**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**Технология разработки и защиты баз данных**

Студент: Мошников А.В.

Группа: ИСПП-21

Преподаватель: Маломан Ю.С.

Архангельск 2024

**Лабораторная работа №1**

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

**1 Цель работы**

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

1.2 Изучить процесс создания диаграммы вариантов использования.

**2 Контрольные вопросы**

2.1 Для чего используется диаграмма вариантов использования?

Диаграмма вариантов использования используется для сбора и визуализации функциональных требований и взаимодействий внутри системы.

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

Актор — это любой объект, субъект или система, взаимодействующая с моделируемой бизнес-системой извне. Обозначается как фигурка человечка.

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

Прецедент — это возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Обозначается в виде эллипса с надписью.

2.4 Что обозначает «отношение ассоциации»?

Взаимодействие актора и прецедента.

2.5 Что обозначает «отношение обобщения»?

Указывает, что один элемент диаграммы является частным случаем другого.

2.6 Что обозначает «отношение включения»?

Показывает,что некоторый вариант использования включает в себя другой вариант использования в качестве составной части.

2.7 Что обозначает «отношение расширения»?

Один вариант использования связан с другим, расширяющим основной вариант дополнительными функциями.

**3 Вывод**

Мы изучили процесс описания требований к системе методом use-case и процесс создания диаграммы вариантов использования.

**Лабораторная работа №2**

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

**1 Цель работы**

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

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

**2 Контрольные вопросы**

2.1 Что такое «сущность»?

Сущность — это объект, который может быть диагностирован и сохранён внутри базы данных.

2.2 Что такое «атрибут»?

Атрибут — это свойство, которое предоставляет более подробную информацию о сущности.

2.3 Что такое «ключевое поле»?

Ключевое поле — это поле, значение которого однозначно определяет каждую запись в таблице.

2.4 Каково назначение первичных и внешних ключей?

Первичный ключ - это ключ, который помогает однозначно идентифицировать кортеж базы данных. Внешний ключ - это ключ, используемый для определения взаимосвязи между таблицами через первичный ключ одной таблицы, действующим как внешний ключ для другой таблицы.

2.5 Что такое «связь»?

Связь — это организованное расположение информации по строкам и столбцам.

2.6 Какие виды связей между сущностями существуют?

Один к одному, одни ко многим и многие ко многим.

2.7 Какие элементы входят в ER-диаграммы?

Сущности, атрибуты и связи.

2.8 Для чего применяются ER-диаграммы?

ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных.

**3 Вывод**

Мы научились применять MySQL Workbench в процессе создания схем моделей БД и представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**Лабораторная работа №3**

**Приведение БД к нормальной форме**

**1 Цель работы**

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

1.2 Изучить процесс декомпозии отношений.

**2 Контрольные вопросы**

2.1 Что называется первичным ключом отношения?

Первичный ключ отношения — это атрибут отношения, однозначно идентифицирующий каждый из его кортежей.

2.2 Что называется внешним ключом отношения?

Внешний ключ отношения — это поле (или набор полей) в одной таблице, которое ссылается на первичный ключ в другой таблице.

2.3 В чем заключается процесс нормализации отношений?

Процесс нормализации заключается в избавлении от избыточности в отношениях и модификации их структуры.

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

Атрибут А функционально зависит от атрибута В, если каждому значению А соответствует в точности одно значение В.

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

Атрибут A транзитивно зависит от атрибута B, если A функционально зависит от C, а C функционально зависит от B.

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

1НФ: атомарность значений.

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

2НФ: должна быть в 1НФ, каждый неключевой атрибут должен находиться в полной функциональной зависимости от ключа.

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

3НФ: должна быть во 2НФ, и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

**3 Вывод**

Мы изучили процесс приведения отношений от ненормализованного вида к четвертой нормальной форме и процесс декомпозии отношений.

**Лабораторная работа №4**

**Установка SQL сервера**

**1 Цель работы**

1.1 Научиться устанавливать разные СУБД, используя docker.

**2 Контрольные вопросы**

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

MSSQL: 1433; MySQL: 3306; Postgres: 5432.

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

MSSQL: /var/opt/mssql; MySQL: /var/lib/mysql; Postgres: /var/lib/postgresql/data

2.3 Для чего используется Docker?

Docker используется для оптимизации управления приложениями и разработки программного обеспечения. Платформа позволяет создавать контейнеры, автоматизировать их запуск и развёртывание, управлять жизненным циклом.

**3 Вывод**

Мы научились устанавливать разные СУБД, используя docker.

**Лабораторная работа №5**

**Создание базы данных в среде разработки.**

**1 Цель работы**

1.1 Изучить способы обеспечения целостности данных в MS SQL Server;

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

**2 Контрольные вопросы**

2.1 Что такое SQL Server Management Studio?

SQL Server Management Studio — это интегрированная среда для управления любой инфраструктурой SQL.

2.2 Какие виды авторизации поддерживаются в MS SQL Server?

Аутентификация Windows и смешанный режим аутентификации.

2.3 Что такое первичный ключ?

Первичный ключ — это атрибут отношения, однозначно идентифицирующий каждый из его кортежей.

2.4 Как указать заполнение столбца автоинкрементными значениями?

Чтобы указать заполнение столбца автоинкрементными значениями в SQL Server, нужно использовать свойство IDENTITY при добавлении столбца.

2.5 Как указать значение по умолчанию?

Чтобы указать значение по умолчанию для столбца в SQL Server, можно использовать SQL Server Management Studio (SSMS):

1. Подключиться к экземпляру SQL Server в SSMS.

2. В обозревателе объектов щёлкнуть правой кнопкой мыши таблицу со столбцами, масштаб которых необходимо изменить, и выбрать «Конструктор».

3. Выбрать столбец, для которого нужно задать значение по умолчанию.

4. На вкладке «Свойства столбца» ввести новое значение по умолчанию в свойстве «Значение по умолчанию или привязка».

5. В меню «Файл» выбрать пункт «Сохранить» (имя таблицы).

2.6 Как задать проверочное ограничение?

Чтобы задать проверочное ограничение в SQL Server, можно использовать SQL Server Management Studio:

1. В обозревателе объектов развернуть таблицу, в которую необходимо добавить проверочное ограничение.

2. Щёлкнуть правой кнопкой мыши на пункте «Ограничения» и выбрать команду «Создать ограничение».

3. В диалоговом окне «Проверочные ограничения» установить курсор в поле «Выражение» и затем нажать кнопку с многоточием (…).

4. В диалоговом окне «Выражение проверочного ограничения» ввести выражения SQL, соответствующие проверочному ограничению.

5. Нажать «ОК».

6. В категории «Идентификация» можно изменить имя проверочного ограничения и добавить описание (расширенное свойство) ограничения.

7. В категории «Конструктор таблиц» можно задать время принудительного выполнения проверочного ограничения.

2.7 Как обеспечить уникальность значений в столбце или наборе столбцов?

Чтобы обеспечить уникальность значений в столбце или наборе столбцов в Microsoft SQL Server, можно использовать ограничение UNIQUE. Оно позволяет исключить повторяющиеся значения в столбце.

2.8 Что такое внешний ключ?

Внешний ключ — это поле (или набор полей) в одной таблице, которое ссылается на первичный ключ в другой таблице.

2.9 Какие значения может принимать внешний ключ?

Внешний ключ может принимать следующие значения: NULL: при удалении связанной строки из главной таблицы для столбца внешнего ключа устанавливается значение NULL; Значение по умолчанию: при удалении связанной строки из главной таблицы для столбца внешнего ключа устанавливается значение по умолчанию, которое задаётся с помощью атрибута DEFAULT. Если для столбца не задано значение по умолчанию, то в качестве него применяется значение NULL.

**3 Вывод**

Мы изучили способы обеспечения целостности данных в MS SQL Server и научились работать в среде SQL Server Management Studio (SSMS).

**Лабораторная работа №6**

**Создание представлений в СУБД**

**1 Цель работы**

1.1 Научиться создавать и использовать представления в MS SQL Server;

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

**2 Контрольные вопросы**

2.1 Что такое представления и чем они отличаются от таблиц?

Представление — это виртуальная таблица, определение которой хранится в базе данных. В отличие от таблиц представления не содержат никаких данных. Вместо этого они обеспечивают способ хранения в базе данных часто используемых сложных запросов.

2.2 Для чего применяются представления?

Представления применяются для упрощения доступа к базе данных, ограничения доступа к данным и улучшения производительности.

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

Создание представления, включающего информацию из нескольких таблиц одновременно возможно,если в запросе написать JOIN.

2.4 Какие требование предъявляются к обновляемым представлениям?

Любые изменения должны ссылаться на столбцы только одной базовой таблицы и таблицы должны быть объединены только с помощью INNER JOIN.

2.5 Как создать обновляемое представление?

CREATE VIEW ИмяПредставления

**3 Вывод**

Мы научились создавать и использовать представления в MS SQL Server и закрепили навык создания запросов на выборку в MS SQL Server.

**Лабораторная работа №7**

**Создание функций пользователя в СУБД**

**1 Цель работы**

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

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

**2 Контрольные вопросы**

2.1 Что такое функции пользователя и для чего они применяются?

Функции пользователя — это самостоятельные объекты базы данных, такие как хранимые процедуры или триггеры. Они применяются для сохранения разработанного кода для дальнейшего применения и улучшения организации кода и возможности повторного использования

2.2 Чем отличается скалярная функция от табличной?

Скалярные функции — это функции, которые возвращают одно значение заданного типа данных. Табличные функции — это функции, которые возвращают табличные данные.

2.3 Как объявить переменную и присвоить ей значение?

DECLARE @имяПеременной типДанных

2.4 Как указать в функции параметр по умолчанию?

DECLARE @имяПеременной типДанных = значение

2.5 Как вызвать скалярную функцию?

SELECT имяСкалярнойФункции FROM имяТаблицы

2.6 Как вызвать табличную функцию?

SELECT \* FROM имяТабличнойФункции

**3 Вывод**

Мы научились создавать и использовать скалярные и табличные функции пользователя и закрепили навык создания запросов на выборку в MS SQL Server.

**Лабораторная работа №8**

**Создание хранимых процедур в СУБД**

**1 Цель работы**

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

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

**2 Контрольные вопросы**

2.1 Что такое хранимые процедуры и для чего они применяются?

Хранимые процедуры — это предварительно скомпилированный набор операторов SQL, который хранится в базе данных и может быть выполнен одной командой. Они применяются для улучшения производительности, упрощения сложных операций и обеспечения безопасности данных.

2.2 Чем отличается функция пользователя от хранимой процедуры?

Функции пользователя вызываются посредством оператора SELECT и возвращают значение.

2.3 Как задать входные параметры хранимой процедуры?

CREATE PROCEDURE имяПроцедуры @имяПеременной1 тип

2.4 Как задать выходные параметры хранимой процедуры?

После типа переменной процедуры указать параметр OUTPUT

CREATE PROCEDURE имяПроцедуры @имяПеременной1 тип OUTPUT

2.5 Как вызвать выполнение хранимой процедуры?

EXECUTE имяПроцедуры

**3 Вывод**

Мы научились создавать и использовать хранимые процедуры в MS SQL Server и закрепили навык создания запросов на выборку и модификацию данных в MS SQL Server.

**Лабораторная работа №9**

**Создание триггеров в СУБД**

**1 Цель работы**

1.1 Научиться создавать и использовать триггеры в MS SQL Server;

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

**2 Контрольные вопросы**

2.1 Что такое триггер?

Триггер — это хранимая процедура особого типа, которую пользователь не вызывает непосредственно. Её исполнение обусловлено действием по модификации данных.

2.2 Чем триггер отличается от хранимой процедуры?

Триггер — это процедура,выполняющаяся автоматически как реакция на событие.

2.3 Как запустить триггер на выполнение?

ALTER TRIGGER имяТриггера ENABLE

2.4 Каково назначение триггеров?

Назначение триггеров в БД заключается в обеспечении целостности данных и реализации сложной бизнес-логики.

2.5 Чем отличаются триггеры INSTEAD OF и AFTER?

INSTEAD OF: выполнить что-то вместо DML-команды.

AFTER: выполнить что-то после DML-команды.

**3 Вывод**

Мы научились создавать и использовать триггеры в MS SQL Server и закрепили навык создания запросов на выборку и модификацию данных в MS SQL Server.

**Лабораторная работа №10**

**Реализация доступа пользователей к БД**

**1 Цель работы**

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

1.2 Научиться назначать привилегии пользователю БД;

1.3 Закрепить навык создания объектов БД.

**2 Контрольные вопросы**

2.1 В чем отличие между именами входа и пользователями БД?

Имя входа — это глобальный логин для всего экземпляра SQL Server. С помощью него вы проходите процесс аутентификации. Пользователь – это участник базы данных, привязанный к определенному Имени Входа.

2.2 Как идентифицируются пользователи в MS SQL Server?

Идентификация пользователей в MS SQL Server происходит через аутентификацию.

2.3 На какие уровни разделяется система безопасности MS SQL Server?

Уровень сервера, уровень БД и уровень схемы.

2.4 Каково назначение ролей сервера?

Роли сервера в БД определяют уровень доступа к объектам сервера. Они позволяют упростить управление привилегиями пользователей и структурировать этот процесс.

2.5 Каково назначение ролей БД?

Роли базы данных — это специальные объекты базы данных, которые используются для упрощения предоставления разрешений в базах данных.

**3 Вывод**

Мы научились использовать системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД, назначать привилегии пользователю БД и закрепили навык создания объектов БД.

**Лабораторная работа №11**

**Выполнение резервного копирования и восстановления БД**

**1 Цель работы**

1.1 Научиться выполнять резервное копирование БД;

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

1.3 Научиться выполнять восстановление БД.

**2 Контрольные вопросы**

2.1 Для чего создаются резервные копии БД?

Для сохранения вносимых изменений на регулярной основе.

2.2 В чем отличие между полным и разностным резервным копированием?

Полное резервное копирование затрагивает всю систему и все файлы. В нём содержатся все данные, необходимые для полного восстановления данных. Разностное резервное копирование сохраняет только те изменения данных, которые произошли после создания полной резервной копии.

2.3 Как часто должен выполняться каждый из видов резервного копирования БД?

Частота резервного копирования базы данных (БД) зависит от важности данных, бизнес-требований и уровня допустимой потери данных.

Для критически важных баз данных резервное копирование может выполняться несколько раз в день. Для менее критичных баз данных может быть достаточно ежедневного или даже еженедельного резервного копирования.

2.4 Какие скрипты можно сформировать для объектов БД?

Скрипты создания и обновления.

2.5 Как выполнить восстановление БД из резервной копии?

БД → Восстановить БД → Выбрать БД и выбрать из списка РК нужную для восстановления → Выбрать опцию "Закрыть существующие подключения к целевой БД" → Нажать "ОК".

2.6 Какая команда выполняет восстановление БД из резервной копии?

RESTORE DATABASE

2.7 В каком порядке надо восстанавливать резервные копии?

При восстановлении базы данных вручную  информацию сначала восстанавливают с полной копии, а затем последовательно со всех частичных копий.

При разностном методе восстановление информации выполняется быстрее, поскольку используются только две копии: полная и последняя разностная.

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

RECOVERY/NORECOVERY: запускает/отключает восстановление

2.9 Для чего используется мастер импорта и экспорта?

Мастер импорта и экспорта используется для простого копирования данных из источника в место назначения.

**3 Вывод**

Мы научились использовать системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД, назначать привилегии пользователю БД и закрепили навык создания объектов БД.

**Лабораторная работа №12**

**Экспорт базы данных в документы пользователя**

**1 Цель работы**

1.1 Научиться выполнять экспорт данных из БД.

**2 Контрольные вопросы**

2.1 Как выполнить экспорт csv-файла в SSMS?

БД → Задачи → Экспорт данных → В источнике данных выбрать БД → В выборе назначения указать "Flat File Source" → Указать расположение файла → Следовать инструкциям экспорта данных.

2.2 Как выполнить экспорт xlsx-файла в SSMS?

БД → Задачи → Экспорт данных → В источнике данных выбрать БД → В выборе назначения указать "Microsoft Excel" → Указать расположение файла → Следовать инструкциям экспорта данных.

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

Для xml: Excel → Открыть → Выбрать файл формата xml.

Для txt/csv: Excel → Данные → Выбрать "Из текста/CSV" → Выбрать файл → Выбрать разделитель, указанный в файле → Следовать инструкциям Excel.

2.4 Данные из скольких таблиц могут храниться в файле csv?

В csv-файле могут храниться данные одной или нескольких таблиц.

2.5 Как экспортировать данные в файл формата JSON?

БД → Задачи → Экспорт данных → В выборе назначения указать "Flat File Source" → Указать расположение файла и формат .json → Следовать инструкциям экспорта данных.

2.6 Как экспортировать данные в файл формата XML?

БД → Задачи → Экспорт данных → В выборе назначения указать "Flat File Source" → Указать расположение файла и формат .xml → Следовать инструкциям экспорта данных.

**3 Вывод**

Мы научились выполнять экспорт данных из БД.

**Лабораторная работа №13**

**Импорт данных пользователя в базу данных**

**1 Цель работы**

1.1 Научиться выполнять импорт данных в БД.

**2 Контрольные вопросы**

2.1 Как выполнить импорт xlsx-файла в Management Studio?

БД → Задачи → Импорт данных → В источнике данных выбрать "Microsoft Excel" → В выборе назначения указать сервер → Следовать инструкциям импорта данных.

2.2 Как выполнить импорт csv-файла в Management Studio?

БД → Задачи → Импорт данных → В источнике данных выбрать "Flat File Source" → В выборе назначения указать сервер → Следовать инструкциям импорта данных.

2.3 Как выполнить импорт файла XML?

БД → Задачи → Импорт данных → В источнике данных выбрать "Flat File Source" и поменять расширение на xml → В выборе назначения указать сервер → Следовать инструкциям импорта данных.

2.4 Как выполнить импорт файла JSON?

БД → Задачи → Импорт данных → В источнике данных выбрать "Flat File Source" и поменять расширение на json → В выборе назначения указать сервер → Следовать инструкциям импорта данных.

**3 Вывод**

Мы научились выполнять импорт данных в БД.

**Лабораторная работа №14**

**Создание слоя доступа к данным БД**

**1 Цель работы**

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

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

**2 Контрольные вопросы**

2.1 Какое назначение у элемента SqlConnection?

Создание строки подключения к БД.

2.2 Какое назначение у элемента SqlCommand?

Создание запросов к БД посредством DML-команд.

2.3 Какое назначение у элемента SqlDataReader?

SqlDataReader предназначен для получения данных из базы.

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

SSMS: System.Data.SqlClient; MySQL: System.Data.MySqlClient.

2.5 Какие методы класса SqlCommand позволяют выполнить SQL-запрос?

ExecuteNonQuery, ExtcuteScalar и ExecuteReader.

**3 Вывод**

Мы научились создавать приложение C# для организации доступа к БД и изучили свойства и методы компонентов SqlConnection, SqlCommand, SqlDataReader, научиться их применять и настраивать.

**Лабораторная работа №15**

**Применение ORM для доступа к данным БД**

**1 Цель работы**

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

1.2 Научиться проектировать приложение, использующее паттерн репозиторий и Dapper.

**2 Контрольные вопросы**

2.1 Что такое Dapper и для чего используется?

Dapper — это библиотека объектно-реляционного сопоставления (ORM) для платформы Microsoft .NET. Используется для упрощения работы с данными в приложении.

2.2 Какие методы Dapper позволяют извлечь данные из БД?

Query и ExecuteScalar.

2.3 Какие методы Dapper позволяют изменить данные в БД?

Execute позволяет выполнять команды, которые не предназначены для возврата наборов результатов, например INSERT, UPDATE и DELETE.

2.4 Зачем используетсяя паттерн «репозиторий»?

Паттерн «репозиторий» используется для разделения бизнес-логики от деталей реализации слоя доступа к данным.

**3 Вывод**

Мы научились создавать приложение C# для организации доступа к БД и проектировать приложение, использующее паттерн репозиторий и Dapper.

**Лабораторная работа №16**

**Применение ORM EF Core**

**1 Цель работы**

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

1.2 Научиться разрабатывать приложение, использующее EF Core.

**2 Контрольные вопросы**

2.1 Что такое «ORM»?

ORM — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных».

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

Entity Framework Core — это простая, кроссплатформенная и расширяемая версия популярной технологии доступа к данным Entity Framework.

2.3 Как получить данные из БД, используя EF Core?

Чтобы получить данные из базы данных с помощью EF Core, можно использовать метод FromSqlRaw(). Он принимает в качестве параметра SQL-выражение и набор параметров и возвращает набор полученных из БД объектов.

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

Чтобы выполнить редактирование с помощью EF Core, нужно использовать метод SaveChanges().

**3 Вывод**

Мы научились создавать приложение C# для организации доступа к БД и разрабатывать приложение, использующее EF Core.

**Лабораторная работа №17**

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

**1 Цель работы**

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

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

**2 Контрольные вопросы**

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

OrderBy – сортировка по возрастанию; OrderByDescending – сортировка по убыванию; ThenBy – используется после OrderBy / OrderByDescending, сортирует по второму критерию по возрастанию; ThenByDescending – используется после OrderBy / OrderByDescending, сортирует по второму критерию по убыванию.

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

Take(n) – выбирает первые n данных из списка; Skip(n) – позволяет пропустить n данных для реализации постраничного вывода (то есть берёт, начиная с n + 1).

2.3 Для чего используется метод Where?

Для выборки данных по определённому условию.

2.4 Какие логические операторы могут применяться при составлении условий?

&& – и, || – или и ! – не.

2.5 Как проверить, что значение есть в списке?

Используя метод Exists.

2.6 Как проверить, что строка начинается с определенного текста?

Используя метод StartsWith.

2.7 Как проверить, что строка содержит определенный текст?

Используя метод Contains.

2.8 Как указать в Select список требуемых данных?

var selectedFields = dbContext.ИмяТаблицы.Select(item => new { item.Field1, item.Field2 }).ToList()

**3 Вывод**

Мы научились создавать приложение для доступа к БД средствами EF Core и выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.

**Лабораторная работа №18**

**Разработка приложения для редактирования данных**

**1 Цель работы**

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

1.2 Научиться обеспечивать обратную связь при редактировании данных.

**2 Контрольные вопросы**

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

Для добавления объекта или массива объектов в БД.

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

Для обновления данных объекта в БД.

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

Для удаления данных объекта из БД.

2.4 Как сохранить изменения в БД, используя EF Core?

Используя метод SaveChanges.

2.5 Как изменить значения полей объекта?

Выбрать объект из БД, например по идентификатору и занести в объект класса. Затем изменить у объекта значение поля или нескольких полей и сохранить изменения.

2.6 Какое значение по умолчанию присваивается идентификатору нового объекта?

Автоинкрементное значение от предыдущего.

2.7 Как передать объект с одной формы на другую?

Используя статический глобальный массив или передавать ссылку на объект.

**3 Вывод**

Мы научились создавать приложение для доступа к БД средствами EF Core и выполнять сортировку, фильтрацию и постраничный вывод данных, используя LINQ-запросы.

**Лабораторная работа №19**

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

**1 Цель работы**

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

**2 Контрольные вопросы**

2.1 Как выполнить команду на выборку данных в EF Core?

Используя Where для поиска одного значения.

2.2 Как выполнить команду на модификацию данных в EF Core?

Создать новый объект класса и, выбрав из БД объект, например, по идентификатору, вызвать метод Update с параметром объекта.

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

var sql = @"Update [User] SET FirstName = @FirstName WHERE Id = @Id";

dbContext.Database.ExecuteSqlCommand(

sql,

new SqlParameter("@FirstName", firstname),

   new SqlParameter("@Id", id)

);

2.4 Как вызвать табличную функцию в EF Core?

В OnModelCreating определить функцию с помощью HasDbFunction(() => ИмяФункции(default)) и вызывать по имени функции.

2.5 Как вызвать хранимую процедуру в EF Core?

Через DbSet.FromSql или через dbContext.Database.ExecuteSqlCommand.

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

var sql = "exec ИмяПроцедуры @Param1, @Param2, @Param3";

**3 Вывод**

Мы научились выполнять SQL-команды и вызывать хранимые процедуры и функции пользователя средствами EF Core.

**Лабораторная работа №20**

**Реализация разграничения прав доступа пользователей**

**1 Цель работы**

1.1 Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;

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

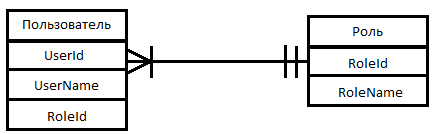
**2 Контрольные вопросы**

2.1 Как изменить настройки подключения к БД в клиентском приложении?

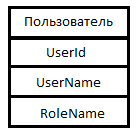
В appsettings.json переопределить строку подключения.

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

1 способ – хранение пользователей и ролей в разных таблицах



2 способ – хранение пользователей и их ролей



2.3 Что такое «авторизация»?

Авторизация — предоставление определённому лицу или группе лиц прав на выполнение определённых действий, а также процесс проверки этих прав при попытке выполнения этих действий.

2.4 Что такое «регистрация»?

Регистрация — это процесс идентификации пользователя с использованием пользовательских данных.

**3 Вывод**

Мы научились разграничивать права доступа пользователей на уровне интерфейса приложения и изменять настройки подключения к БД средствами EF Core.

**Лабораторная работа №21**

**Разработка приложения для импорта данных**

**1 Цель работы**

1.1 Импорт данных пользователя в базу данных средствами EF Core.

**2 Контрольные вопросы**

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

2.2 Как импортировать изображение в БД через приложение?

2.3 Как выполнить импорт файла JSON в БД через приложение?

**3 Вывод**

Мы научились импортировать данные пользователя в БД средствами EF Core.

**Лабораторная работа №22**

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

**1 Цель работы**

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

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

**2 Контрольные вопросы**

2.1 Какое пространство имен требуется подключить для работы с Word?

Microsoft.Office.Interop.Word.

2.2 Какие библиотеки позволяют сохранять данные в формате docx?

Данные в формате docx позволяет сохранять библиотека DocX.

2.3 Как выполнить экспорт данных в формате txt?

С помощью SQLCMD или через PowerShell.

**3 Вывод**

Мы научились выполнять экспорт данных из БД в формате текстового файла и выполнять экспорт данных из БД с применением Microsoft Office Word.

**Лабораторная работа №23**

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

**1 Цель работы**

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

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

**2 Контрольные вопросы**

2.1 Какое пространство имен требуется подключить для работы с Excel?

Microsoft.Office.Interop.Excel.

2.2 Какие библиотеки позволяют сохранять данные в формате xlsx?

Данные в формате xlsx позволяет сохранять библиотеки EPPlus, NPOI, ClosedXML и др.

**3 Вывод**

Мы научились выполнять экспорт данных из БД в формате табличного файла и выполнять экспорт данных из БД с применением Microsoft Office Excel.

**Практическая работа №1**

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

**1 Цель работы**

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

**2 Контрольные вопросы**

2.1 Что такое REST-запрос?

REST-запрос — это HTTP-запрос, используемый в архитектурном стиле REST для получения или изменения данных на сервере.

2.2 Что такое RESTful?

RESTful — это архитектурный стиль взаимодействия компонентов распределённого приложения в сети, который использует протокол HTTP для передачи данных между сервером и клиентом.

2.3 Для чего используется метод GET?

Метод GET используется для получения данных из БД.

2.4 Для чего используется метод POST?

Метод POST используется для добавления данных в БД.

2.5 Для чего используется метод PUT?

Метод PUT используется для изменения данных в БД.

2.6 Для чего используется метод DELETE?

Метод DELETE используется для удаления данных из БД.

**3 Вывод**

Мы научились выполнять разработку web-API для доступа к БД.

**Практическая работа №2**

**Вызов методов REST API**

**1 Цель работы**

1.1 Научиться проверять работоспособность RESTful API в клиентском приложении;

1.2 Научиться выполнять тестирование RESTful API методом черного ящика.

**2 Контрольные вопросы**

2.1 Как указать у объекта HttpClient базовый адрес?

var client = new HttpClient();

client.BaseAddress = new Uri("адрес");

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

200 – OK, 201 – Created, 204 – No Content, 400 – Bad Request, 404 – Not Found, 500 – Internal Server Error.

2.3 Какой метод класса HttpClient используется для получения данных?

Метод GET используется для получения данных из БД.

2.4 Какой метод класса HttpClient используется для вставки данных?

Метод POST используется для добавления данных в БД.

2.5 Какой метод класса HttpClient используется для изменения данных?

Метод PUT используется для изменения данных в БД.

2.6 Какой метод класса HttpClient используется для удаления данных?

Метод DELETE используется для удаления данных из БД.

**3 Вывод**

Мы научились проверять работоспособность RESTful API в клиентском приложении и выполнять тестирование RESTful API методом черного ящика.

**Практическая работа №3**

**Разграничение прав доступа на уровне REST API**

**1 Цель работы**

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

1.2 Научиться работать с JWT.

**2 Контрольные вопросы**

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

Authorize, AllowAnonymous и Roles.

2.2 Для чего используется JWT?

JSON Web Token — это открытый стандарт для создания токенов доступа, основанный на формате JSON.

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

Авторизация с использованием cookie менее безопасна, так как файлы cookie уязвимы для атак с использованием межсайтового скриптинга (XSS) и подделки межсайтовых запросов (CSRF). Авторизация с использованием JWT более безопасна, так как токены генерируются и подписываются только со стороны веб-приложения.

2.4 Какие настройки можно указать при создании токена?

ValidateIssuer, ValidateAudience, ValidateLifetime, ValidateIssuerSigningKey и др.

2.5 Почему для авторизации следует использовать метод POST?

Параметры запроса с методом POST не видны пользователю без использования специального ПО, что повышает конфиденциальность и снижает вероятность утечки информации.

**3 Вывод**

Мы научились выполнять разработку web-API для доступа к БД и работать с JWT.

**Практическая работа №4**

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

**1 Цель работы**

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

1.2 Научиться проверять работоспособность RESTful API в клиентском приложении.

**2 Контрольные вопросы**

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

Чтобы выполнить HTTP-запрос к API с помощью Fetch API, нужно использовать глобальный метод fetch().

2.2 Как обрабатывать ошибки при вызове API в JavaScript?

Для обработки ошибок при вызове API в JavaScript можно использовать конструкцию try/catch.

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

Асинхронные функции нужны для выполнения асинхронных операций, таких как работа с API, базами данных, чтение файлов и т. д.. Асинхронные операции выполняются не сразу: код отправил запрос к API и ждёт, пока сервер пришлёт ответ. Связь асинхронных функций с вызовом API заключается в том, что асинхронные API позволяют отправлять множество запросов одновременно, не дожидаясь ответа. Это позволяет различным приложениям взаимодействовать более эффективно, что приводит к повышению производительности и времени ответа.

2.4 Что такое Razor Pages?

Razor Pages — это новая улучшенная версия MVC. Она позволяет создавать файлы с расширением .cshtml, которые являются смесью C# и HTML.

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

Создать папку Pages → Добавить → Новый объект → Выбрать шаблон Razor Page и создать файл.

**3 Вывод**

Мы научились разрабатывать клиентское веб-приложение для доступа к БД и проверять работоспособность RESTful API в клиентском приложении.