**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**по МДК.11.01**

Студент: Н.Д. Попов

Группа: ИСПП-21

Преподаватель: Ю.С. Маломан

Архангельск 2024

**Лабораторная работа №1**

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

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

Ответ: Определение функциональности системы, Упрощение общения с заказчиком, Моделирование поведения системы, Определение ролей пользователей, База для проектирования.

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

Ответ: это сущность, которая взаимодействует с системой, изображается в виде человечка.

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

Ответ: это описание конкретной функциональности системы, которая предоставляет ценность актору, изображается в виде овалов (эллипсов).

* Что обозначает «отношение ассоциации»?

Ответ: это связь, которая показывает взаимодействие между актором и прецедентом на диаграмме вариантов использования.

* Что обозначает «отношение обобщения»?

Ответ: на диаграмме вариантов использования обозначает наследование или иерархическую связь между элементами, такими как акторы или прецеденты.

* Что обозначает «отношение включения»?

Ответ: в диаграмме вариантов использования указывает, что один прецедент всегда использует функциональность другого прецедента.

* Что обозначает «отношение расширения»?

Ответ: в диаграмме вариантов использования показывает, что дополнительное (расширенное) поведение может быть добавлено к основному прецеденту при выполнении определённых условий.

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

**Лабораторная работа №2**

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

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

Ответ: это фундаментальная концепция, используемая в моделировании и проектировании систем, особенно в базах данных и диаграммах.

* Что такое «атрибут»?

Ответ: это характеристика или свойство, которое описывает сущность.

* Что такое «ключевое поле»??

Ответ: это атрибут (или комбинация атрибутов) в базе данных, который используется для уникальной идентификации записи в таблице.

* Каково назначение первичных и внешних ключей?

Ответ: играют важную роль в организации, идентификации и связывании данных в реляционных базах данных.

* Что такое «связь»?

Ответ: в контексте реляционных баз данных и моделирования данных обозначает взаимосвязь между двумя или более сущностями (объектами).

* Какие виды связей между сущностями существуют?

Ответ: Один к одному (1:1), Один ко многим (1:N), Многие ко многим (M:N), Самосвязь (рефлексивная связь),Сложная связь (составная связь).

* Какие элементы входят в ER-диаграммы?

Ответ: Сущности (Entities), Атрибуты (Attributes), Связи (Relationships), Кардинальности связей (Cardinality), Ключевые атрибуты (Key Attributes), Сложные связи (Composite Relationships), Иерархические или обобщённые связи (Generalization/ Specialization).

* Для чего применяются ER-диаграммы?

Ответ: Проектирование базы данных,Понимание и визуализация структуры данных, Обеспечение целостности данных, Документация системы.

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

**Лабораторная работа №3**

**Приведение БД к нормальной форме**

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

Ответ: это один или несколько атрибутов (полей) в таблице базы данных, которые уникально идентифицируют каждую запись (строку) в этой таблице.

* Что называется внешним ключом отношения?

Ответ: это атрибут или набор атрибутов в таблице базы данных, который ссылается на первичный ключ другой таблицы.

* В чем заключается процесс нормализации отношений?

Ответ: это процесс организации данных в базе данных для устранения избыточности и зависимости между данными, а также для улучшения целостности данных.

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

Ответ: Атрибут А функционально зависит от атрибута В, если для каждого значения атрибута В существует только одно значение атрибута А.

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

Ответ: Атрибут А транзитивно зависит от атрибута В, если существует промежуточный атрибут С, такой что:

В → С (атрибут С функционально зависит от атрибута В),

С → А (атрибут А функционально зависит от атрибута С).

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

Ответ: Отношение считается находящимся в первой нормальной форме (1НФ), если оно удовлетворяет следующим требованиям: Атомарность атрибутов, Однозначность строк, Отсутствие повторяющихся групп.

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

Ответ: Отношение считается находящимся во второй нормальной форме (2НФ), если оно удовлетворяет следующим требованиям: Соответствие первой нормальной форме (1НФ), Отсутствие частичных зависимостей.

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

Ответ: Отношение считается находящимся в третьей нормальной форме (3НФ), если оно удовлетворяет следующим требованиям: Соответствие второй нормальной форме (2НФ), Отсутствие транзитивных зависимостей.

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

**Лабораторная работа №4**

**Установка SQL сервера**

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

Ответ: MSSQL (Microsoft SQL Server):

Стандартный порт: 1433 (TCP)

MySQL:

Стандартный порт: 3306 (TCP)

PostgreSQL:

Стандартный порт: 5432 (TCP)

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

Ответ: 1)Стандартная папка для данных:

На Windows:

C:\ProgramFiles\MicrosoftSQLServe\MSSQLXX.MSSQLSERVER\MSSQL\DATA\ где XX — версия SQL Server (например, MSSQL15 для SQL Server 2019).

В этой папке хранятся файлы базы данных с расширениями .mdf (основной файл данных) и .ldf (журнал транзакций).

Стандартная папка для журналов транзакций:

Обычно они находятся в той же папке, что и файлы данных:

C:\ProgramFiles\MicrosoftSQLServe\MSSQLXX.MSSQLSERVER\MSSQL\DATA\

2)Стандартная папка для данных:

На Linux: /var/lib/mysql/

На Windows: C:\ProgramData\MySQL\MySQL Server X.X\data\

где X.X — версия MySQL (например, 5.7, 8.0).

3)Стандартная папка для данных:

На Linux: /var/lib/postgresql/X.X/main/

где X.X — версия PostgreSQL (например, 12, 13).

На Windows:

C:\Program Files\PostgreSQL\X.X\data\

где X.X — версия PostgreSQL.

В этой папке хранятся все данные PostgreSQL, включая файлы баз данных и журналы транзакций.

В этой папке хранятся файлы базы данных, а также различные файлы журналов.

* Для чего используется Docker?

Ответ: это платформа для разработки, доставки и запуска приложений в контейнерах. Контейнеры позволяют упаковывать приложения и их зависимости в единый исполнимый пакет, который можно запускать на любой системе, поддерживающей Docker, независимо от конфигурации этой системы.

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

**Лабораторная работа №5**

**Создание базы данных в среде разработки**

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

Ответ: это интегрированная среда для управления базами данных Microsoft SQL Server. Она предоставляет графический интерфейс для работы с базами данных, создания и выполнения SQL-запросов, администрирования серверов и баз данных.

* Какие виды авторизации поддерживаются в MS SQL Server?

Ответ: В MS SQL Server поддерживаются два типа авторизации: Windows-авторизация и смешанная авторизация (SQL Server и Windows).

* Что такое первичный ключ?

Ответ: Первичный ключ — это уникальный идентификатор записи в таблице базы данных, который гарантирует, что каждое значение в столбце или комбинации столбцов будет уникальным и не NULL.

* Как указать заполнение столбца автоинкрементными значениями?

Ответ Для автоинкремента используется атрибут IDENTITY при создании столбца.

* Как указать значение по умолчанию?

Ответ: Для указания значения по умолчанию используется ограничение DEFAULT.

* Как задать проверочное ограничение?

Ответ Для создания проверочного ограничения используется ключевое слово CHECK.

* Как обеспечить уникальность значений в столбце или наборе столбцов?

Ответ: Для обеспечения уникальности используется ограничение UNIQUE.

* Что такое внешний ключ?

Ответ: Внешний ключ — это ограничение, которое обеспечивает ссылочную целостность между таблицами, связывая одну таблицу с другой через атрибуты, которые ссылаются на первичный ключ другой таблицы.

* Какие значения может принимать внешний ключ?

Ответ: Внешний ключ может принимать значения, которые существуют в связанной таблице (таблице с первичным ключом), или быть NULL, если разрешена такая настройка.

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

**Лабораторная работа №6**

**Создание представлений в СУБД**

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

Ответ: Представление — это виртуальная таблица, которая хранит результат выполнения SQL-запроса. Представления не содержат данных, в отличие от таблиц, которые хранят данные.

* Для чего применяются представления?

Ответ: Представления используются для упрощения работы с базой данных, объединения нескольких таблиц, фильтрации данных, скрытия сложной логики запроса от пользователей.

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

Ответ: Да, представление может объединять данные из нескольких таблиц, используя операторы JOIN в SQL-запросах.

* Какие требование предъявляются к обновляемым представлениям?

Ответ: Обновляемые представления должны включать столбцы из одной таблицы и не содержать агрегаций, группировок или подзапросов.

* Как создать обновляемое представление?

Ответ: Обновляемое представление создается с использованием простых запросов.

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

**Лабораторная работа №7**

**Создание функций пользователя в СУБД**

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

Ответ: Функции пользователя — это SQL-программы, которые выполняют определенную задачу и могут возвращать значение (скалярные функции) или таблицу (табличные функции). Они применяются для повторного использования логики и упрощения запросов.

* Чем отличается скалярная функция от табличной?

Ответ: Скалярная функция возвращает одно значение, а табличная функция возвращает набор данных в виде таблицы.

* Как объявить переменную и присвоить ей значение?

Ответ: Для объявления переменной используется ключевое слово DECLARE, а для присваивания значения — SET.

* Как указать в функции параметр по умолчанию?

Ответ: Для параметра по умолчанию используется оператор DEFAULT.

* Как вызвать скалярную функцию?

Ответ: Скалярную функцию можно вызвать как часть SQL-запроса SELECT dbo.MyFunction(10);

* Как вызвать табличную функцию?

Ответ: SELECT \* FROM dbo.MyTableFunction();

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

**Лабораторная работа №8**

**Создание хранимых процедур в СУБД**

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

Ответ: Хранимые процедуры — это предопределенные SQL-запросы, которые могут принимать параметры, выполнять операции с данными и возвращать результаты. Они используются для автоматизации задач и повышения производительности.

* Чем отличается функция пользователя от хранимой процедуры?

Ответ: Функция возвращает значение (скалярное или таблицу), а хранимая процедура выполняет действия и может изменять данные.

* Как задать входные параметры хранимой процедуры?

Ответ: ходные параметры задаются при объявлении процедуры.

* Как задать выходные параметры хранимой процедуры?

Ответ: ля выходных параметров используется ключевое слово OUTPUT.

* Как вызвать выполнение хранимой процедуры?

Ответ: Хранимую процедуру можно вызвать с помощью оператора EXEC.

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

**Лабораторная работа №9**

**Создание триггеров в СУБД**

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

Ответ: Триггер — это объект базы данных, который автоматически выполняет заданное действие при наступлении определённого события в базе данных (например, при вставке, обновлении или удалении данных).

* Чем триггер отличается от хранимой процедуры?

Ответ: Хранимая процедура вызывается вручную или по расписанию, тогда как триггер выполняется автоматически в ответ на изменение данных в базе.

* Как запустить триггер на выполнение?

Ответ: Триггер запускается автоматически в ответ на событие, такое как вставка, обновление или удаление записи в таблице.

* Каково назначение триггеров?

Ответ: Триггеры используются для автоматизации выполнения определённых операций, таких как поддержание целостности данных, аудита изменений или выполнения дополнительных действий при изменении данных.

* Чем отличаются триггеры INSTEAD OF и AFTER?

Ответ: Триггер INSTEAD OF выполняется вместо действия (например, вставки), а AFTER выполняется после выполнения действия.

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

**Лабораторная работа №10**

**Реализация доступа пользователей к базе данных**

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

Ответ: Имя входа (login) — это учётная запись для доступа к серверу SQL, тогда как пользователь БД (user) — это учётная запись для работы с конкретной базой данных.

* Как идентифицируются пользователи в MS SQL Server?

Ответ: Пользователи в MS SQL Server идентифицируются через имя входа, которое связано с конкретной базой данных.

* На какие уровни разделяется система безопасности MS SQL Server?

Ответ: Система безопасности делится на уровень сервера и уровень базы данных.

* Каково назначение ролей сервера?

Ответ: Роли сервера задают права на уровне сервера для управления доступом и безопасности.

* Каково назначение ролей БД?

Ответ: Роли БД управляют правами пользователей на уровне конкретной базы данных, например, права на создание таблиц или выполнение запросов.

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

**Лабораторная работа №11**

**Выполнение резервного копирования и восстановления БД**

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

Ответ: Резервные копии создаются для защиты данных от потери в случае сбоев или аварийных ситуаций.

* В чем отличие между полным и разностным резервным копированием?

Ответ: Полное резервное копирование сохраняет все данные, а разностное — только данные, изменённые после последнего полного копирования.

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

Ответ: Полное копирование выполняется периодически (например, раз в неделю), а разностное — ежедневно или чаще в зависимости от изменений.

* Какие скрипты можно сформировать для объектов БД?

Ответ: Для объектов БД можно сформировать скрипты для создания, модификации и удаления объектов, а также для вставки данных.

* Как выполнить восстановление БД из резервной копии?

Ответ: Восстановление выполняется с помощью команды RESTORE DATABASE.

* Какая команда выполняет восстановление БД из резервной копии?

Ответ: Для восстановления базы данных (БД) из резервной копии в SQL Server используется команда RESTORE DATABASE

* В каком порядке надо восстанавливать резервные копии?

Ответ: В простых случаях для восстановления всей базы данных до точки сбоя начинают с резервного копирования журнала действующих транзакций (заключительного фрагмента журнала). Затем восстанавливают последнюю полную резервную копию базы данных, последнюю разностную резервную копию базы данных (если она есть), а потом все последующие резервные копии журналов в порядке их выполнения.

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

Ответ: RESTORE WITH RECOVERY. Восстанавливает базу данных после восстановления последней резервной копии. Это параметр по умолчанию.

RESTORE WITH NORECOVERY. Оставляет базу данных в состоянии восстановления. Это позволяет восстановить дополнительные резервные копии в текущем пути восстановления.

RESTORE WITH STANDBY. Оставляет базу данных в состоянии резерва, в котором база данных доступна в ограниченном режиме только для чтения.

* Для чего используется мастер импорта и экспорта?

Ответ: используется для копирования данных из источника в место назначения.

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

**Лабораторная работа №12**

**Экспорт данных базы в документы пользователя**

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

Ответ: Для экспорта данных в csv-файл используется мастер импорта и экспорта в SQL Server Management Studio (SSMS), выбирая формат CSV.

* Как выполнить экспорт xlsx-файла в Management Studio?

Ответ: Для экспорта в формат Excel (xlsx) можно использовать мастер импорта и экспорта в SSMS, выбрав нужный формат.

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

Ответ: В Excel можно открыть файл через меню "Файл" -> "Открыть" и выбрать нужный формат.

* Данные из скольких таблиц могут храниться в файле csv?

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

* Как экспортировать данные в файл формата JSON?

Ответ:

Для экспорта данных в JSON используется команда SQL, которая может быть адаптирована для генерации JSON-строки.

* Как экспортировать данные в файл формата XML?

Ответ: Выберите меню «Файл» > «Экспорт». В поле «Тип файла» (Windows) диалогового окна Сохранить как выберите «Xml». Введите название и выберите местоположение XML-файла, затем нажмите кнопку «Сохранить». В диалоговом окне «Экспорт XML» выберите необходимую кодировку: UTF-8 или UTF-16. Выполните одно из следующих действий: Включить объявление определения типа документа. Вместе с XML-файлом экспортирует загруженный DTD-файл.

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

**Лабораторная работа №13**

**Импорт данных пользователя в базу данных**

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

Ответ: Для импорта xlsx-файла в SSMS используется мастер импорта и экспорта, который позволяет выбрать источник данных Excel.

* Как выполнить импорт csv-файла в Management Studio?

Ответ: Для импорта csv-файла используется тот же мастер, что и для xlsx, выбирая формат CSV в качестве источника данных.

* Как выполнить импорт файла XML?

Ответ: Для импорта XML-файла можно использовать мастер импорта, который поддерживает формат XML.

* Как выполнить импорт файла JSON?

Ответ: Для импорта JSON в MS SQL Server можно использовать функции OPENJSON или специализированные процедуры для работы с этим форматом.

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

**Лабораторная работа №14**

**Создание слоя доступа к данным БД**

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

Ответ: SqlConnection используется для установления и управления соединением с базой данных SQL Server. Он открывает канал связи между приложением и сервером базы данных.

* Какое назначение у элемента SqlCommand?

Ответ: SqlCommand используется для выполнения SQL-запросов или хранимых процедур в базе данных. Он может быть использован для выборки данных, вставки, обновления или удаления данных.

* Какое назначение у элемента SqlDataReader?

Ответ: SqlDataReader используется для чтения данных из базы данных построчно. Он предоставляет быстрый доступ к результатам запросов, позволяя обработать их в реальном времени.

* 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

* Какие методы класса SqlCommand позволяют выполнить SQL-запрос?

Ответ: ExecuteNonQuery() — выполняет SQL-запрос, не возвращая результатов (например, для операций INSERT, UPDATE, DELETE).

ExecuteReader() — выполняет SQL-запрос и возвращает SqlDataReader для построчного чтения данных.

ExecuteScalar() — выполняет SQL-запрос и возвращает первое значение из первой строки результата.

* **Вывод**
* Научился создавать приложение C# для организации доступа к БД;
* Изучил свойства и методы компонентов SqlConnection, SqlCommand, SqlDataReader, научиться их применять и настраивать.

**Лабораторная работа №15**

**Применение ORM для доступа к данным БД**

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

Ответ: Dapper — это легковесный ORM (Object-Relational Mapper) для .NET, предназначенный для быстрого и простого выполнения запросов к базе данных, маппинга данных на объекты и обратно, с минимальной нагрузкой.

* Какие методы Dapper позволяют извлечь данные из БД?

Ответ: Query<T>() — для выполнения SELECT-запросов и извлечения данных в объекты.

QuerySingle<T>() — для извлечения одного объекта из результата запроса.

QueryFirst<T>() — для извлечения первого объекта из результата запроса.

* Какие методы Dapper позволяют изменить данные в БД?

Ответ: Execute() — для выполнения команд, не возвращающих данных (например, INSERT, UPDATE, DELETE).

ExecuteAsync() — асинхронная версия метода Execute().

* Зачем используется паттерн «репозиторий»?

Ответ: Паттерн "репозиторий" используется для инкапсуляции доступа к данным и для работы с объектами, скрывая детали реализации работы с базой данных. Он позволяет изолировать слой бизнес-логики от конкретной реализации данных.

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

**Лабораторная работа №16**

**Применение ORM EF Core**

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

Ответ: ORM (Object-Relational Mapping) — это технология, позволяющая работать с базой данных через объектно-ориентированное программирование, преобразуя данные из таблиц базы данных в объекты и наоборот.

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

Ответ: EF Core (Entity Framework Core) — это кросс-платформенный ORM для .NET, предназначенный для взаимодействия с базами данных с использованием объектно-ориентированных моделей данных.

* Как получить данные из БД, используя EF Core?

Ответ: Для получения данных из БД используется LINQ-запрос или метод DbSet<T>.ToList() для выборки данных из таблиц, а также FirstOrDefault(), SingleOrDefault() и другие методы.

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

Ответ: Вставка: DbSet<T>.Add() или AddRange()

Обновление: DbSet<T>.Update()

Удаление: DbSet<T>.Remove() или RemoveRange()

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

**Лабораторная работа №17**

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

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

Ответ:

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

Ответ: OrderBy() — сортирует данные по возрастанию.

OrderByDescending() — сортирует данные по убыванию.

ThenBy() — используется для вторичной сортировки (после основной).

ThenByDescending() — вторичная сортировка по убыванию.

* Для чего используется метод Where?

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

* Какие логические операторы могут применяться при составлении условий?

Ответ: Логические операторы: AND, OR, NOT.

* Как проверить, что значение есть в списке?

Ответ: Для проверки наличия значения в списке используется метод Contains().

* Как проверить, что строка начинается с определенного текста?

Ответ: Для проверки используется метод StartsWith().

* Как проверить, что строка содержит определенный текст?

Ответ: Для проверки используется метод Contains().

* Как указать в Select список требуемых данных?

Ответ: В методе Select() указывается выражение, которое выбирает требуемые поля или объекты из коллекции.

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

**Лабораторная работа №18**

**Разработка приложения для редактирования данных**

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

Ответ: Add() используется для добавления одного объекта в коллекцию сущностей, а AddRange() — для добавления нескольких объектов.

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

Ответ: Метод Update() используется для обновления сущности в контексте.

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

Ответ: Remove() используется для удаления одного объекта, а RemoveRange() — для удаления нескольких объектов.

* Как сохранить изменения в БД, используя EF Core?

Ответ: Для сохранения изменений используется метод SaveChanges().

* Как изменить значения полей объекта?

Ответ: Значения полей объекта изменяются непосредственно через свойства объекта.

* Какое значение по умолчанию присваивается идентификатору нового объекта?

Ответ: Обычно, для первичного ключа значение по умолчанию — 0 или null, в зависимости от типа данных.

* Как передать объект с одной формы на другую?

Ответ: Объект можно передать через параметры в методы, через сессии или временные хранилища данных (например, через TempData в ASP.NET).

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

**Лабораторная работа №19**

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

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

Ответ: Используется метод FromSqlRaw() или FromSqlInterpolated() для выполнения необработанных SQL-запросов.

* Как выполнить команду на модификацию данных в EF Core?

Ответ: Для модификации данных можно использовать ExecuteSqlRaw() или ExecuteSqlInterpolated().

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

Ответ: Параметры можно передавать через FromSqlRaw() или ExecuteSqlRaw(), используя синтаксис параметров SQL-запроса.

* Как вызвать табличную функцию в EF Core?

Ответ: Для вызова табличной функции можно использовать метод FromSqlRaw() с соответствующим SQL-запросом.

* Как вызвать хранимую процедуру в EF Core?

Ответ: Хранимую процедуру можно вызвать с помощью метода FromSqlRaw() или ExecuteSqlRaw().

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

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

Ответ: Для работы с выходными параметрами используются методы ExecuteSqlRaw() и ExecuteSqlInterpolated() с указанием параметров в SQL-запросах.

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

**Лабораторная работа №20**

**Реализация разграничения прав доступа пользователей**

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

Ответ: Изменения в настройках подключения могут быть сделаны через строку подключения в конфигурации приложения (например, в файле appsettings.json).

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

Ответ:

Для хранения пользователей и их ролей в базе данных можно использовать несколько подходов. Основной концепцией является создание таблиц для пользователей, ролей и связи между ними.

* Что такое «авторизация»?

Ответ: это процесс проверки и управления правами доступа пользователя к ресурсам или действиям в системе после того, как его личность была подтверждена (например, через процесс аутентификации).

* Что такое «регистрация»?

Ответ: это процесс создания учетной записи пользователя в системе, который обычно включает в себя сбор и сохранение личных данных, таких как имя, электронная почта, логин и пароль.

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

**Лабораторная работа №21**

**Разработка приложения для импорта данных**

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

Ответ: Для импорта данных из CSV-файла можно использовать библиотеку CsvHelper для чтения файла и создания объектов модели, которые затем можно добавить в базу данных с использованием EF Core.

* Как импортировать изображение в БД через приложение?

Ответ: Для импорта изображения в базу данных нужно сначала считать изображение в массив байтов (например, используя File.ReadAllBytes()), а затем сохранить эти данные в столбце типа VARBINARY или BLOB в БД.

* Как выполнить импорт файла JSON в БД через приложение?

Ответ: Для импорта данных из JSON можно использовать библиотеку Newtonsoft.Json для десериализации данных из файла в объекты, а затем вставить эти объекты в базу данных через EF Core.

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

**Лабораторная работа №22**

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

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

Ответ: Для работы с Word в C# через библиотеки Office необходимо подключить пространство имен: using Microsoft.Office.Interop.Word;

* Какие библиотеки позволяют сохранять данные в формате docx?

Ответ: DocumentFormat.OpenXml

Spire.Doc

* Как выполнить экспорт данных в формате txt?

Ответ: using (var writer = new StreamWriter(filePath))

{

foreach (var record in dbContext.MyModels)

{

writer.WriteLine($"{record.Property1}, {record.Property2}");

}

}

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

**Лабораторная работа №23**

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

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

Ответ: using Microsoft.Office.Interop.Word;

* Какие библиотеки позволяют сохранять данные в формате docx?

Ответ: Для работы с форматами .docx используется библиотека DocumentFormat.OpenXml.

* Как выполнить экспорт данных в формате txt?

Ответ: using (var writer = new StreamWriter(filePath))

{

foreach (var record in dbContext.MyModels)

{

writer.WriteLine($"{record.Property1}, {record.Property2}");

}

}

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

**Практическая работа №1**

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

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

Ответ: REST-запрос — это запрос, отправляемый клиентом к серверу, основанный на принципах архитектуры REST. Обычно используется для получения, создания, обновления или удаления данных.

* Что такое RESTful?

Ответ: RESTful — это приложение или API, которое следует принципам архитектуры REST (Representational State Transfer), обеспечивающим взаимодействие через стандартные HTTP-методы (GET, POST, PUT, DELETE).

* Для чего используется метод GET?

Ответ: Метод GET используется для получения данных с сервера.

* Для чего используется метод POST?

Ответ: Метод POST используется для отправки данных на сервер для создания нового ресурса.

* Для чего используется метод PUT?

Ответ: Метод PUT используется для обновления существующего ресурса на сервере.

* Для чего используется метод DELETE?

Ответ: Метод DELETE используется для удаления ресурса на сервере.

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

**Практическая работа №2**

**Вызов REST API**

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

Ответ: HttpClient client = new HttpClient();

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

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

Ответ: 200 OK — запрос выполнен успешно.

201 Created — ресурс успешно создан.

400 Bad Request — ошибка в запросе.

404 Not Found — ресурс не найден.

500 Internal Server Error — ошибка на сервере.

* Какой метод класса HttpClient используется для получения данных?

Ответ: var response = await client.GetAsync("endpoint");

* Какой метод класса HttpClient используется для вставки данных?

Ответ: var response = await client.PostAsync("endpoint", content);

* Какой метод класса HttpClient используется для изменения данных?

Ответ: var response = await client.PutAsync("endpoint", content);

* Какой метод класса HttpClient используется для удаления данных?

Ответ: var response = await client.DeleteAsync("endpoint");

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

**Практическая работа №3**

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

Ответ: В ASP.NET Core для настройки доступа используются атрибуты: [Authorize] — доступ разрешен только авторизованным пользователям. [AllowAnonymous] — доступ разрешен всем, включая неавторизованных пользователей.

* Для чего используется JWT?

Ответ: JWT используется для авторизации и аутентификации пользователей. Он передает информацию о пользователе в зашифрованном виде, и позволяет серверу удостовериться в подлинности пользователя без необходимости хранения сессий.

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

Ответ: Cookie: данные (например, идентификаторы сессий) хранятся на клиенте в браузере. Cookies могут быть уязвимы для атак CSRF (Cross-Site Request Forgery).

JWT: хранится на клиенте (чаще всего в localStorage или sessionStorage), и используется для аутентификации через заголовки запросов. JWT меньше подвержены CSRF, но они могут быть уязвимы к XSS-атакам (Cross-Site Scripting), если не принимаются меры для предотвращения доступа к данным токена.

* Какие настройки можно указать при создании токена?

Ответ: issuer — издатель токена.

audience — целевая аудитория токена.

expiration — срок действия токена.

claims — дополнительные данные (например, роли пользователя).

signing credentials — алгоритм подписи токена.

* Почему для авторизации следует использовать метод POST?

Ответ: Метод POST используется для авторизации, так как при авторизации передается чувствительная информация (например, имя пользователя и пароль), которую не следует передавать через URL (что характерно для GET-запросов). POST-запросы не включают данные в URL, что делает их безопаснее для таких операций.

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

**Практическая работа №4**

**Разработка веб-клиента**

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

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

method: 'GET',

headers: {

'Authorization': `Bearer ${token}`,

'Content-Type': 'application/json'

}

})

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

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

.catch(error => console.error('Error:', error));

* Как обрабатывать ошибки при вызове API в JavaScript?

Ответ: Для обработки ошибок при вызове API можно использовать конструкцию .catch() или проверку статуса ответа:

fetch('https://api.example.com/data')

.then(response => {

if (!response.ok) {

throw new Error('Network response was not ok');

}

return response.json();

})

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

.catch(error => console.error('There has been a problem with your fetch operation:', error));

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

Ответ: Асинхронные функции (использующие ключевое слово async) позволяют выполнять асинхронный код более удобно, ожидая выполнения операций (например, HTTP-запросов) без блокировки основного потока исполнения. Они используют оператор await, который приостанавливает выполнение функции до получения результата асинхронной операции: async function getData() {

const response = await fetch('https://api.example.com/data');

const data = await response.json();

console.log(data);

}

* Что такое Razor Pages?

Ответ: Razor Pages — это способ создания веб-приложений в ASP.NET Core, который представляет собой парадигму, основанную на страницах (Pages). Каждая страница в Razor Pages состоит из Razor-шаблона (HTML с embedded C#) и страницы-класса, который управляет поведением.

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

Ответ: Создайте новую папку в проекте Pages.

Добавьте новый файл .cshtml в эту папку.

Создайте связанный с ним класс в C# (например, Index.cshtml.cs), который будет отвечать за обработку логики для этой страницы.

public class IndexModel : PageModel

{

public void OnGet()

{

// Логика для обработки GET-запроса

}

}

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