**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

# Отчеты по лабораторным и практическим работам

**по МДК.11.01**

Студент: Д.М. Мичурин

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: Для моделирования функциональных требований системы с точки зрения пользователя.

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

Ответ: Сущность, взаимодействующая с системой; обозначается стилизованным человечком.

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

Ответ: Функция системы, выполняемая для пользователя; обозначается овалом.

* 1. Что обозначает «отношение ассоциации»?

Ответ: Простое взаимодействие между актором и прецедентом.

* 1. Что обозначает «отношение обобщения»?

Ответ: Связь между общим и специфическим актором/прецедентом (наследование).

* 1. Что обозначает «отношение включения»?

Ответ: Обязательная часть одного прецедента в другом.

* 1. Что обозначает «отношение расширения»?

Ответ: Дополнительная функциональность к прецеденту при определенных условиях.

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: Объект, о котором нужно хранить данные.

* 1. Что такое «атрибут»?

Ответ: Свойство или характеристика сущности.

* 1. Что такое «ключевое поле»??

Ответ: Атрибут, однозначно идентифицирующий запись.

* 1. Каково назначение первичных и внешних ключей?

Ответ: Первичный ключ - уникальная идентификация записи; внешний ключ - связь между таблицами.

* 1. Что такое «связь»?

Ответ: Ассоциация между сущностями.

* 1. Какие виды связей между сущностями существуют?

Ответ: Один к одному, один ко многим, многие ко многим.

* 1. Какие элементы входят в ER-диаграммы?

Ответ: Сущности, атрибуты, связи, кардинальность.

* 1. Для чего применяются ER-диаграммы?

Ответ: Для моделирования структуры базы данных.

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: Атрибут(ы), однозначно идентифицирующий каждую запись в отношении.

* 1. Что называется внешним ключом отношения?

Ответ: Атрибут(ы), ссылающийся на первичный ключ в другом отношении.

* 1. В чем заключается процесс нормализации отношений?

Ответ: Процесс устранения избыточности и аномалий данных путём декомпозиции отношений.

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

Ответ: Когда для каждого значения B существует только одно значение A.

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

Ответ: Когда A зависит от C, а C зависит от B (A -> C -> B).

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

Ответ: Каждый атрибут должен быть атомарным (неделимым), нет повторяющихся групп.

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

Ответ: Должно быть в 1НФ и не иметь неключевых атрибутов, зависящих от части первичного ключа.

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

Ответ: Должно быть во 2НФ и не иметь транзитивной зависимости неключевых атрибутов от первичного ключа.

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: Обычно C:\Program Files\Microsoft SQL Server\MSSQLXX.MSSQLSERVER\MSSQL\DATA, где XX - версия.

MySQL: Обычно /var/lib/mysql (Linux) или C:\ProgramData\MySQL\MySQL Server X.X\data (Windows).

Postgres: Обычно /var/lib/postgresql/X.X/main (Linux), где X.X - версия.

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

Ответ:

* 1. Для чего используется Docker?

Ответ: Для контейнеризации приложений, упрощения развертывания и управления.

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: Инструмент для управления SQL Server и работы с базами данных.

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

Ответ: Windows Authentication и SQL Server Authentication.

* 1. Что такое первичный ключ?

Ответ: Уникальный идентификатор записи в таблице.

* 1. Как указать заполнение столбца автоинкрементными значениями?

Ответ: Использовать IDENTITY(seed, increment) 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 столбца или таблицы.

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

Ответ: Использовать UNIQUE ограничение.

* 1. Что такое внешний ключ?

Ответ: Ссылка на первичный ключ в другой таблице.

* 1. Какие значения может принимать внешний ключ?

Ответ: Значение первичного ключа из связанной таблицы или 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: Представление - это виртуальная таблица, основанная на запросе. От таблицы отличается тем, что не хранит данные физически.

* 1. Для чего применяются представления?

Ответ: Для упрощения сложных запросов, контроля доступа к данным, предоставления логически сгруппированной информации.

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

Ответ: Да, возможно.

* 1. Какие требование предъявляются к обновляемым представлениям?

Ответ: Должно быть основано на одной таблице, не содержать агрегатных функций, GROUP BY, DISTINCT, UNION, не иметь вычисляемых столбцов.

* 1. Как создать обновляемое представление?

Ответ: Используя CREATE VIEW AS SELECT с соблюдением условий обновляемости.

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: Пользовательские функции - это подпрограммы, созданные пользователем для выполнения конкретных задач. Применяются для повторного использования логики и расширения функционала SQL.

* 1. Чем отличается скалярная функция от табличной?

Ответ: Скалярная функция возвращает одно значение, табличная - набор строк (таблицу).

* 1. Как объявить переменную и присвоить ей значение?

Ответ: Используя DECLARE @имя\_переменной тип; и SET @имя\_переменной = значение;

* 1. Как указать в функции параметр по умолчанию?

Ответ: Используя параметр тип = значение\_по\_умолчанию в объявлении параметра.

* 1. Как вызвать скалярную функцию?

Ответ: Как часть выражения, например SELECT имя\_функции(аргументы);

* 1. Как вызвать табличную функцию?

Ответ: Как часть запроса SELECT \* FROM имя\_функции(аргументы); или JOIN имя\_функции(аргументы) ON ....

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 наборы SQL-инструкций, сохраненные в базе данных. Применяются для инкапсуляции логики, повышения производительности и безопасности.

* 1. Чем отличается функция пользователя от хранимой процедуры?

Ответ: Функция обычно возвращает одно значение или таблицу, а процедура может выполнять действия и возвращать несколько значений через выходные параметры.

* 1. Как задать входные параметры хранимой процедуры?

Ответ: Используя @имя\_параметра тип в объявлении процедуры.

* 1. Как задать выходные параметры хранимой процедуры?

Ответ: Используя @имя\_параметра тип OUTPUT в объявлении процедуры.

* 1. Как вызвать выполнение хранимой процедуры?

Ответ: Используя EXEC имя\_процедуры @параметр1 = значение1, ...; или просто EXEC имя\_процедуры значение1, ...;

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 специальная хранимая процедура, которая выполняется автоматически в ответ на определённые события в базе данных, например, вставка, изменение или удаление данных в таблице.

* 1. Чем триггер отличается от хранимой процедуры?

Ответ: Триггер запускается автоматически при определённом событии, в то время как хранимую процедуру нужно вызывать явно. Триггеры обычно выполняют действия, связанные с изменениями данных в таблице, а хранимые процедуры могут выполнять различные задачи, не только связанные с данными.

* 1. Как запустить триггер на выполнение?

Ответ: Триггеры запускаются автоматически при наступлении события, на которое они реагируют (например, при INSERT, UPDATE или DELETE). Их не нужно вызывать явно.

* 1. Каково назначение триггеров?

Ответ: Триггеры используются для поддержания целостности данных, ведения аудита изменений, выполнения сложных проверок на входе или выходе, и для многих других задач, связанных с обновлением данных. Они обеспечивают автоматическое реагирование на изменения данных, без необходимости дополнительного кода или запросов.

* 1. Чем отличаются триггеры INSTEAD OF и AFTER?

Ответ: Триггеры INSTEAD OF изменяют событие, заменяя его своим выполнением. Триггеры AFTER выполняют действие после наступления события. Ключевое отличие заключается в том, что триггеры INSTEAD OF блокируют исходное событие, в то время как триггеры AFTER позволяют исходному событию выполниться, а затем уже запускают триггер. Триггеры INSTEAD OF особенно полезны, когда нужно изменить поведение SQL-операции.

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. ;
   3. .
2. **Ответы на контрольные вопросы**
   1. В чем отличие между именами входа и пользователями БД?

Ответ: Имя входа (логин) - это аутентификация на уровне сервера, а пользователь БД - это аутентификация внутри конкретной базы данных.

* 1. Как идентифицируются пользователи в MS SQL Server?

Ответ: Через имена входа (логины) и пользователей, связанных с этими логинами в базах данных.

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

Ответ: На уровень сервера (логины, роли сервера) и уровень базы данных (пользователи, роли БД).

* 1. Каково назначение ролей сервера?

Ответ: Для управления доступом к ресурсам сервера и административным функциям (например, создание БД, управление логинами).

* 1. Каково назначение ролей БД?

Ответ: Для управления доступом к объектам и операциям внутри базы данных (например, чтение, запись, выполнение).

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, ошибок, повреждения или потери данных.

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

Ответ: Полное копирование сохраняет все данные, а разностное - только изменения с момента последнего полного копирования.

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

Ответ: Полное: Раз в неделю (например, в воскресенье). \* Разностное: Ежедневно (например, в будние дни) или через день.

* 1. Какие скрипты можно сформировать для объектов БД?

Ответ: Скрипты для создания таблиц, представлений, процедур, функций, триггеров, индексов и т.д.

* 1. Как выполнить восстановление БД из резервной копии?

Ответ: Используя средства управления БД (например, SQL Server Management Studio) или SQL-команды.

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

Ответ: RESTORE DATABASE (в T-SQL).

* 1. В каком порядке надо восстанавливать резервные копии?

Ответ: Сначала полное, затем последовательно разностные в хронологическом порядке.

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

Ответ: Параметр RECOVERY/NORECOVERY (в RESTORE DATABASE).

* 1. Для чего используется мастер импорта и экспорта?

Ответ: Для переноса данных между различными источниками (файлы, другие БД) и для конвертации форматов.

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: Через контекстное меню на таблице (Tasks -> Export Data) или с помощью SQL-запроса и утилиты bcp.

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

Ответ: Через контекстное меню на таблице (Tasks -> Export Data) или через использование Linked Server для подключения к Excel.

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

Ответ: Через “Данные” -> “Из текста/CSV” или “Из XML” на вкладке “Данные”.

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

Ответ: Обычно данные из одной таблицы.

* 1. Как экспортировать данные в файл формата JSON?

Ответ: Используя SQL-запрос и функцию FOR JSON (доступно в MS SQL Server 2016 и новее), а затем выводя результат в файл.

* 1. Как экспортировать данные в файл формата XML?

Ответ: Используя SQL-запрос и функцию FOR XML, а затем выводя результат в файл.

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: Через контекстное меню на базе данных (Tasks -> Import Data) или через использование Linked Server для подключения к Excel и импорта данных.

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

Ответ: Через контекстное меню на базе данных (Tasks -> Import Data) или с помощью SQL-запроса BULK INSERT.

* 1. Как выполнить импорт файла XML?

Ответ: Используя OPENROWSET(BULK ...) или OPENXML.

* 1. Как выполнить импорт файла JSON?

Ответ: Используя OPENJSON (доступно в MS SQL Server 2016 и новее) или импортируя через SSIS.

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: Устанавливает соединение с базой данных SQL Server.

* 1. Какое назначение у элемента SqlCommand?

Ответ: Представляет SQL-запрос или хранимую процедуру для выполнения.

* 1. Какое назначение у элемента SqlDataReader?

Ответ: Предоставляет поток чтения данных, полученных из базы данных, после выполнения запроса.

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

Ответ: MS SQL Server: System.Data.SqlClient

MySQL: MySql.Data.MySqlClient

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

Ответ: \* ExecuteNonQuery() - для запросов, не возвращающих данных (INSERT, UPDATE, DELETE). \* ExecuteReader() - для запросов, возвращающих данные (SELECT). \* ExecuteScalar() - для запросов, возвращающих одно значение (например, COUNT).

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 (Object-Relational Mapper), который используется для простого и быстрого доступа к данным в базе данных из .NET приложений, является микро-ORM, акцентирующимся на производительности.

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

Ответ: Query<T>() - для запросов, возвращающих коллекцию объектов типа T.

QuerySingle<T>(), QuerySingleOrDefault<T>() - для запросов, возвращающих один объект типа T.

QueryFirst<T>(), QueryFirstOrDefault<T>() - для запросов, возвращающих первый объект из коллекции типа T.

Query() - для динамических результатов.

QueryMultiple() - для запросов, возвращающих несколько наборов результатов.

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

Ответ: Execute() - для запросов, не возвращающих данные (INSERT, UPDATE, DELETE).

* 1. Зачем используется паттерн «репозиторий»?

Ответ: Для инкапсуляции логики доступа к данным, отделения ее от бизнес-логики приложения, улучшения тестируемости и поддерживаемости.

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 Mapper) — это инструмент, который преобразует данные между объектно-ориентированным представлением (в коде) и реляционным представлением (в базе данных).

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

Ответ: EF Core (Entity Framework Core) — это кроссплатформенный ORM от Microsoft для .NET, предназначенный для упрощения работы с базами данных, путем предоставления удобного API и механизмов для взаимодействия с реляционными СУБД.

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

Ответ: Используя методы DbSet<T> (например, context.Users.ToList(), context.Users.Where(u => u.Id == 1).FirstOrDefault())

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

Ответ: \* Вставка: context.Add(entity); context.SaveChanges(); \* Обновление: context.Update(entity); context.SaveChanges(); \* Удаление: context.Remove(entity); context.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: Сортирует коллекцию по возрастанию. \* OrderByDescending: Сортирует коллекцию по убыванию. \* ThenBy: Указывает дополнительную сортировку по возрастанию в рамках предыдущей сортировки. \* ThenByDescending: Указывает дополнительную сортировку по убыванию в рамках предыдущей сортировки. \* Отличие: OrderBy и OrderByDescending начинают сортировку, а ThenBy и ThenByDescending добавляют вторичную сортировку.

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

Ответ: \* Take: Возвращает указанное количество элементов с начала последовательности. \* Skip: Пропускает указанное количество элементов и возвращает остальные. \* Применяются при пагинации для выборки данных для конкретной страницы.

* 1. Для чего используется метод Where?

Ответ: Для фильтрации последовательности на основе заданного условия.

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

Ответ: && (AND), || (OR), ! (NOT).

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

Ответ: Используя метод Contains() (например, myList.Contains(value)).

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

Ответ: Используя метод StartsWith() (например, myString.StartsWith("text")).

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

Ответ: Используя метод Contains() (например, myString.Contains("text")).

* 1. Как указать в Select список требуемых данных?

Ответ: Указав анонимный объект, свойства которого соответствуют необходимым полям (например, Select(x => new { x.Name, x.Email })) или используя проекцию.

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 один новый объект в контекст для последующей вставки в базу данных. \* AddRange(): Добавляет несколько новых объектов в контекст для последующей вставки в базу данных.

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

Ответ: Помечает существующий объект в контексте для последующего обновления в базе данных.

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

Ответ: \* Remove(): Помечает один существующий объект в контексте для последующего удаления из базы данных. \* RemoveRange(): Помечает несколько существующих объектов в контексте для последующего удаления из базы данных.

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

Ответ: Вызвать метод SaveChanges() у контекста (например, context.SaveChanges()).

* 1. Как изменить значения полей объекта?

Ответ: Просто присвоив новые значения соответствующим свойствам объекта (например, myObject.Name = "New Name";).

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

Ответ: Зависит от типа идентификатора, если это автоинкрементное число, то по умолчанию база данных присвоит значение автоматически. Если это GUID, то создастся новый GUID.

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

Ответ: Можно передать объект как параметр в конструктор новой формы или через свойство новой формы.

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 методы DbSet<T> и LINQ-запросы (например, context.Users.ToList(), context.Users.Where(u => u.Id == 1).FirstOrDefault())

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

Ответ: С помощью методов Add(), Update(), Remove() и последующего вызова SaveChanges().

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

Ответ: Использовать FromSqlRaw или FromSqlInterpolated, передавая параметры как именованные аргументы или в формате string interpolation (например, context.Users.FromSqlRaw("SELECT \* FROM Users WHERE Id = {0}", userId))

* 1. Как вызвать табличную функцию в EF Core?

Ответ: Использовать FromSqlRaw или FromSqlInterpolated с указанием SQL-запроса, вызывающего функцию (например, context.Set<MyView>().FromSqlRaw("SELECT \* FROM MyFunction(@parameter)", parameter))

* 1. Как вызвать хранимую процедуру в EF Core?

Ответ: Использовать FromSqlRaw или FromSqlInterpolated с SQL-запросом для вызова процедуры, а для получение результата использовать ExecuteSqlRaw (например, context.Database.ExecuteSqlRaw("EXEC MyStoredProcedure @Parameter", parameter)).

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

Ответ: Через использование SqlParameter, а результат прочитать через свойство Value. Применяется вместе с context.Database.ExecuteSqlRaw

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: \* Файл конфигурации: Изменив строку подключения (connection string) в файле конфигурации (например, appsettings.json или web.config). \* Настройки приложения: Предоставив пользователю интерфейс для изменения параметров подключения и сохранения их в настройках приложения. \* Переменные среды: Настройка строки подключения через переменные среды, что часто используется в облачных окружениях.

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

Ответ: Таблица с информацией о пользователях (UserID - Primary Key, UserName, Password, Email и другие данные пользователя). \* Roles: Таблица с ролями (RoleID - Primary Key, RoleName - название роли). \* UserRoles: Связующая таблица между Users и Roles, чтобы задать какие роли у пользователя. (UserID, RoleID - Composite Foreign key, ссылающийся на Users и Roles).

* 1. Что такое «авторизация»?

Ответ: Процесс проверки, имеет ли пользователь право доступа к определенным ресурсам или функциям после успешной аутентификации.

* 1. Что такое «регистрация»?

Ответ: Процесс создания нового пользователя в системе, включающий ввод данных (имя, пароль, email и др.) и их сохранение для последующей 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: Прочитать содержимое CSV-файла с помощью классов StreamReader или библиотек для работы с CSV (например, CsvHelper)

* 1. Как импортировать изображение в БД через приложение?

Ответ: Прочитать содержимое файла изображения в виде массива байтов (byte[]) используя File.ReadAllBytes().

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

Ответ: Прочитать содержимое JSON-файла с помощью StreamReader или File.ReadAllText()

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: Для работы с Word через COM-интероп (старый способ) используется Microsoft.Office.Interop.Word. Но это не рекомендуется для серверных приложений и может потребовать установки Microsoft Office. Для более современных и кроссплатформенных решений используются другие библиотеки.

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

Ответ: Open XML SDK: Официальная библиотека от Microsoft для работы с форматами Office Open XML (.docx, .xlsx, .pptx). Подходит для генерации и модификации документов. \* DocX: Простая библиотека для работы с docx-файлами, предоставляющая более удобный API, чем Open XML SDK. \* NPOI: Кроссплатформенная библиотека для работы с форматами Microsoft Office (xls, xlsx, doc, docx, ppt, pptx).

* 1. Как выполнить экспорт данных в формате txt?

Ответ: 1. Создание или открытие файла: Используя File.Create() или File.Open() для создания нового или открытия существующего текстового файла. 2. Запись данных: Используя StreamWriter для записи строк в файл (например, using (StreamWriter writer = new StreamWriter(filePath)) { writer.WriteLine("My text"); }). 3. Закрытие файла: (Обеспечивается с помощью using).

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: Для работы с Word через COM-интероп (старый способ) используется Microsoft.Office.Interop.Word. Однако, как я уже упоминал ранее, этот подход не рекомендуется для серверных приложений и требует наличия установленного Microsoft Office. Для более современных и кроссплатформенных решений следует использовать сторонние библиотеки.

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

Ответ: Open XML SDK, DocumentFormat.OpenXml.

* 1. Как выполнить экспорт данных в формате txt?

Ответ: Создание или открытие файла: Используйте File.Create() или File.Open() для создания нового или открытия существующего текстового файла.

Запись данных: Используйте StreamWriter для записи данных в файл построчно

Закрытие файла: Блок using гарантирует, что файл будет закрыт и ресурсы освобождены после завершения работы с StreamWriter.

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 (Representational State Transfer) — это запрос к веб-сервису, который соответствует принципам REST, включая использование HTTP-методов (GET, POST, PUT, DELETE), URI для идентификации ресурсов и стандартные форматы данных (например, JSON, XML) для обмена информацией.

* 1. Что такое RESTful?

Ответ: RESTful - это архитектурный стиль веб-сервисов, который соответствует принципам REST и использует стандарты HTTP для предоставления доступа к ресурсам. RESTful API - это API, разработанный в стиле REST.

* 1. Для чего используется метод GET?

Ответ: Метод GET используется для получения (чтения) данных с сервера по указанному URI.

* 1. Для чего используется метод POST?

Ответ: Метод POST используется для создания нового ресурса на сервере или отправки данных для обработки.

* 1. Для чего используется метод PUT?

Ответ: Метод PUT используется для обновления (замены) существующего ресурса на сервере по указанному URI.

* 1. Для чего используется метод DELETE?

Ответ: Метод DELETE используется для удаления ресурса на сервере по указанному URI.

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: При создании экземпляра HttpClient можно задать базовый адрес в его конструкторе через свойство BaseAddress:

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

Ответ: HTTP: \* 200 OK: Запрос выполнен успешно. \* 201 Created: Ресурс успешно создан (обычно после POST). \* 204 No Content: Запрос успешно выполнен, но нет содержимого для отправки в ответе (например, после DELETE). \* 301 Moved Permanently: Ресурс перемещен на новый URI. \* 400 Bad Request: Запрос не может быть обработан из-за синтаксической ошибки или неверных данных. \* 401 Unauthorized: Запрос требует аутентификации. \* 403 Forbidden: Доступ к ресурсу запрещен. \* 404 Not Found: Ресурс не найден по указанному URI. \* 500 Internal Server Error: Внутренняя ошибка на сервере. \* 503 Service Unavailable: Сервер временно недоступен.

* 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: В .NET API (например, ASP.NET Core) используются следующие атрибуты: \* [Authorize]: Требует, чтобы пользователь был аутентифицирован (вошел в систему). Может применяться к контроллеру или отдельному методу. \* [AllowAnonymous]: Разрешает доступ к методу без аутентификации. Применяется, когда нужно отменить действие [Authorize] для отдельного метода в авторизованном контроллере. \* [Authorize(Roles = "Admin,User")]: Требует аутентификацию и наличие у пользователя одной из указанных ролей (в данном примере Admin или User). \* [Authorize(Policy = "MyPolicy")]: Требует, чтобы пользователь соответствовал заданной политике авторизации.

* 1. Для чего используется JWT?

Ответ: JWT (JSON Web Token) — это стандарт для создания токенов доступа, которые представляют собой JSON-объекты с цифровой подписью. JWT используется для безопасной передачи информации об аутентификации и авторизации между клиентом и сервером.

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

Ответ: Cookies обычно используются для аутентификации на основе сессий. Сессии хранятся на сервере, а браузер отправляет cookie-файл в каждом запросе. Недостатки: уязвимость к CSRF-атакам (Cross-Site Request Forgery), сложность при работе с несколькими серверами.

* 1. Какие настройки можно указать при создании токена?

Ответ: При создании JWT можно указать: \* Issuer (издатель): Кто выдает токен. \* Audience (получатель): Для кого предназначен токен. \* Subject (субъект): Идентификатор пользователя. \* Expiration (время истечения срока действия): Дата и время, когда токен станет недействительным. \* Claims (утверждения): Дополнительные данные, такие как имя, роль и т. д. \* Signing key (секретный ключ): Используется для подписи токена. \* Algorithm (алгоритм подписи): Например, HS256, RS256.

* 1. Почему для авторизации следует использовать метод POST?

Ответ: Для авторизации (аутентификации) следует использовать метод POST, потому что: \* Безопасность: Данные для аутентификации (логин, пароль) передаются в теле запроса, а не в URL, что более безопасно. \* Семантика: POST предназначен для отправки данных на сервер для обработки и создания нового ресурса (в данном случае, создание токена доступа или создание сессии). \* Кеширование: GET запросы часто кешируются, что делает нежелательным использование его для аутентификации.

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:

fetch('https://api.example.com/data', {

method: 'GET', // Или POST, PUT, DELETE

headers: {

'Content-Type': 'application/json',

'Authorization': 'Bearer YOUR\_TOKEN' // Если требуется

},

// body: JSON.stringify({ key: 'value' }) // Если POST, PUT

})

.then(response => {

if (!response.ok) {

throw new Error(`HTTP error! status: ${response.status}`);

}

return response.json(); // Или response.text(), response.blob()

})

.then(data => {

console.log(data); // Обработка полученных данных

})

.catch(error => {

console.error('Error:', error); // Обработка ошибок

});

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

Ответ: try...catch блоки: Для перехвата ошибок, возникающих при выполнении кода. \* Проверка response.ok: Проверка статуса HTTP ответа. Если response.ok is false нужно выбросить исключение throw new Error(). \* catch в цепочке Promise: Для перехвата ошибок, возникающих при выполнении fetch или в обработчиках then. \* Обработка статуса ошибки: Вывод сообщения об ошибке на экран или запись в журнал.

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

Ответ: Асинхронные функции (async) в JavaScript позволяют писать код, который не блокирует основной поток выполнения при выполнении длительных операций, таких как вызов API. \* await: Ключевое слово await используется внутри async функции, чтобы “дождаться” завершения Promise (например, результата fetch) и вернуть его значение. \* Связь с API: При вызове API через fetch мы получаем Promise, который будет разрешен с результатом или отклонен с ошибкой. Используя async/await, мы делаем работу с промисами более читаемой и простой.

* 1. Что такое Razor Pages?

Ответ: Razor Pages — это модель разработки веб-приложений в ASP.NET Core, основанная на страницах. Каждая страница Razor представляет собой файл .cshtml, сочетающий разметку HTML с кодом C#. Это упрощает разработку отдельных страниц с их собственным поведением и бизнес-логикой.

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

Ответ: Создание файла: В папке Pages проекта создайте новый файл с расширением .cshtml (например, MyPage.cshtml).

Добавление модели страницы: Создайте файл с именем MyPage.cshtml.cs и определите класс, который наследуется от PageModel (например, public class MyPageModel : PageModel { ... }).

Разметка: В файле MyPage.cshtml разместите HTML-код, используя Razor синтаксис для внедрения кода C# (например, @Model.Message).

Код: В файле MyPage.cshtml.cs можно добавить методы обработки запросов (OnGet, OnPost и другие), свойства и логику страницы.

Доступ: Страница будет доступна по URL, который соответствует ее пути (например, /MyPage).

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