


Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		



УТВЕРЖДЕНО

решением Ученого совета ФМИАТ
от «16» мая 2023 г., протокол № 4/23
Председатель _____ Волков М.А.
(подпись, расшифровка подписи)
«16» мая 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Дисциплина	Безопасность систем баз данных
Факультет	Математики, информационных и авиационных технологий
Кафедра	Информационной безопасности и теории управления (ИБиТУ)
Курс	4

Специальность: 10.05.03 "Информационная безопасность автоматизированных систем"
(код специальности (направления), полное наименование)

Специализация: "Безопасность открытых информационных систем"
полное наименование

Форма обучения: очная
очная, заочная, очно-заочная (указать только те, которые реализуются)

Дата введения в учебный процесс УлГУ: « 01 » 09 2023 г.

Программа актуализирована на заседании кафедры: протокол № 12 от 12.04.2023 г.

Программа актуализирована на заседании кафедры: протокол № 10 от 15.04.2024 г.


Программа актуализирована на заседании кафедры: протокол № __ от _____ 20__ г.

Сведения о разработчиках:

ФИО	Кафедра	Должность, ученая степень, звание
Иванцов Андрей Михайлович	ИБ и ТУ	Кандидат технических наук, доцент
Клочков Андрей Евгеньевич	ИБ и ТУ	Старший преподаватель

СОГЛАСОВАНО

Заведующий выпускающей кафедрой
«Информационная безопасность и теория
управления»

 / Андреев А.С. /
(подпись) (Ф.И.О.)
« 11 » 05 2023 г.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цели освоения дисциплины:

Основной целью освоения дисциплины «Безопасность систем баз данных» является формирование у студентов знаний и умений по работе с базами

Задачи освоения дисциплины:

- принципы проектирования структур БД на основе реляционной алгебры и метода ER-диаграмм;
- освоить основы построения реляционных баз данных;
- получить навыки основ языка SQL и выполнения основных операций с таблицами с помощью операторов SQL.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина «Безопасность систем баз данных» изучается в 7 и 8 семестрах и относится к обязательной части дисциплин блока Б1.О специальности 10.05.03 "Информационная безопасность автоматизированных систем".


Курс учебной дисциплины тесно увязан с другими учебными дисциплинами, в первую очередь с курсами «Информатика», «Открытые информационные системы», «Безопасность операционных систем», «Основы информационной безопасности», «Языки программирования», позволяющими понять сущность методы обеспечения безопасности баз данных

Для освоения дисциплины студент должен иметь следующие «входные» знания, умения, навыки и компетенции:

- знание базовых понятий в области информатики, вычислительной техники и безопасности информации;
- способность использовать нормативные правовые документы;
- способность анализировать проблемы и процессы;
- способность использовать основные законы естественно-научных дисциплин, применять методы математического анализа и моделирования.

Основные положения дисциплины используются в дальнейшем при изучении таких дисциплин как: «Безопасность сетей ЭВМ»; «Разработка и эксплуатация автоматизированных систем в защищенном исполнении»; «Безопасность открытых информационных систем».

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СОТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины направлен на формирование следующих компетенций:


Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
ОПК-12 - Способен применять знания в области безопасности вычислительных сетей, операционных систем и баз данных при разработке автоматизированных систем	<p>Знать: основные принципы обеспечения безопасности вычислительных сетей, операционных систем и баз данных при разработке автоматизированных систем</p> <p>Уметь: применять знания в области безопасности вычислительных сетей, операционных систем и баз данных при разработке автоматизированных систем</p> <p>Владеть: навыками применения знаний в области безопасности вычислительных сетей, операционных систем и баз данных</p>
ОПК-15 - Способен осуществлять администрирование и контроль функционирования средств и систем защиты информации автоматизированных систем, инструментальный мониторинг защищенности автоматизированных систем	<p>Знать: порядок администрирования и контроля функционирования средств и систем защиты информации автоматизированных систем</p> <p>Уметь: осуществлять администрирование и контроль функционирования средств и систем защиты информации автоматизированных систем, инструментальный мониторинг защищенности автоматизированных систем</p> <p>Владеть: навыками администрирования и контроля функционирования средств и систем защиты информации автоматизированных систем, инструментального мониторинга защищенности автоматизированных систем</p>

4. ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

4.1. Объем дисциплины в зачетных единицах (всего) 9

4.2. Объем дисциплины по видам учебной работы (в часах):


Вид учебной работы	Количество часов (форма обучения <u>дневная</u>)			
	Всего по плану	В т.ч. по семестрам		
		7 семестр	8 семестр	
Контактная работа обучающихся с преподавателем	162/162	72/72*	90/90*	
Аудиторные занятия:	162/162	72/72*	90/90*	
Лекции	72/72	36/36*	36/36*	
Практические и	18/18		18/18*	

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

семинарские занятия				
Лабораторные работы (лабораторный практикум)	72/72	36/36*	36/36*	
Самостоятельная работа	90	36	54	
Форма текущего контроля знаний и контроля самостоятельной работы: тестирование, контр. работа, коллоквиум, реферат и др. (не менее 2 видов)		-Тестирование на семинарах; - вопросы при защите лабораторных работ		
Курсовая работа				
Виды промежуточной аттестации (экзамен, зачет)	Экзамен	Экзамен 36	Экзамен 36	
Всего часов по дисциплине:	324	144	180	

* В случае необходимости использования в учебном процессе частично/исключительно дистанционных образовательных технологий в таблице через слеш указывается количество часов работы ППС с обучающимися для проведения занятий в дистанционном формате с применением электронного обучения.


4.3. Содержание дисциплины (модуля.) Распределение часов по темам и видам

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

учебной работы:

Форма обучения _____ очная _____

Название разделов и тем	Всего	Виды учебных занятий					Форма текущего контроля знаний
		Аудиторные занятия			Занятия в интерактивной форме	Самостоятельная работа	
		Лекции	Практ. занятия, семинары	Лабораторные работы			
Раздел 1. Общие понятия о СУБД							
1. Основные определения и принципы построения СУБД	4	2				2	Тесты Т1
2. Распределенные БД	4	2				2	Тесты Т2
3. Назначение и возможности СУБД	8	4				4	Тесты Т3
Раздел 2. Языковые средства							
4. Языковые средства для различных моделей СУБД	16	4		8	4	4	Тесты Т4, Лаб. раб. № 1
5. Манипулирование в реляционных БД	8	2		4		2	Тесты Т5, Лаб. раб. № 2
6. Языковые средства описания данных	12	2		8		2	Тесты Т6, Лаб. раб. № 3-4
7. Особенности языковых средств	14	2		8		4	Тесты Т7, Лаб. раб. № 5-6
Раздел 3. Доступ к БД							
8. Идентификация и аутентификация в СУБД	6	4				2	Тесты Т8
9. Разграничение доступа	6	4				2	Тесты Т9
10. Механизм ролей	12	2		4	4	6	Тесты Т10, Лаб. раб. № 7
11. Аудит событий	6	4				2	Тесты Т11
12. Контроль целостности БД	12	4		4		4	Лаб. раб. № 8
Раздел 4. Администрирование СУБД							
13. Взаимодействие СУБД с базовой ОС	26	6	2	8		10	Тесты Т13, Лаб. раб. № 9-10
14. Журнализация	14	6	2			6	Тесты Т14
15. Средства создания резервных копий и восстановление баз данных	52	6	4	24	24	18	Тесты Т15, Лаб. раб. № 11-13
16. Тиражирование и синхронизация в распределен-	14	6	2			6	Тесты Т16

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

ных БД							
17. Подготовка отчетов в языках СУБД	16	6	4			6	Тесты Т17
18. Средства автоматизации проектирования СУБД	22	6	4	4	4	8	Тесты Т18, Лаб. раб. № 14
Итого:	252	72	18	72	36	90	

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Раздел 1. Общие понятия о СУБД

Тема 1. Основные определения и принципы построения СУБД

1. Общие понятия о СУБД. Определение систем управления баз данных. Отличия СУБД от других систем управления. Назначение СУБД. Архитектура и типы СУБД. Трехуровневая архитектура БД. История создания. Реляционная, иерархическая и сетевая модель баз данных. Нормальные формы.

Тема 2. Распределенные БД

Архитектура и принципы распределённого подхода. Требования и критерии построения информационных систем на базе распределённых баз данных. Многомерное представление данных. Общая схема организации хранилища данных. Распределенные базы данных. Назначение, потребность, принципы организации. Проектирование данных на концептуальном и логическом уровнях. Примеры использования.

Тема 3. Назначение и возможности СУБД

1. Характеристика СУБД, области использования, реализации. Необходимость использования СУБД. Возможности СУБД. Архитектурные решения, используемые при реализации многопользовательских СУБД. Принципы действия централизованной архитектуры, архитектуры «файл-сервер», архитектуры «клиент-сервер». Недостатки и преимущества различных архитектур СУБД

Раздел 2. Языковые средства

Тема 4. Языковые средства для различных моделей СУБД

1. Введение в SQL и другие языки для работы с СУБД. Сравнительные характеристики T-SQL и PL-SQL. Процедурное расширение PL/SQL. Триггеры. Технологии разработки клиентского приложения

Тема 5. Манипулирование в реляционных БД

Описание реляционных баз данных. Понятие таблиц и хранение данных. Возможности СУБД при работе с данными. Язык манипулирования данными (DML). Операции манипулирования.

Тема 6. Языковые средства описания данных


Возможности языка SQL и TSQL. Синтаксис языков для работы с данными. Технологии объектно-реляционного отображения JRM. Разработка консольного приложения. Клиентский интерфейс. Триггеры защиты. Примеры описания данных.

Тема 7. Особенности языковых средств

Отличия различных языков использующихся для СУБД. Описание данных на основе языка SQL. Описание данных на основе языка SQL. Основные отличия и особенности. Отличия SQL от процедурных языков программирования. Инструментарий web-разработчика. Проблема совместимости.

Раздел 3. Доступ к БД

Тема 8. Идентификация и аутентификация в СУБД

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

Основные способы аутентификации. Управление наборами привилегий. Передача привилегий. Понятия пользователя, роли в СУБД. Назначение прав для объектов данных в БД. Процедура проверки пользователя.

Тема 9. Разграничение доступа

Установка доступа к объектам в СУБД. Проблемы доступа. Политика предоставления прав на объекты.

Тема 10. Механизм ролей

Понятие ролей в СУБД. Отличие от групп пользователей. Делегация прав роли. Отслеживания действия роли.

Тема 11. Аудит событий

Настройка политики аудита событий в СУБД. Аудит доступа к объектам. Составление политики аудита на основании требований и модели злоумышленника. Понятие разумного аудита в СУБД.

Тема 12. Контроль целостности БД

Механизм обнаружения проблем в СУБД. Понятие целостности БД, возможности организации бесперебойного функционирования СУБД.

Раздел 4. Администрирование СУБД

Тема 13. Взаимодействие СУБД с базовой ОС

Настройка СУБД в базовой ОС. Выделение ресурсов ОС для нормальной работы СУБД. Определение правил функционирования СУБД в ОС.

Тема 14. Журнализация

Организация ведения журналов событий в СУБД. Анализ журналов, выявления проблем и причин их вызвавших. Корректная настройка журналов.

Тема 15. Средства создания резервных копий и восстановление баз данных

Механизмы резервного копирования и восстановления данных. Разработка политики создания резервных копий. Определение данных подлежащих резервному копированию.

Тема 16. Тиражирование и синхронизация в распределенных БД

Функционирование распределенных БД. Проблемы синхронизации распределенных БД. Понятие тиражирования, основные принципы и концепции.

Тема 17. Подготовка отчетов в языках СУБД

Системы подготовки отчетов в СУБД. Функциональность языков СУБД для подготовки отчетов. Дополнительные модули и компоненты для работы с отчетами.

Тема 18. Средства автоматизации проектирования СУБД

Проектирование СУБД, основные подходы, тонкие моменты. Цели и задачи проектирования. Определение необходимости использования «нормальных форм».

6. ТЕМЫ ПРАКТИЧЕСКИХ И СЕМИНАРСКИХ ЗАНЯТИЙ

6.1 Практические занятия не предусмотрены учебным планом дисциплины.

6.2 Темы семинарских занятий:

Раздел 4. Администрирование СУБД

Тема 13. Взаимодействие СУБД с базовой ОС (семинар).

1. Настройка СУБД в базовой ОС.
2. Выделение ресурсов ОС для нормальной работы СУБД.
3. Определение правил функционирования СУБД в ОС.

Тема 14. Журнализация (семинар).

1. Организация ведения журналов событий в СУБД.
2. Анализ журналов, выявления проблем и причин их вызвавших.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

3. Корректная настройка журналов.

Тема 15. Средства создания резервных копий и восстановление баз данных (семинар).

1. Механизмы резервного копирования и восстановления данных.
2. Разработка политики создания резервных копий.
3. Определение данных подлежащих резервному копированию.

Тема 16. Тиражирование и синхронизация в распределенных БД (семинар).

1. Функционирование распределенных БД.
2. Проблемы синхронизации распределенных БД.
3. Понятие тиражирования, основные принципы и концепции.

Тема 17. Подготовка отчетов в языках СУБД (семинар).

1. Системы подготовки отчетов в СУБД.
2. Функциональность языков СУБД для подготовки отчетов.
3. Дополнительные модули и компоненты для работы с отчетами.

Тема 18. Средства автоматизации проектирования СУБД (семинар).

1. Проектирование СУБД, основные подходы, тонкие моменты.
2. Цели и задачи проектирования.
3. Определение необходимости использования «нормальных форм».

7. ЛАБОРАТОРНЫЕ РАБОТЫ (ЛАБОРАТОРНЫЙ ПРАКТИКУМ)

Раздел 2. Языковые средства

Тема 4. Языковые средства для различных моделей СУБД

Лабораторная работа № 1 (8 часов). «Проектирование и создание базы данных».

Лабораторная работа должна быть выполнена на MS SQL Server Express и на PostgreSQL или MySQL.

Варианты заданий

Необходимо произвести проектирование реляционной базы данных в любой среде проектирования (например, Microsoft Visio). База данных должна состоять минимум из 10 таблиц, которые должны быть связаны всеми типами отношений (1:1 1:n n:n). Минимум 5 из этих таблиц должны быть в третьей нормальной форме. Все таблицы должны быть заполнены разумными данными, не менее 20 записей в каждой. Одна из таблиц должна содержать 1 000 000 (один миллион записей) Одна из таблиц должны содержать следующие сведения:

- Пол
- Фотография
- Место жительства
- Дата рождения
- Номер и серия паспорта (в установленном формате)

Основные этапы проектирования баз данных

А) Концептуальное (инфологическое) проектирование

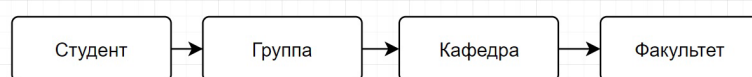
Концептуальное (инфологическое) проектирование — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создается без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и

образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

1. описание информационных объектов или понятий предметной области и связей между ними.
2. описание ограничений целостности, то есть требований к допустимым значениям данных и к связям между ними.

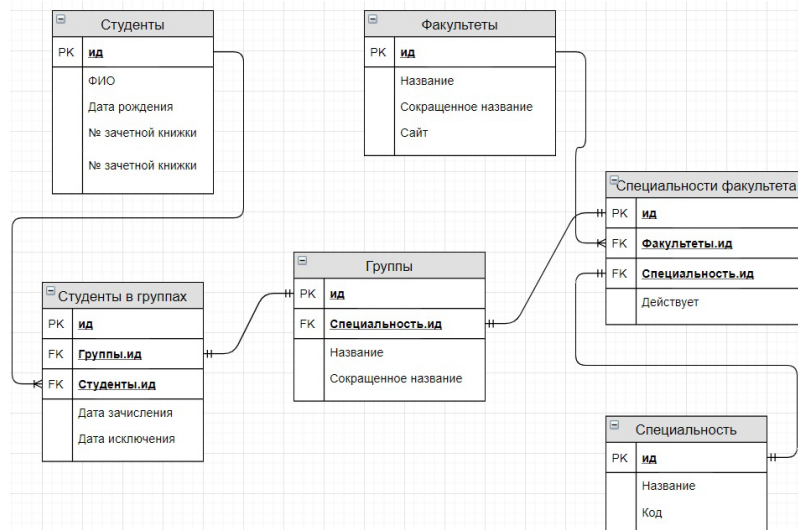


Б) Логическое (даталогическое) проектирование

Логическое (даталогическое) проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.



В) Физическое проектирование

Физическое проектирование — создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д.

Результатом физического проектирования логической схемы выше на языке SQL может являться следующий скрипт:

Дополнительная информация: [Физическое проектирование базы данных MS SQL](#)

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

Server

1. Ход работы

Для проектирования реляционной базы данных можно использовать специализированное программное обеспечение:

- Microsoft Visio
- JetBrains DataGrip
- Diagrams.net

1. Запустим Diagrams.net

2. По заданной предметной области необходимо провести концептуальное и логическое проектирование базы данных на отдельных листах и продемонстрировать результат работы.

3. Провести физическое проектирование базы данных для каждой из СУБД.

4. Создать дополнительные таблицы для генерации фамилий имен и отчеств из архива. [russian_names.zip](#)

* Разработать программу на любом языке программирования, которая добавит в таблицу 1 000 000 записей Фамилий и Имен.

Задание можно выполнить на любом языке программирования.

Тема 5. Манипулирование в реляционных БД

Лабораторная работа № 2 (4 часа). «Основы оператора SELECT»

Возвращает строки из базы данных и позволяет делать выборку одной или нескольких строк или столбцов из одной или нескольких таблиц в SQL Server. Полный синтаксис инструкции SELECT сложен, однако основные предложения можно вкратце описать следующим образом:

Например, сделаем выборку всех записей из таблицы Студенты

Логический порядок обработки инструкции SELECT

Следующие действия демонстрируют логический порядок обработки или порядок привязки инструкции SELECT. Этот порядок определяет, когда объекты, определенные в одном шаге, становятся доступными для предложений в последующих шагах. Например, если обработчик запросов можно привязать (для доступа) к таблицам или представлениям, определенным в предложении FROM, эти объекты и их столбцы становятся доступными для всех последующих шагов. И наоборот, поскольку предложение SELECT является шагом 8, любые псевдонимы столбцов или производных столбцов, определенные в этом предложении, не могут быть объектом для ссылки предыдущих предложений. Вместе с тем к ним могут обращаться последующие предложения, например предложение ORDER BY. Фактическое физическое выполнение инструкции определяется обработчиком запросов и порядок из этого списка может значительно отличаться.

- FROM
- ON
- JOIN
- WHERE
- GROUP BY
- WITH CUBE или WITH ROLLUP
- HAVING
- SELECT
- DISTINCT
- ORDER BY
- В начало

Примеры работы

1. Использование SELECT для получения строк и столбцов

```
SELECT *
FROM [Студенты]
```

Пример с использованием псевдонимов

```
SELECT e.*
FROM [Студенты] as e
```

В ходе выполнения данного примера кода возвращаются все строки и подмножества столбцов ([ид],[ФИО],[Дата рождения] ,[Зачётная книжка]) таблицы [Студенты]. Заголовок последних столбцов переименовываем в [Логин], [Пароль]

```
SELECT [ид]
,[ФИО]
,[Дата рождения]
,[Зачётная книжка]
,[login] AS [Логин]
,[pass] AS [Пароль]
FROM [Студенты]
```

2. Использование SELECT с заголовками столбцов и вычислениями

В следующем примере возвращаются все строки из таблицы Должности и вычисляется заработная плата до вычетов для каждого сотрудника на основе их Оклада умноженные на 12 месяцев

```
SELECT [ид]
,[Название]
,[Оклад] * 12 AS [Годовой доход]
FROM [Должности]
```

3. Совместное использование DISTINCT и SELECT

В следующем примере используется DISTINCT для создания списка всех уникальных должностей в таблице Студенты.

```
SELECT DISTINCT [ФИО]
FROM [Студенты]
```

Подробная информация

Детальная информация о операторе SELECT в TSQL <https://docs.microsoft.com/ru-ru/sql/t-sql/queries/select-transact-sql?view=sql-server-ver15>

Тема 6. Языковые средства описания данных

Лабораторная работа № 3 (4 часа). «Возможности условного оператора (WHERE)».

<https://docs.microsoft.com/ru-ru/sql/t-sql/queries/where-transact-sql?view=sql-server-ver15>


Агрегатные функции

Функция SUM

Функция SUM возвращает сумму значений столбца таблицы базы данных. Она может применяться только к столбцам, значениями которых являются числа. Запросы SQL для получения результирующей суммы начинаются так:

```
SELECT SUM([Оклад])
FROM [Должности]
```

После этого выражения следует FROM (ИМЯ_ТАБЛИЦЫ), а далее с помощью конструкции WHERE может быть задано условие. Кроме того, перед именем столбца

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

может быть указано DISTINCT, и это означает, что учитываться будут только уникальные значения. По умолчанию же учитываются все значения (для этого можно особо указать не DISTINCT, а ALL, но слово ALL не является обязательным).

Функция MIN

Функция SQL MIN также действует в отношении столбцов, значениями которых являются числа и возвращает минимальное среди всех значений столбца. Эта функция имеет синтаксис аналогичный синтаксису функции SUM.

Функция MAX

Аналогично работает и имеет аналогичный синтаксис функция SQL MAX, которая применяется, когда требуется определить максимальное значение среди всех значений столбца.

Функция AVG

Указанное в отношении синтаксиса для предыдущих описанных функций верно и в отношении функции SQL AVG. Эта функция возвращает среднее значение среди всех значений столбца.

Функция COUNT

Функция SQL COUNT возвращает количество записей таблицы базы данных. Если в запросе указать SELECT COUNT(ИМЯ_СТОЛБЦА) ..., то результатом будет количество записей без учёта тех записей, в которых значением столбца является NULL (неопределённое). Если использовать в качестве аргумента звёздочку и начать запрос SELECT COUNT(*) ..., то результатом будет количество всех записей (строк) таблицы.

Агрегатные функции вместе с SQL GROUP BY (группировкой)

Теперь рассмотрим применение агрегатных функций вместе с оператором SQL GROUP BY. Оператор SQL GROUP BY служит для группировки результирующих значений по столбцам таблицы базы данных.

Оператор SQL GROUP BY служит для распределения строк - результата запроса - по группам, в которых значения некоторого столбца, по которому происходит группировка, являются одинаковыми. Группировку можно производить как по одному столбцу, так и по нескольким.

Часто оператор SQL GROUP BY применяется вместе с агрегатными функциями (COUNT, SUM, AVG, MAX, MIN). В этих случаях агрегатные функции служат для вычисления соответствующего агрегатного значения ко всему набору строк, для которых некоторый столбец - общий.

Оператор HAVING


Определяет условие поиска для группы или статистического выражения. Предложение HAVING можно использовать только в инструкции SELECT. HAVING обычно используется с предложением GROUP BY. Если предложение GROUP BY не используется, имеется одна неявная агрегированная группа.

Пример

Найдём группу с минимальным количеством студентов.

– **Способ первый. Выборка всех групп с количеством студентов. Упорядочиваем по возрастанию и выбираем первый элемент. Минусы данного способа - возможно что групп с минимальным количеством студентов больше чем 1.**

```
SELECT TOP 1 -- Выбрать только один первый элемент
[Группы_ид], -- Номер группы
COUNT([Группы_ид]) as [Количество студентов в группах] -- Подсчитать количество
студентов в группе
FROM [dbo].[Студенты] в [группах]
GROUP BY [Группы_ид] -- Агрегатная функция COUNT будет работать для по
```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

группировке *данного* *поля*
ORDER BY [Количество студентов в группах] ASC -- Отсортировать данные по возрастанию значения количества

- Способ второй. Найдем чему равно минимальное количества студентов в группе. Найдем группы с минимальным количеством студентов.
- Способ третий. Оператор HAVING. Может быть использован для фильтрации после применения агрегатных функций.

Лабораторная работа № 4 (4 часа). «Процедуры и функции».

Теоретическая часть. Для упрощения работы с SQL сервером возможно создание пользовательских функций и процедур. Функции имеют возвращаемое значение, а процедуры только выполняю задачи.

Хранимые процедуры представляют собой набор команд, состоящий из одного или нескольких операторов SQL или функций и сохраняемый в базе данных в откомпилированном виде. Процедуры и функции - это объекты базы данных и, следовательно, они создаются командой CREATE и уничтожаются командой DROP. При создании процедуры и функции должны быть определены: имя объекта, перечень и типы параметров и логика работы процедуры или функции, описанные на языке PL/SQL


Чтобы создать процедуру или функцию, необходимо иметь системные привилегии CREATE PROCEDURE. Для создания процедуры, функции или пакета в схеме, отличной от текущей схемы пользователя, требуются системные привилегии на CREATE ANY PROCEDURE. После определения имени новой процедуры или функции необходимо задать ее имена, типы и виды параметров. Для каждого параметра должен быть указан один из видов параметра - IN, OUT или IN OUT. Вид параметра IN предполагает, что значение параметра должно быть определено при обращении к процедуре и не изменяется процедурой.

Хранимая процедура - это наиболее часто используемая в базах данных программная структура, представляющая собой оформленный особым образом сценарий (вернее, пакет), который хранится в базе данных, а не в отдельном файле. Хранимые процедуры отличаются от сценариев тем, что в них допускается использование входных и выходных параметров, а также возвращаемых значений, которые фактически не могут использоваться в обычном сценарии.

Хранимая процедура представляет собой просто имя, связанное с программным кодом SQL, который хранится и исполняется на сервере. Она может содержать практически любые конструкции или команды, исполнение которых поддерживается в СУБД. Процедуры можно использовать для изменения данных, возврата скалярных значений или целых результирующих наборов. Хранимые процедуры, являются основным интерфейсом, который должен использоваться приложениями для обращения к любым данным в базах данных. Хранимые процедуры позволяют не только управлять доступом к базе данных, но также изолировать код базы данных для упрощения обслуживания.

Как серверные программы хранимые процедуры имеют ряд преимуществ.

- Хранимые процедуры хранятся в компилированном виде, поэтому выполняются быстрее, чем пакеты или запросы.
- Выполнение обработки данных на сервере, а не на рабочей станции, значительно снижает нагрузку на локальную сеть.
- Хранимые процедуры имеют модульный вид, поэтому их легко внедрять и изменять. Если клиентское приложение вызывает хранимую процедуру для выполнения некоторой операции, то модификация процедуры в одном месте влияет на ее выполнение у всех пользователей.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

– Хранимые процедуры можно рассматривать как важный компонент системы безопасности базы данных. Если все клиенты осуществляют доступ к данным с помощью хранимых процедур, то прямой доступ к таблицам может быть запрещен, и все действия пользователей будут находиться под контролем. Что еще важнее, хранимые процедуры скрывают от пользователя структуру базы данных и разрешают ему выполнение только тех операций, которые запрограммированы в хранимой процедуре.

Порядок выполнения работы

1. Создадим процедуру без параметров:
2. Создадим процедуру с входным параметром
3. Создадим процедуру с входными параметрами
4. Создадим процедуру с входными параметрами и выходным параметром:
5. Создадим процедуру с входными параметрами и RETURN:

Задание

Написать функции возвращающие значения определённые в пункте 5 лабораторной работы №3.

Тема 7. Особенности языковых средств

Лабораторная работа № 5 (4 часа). «Резервное копирование базы данных».

Цель работы: Ознакомление с возможностями по созданию резервных копий баз данных различных типов.

Задание

- Создать резервную копию базы данных.
- Удалить базу данных.
- Восстановить базу данных из резервной копии.
- Создать инкрементальный бекап базы со следующими параметрами:
 - Дифференциальный - в понедельник в 8:00.
 - Инкрементальный - каждые 4 часа
 - Журнал транзакций - каждые 10 минут.
- Остановить базу данных.
- Восстановить базу данных на 15:30 вторника.

Дополнительная информация:

1. [Восстановление баз данных из резервной копии на Powershell](#)
2. [Создание резервной копии базы данных на Powershell](#)

Модель восстановления базы данных MS SQL

В MS SQL базе данных есть несколько моделей восстановления баз данных.

Модели восстановления баз данных


Просмотр действующей модели восстановления данных

```
SELECT name, recovery_model_desc
FROM sys.databases
WHERE name = 'V14' ;
```

Для включения системы полного восстановления базы данных следует указать это в настройках БД.

Ход работы

1. Создадим резервную копию базы данных.
2. Резервную копию базы данных также можно сделать из консоли, например на PowerShell

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

3. Производим восстановление базы данных из резервной копии
4. Восстановление можно провести из Powershell

```
Restore-SqlDatabase -ServerInstance "MSSQL\SQLEXPRESS" -Database "V14" -
BackupFile "C:\Backup\V14.bak"
```

4. Создание инкрементальных резервных копий

Внимание! В редакции MSSQL Server Express служба MSSQL Agent не работает.

Для выполнения задания воспользуемся "Планировщик заданий".

1. Напишем скрипт для выполнения полного резервного копирования

```
Backup-SqlDatabase -ServerInstance "MSSQL\SQLEXPRESS" -Database "V14" -
BackupFile "c:\Backup\V14.bak"
```

2. Сохраняем скрипт в файл с расширением ps1
3. Напишем скрипт для выполнения инкрементального резервного копирования

```
Backup-SqlDatabase -ServerInstance "MSSQL\SQLEXPRESS" -Database "V14" -
BackupFile "c:\Backup\V14_inc.bak" -Incremental
```

4. Сохраняем скрипт в файл с расширением ps1
5. Запускаем Планировщик заданий
6. Создаём простую задачу
7. Выбираем расписание запуска задачи.
8. Все дополнительные условия запуска задания оставляем по-умолчанию. Их можно изменить в дальнейшем.
9. Устанавливаем задачу: Запустить программу
10. Указываем путь до Powershell, в аргументах указываем путь до скрипта, выполняющего резервную копию данных.
11. Сохраняем задачу.
12. Задача появилась в списке задач. Её можно отредактировать и указать точное расписание запуска программы, а также учётную запись пользователя от имени которого будет происходить вход в MSSQL сервер.
13. Создаёте вторую задачу для выполнения резервного копирования журнала транзакций.
14. Запустите созданные задачи.
15. Проверьте правильность создания резервных копий.

Лабораторная работа № 6 (4 часа). «Моделирование нагрузки и оценка производительности реляционных баз данных».

Цель задания: разработка приложения с максимальным быстродействием, которое принимает HTTP-запросы от клиентов, выполняет действие, предусмотренное вариантом задания и записывает результаты в БД.

Спецификация задания

- Язык разработки приложения – любой язык высокого уровня
- СУБД – MySQL/PostgreSQL и MS SQL
- Клиент – внешняя программа, генерирующая поток сообщений фиксированного содержания в формате HTTP-пакета (предлагается использовать Jmeter, <http://jmeter.apache.org/>)
- Протокол обмена сообщениями – HTTP

Требования к клиентской части

1. Клиентский модуль должен генерировать поток HTTP-запросов с максимально

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

возможной скоростью через запрос типа GET, например: http://localhost/?cl_cod=m1

2. Формат сообщения: **Код клиента** - символьная константа, по которой можно однозначно идентифицировать клиента, приславшего сообщение. Каждый клиент имеет свой уникальный код. Дополнительные параметры запроса, которые зависят от конкретного варианта задания (см. Таблицу 1).

3. Клиентская часть должна управляться следующими параметрами:

1. Имя модуля – идентификатор, уникально идентифицирующая клиента (код клиента). Например, m1.

2. Время работы модуля в секундах – параметр, определяющий время, в течении которого модуль должен генерировать пакеты.

3. Интервал задержки – параметр, определяющий время задержки между отправками пакетов в микросекундах. Если этот параметр не указан, то модуль генерирует пакеты с максимально возможной интенсивностью без задержек.

4. В конце работы, модуль выдает информацию о том, сколько пакетов было отправлено на сервер.

Внимание! Все эти перечисленные требования позволяет удовлетворить Jmeter (см. Выше), поэтому реализовывать клиентскую часть не обязательно.

Требования к серверной части

Серверная часть представляет собой многопоточное приложение, которое позволяет принимать запросы от клиента, выполнять вычисление согласно варианту задания и помещать результаты работы в БД.

Алгоритм работы серверного приложения

- Принять HTTP-запрос
- Разобрать параметры GET запроса, выделив из него код клиента и номер пакета
- Определить IP-адрес источника сообщения
- Сформировать дату и время прихода запроса, как текущее время на стороне сервера

- Выполнить действие, предусмотренное вариантом задания (см. Таблицу 1)

- Записать в таблицу БД полученную от клиента информацию + результаты вычислений + IP адрес источника + дату и время получения запроса. (Структуру данных для хранения результатом разработать самостоятельно).

Требования к проведению эксперимента

- Эксперимент следует проводить таким образом, чтобы было запущено несколько клиентов с различных хостов.

- Цель эксперимента – «выжать» максимальное быстродействие из серверного приложения Ориентировочные показатели: Минимальное быстродействие – 1000 запросов/сек

Предоставление результата


По окончании работы предоставить рабочую программу и отчет о результатах проведения эксперимента с указанием следующих параметров:

- описание схемы проведения эксперимента
- указание количества хостов клиентов, с которых осуществлялась отправка пакетов
- максимально полученное быстродействие (запросы/сек)
- процент потерь пакетов (если они произошли в результате эксперимента)
- указать причину, по которой дальнейшее увеличение быстродействия невозможно (найти узкое место).
- Описание выполняемого действия

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

Параметры запроса

1. Вычисление определенного интеграла $f(x) = x \cdot \sin(x)$ методом прямоугольников А и В — пределы интегрирования, h - шаг
2. Вычисление определенного интеграла $f(x) = x \cdot \cos(x)$ методом прямоугольников А и В — пределы интегрирования, h - шаг
3. Вычисление определенного интеграла $f(x) = x \cdot e^{-x}$ методом прямоугольников А и В — пределы интегрирования, h - шаг
4. Вычисление определенного интеграла $f(x) = \sin^2(x)$ методом прямоугольников А и В — пределы интегрирования, h - шаг
5. Вычисление определенного интеграла $f(x) = \cos^2(x)$ методом прямоугольников А и В — пределы интегрирования, h - шаг
6. Вычисление определенного интеграла $f(x) = x \cdot \sin(x^2)$ методом прямоугольников А и В — пределы интегрирования, h - шаг
7. Вычисление определенного интеграла $f(x) = x \cdot \cos(x^2)$ методом прямоугольников А и В — пределы интегрирования, h - шаг
8. Вычисление определенного интеграла $f(x) = x^2(e^{-2x})$ методом прямоугольников А и В — пределы интегрирования, h - шаг
9. Вычисление определенного интеграла $f(x) = x \cdot \sin(x)$ методом трапеций А и В — пределы интегрирования, h - шаг
10. Вычисление определенного интеграла $f(x) = x \cdot \cos(x)$ методом трапеций А и В — пределы интегрирования, h - шаг
11. Вычисление определенного интеграла $f(x) = x \cdot e^{-x}$ методом трапеций А и В — пределы интегрирования, h - шаг
12. Вычисление определенного интеграла $f(x) = \sin^2(x)$ методом трапеций А и В — пределы интегрирования, h - шаг
13. Вычисление определенного интеграла $f(x) = \cos^2(x)$ методом трапеций А и В — пределы интегрирования, h - шаг
14. Вычисление определенного интеграла $f(x) = x \cdot \sin(x^2)$ методом трапеций А и В — пределы интегрирования, h - шаг
15. Вычисление определенного интеграла $f(x) = x \cdot \cos(x^2)$ методом трапеций А и В — пределы интегрирования, h - шаг
16. Вычисление определенного интеграла $f(x) = x^2(e^{-2x})$ методом трапеций А и В — пределы интегрирования, h - шаг
17. Вычисление определенного интеграла $f(x) = x \cdot \sin(x)$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
18. Вычисление определенного интеграла $f(x) = x \cdot \cos(x)$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
19. Вычисление определенного интеграла $f(x) = x \cdot e^{-x}$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
20. Вычисление определенного интеграла $f(x) = \sin^2(x)$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
21. Вычисление определенного интеграла $f(x) = \cos^2(x)$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
22. Вычисление определенного интеграла $f(x) = x \cdot \sin(x^2)$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
23. Вычисление определенного интеграла $f(x) = x \cdot \cos(x^2)$ методом парабол (метод Симпсона) А и В — пределы интегрирования, h - шаг
24. Вычисление определенного интеграла $f(x) = x^2(e^{-2x})$ методом парабол

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

(метод Симпсона) А и В — пределы интегрирования, h - шаг

Раздел 3. Доступ к БД

Тема 10. Механизм ролей

Лабораторная работа № 7 (4 часа). «Работа с нереляционными базами данных (noSQL)».

Цель задания: научиться использовать noSQL БД для хранения и обработки информации.

Задание: Модифицировать л/р №6 таким образом чтобы в качестве СУБД использовалась noSQL технология. Провести аналогичные эксперименты и поместить результаты экспериментов в отчет.

Спецификация задания: Язык разработки приложения – любой язык высокого уровня СУБД – noSQL (Redis, Hadoop, MongoDB и т.п.) Клиент – внешняя программа, генерирующая поток сообщений фиксированного содержания в формате HTTP-пакета (предлагается использовать Jmeter, <http://jmeter.apache.org/>) Протокол обмена сообщениями – HTTP Требования к серверной части:

Модифицировать серверную часть таким образом чтобы велась обработка запросов.

Алгоритм работы серверного приложения 1.

Принять HTTP-запрос 2. Разобрать параметры GET запроса, выделив из него код клиента и номер пакета 3. Определить IP-адрес источника сообщения 4. Сформировать дату и время прихода запроса, как текущее время на стороне сервера 5. Выполнить действие, предусмотренное вариантом задания (см. Таблицу 1) 6. Записать в таблицу БД полученную от клиента информацию + результаты вычислений + IP адрес источника + дату и время получения запроса. (Структуру данных для хранения результатов разработать самостоятельно).

Тема 12. Контроль целостности БД

Лабораторная работа № 8 (4 часа). «Триггеры».

Теоретическая часть. Триггер, подобно хранимой процедуре, представляет собой сохраненный на сервере набор инструкций SQL. Главное отличие заключается в том, что его невозможно выполнить вручную с помощью команды EXEC. Триггер вызывается на выполнение не пользователем, а прикрепляется к определенной таблице и инициируется самим сервером баз данных как отклик на события вставки, обновления и удаления данных из этой таблицы, т.е. триггер выполняется автоматически как часть самого оператора модификации данных. Триггер на вставку запускается, когда в таблицу вставляется новая запись. Триггер на удаление запускается, когда из таблицы удаляется некоторая запись. Триггер на обновление запускается, когда некоторая запись таблицы изменяется. Кроме того, можно определить триггер, реагирующий сразу на несколько разных типов операций модификации, например на обновление и вставку. Триггер выполняется внутри того же пространства транзакции, что и оператор модификации данных, поэтому откат транзакции в триггере отменяет и саму исходную операцию модификации данных.

Триггеры – наиболее мощный инструмент поддержания целостности базы данных, поскольку они могут выполнять любые необходимые для поддержания целостности данных действия:

- Сравнивать предшествующие и новые версии данных. В большинстве случаев в момент выполнения триггера необходимо знать, какие изменения были выполнены исходным оператором модификации данных. Сведения об этом можно найти в

таблицах inserted (вставленные) и deleted (удаленные), которые становятся доступны внутри триггера. Эти таблицы – фактически представления строк в файле регистрации транзакции, которые были изменены оператором и имеют структуры и имена столбцов, идентичные таблице, которая изменилась. Таким образом, оценить, какие именно изменения были произведены в таблице, можно исследуя содержимое таблиц inserted и deleted, как показано в следующей таблице, и соответственно предпринимать те или иные действия в зависимости от обнаруженных различий.

Оператор	Содержимое таблицы inserted	Содержимое таблицы deleted
INSERT	Добавленные строки	Пусто
UPDATE	Новые строки	Старые строки
DELETE	Пусто	Удалённые строки

* Осуществлять отмену недопустимых модификаций посредством отката транзакции. Это возможно благодаря тому, что триггер запускается в рамках транзакции исходной операции модификации данных.

- Осуществлять считывание из других таблиц.
- Изменять другие таблицы.
- Выполнять хранимые процедуры и функции.

Как правило триггеры используют для реализации сложных ограничений целостности, которые не могут быть обработаны посредством типов данных, обычных ограничений, декларативной ссылочной целостностью, например, реализация сложных ограничений столбцов или генерирование сложных значений по умолчанию, основанных на данных в других строках или таблицах.

Потребность в триггерах возникает тогда, когда требуется автоматическая реакция базы данных на определенные действия пользователя. К примеру, когда из рабочей таблицы удаляется запись, вполне возможно, что необходимо сохранить ее в некоторой дополнительной архивной таблице для последующего аудита. Это можно осуществить, создав триггер на удаление в первой таблице. В момент удаления записи этот триггер будет вызван на выполнение. В это время он еще будет иметь доступ ко всем удаляемым данным и сможет их скопировать в какое-либо другое место.

Триггеры также используют в более сложной и гибкой форме ограничений. Ограничения лимитированы только пределами одной таблицы, в то время как триггеры потенциально имеют доступ ко всей базе данных. Предположим, что необходимо разрешить принимать заказы только от тех клиентов, которые не имеют задолженностей по уже имеющимся счетам. В этом случае можно создать триггер на вставку, который проверяет данное правило и в случае необходимости отменяет исходную операцию вставки (откатывает транзакцию) с выбросом соответствующего сообщения об ошибке.

Общий синтаксис запроса на создания триггера в T-SQL выглядит следующим образом:

```
CREATE TRIGGER имя_триггера ON имя_таблицы AFTER { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] } AS { инструкции T-SQL }
```

При создании триггера указывается его имя, имя таблицы, для которой он определяется, и события, при возникновении которых он должен выполняться (перечисление после ключевого слова AFTER). Эти события соответствуют трем операциям модификации данных: вставка, обновление и удаление. При этом триггер запускается один раз на каждую операцию модификации данных вне зависимости от

количества затронутых этой операцией записей.

Задание

Внимание! Все триггеры должны выполнять осмысленные операции с вашей базой данных

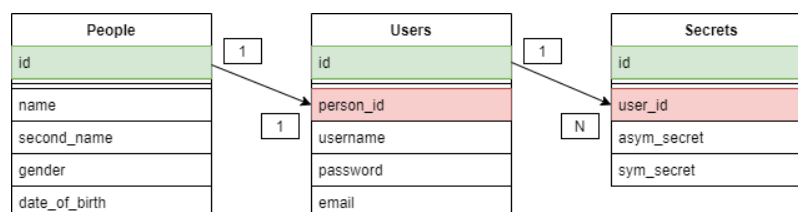
1. Создать триггер на insert
2. Создать триггер на update
3. Создать триггер на delete

Раздел 4. Администрирование СУБД

Тема 13. Взаимодействие СУБД с базовой ОС

Лабораторная работа № 9 (4 часа). «Создание базы данных».

Если вы еще не создавали на сервере базу данных - можно воспользоваться скриптом для быстрого создания базы с таблицами.



Пример создания 3-х таблиц: People, Users, Secrets.

Лабораторная работа № 10 (4 часа). «Организация доступа к СУБД».

Установить возможность проведения аутентификации в СУБД средствами Операционной системы и СУБД. Создать пользователя user_os который использует авторизацию ОС и предоставить ему права для чтения 3-х таблиц из базы данных и запрет на любые действия в других таблицах. Продемонстрировать возможность получения доступа к данным пользователем и отказ доступа. Создать пользователя user_db и предоставить ему доступ к другим 3-м таблицам базы данных и разрешить добавление записей в таблицу с персональными данными, но запретить изменение и чтение данных из этой таблицы.

1. Создать новую роль в СУБД, создать новых пользователей user_role_os и user_role_db и сопоставить их с новой ролью сервера. Назначить права для роли - возможность создания резервных копий баз данных. Произвести резервное копирование базы данных и восстановить базу данных из резервной копии.

2. Проверяем, что в вашей базе данных созданы как минимум 4 таблицы, одна из которых содержит персональные данные. См. Лабораторная работа №1.

3. В свойствах SQL сервера, на вкладке безопасность проверяем, что включена возможность аутентификации средствами ОС, а также средствами SQL Server.

Внимание! Выполняя лабораторную работу в компьютерном классе вы не можете изменить данную настройку сервера.

4. Предоставление прав пользователем.

login: mssql_user_ad@lab24b.ulsu.local

pass: Gfhjkm24, (Пароль24б - если смотреть на русские буквы на клавиатуре)

Запускаем MMS из под учётной записи mssql_user_ad.

runas /user:mssql_user_ad@lab24b.ulsu.local cmd

Нажимаем на свойства ярлыка MMS и копируем его путь.

Вставляем в терминал, запущенный от имени пользователя mssql_user_a

5. Предоставление доступа пользователю к вашей базе данных.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

Под своей учётной записью, в своей базе данных переходим: Безопасность->Пользователи. В контекстном меню выбираем «Создать пользователя».

Изменяем тип пользователя на Windows и вводим полное имя пользователя или производим его поиск.

6. Предоставление доступа пользователю к Объектам базы данных.

Под своей учётной записью, в своей базе данных переходим: Безопасность->Пользователи. В контекстном меню выбираем пользователя для которого мы хотим изменить права доступа к объектам базы данных.

В контекстном меню нажимаем «Свойства» и переходим в вкладку «Защищаемые объекты». Нажимаем «Найти» и выбираем «Все типы объектов». В окне выбираем нужные вам объекты базы данных.

Устанавливаем нужный уровень доступа к объекту.

Обратите внимание, что доступ к Таблице можно предоставить для конкретных столбцов.

7. Проверяем доступ к объекту из-под учётной записи пользователя.

Обратите внимание, что доступ к Таблице можно предоставить для конкретных столбцов.

8. Проверяем отсутствие доступа к другим таблицам базы данных.

9. Создать пользователя с авторизацией через базу данных MSSQL.

10. Для входа из-под учётной записи данного пользователя открываем еще одну MMS и выбираем проверку подлинности SQL сервера.

11. Под своей учётной записью добавляем пользователя к базе данных. И устанавливаем нужные права.

12. Проверяем права доступа из-под учётной записи user_db.

13. Создаём новую роль в базе данных.

14. Добавляем право на создание резервной копии базы данных.

15. Добавляем роль к пользователю, например user_db.

16. Создаём резервную копию базы данных.

Внимание! Сохранение резервной копии на сервере нужно производить только в каталог C:\Backup\

17. Резервные копии базы данных можно скачать с сервера базы данных по адресу: \\mssql\Backup\

18. Произвести восстановление базы данных из резервной копии.

Тема 15. Средства создания резервных копий и восстановление баз данных Лабораторная работа № 11 (8 часов). «Создание хранилища информации о логинах и паролях пользователей в БД».

1. Хранение паролей должно быть в защищенном виде.

2. Доступ данным пользователя может быть предоставлено одновременно только для одного пользователя.

3. Вся информация о получении доступа к данным пользователя должна записываться в соответствующий журнал доступа. **Ход работы.**

Ход работы

Microsoft SQL:

1. Добавляем поле login и pass в любую таблицу, содержащую данные о пользователях.

2. Для защиты хранения паролей в базе данных можно использовать [Хеш функции](#)

3. Для языка T-SQL можно использовать следующий код

```
DECLARE                                @HashThis                                nvarchar(32);
```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

```
SET @HashThis = CONVERT(nvarchar(32), 'Пароль');
SELECT HASHBYTES('SHA2_256', @HashThis);
```

4. Разработаем триггер для таблицы базы данных, который будет хешировать любой передаваемый пароль.

5. Проверяем работу триггера.

```
UPDATE [dbo].[Студенты]
SET [pass] = 'MyNewPassword1'
WHERE [ид] = 2
```

```
SELECT [pass]
FROM dbo.Студенты
WHERE [ид]=2
```

PostgreSQL

1. Загрузите модуль PostgreSQL для работы с криптографическими функциями.

```
CREATE extension IF NOT EXISTS pgcrypto;
```

2. Создание триггера для хеширования пароля.

```
CREATE FUNCTION hash_password() RETURNS TRIGGER AS $func$
BEGIN
IF tg_op = 'INSERT' OR tg_op = 'UPDATE' THEN
NEW.passwd = digest(NEW.passwd, 'sha256');
RETURN NEW;
END IF;
END;
```

```
$func$ LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER users_hash_update
```

```
BEFORE INSERT OR UPDATE ON lab.users
```

```
FOR each ROW EXECUTE PROCEDURE hash_password();
```

Результат:

3. Добавление нового человека в БД ('Alex Ivanov'):

```
INSERT INTO lab.people (name, second_name, gender, date_of_birth)
```

```
VALUES (
```

```
'Alex',
```

```
'Ivanov',
```

```
'M',
```

```
-- 11 - месяц, 05 - день
```

```
'11.05.1989'::DATE
```

```
);
```

Результат:

4. Проверка работы триггера. Создание нового пользователя для 'Alex Ivanov'

Лабораторная работа № 12 (8 часов). «Шифрование данных в БД».

Цель работы

Цель работы: Ознакомление с базовыми механизмами шифрования данных в СУБД

Задание

Для Microsoft SQL Server

– Создать три симметричных открытых ключа в БД.

– Зашифровать три разных столбца таблицы разными симметричными ключами.

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

- Создать MASTER KEY для базы данных.
- Создать сертификат.
- Зашифровать Базу Данных используя AES.

Для PostgreSQL

Написать триггер, реализующий следующие действия.

3. При попытке чтения данных из столбца, произвести его расшифровку на алгоритме AES.

4. При записи данных зашифровать данные столбца на алгоритме AES.

5. Продемонстрировать работу триггера.

Ссылки на описание и примеры

- <https://www.osp.ru/winitpro/2013/05/13035359>
- <https://www.sql.ru/blogs/dbasimple/2007>
- <https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-symmetric-key-transact-sql?view=sql-server-ver15>
- <https://docs.microsoft.com/ru-ru/sql/relational-databases/security/encryption/encryption-hierarchy?view=sql-server-ver15>

Ход работы

Microsoft SQL Server

PostgreSQL

1. Создание пары ключей в терминале:

```
gpg --gen-key
```

2. Экспортировать публичный ключ в файл:

```
gpg --output public.pgp --armor --export dbsec@email.com
```

3. Экспортировать секретный ключ в файл:

```
gpg --output private.pgp --armor --export-secret-key dbsec@email.com
```

4. Создать файл с паролем:

```
echo "Pa$$w0rd" >> password.txt
```

5. Под пользователем «postgres» перейти в директорию конфигурации postgresql (/var/lib/postgresql/12/main):

```
cd /var/lib/postgresql/12/main
```

Создать директорию для сертификатов:

```
mkdir encryption
```

6. Переместите файлы сертификатов(public.pgp, private.pgp, password.txt) в эту директорию:

```
sudo mv public.pgp /var/lib/postgresql/12/main/encryption/
sudo mv private.pgp /var/lib/postgresql/12/main/encryption/
sudo mv password.txt /var/lib/postgresql/12/main/encryption/
```

7. Дайте право на доступ к этим файлам пользователю (postgres): От пользователя root:

```
sudo chown postgres:postgres /var/lib/postgresql/12/main/encryption/*
```

8. Теперь можно читать файлы при помощи запросов к postgresql: Открытый ключ:

```
SELECT pg_read_file('encryption/public.pgp')::text dearmor(
);
```

Закрытый ключ:

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

```
SELECT dearmor(
pg_read_file('encryption/private.pgp')::text
);
```

Файл с паролем от закрытого ключа: (В данном случае длина пароля («Pa\$\$w0rd») - 8 символов)

```
SELECT pg_read_file('encryption/password.txt')::VARCHAR(8);
```

Пример асимметричного шифрования:

```
SELECT pgp_pub_encrypt(
'Hello',
dearmor(
pg_read_file('encryption/public.pgp')::text
)
);
```

Пример симметричного шифрования:

```
SELECT pgp_sym_encrypt(
'Hello',
pg_read_file('encryption/password.txt')::VARCHAR(8)
);
```

9. Триггер для шифрования:

10. Проверка работы триггера: (Создание секрета для пользователя 'AlexIvanov1989')

11. Расшифровка значений: (В Postgresql нельзя создавать триггеры для операции SELECT, но можно создать репрезентацию (VIEW))

Теперь получить расшифрованные значения можно командой

```
SELECT * FROM secrets_decrypted;
```

Лабораторная работа № 13 (8 часов). «Настройка брандмауэра для SQL сервера».

Внимание. Для выполнения лабораторной работы требуется установка SQL сервера на Виртуальную машину.

Цель работы: научиться конфигурировать настройку безопасности сети для SQL сервера

Ход работы

Настройка сетевого доступа к MS SQL Server

1. Настройки работы SQL сервера производятся через оснастку (mmc) SQL Server Configuration Manager

2. Включим сетевые протоколы доступа к SQL серверу.

3. Установим соответствующее значение

4. Настроим доступ к серверу по протоколам IPv4 и IPv6. Порт по умолчанию 1433. IP адрес сервера у каждого компьютера свой

5. Перезапустим службу SQL сервера.


6. Настройки безопасности Microsoft Firewall можно произвести через соответствующую оснастку: wf.msc

7. Необходимо добавить два правила доступа к SQL Server по TCP\1433 и по UDP\1434. Это можно сделать в графическом режиме или в командном

8. Проверить созданные правила

9. Проверить доступ через Microsoft Management Console.

Внимание! Если Вы установили MS SQL сервер в неименованный экземпляр (unnamed instance), то при подключении по сети следует использовать только имя сервера

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

без имени экземпляра.

Тема 18. Средства автоматизации проектирования СУБД
Лабораторная работа № 14 (4 часа). «Аудит информационной безопасности СУБД».

Внимание. Для выполнения лабораторной работы требуется установка SQL сервера на Виртуальную машину.

Подготовка VM

- Сделайте snapshot виртуальной машины. [Снимок виртуальной машины](#)
- Скачать дистрибутив SQL сервера на виртуальную машину.
\\nas\Distr\SQL2019-Express
- Установить Microsoft SQL сервер. [Установка Microsoft SQL Сервера](#)

Цель работы

Цель работы: Аудит информационной безопасности событий в СУБД.

Задание:

Настроить работу системы аудита в SQL сервере для регистрации событий входа в систему и отказа входа в систему, а также регистрацию новых пользователей.

Выполнить вход на SQL сервер с правильной парой логин и пароль. Продемонстрировать запись в журнале аудита. Выполнить вход на SQL сервер с неправильными реквизитами. Продемонстрировать запись в журнале аудита. Создать нового пользователя. Продемонстрировать запись в журнале.

Ход работы

Выполнение данной работы в MS SQL Server и в PostgreSQL сильно отличаются.

В MSSQL есть несколько системных функций позволяющих прочитать журналы аудита системы.

Данные способы подходят для быстрого получения данных о работе СУБД, но они не позволяют настроить журналирование операций под требования пользователя.

Программное обеспечение Microsoft SQL Profiler позволяет просматривать в режиме реального времени все события происходящие в MS SQL сервере и проводить их анализ. Он используется для моделирования различных ситуаций и отладки работы СУБД и её настроек.

Проведение аудита Microsoft SQL Server критично для определения рисков информационной безопасности, а также может быть одним из требований регуляторов, например для PCI DSS или HIPAA.

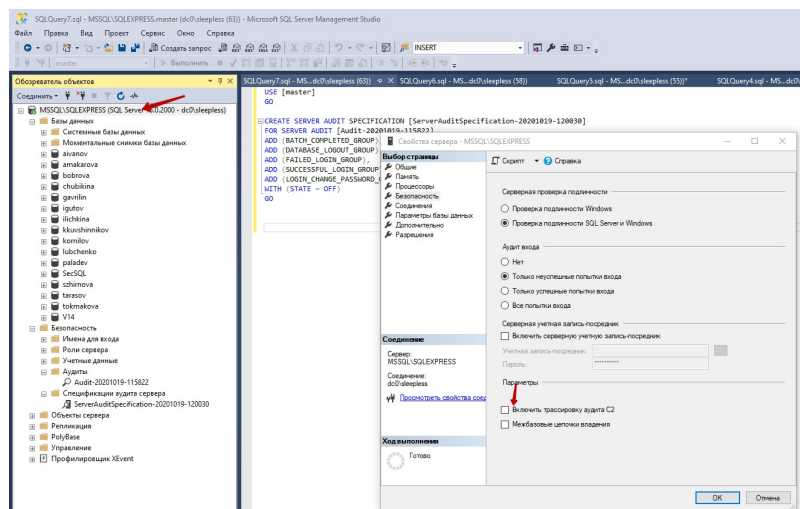
Шаг 1

Определите, какие параметры важны для вашего аудита, например, только вход пользователей в систему, изменение конфигурации сервера, изменение данных и т.д.

Полезные типы аудита:

- C2 Auditing
- Common Compliance Criteria
- Login Auditing
- SQL Server Auditing
- SQL Trace
- Extended Events
- Change Data Capture
- DML, DDL, and Logon Triggers

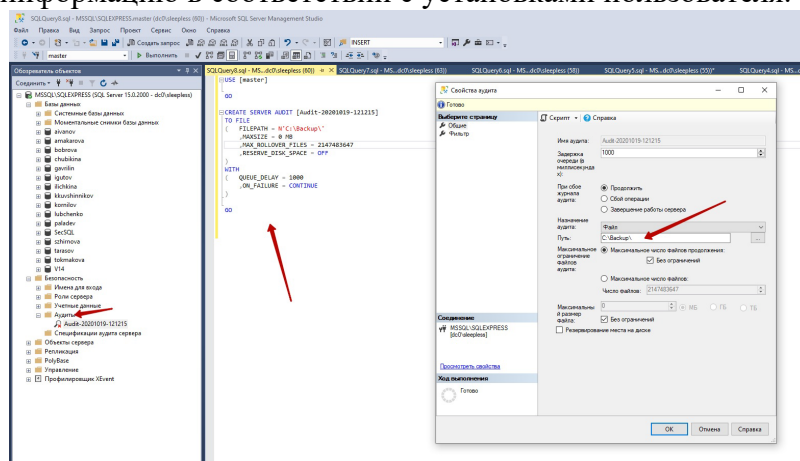
а. **Шаг 2**



Включение отдельных типов специальных аудитов. Для проведения аудитов может потребоваться подключение специальных функций аудита для вашего сервера и для базы данных.

Шаг 3

Создание Аудита. Для создания аудита в MS SQL используется отдельный тип объектов - аудиты. Они работают в фоновом режиме во время работы СУБД и регистрируют информацию в соответствии с установками пользователя.



Внимание! При выполнении лабораторных работ в компьютерном классе следует указывать путь: c:\Vbackup\

Шаг 4

Создание правил для аудита данных.

Если правила аудита затрагивают отдельную базу данных, то они создаются в контексте базы данных. Создадим аудит отслеживания операций чтения из таблицы [Студенты] пользователями с ролью [db_owner]

Включим систему аудитов в SQL Server и включим аудит базы данных. Проверим работу системы, зайдем под пользователем tuser с правами db_owner и выполним выборку из таблицы [Студенты]

Откроем просмотр журналов и убедимся в присутствии записи о выборке данных.

Шаг 5

Если правила аудита затрагивают всю СУБД, то они создаются в контексте СУБД. Создадим аудит отслеживания операций входа в систему.

Включаем. Проверяем.

Шаг 6

- Самостоятельно настраиваем отслеживание пользователей выполняющих резервное копирование базы данных.
- Самостоятельно настраиваем доступ к ключам шифрования данных БД.

PostgreSQL

1. Установка pg-audit

```
sudo apt install postgresql-12-pgaudit
```

2. Изменение файла конфигурации: В файле postgresql.conf необходимо заменить строку:

```
shared_preload_libraries = "
```

на

```
shared_preload_libraries = 'pgaudit'
```

Добавить строки:

```
pgaudit.log_catalog = off
pgaudit.log = 'all, -misc'
pgaudit.log_relation = 'on'
pgaudit.log_parameter = 'on'
```

Раскомментировать и изменить строки на значения:

```
log_destination = 'stderr'
logging_collector = on
log_directory = 'log'
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
```

В директории PostgreSQL «/var/lib/postgresql/12/main» (может отличаться, поэтому при необходимости узнать её можно при помощи SQL запроса - «SHOW data_directory;») создать директорию log с правами доступа для пользователя «postgres»

```
mkdir log
chown postgres:postgres log
```

3. Перезапустить postgresql:

```
sudo systemctl restart postgresql
```

4. Активация расширения


```
CREATE extension IF NOT EXISTS pgaudit;
```

Аудит сессии поддерживает языки манипулирования данными (DML), языки описания данных (DDL), misc sql команды разделённые на классы:

- READ (select, copy from)
- WRITE (insert, update, delete, truncate, copy to)
- FUNCTION (функции и DO блоки)
- ROLE (grant, revoke, create/alter/drop role)
- DDL (все DDL кроме тех, которые находятся в ROLE)
- MISC (discard, fetch, checkpoint, vacuum)

5. Проверка логирования:

```
INSERT INTO lab.people (name, second_name, gender, date_of_birth)
VALUES
('Bob',
'Bush',
'M',
'12.12.1977'::DATE)
```

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

);

Чтение логов:

```
tail -f /var/lib/postgresql/12/main/log/postgresql-2021-01-...-.log | grep AUDIT
```

6. Создание отдельной роли аудита:

```
CREATE ROLE auditor;
ALTER ROLE auditor WITH NOSUPERUSER INHERIT NOCREATEROLE NOCREATEDB
NOLOGIN NOREPLICATION NOBYPASSRLS CONNECTION LIMIT 0;
```

Измените файл postgresql.conf Поменять строку

```
pgaudit.log = none
```

Добавить строку

```
pgaudit.role = auditor
```

Перезапустить postgresql:

```
sudo systemctl restart postgresql
```

Дать разрешения на логирование таблиц пользователем/ролью auditor

```
GRANT ALL ON lab.people TO auditor ;
GRANT ALL ON lab.users TO auditor ;
GRANT ALL ON lab.secrets TO auditor ;
```

Совершите изменения в таблицах в БД (например добавьте нового человека в БД)
(Дополнительно проверьте ведение логов для других операций (пр. SELECT, UPDATE и.т.д.)

```
INSERT INTO lab.people (name, second_name, gender, date_of_birth)
VALUES
('Sara',
'Whatever',
'F',
'11.11.1999'::DATE
);
```

Проверьте ведение логов:

```
tail -f /var/lib/postgresql/12/main/log/postgresql-2021-01-...-.log | grep AUDIT
```

8. ТЕМАТИКА КУРСОВЫХ, КОНТРОЛЬНЫХ РАБОТ, РЕФЕРАТОВ


8.1 Контрольные работы и рефераты не предусмотрены учебным планом дисциплины.

8.2 Курсовые работы не предусмотрены учебным планом дисциплины.

9. ПЕРЕЧНИ ВОПРОСОВ К ЭКЗАМЕНАМ

9. 1. ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ № 1

1. Общие понятия о СУБД
2. Назначение СУБД. История создания
3. Принципы построения СУБД
4. Модели баз данных
5. Распределенные базы данных
6. Назначение, потребность, принципы организации баз данных
7. Примеры использования баз данных
8. Характеристика СУБД, области использования, реализации
9. Возможности СУБД
10. SQL и другие языки для работы с СУБД

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

11. Синтаксис языков для работы с данными
12. Отличия различных языков использующихся для СУБД
13. Понятия пользователя, роли в СУБД
14. Назначение прав для объектов данных в БД
15. Процедура проверки пользователя
16. Установка доступа к объектам в СУБД
17. Политика предоставления прав на объекты
18. Понятие ролей в СУБД
19. Делегирование прав роли
20. Отслеживания действия роли
21. Настройка политики аудита событий в СУБД
22. Аудит доступа к объектам
23. Составление политики аудита на основании требований и модели злоумышленника
24. Понятие разумного аудита в СУБД
25. Описание реляционных баз данных
26. Понятие таблиц и хранение данных

9.2 ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ № 2


1. Понятие безопасности БД. Угрозы безопасности БД: общие и специфичные
2. Требования к безопасности БД
3. Критерии оценки надежных компьютерных систем (TCSEC)
4. Интерпретация TCSEC для надежных СУБД (TDI). Оценка надежности СУБД как компоненты вычислительной системы
5. Понятие политики безопасности. Совместное применение различных политик безопасности в рамках единой модели
6. Транзакции как средство изолированности пользователей. Сериализация транзакций. Методы сериализации транзакций
7. Режимы блокировок. Правила согласования блокировок. Двухфазный протокол синхронизационных блокировок
8. Причины, виды, основные методы нарушения конфиденциальности. Типы утечки конфиденциальной информации из СУБД, частичное разглашение
9. Механизм обнаружения проблем в СУБД
10. Понятие целостности БД
11. Возможности организации бесперебойного функционирования СУБД
12. Настройка СУБД в базовой ОС
13. Выделение ресурсов ОС для нормальной работы СУБД
14. Определение правил функционирования СУБД в ОС
15. Организация ведения журналов событий в СУБД
16. Анализ журналов, выявления проблем и причин их вызвавших
17. Корректная настройка журналов
18. Механизмы резервного копирования и восстановления данных
19. Разработка политики создания резервных копий
20. Определение данных подлежащих резервному копированию
21. Функционирование распределенных БД
22. Проблемы синхронизации распределенных БД
23. Понятие тиражирования, основные принципы и концепции
24. Системы подготовки отчетов в СУБД
25. Функциональность языков СУБД для подготовки отчетов
26. Дополнительные модули и компоненты для работы с отчетами

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		


27. Проектирование СУБД, основные подходы
28. Цели и задачи проектирования СУБД
29. Определение необходимости использования «нормальных форм»
30. Совместное применение средств идентификации и аутентификации, встроенных в СУБД и в ОС

8. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТОВ

Название разделов и тем	Вид самостоятельной работы	Объем в часах	Форма контроля
Раздел 1. Общие понятия о СУБД			
Тема 1. Основные определения и принципы построения СУБД	Подготовка к лекции, семинару, подготовка к сдаче зачёта	2	Тесты перед лекцией, зачёт
Тема 2. Распределенные БД	Подготовка к лекции, семинару, подготовка к сдаче зачёта	2	Тесты перед лекцией, зачёт
Тема 3. Назначение и возможности СУБД	Подготовка к лекции, семинару, подготовка к сдаче зачёта	4	Тесты перед лекцией, зачёт
Раздел 2. Языковые средства			
Тема 4. Языковые средства для различных моделей СУБД	Подготовка к лекции, семинару, подготовка к лабораторным работам, подготовка к сдаче зачёта	4	Тесты перед лекцией, тесты на лабораторных работах, зачёт
Тема 5. Манипулирование в реляционных БД	Подготовка к лекции, семинару, подготовка к сдаче зачёта	2	Тесты перед лекцией, зачёт
Тема 6. Языковые средства описания данных	Подготовка к лекции, семинару, подготовка к лабораторным работам, подготовка к сдаче зачёта	2	Тесты перед лекцией, тесты на лабораторных работах, зачёт
Тема 7. Особенности языковых средств	Подготовка к лекции, семинару, подготовка к лабораторным работам, подготовка к сдаче зачёта	4	Тесты перед лекцией, тесты на лабораторных работах, зачёт
Раздел 3. Доступ к БД			
Тема 8. Идентификация и аутентификация в СУБД	Подготовка к лекции, семинару, подготовка к сдаче зачёта	2	Тесты перед лекцией, зачёт
Тема 9. Разграничение доступа	Подготовка к лекции, семинару, подготовка к сдаче зачёта	2	Тесты перед лекцией, зачёт
Тема 10. Механизм ролей	Подготовка к лекции, подготовка к лабораторным работам, подготовка к сдаче зачёта	6	Тесты перед лекцией, тесты на лабораторных работах, зачёт
Тема 11. Аудит событий	Подготовка к лекции, семинару, подготовка к	2	Тесты перед лекцией, зачёт

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

	сдаче зачёта		
Тема 12. Контроль целостности БД	Подготовка к лекции, семинару, подготовка к сдаче зачёта	4	Тесты перед лекцией, зачёт
Раздел 4. Администрирование СУБД			
Тема 13. Взаимодействие СУБД с базовой ОС	Подготовка к лекции, подготовка к лабораторным работам, подготовка к сдаче экзамена	10	Тесты перед лекцией, тесты на лабораторных работах, экзамен
Тема 14. Журнализация	Подготовка к лекции, семинару, подготовка к сдаче экзамена	6	Тесты перед лекцией, экзамен
Тема 15. Средства создания резервных копий и восстановление баз данных	Подготовка к лекции, подготовка к лабораторным работам, подготовка к сдаче экзамена	18	Тесты перед лекцией, тесты на лабораторных работах, экзамен
Тема 16. Тиражирование и синхронизация в распределенных БД	Подготовка к лекции, семинару, подготовка к сдаче экзамена	6	Тесты перед лекцией, экзамен
Тема 17. Подготовка отчетов в языках СУБД	Подготовка к лекции, семинару, подготовка к сдаче экзамена	6	Тесты перед лекцией, экзамен
Тема 18. Средства автоматизации проектирования СУБД	Подготовка к лекции, подготовка к лабораторным работам, подготовка к сдаче экзамена	8	Тесты перед лекцией, тесты на лабораторных работах, экзамен

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		

б) Программное обеспечение

- операционная среда ОС Windows/ Альт Рабочая станция 8;
- Microsoft Office / МойОфис Стандартный.

в) Профессиональные базы данных, информационно-справочные системы

1. Электронно-библиотечные системы:

1.1. Цифровой образовательный ресурс IPRsmart : электронно-библиотечная система : сайт / ООО Компания «Ай Пи Ар Медиа». - Саратов, [2023]. – URL: <http://www.iprbookshop.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.2. Образовательная платформа ЮРАЙТ : образовательный ресурс, электронная библиотека : сайт / ООО Электронное издательство «ЮРАЙТ». – Москва, [2023]. - URL: <https://urait.ru>. – Режим доступа: для зарегистрир. пользователей. - Текст : электронный.

1.3. База данных «Электронная библиотека технического ВУЗа (ЭБС «Консультант студента») : электронно-библиотечная система : сайт / ООО «Политехресурс». – Москва, [2023]. – URL: <https://www.studentlibrary.ru/cgi-bin/mb4x>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.4. Консультант врача. Электронная медицинская библиотека : база данных : сайт / ООО «Высшая школа организации и управления здравоохранением-Комплексный медицинский консалтинг». – Москва, [2023]. – URL: <https://www.rosmedlib.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.5. Большая медицинская библиотека : электронно-библиотечная система : сайт / ООО «Букап». – Томск, [2023]. – URL: <https://www.books-up.ru/ru/library/> . – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.6. ЭБС Лань : электронно-библиотечная система : сайт / ООО ЭБС «Лань». – Санкт-Петербург, [2023]. – URL: <https://e.lanbook.com>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

1.7. ЭБС Znanium.com : электронно-библиотечная система : сайт / ООО «Знаниум». - Москва, [2023]. - URL: <http://znanium.com> . – Режим доступа : для зарегистрир. пользователей. - Текст : электронный.

2. КонсультантПлюс [Электронный ресурс]: справочная правовая система. / ООО «Консультант Плюс» - Электрон. дан. - Москва : КонсультантПлюс, [2023].


3. Базы данных периодических изданий:

3.1. eLIBRARY.RU: научная электронная библиотека : сайт / ООО «Научная Электронная Библиотека». – Москва, [2023]. – URL: <http://elibrary.ru>. – Режим доступа : для авториз. пользователей. – Текст : электронный

3.2. Электронная библиотека «Издательского дома «Гребенников» (Grebinnikon) : электронная библиотека / ООО ИД «Гребенников». – Москва, [2023]. – URL: <https://id2.action-media.ru/Personal/Products>. – Режим доступа : для авториз. пользователей. – Текст : электронный.

4. Федеральная государственная информационная система «Национальная электронная библиотека» : электронная библиотека : сайт / ФГБУ РГБ. – Москва, [2023]. – URL: <https://нэб.рф>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

5. Российское образование : федеральный портал / учредитель ФГАУ «ФИЦТО». –

Министерство науки и высшего образования РФ Ульяновский государственный университет	Форма	
Ф-Рабочая программа по дисциплине		



URL: <http://www.edu.ru>. – Текст : электронный.

6. Электронная библиотечная система УлГУ : модуль «Электронная библиотека» АБИС Мега-ПРО / ООО «Дата Экспресс». – URL: <http://lib.ulsu.ru/MegaPro/Web>. – Режим доступа : для пользователей научной библиотеки. – Текст : электронный.

Согласовано:

Инженер ведущий / Щуренко Ю.В. /  / 04.05.2023
Должность сотрудника УИТТ ФИО подпись дата

ЛИСТ ИЗМЕНЕНИЙ

№ п/п	Содержание изменения или ссылка на прилагаемый текст изменения	ФИО заведующего кафедрой, реализующей дисциплину/вы- пускающей кафедрой	Подпись	Дата
1.	Утверждение РПД и ФОС для набора 2023 года (10.05.01 и 10.05.03). Актуализация РПД и ФОС для наборов 2022 года 10.05.01 и 10.05.03 (без изменений)	Андреев А.С.		12.04.2023 Протокол заседания кафедры № 12
2.	Утверждение РПД и ФОС для набора 2024 года (10.05.03). Актуализация РПД и ФОС для наборов 2023 года 10.05.01 и 10.05.03 (без изменений)	Андреев А.С.		15.04.2024 Протокол заседания кафедры № 10