select список требуемых данных?

Ответ: Пример:

SELECT column1, column2, column3

FROM your\_table;

**Вывод:** В ходе лабораторной работы создали приложение для доступа к БД средствами EF Core, а также выполнили сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.

**Лабораторная работа №18**

**Разработка приложения для редактирования данных**

**1 Цель работы**

1.1 Научиться выполнять вставку, обновление и удаление записей средствами EF Core;

1.2 Научиться обеспечивать обратную связь при редактировании данных.

**8 Контрольные вопросы**

8.1 Для чего используются методы Add() и AddRange() в EF Core?

Ответ: Они используются для добавления новых сущностей в контекст базы данных

8.2 Для чего используются методы Update() в EF Core?

Ответ: Он используется для обновления существующей сущности в контексте базы данных.

8.3 Для чего используются методы Remove() и RemoveRange() в EF Core?

Ответ: Они используются для удаления сущностей из контекста базы данных.

8.4 Как сохранить изменения в БД, используя EF Core?

Ответ: Нужно вызвать метод SaveChanges() на вашем контексте базы данных.

8.5 Как изменить значения полей объекта?

Ответ: Сначала нужно получить объект из базы данных, затем измените его свойства и сохраните изменения с помощью SaveChanges().

8.6 Какое значение по умолчанию присваивается идентификатору нового объекта?

Ответ:

8.7 Как передать объект с одной формы на другую?

Ответ: Можно передать объект через конструктор второй формы.

**Вывод:** В ходе лабораторной работы выполнили вставку, обновление и удаление записей средствами EF Core, а также обеспечили обратную связь при редактировании данных.

**Лабораторная работа №19**

**Выполнение SQL-команд и SQL-подпрограмм**

**1 Цель работы**

1.1 Научиться выполнять SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.

**8 Контрольные вопросы**

8.1 Как выполнить команду на выборку данных в EF Core?

Ответ: Можно использовать методы ToList(), FirstOrDefault(), SingleOrDefault()

8.2 Как выполнить команду на модификацию данных в EF Core?

Ответ: Получите объект из базы данных>Измените необходимые свойства объекта>Вызовите метод SaveChanges() для сохранения изменений.

8.3 Как объявить и передать параметр в SQL-команду в EF Core?

Ответ:Можно использовать метод FromSqlRaw() или ExecuteSqlRaw() для выполнения необработанных SQL-запросов с параметрами.

8.4 Как вызвать табличную функцию в EF Core?

Ответ: Пример:

Создайте класс для представления результата функции:

public class MyResult

{

public int Id { get; set; }

public string Name { get; set; }

}

Вызовите табличную функцию:

using (var context = new YourDbContext())

{

var results = context.Set<MyResult>()

.FromSqlRaw("SELECT \* FROM dbo.MyTableFunction(@param)", new SqlParameter("@param", paramValue))

.ToList();

}

8.5 Как вызвать хранимую процедуру в EF Core?

Ответ: Пример:

Для выборки данных:

using (var context = new YourDbContext())

{

var results = context.MyEntities

.FromSqlRaw("EXEC dbo.MyStoredProcedure @param1, @param2",

new SqlParameter("@param1", value1),

new SqlParameter("@param2", value2))

.ToList();

}

Для выполнения хранимой процедуры без возврата данных:

using (var context = new YourDbContext())

{

context.Database.ExecuteSqlRaw("EXEC dbo.MyStoredProcedure @param1, @param2",

new SqlParameter("@param1", value1),

new SqlParameter("@param2", value2));

}

8.6 Как объявить и передать выходной параметр из хранимой процедуры в EF Core?

Ответ: Объявите выходной параметр и вызовите хранимую процедуру:

using (var context = new YourDbContext())

{

// Объявление выходного параметра

var outputParam = new SqlParameter

{

ParameterName = "@outputParam",

SqlDbType = System.Data.SqlDbType.Int, // Укажите тип данных

Direction = System.Data.ParameterDirection.Output

};

// Вызов хранимой процедуры

context.Database.ExecuteSqlRaw("EXEC dbo.MyStoredProcedure @inputParam, @outputParam OUT",

new SqlParameter("@inputParam", inputValue),

outputParam);

// Получение значения выходного параметра

var result = (int)outputParam.Value;

}

**Вывод:** В ходе лабораторной работы выполнили SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.

**Лабораторная работа №20**

**Реализация разграничения прав доступа пользователей**

**1 Цель работы**

1.1 Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;

1.2 Научиться изменять настройки подключения к БД средствами Entity Framework Core.

**8 Контрольные вопросы**

8.1 Как изменить настройки подключения к БД в клиентском приложении?

Ответ: Изменить строку подключения в файле конфигурации>Обновите контекст базы данных>Настройте сервисы в Startup.cs

8.2 Какими способами можно обеспечить хранение пользователей и ролей пользователей в БД (отобразить в виде ERD)?

Ответ:

8.3 Что такое «авторизация»?

Ответ: Авторизация — это процесс, который определяет, имеет ли пользователь право на доступ к определенным ресурсам или выполнению определенных действий в системе после того, как его личность была подтверждена (аутентификация).

8.4 Что такое «регистрация»?

Ответ: Регистрация — это процесс создания учетной записи пользователя в системе или приложении. Во время регистрации пользователи предоставляют свои данные, такие как имя, адрес электронной почты, пароль и другие необходимые сведения.

**Вывод:** В ходе лабораторной работы разграничили права доступа пользователей на уровне интерфейса приложения, а также изменили настройки подключения к БД средствами Entity Framework Core.

**Лабораторная работа №21**

**Разработка приложения для импорта данных**

**1 Цель работы**

1.1 Импорт данных пользователя в базу данных средствами EF Core.

**8 Контрольные вопросы**

8.1 Как выполнить импорт из csv-файла в БД через приложение?

Ответ: Чтение CSV-файла:

using (var reader = new StreamReader("path/to/file.csv"))

using (var csv = new CsvReader(reader, CultureInfo.InvariantCulture))

{

var records = csv.GetRecords<YourEntity>().ToList();

}

Подключение к базе данных:

Используйте Entity Framework Core или ADO.NET для подключения к базе данных

Сохранение данных в БД:using (var context = new YourDbContext())

{

context.YourEntities.AddRange(records);

context.SaveChanges();

}

8.2 Как импортировать изображение в БД через приложение?

Ответ: Получите изображение: Используйте диалоговое окно для выбора файла или загрузите изображение через интерфейс приложения. Чтение изображения в массив байтов: Прочитайте файл изображения в массив байтов.

byte[] imageData;

using (var fs = new FileStream("path/to/image.jpg", FileMode.Open, FileAccess.Read))

{

using (var br = new BinaryReader(fs))

{

imageData = br.ReadBytes((int)fs.Length);

}

}

Сохранение изображения в БД:

using (var context = new YourDbContext())

{

var entity = new YourEntity

{

ImageData = imageData // Поле для хранения изображения в БД

};

context.YourEntities.Add(entity);

context.SaveChanges();

}

8.3 Как выполнить импорт файла JSON в БД через приложение?

Ответ: Чтение JSON-файла:

Используйте библиотеку System.Text.Json или Newtonsoft.Json для чтения и десериализации JSON.

var jsonString = File.ReadAllText("path/to/file.json");

var records = JsonSerializer.Deserialize<List<YourEntity>>(jsonString);

Подключение к базе данных:

Используйте Entity Framework Core или ADO.NET для подключения к базе данных.

Сохранение данных в БД:

Добавьте десериализованные записи в контекст базы данных и сохраните изменения.

using (var context = new YourDbContext())

{

context.YourEntities.AddRange(records);

context.SaveChanges();

}

**Вывод:** В ходе лабораторной работы импортировали данные пользователя в базу данных средствами EF Core.

**Лабораторная работа №22**

**Разработка приложения для экспорта данных в текстовые файлы**

**1 Цель работы**

1.1 Научиться выполнять экспорт данных из БД в формате текстового файла.

1.2 Научиться выполнять экспорт данных из БД с применением Microsoft Office Word.

**8 Контрольные вопросы**

8.1 Какое пространство имен требуется подключить для работы с Word?

Ответ: Необходимо подключить пространство имен Microsoft.Office.Interop.Word

8.2 Какие библиотеки позволяют сохранять данные в формате docx?

Ответ: DocX, Open XML SDK, NPOI

8.3 Как выполнить экспорт данных в формате txt?

Ответ: Получите данные: Извлеките данные, которые вы хотите экспортировать, например, из базы данных.

Запишите данные в файл TXT: Используйте StreamWriter для записи данных в текстовый файл.

using System.IO;

// Получение данных (например, из базы данных)

var data = new List<string> { "Line 1", "Line 2", "Line 3" };

// Запись данных в файл TXT

using (var writer = new StreamWriter("path/to/file.txt"))

{

foreach (var line in data)

{

writer.WriteLine(line);

}

}

**Вывод:** В ходе лабораторной работы выполнили экспорт данных из БД в формате текстового файла, а также выполнили экспорт данных из БД с применением Microsoft Office Word.

**Лабораторная работа №23**

**Разработка приложения для экспорта данных в табличные файлы**

**1 Цель работы**

1.1 Научиться выполнять экспорт данных из БД в формате табличного файла.

1.2 Научиться выполнять экспорт данных из БД с применением Microsoft Office Excel.

**8 Контрольные вопросы**

8.1 Какое пространство имен требуется подключить для работы с Word?

Ответ: Необходимо подключить пространство имен Microsoft.Office.Interop.Word

8.2 Какие библиотеки позволяют сохранять данные в формате docx?

Ответ: DocX, Open XML SDK, NPOI

8.3 Как выполнить экспорт данных в формате txt?

Ответ: Получите данные: Извлеките данные, которые вы хотите экспортировать, например, из базы данных.

Запишите данные в файл TXT: Используйте StreamWriter для записи данных в текстовый файл.

using System.IO;

// Получение данных (например, из базы данных)

var data = new List<string> { "Line 1", "Line 2", "Line 3" };

// Запись данных в файл TXT

using (var writer = new StreamWriter("path/to/file.txt"))

{

foreach (var line in data)

{

writer.WriteLine(line);

}

}

**Вывод:** В ходе лабораторной работы выполнили экспорт данных из БД в формате табличного файла, а также выполнили экспорт данных из БД с применением Microsoft Office Excel.

**Практическая работа №1**

**Разработка web-API для доступа к данным**

**1 Цель работы**

1.1 Научиться выполнять разработку web-API для доступа к БД.

**8 Контрольные вопросы**

8.1 Что такое REST-запрос?

Ответ: запрос, отправляемый клиентом на сервер в рамках архитектурного стиля REST (Representational State Transfer).

8.2 Что такое RESTful?

Ответ: это термин, описывающий веб-сервисы, которые следуют принципам архитектурного стиля REST (Representational State Transfer).

8.3 Для чего используется метод GET?

Ответ: Метод GET используется в HTTP-запросах для получения данных с сервера.

8.4 Для чего используется метод POST?

Ответ: Метод POST используется в HTTP-запросах для отправки данных на сервер.

8.5 Для чего используется метод PUT?

Ответ: Метод PUT используется в HTTP-запросах для обновления существующего ресурса на сервере или для создания нового ресурса, если он не существует.

8.6 Для чего используется метод DELETE?

Ответ: Метод DELETE используется в HTTP-запросах для удаления существующего ресурса на сервере.

**Вывод:** В ходе практической работы выполнять разработку web-API для доступа к БД.

**Практическая работа №2**

**Вызов методов REST API**

**1 Цель работы**

1.1 Научиться проверять работоспособность RESTful API в клиентском приложении.

1.2 Научиться выполнять тестирование RESTful API методом черного ящика.

**8 Контрольные вопросы**

8.1 Как указать у объекта HttpClient базовый адрес?

Ответ: использовать свойство BaseAddress:

using System;

using System.Net.Http;

class Program

{

static void Main()

{

var client = new HttpClient

{

BaseAddress = new Uri("https://api.example.com/")

};

// Теперь можно использовать client для отправки запросов

var response = client.GetAsync("endpoint").Result; // Запрос к https://api.example.com/endpoint

}

}

8.2 Какие коды ответа могут быть получены при Http-запросе (указать коды и значения)?

Ответ: 100 (Continue): Сервер получил начальные заголовки запроса и клиент может продолжать отправку тела запроса; 101 (Switching Protocols): Сервер согласен переключить протоколы; 200 (OK): Запрос успешно выполнен; 201 (Created): Запрос выполнен, и ресурс был создан; 204 (No Content): Запрос выполнен, но нет содержимого для возврата; 301 (Moved Permanently): Запрашиваемый ресурс был перемещен на новый URL; 302 (Found): Запрашиваемый ресурс временно доступен по другому URL; 304 (Not Modified): Ресурс не изменялся с момента последнего запроса; 400 (Bad Request): Запрос не может быть обработан из-за неверного синтаксиса; 401 (Unauthorized): Необходима аутентификация для доступа к ресурсу; 403 (Forbidden): Доступ к ресурсу запрещен; 404 (Not Found): Запрашиваемый ресурс не найден; 500 (Internal Server Error): Внутренняя ошибка сервера; 502 (Bad Gateway): Сервер, действующий как шлюз, получил недопустимый ответ от вышестоящего сервера; 503 (Service Unavailable): Сервер временно недоступен, обычно из-за перегрузки или технического обслуживания.

8.3 Какой метод класса HttpClient используется для получения данных?

Ответ: Метод класса HttpClient, используемый для получения данных, — это GetAsync

8.4 Какой метод класса HttpClient используется для вставки данных?

Ответ: Метод класса HttpClient, используемый для вставки данных, — это PostAsync.

8.5 Какой метод класса HttpClient используется для изменения данных?

Ответ: Метод класса HttpClient, используемый для изменения данных, — это PutAsync.

8.6 Какой метод класса HttpClient используется для удаления данных?

Ответ: Метод класса HttpClient, используемый для удаления данных, — это DeleteAsync

**Вывод:** В ходе практической работы проверили работоспособность RESTful API в клиентском приложении, а также выполнили тестирование RESTful API методом черного ящика.

**Практическая работа №3**

**Разграничение прав доступа на уровне REST API**

**1 Цель работы**

1.1 Научиться выполнять разработку web-API для доступа к БД.

1.2 Научиться работать с JWT.

**8 Контрольные вопросы**

8.1 Какие атрибуты можно указать у методов REST для настройки доступа для авторизованных и неавторизованных пользователей?

Ответ: [Authorize], [AllowAnonymous], [Authorize(Roles = "Admin")]

8.2 Для чего используется JWT?

Ответ: Используется для безопасной передачи информации между сторонами в виде JSON-объекта.

8.3 В чем отличие между авторизацией с использованием cookie и с использованием JWT с точки зрения безопасности?

Ответ:  **Хранение данных:**

Cookie: Хранит сессионные данные на стороне клиента. Сервер создает сессию и хранит информацию о пользователе на сервере. Cookie содержит идентификатор сессии.

JWT: Хранит все необходимые данные (например, идентификатор пользователя, роли) в самом токене, который подписан и может быть проверен на сервере. Токен передается клиентом при каждом запросе.

### **Статус сессии:**

Cookie: Сессия является состоянием, и сервер должен отслеживать активные сессии. Это может привести к проблемам с масштабируемостью.

JWT: Без состояния. Сервер не хранит информацию о сессии, что упрощает масштабирование, но может усложнить отзыв токенов.

### **Уязвимости:**

Cookie: Уязвимы к атакам CSRF (Cross-Site Request Forgery), если не используются соответствующие меры защиты (например, SameSite атрибут).

JWT: Уязвимы к атакам XSS (Cross-Site Scripting), если токен хранится в локальном хранилище. Также могут быть уязвимы к атакам на подделку токенов, если не используется надежная подпись.

8.4 Какие настройки можно указать при создании токена?

Ответ: Заголовок (Header), Полезная нагрузка (Payload), Секретный ключ (Secret Key), Срок действия (Expiration):

8.5 Почему для авторизации следует использовать метод POST?

Ответ: Безопасность данных: Данные, такие как имя пользователя и пароль, передаются в теле запроса, что делает их менее видимыми в URL, в отличие от метода GET, где параметры передаются в строке запроса.

Изменение состояния: POST предназначен для создания или изменения состояния на сервере. При авторизации сервер проверяет учетные данные и может создавать сессию или токен, что соответствует семантике POST.

**Вывод:** В ходе практической работы выполнили разработку web-API для доступа к БД, а также работали с JWT.

**Практическая работа №4**

**Разработка веб-клиента**

**1 Цель работы**

1.1 Научиться разрабатывать клиентское веб-приложение для доступа к БД.

1.2 Научиться проверять работоспособность RESTful API в клиентском приложении.

**8 Контрольные вопросы**

8.1 Как выполнить HTTP-запрос к API с помощью Fetch API?

Ответ: Чтобы выполнить HTTP-запрос к API с помощью Fetch API, используйте функцию fetch()

8.2 Как обрабатывать ошибки при вызове API в JavaScript?

Ответ: Для обработки ошибок при вызове API в JavaScript с использованием Fetch API можно использовать блоки then() и catch()

8.3 Что такое асинхронные функции и как они связаны с вызовом API?

Ответ: Асинхронные функции в JavaScript — это функции, которые позволяют выполнять асинхронные операции, не блокируя основной поток выполнения. Они объявляются с помощью ключевого слова async и могут использовать оператор await для ожидания завершения промисов.

8.4Что такое Razor Pages?

Ответ: Razor Pages — это фреймворк для создания веб-приложений на платформе ASP.NET Core, который упрощает разработку страниц с динамическим контентом. Он основан на паттерне "страница как контроллер" и использует синтаксис Razor для генерации HTML.

### 

**Вывод:** В ходе практической работы разработали клиентское веб-приложение для доступа к БД, а также проверили работоспособность RESTful API в клиентском приложении.