ВОРОНЕЖСКИЙ ИНСТИТУТ МВД РОССИИ

Кафедра автоматизированных информационных систем
органов внутренних дел

Системы управления базами данных специального назначения

Методические рекомендации
для слушателей факультета заочного обучения по направлению подготовки 09.03.01 Информатика и вычислительная техника

Воронеж
2024

ББК 32.972.134

Обсуждено и одобрено на заседании кафедры автоматизированных информационных систем, протокол № 6 от 14.02.24 г.

Обсуждено и одобрено на заседании методического совета, протокол № 6 от 19.02.24 г.

Рассмотрено и одобрено к изданию на заседании редакционно-издательского совета, протокол № 2 от 27.02.2024 г.

Рецензенты: начальник информационного центра Главного управления МВД России по Воронежской области полковник внутренней службы Ю.П. Демченко;

старший преподаватель кафедры математики и моделирования систем Воронежского института МВД России, кандидат технических наук, капитан полиции Н.Е. Чиркова

Зверев Г.И. Системы управления базами данных специального назначения: методические рекомендации / Г.И. Зверев. – Воронеж: Воронежский институт МВД России, 2024. – 43 с.

В методических рекомендациях приведены основы функционирования систем управления базами данных специального назначения, возможности их применения в деятельности органов внутренних дел, а также описаны практические работы для выработки умений и навыков для решения профессиональных прикладных задач.

Издание предназначено для слушателей Воронежского института МВД России, обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Цель изучения дисциплины: подготовка обучающихся к оперативно-служебной деятельности и усвоение компетенций, необходимых для профессиональной эксплуатации современных систем управления базами данных (СУБД).

Задачи дисциплины:

– наделить обучающихся знаниями о принципах функционирования СУБД;

– наделить обучающихся знаниями по проектированию баз данных;

– наделить обучающихся знаниями об особенностях СУБД специального назначения, нормативно-правовой базе их функционирования;

– наделить обучающихся знаниями по применению DDL- и DML-подмножеств SQL;

– научить обучающихся создавать объекты СУБД;

– научить обучающихся применять DDL- и DML-подмножества SQL при реализации задач по созданию и сопровождению БД.

– научить обучающихся использовать СУБД специального назначения.

Изучение дисциплины направлено на приобретение обучающимися следующих компетенций:

– способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;

– способен использовать специальную литературу и научно-техническую информацию, отражающую достижения отечественной и зарубежной науки и техники в области информатики и вычислительной техники;

– способен к написанию программного кода с использованием языков и систем программирования, средств манипулирования данными.

Место дисциплины в структуре основной профессиональной образовательной программы.

Дисциплина «Системы управления базами данных специального назначения» относится к блоку дисциплин по выбору образовательной

программы, преподается на четвертом курсе. Изучение дисциплины базируется на знании материала курса «Информатика». Приступая к изучению дисциплины обучаемые должны:

знать:

– основные понятия информатики, основы программирования, основные методы разработки алгоритмов и программ, структуры данных, типовые алгоритмы обработки данных.

уметь:

– применять инструментальные среды программирования.

владеть:

– основными методами работы на компьютере с прикладными программными средствами.

Дисциплина «Системы управления базами данных специального назначения» является предшествующей дисциплиной для последующего изучения:

– Применение и эксплуатация автоматизированных систем;
– Программирование в системах управления базами данных.

1. Основы функционирования СУБД специального назначения

1.1. Основы теории реляционных баз данных

1. Введение в реляционные базы данных

Определение реляционных баз данных:

Реляционные базы данных (RDBMS) – это тип баз данных, ориентированных на организацию данных в таблицы (или реляционные отношения) с использованием строк и столбцов. Каждая строка представляет собой запись, а каждый столбец – атрибут или поле данных. Реляционные базы данных основаны на принципах реляционной алгебры и теории множеств.

Основные понятия и терминология:

Таблицы являются основной структурой данных в реляционных базах данных. Они представляют собой двумерные структуры, где данные организованы в виде строк и столбцов. Каждая строка таблицы представляет собой отдельную запись, а каждый столбец – атрибут или поле данных. Таблицы могут содержать разнообразные типы данных, такие как числа, строки, даты и так далее.

Атрибуты, также известные как поля, представляют собой конкретные характеристики или аспекты данных, хранящихся в таблицах. Например, если у вас есть таблица «Сотрудники», то атрибуты могут включать в себя «Имя», «Фамилию», "Дата рождения", "Заработную плату" и другие характеристики сотрудников.

Запросы представляют собой механизм для взаимодействия с данными в реляционных базах данных. Они используются для извлечения, обновления, вставки и удаления данных из таблиц. Запросы составляются с использованием языка SQL (Structured QueryLanguage), который предоставляет различные операторы и предложения для работы с данными.

Транзакции представляют собой логические операции, выполняемые над данными в базе данных. Транзакции обеспечивают целостность данных, что означает, что они либо выполняются полностью, либо не выполняются вообще. Примерами операций транзакций являются внесение изменений в несколько таблиц сразу или отмена всех изменений, если произошла ошибка.

2. Нормализация баз данных

Нормализация баз данных – это процесс проектирования баз данных с целью уменьшения избыточности данных и обеспечения целостности данных. Это происходит путем организации данных в таблицах так, чтобы каждая таблица имела четко определенный набор атрибутов (полей) и чтобы данные в каждой записи были связаны с одной темой. Основная идея – избежать хранения одной и той же информации в разных местах, что может привести к аномалиям при обновлении данных.

Нормализация проводится в соответствии с нормальными формами, которые определяют, насколько данные разделены и организованы в таблицах. К основным нормальным формам относят:

Первая нормальная форма (1НФ) – все атрибуты должны быть атомарными, то есть они не могут содержать набор значений.

Вторая нормальная форма (2НФ) – данные должны быть в первой нормальной форме, и каждый неключевой атрибут должен полностью зависеть от ключевого атрибута.

Третья нормальная форма (3НФ) – данные должны быть во второй нормальной форме, и неключевые атрибуты не должны зависеть друг от друга.

Продвинутые нормальные формы включают в себя нормальную форму Бойса–Кодда (BCNF) и четвертую нормальную форму (4НФ).

Рассмотрим пример нормализации на основе не нормализованной таблицы. Предположим, у нас есть таблица, представляющая информацию о заказах, и она выглядит следующим образом:

Таблица «Заказы» (не нормализованная):

Номер заказа	Имя клиента	Адрес клиента	Телефон клиента	Название товара	Количество	Цена
1	Анна	Ул. Центральная, 123	555–1234	Ноутбук	2	800
2	Максим	Ул. Парковая, 456	555–5678	Смартфон	3	400
3	Анна	Ул. Центральная, 123	555–1234	Планшет	1	300

Эта таблица не нормализована, и мы видим избыточность информации. Например, информация о клиенте (Имя клиента, Адрес

клиента, Телефон клиента) дублируется для заказов, сделанных одним и тем же клиентом (например, Анной). Это создает потенциал для аномалий при обновлении данных, и это неэффективно.

Чтобы нормализовать эту таблицу, мы можем выполнить следующие шаги:

1. Разделите таблицу «Заказы» на две отдельные таблицы: «Клиенты» и «Заказы». Это уберет избыточность информации о клиентах.

Таблица «Клиенты» (1НФ):

Имя клиента	**Адрес клиента**	**Телефон клиента**
Анна	Ул. Центральная, 123	555–1234
Максим	Ул. Парковая, 456	555–5678

Таблица «Заказы» (1НФ):

Номер заказа	**Имя клиента**	**Название товара**	**Количество**	**Цена**
1	Анна	Ноутбук	2	800
2	Максим	Смартфон	3	400
3	Анна	Планшет	1	300

2. В таблице «Заказы» все неключевые атрибуты должны полностью зависеть от ключевого атрибута, который в данном случае может быть определен как (Номер заказа, Название товара). Создайте таблицу «Товары», чтобы разделить информацию о товарах.

Таблица «Товары» (2НФ):

Название товара	**Цена**
Ноутбук	800
Смартфон	400
Планшет	300

Таблица «Заказы» (2НФ):

Номер заказа	Имя клиента	Название товара	Количество
1	Анна	Ноутбук	2
2	Максим	Смартфон	3
3	Анна	Планшет	1

Теперь данные разделены на три таблицы: «Клиенты», «Товары» и «Заказы», и информация больше не дублируется. Это нормализованная база данных, которая обеспечивает лучшую структуру и уменьшает риск аномалий при изменении данных.

После проведения нормализации базы данных, получается следующее итоговое разбиение на таблицы:

Таблица «Клиенты»:

Имя клиента	Адрес клиента	Телефон клиента
Анна	Ул. Центральная, 123	555–1234
Максим	Ул. Парковая, 456	555–5678

Таблица «Товары»:

Название товара	Цена
Ноутбук	800
Смартфон	400
Планшет	300

Таблица «Заказы»:

Номер заказа	Имя клиента	Название товара	Количество
1	Анна	Ноутбук	2
2	Максим	Смартфон	3
3	Анна	Планшет	1

Преимущества нормализации включают в себя снижение избыточности данных, облегчение обновления и поддержки базы данных, а также уменьшение вероятности аномалий данных.

К недостаткам можно отнести более сложные запросы для объединения данных из разных таблиц и некоторую потерю производительности при выполнении сложных запросов.

3. Отношения между таблицами

В контексте реляционных баз данных ключи играют важную роль. Ключи используются для установления связей между таблицами. Введение ключей обеспечивает уникальность и целостность данных.

Первичный ключ (Primary Key) – это атрибут или набор атрибутов, который однозначно идентифицирует каждую запись в таблице. Он должен быть уникальным и не пустым для каждой записи. Первичные ключи помогают быстро находить и связывать данные в таблицах.

Внешний ключ (Foreign Key) – это атрибут в одной таблице, который связан с первичным ключом в другой таблице. Внешний ключ используется для создания отношений между таблицами. Это позволяет связывать данные из разных таблиц и обеспечивает целостность ссылочных данных.

В реляционных базах данных существует несколько типов отношений между таблицами:

Один–ко–многим (One–to–Many) – это тип отношения, при котором одна запись в первой таблице может иметь много связанных записей во второй таблице. Например, один клиент может иметь много заказов.

Многие–ко–многим (Many–to–Many) – это более сложное отношение, при котором множество записей в первой таблице связано с множеством записей во второй таблице. Обычно используется промежуточная (ассоциативная) таблица для связывания данных. Например, много студентов может учиться во многих курсах, и наоборот.

Один–ко–одному (One–to–One) – это отношение, при котором одна запись в первой таблице связана с одной записью во второй таблице. Обычно это используется для разделения данных, когда определенные атрибуты могут быть вынесены в отдельную таблицу.

Каскадное удаление и обновление – это дополнительные действия, которые можно настроить при использовании внешних ключей. Каскадное удаление означает, что при удалении записи в родительской таблице,

связанные записи в дочерних таблицах также будут автоматически удалены. Каскадное обновление позволяет автоматически обновлять связанные записи в дочерних таблицах при обновлении родительской записи.

4. Транзакции и управление конкурентным доступом

Транзакция в контексте баз данных – это логическая операция или набор операций, которые выполняются над данными в базе данных. Транзакции могут включать операции чтения (SELECT) и операции записи (INSERT, UPDATE, DELETE). Важно понимать, что транзакции должны быть атомарными, целостными, изолированными и долговечными, что образует акроним ACID (Atomicity, Consistency, Isolation, Durability):

Atomicity (Атомарность) – транзакция должна быть выполнена целиком, или не выполнена вовсе. Нет промежуточных состояний.

Consistency (Согласованность) – транзакция должна приводить базу данных из одного согласованного состояния в другое с соблюдением всех ограничений и правил целостности.

Isolation (Изоляция) – данные в процессе выполнения транзакции должны быть изолированы от других транзакций, чтобы предотвратить конфликты и взаимное воздействие.

Durability (Долговечность) – когда транзакция завершается успешно, ее изменения должны сохраняться даже в случае сбоя системы.

5. Оптимизация запросов и индексирование

Оптимизация запросов в реляционных базах данных является процессом повышения производительности и эффективности выполнения SQL–запросов. Оптимизация запросов включает в себя следующие аспекты:

Индексы позволяют ускорить поиск данных в таблицах. Важно выбирать индексы тщательно, исходя из типов запросов, которые будут выполняться. Индексы могут быть созданы на одном или нескольких столбцах таблицы.

Использование правильных SQL–операторов и предложений, таких как JOIN, WHERE, GROUP BY и HAVING, может улучшить производительность запросов.

Подзапросы и объединения позволяют создавать более сложные запросы. Оптимизация таких запросов включает в себя выбор оптимального способа объединения таблиц и фильтрации данных.

Представления могут упростить выполнение запросов, и оптимизация представлений может повысить производительность.

Индексирование – это процесс создания индексов для столбцов таблицы с целью ускорения поиска и фильтрации данных. Индексы позволяют базе данных быстро находить записи, соответствующие определенным критериям.

Существует несколько типов индексов, включая одиночные индексы, составные индексы, уникальные индексы и полнотекстовые индексы. Каждый из них имеет свои преимущества и ограничения.

Важно выбирать столбцы, которые часто используются в запросах, для индексирования. Индексация редко используемых столбцов может привести к избыточной нагрузке на базу данных.

1.2. Создание и сопровождение базы данных средствами СУБД

1. Основные компоненты PostgreSQL

PostgreSQL работает в клиент-серверной архитектуре, где сервер управляет доступом к данным и выполнением SQL-запросов. Сервер PostgreSQL поддерживает множество одновременных клиентских соединений.

pg_catalog – это системный каталог, который содержит метаданные о базе данных, таблицах, индексах, пользователях и других объектах базы данных. Позволяет администраторам и разработчикам получать информацию о структуре базы данных.

Метаданные представляют собой информацию о самой базе данных и ее объектах. Информация о метаданных хранится в таблицах pg_catalog и используется PostgreSQL для управления базой данных и выполнения запросов.

Таблицы – основные объекты хранения данных в PostgreSQL. Они используются для организации структурированных данных.

Индексы – структуры данных, ускоряющие выполнение запросов, позволяя быстро находить и извлекать данные из таблиц.

PostgreSQL разделяет хранилище данных на различные пространства (tablespaces), что позволяет лучше управлять распределением данных и их хранением.

В PostgreSQL существует несколько процессов фонового выполнения, которые выполняют различные задачи, такие как автовакуум, анализ журнала изменений (WAL), сбор статистики и другие. Эти процессы обеспечивают целостность данных, поддерживают производительность и выполняют ряд других важных функций.

PostgreSQL разработана с учетом работы в многопользовательской среде, что позволяет множеству клиентов одновременно подключаться к базе данных. Каждый клиент имеет собственный процесс Backend, который обслуживает его запросы.

PostgreSQL поддерживает пул соединений (connection pool), который позволяет эффективно управлять и переиспользовать соединения с базой данных, уменьшая нагрузку на сервер и улучшая производительность.

Транзакция – это логическая единица работы с базой данных, которая может включать один или несколько SQL-запросов. PostgreSQL

поддерживает транзакции, которые обеспечивают целостность данных и изоляцию между параллельными транзакциями.

PostgreSQL поддерживает разные уровни изоляции транзакций, такие как Read Uncommitted, Read Committed, Repeatable Read и Serializable. Выбор уровня изоляции позволяет балансировать между целостностью данных и производительностью.

PostgreSQL использует механизм блокировок для обеспечения согласованного доступа к данным при одновременных изменениях. Разные виды блокировок, такие как эксклюзивные и разделяемые, позволяют контролировать доступ к данным.

PostgreSQL предоставляет средства для управления конфликтами блокировок, что позволяет предотвращать или разрешать конфликты при работе с данными.

PostgreSQL позволяет определять, насколько долго блокировка будет действительной, что уменьшает возможность блокировки данных на долгий срок.

2. Архитектурные принципы PostgreSQL

PostgreSQL разработана с учетом модульности, что позволяет легко добавлять новую функциональность с помощью расширений (extensions). Расширения позволяют интегрировать сторонние библиотеки и плагины, расширяя возможности СУБД без изменения ее исходного кода.

Разработчики могут создавать собственные функции, типы данных, операторы и агрегаты, что позволяет адаптировать PostgreSQL под конкретные потребности проекта.

Write-Ahead Logging (WAL) – это механизм журналирования, используемый PostgreSQL для обеспечения целостности данных, восстановления после сбоев и поддержки репликации. Принцип работы WAL заключается в том, что перед фактической записью изменений в таблицы, они записываются в журнал.

Перед тем как изменения данных будут внесены в таблицы, они записываются в специальный журнал, называемый журналом изменений (WAL log). Этот журнал сохраняет информацию о внесенных изменениях, таких как вставка, обновление или удаление записей.

Записи в журнале представляют собой точный лог изменений, что позволяет восстанавливать базу данных после сбоев или отказов. В случае сбоя можно использовать WAL для восстановления данных до момента сбоя.

WAL также играет важную роль в репликации данных. Он может быть использован для передачи изменений данных на другие серверы, обеспечивая репликацию между мастером и репликами.

Запись в журнал происходит асинхронно, что позволяет ускорить операции записи данных. Тем не менее, WAL обеспечивает полную целостность данных.

Многоверсионное управление конфликтами (MVCC) – это механизм, который PostgreSQL использует для обеспечения изоляции данных в многопользовательской среде, где множество транзакций одновременно работает с базой данных.

Вместо блокировки данных для других транзакций, PostgreSQL создает снимки данных, которые представляют состояние данных на определенный момент времени. Каждая транзакция видит данные в соответствии с ее собственным снимком.

MVCC обеспечивает уровень изоляции транзакций, который позволяет параллельно выполняющимся транзакциям не видеть изменений, внесенных другими транзакциями, пока они не будут явно подтверждены.

Транзакции, выполняющие только чтение данных, не блокируют другие транзакции и не мешают им одновременно вносить изменения.

MVCC позволяет избегать конфликтов между транзакциями, так как они работают с разными снимками данных.

PostgreSQL позволяет транзакциям иметь разные продолжительности жизни, что обеспечивает гибкость в работе с данными.

PostgreSQL оптимизирована для работы с данными в памяти, что обеспечивает высокую производительность при запросах к часто используемым данным.

Буферный кэш в PostgreSQL позволяет кэшировать данные в оперативной памяти, уменьшая количество обращений к диску и ускоряя выполнение запросов.

PostgreSQL использует статистику и планировщик запросов для выбора наилучшего способа выполнения SQL-запросов.

3. Запросы и оптимизация

Планировщик запросов PostgreSQL – это ключевой компонент, отвечающий за оптимизацию и выполнение SQL-запросов. Его задача – выбрать наилучший план выполнения запроса для оптимальной производительности.

Планировщик запросов проходит через несколько стадий оптимизации, включая синтаксический анализ запроса, построение дерева запроса и выбор физического плана выполнения.

Планировщик использует статистическую информацию о данных, такую как распределение значений в столбцах, для выбора оптимального плана. Эвристики помогают учитывать различные факторы, такие как размер данных, индексы и доступные ресурсы сервера.

Индексы играют ключевую роль в оптимизации производительности запросов в PostgreSQL. Они представляют собой структуры данных, которые позволяют быстро находить и извлекать нужные записи из таблицы.

Индексы обеспечивают быстрый доступ к данным, позволяя избегать полного сканирования таблицы. Вместо того, чтобы просматривать каждую запись, PostgreSQL может использовать индекс для быстрого поиска нужных строк.

Использование подходящих индексов может существенно ускорить выполнение SQL-запросов. Например, индексы могут ускорить SELECT-запросы, а также условия в WHERE-клаузе и соединения таблиц.

Использование индексов снижает количество операций ввода-вывода на диске, так как запросы могут быть выполнены, просматривая индексы, а не саму таблицу.

Индексы позволяют параллельным транзакциям быстро находить данные без блокировки друг друга, что способствует изоляции данных и конфликтов.

PostgreSQL поддерживает разнообразные типы индексов, позволяя выбирать наилучший индекс для конкретных запросов. Некоторые из наиболее распространенных типов индексов в PostgreSQL включают:

B-дерево (B-tree) индексы – подходят для равенственных сравнений и диапазонных запросов.

Хеш (Hash) индексы – используются для равенственных сравнений и подходят для операций с хеш-соединениями.

GiST (Generalized Search Tree) и GIN (Generalized Inverted Index) индексы – подходят для поиска по тексту, географическим данным и другим специализированным типам данных.

SP-GiST индексы – используются для специализированных структур данных, таких как диапазоны IP-адресов.

BRIN (Block Range INdex) индексы – подходят для больших таблиц с упорядоченными данными, такие как временные ряды.

Выбор правильных индексов играет важную роль в оптимизации запросов. PostgreSQL предоставляет оптимизатору много информации о структуре данных и статистике, чтобы помочь принимать решения о выборе индексов. Оптимизатор анализирует эту информацию и решает, какие индексы следует использовать для наилучшей производительности запроса.

Написание эффективных SQL-запросов – важный аспект оптимизации производительности. Проектирование правильных запросов с учетом индексов и структуры таблиц может существенно улучшить производительность. В PostgreSQL существуют инструменты для анализа и профилирования выполнения запросов, такие как EXPLAIN и pg_stat_statements. Они помогают разработчикам и администраторам оптимизировать запросы и выявить узкие места в производительности.

4. Высокодоступность и резервное копирование

Высокодоступность (High Availability, HA) – это характеристика системы, обеспечивающая ее непрерывную работу и доступность для пользователей даже в случае сбоев или отказов в работе отдельных компонентов.

PostgreSQL предоставляет несколько механизмов для обеспечения высокодоступности, включая репликацию, кластеризацию и горизонтальное масштабирование.

Репликация PostgreSQL – это процесс создания копий данных (реплик) с одного сервера (мастера) на другие сервера (реплики). Чтение данных может быть распределено между репликами, что уменьшает нагрузку на мастер.

PostgreSQL поддерживает механизмы отказоустойчивости, включая автоматическое переключение (failover) с мастера на реплику в случае отказа мастера.

Резервное копирование – это важная составляющая высокодоступности и безопасности данных. Оно обеспечивает возможность восстановления данных после случайного удаления, сбоев в работе или катастрофических событий.

PostgreSQL поддерживает разные виды резервного копирования, включая полное (full), дифференциальное (differential) и инкрементное (incremental).

Сам PostgreSQL предоставляет инструменты для резервного копирования, такие как pg_dump и pg_basebackup. Существуют также сторонние инструменты, предоставляющие расширенные функциональные возможности для резервного копирования.

Хранение резервных копий на отдельных удаленных серверах или в облаке может обеспечить защиту данных от различных катастрофических событий, таких как пожары, наводнения и кражи.

Важно обеспечивать шифрование резервных копий для защиты конфиденциальности данных.

Критической частью резервного копирования является разработка и тестирование плана восстановления, который описывает процедуры восстановления данных после сбоев.

Для обеспечения высокодоступности и резервного копирования PostgreSQL рекомендуется использовать системы мониторинга и управления, такие как pgBouncer, Patroni и другие. Эти инструменты позволяют автоматизировать многие процессы, связанные с обслуживанием баз данных, включая мониторинг состояния серверов, управление репликацией и автоматическое восстановление после отказов.

5. Масштабируемость и производительность

Масштабируемость – это способность системы обрабатывать ростущую нагрузку путем увеличения ресурсов, таких как процессоры, память и хранилище.

Горизонтальная масштабируемость (scaling out) – это увеличение производительности путем добавления новых серверов или узлов к системе.

Вертикальная масштабилизация (scaling up) – это увеличение производительности путем улучшения ресурсов на существующих серверах, таких как увеличение количества ядер процессора или объема памяти.

PostgreSQL поддерживает как горизонтальную, так и вертикальную масштабируемость. Горизонтальная масштабируемость может быть достигнута с использованием репликации и разделения данных между серверами.

Производительность – это способность системы выполнять операции и запросы с минимальной задержкой и максимальной эффективностью.

Важным аспектом обеспечения высокой производительности PostgreSQL является оптимизация SQL-запросов. Эффективные запросы,

правильно спроектированные с учетом индексов и структуры данных, могут существенно улучшить производительность.

PostgreSQL предоставляет множество параметров конфигурации, которые позволяют настроить систему под конкретные требования и ресурсы сервера. Настройка буферных кешей, рабочих процессов и других параметров может улучшить производительность.

Эффективное использование индексов может значительно увеличить скорость выполнения запросов. PostgreSQL предоставляет механизмы для анализа и оптимизации индексов.

Использование репликации и параллелизма позволяет распределять нагрузку между серверами и увеличивать производительность при одновременном выполнении запросов.

Для оценки и улучшения производительности PostgreSQL полезно использовать инструменты мониторинга и профилирования, такие как pg_stat_statements и pg_stat_monitor. Они позволяют анализировать выполнение запросов и выявлять узкие места в производительности.

6. Современные тенденции и возможности PostgreSQL. Примеры использования PostgreSQL

PostgreSQL является проектом с открытым исходным кодом (Open Source), что означает, что его исходный код доступен для просмотра, изменений и распространения.

PostgreSQL имеет активное мировое сообщество разработчиков, пользователей и администраторов, которые вносят свой вклад в развитие и поддержку системы.

Благодаря открытости и настраиваемости PostgreSQL, пользователи имеют полный контроль над настройками и функциональностью СУБД, что делает ее универсальным решением для различных задач.

Внедрение поддержки JSON и JSONB (бинарный JSON) позволяет хранить и работать с полуструктурированными данными и расширяет возможности PostgreSQL в области обработки JSON-данных.

Введение логической репликации позволяет более гибко управлять данными и распределять их между серверами, обеспечивая более высокую доступность и масштабируемость.

Появление поддержки SQL/JSON позволяет выполнять запросы к JSON-данным с использованием SQL-запросов, что облегчает работу с JSON-структурами.

Асинхронное выполнение запросов позволяет параллельно выполнять несколько запросов, что повышает производительность PostgreSQL.

Новые инструменты и системные каталоги помогают в мониторинге и профилировании выполнения запросов и работы PostgreSQL.

PostgreSQL поддерживает механизм расширений, который позволяет интегрировать сторонние модули и функциональность в СУБД без изменения ее исходного кода.

Вокруг PostgreSQL сформировалась богатая экосистема инструментов и библиотек, которые облегчают администрирование, мониторинг, разработку и интеграцию с другими системами.

PostgreSQL активно работает над улучшением безопасности, включая внедрение возможности шифрования данных, аутентификации и авторизации. Поддержка шифрования на уровне столбцов и дополнительные средства защиты данных.

PostgreSQL интегрируется с различными облачными платформами, что облегчает развертывание и управление PostgreSQL в облаке.

Облачные окружения позволяют легко настраивать горизонтальное масштабирование и репликацию для обеспечения высокой доступности и производительности.

2. Применение СУБД специального назначения в ОВД

2.1. Организация работы ИСОД МВД России

Цель и задачи ИСОД МВД России

Создаваемая ИСОД МВД России является эволюционным развитием Единой информационно-телекоммуникационной системы ОВД (ЕИТКС ОВД) и представляет собой сегмент глобальной общегосударственной информационно-телекоммуникационной системы (электронного правительства) на более высоком технологическом уровне и направлена на обеспечение реализации в системе МВД России государственной политики Российской Федерации в области информатизации, связи и технической защиты информации. Её реализация призвана максимально оптимизировать служебную деятельность практически каждого сотрудника ОВД, избавить его от «рутины» промежуточных этапов «ручного» сбора, обобщения и анализа необходимой информации, существенно сократить время передачи и доведения актуальной информации до компетентного субъекта принятия управленческого решения, на несколько порядков снизить материально-технические затраты.

Вместе с тем, эта же система минимизирует непосредственное личное взаимодействие сотрудников между собой и представителями иных ветвей государственной власти по служебной компетенции, а также между заявителями, максимально поставив с использование информационных технологий под так называемый «электронный контроль» их профессиональную деятельность. При масштабном развитии Системы наиболее важные служебные правоотношения должностных лиц максимально сведутся к формуле: «Должностное лицо – техническая Система – должностное лицо». В такой Системе принятое решение любого должностного лица в рамках служебной компетенции для придания ему юридической силы, удостоверяется электронной подписью и чётко фиксируется во времени, а также сохраняется весь последующий механизм его исполнения с возможностью контроля соответствующими субъектами.

Концепция построения ИСОД:

– Реализация сервис-ориентированной архитектуры для создания системы автоматизации прикладных задач ИСОД.

– Использование технологии «облачных вычислений» – технологии обеспечения повсеместного и удобного сетевого доступа по требованию к общему пулу конфигурируемых вычислительных ресурсов.

– Построение централизованной схемы инфраструктуры на базе центров обработки данных (ЦОД) единого, масштабируемого, программно-аппаратного решения на всех уровнях структурной организации ОВД.

– Максимальное использование имеющихся в ОВД программно-технических комплексов обеспечения деятельности (ПТК), развертывание на их основе типового, масштабируемого решения для обеспечения деятельности ОВД в составе системы прикладных задач ИСОД.

Цель ИСОД МВД России заключается в повышении уровня информационно-аналитического обеспечения деятельности МВД России, выражающаяся в обеспечении сотрудникам основных направлений деятельности ОВД и в соответствии со служебной компетенцией возможности использования:

– увеличенного количества видов актуальной структурированной информации в электронной форме, одновременно доступной с автоматизированных рабочих мест (АРМ) и обеспечении быстрого поиска необходимой информации,

– обобщённых актуальных данных при принятии управленческих решений.

Задачей ИСОД является автоматизация основных видов деятельности сотрудников подразделений МВД России, которая призвана:

– интегрировать информационные ресурсы МВД России на основе единых информационных банков данных, в том числе обеспечить совместное ведение учёта преступлений, сообщений о происшествиях, специальных учётов;

– создать единую систему централизованной обработки данных для информационно-аналитической поддержки деятельности подразделений МВД России при принятии решений и обеспечить защищённый регламентированный доступ к ней пользователей в режиме реального времени, а также контроль за деятельностью сотрудников на основе анализа произведённых ими действий в процессе эксплуатации;

– предоставить возможность ведомственного электронного обмена информацией и внешнего электронного взаимодействия подразделений МВД России с органами государственной власти, местного самоуправления, а также гражданами и организациями;

– создать подсистему обеспечения информационно-аналитической деятельности оперативно-технических подразделений МВД России;

– обеспечить предоставление государственных услуг в электронном виде и развитие соответствующих информационных систем, интегрируя их с компонентами инфраструктуры электронного правительства.

– совершенствовать профессиональную подготовку и переподготовку в области информационных технологий.

Технологии ИСОД МВД России

Технологическую основу внедряемой ИСОД МВД России составляют «облачные вычисления», сущность которых заключается в том, что при обработке информации компьютерные ресурсы (то есть вычислительные мощности, программное обеспечение, данные) предоставляются пользователю в режиме реального времени, как распределённый в вычислительной сети сервис (так называемый онлайн Net-сервис).

Указанные технологии при обработке данных способны динамически перераспределять в глобальной сети вычислительную нагрузку между структурными сегментами и обеспечивать требуемый уровень защиты, катастрофоустойчивости и доступности информации.

Указанные технологии позволяют пользователю иметь электронный доступ со своего АРМ к необходимым данным, но не могут и не должны заставлять его заботиться о структуре хранения информации, специальном программном обеспечении своего компьютера. Пользователю необходимо только санкционированное подключение АРМ в основном с типовым программным обеспечением и средствами защиты информации в вычислительную сеть, в которой распределены необходимые ему специальные программы и данные для работы в соответствующей его должностного статуса информационной системе – электронном сервисе.

При использовании облачных технологий в деятельности МВД России служебная информация, введённая надлежащим образом в создаваемую Систему, может быть изменена только установленными способами, а сами данные могут храниться и обрабатываться в ином субъекте Российской Федерации на техническом оборудовании, созданных и объединённых в ведомственную катастрофоустойчивуюединую информационную систему центров обработки данных (Далее – ЕИС ЦОД).

Санкционированный электронный доступ любого сотрудника ОВД (с помощью электронной подписи) в указанную систему будет осуществляться с АРМ (в перспективе мобильные терминалы, планшеты),

подключенных по ранее созданным ведомственным каналам связи (ИМТС ОВД) через региональные программно-технического комплекса единого информационного пространства.

Механизм электронного доступа регламентирован процедурами идентификации и аутентификации пользователей (в соответствии с персональной электронной подписью сотрудника), а также контролем за их действиями в соответствии с создаваемыми в настоящее время на каждого пользователя ИСОД МВД России электронными личными кабинетами и единым каталогом.

Электронная подпись сотрудника МВД России являться неотъемлемым условием для доступа к внедряемым электронным сервисам Системы. Она удостоверяет правомочность действий пользователя в соответствии с его занимаемой должностью. Единый каталог пользователей будет интегрирован с системой удостоверяющих центров МВД России, необходимых для функционирования электронной подписи.

Состав ИСОД МВД России

– ИМТС – интегрированная мультисервисная телекоммуникационная сеть
– ЦОД и ПТК
– ПОИБ – подсистема обеспечения информационной безопасности
– Сервисы повседневной деятельности
– Сервисы оперативно-служебной деятельности
– ЕЦЭ – единый центр эксплуатации сервисов ИСОД

Технологическая инфраструктура ЦОД ИСОД МВД России

Подсистема обеспечения информационной безопасности ПОИБ

– Криптографическая защита каналов связи

– Антивирусная система
– Сервис СУДИС
– Средства защиты от несанкционированного доступа
– Аппаратный идентификаторы rutoken, крипто-про
– Комплекс организационных мер

Криптографическая защита каналов связи

Программное обеспечение

– ViPNet Administrator.
– ViPNet StateWatcher.
– ViPNet Client.

Программно-аппаратные комплексы (ПАК)

– ViPNet Coordinator HW 2000
– ViPNet Coordinator HW 1000

Сервис управления доступом к информационным системам

Сервис обеспечивает:

– Управление доступом пользователей в систему
– Управление доступом пользователей к сервисам ИСОД
– Единую точку входа в сервисы ИСОД
– Регистрацию событий безопасности

Сервис позволяет:

– Организовать контроль пользователей, имеющих доступ к сервисам ИСОД МВД России
– Обеспечить доступ к ресурсам по электронной подписи

Данный сервис является одним из ключевых элементов подсистемы информационной безопасности ИСОД МВД России

Сервисы ИСОД МВД России

Прикладные сервисы обеспечения повседневной деятельности подразделений МВД включают:

- сервис электронного документооборота (СЭД);
- сервис электронной почты (СЭП);
- ведомственный информационно-справочный портал (ВИСП);
- систему видеоконференцсвязи.

Прикладные сервисы обеспечения оперативно-служебной деятельности включают:

- информационно-поисковый сервис «Следопыт-М»;
- сервис обеспечения охраны общественного порядка (СООП);
- сервис обеспечения деятельности дежурных частей (СОДЧ);
- сервис обеспечения деятельности подразделений материально-технического обеспечения МВД (СОМТО);
- федеральную информационную систему ГИБДД (ГИБДД-М);
- сервис обеспечения экономической безопасности (СОЭБ);
- сервис НЦБ Интерпола (СОДИ).

Прикладные сервисы обеспечения оперативно-служебной деятельности включают:

- сервис экспертно-криминалистической деятельности (ЕАИС ЭКП);
- сервис обеспечения государственной защиты лиц (СУОГЗ);
- сервис оформления проезда сотрудников (СОПС);
- сервис ГУ собственной безопасности МВД (СОПД ГУСБ);
- сервис статистической отчетности (МОСТ);
- банк отпечатков пальцев (ЦИАДИС).

Подсистема поддержки взаимодействия с населением, а также межведомственного взаимодействия с целью предоставления госуслуг включает:

- сервис предоставления госуслуг (СПГУ);
- систему централизованного учета оружия (СЦУО);
- единый банк данных архивной информации (Ретроспектива);
- интегрированный банк данных.

Перспективы развития ИСОД МВД России

Отдельным направление развития станет мобильный доступ сотрудников ОВД, входящих в состав нарядов ППС, ДПС ГИБДД, УУП к

информационным ресурсам Системы с использованием сетей операторов сотовой связи либо систем спутниковой связи.

Кроме доступа к информационным ресурсам Системы, подразделениям МВД России будет обеспечен доступ к глобальной навигационной спутниковой системе (ГЛОНАСС) с использованием соответствующего оборудования, для подсистемы мониторинга и управления силами и средствами МВД России. Будет создана возможность санкционированного доступа любого пользователя МВД России к навигационной информации, необходимой для решения его функциональных задач.

Итогом указанного накопления и обработки информации должен стать требуемый уровень информационно-аналитической поддержки принятия решений по направлениям деятельности МВД России на всех этапах управления (прогнозирования, планирования, финансирования, регулирования, контроля) заключающийся в наличие у соответствующего руководителя в режиме реального времени:

– результатов обобщения большого количества разноформатных данных, представленных в требуемом виде (справка, отчёт) как итог информационно-справочного обслуживания,

Итогом указанного накопления и обработки информации должен стать требуемый уровень информационно-аналитической поддержки принятия решений по направлениям деятельности МВД России на всех этапах управления (прогнозирования, планирования, финансирования, регулирования, контроля) заключающийся в наличие у соответствующего руководителя в режиме реального времени:

– результаты служебной деятельности ОВД после проведения над разноформатными данными в автоматическом режиме сложных модельных расчётов (для получения новых знаний) в наглядной форме (график, диаграмма), как итог информационно-аналитического обслуживания.

– результаты обработки разноформатной оперативно-технической информации, полученной в ходе оперативных мероприятий для решения задач оперативно-технических подразделений МВД России.

Практические занятия № 1-2.

Создание базы данных и управление экземпляром PostgreSQL

1. Установка и настройка PostgreSQL

Загрузка и установка PostgreSQL

Первым шагом в работе с PostgreSQL является его установка. PostgreSQL можно установить на различных операционных системах, включая Linux, Windows и macOS.

На Linux-системах, вы можете использовать пакетные менеджеры, такие как APT (для Debian/Ubuntu) или YUM (для CentOS/Red Hat), чтобы установить PostgreSQL. Например, для Debian/Ubuntu можно выполнить команду:

На Windows и macOS можно скачать инсталлятор с официального сайта PostgreSQL (https://postgresql.org/download/) и следовать инструкциям по установке.

Во время установки, вы должны выбрать пароль для суперпользователя (обычно называется «postgres»). Этот пароль будет использоваться для управления экземпляром PostgreSQL.

PostgreSQL имеет несколько конфигурационных файлов, которые определяют поведение сервера. Один из основных файлов – это postgresql.conf. Вы можете найти его в каталоге данных PostgreSQL (обычно /etc/postgresql/версия/мажорная_версия/main/ на Linux).

В postgresql.conf можно настроить различные параметры, такие как порт, размер буфера, максимальное количество подключений и другие. Эти параметры зависят от потребностей и ресурсов сервера.

Кроме того, существует файл pg_hba.conf, который управляет методами аутентификации и разрешениями для подключения к серверу.

После установки PostgreSQL, можно запустить сервер с помощью команды, например, на Linux:

или на Windows:

Для остановки сервера используются аналогичные команды, заменяя «start» на «stop».

После успешного запуска, сервер PostgreSQL будет работать на определенном порту (по умолчанию 5432) и готов к принятию подключений.

Чтобы убедиться, что сервер работает, можно использовать команду «pg_isready»:

Здесь «dbname» – имя базы данных, «username» – имя пользователя, «localhost» – адрес сервера, а «5432» – порт. Команда сообщит, готов ли сервер к подключению.

2. Создание и удаление баз данных

Определение цели создания новой базы данных: перед тем как начать процесс создания новой базы данных, важно определить цели и требования для этой базы данных.

Для создания новой базы данных в PostgreSQL, используйте команду SQL CREATE DATABASE. Пример:

Вы можете также определить различные параметры базы данных, такие как кодировка, сортировка, владелец и другие.

После создания базы данных, вы можете начать создавать таблицы и определять структуру данных, которая соответствует вашим целям.

Важно осторожно подходить к удалению базы данных, так как это может привести к потере всех данных, которые она содержит. Убедитесь, что вы абсолютно уверены в необходимости удаления базы данных.

Для удаления базы данных в PostgreSQL, используйте команду SQL DROP DATABASE. Пример:

Прежде чем удалить базу данных, убедитесь, что все соединения с ней завершены, и что никакие процессы не используют эту базу данных. В противном случае, удаление будет невозможно.

PostgreSQL обычно запрашивает подтверждение перед удалением базы данных, чтобы избежать случайных удалений.

Перед удалением базы данных рекомендуется создать резервную копию всех важных данных, чтобы иметь возможность восстановить информацию в случае ошибки. Удаление базы данных также может потребовать административных прав доступа, поэтому удостоверьтесь, что у вас есть необходимые права, чтобы выполнить эту операцию.

3. Управление пользователями и ролями

Для создания нового пользователя используется команда SQL CREATE ROLE. Пример:

Здесь username – имя нового пользователя, LOGIN указывает, что пользователь имеет право входа в систему, PASSWORD 'password' - устанавливает пароль для пользователя.

После создания пользователя, вы можете назначать ему различные роли и привилегии. Роли определяют уровень доступа и права пользователя. PostgreSQL предоставляет несколько встроенных ролей, таких как SUPERUSER, CREATEDB, CREATEROLE, и другие.

Для назначения прав доступа используется команда SQL GRANT. Пример:

Эта команда назначает разрешения SELECT, INSERT, UPDATE и DELETE на указанную таблицу table_name пользователю username.

Для отзыва прав доступа используется команда SQL REVOKE. Пример:

Эта команда отзывает право SELECT на указанной таблице table_name у пользователя username.

Роли могут быть созданы с помощью команды **CREATE ROLE**, как и пользователи. Однако, роли могут быть присвоены другим ролям и использованы для организации пользователей в иерархических группах.

Роли могут быть назначены пользователям и другим ролям с использованием команды **GRANT**. Это позволяет наследовать права и настройки от одной роли к другой.

PostgreSQL предоставляет несколько встроенных ролей, таких как **PUBLIC**, **CONNECT**, **CREATE**, **TEMP**, и другие. Они имеют определенные права и ограничения, которые могут быть настроены в зависимости от требований.

Для удаления пользователя используется команда SQL **DROP ROLE**. Пример:

Перед удалением пользователя, убедитесь, что он не владеет какими-либо объектами в базе данных и не имеет активных сеансов.

Роли также могут быть удалены с использованием команды **DROP ROLE**. Удаление ролей может потребовать особого внимания, так как оно может повлиять на множество пользователей.

4. Работа с таблицами

Прежде чем создать таблицу, важно определить её структуру. Это включает в себя определение имени таблицы и столбцов, а также их типы данных. Пример создания таблицы:

В этом примере создаётся таблица «products» с тремя столбцами: «id», «name», и «price». Столбец «id» определён как SERIAL, который автоматически генерирует уникальные значения.

Вы также можете определять ограничения (constraints) для таблицы, такие как первичный ключ (PRIMARY KEY), внешний ключ (FOREIGN KEY), уникальность (UNIQUE), и другие, чтобы обеспечить целостность данных и связи между таблицами.

PostgreSQL предоставляет разнообразные типы данных, которые позволяют точно определить, какие виды данных могут храниться в столбцах таблицы. Некоторые распространенные типы данных включают:

Целые числа: INT, SMALLINT, BIGINT.

Десятичные числа: NUMERIC, DECIMAL.

Символьные строки: VARCHAR, CHAR, TEXT.

Дата и время: DATE, TIME, TIMESTAMP.

Булевы значения: BOOLEAN.

Бинарные данные: BYTEA.

JSON, JSONB и другие для хранения JSON-данных.

Геоданные: географические типы (GEOMETRY, GEOGRAPHY).

Для вставки данных в таблицу в PostgreSQL используется команда SQL **INSERT INTO**. Пример:

Эта команда вставляет новую запись в таблицу «products» с указанными значениями для столбцов «name» и «price».

Для обновления данных в таблице используется команда SQL **UPDATE**. Пример:

```
UPDATE products SET price = 899.99 WHERE name = 'Ноутбук';
```

Эта команда обновляет цену продукта с именем 'Ноутбук'.

Для удаления данных из таблицы используется команда SQL **DELETE**. Пример:

Эта команда удаляет все записи о продуктах с именем 'Ноутбук' из таблицы «products».

Для извлечения данных из таблицы используется команда SQL SELECT. Пример:

Эта команда извлекает все записи из таблицы «products», где цена меньше 1000.

SQL-запросы могут быть более сложными, включая объединение таблиц, сортировку, фильтрацию и агрегацию данных для получения необходимой информации.

5. Резервное копирование и восстановление данных

В PostgreSQL существуют различные способы создания резервных копий данных:

pg_dump – эта команда создает текстовую резервную копию данных, которую можно восстановить с помощью команды **pg_restore**.

pg_dumpall – эта команда создает резервную копию всех баз данных и пользователей в системе.

Резервные копии должны храниться в надежном месте, отдельно от сервера базы данных. Это может быть на физических носителях (например, жестких дисках, сетевых хранилищах) или в облаке.

Для восстановления данных из текстовой резервной копии, созданной с помощью **pg_dump**, вы можете использовать команду **pg_restore**. Пример:

```
UPDATE products SET price = 899.99 WHERE name = 'Ноутбук';
```

Здесь **target_database** – это имя базы данных, в которую вы хотите восстановить данные, а **backup_file** - путь к файлу резервной копии.

Восстановление данных с использованием команды **pg_dumpall**:

Если вы создали резервную копию всей системы с помощью **pg_dumpall**, вы можете использовать эту команду для восстановления всей системы.

В PostgreSQL также можно создавать физические резервные копии (например, с использованием инструмента **pg_basebackup**). Восстановление физических копий обычно требует более сложных процедур, и оно используется, например, для создания репликации или восстановления на новом сервере.

После создания резервных копий, важно регулярно проверять, что процедуры восстановления работают как ожидалось. Это можно сделать путем тестирования восстановления на отдельной тестовой системе.

Автоматизация процесса резервного копирования и восстановления может упростить управление данными и обеспечить надежность восстановления при сбоях.

Помните, что восстановление данных может потребовать некоторого времени, особенно для больших баз данных. Планируйте восстановление так, чтобы минимизировать простои и потери данных в случае сбоев.

Практические занятия № 3-4.

Использование сервисов повседневной деятельности ИСОД МВД России

Прикладные сервисы обеспечения повседневной деятельности подразделений МВД России включают:

- сервис электронной почты (СЭП);
- сервис электронного документооборота (СЭД);
- ведомственный информационно-справочный портал (ВИСП);
- систему видеоконференцсвязи.

СЭП предназначен для обмена электронными сообщениями между сотрудниками Заказчика, между сотрудниками Заказчика и внешними адресатами. Обмен сообщениями осуществляется с использованием стандартных протоколов сети Интернет: SMTP, LMTP. Доступ пользователей к почтовым сообщениям осуществляется с использованием стандартных протоколов сети Интернет: POP3, IMAP, HTTPS. Доступ пользователей к СЭП осуществляется с компьютеров, подключенных к Интегрированной мультисервисной телекоммуникационной сети.

Функции сервиса СЭП:

СЭП обеспечивает функции доступа пользователей к почтовому ящику:

- аутентификация пользователей при доступе к почтовому ящику;
- просмотр списка папок в почтовом ящике;
- просмотр списка сообщений;
- просмотр сообщений;
- создание новых сообщений;
- прикрепление к новым сообщениям файлов документов в качестве вложений;
- отправка сообщений;
- просмотр личной адресной книги и адресной книги организации;
- управление почтой:
- создание правила автоматического ответа;
- создание правил, позволяющих выполнять различные действия в зависимости от условий;
- настройка переадресации почтовых сообщений;
- работа со средствами совместного доступа:

– работа с календарями;
– работа со списками рассылки;
– работа с задачами.

СЭП обеспечивает функции доставки электронных сообщений от отправителя получателю, а именно:

– отправку исходящих сообщений от пользователей СЭП;
– прием входящих сообщений для пользователей СЭП;
– хранение сообщений в очереди в случае временной невозможности доставки;
– уведомление пользователей о невозможности доставки сообщений.

Минимальный состав используемых технических (аппаратных) средств для клиентской части:

– Наличие подключения к ИМТС.
– Конфигурация АРМ, позволяющая запустить:
  - веб-браузер с поддержкой HTTP/1.1 (для базового интерфейса), Adobe Flash (для полного интерфейса);
  - почтовый клиент, поддерживающий протоколы IMAP, POP3, SMTP.

Минимальный состав программных средств для клиентской части

– Операционная система – Microsoft Windows, Linux, FreeBSD, MacOS и иные. Дополнительное ПО:
  - веб-браузер, поддерживающий HTTP/1.1 (для базового интерфейса), Adobe Flash (для полного интерфейса);
  - почтовый клиент, поддерживающий протоколы IMAP, POP3, SMTP.

Требования к персоналу (оператору)

Конечный пользователь сервиса «СЭП» (оператор) должен обладать практическими навыками работы с пользовательским интерфейсом операционной системы, установленной на компьютере.

Прочие требования

Работа с сервисом «СЭП» возможна несколькими способами:

– с использованием веб-браузера;
– с использованием почтового клиента. Во всех случаях для использования сервиса «СЭП» пользователю необходимо знать следующие данные:
– электронный почтовый адрес;

– имя учетной записи для аутентификации;

– пароль учетной записи для аутентификации;

– имя почтового сервера для подключения через веб-браузер - post.mvd.ru;

– имя почтового сервера для подключения по протоколу IMAP - post.mvd.ru;

– имя почтового сервера для подключения по протоколу SMTP - post.mvd.ru;

– имя почтового сервера для подключения по протоколу POP3 - post.mvd.ru.

Имена почтового сервера для различных подключений совпадают и представляют собой DNS-имя post.mvd.ru. Работа с сервисом «СЭП» возможна с компьютера, имеющего подключение к внутренней сети (ИМТС) МВД России.

Предоставление доступа к СЭП

Для работы с СЭП пользователь должен иметь действительную учетную запись (логин и пароль) пользователя ИСОД, которая создается в СУДИС. В случае отсутствия учетной записи, необходимо подать заявку на ее создание в соответствии с действующими процедурами и регламентом ИСОД.

Начало работы:

– Убедиться, что рабочее место подключено к ИМТС в зависимости от способа доступа):

– через WEB-интерфейс:

– открыть браузер (Internet Explorer или другой)

– Перейти по адресу: https://post.mvd.ru

– через почтовый клиент:

– открыть почтовый клиент (клиент предварительно должен быть настроен).

Сервис электронного документооборота (СЭД) предназначен для повышения эффективности организационно-управленческой (административной) деятельности органов внутренних дел, связанных с документационным обеспечением органов внутренних дел и представлением юридически значимого документооборота.

Доступ пользователей к Сервису осуществляется исключительно с компьютеров, подключенных к ИМТС.

Сервис решает следующие задачи:

– повышение полноты и уровня оперативности информационной поддержки принятия управленческих решений в системе МВД России;

– повышение уровня информационной поддержки и эффективности организационно-управленческой (административной) деятельности подразделений делопроизводства и режима территориальных подразделений органов внутренних дел Российской Федерации;

– повышение уровня информационного взаимодействия в части обмена юридически значимыми электронными документами;

– создание единого информационного пространства документационного обеспечения в органах внутренних дел Российской Федерации;

– оптимизация потоков документов между различными уровнями органов внутренних дел Российской Федерации;

– обеспечение единообразия работы с электронными документами в органах внутренних дел Российской Федерации с сохранением защищённости, управляемости и доступности документов;

– обеспечение сквозного контроля прохождения документов и повышение уровня исполнительской дисциплины сотрудников;

– обеспечение безопасного обмена документами в рамках Сервиса;

– обеспечение безопасного хранения и разграничения доступа к информации, включая журналирование действий пользователей и электронную подпись.

Областью применения СЭД является деятельность сотрудников, федеральных государственных гражданских служащих и работников центрального аппарата МВД России, территориальных органов МВД России, а также иных организаций и подразделений, созданных для выполнения задач и осуществления полномочий, возложенных на органы внутренних дел, направленная на подготовку, обработку, хранение и использование документов (в том числе и документов в электронной форме), образующихся в ходе деятельности органов внутренних дел.

Реализация функций в СЭД МВД выполняется в определённом Сервисе, ниже перечислены функции:

– Регистрация документа;
– Редактирование документа;
– Настраивание нумерации документа;
– Добавление штрих-кода;

- Создание ссылки на документ;
- Размножение документа;
- Возврат документа с размножения;
- Передача документа на доклад должностному лицу;
- Отправка документа;
- Выдача документа по разносной книге;
- Формирование сопроводительного письма;
- Пересылка документа;
- Досылка документа;
- Формирование отчётов по документам;
- Отправка документа на размножение;
- Фиксирование информации о прохождении документа;
- Подшивка документа в дело;
- Отмена подшивки документа;
- Формирование описи дела;
- Добавление электронного образа;
- Просмотр электронного образа;
- Удаление электронного образа;
- Выгрузка электронного образа;
- Генерация номеров штрих-кодов;
- Печать штрих-кода;
- Идентификация документа по штрих-коду;
- Добавление резолюции;
- Просмотр и редактирование резолюции;
- Удаление резолюции;
- Утверждение и отклонение резолюции;
- Постановка документа на контроль;
- Просмотр данных о контроле документа;
- Постановка документа на параллельный контроль;
- Снятие документа с контроля;
- Печать отчётных форм по контрольным документам;
- Создание проекта документа;
- Редактирование проекта документа;
- Формирование маршрута согласования;
- Согласование (визирование) проекта документа;

– Подписание проекта документа;
– Создание проекта резолюции;
– Фильтрация документов;
– Поиск документов;
– Формирование реестра и списка;
– Добавление документа в реестр и список;
– Удаление документа из реестра и списка;
– Формирование отчёта;
– Просмотр информации о совершённых действиях;
– Добавление значения в справочник;
– Удаления значения из справочника;
– Редактирование значения в справочнике;
– Настройка отображения документов.

СЭД разделён на следующие группы сервисов:

– интеграционные сервисы – набор сервисов, обеспечивающих интеграцию между документационными подразделениями и единое пространство документооборота в рамках МВД России:
– сервис взаимодействия с СИА;
– сервис хранения документов;
– сервис работы с адресными справочниками;
– сервис ведения организационно-штатной структуры МВД России (ОШС МВД);
– сервис интеграции с внешними системами (МЭДО).

Задание 1. Использование сервиса электронной почты

1. Открыть браузер и зайти на сайт https://post.mvd.ru

2. Для авторизации в СЭП необходимо ввести данные своей учетной записи пользователя ИСОД в соответствующие поля формы

3. Учетную запись (Имя пользователя/логин и пароль) пользователя ИСОД можно получить у администратора доступа Вашего региона.

4. Далее необходимо выбрать тип интерфейса:

– «Полный» - рекомендованный вид интерфейса;
– «Базовый» - для устаревших АРМ, работающих через медленные или нестабильные сетевые подключения;

и нажать кнопку “Войти”.

После успешной авторизации вы будете перенаправлены на главную страницу СЭП.

5. Самостоятельно просмотреть интерфейс сервиса.

Задание 2. Использование сервиса электронного документооборота

Регистрация входящего документа.

Для регистрации входящего документа необходимо в форме регистрации входящего документа заполнить поля, после чего выполнить одно из нижеуказанных действий:

1. Нажать кнопку **«Сохранить»**, экранная форма РКФ обновится, и добавятся доступные действия с РКФ ().

2. После сохранения документа для первого указанного в таблице «Прохождение документа» адресата значение поля «Отметка о прохождении» будет «Первичное рассмотрение» (если указанный адресат – должностное лицо, ответственное за первичное рассмотрение – см. «Руководство администратора», раздел «Первичное рассмотрение по умолчанию»).

3. Если в поле «Кому» было указано более одного адресата, то для остальных адресатов поле «Отметка о прохождении» после сохранения заполняется следующим образом:

– Руководитель подразделения или заместители руководителя подразделения (должностные лица из групп «Руководство <Наименование текущего подразделения>» и «Заместители руководителя <Наименование текущего подразделения>»), а также должностные лица из группы «Должностные лица, которым документы отправляются на доклад» – *«На доклад»*;

– Рядовые сотрудники – *«На исполнение»*;

– Сотрудники других подразделений – *«На исполнение»*.

ЗАКЛЮЧЕНИЕ

В методических рекомендациях «Системы управления базами данных специального назначения» приведены основы функционирования СУБД специального назначения, возможности их применения в деятельности органов внутренних дел, а также описаны практические работы для выработки умений и навыков для решения профессиональных прикладных задач.

В представленных методических рекомендациях содержатся систематизированные сведения о различных аспектах применения СУБД в деятельности органов внутренних дел Российской Федерации.

Методические рекомендации включают в себя теоретические сведениях и практические занятия для слушателей Воронежского института МВД России, обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника.

Также необходимо отметить, что данные методические рекомендации могут быть полезны для организации служебной подготовки, переподготовки и повышения квалификации для сотрудников и работников инженерного профиля подразделений органов внутренних дел.