**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

# Отчеты по лабораторным и практическим работам

**по МДК.11.01**

Студент: А.В. Речицкий

Группа: ИСПП-21

Преподаватель: Ю.С. Маломан

Архангельск 2024

# Лабораторная работа №1

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

1. **Цель работы**
   1. Изучить процесс описания требований к системе методом use-case;
   2. Изучить процесс создания диаграммы вариантов использования.
2. **Ответы на контрольные вопросы**
   1. Для чего используется диаграмма вариантов использования?

Ответ: Диаграмма вариантов используется для визуализации функциональности системы с точки зрения пользователя (акторов)

* 1. Что такое «актор» и как он обозначается на диаграмме вариантов использования?

Ответ: Актор – это сущность, взаимодействующая с системой (пользователь, устройство и т.д.). на диаграмме означается стилизованным человечком

* 1. Что такое «прецедент» и как он обозначается на диаграмме вариантов использования?

Ответ: Прецедент — это описание функциональности, которую система предоставляет актору. Обозначается эллипсом с названием действия.

* 1. Что обозначает «отношение ассоциации»?

Ответ: Ассоциация показывает взаимодействие актора с прецедентом. Обозначается линией между ними.

* 1. Что обозначает «отношение обобщения»?

Ответ: Показывает, что один актор или прецедент является частным случаем другого. Обозначается стрелкой с пустым треугольником.

* 1. Что обозначает «отношение включения»?

Ответ: Показывает, что выполнение одного прецедента всегда включает выполнение другого. Обозначается стрелкой с надписью <<include>>.

* 1. Что обозначает «отношение расширения»?

Ответ: Указывает, что один прецедент может расширять функциональность другого. Обозначается стрелкой с надписью <<extend>>.

1. **Вывод**
   1. В ходе проделанной лабораторной работы я изучил процесс описания требований к системе методом use-case.
   2. В ходе проделанной лабораторной работы я изучил процесс создания диаграммы вариантов использования.

# Лабораторная работа №2

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

1. **Цель работы**
   1. Научиться применять MySQL Workbench в процессе создания схем моделей БД;
   2. Научиться представлять логическую модель данных согласно нотациям ERD и IDEF1X.
2. **Ответы на контрольные вопросы**
   1. Что такое «сущность»?

Ответ: Сущность — это объект или понятие, для которого необходимо хранить данные. Например, клиент, продукт.

* 1. Что такое «атрибут»?

Ответ: Атрибут — это характеристика сущности, описывающая её свойства. Например, для клиента: имя, адрес.

* 1. Что такое «ключевое поле»??

Ответ: Ключевое поле — это атрибут или их комбинация, уникально идентифицирующая запись в таблице.

* 1. Каково назначение первичных и внешних ключей?

Ответ: Первичный ключ обеспечивает уникальность записей. Внешний ключ связывает записи разных таблиц, указывая на первичный ключ другой таблицы.

* 1. Что такое «связь»?

Ответ: Связь определяет отношение между сущностями, например, "один ко многим" или "многие ко многим".

* 1. Какие виды связей между сущностями существуют?

Ответ: Один к одному (1:1). Один ко многим (1:N). Многие ко многим (M:N).

* 1. Какие элементы входят в ER-диаграммы?

Ответ: Сущности. Атрибуты. Связи между сущностями.

* 1. Для чего применяются ER-диаграммы?

Ответ: Для визуального представления структуры базы данных, включая сущности, их атрибуты и связи.

1. **Вывод**
   1. Научился применять MySQL Workbench в процессе создания схем моделей БД;
   2. Научился представлять логическую модель данных согласно нотациям ERD и IDEF1X.

# Лабораторная работа №3

**Приведение БД к нормальной форме**

1. **Цель работы**
   1. Изучить процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;
   2. Изучить процесс декомпозиции отношений.
2. **Ответы на контрольные вопросы**
   1. Что называется первичным ключом отношения?

Ответ: Это поле, которое уникально идентифицирует каждую запись в таблице.

* 1. Что называется внешним ключом отношения?

Ответ: Это поле, ссылающееся на первичный ключ другой таблицы.

* 1. В чем заключается процесс нормализации отношений?

Ответ: Процесс разбиения таблиц на более мелкие, чтобы устранить избыточность данных и избежать аномалий обновления.

* 1. В каком случае атрибут А функционально зависит от атрибута В?

Ответ: Если значение атрибута B однозначно определяет значение атрибута A.

* 1. В каком случае атрибут А транзитивно зависит от атрибута В?

Ответ: Если атрибут A зависит от атрибута B через посредника, т.е., через другой атрибут.

* 1. Каким требованиям должно отвечать отношение, находящееся в 1НФ?

Ответ: Все значения атрибутов атомарны. Нет повторяющихся групп данных.

* 1. Каким требованиям должно отвечать отношение, находящееся во 2НФ?

Ответ: Находится в 1НФ. Все неключевые атрибуты зависят от всего первичного ключа.

* 1. Каким требованиям должно отвечать отношение, находящееся в 3НФ?

Ответ: Находится во 2НФ. Нет транзитивной зависимости между атрибутами.

1. **Вывод**
   1. Изучил процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;
   2. Изучил процесс декомпозиции отношений.

# Лабораторная работа №4

**Установка SQL сервера**

1. **Цель работы**
   1. Научиться устанавливать разные СУБД, используя docker.
2. **Ответы на контрольные вопросы**
   1. Какие стандартные порты для подключения к СУБД MSSQL, MySQL, Postgres?

Ответ: MSSQL — 1433. MySQL — 3306. Postgres — 5432.

* 1. Какие стандартные папки для хранения данных СУБД MSSQL, MySQL, Postgres?

Ответ: MSSQL: C:\Program Files\Microsoft SQL Server\MSSQL\Data.

MySQL: /var/lib/mysql (Linux) или C:\ProgramData\MySQL\MySQL Server x.x\Data (Windows).

Postgres: /var/lib/postgresql/data.

* 1. Для чего используется Docker?

Ответ: Docker используется для контейнеризации приложений, что позволяет быстро разворачивать, изолировать и масштабировать программное обеспечение, включая СУБД.

1. **Вывод**
   1. Научился устанавливать разные СУБД, используя docker.

# Лабораторная работа №5

**Создание базы данных в среде разработки**

1. **Цель работы**
   1. Изучить способы обеспечения целостности данных в MS SQL Server;
   2. Научиться работать в среде SQL Server Management Studio (SSMS).
2. **Ответы на контрольные вопросы**
   1. Что такое SQL Server Management Studio?

Ответ: Это интегрированная среда для управления базами данных SQL Server, выполнения запросов и администрирования.

* 1. Какие виды авторизации поддерживаются в MS SQL Server?

Ответ: Авторизация Windows. Авторизация SQL Server.

* 1. Что такое первичный ключ?

Ответ: Поле или комбинация полей, обеспечивающее уникальность записей в таблице.

* 1. Как указать заполнение столбца автоинкрементными значениями?

Ответ: Использовать свойство IDENTITY(начало, шаг) при создании столбца.

* 1. Как указать значение по умолчанию?

Ответ: Использовать DEFAULT значение при создании столбца.

* 1. Как задать проверочное ограничение?

Ответ: Использовать CHECK (условие) при создании или изменении столбца.

* 1. Как обеспечить уникальность значений в столбце или наборе столбцов?

Ответ: Создать уникальный индекс или ограничение UNIQUE.

* 1. Что такое внешний ключ?

Ответ: Поле в одной таблице, ссылающееся на первичный ключ другой таблицы.

* 1. Какие значения может принимать внешний ключ?

Ответ: Либо значение, соответствующее первичному ключу в связанной таблице, либо NULL.

1. **Вывод**
   1. Изучил способы обеспечения целостности данных в MS SQL Server;
   2. Научился работать в среде SQL Server Management Studio (SSMS).

# Лабораторная работа №6

**Создание представлений в СУБД**

1. **Цель работы**
   1. Научиться создавать и использовать представления в MS SQL Server;
   2. Закрепить навык создания запросов на выборку в MS SQL Server.
2. **Ответы на контрольные вопросы**
   1. Что такое представления и чем они отличаются от таблиц?

Ответ: Представления — это виртуальные таблицы, создаваемые на основе SQL-запросов. Они не хранят данные, а отображают их из других таблиц.

* 1. Для чего применяются представления?

Ответ: Для упрощения сложных запросов, повышения безопасности (ограничение доступа к определенным данным) и представления данных в удобном формате.

* 1. Возможно ли создание представления, включающего информацию из нескольких таблиц одновременно?

Ответ: Да, с использованием соединений (JOIN) в SQL-запросах.

* 1. Какие требование предъявляются к обновляемым представлениям?

Ответ: Должны основываться на одной таблице. Должны содержать все столбцы, участвующие в ограничениях.

* 1. Как создать обновляемое представление?

Ответ: Использовать оператор CREATE VIEW с SQL-запросом, отвечающим требованиям для обновления.

1. **Вывод**
   1. Научился создавать и использовать представления в MS SQL Server;
   2. Закрепил навык создания запросов на выборку в MS SQL Server.

# Лабораторная работа №7

**Создание функций пользователя в СУБД**

1. **Цель работы**
   1. Научиться создавать и использовать скалярные и табличные функции пользователя в MS SQL Server;
   2. Закрепить навык создания запросов на выборку в MS SQL Server.
2. **Ответы на контрольные вопросы**
   1. Что такое функции пользователя и для чего они применяются?

Ответ: Функции пользователя — это программные конструкции, выполняющие вычисления и возвращающие значения. Используются для упрощения и повторного использования кода.

* 1. Чем отличается скалярная функция от табличной?

Ответ: Скалярная функция возвращает одно значение. Табличная функция возвращает набор строк.

* 1. Как объявить переменную и присвоить ей значение?

Ответ: Использовать конструкцию DECLARE @имя\_переменной тип; SET @имя\_переменной = значение;.

* 1. Как указать в функции параметр по умолчанию?

Ответ: При объявлении параметра использовать синтаксис @параметр тип = значение\_по\_умолчанию.

* 1. Как вызвать скалярную функцию?

Ответ: Использовать ее в SQL-запросе, например: SELECT dbo.ИмяФункции(параметры).

* 1. Как вызвать табличную функцию?

Ответ: Использовать в выражении SELECT \* FROM dbo.ИмяФункции(параметры).

1. **Вывод**
   1. Научился создавать и использовать скалярные и табличные функции пользователя в MS SQL Server;
   2. Закрепил навык создания запросов на выборку в MS SQL Server.

# Лабораторная работа №8

**Создание хранимых процедур в СУБД**

1. **Цель работы**
   1. Научиться создавать и использовать хранимые процедуры в MS SQL Server;
   2. Закрепить навык создания запросов на выборку и модификацию данных в MS SQL Server.
2. **Ответы на контрольные вопросы**
   1. Что такое хранимые процедуры и для чего они применяются?

Ответ: Хранимые процедуры — это заранее скомпилированные SQL-скрипты, которые хранятся в БД. Используются для автоматизации и повышения производительности.

* 1. Чем отличается функция пользователя от хранимой процедуры?

Ответ: Функция возвращает значение или таблицу, используется в SQL-запросах. Процедура может выполнять действия (например, модификацию данных) и возвращать несколько выходных параметров.

* 1. Как задать входные параметры хранимой процедуры?

Ответ: Указать параметры в объявлении процедуры, например: CREATE PROCEDURE ИмяПроцедуры (@параметр тип).

* 1. Как задать выходные параметры хранимой процедуры?

Ответ: Использовать ключевое слово OUTPUT, например: @параметр тип OUTPUT.

* 1. Как вызвать выполнение хранимой процедуры?

Ответ: Использовать команду EXEC ИмяПроцедуры параметры.

1. **Вывод**
   1. Научился создавать и использовать хранимые процедуры в MS SQL Server;
   2. Закрепил навык создания запросов на выборку и модификацию данных в MS SQL Server.

# Лабораторная работа №9

**Создание триггеров в СУБД**

1. **Цель работы**
   1. Научиться создавать и использовать триггеры в MS SQL Server;
   2. Закрепить навык создания запросов на выборку и модификацию данных в MS SQLServer.
2. **Ответы на контрольные вопросы**
   1. Что такое триггер?

Ответ: Триггер — это объект базы данных, автоматически выполняющийся при определенных событиях (INSERT, UPDATE, DELETE).

* 1. Чем триггер отличается от хранимой процедуры?

Ответ: Триггер выполняется автоматически при возникновении события, а процедура вызывается явно.

* 1. Как запустить триггер на выполнение?

Ответ: Триггер запускается автоматически при выполнении операции, для которой он настроен.

* 1. Каково назначение триггеров?

Ответ: Автоматизация проверок целостности данных, логирование изменений, выполнение сложных бизнес-правил.

* 1. Чем отличаются триггеры INSTEAD OF и AFTER?

Ответ: INSTEAD OF выполняется вместо заданного события. AFTER выполняется после завершения события.

1. **Вывод**
   1. Научился создавать и использовать триггеры в MS SQL Server;
   2. Закрепил навык создания запросов на выборку и модификацию данных в MS SQLServer.

# Лабораторная работа №10

**Реализация доступа пользователей к базе данных**

1. **Цель работы**
   1. Научиться использовать системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД;
2. **Ответы на контрольные вопросы**
   1. В чем отличие между именами входа и пользователями БД?

Ответ: Имя входа (логин) используется для аутентификации на сервере БД. Пользователь БД привязан к логину и используется для авторизации в конкретной базе данных.

* 1. Как идентифицируются пользователи в MS SQL Server?

Ответ: Идентификация происходит через связку логина и имени пользователя в базе данных.

* 1. На какие уровни разделяется система безопасности MS SQL Server?

Ответ: Уровень сервера (логины и роли сервера). Уровень базы данных (пользователи, роли и разрешения в БД).

* 1. Каково назначение ролей сервера?

Ответ: Управление правами доступа на уровне всего сервера, например, администрирование, управление логинами.

* 1. Каково назначение ролей БД?

Ответ: Управление правами на уровне конкретной базы данных, например, доступ к таблицам, хранимым процедурам.

1. **Вывод**
   1. Научился использовать системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД;
   2. Научился назначать привилегии пользователю БД;
   3. Закрепил навык создания объектов БД.

# Лабораторная работа №11

**Выполнение резервного копирования и восстановления БД**

1. **Цель работы**
   1. Научиться выполнять резервное копирование БД;
   2. Научиться сохранять во внешних файлах описание структуры и данные БД;
   3. Научиться выполнять восстановление БД.
2. **Ответы на контрольные вопросы**
   1. Для чего создаются резервные копии БД?

Ответ: Для восстановления данных в случае сбоев, утрат или повреждений.

* 1. В чем отличие между полным и разностным резервным копированием?

Ответ: Полное копирование сохраняет всю базу данных. Разностное копирование сохраняет изменения с момента последнего полного копирования.

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

Ответ: Полное — раз в сутки (например, ночью). Разностное — каждые 1–4 часа, в зависимости от интенсивности изменений.

* 1. Какие скрипты можно сформировать для объектов БД?

Ответ: Скрипты создания объектов (таблиц, процедур), экспорта данных, восстановления структуры.

* 1. Как выполнить восстановление БД из резервной копии?

Ответ: Использовать команду RESTORE DATABASE с указанием пути к файлу резервной копии.

* 1. Какая команда выполняет восстановление БД из резервной копии?

Ответ: RESTORE DATABASE.

* 1. В каком порядке надо восстанавливать резервные копии?

Ответ: Полную копию. Последовательность разностных копий (если применимо).

* 1. Какой параметр у команды восстановления данных отключает/запускает восстановление БД?

Ответ: Параметры NORECOVERY (отключить) и RECOVERY (запустить).

* 1. Для чего используется мастер импорта и экспорта?

Ответ: Для переноса данных между разными источниками и целями (например, между серверами, файлами, базами).

1. **Вывод**
   1. Научился выполнять резервное копирование БД;
   2. Научился сохранять во внешних файлах описание структуры и данные БД;
   3. Научился выполнять восстановление БД.

# Лабораторная работа №12

**Экспорт данных базы в документы пользователя**

1. **Цель работы**
   1. Научиться выполнять экспорт данных из БД.
2. **Ответы на контрольные вопросы**
   1. Как выполнить экспорт csv-файла в Management Studio?

Ответ: Использовать мастер импорта и экспорта, выбрав CSV в качестве целевого формата.

* 1. Как выполнить экспорт xlsx-файла в Management Studio?

Ответ: Использовать мастер импорта и экспорта, выбрав Excel как целевой формат.

* 1. Как в Excel открыть файл формата txt/csv и xml?

Ответ: Через меню "Открыть" или "Импорт данных", указав соответствующий формат.

* 1. Данные из скольких таблиц могут храниться в файле csv?

Ответ: В одном CSV-файле могут храниться данные только из одной таблицы.

* 1. Как экспортировать данные в файл формата JSON?

Ответ: Использовать функцию FOR JSON в SQL-запросах.

* 1. Как экспортировать данные в файл формата XML?

Ответ: Использовать функцию FOR XML в SQL-запросах.

1. **Вывод**
   1. Научился выполнять экспорт данных из БД.

# Лабораторная работа №13

**Импорт данных пользователя в базу данных**

1. **Цель работы**
   1. Научиться выполнять импорт данных в БД.
2. **Ответы на контрольные вопросы**
   1. Как выполнить импорт xlsx-файла в Management Studio?

Ответ: Использовать мастер импорта данных, выбрав Excel в качестве источника данных.

* 1. Как выполнить импорт csv-файла в Management Studio?

Ответ: Использовать мастер импорта данных, указав CSV-файл как источник и настроив разделители.

* 1. Как выполнить импорт файла XML?

Ответ: Использовать функции XML в SQL Server, например, OPENROWSET или OPENXML.

* 1. Как выполнить импорт файла JSON?

Ответ: Использовать функцию OPENJSON для парсинга и загрузки данных из файла JSON.

1. **Вывод**
   1. Научился выполнять импорт данных в БД.

# Лабораторная работа №14

**Создание слоя доступа к данным БД**

1. **Цель работы**
   1. Научиться создавать приложение C# для организации доступа к БД;
   2. Изучить свойства и методы компонентов SqlConnection, SqlCommand, SqlDataReader, научиться их применять и настраивать.
2. **Ответы на контрольные вопросы**
   1. Какое назначение у элемента SqlConnection?

Ответ: SqlConnection используется для установления подключения к базе данных.

* 1. Какое назначение у элемента SqlCommand?

Ответ: SqlCommand выполняет SQL-запросы или команды к базе данных.

* 1. Какое назначение у элемента SqlDataReader?

Ответ: SqlDataReader используется для чтения данных из базы данных построчно, без сохранения их в памяти.

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

Ответ: Для MS SQL Server: System.Data.SqlClient. Для MySQL: MySql.Data.MySqlClient.

* 1. Какие методы класса SqlCommand позволяют выполнить SQL-запрос?

Ответ: ExecuteReader() — для выполнения запросов на выборку. ExecuteNonQuery() — для команд, не возвращающих данных (INSERT, UPDATE, DELETE). ExecuteScalar() — для получения одиночного значения.

1. **Вывод**
   1. Научился создавать приложение C# для организации доступа к БД;
   2. Изучил 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. Научиться создавать приложение C# для организации доступа к БД;
   2. Научиться проектировать приложение, использующее паттерн репозиторий и Dapper.
2. **Ответы на контрольные вопросы**
   1. Что такое Dapper и для чего используется?

Ответ: Dapper — это ORM-библиотека для .NET, упрощающая выполнение SQL-запросов и маппинг данных в объекты.

* 1. Какие методы Dapper позволяют извлечь данные из БД?

Ответ: Query() — для выполнения запросов, возвращающих коллекцию данных. QueryFirst() — для получения первой записи. QuerySingle() — для получения единственной записи.

* 1. Какие методы Dapper позволяют изменить данные в БД?

Ответ: Execute() — для выполнения команд INSERT, UPDATE, DELETE.

* 1. Зачем используется паттерн «репозиторий»?

Ответ: Для абстракции доступа к данным, упрощения управления данными и уменьшения зависимости между бизнес-логикой и базой данных.

1. **Вывод**
   1. Научился создавать приложение C# для организации доступа к БД;
   2. Научился проектировать приложение, использующее паттерн репозиторий и Dapper.

# Лабораторная работа №16

**Применение ORM EF Core**

1. **Цель работы**
   1. Научиться создавать приложение C# для организации доступа к БД;
   2. Научиться разрабатывать приложение, использующее EF Core.
2. **Ответы на контрольные вопросы**
   1. Что такое «ORM»?

Ответ: ORM (Object-Relational Mapping) — это технология, связывающая объекты программирования с таблицами базы данных.

* 1. Что такое «EF Core» и для чего он предназначен?

Ответ: EF Core — это ORM-библиотека от Microsoft для .NET, предназначенная для работы с базами данных через объектную модель.

* 1. Как получить данные из БД, используя EF Core?

Ответ: Использовать метод DbContext.DbSet<TEntity>.ToList() или LINQ-запросы.

* 1. Как выполнить редактирование (вставку, обновление, удаление данных), используя EF Core?

Ответ: Вставка: DbContext.Add(entity) → DbContext.SaveChanges(). Обновление: Изменить свойства объекта → DbContext.SaveChanges(). Удаление: DbContext.Remove(entity) → DbContext.SaveChanges().

1. **Вывод**
   1. Научился создавать приложение C# для организации доступа к БД;
   2. Научился разрабатывать приложение, использующее EF Core.

# Лабораторная работа №17

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

1. **Цель работы**
   1. Научиться создавать приложение C# для организации доступа к БД средствами EF Core;
   2. Научиться выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.
2. **Ответы на контрольные вопросы**
   1. Для чего используются метод OrderBy, OrderByDescending, ThenBy, ThenByDescending и в чем их отличие?

Ответ: OrderBy — сортировка по возрастанию. OrderByDescending — сортировка по убыванию. ThenBy — дополнительная сортировка по возрастанию после первичной. ThenByDescending — дополнительная сортировка по убыванию после первичной.

* 1. Для чего используются методы Take и Skip и как они применяются при пагинации?

Ответ: Take(n) — извлекает первые n элементов из выборки. Skip(n) —пропускает первые n элементов. Они используются вместе для реализации постраничного вывода.

* 1. Для чего используется метод Where?

Ответ: Метод Where используется для фильтрации данных по заданным условиям.

* 1. Какие логические операторы могут применяться при составлении условий?

Ответ: AND, OR, NOT.

* 1. Как проверить, что значение есть в списке?

Ответ: Использовать метод Contains, например: list.Contains(value).

* 1. Как проверить, что строка начинается с определенного текста?

Ответ: Использовать метод StartsWith, например: string.StartsWith("text").

* 1. Как проверить, что строка содержит определенный текст?

Ответ: Использовать метод Contains, например: string.Contains("text").

* 1. Как указать в Select список требуемых данных?

Ответ: Использовать метод Select, например: dbContext.Entities.Select(e => new { e.Field1, e.Field2 }).

1. **Вывод**
   1. Научился создавать приложение C# для организации доступа к БД средствами EF Core;
   2. Научился выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.

# Лабораторная работа №18

**Разработка приложения для редактирования данных**

1. **Цель работы**
   1. Научиться выполнять вставку, обновление и удаление записей средствами EF Core;
   2. Научиться обеспечивать обратную связь при редактировании данных.
2. **Ответы на контрольные вопросы**
   1. Для чего используются методы Add() и AddRange() в EF Core?

Ответ: Add() добавляет одну запись в контекст. AddRange() добавляет несколько записей за один вызов.

* 1. Для чего используются методы Update() в EF Core?

Ответ: Update() обновляет состояние объекта в контексте для последующего сохранения изменений в БД.

* 1. Для чего используются методы Remove() и RemoveRange() в EF Core?

Ответ: Remove() удаляет одну запись из контекста. RemoveRange() удаляет несколько записей.

* 1. Как сохранить изменения в БД, используя EF Core?

Ответ: Вызвать метод DbContext.SaveChanges().

* 1. Как изменить значения полей объекта?

Ответ: Изменить свойства объекта и вызвать DbContext.SaveChanges().

* 1. Какое значение по умолчанию присваивается идентификатору нового объекта?

Ответ: Обычно 0 или null, в зависимости от типа.

* 1. Как передать объект с одной формы на другую?

Ответ: Использовать свойства или параметры конструктора для передачи объекта между формами.

1. **Вывод**
   1. Научился выполнять вставку, обновление и удаление записей средствами EF Core;
   2. Научился обеспечивать обратную связь при редактировании данных.

# Лабораторная работа №19

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

1. **Цель работы**
   1. Научиться выполнять SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.
2. **Ответы на контрольные вопросы**
   1. Как выполнить команду на выборку данных в EF Core?

Ответ: Использовать LINQ-запрос, например: dbContext.Entities.ToList().

* 1. Как выполнить команду на модификацию данных в EF Core?

Ответ: Изменить данные в объекте и вызвать DbContext.SaveChanges().

* 1. Как объявить и передать параметр в SQL-команду в EF Core?

Ответ: Использовать метод FromSqlRaw с параметрами, например:

dbContext.Entities.FromSqlRaw("SQL-запрос", параметры).

* 1. Как вызвать табличную функцию в EF Core?

Ответ: Использовать метод FromSqlRaw с вызовом функции.

* 1. Как вызвать хранимую процедуру в EF Core?

Ответ: Использовать метод FromSqlRaw, передав вызов процедуры.

* 1. Как объявить и передать выходной параметр из хранимой процедуры в EF Core?

Ответ: Использовать выходные параметры через SqlParameter и обработать их после выполнения команды.

1. **Вывод**
   1. Научился выполнять SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.

# Лабораторная работа №20

**Реализация разграничения прав доступа пользователей**

1. **Цель работы**
   1. Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;
   2. Научиться изменять настройки подключения к БД средствами Entity Framework Core.
2. **Ответы на контрольные вопросы**
   1. Как изменить настройки подключения к БД в клиентском приложении?

Ответ: Настройки подключения можно изменить в файле конфигурации приложения (например, appsettings.json) или передать в строку подключения вручную в коде.

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

Ответ: Создать таблицу пользователей с атрибутами (ID, имя, пароль и т. д.). Создать таблицу ролей. Создать таблицу связей "пользователи-роли" для реализации отношений "многие ко многим".

* 1. Что такое «авторизация»?

Ответ: Авторизация — это процесс проверки прав пользователя на выполнение определенных действий или доступ к ресурсам

* 1. Что такое «регистрация»?

Ответ: Регистрация — это процесс создания учетной записи пользователя в системе, включая предоставление данных для аутентификации.

1. **Вывод**
   1. Научился разграничивать права доступа пользователей на уровне интерфейса приложения;
   2. Научился изменять настройки подключения к БД средствами Entity Framework Core.

# Лабораторная работа №21

**Разработка приложения для импорта данных**

1. **Цель работы**
   1. Импорт данных пользователя в базу данных средствами EF Core.
2. **Ответы на контрольные вопросы**
   1. Как выполнить импорт из csv-файла в БД через приложение?

Ответ: Использовать библиотеки для работы с CSV (например, CsvHelper), прочитать файл и записать данные в БД через ORM или SQL-запросы.

* 1. Как импортировать изображение в БД через приложение?

Ответ: Считать изображение как массив байтов и сохранить его в поле типа varbinary или аналогичного.

* 1. Как выполнить импорт файла JSON в БД через приложение?

Ответ: Использовать библиотеки для работы с JSON (например, Newtonsoft.Json) для парсинга файла и ORM для записи данных в БД.

1. **Вывод**
   1. Импорт данных пользователя в базу данных средствами EF Core.

# Лабораторная работа №22

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

1. **Цель работы**
   1. Научиться выполнять экспорт данных из БД в формате текстового файла;
   2. Научиться выполнять экспорт данных из БД с применением Microsoft Office Word.
2. **Ответы на контрольные вопросы**
   1. Какое пространство имен требуется подключить для работы с Word?

Ответ: Microsoft.Office.Interop.Word.

* 1. Какие библиотеки позволяют сохранять данные в формате docx?

Ответ: OpenXML SDK, NPOI, ClosedXML.

* 1. Как выполнить экспорт данных в формате txt?

Ответ: Считать данные из БД и сохранить их в текстовый файл с использованием классов StreamWriter или File.

1. **Вывод**
   1. Научился выполнять экспорт данных из БД в формате текстового файла;
   2. Научился выполнять экспорт данных из БД с применением Microsoft Office Word.

# Лабораторная работа №23

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

1. **Цель работы**
   1. Научиться выполнять экспорт данных из БД в формате табличного файла;
   2. Научиться выполнять экспорт данных из БД с применением Microsoft Office Excel.
2. **Ответы на контрольные вопросы**
   1. Какое пространство имен требуется подключить для работы с Word?

Ответ: Microsoft.Office.Interop.Excel или альтернативные библиотеки (например, ClosedXML).

* 1. Какие библиотеки позволяют сохранять данные в формате docx?

Ответ: ClosedXML, EPPlus, NPOI.

* 1. Как выполнить экспорт данных в формате txt?

Ответ: Считать данные из БД и сохранить их в файл с использованием StreamWriter или File.

1. **Вывод**
   1. Научился выполнять экспорт данных из БД в формате табличного файла;
   2. Научился выполнять экспорт данных из БД с применением Microsoft Office Excel.

# Практическая работа №1

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

1. **Цель работы**
   1. Научиться выполнять разработку web-API для доступа к БД.
2. **Ответы на контрольные вопросы**
   1. Что такое REST-запрос?

Ответ: REST-запрос — это HTTP-запрос, соответствующий принципам архитектуры REST, используемый для взаимодействия с ресурсами веб-API.

* 1. Что такое RESTful?

Ответ: RESTful — это стиль разработки веб-сервисов, основанный на REST-принципах, таких как использование стандартных методов HTTP (GET, POST, PUT, DELETE) и четкая идентификация ресурсов через URL.

* 1. Для чего используется метод GET?

Ответ: Для получения данных с сервера.

* 1. Для чего используется метод POST?

Ответ: Для передачи данных на сервер и их создания.

* 1. Для чего используется метод PUT?

Ответ: Для обновления существующих данных на сервере.

* 1. Для чего используется метод DELETE?

Ответ: Для удаления данных с сервера.

1. **Вывод**
   1. Научился выполнять разработку web-API для доступа к БД.

# Практическая работа №2

**Вызов REST API**

1. **Цель работы**
   1. Научиться проверять работоспособность RESTful API в клиентском приложении;
   2. Научиться выполнять тестирование RESTful API методом черного ящика.
2. **Ответы на контрольные вопросы**
   1. Как указать у объекта HttpClient базовый адрес?

Ответ: Установить свойство BaseAddress объекта HttpClient, например: httpClient.BaseAddress = new Uri("https://api.example.com");.

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

Ответ: 200 OK — успешный запрос. 201 Created – успешно создан ресурс. 400 Bad Request – ошибка клиента. 401 Unauthorized – требуется авторизация. 404 Not Found – ресурс не найден. 500 Internal Server Error – ошибка сервера.

* 1. Какой метод класса HttpClient используется для получения данных?

Ответ: GetAsync().

* 1. Какой метод класса HttpClient используется для вставки данных?

Ответ: PostAsync()

* 1. Какой метод класса HttpClient используется для изменения данных?

Ответ: PutAsync()

* 1. Какой метод класса HttpClient используется для удаления данных?

Ответ: DeleteAsync()

1. **Вывод**
   1. Научиться проверять работоспособность RESTful API в клиентском приложении;
   2. Научиться выполнять тестирование RESTful API методом черного ящика.

# Практическая работа №3

1. **Цель работы**
   1. Научиться выполнять разработку web-API для доступа к БД;
   2. Научиться работать с JWT.
2. **Ответы на контрольные вопросы**
   1. Какие атрибуты можно указать у методов REST для настройки доступа для авторизованных и неавторизованных пользователей?

Ответ: [Authorize] — для доступа только авторизованных пользователей. [AllowAnonymous] — для доступа без авторизации.

* 1. Для чего используется JWT?

Ответ: JWT (JSON Web Token) используется для передачи данных между сторонами в формате токена с целью аутентификации и авторизации.

* 1. В чем отличие между авторизацией с использованием cookie и с использованием JWT с точки зрения безопасности?

Ответ: Cookie: Данные хранятся на клиенте и автоматически отправляются браузером на сервер. JWT: Токен передается явно, меньше зависимости от состояния сервера.

* 1. Какие настройки можно указать при создании токена?

Ответ: Срок действия, данные пользователя, ключ шифрования, алгоритм подписи.

* 1. Почему для авторизации следует использовать метод POST?

Ответ: Метод POST более безопасен, так как передает данные в теле запроса, а не в URL, снижая риск их утечки.

1. **Вывод**
   1. Научился выполнять разработку web-API для доступа к БД;
   2. Научился работать с JWT.

# Практическая работа №4

**Разработка веб-клиента**

1. **Цель работы**
   1. Научиться разрабатывать клиентское веб-приложение для доступа к БД;
   2. Научиться проверять работоспособность RESTful API в клиентском приложении.
2. **Ответы на контрольные вопросы**
   1. Как выполнить HTTP-запрос к API с помощью Fetch API?

Ответ: Использовать метод fetch(url, options), например:

fetch(“https://api.example.com/resource”, { method: “GET” })

.then(response => response.json())

.then(data => console.log(data));

* 1. Как обрабатывать ошибки при вызове API в JavaScript?

Ответ: Использовать конструкцию try...catch или проверять статус ответа в промисах:

fetch(url)

.then(response => {

if (!response.ok) throw new Error(response.statusText);

return response.json();

})

.catch(error => console.error(error));

* 1. Что такое асинхронные функции и как они связаны с вызовом API?

Ответ: Асинхронные функции (объявляемые с async) позволяют выполнять операции, которые могут занимать время (например, запросы к API), используя await для ожидания их завершения.

* 1. Что такое Razor Pages?

Ответ: Razor Pages — это модель разработки в ASP.NET Core для создания веб-страниц с серверной логикой и встроенным HTML-кодом.

* 1. Как создать и настроить новую страницу Razor в ASP.NET Core?

Ответ: Добавить файл .cshtml и связанный файл .cshtml.cs. Настроить маршрутизацию в файле Startup.cs или Program.cs.

1. **Вывод**
   1. Научился разрабатывать клиентское веб-приложение для доступа к БД;
   2. Научился проверять работоспособность RESTful API в клиентском приложении.