**АКТ(ф) СПбГУТ**

ОТЧЁТЫ ПО

ЛАБОРАТОРНЫМ И ПРАКТИЧЕСКИМ

РАБОТАМ

ТРиЗД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИСПП-21 |  | 20.12.2024 | Баранов Е.А. |
|  | (группа) | (подпись) | (дата) | (Фамилия И.О.) |
| Студент | ИСПП-21 |  | 20.12.2024 | Хомутов М.A. |
|  | (группа) | (подпись) | (дата) | (Фамилия И.О.) |
| Преподаватель | |  | 20.12.2024 | Маломан Ю.С. |
|  |  | (подпись) | (дата) | (Фамилия И.О.) |

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. Контрольные вопросы
   1. Для чего используется диаграмма вариантов использования?

Ответ: Диаграмма вариантов использования используется для определения общих границ функциональности проектируемой системы в контексте моделируемой предметной области.

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

Ответ: Актор на диаграмме вариантов использования (прецедентов) — это любая внешняя по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач. Это может быть человек, техническое устройство, программа или любая другая система.

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

Ответ: Прецедент — это возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой.

* 1. Что обозначает «отношение ассоциации»?

Ответ: Ассоциация - это отношение, при котором объекты одного типа неким образом связаны с объектами другого типа. Например, объект одного типа содержит или использует объект другого типа.

* 1. Что обозначает «отношение обобщения»?

Ответ: Отношение обобщения обозначает, что один элемент модели (дочерний) основан на другом элементе модели (родительском).

* 1. Что обозначает «отношение включения»?

Ответ: «Отношение включения» в контексте диаграмм вариантов использования указывает на то, что заданное поведение одного варианта использования обязательно включается в качестве составного компонента в последовательность поведения другого варианта использования.

* 1. Что обозначает «отношение расширения»?

Ответ: «Отношение расширения» в моделировании UML обозначает взаимосвязь базового варианта использования с другим вариантом использования, функциональное поведение которого задействуется базовым не всегда, а только при выполнении дополнительных условий.

1. Вывод:
   1. В ходе лабораторной работы изучен процесс описания требований к системе методом use-case.
   2. В ходе лабораторной работы изучен 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. Контрольные вопросы
   1. Что такое «сущность»?

Ответ: Сущность — это класс однотипных объектов, информация о которых должна быть учтена в модели.

* 1. Что такое «атрибут»?

Ответ: Атрибут — это наименьшая единица структуры данных, информационное отображение свойств объекта.

* 1. Что такое «ключевое поле»?

Ответ: Ключевое поле — это поле, значения которого однозначно определяют каждую запись в таблице.

* 1. Каково назначение первичных и внешних ключей?

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

* 1. Что такое «связь»?

Ответ: Связь в базе данных (БД) — это ассоциация между таблицами, которая позволяет сопоставлять данные в ключевых столбцах. В большинстве случаев связь соединяет основной ключ или уникальный столбец идентификатора для каждой строки от одной таблицы к полю в другой таблице.

* 1. Какие виды связей между сущностями существуют?

Ответ:

Существуют три вида связей между сущностями:

1. **«Один-к-одному»**. Один экземпляр сущности связан только с одним экземпляром другой сущности.
2. **«Один-ко-многим»**. Один экземпляр сущности связан со множеством экземпляров другой сущности.
3. **«Многие-ко-многим»**. Множество экземпляров одной сущности связаны со множеством экземпляров другой сущности.
   1. Какие элементы входят в ER-диаграммы?

Ответ:

В ER-диаграмме входят **три основных элемента**: [4](https://creately.com/blog/ru/uncategorized-ru/%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%BE%D0%BC-%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%B8%D0%B8-%D0%BF%D0%BE-er-%D0%B4%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D0%BC/)

1. **Сущность** - это класс однотипных объектов, информация о которых должна быть учтена в модели.
2. **Атрибут** - это именованная характеристика, являющаяся некоторым свойством сущности.
3. **Связь** - описывает, как взаимодействуют сущности.
   1. Для чего применяются ER-диаграммы?

Ответ:

ER-диаграммы применяются для визуализации и упрощения сложных структур данных. Они помогают:

* **Чётко определить элементы, из которых состоит система.** Это позволяет визуализировать сложные связи и взаимозависимости между этими элементами.
* **Упростить анализ и улучшение моделей баз данных.** ER-диаграммы способствуют более эффективной коммуникации между проектировщиками и пользователями, уменьшению ошибок и улучшению структуры данных.
* **Отобразить, как объекты связаны друг с другом.** Это помогает понять, как работают эти связи и какие аспекты могут быть улучшены.

1. Вывод:
   1. В ходе лабораторной работы научились применять MySQL Workbench в процессе создания схем моделей БД;
   2. В ходе лабораторной работы научились 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 декомпозии отношений.
2. Контрольные вопросы
   1. Что называется первичным ключом отношения?

Ответ: Первичный ключ отношения — в реляционной модели данных один из потенциальных ключей отношения, выбранный в качестве основного ключа

* 1. Что называется внешним ключом отношения?

Ответ: Внешний ключ отношения — это один или несколько столбцов из одной таблицы, который одновременно является потенциальным ключом из другой таблицы.

* 1. В чем заключается процесс нормализации отношений?

Ответ: Процесс нормализации отношений заключается в преобразовании отношений реляционной базы данных к виду, отвечающему одной из нормальных форм.

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

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

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

Ответ: Атрибут С транзитивно зависит от атрибута А, если для атрибутов А, В и С выполняются условия А → В и В → С, но отсутствует обратная зависимость.

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

Ответ: Отношение находится в первой нормальной форме (1НФ), если все атрибуты отношения являются простыми (требование атомарности атрибутов в реляционной модели), т.е. не имеют компонентов.

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

Ответ:

Отношение, находящееся во второй нормальной форме (2НФ), должно отвечать следующим требованиям:

1. Быть в первой нормальной форме (1НФ).
2. Все неключевые атрибуты отношения функционально полно зависят от составного ключа отношения. Иными словами, отношение не должно содержать частичных функциональных зависимостей.
   1. Каким требованиям должно отвечать отношение, находящееся в 3НФ?

Ответ:

Отношение, находящееся в 3НФ, должно отвечать следующим требованиям:

1. Быть во второй нормальной форме (2НФ). [1](https://intuit.ru/studies/professional_skill_improvements/17411/courses/191/lecture/4977?page=4)[3](https://appmaster.io/ru/glossary/tret-ia-normal-naia-forma-3nf)
2. Все неключевые атрибуты отношения зависят только от первичного ключа. Иными словами, 3НФ требует, чтобы отношение не содержало транзитивных зависимостей неключевых атрибутов от ключа.
3. Вывод:
   1. В ходе лабораторной работы изучен процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;
   2. В ходе лабораторной работы изучен процесс декомпозии отношений.

**Лабораторная работа №4**

**«Установка SQL сервера»**

1. Цель работы:
   1. Научиться устанавливать разные СУБД, используя docker.
2. Контрольные работы
   1. Какие стандартные порты для подключения к СУБД MSSQL, MySQL, Postgres?

Ответ:

Стандартные порты для подключения к некоторым СУБД:

* MySQL — 3306.
* PostgreSQL — 5432.
* SQL Server — 1433.
  1. Какие стандартные папки для хранения данных СУБД MSSQL, MySQL, Postgres?

Ответ:

Стандартные папки для хранения данных СУБД MSSQL, MySQL и PostgreSQL:

1. **Для MSSQL**: если не указано иное, то файлы данных и журналов создаются в том же каталоге, что и системные базы данных SQL Server, то есть **<диск>:\Program Files\Microsoft SQL Server\MSSQL.MSSQLSERVER\MSSQL\DATA**. Например, для экземпляра SQL Server 2014, установленного на диске C, файлы данных и журналов по умолчанию будут находиться в каталоге **C:\Program Files\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQL\DATA**.
2. **Для MySQL**: в Linux по умолчанию файлы с базами данных хранятся в директории **/var/lib/mysql**. В Windows — **C:\ProgramData\MySQL\MySQL Server{version}\Data**. В macOS — **/usr/local/mysql/data/**.
3. **Для PostgreSQL**: обычно PGDATA находится в директории **/var/lib/pgsql/data** на Linux и **\Program Files\PostgreSQL[version]\data** на Windows.
   1. Для чего используется Docker?

Ответ: Docker используется для упрощения, автоматизации и ускорения разработки и развёртывания приложений.

1. Вывод:
   1. В ходе лабораторной рабоы научились 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. Контрольные вопросы
   1. Что такое SQL Server Management Studio?

Ответ: SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL.

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

Ответ:

В Microsoft SQL Server поддерживаются два вида авторизации:

1. **Аутентификация Windows (Windows Authentication)**. Осуществляется с помощью системы безопасности Windows. Пользователям, которые уже аутентифицированы в Windows и имеют права на SQL Server, не нужно предоставлять дополнительные учётные данные.
2. **Смешанный режим аутентификации (Mixed Mode Authentication)**. В этом режиме помимо аутентификации Windows поддерживается аутентификация самого SQL Server через логин и пароль.
   1. Что такое первичный ключ?

Ответ: Первичный ключ (primary key) — поле в таблице, которое однозначно идентифицирует каждую запись в ней.

* 1. Как указать заполнение столбца автоинкрементными значениями?

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

* 1. Как указать значение по умолчанию?

Ответ:

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).
   1. Как задать проверочное ограничение?

Ответ:

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. Например, чтобы ограничить записи в столбце SellEndDate таблицы Product значением, превышающим или равным дате в столбце SellStartDate, или значением NULL, нужно ввести:

SellEndDate >= SellStartDate

1. Нажать «ОК».
2. В категории «Идентификация» можно изменить имя проверочного ограничения и добавить описание (расширенное свойство) ограничения.
3. В категории «Конструктор таблиц» можно задать время принудительного выполнения проверочного ограничения.
   1. Как обеспечить уникальность значений в столбце или наборе столбцов?

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

* 1. Что такое внешний ключ?

Ответ: Внешний ключ — это ограничение целостности, обеспечивающее соответствие между полями разных таблиц.

* 1. Какие значения может принимать внешний ключ?

Ответ:

**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.

1. Вывод
   1. В ходе лабораторной работы изучен способы обеспечения целостности данных в MS SQL Server;
   2. В ходе лабораторной работы научились 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. Контрольные работы
   1. Что такое представления и чем они отличаются от таблиц?

Ответ: Представление — это виртуальная таблица, определение которой хранится в базе данных. Но в отличие от таблиц представления не содержат никаких данных.

* 1. Для чего применяются представления?

Ответ:

Представления применяются для следующих целей:

* **Упрощение сложных запросов**. Представления скрывают сложность структуры данных и предоставляют упрощённый интерфейс для доступа к ним.
* **Улучшение производительности**. Создание представлений, которые инкапсулируют сложные запросы, может помочь оптимизировать их выполнение. Это приводит к более быстрому выполнению запросов и улучшению общей производительности базы данных.
* **Обеспечение безопасности**. Представления могут использоваться для ограничения доступа к конфиденциальным данным. Это помогает гарантировать, что только авторизованные пользователи имеют доступ к чувствительной информации.
* **Структурная простота**. С помощью представлений можно создать собственную структуру базы данных для каждого пользователя.
* **Защита от изменений**. Представление может возвращать непротиворечивый и неизменный образ структуры базы данных, даже если исходные таблицы разделяются, реструктуризуются или переименовываются.
  1. Возможно ли создание представления, включающего информацию из нескольких таблиц одновременно?

Ответ: Да, возможно создание представления, включающего информацию из нескольких таблиц одновременно.

* 1. Какие требование предъявляются к обновляемым представлениям?

Ответ:

Некоторые требования к обновляемым представлениям в Microsoft SQL Server:

1. Любое изменение (UPDATE, INSERT или DELETE) должно касаться столбцов только одной базовой таблицы.
2. Изменяемые в представлении столбцы должны напрямую ссылаться на данные столбцов базовой таблицы. То есть нельзя внести изменения в столбцы, которые были сформированы в представлении.
   1. Как создать обновляемое представление?

Ответ:

Чтобы создать обновляемое представление в Microsoft SQL Server, необходимо выполнить следующие условия:

1. Любое изменение (UPDATE, INSERT или DELETE) должно касаться столбцов только одной базовой таблицы.
2. Столбцы, данные которых нужно изменить, должны напрямую ссылаться на столбцы базовой таблицы. То есть нельзя внести изменения в столбцы, которые были сформированы в представлении, например, агрегатной функцией или другими вычислениями.
3. Вывод:
   1. В ходе лабораторной работы научились создавать и использовать представления в MS SQL Server;
   2. В ходе лабораторной работы закрепили 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. Контрольные вопросы:
   1. Что такое функции пользователя и для чего они применяются?

Ответ:

Функции пользователя — это самостоятельные объекты базы данных, которые представляют собой группу произвольных операторов, предназначенных для выполнения определённой задачи. Они располагаются в определённой базе данных и доступны только в её контексте.

* 1. Чем отличается скалярная функция от табличной?

Ответ:

Скалярная функция возвращает одно значение (например, число, строку, дату) и используется в запросе для преобразования входных значений в выходные.

Табличная функция возвращает таблицу в качестве своего набора результатов.

* 1. Как объявить переменную и присвоить ей значение?

Ответ:

Чтобы объявить переменную в SQL и немедленно присвоить ей значение, нужно использовать код:

DECLARE @MyVar INT = 10;

* 1. Как указать в функции параметр по умолчанию?

Ответ:

Чтобы указать в функции SQL Server параметр по умолчанию, нужно использовать ключевое слово DEFAULT.

* 1. Как вызвать скалярную функцию?

Ответ: Для вызова скалярной функции в SQL Server используется ключевое слово SELECT.

* 1. Как вызвать табличную функцию?

Ответ: Чтобы вызвать табличную функцию в SQL Server, нужно использовать её в предложении FROM инструкции SELECT. При этом перед именем функции ставится префикс имени схемы, например «dbo».

1. Вывод
   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.

**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. Контрольные вопросы
   1. Что такое хранимые процедуры и для чего они применяются?

Ответ:

Хранимая процедура — это предварительно скомпилированный набор операторов SQL, который хранится в базе данных и может быть выполнен одной командой.

* 1. Чем отличается функция пользователя от хранимой процедуры?

Ответ:

Некоторые отличия функции пользователя от хранимой процедуры:

* **Возвращаемое значение**. Функции всегда возвращают значение, в то время как хранимые процедуры могут этого не делать.
* **Входные параметры**. Функции могут иметь входные параметры, но не могут иметь выходные. Хранимые процедуры могут иметь как входные, так и выходные параметры.
* **Выполнение**. SQL-функции выполняются как часть оператора, в то время как хранимые процедуры выполняются как независимые единицы кода.
* **Сложность**. Функции обычно проще и направлены на выполнение конкретных задач, в то время как хранимые процедуры могут быть более сложными и включать в себя несколько операторов SQL и логику программирования.
  1. Как задать входные параметры хранимой процедуры?

Ответ:

Чтобы задать входные параметры хранимой процедуры, нужно указать их после команды CREATE PROC и выше команды AS.

Каждая спецификация входного параметра состоит из имени параметра, за которым следует тип данных. Имена параметров должны начинаться с символа @.

* 1. Как задать выходные параметры хранимой процедуры?

Ответ:

Чтобы задать выходные параметры хранимой процедуры, нужно использовать ключевое слово OUTPUT. Оно означает, что параметр предназначен для возвращения данных из хранимой процедуры.

* 1. Как вызвать выполнение хранимой процедуры?

Ответ:

* В обозревателе объектов подключитесь к экземпляру SQL Server или базе данных SQL Azure.
* Разверните нужную базу данных, разверните узлы «Программирование» и «Хранимые процедуры».
* Щёлкните правой кнопкой мыши на хранимой процедуре, которую хотите запустить, и выберите команду «Выполнить хранимую процедуру».
* В диалоговом окне «Процедура выполнения» укажите имя каждого параметра, тип данных и, если процедура является выходной, нужно указать, является ли она выходной.
* Для каждого параметра в поле Value введите значение, используемое для параметра. В разделе «Передать значение NULL» выберите, следует ли передавать значение NULL в качестве значения параметра.
* Нажмите кнопку «ОК», чтобы выполнить хранимую процедуру. Если процедура не имеет параметров, просто нажмите кнопку «ОК».

1. Вывод
   1. В ходе лабораторной работы научились создавать и использовать хранимые процедуры в MS SQL Server;
   2. В ходе лабораторной работы закрепили 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. Контрольные вопросы
   1. Что такое триггер?

Ответ: Триггер в SQL Server — это специальный тип хранимой процедуры, который вызывается автоматически при выполнении определённого действия над таблицей или представлением.

* 1. Чем триггер отличается от хранимой процедуры?

Ответ:Триггер выполняется автоматически при возникновении события (вставка, удаление и обновление) в таблице, в которой определён триггер. Хранимая процедура запускается при её вызове.

* 1. Как запустить триггер на выполнение?

Ответ:

Триггер запускается автоматически при наступлении определённого события. Событие триггера описывается в виде логических условий.

* 1. Каково назначение триггеров?

Ответ: Назначение триггеров в SQL Server — это автоматическое выполнение определённых действий при выполнении определённого действия над таблицей или представлением, в частности, при добавлении, изменении или удалении данных, то есть при выполнении команд INSERT, UPDATE, DELETE.

* 1. Чем отличаются триггеры INSTEAD OF и AFTER?

Ответ:Триггеры типа AFTER вызываются после выполнения действия, запускающего триггер, а триггеры типа INSTEAD OF выполняются вместо действия, запускающего триггер. Триггеры AFTER можно создавать только для таблиц, а триггеры INSTEAD OF - как для таблиц, так и для представлений.

1. Вывод
   1. В ходе лабораторной работы научились создавать и использовать триггеры в MS SQL Server;
   2. В ходе лабораторной работы закрепили 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. Научиться назначать привилегии пользователю БД;
   3. Закрепить навык создания объектов БД.
2. Контрольные вопросы
   1. В чем отличие между именами входа и пользователями БД?

Ответ:

Отличие имён входа и пользователей базы данных (БД) в SQL Server заключается в следующем:

1. **Имена входа** — это отдельные учётные записи пользователей для входа в ядро СУБД SQL Server. С их помощью проходит процесс аутентификации.
2. **Пользователи базы данных** — это участники базы данных, привязанные к определённому имени входа. Как правило, имя пользователя базы данных совпадает с именем входа, хотя оно не должно совпадать. Один пользователь базы данных сопоставляется с одним именем входа.
   1. Как идентифицируются пользователи в MS SQL Server?

Ответ:

Идентификация пользователей в MS SQL Server происходит в два этапа:

1. **На первом уровне** создаётся учётная запись пользователя (login). Она позволяет подключиться к серверу, но не даёт автоматического доступа к базам данных.
2. **На втором уровне** для каждой базы данных SQL-сервера на основании учётной записи создаётся запись пользователя. На основе прав, выданных пользователю как пользователю базы данных (user), его регистрационное имя (login) получает доступ к соответствующей базе данных.
   1. На какие уровни разделяется система безопасности MS SQL Server?

Ответ:

Система безопасности MS SQL Server разделяется на три уровня:

1. **Уровень сервера**. На этом уровне можно раздать права на базы данных, учётные записи, роли сервера и группы доступности.
2. **Уровень базы данных**. Включает в себя схемы, пользователей базы данных, роли базы данных и полнотекстовые каталоги.
3. **Уровень схемы**. Включает такие объекты, как таблицы, представления, функции и хранимые процедуры.
   1. Каково назначение ролей сервера?

Ответ:Назначение ролей сервера — это определение основной функции компьютера и предоставление определённых возможностей для нескольких пользователей или других компьютеров в сети.

* 1. Каково назначение ролей БД?

Ответ: Назначение ролей базы данных заключается в управлении правами доступа к ней. Роль может рассматриваться как пользователь базы данных или группа пользователей, в зависимости от того, как она установлена.

1. Вывод
   1. В ходе лабораторной работы научились использовать системные хранимые процедуры и DDL-команды для управления именами входа и пользователями БД в СУБД;
   2. В ходе лабораторной работы научились назначать привилегии пользователю БД;
   3. В ходе лабораторной работы закрепили 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. Контрольные вопросы
   1. Для чего создаются резервные копии БД?

Ответ:Резервные копии баз данных (БД) создаются для защиты данных от потери или повреждения в случае сбоев в работе системы, вирусных атак, ошибок администрирования и других непредвиденных ситуаций.

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

Ответ:

Отличие между полным и разностным резервным копированием заключается в том, как сохраняются изменения данных.

**Полное резервное копирование** создаёт полную копию всех данных на диск или ленту для хранения. В ней содержатся данные, необходимые для полного восстановления базы данных.

**Разностное резервное копирование** сохраняет только те изменения данных, которые произошли после создания полной резервной копии.

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

Ответ:

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

Пример расписания резервного копирования для небольшой системы в условиях «8 на 5» (8-часовой рабочий день 5 дней в неделю): полное резервное копирование каждый вечер, журнал транзакций нужно копировать только раз в день (в зависимости от размера журнала транзакций и количества выполненных транзакций).

* 1. Какие скрипты можно сформировать для объектов БД?

Ответ:

Для объектов базы данных можно сформировать различные скрипты, например:

* **Скрипты для таблиц**. Они могут содержать инструкции создания таблиц (CREATE) и заполнения их данными (инструкции INSERT).
* **Скрипты для представлений**.В них можно определить представления, функции, хранимые процедуры, триггеры, ограничения и индексы.
* **Скрипты для хранимых процедур**. Для их создания нужно развернуть опцию «Хранимые процедуры» и выбрать нужные процедуры.
  1. Как выполнить восстановление БД из резервной копии?

Ответ:

**Чтобы восстановить базу данных из резервной копии с помощью SQL Server Management Studio, нужно**:

1. Запустить утилиту SQL Server Management Studio.
2. Подключиться к серверу под учётной записью администратора или владельца БД.
3. Нажать правой кнопкой мыши на разделе «Базы данных» и выбрать меню «Восстановить базу данных».
4. На странице «Общие» выполнить следующие действия:
   * В поле «В базу данных» ввести имя для восстанавливаемой базы.
   * Установить переключатель «С устройства» и указать путь к файлу резервной копии, нажав кнопку «…».
   * Установить галочку «Восстановить» в нужной строке (которых может быть несколько, если один файл \*.bak содержит несколько резервных копий базы).
5. На странице «Параметры» установить галочку «Перезаписать существующую базу данных» и проверить пути в списке «Восстановить файлы базы данных как» (должны указывать на существующую папку на SQL-сервере, к которой предоставлены права на запись).
6. Нажать кнопку «ОК».
   1. Какая команда выполняет восстановление БД из резервной копии?

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

* 1. В каком порядке надо восстанавливать резервные копии?

Ответ:

1. Проверка исправности и работоспособности средств обеспечения функционирования ИС. При необходимости — восстановление работоспособности (ремонт или замена).
2. Проверка исправности и работоспособности средств вычислительной техники. При необходимости — восстановление работоспособности (ремонт или замена).
3. Проверка правильности функционирования общего программного обеспечения ИС. При необходимости — восстановление нормального функционирования общего программного обеспечения ИС с использованием дистрибутивов и обновлений к ним или резервных копий настроек.
4. Проверка правильности функционирования средств защиты информации. При необходимости — восстановление нормального функционирования средств защиты информации с использованием дистрибутивов и обновлений к ним.
5. Проверка правильности функционирования специального программного обеспечения ИС. При необходимости — восстановление нормального функционирования специального программного обеспечения ИС с использованием дистрибутивов и обновлений к ним.
6. Восстановление баз защищаемой информации с использованием резервной копии, при необходимости.
   1. Какой параметр у команды восстановления данных отключает/запускает восстановление БД?

Ответ:

**Параметры RESTORE WITH RECOVERY и RESTORE WITH NORECOVERY** в команде восстановления данных отключают и запускают восстановление базы данных.

**RESTORE WITH RECOVERY** восстанавливает базу данных после восстановления последней резервной копии. Это параметр по умолчанию.

**RESTORE WITH NORECOVERY** оставляет базу данных в состоянии восстановления, что позволяет восстановить дополнительные резервные копии в текущем пути восстановления.

* 1. Для чего используется мастер импорта и экспорта?

Ответ: Мастер импорта и экспорта используется в SQL Server для копирования данных из источника в место назначения.

1. Вывод
   1. В ходе лабораторной работы научились выполнять резервное копирование БД;
   2. В ходе лабораторной работы научились сохранять во внешних файлах описание структуры и данные БД;
   3. В ходе лабораторной работы научились выполнять восстановление БД.

**Лабораторная работа №12**

**«Экспорт данных базы в документы пользователя»**

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

Ответ:

Чтобы выполнить экспорт CSV-файла в SQL Server Management Studio (SSMS), нужно:

1. Подключиться к базе данных SQL Server с помощью SSMS.
2. Нажать правой кнопкой мыши на базу данных, из которой нужно экспортировать данные, и выбрать «Задачи» > «Экспорт данных».
3. Следовать шагам мастера, чтобы указать источник данных, назначение и параметры форматирования. В качестве типа назначения выбрать «Плоский файл», а формата файла — CSV.
4. Сопоставить столбцы источника с столбцами назначения и указать дополнительные параметры форматирования.
5. Просмотреть сводку и нажать «Finish» для экспорта данных в CSV.
   1. Как выполнить экспорт xlsx-файла в Management Studio?

Ответ:

Чтобы выполнить экспорт xlsx-файла в Management Studio, можно использовать мастер импорта и экспорта SQL Server:

1. В SQL Server Management Studio подключитесь к экземпляру ядра СУБД SQL Server.
2. Разверните узел «Базы данных».
3. Щёлкните базу данных правой кнопкой мыши.
4. Выберите «Задачи».
5. Выберите «Импортировать данные» или «Экспортировать данные».
6. В открывшемся мастере подключитесь к источнику данных Excel.
   1. Как в Excel открыть файл формата txt/csv и xml?

Ответ:

**Чтобы открыть в Excel файл формата TXT или CSV**, нужно:

1. Перейти в раздел «Файл» и выбрать «Открыть».
2. Перейти в расположение, содержащее текстовый файл.
3. Выбрать «Текстовые файлы» в раскрывающемся списке «Тип файла» в диалоговом окне «Открыть».
4. Найти нужный файл и дважды щёлкнуть его.
5. Если это текстовый файл (с расширением TXT), запустится мастер импорта текста. Окончив работу с мастером, нажать кнопку «Готово», чтобы завершить импорт.
   1. Данные из скольких таблиц могут храниться в файле csv?

Ответ: В файле CSV могут храниться данные из нескольких таблиц.

* 1. Как экспортировать данные в файл формата JSON?

Ответ:

Чтобы экспортировать данные в файл JSON, например, в Firebase, нужно:

1. Выбрать узел, который нужно экспортировать. Вместе с ним будут экспортированы все дочерние узлы.
2. Нажать на значок меню действий и выбрать «Экспорт в файл JSON».
3. Браузер начнёт скачивание файла с данными.
   1. Как экспортировать данные в файл формата XML?

Ответ:

Чтобы экспортировать данные в файл формата XML из Excel, нужно:

1. На вкладке «Разработчик» нажать кнопку «Экспорт».
2. В небольшом диалоговом окне «Экспорт XML» щёлкнуть карту XML, которую нужно использовать, и нажать кнопку «ОК».
3. В большом диалоговом окне «Экспорт XML» в поле «Имя файла» ввести имя файла XML-данных.
4. Нажать кнопку «Экспорт».
5. Вывод
   1. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Как выполнить импорт xlsx-файла в Management Studio?

Ответ:

**Чтобы выполнить импорт xlsx-файла в Management Studio, нужно**:

1. Открыть SQL Server Management Studio (SSMS) и убедиться, что подключён правильный экземпляр SQL Server. На левой панели должен быть список баз данных.
2. Создать новую базу данных для хранения данных. Для этого нужно нажать правой кнопкой мыши на «Базы данных» и выбрать «Новая база данных». Название базы должно быть связано с импортируемыми данными. Например, если файл Excel содержит данные о продажах, можно назвать его «SalesData».
3. Создать таблицу в новой базе данных для хранения данных из Excel. Для этого нужно нажать правой кнопкой мыши на папку «Таблицы» и выбрать «Новая таблица». Определить столбцы в соответствии с полями в файле Excel. Нужно указать типы данных для каждого столбца, например varchar, int или дата.
4. Открыть мастер импорта. Для этого нужно нажать правой кнопкой мыши на базу данных, выбрать «Задачи» и «Импорт данных». В мастере выбрать источник данных как Microsoft Excel и выбрать файл Excel.
5. Соотнести источник и назначение. Для этого нужно сопоставить столбцы из файла Excel со столбцами в таблице SQL. Нужно дважды проверить сопоставление, чтобы всё было правильно. Если в файле Excel есть заголовки, следует проверить соответствующее поле.
6. Запустить процесс импорта. Для этого нужно следовать подсказкам мастера импорта. Чтобы выполнить импорт, нажать «Finish».
   1. Как выполнить импорт csv-файла в Management Studio?

Ответ:

Чтобы выполнить импорт CSV-файла в Management Studio, можно следовать таким шагам:

1. **Убедитесь, что в базе данных уже есть таблица**, в которую будет импортироваться CSV-файл. Если таблицы нет, создайте её в инструменте Management Studio.
2. **Откройте Management Studio и войдите в целевую базу данных**. Правой кнопкой мыши откройте обозреватель объектов, выберите всю базу, а не одну из её таблиц.
3. **В обозревателе объектов наведите курсор на опцию «Задача» и найдите «Импорт данных»**. Выберите эту опцию.
4. **Нажмите кнопку «Далее»**, чтобы открыть страницу «Выбор источника данных».
5. **В окне «Выбор источника данных» выберите «Flat File Source»**. Нажмите кнопку «Обзор» рядом с именем файла, чтобы открыть страницу «Проводник Windows».
6. **Выберите CSV-файл**, который нужно импортировать. После выбора файла настройте, как нужно импортировать данные в базу данных.
7. **Выберите флажок рядом с опцией «Названия столбцов в первой строке данных»**. Убедитесь, что названия столбцов соответствуют заголовкам в первой строке импортируемого CSV-файла.
8. **Нажмите кнопку «Далее»**. На следующей странице выберите детали базы данных назначения, такие как имя базы данных, имя сервера, тип аутентификации и тип базы данных назначения.
9. **Нажмите «Далее»**, и на следующей странице сервер SQL от вашего имени выберет таблицу. Если это не так, создайте таблицу или выберите другую из списка.
10. **Нажмите кнопку «Изменить сопоставления»**, чтобы изменить детали таблицы.
11. **Нажмите кнопку «Далее»**. Система предложит сохранить импортируемый файл как пакет SSIS. Если не нужно сохранять файл как пакет SSIS, оставьте эту опцию незадействованной.
12. **Проверьте опцию «Выполнить немедленно»** и нажмите кнопку «Далее». Система предложит экран проверки. Если всё в порядке, нажмите кнопку «Завершить» для импорта CSV-файла в выбранную базу данных назначения.
    1. Как выполнить импорт файла XML?

Ответ:

Чтобы выполнить импорт файла XML в программе Excel, нужно:

1. Перейти во вкладку «Разработчик».
2. На ленте в блоке инструментов «XML» нажать на кнопку «Импорт».
3. В открывшемся окне импорта перейти в директорию, где располагается нужный документ.
4. Выбрать его и нажать на кнопку «Импорт».
   1. Как выполнить импорт файла JSON?

Ответ: Для импорта файла JSON в SQL Server можно использовать функцию OPENROWSET(BULK). Она считывает данные из любого файла на локальном диске или в сети, если у SQL Server есть доступ на чтение к этому расположению. Функция возвращает таблицу с одним столбцом, включающим содержимое файла.

1. Вывод
   1. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Какое назначение у элемента SqlConnection?

Ответ: Назначение класса SqlConnection — установление соединения с базой данных SQL Server.

* 1. Какое назначение у элемента SqlCommand?

Ответ: Элемент SqlCommand используется для выполнения SQL-запросов и хранимых процедур в базе данных.

* 1. Какое назначение у элемента SqlDataReader?

Ответ: Назначение элемента SqlDataReader в ADO.NET — быстрая и эффективная загрузка данных из баз данных SQL Server. Он позволяет читать данные в однонаправленном потоке, минимизируя использование памяти и делая его идеальным для больших наборов данных.

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

Ответ: Для реализации подключения к СУБД MS SQL Server требуется подключить пространство имён System.Data.SqlClient.

Для подключения к MySQL используется пространство имён MySql.Data.MySqlClient.

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

Ответ:

Некоторые методы класса SqlCommand, позволяющие выполнить SQL-запрос:

* **ExecuteNonQuery**. Выполняет SQL-выражение и возвращает количество изменённых записей. Подходит для SQL-выражений INSERT, UPDATE, DELETE.
* **ExecuteReader**. Выполняет SQL-выражение и возвращает строки из таблицы. Подходит для SQL-выражения SELECT.
* **ExecuteScalar**. Выполняет SQL-выражение и возвращает одно скалярное значение, например, число. Подходит для SQL-выражения SELECT в паре с одной из встроенных функций SQL, как например, Min, Max, Sum, Count.

1. Вывод
   1. В ходе лабораторной работы научились создавать приложение C# для организации доступа к БД.
   2. В ходе лабораторной работы 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. Научиться создавать приложение C# для организации доступа к БД.
   2. Научиться проектировать приложение, использующее паттерн репозиторий и Dapper.
2. Контрольные вопросы
   1. Что такое Dapper и для чего используется?

Ответ: Dapper — это объектно-реляционный маппер (ORM) для платформы Microsoft .NET. Он обеспечивает лёгкий, высокопроизводительный доступ к данным с минимальной абстракцией.

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

Ответ:

Некоторые методы Dapper, позволяющие извлечь данные из базы данных:

* **Query**. Выполняет SQL-запрос и преобразует его в список объектов, переданных через тип шаблона.
* **QueryMultiple**. Позволяет выполнять пакетные запросы и получать несколько наборов результатов за одно обращение к базе данных.
* **Get**. Извлекает одну запись из базы данных на основе переданного в качестве аргумента идентификатора и заполняет данные в объектной модели.
  1. Какие методы Dapper позволяют изменить данные в БД?

Ответ:

Для изменения данных в базе данных с помощью Dapper можно использовать метод Execute.

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

* 1. Зачем используется паттерн «репозиторий»?

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

1. Вывод
   1. В ходе лабораторной работы научились создавать приложение C# для организации доступа к БД.
   2. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Что такое «ORM»?

Ответ: ORM (Object-Relational Mapping) — это технология, которая облегчает взаимодействие между объектно-ориентированным программированием и реляционными базами данных.

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

Ответ: Entity Framework (EF) Core — это простая, кроссплатформенная и расширяемая версия популярной технологии доступа к данным Entity Framework с открытым исходным кодом.

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

Ответ: 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 базы данных объектов.

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

Ответ: Чтобы выполнить редактирование (вставку, обновление, удаление данных) с помощью EF Core, нужно использовать метод DbContext.SaveChanges(). Он сохраняет изменения в базе данных.

1. Вывод
   1. В ходе лабораторной работы научились создавать приложение C# для организации доступа к БД.
   2. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Для чего используются метод OrderBy, OrderByDescending, ThenBy, ThenByDescending и в чем их отличие?

Ответ:

Методы OrderBy, OrderByDescending, ThenBy и ThenByDescending используются для сортировки данных в LINQ.

OrderBy сортирует значения в порядке возрастания на основе ключа. OrderByDescending сортирует значения в порядке убывания. ThenBy выполняет вторичную сортировку в порядке возрастания. ThenByDescending выполняет вторичную сортировку в порядке убывания.

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

Ответ:

**Методы Take и Skip используются для ограничения объёма данных**. Метод Take выбирает определённое количество элементов из последовательности, начиная с начала, а Skip пропускает указанное количество элементов в последовательности и возвращает оставшиеся элементы.

**При пагинации эти методы применяются следующим образом**:

1. **Skip** пропускает нужное количество элементов, чтобы достичь нужной страницы. [1](https://metanit.com/sharp/aspnet5/12.7.php)
2. **Take** выбирает нужную порцию элементов, которая равна размеру страницы.
   1. Для чего используется метод Where?

Ответ: Оператор WHERE в SQL обеспечивает фильтрацию данных на основе заданных условий, что позволяет извлекать, обновлять или удалять именно те данные, которые соответствуют определенным критериям.

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

Ответ:

При составлении условий могут применяться следующие логические операторы:

* **and** (логическое «И») для двух условий. Возвращает True, если оба условия истинны, иначе возвращает False.
* **or** (логическое «ИЛИ») для двух условий. Возвращает False, если оба условия ложны, иначе возвращает True.
* **not** (логическое «НЕ») для одного условия. Возвращает False для истинного условия, и наоборот.
  1. Как проверить, что значение есть в списке?

Ответ: Чтобы проверить, что значение есть в списке на C#, можно использовать метод Contains.

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

Ответ: Чтобы проверить, что строка начинается с определённого текста в C#, можно использовать метод StartsWith().

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

Ответ:

Чтобы проверить, что строка содержит определённый текст, можно использовать следующие методы:

* **Оператор in**. Возвращает True, если указанная подстрока является частью строки, в противном случае — False. [3](https://pythonist.ru/python-poisk-v-stroke/)
* **Метод find**. Возвращает целое число — индекс начала подстроки в строке, если она есть, или -1 — если подстрока не найдена. [3](https://pythonist.ru/python-poisk-v-stroke/)
* **Метод includes()**. Проверяет с учётом регистра, содержит ли строка заданную подстроку, и возвращает true или false.
  1. Как указать в Select список требуемых данных?

Ответ:

**Чтобы указать список требуемых данных в операторе select на C#, нужно передать ему делегат, в котором выбираются нужные свойства модели**.

Например, чтобы получить только имена сотрудников из списка, можно использовать следующее выражение:

var names = employees.Select(e => e.Name);

1. Вывод
   1. В ходе лабораторной работы научились создавать приложение C# для организации доступа к БД средствами EF Core.
   2. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Для чего используются методы Add() и AddRange() в EF Core?

Ответ:

Метод Add() в EF Core используется для добавления одного объекта. Он устанавливает значение Added в качестве состояния нового объекта, после чего метод SaveChanges() сгенерирует выражение INSERT для вставки модели в таблицу.

Метод AddRange() позволяет добавить сразу несколько объектов. Он прикрепляет коллекцию объектов к объекту контекста в состоянии Added, и при вызове метода SaveChanges() для всех объектов выполняется SQL-команда INSERT в базе данных.

Таким образом, основное различие в использовании методов в том, что Add() добавляет один объект, а AddRange() — сразу несколько.

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

Ответ:

Метод Update() в Entity Framework Core используется для того, чтобы прикрепить объект к контексту в изменённом состоянии.

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

Ответ:

Методы Remove() и RemoveRange() в EF Core используются для удаления объектов.

Метод Remove() устанавливает статус объекта в Deleted и прикрепляет его к DbContext, после чего при выполнении метода SaveChanges() Entity Framework сгенерирует SQL-выражение DELETE.

Метод RemoveRange() используется для удаления сразу нескольких объектов, прикрепляя к DbContext коллекцию или массив объектов в состоянии Deleted и начиная их отслеживать.

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

Ответ: Чтобы сохранить изменения в базе данных с помощью EF Core, можно использовать метод DbContext.SaveChanges. Он сохраняет все изменения, сделанные в этом контексте, в базу данных.

* 1. Как изменить значения полей объекта?

Ответ: Чтобы изменить значения полей объекта, можно использовать установщик, если он есть у класса. Также можно создать новый объект, используя существующий (например, с помощью шаблона builder).

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

Ответ: По умолчанию для свойств первичных ключей, которые представляют типы int или GUID и которые имеют значение по умолчанию, генерируется значение при вставке в базу данных. Например, для свойств типа int это значение **0.**

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

Ответ:

Несколько способов передать данные с одной формы на другую:

1. Через конструктор дочерней формы.
2. Через public-переменную или свойство класса дочерней формы.
3. Через временное хранилище.
4. Через оповещения.
5. Вывод
   1. В ходе лабораторной работы научились выполнять вставку, обновление и удаление записей средствами EF Core;
   2. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Как выполнить команду на выборку данных в EF Core?

Ответ:

Чтобы выполнить команду на выборку данных в EF Core, можно использовать метод LINQ Select(). Он позволяет выбрать один или несколько столбцов из таблицы. Для выборки одного объекта можно использовать метод Find().Также для получения первого элемента выборки, который соответствует определённому условию или набору условий, можно использовать методы Linq First() или FirstOrDefault().

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

Ответ:

Чтобы выполнить команду на модификацию данных в EF Core, нужно:

1. Получить объект, запросив его из базы данных.
2. Сделать изменения в объекте.
3. Вызвать метод SaveChanges() DbContext для распространения изменений в базу данных.
   1. Как объявить и передать параметр в SQL-команду в EF Core?

Ответ: Чтобы объявить и передать параметр в SQL-команду в EF Core, можно использовать класс SqlParameter. Он позволяет задать параметр, который затем передаётся в запрос SQL.

* 1. Как вызвать табличную функцию в EF Core?

Ответ:

Чтобы вызвать табличную функцию в EF Core, можно использовать следующий подход:

1. **Создать в классе контекста метод, который проецируется на хранимую функцию**. Через этот метод можно вызывать функцию. Например, метод должен принимать параметр и возвращать объект IQueryable.
2. **Зарегистрировать метод в переопределённом методе OnModelCreating() класса контекста**. Для этого нужно вызвать метод HasDbFunction() с помощью вызова метода GetUsersByAge().
3. **Обратиться к хранимой функции в коде**. Например, с помощью метода db.Users.FromSqlRaw("SELECT \* FROM GetUsersByAge (@age)", param).ToList().
   1. Как вызвать хранимую процедуру в EF Core?

Ответ:

Чтобы вызвать хранимую процедуру в EF Core, нужно:

1. **Создать новую миграцию**, которая управляет хранимой процедурой в базе данных. Для этого нужно запустить команду «Add-Migration» в Package Console Manager. В сгенерированном классе миграции метод Up() создаёт хранимую процедуру, а метод Down() — удаляет её.
2. **Выполнить команду «Update-Database»**, чтобы создать хранимую процедуру в базе данных.
3. **Использовать метод FromSqlRaw()** DbSet для выполнения хранимой процедуры. При этом результат, возвращённый хранимой процедурой, должен быть материализован в соответствующую сущность DbSet.
   1. Как объявить и передать выходной параметр из хранимой процедуры в EF Core?

Ответ: Чтобы объявить и передать выходной параметр из хранимой процедуры в EF Core, нужно определить его в самой процедуре с ключевым словом OUTPUT.

1. Вывод
   1. В ходе лабораторной работы 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. Научиться разграничивать права доступа пользователей на уровне интерфейса приложения;
   2. Научиться изменять настройки подключения к БД средствами Entity Framework Core.
2. Контрольные вопросы
   1. Как изменить настройки подключения к БД в клиентском приложении?

Ответ:

Чтобы изменить настройки подключения к базе данных в клиентском приложении, можно использовать конфигурационный файл приложения. В нём можно переопределить базовые соглашения и точно указать строку подключения к базе данных.

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

Ответ:

Для обеспечения хранения пользователей и ролей пользователей в базе данных и их отображения в виде ERD (диаграммы сущность-связь) можно использовать следующие подходы:

1. **Разделение системы на три основные группы**: [1](https://code-boxx.com/user-roles-login-erd/)
   * **Пользователи**. Представляются как сущности, например, клиенты, сотрудники, администраторы и другие пользователи системы.
   * **Роли и разрешения**. Роли используются для контроля доступа и настройки разрешений.
   * **Сессии и история**. Указывают, кто вошёл в систему, с какого устройства и что сделал пользователь.
2. **Создание таблиц для хранения данных**. Например, таблица для пользователей может содержать информацию о них, а таблица для ролей — данные о ролях (полное имя, краткое описание и т. д.).
3. **Использование искусственного первичного ключа**. Это атрибут, созданный с целью идентификации конкретных экземпляров сущности. Например, искусственный первичный ключ может формироваться простой последовательной нумерацией каждого из экземпляров сущности.
   1. Что такое «авторизация»?

Ответ: Авторизация в EF Core — это процесс определения прав пользователя в системе, к каким ресурсам приложения он имеет право доступа и при каких условиях.

* 1. Что такое «регистрация»?

Ответ: «Регистрация» в Entity Framework Core — это регистрация контекста данных, через который идёт взаимодействие с базой данных.

1. Вывод
   1. В ходе лабораторной работы научились разграничивать права доступа пользователей на уровне интерфейса приложения;
   2. В ходе лабораторной работы научились 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. Контрольные вопросы
   1. Как выполнить импорт из csv-файла в БД через приложение?

Ответ:

Чтобы выполнить импорт из CSV-файла в базу данных через приложение SQL Server, можно использовать мастер импорта неструктурированных файлов:

1. Откройте SQL Server Management Studio.
2. Подключитесь к экземпляру ядра СУБД SQL Server или к узлу localhost.
3. Разверните базы данных, щёлкните правой кнопкой мыши базу данных, наведите указатель на задачи и выберите «Импортировать неструктурированный файл».
4. Выберите пункт «Обзор», чтобы выбрать входной файл. По умолчанию мастер ищет файлы в форматах CSV и TXT.
5. Укажите уникальное имя таблицы, в которую будут импортироваться данные.
6. Нажмите кнопку «Далее».
   1. Как импортировать изображение в БД через приложение?

Ответ:

Чтобы импортировать изображение в базу данных через приложение, можно следовать таким шагам:

1. Сохранить картинку на диске или в облаке и получить её путь или ссылку.
2. Открыть соединение с базой данных и выполнить SQL-запрос на вставку новой записи в таблицу картинок. В запросе нужно указать информацию о картинке, например, название, путь к файлу и описание.
   1. Как выполнить импорт файла JSON в БД через приложение?

Ответ:

1. С помощью SQL Server Integration Services (SSIS).
2. С помощью OPENROWSET.
3. С помощью BULK INSERT.
4. Вывод
   1. В ходе лабораторной работы мы импортировали данные 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. Контрольные вопросы
   1. Какое пространство имен требуется подключить для работы с Word?

Ответ: Для работы с Word требуется подключить пространство имён Microsoft.Office.Interop.Word.

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

Ответ:

Несколько библиотек для сохранения данных в формате DOCX на C#:

* **DocX**.
* **Open XML SDK**.
* **IronWord**.
* **Spire.Doc**.
  1. Как выполнить экспорт данных в формате txt?

Ответ:

Для экспорта данных в формате txt можно воспользоваться следующими методами:

1. **В SQL Server Management Studio (SSMS)**. На панели инструментов выбрать «Запрос» — «Результаты в» — «Результаты в файл». Затем выполнить SQL-запросы, после чего появится предложение сохранить результаты в текстовый файл с расширением .rpt.
2. **С помощью SQLCMD**. Это утилита командной строки SQL Server, с её помощью можно сохранить результаты в файл. Для этого нужно использовать команду:

sqlcmd -i c:\sql\myquery.sql -o c:\sql\myoutput.txt

1. **С помощью PowerShell**. Для экспорта результатов запроса SQL Server в текстовый файл нужно выполнить команду:

Invoke-Sqlcmd -InputFile "C:\sql\myquery.sql" | Out-File -filePath "C:\sql\powershelloutput.txt"

1. Вывод
   1. В ходе лабораторной работе научились выполнять экспорт данных из БД в формате текстового файла.
   2. В ходе лабораторной работе научились 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. Контрольные вопросы
   1. Какое пространство имен требуется подключить для работы с Word?

Ответ: Для работы с Word требуется подключить пространство имён Microsoft.Office.Interop.Word.

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

Ответ:

Несколько библиотек для сохранения данных в формате DOCX на C#:

* **DocX**.
* **Open XML SDK**.
* **IronWord**.
* **Spire.Doc**.
  1. Как выполнить экспорт данных в формате txt?

Ответ:

Для экспорта данных в формате txt можно воспользоваться следующими методами:

1. **В SQL Server Management Studio (SSMS)**. На панели инструментов выбрать «Запрос» — «Результаты в» — «Результаты в файл». Затем выполнить SQL-запросы, после чего появится предложение сохранить результаты в текстовый файл с расширением .rpt.
2. **С помощью SQLCMD**. Это утилита командной строки SQL Server, с её помощью можно сохранить результаты в файл. Для этого нужно использовать команду:

sqlcmd -i c:\sql\myquery.sql -o c:\sql\myoutput.txt

1. **С помощью PowerShell**. Для экспорта результатов запроса SQL Server в текстовый файл нужно выполнить команду:

Invoke-Sqlcmd -InputFile "C:\sql\myquery.sql" | Out-File -filePath "C:\sql\powershelloutput.txt"

1. Вывод
   1. В ходе лабораторной работе научились выполнять экспорт данных из БД в формате табличного файла.
   2. В ходе лабораторной работе научились 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. Контрольные вопросы
   1. Что такое REST-запрос?

Ответ: REST-запрос — это запрос на сервер для получения или изменения данных на сервере в архитектурном стиле REST (Representational State Transfer).

* 1. Что такое RESTful?

Ответ: RESTful API — это интерфейс, используемые двумя компьютерными системами для безопасного обмена информацией через Интернет.

* 1. Для чего используется метод GET?

Ответ: Метод GET используется для запроса данных с сервера. Он предназначен только для чтения, поэтому риск изменения или повреждения данных отсутствует.

* 1. Для чего используется метод POST?

Ответ: POST — метод отправки данных на сервер, например, после заполнения формы регистрации или авторизации на сайте. Информация передаётся в теле запроса и не может быть перехвачена злоумышленниками.

* 1. Для чего используется метод PUT?

Ответ: Метод PUT используется для обновления существующих ресурсов или создания новых, если они не существуют.

* 1. Для чего используется метод DELETE?

Ответ: Метод DELETE в Web API используется для удаления ресурса, который указывается с помощью его URI.

1. Вывод
   1. В ходе практической работы научились 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. Контрольные вопросы
   1. Как указать у объекта HttpClient базовый адрес?

Ответ: Чтобы указать базовый адрес у объекта HttpClient, нужно установить свойство BaseAddress. Оно возвращает или устанавливает базовый адрес в виде объекта URI, который будет использоваться при отправке запросов.

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

Ответ:

При HTTP-запросе могут быть получены **трёхзначные цифровые коды**. Первая цифра в ответе обозначает класс состояния, другие две — причину, по которой мог появиться такой ответ. [1](https://journal.topvisor.com/ru/seo-kitchen/list-of-http-status-codes/)

**Некоторые коды и их значения:**

* **200 OK**. Запрос успешно выполнен. Значение результата «успех» зависит от метода HTTP.
* **201 Created**. Запрос выполнен успешно, и в результате был создан новый ресурс.
* **202 Accepted**. Запрос получен, но ещё не обработан.
* **203 Non-Authoritative Information**. Возвращённые метаданные не полностью совпадают с теми, которые доступны на исходном сервере, а получены из другого источника.
* **204 No Content**. Для этого запроса нет содержимого для отправки, но заголовки ответа могут быть полезны.
* **205 Reset Content**. Запрос успешно выполнен, и браузер должен сбросить введённый контент.
* **206 Partial Content**. Сервер успешно обработал часть GET-запроса, а другую часть вернул.
  1. Какой метод класса HttpClient используется для получения данных?

Ответ: Для получения данных с сервера в классе HttpClient используется метод GetAsync().

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

Ответ: Для вставки данных в классе HttpClient используется метод PostAsync() или PostAsJsonAsync(). В эти методы передаётся адрес ресурса и сами добавляемые данные.

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

Ответ:

* **HttpClient.PutAsync**. Отправляет PUT-запрос на указанный адрес и получает ответ сервера в виде объекта HttpResponseMessage. Метод заменяет существующий ресурс или создаёт новый с помощью полезных данных текста запроса.
* **HttpClient.PatchAsync**. Отправляет Patch-запрос на указанный адрес и получает ответ сервера в виде объекта HttpResponseMessage. Запрос PATCH является частичным обновлением существующего ресурса.
  1. Какой метод класса HttpClient используется для удаления данных?

Ответ: Для удаления данных с помощью класса HttpClient в C# используется метод DeleteAsync.

1. Вывод
   1. В ходе практической работы научились проверять работоспособность RESTful API в клиентском приложении.
   2. В ходе практической работы научились выполнять тестирование RESTful API методом черного ящика.

**Практическая работа №3**

**«Разграничение прав доступа на уровне REST API»**

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

Ответ:

Для настройки доступа к методам REST для авторизованных и неавторизованных пользователей можно указать следующие атрибуты:

* **AllowAny**. Позволяет неограниченный доступ, независимо от того, аутентифицирован ли запрос.
* **IsAuthenticated**. Отказывает в доступе любому неаутентифицированному пользователю, в обратной ситуации разрешает.
* **IsAuthenticatedOrReadOnly**. Разрешает любые запросы для авторизованных пользователей. Запросы для неаутентифицированных пользователей разрешены только для методов GET, HEAD или OPTIONS.
* **DjangoModelPermissions**. Авторизация происходит только если пользователь авторизован и обладает соответствующим разрешением.
  1. Для чего используется JWT?

Ответ: JWT (JSON Web Token) используется для безопасной передачи данных между клиентом и сервером. Как правило, применяется для передачи информации о пользователях для аутентификации, обработки сеансов и контроля доступа.

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

Ответ:

Некоторые отличия авторизации с использованием cookie и JWT с точки зрения безопасности:

* **Хранение**. При использовании JWT состояние аутентификации не сохраняется на стороне сервера, а сохраняется на стороне клиента, в то время как при использовании cookie аутентификация сохраняется на стороне сервера.
* **Отзыв и аннулирование**. С файлами cookie отозвать сессию пользователя легко, в то время как с JWT это сделать сложнее.
* **Уязвимости**. Файлы cookie уязвимы для атак с использованием межсайтового скриптинга (XSS) и подделки межсайтовых запросов (CSRF). JWT, сохранённые в браузере, могут быть уязвимы для атак XSS, если приложение позволяет внедрять внешние сценарии JavaScript.

Выбор между этими методами авторизации с точки зрения безопасности зависит от конкретных требований приложения и необходимого уровня защиты.

* 1. Какие настройки можно указать при создании токена?

Ответ:

Некоторые настройки, которые можно указать при создании токена:

* **Имя токена**. Правильное именование поможет эффективно управлять токенами и, возможно, удалять их, когда они больше не нужны.
* **Время жизни токена**. Указывает, через сколько секунд истечёт время работы токена.
* **Алгоритм для подписи токена**. Например, HMAC (вычисление подписи на основании секретного ключа), RSA (вычисление подписи с помощью асинхронного хеширования RSA) или ECDSA (вычисление подписи с помощью асинхронного хеширования ECDSA).
* **Идентификатор пользователя**. Затем это поле можно будет использовать для идентификации пользователя.
* **Название организации, которая выдала токен**.

Конкретные настройки зависят от используемого инструмента для создания токена.

* 1. Почему для авторизации следует использовать метод POST?

Ответ:

Метод POST следует использовать для авторизации, потому что он обеспечивает большую степень защиты данных, чем GET. [5](https://guruweba.com/html/metody-get-i-post-ispolzovanie-i-otlichiya/)

Запрос POST передаёт данные в теле запроса, и параметры не видны пользователю без использования специального ПО, что даёт методу преимущество при пересылке конфиденциальных данных, например в формах авторизации.

1. Вывод
   1. В ходе практической работы научились выполнять разработку web-API для доступа к БД.
   2. В ходе практической работы научились 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. Контрольные вопросы
   1. Как выполнить HTTP-запрос к API с помощью Fetch API?

Ответ: Чтобы выполнить HTTP-запрос к API с помощью Fetch API, нужно вызвать метод fetch(). 2 Он принимает два аргумента: URL-адрес ресурса и необязательный параметр options, который содержит дополнительные настройки запроса.

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

Ответ: Для обработки ошибок при вызове API в JavaScript можно использовать конструкцию try...catch. Она позволяет «ловить» ошибки и вместо падения скрипта делать что-то более осмысленное. 4

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

Ответ: Асинхронные функции позволяют выполнять долгие операции (например, вызовы API) без блокировки программы. Они возвращают “промис”, а await позволяет дождаться результата. При вызове API асинхронные функции отправляют запрос и позволяют программе продолжать работу, а await “ждет” ответа сервера, не замораживая интерфейс.

* 1. Что такое Razor Pages?

Ответ: Razor Pages - это упрощенный подход к созданию веб-страниц в ASP.NET Core. Вместо традиционного разделения на контроллеры и представления, Razor Pages организуют код, связанный с конкретной страницей, в одном файле .cshtml. Это делает разработку проще, особенно для простых веб-приложений, и уменьшает количество файлов в проекте.

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

Ответ:

1. **Создать файл:** Добавьте .cshtml файл в папку Pages (или подпапку).
2. **Код:**
   * @page - директива в начале файла, делающая его Razor Page.
   * @model - (опционально) указывает класс модели страницы.
   * C# код в блоках @{}, HTML в остальной части.
3. **Маршрутизация:** Доступ к странице по имени файла (например, /MyPage) или настраивается в Startup.cs.
4. **Обработчики:** Методы (OnGet, OnPost) для обработки запросов
5. Вывод
   1. В ходе практической работы научились разрабатывать клиентское веб-приложение для доступа к БД.
   2. В ходе практической работы научились проверять работоспособность RESTful API в клиентском приложении.