

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **КУРСОВАЯ РАБОТА** | | | | | | | | | | | | | |
| по: | ПМ. 07 Соадминистрирование баз данных и серверов | | | | | | | | | | | | |
| междисциплинарного курса: | | | | | | МДК.07.01 Управление и автоматизация баз  данных | | | | | | | |
| на тему: | | | | | | | | | | | | | |
| «Администрирование и автоматизация базы данных для управления процессом городского планирования и развития.» | | | | | | | | | | | | | |
| Выполнена студентом: | | | | | Максимом Павловичем Кириллиным | | | | | | | | |
|  | | | | | И.О. Фамилия | | | | | | | | |
| Группы: | | 4ИСП9-14 | | | | | | | | | | | |
|  |  | номер группы | | | | | | | | | | | |
| Основная профессиональная образовательная программа по специальности 09.02.07 Информационные системы и программирование (администратор баз данных) | | | | | | | | | | | | | |
| шифр и наименование специальности | | | | | | | | | | | | | |
| Форма обучения | | | | очная | | | | | | | | | |
| Руководитель | | |  | преподаватель | | | |  | Даниил Сергеевич Щур | | | | |
|  | | | ученая степень, должность | | | | | | И.О. Фамилия | | | | |
|  | | |  | | | | | | оценка | | | | |
|  | | |  | | | | | | подпись, дата | | | | |
| Зав. отделением | | |  |  | | | |  | Ирина Юрьевна Васильева | | | | |
|  | | |  | | | | | | И.О. Фамилия | | | | |
|  |  |  | | |  | |  | подпись, дата | | | | | |
|  |  |  | | |  | |  |  | |  |  |  |  |
| г. Москва, 2024 г.   |  |  | | --- | --- | |  | **УТВЕРЖДАЮ**  Зам.директора по ОУП  \_\_\_\_\_\_\_\_\_\_И.Г.Бозрова  «\_\_\_»\_\_\_\_\_\_\_\_\_\_2024 г. |   **ЗАДАНИЕ**  к курсовой работе студента  Кириллина Максима Павловича  на тему: «Администрирование и автоматизация базы данных для управления процессом городского планирования и развития»   |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | Тема курсовой работы утверждена приказом директора №\_\_\_от \_\_\_\_\_\_\_\_\_\_\_ | | | | | | | | | | | | | Цель работы: | | Администрирование и автоматизация базы данных для автоматизации основных бизнес-процессов управления процессом городского планирования и развития | | | | | | | | | | | Основные вопросы, подлежащие разработке: | | | | | | | | | | | | | 1. Создание ролей пользователей и прав доступа. | | | | | | | | | | | | | 2. Настройка автоматизации и управления данными посредством применения  сценариев резервного копирования. | | | | | | | | | | | | | Теоретическая часть: | | | |  | | | | | | | | | 1. Необходимость создания ролей и прав доступа к данным. | | | | | | | | | | | | | 2. Анализ методов резервного копирования. | | | | | | | | | | | | | Практическая часть: | | | |  | | | | | | | | | 1. Создание ролей и распределение прав доступа. | | | | | | | | | | | | | 1. Разработка сценариев резервного копирования. | | | | | | | | | | | | | 1. Разработка требований к авторизации пользователей посредством написания триггера на подключение к базе данных. | | | | | | | | | | | | | Основная литература: | | | | | | | | | | | | | 1. Разработка, администрирование и защита баз данных: учебник. Федорова Г.Н., Академия, 2021 г. | | | | | | | | | | | | | 2. Базы данных. Проектирование, программирование, управление и администрирование. Волк В.К., Лань, 2021 г. | | | | | | | | | | | | | 3. Базы данных: модели, разработка, реализация: учебное пособие. Карпова Т. С., ИНТУИТ, 2022 г. | | | | | | | | | | | | | Руководитель: | | | Даниил Сергеевич Щур | | | | | | |  | | |  |  | |  | |  |  |  |  |  |  | подпись | | Дата выдачи задания: | | | | | дд.мм.2024 г. | | | | | | | | Задание получил: | | | Максим Павлович Кириллин | | | | | | |  |  | |  |  | |  | | | | | | |  | подпись | | Дата получения задачи: | | | | | дд.мм.2024 г. | | | | | | | | | | | | | | | | | | | | |

**ПЛАН-ГРАФИК**

курсовой работы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Срок выполнения курсовой работы | |  | | |
| Выбор темы курсовой работы | |  | | |
| Подбор и анализ исходной информации | |  | | |
| Подготовка и утверждение плана курсовой работы | |  | | |
| Выполнение задач курсовой работы | |  | | |
| Оформление курсовой работы | |  | | |
| Исправление замечаний курсовой работы | |  | | |
| Предоставление готовой курсовой работы для рецензирования | |  | | |
| Предзащита курсовой работы | |  | | |
| Защита курсовой работы | |  | | |
|  |  | |  |  |

|  |  |
| --- | --- |
| Руководитель | Даниил Сергеевич Щур |
|  | подпись |
| План принял к исполнению | Максим Павлович Кириллин |
|  | подпись |
|  | дд.мм.2024 г. |
|  |  |

**СОДЕРЖАНИЕ**

[Введение 5](#_Toc158716564)

[Глава 1. Анализ настройки групповых политик. Типы резервных копий 8](#_Toc158716565)

[1.1. Анализ вариантов доступа к данным и доступного функционала 8](#_Toc158716566)

[1.1.2. Роли уровня базы данных 9](#_Toc158716567)

[1.2. Типы резервных копий. Анализ способов резервного копирования и управления сценариями копий в sql server 10](#_Toc158716568)

[Глава 2. Разработка средств администрирования баз данных 13](#_Toc158716569)

[2.1. Создание ролей и настройка прав доступа 13](#_Toc158716570)

[2.2. Разработка сценариев резервного копирования 16](#_Toc158716571)

[2.3. Разработка требований к авторизации пользователей посредством написания триггера на подключение к базе данных 17](#_Toc158716572)

[Заключение 21](#_Toc158716573)

[Список использованной литературы 22](#_Toc158716574)

**ВВЕДЕНИЕ**

Цель данной курсовой работы администрирование и автоматизация базы данных для управления процессом городского планирования и развития.

Современные базы данных представляют собой сложные программные системы, доступные для использования в сфере бизнеса, обладающие широким спектром возможностей для управления данными и создания прикладных программ.

Администрирование баз данных включает в себя процессы управления и поддержки данных, такие как их создание, обновление, удаление и контроль. Базы данных играют ключевую роль в хранении и организации данных и считаются важной частью разработки программного обеспечения.

Основные системы управления базами данных такие как Microsoft SQL Server, Oracle и другие, требуют постоянного внимания и управления. Уровень автоматизации администрирования определяет необходимость квалифицированного персонала для работы с базой данных. Автоматизация процессов может значительно снизить требования к персоналу, освободив их от многих рутинных задач.

Администрация базы данных может иметь множество целей, вот основные из них:

1. Обеспечение безопасности данных: защита данных от несанкционированного доступа, взлома или утечки. Это включает в себя установку и обновление систем безопасности, контроль доступа к данным и мониторинг активности пользователей.
2. Управление производительностью: обеспечение эффективного функционирования базы данных, оптимизация запросов к данным и минимизация времени ответа на запросы пользователей.
3. Управление резервным копированием и восстановлением: гарантия, что данные регулярно резервируются и могут быть восстановлены в случае сбоя или потери данных. Это включает в себя создание и хранение резервных копий данных, а также разработку и тестирование планов восстановления.
4. Управление изменениями: контроль и управление изменениями в структуре базы данных, включая добавление, удаление и изменение таблиц, столбцов и других элементов. Это включает в себя планирование и реализацию изменений, а также тестирование и документирование изменений.
5. Управление пользователями и правами доступа: контроль доступа пользователей к данным и ресурсам базы данных. Это включает в себя создание и управление учетными записями пользователей, установку прав доступа и мониторинг активности пользователей.
6. Мониторинг и анализ производительности: отслеживание и анализ производительности базы данных, чтобы выявлять проблемы и оптимизировать работу.
7. Управление пространством данных: эффективное управление пространством данных, чтобы обеспечить достаточное пространство для хранения данных и минимизировать затраты на хранение. Это включает в себя управление размером таблиц, индексов и других элементов базы данных.
8. Управление резервированием и масштабированием: обеспечение возможности масштабирования базы данных для удовлетворения растущих потребностей бизнеса. Это включает в себя планирование и реализацию резервирования и масштабирования, а также управление ресурсами в распределенных средах.

Конкретные цели могут варьироваться в зависимости от потребностей и требований организации.

Целью данной курсовой работы является администрирование и автоматизация базы данных для управления процессом городского планирования и развития. Деятельность организации заключается в учёте данных об объектах недвижимости, их продаже и зависимости друг от друга. В работе будут рассмотрены основные аспекты администрирование баз данных, а также приведены примеры их реализации для рассматриваемой сферы

Для достижения цели были поставлены следующие задачи:

* изучить особенности предметной области, связанные с предоставлением услуг доставки готовой еды;
* администрирование базы данных в СУБД Microsoft SQl-Server;
* изучения процесса создания резервных копий;
* изучение способ ограничения доступа к БД.

Предметом исследования является администрирование и автоматизация базы данных для управления процессом городского планирования и развития.

**ГЛАВА 1. АНАЛИЗ НАСТРОЙКИ ГРУППОВЫХ ПОЛИТИК. ТИПЫ РЕЗЕРВНЫХ КОПИЙ**

1.1. Анализ вариантов доступа к данным и доступного функционала

SQL Server предлагает роли на уровне сервера, которые помогают управлять разрешениями на сервере. Эти роли являются субъектами безопасности, которые объединяют других участников. Разрешения ролей уровня сервера применяются ко всему серверу.

Дальше описаны предопределённые роли уровня сервера и их возможности:

1. Sysadmin. Участник роли имеет полные права ко всем ресурсам SQL Server;
2. Serveradmin. Участники роли могут изменять параметры конфигурации на уровне сервера и выключать сервер;
3. Securityadmin. Участники роли управляют логинами и их свойствами. Они могут предоставлять права доступа GRANT, DENY и REVOKE на уровне сервера и на уровне базы данных, если имеют к ней доступ. Мало чем отличается от роли sysadmin, потому что участники этой роли потенциально могут получить доступ ко всем ресурсам SQL Server;
4. Processadmin. Участники роли могут завершать процессы, запущенные в SQL Server;
5. Setupadmin. Участники роли могут добавлять и удалять связанные серверы с помощью TSQL;
6. Bulkadmin. Участники роли могут запускать BULK INSERT операции;
7. Diskadmin. Участники роли могут управлять устройствами резервного копирования. На практике эта роль практически не применяется;
8. Dbcreator. Участники роли могут создавать, изменять, удалять и восстанавливать базы данных;
9. Public Каждый логин SQL Server находится в этой роли. Изменить членство public нельзя. Когда у пользователя нет разрешения для объекта, к которому он получает доступ, пользователь наследует разрешения public роли для этого объекта.

1.1.2. Роли уровня базы данных

SQL Server предлагает несколько ролей для удобства управления разрешениями в базах данных. Эти роли являются субъектами безопасности, которые объединяют других участников.

Существуют два типа ролей на уровне базы данных: предопределённые роли базы данных, которые являются стандартными для каждой базы данных, и пользовательские роли базы данных, которые можно создавать. Предопределённые роли базы данных устанавливаются на уровне базы данных и присутствуют в каждой базе данных.

Дальше описаны предопределённые роли базы данных и их возможности:

1. Db\_owner. Участники роли могут выполнять все действия по настройке и обслуживанию базы данных, включая удаление;
2. Db\_securityadmin. Участники роли могут менять членство других ролей. Участники этой группы потенциально могут увеличить свои права до db\_owner, поэтому стоит считать эту роль эквивалентной db\_owner;
3. Db\_accessadmin. Участники роли могут управлять доступом к базе данных для существующих на сервере логинов;
4. Db\_backupoperator. Участники роли могут выполнять резервное копирование базы данных;
5. Db\_ddladmin. Участники роли могут выполнять любую DDL команду в базе данных;
6. Db\_datawriter. Участники роли могут создавать/изменять/удалять данные во всех пользовательских таблицах в базе данных;
7. Db\_datareader. Участники роли могут считывать данные со всех пользовательских таблиц;
8. Db\_denydatawriter. Участники роли не могут добавлять, изменять или удалять данные в пользовательских таблицах базы данных;
9. Db\_denydatareader. Участникам роли запрещён доступ к пользовательским таблицам базы данных.

1.2. Типы резервных копий. Анализ способов резервного копирования и управления сценариями копий в sql server

Одной из основополагающих обязанностей администратора баз данных является обеспечение доступности резервных копий. Она помогает сохранить БД в случае сбоев, повреждений или разрушений в месте хранения.

SQL Server может производить несколько принципиально различных типов бэкапов. Пользователям доступно:

* полная резервная копия (Full Backup);
* дифференциальная резервная копия;
* резервная копия файлов и файловых групп;
* резервная копия журналов транзакций;
* только копирование резервных копий.

Полный бэкап базы данных создает резервную копию всей структуры базы данных и всех данных. При старте полного резервирования записывается Log Sequence Number (LSN - последовательный номер журнала), который также записывается при завершении полного резервирования. SQL Server использует LSN для определения порядка выполнения операторов INSERT, UPDATE или DELETE. Наличие LSN начала и окончания в полной резервной копии обеспечивает согласованное с точки зрения транзакций резервное копирование, так как при полном резервном копировании учитываются изменения, произошедшие во время резервного копирования. Это позволяет обрабатывать такие транзакции в процессе восстановления бэкапа. Полная резервная копия служит базовым бэкапом или отправной точкой для всех последующих дифференциальных резервных копий или бэкапов журналов.

Дифференциальное резервное копирование основывается на концепции разностного растрового изображения. SQL Server использует 8-килобайтные страницы, которые собираются в группы по 8, называемые «экстентами» (Extents). Существует битовая карта, которая используется для пометки каждого экстента, который был изменен с момента последнего полного резервного копирования. При выполнении дифференциального бэкапа именно эта битовая карта определяет, какие данные должны быть включены в резервную копию.

В зависимости от размера и интенсивности изменений в базе данных дифференциальные резервные копии зачастую готовятся быстрей, чем полный бэкап. Кроме того, дифференциальные бэкапы будут меньших размеров, поскольку включают только изменённые экстенты, а не всю базу данных.

Бэкап журнала транзакций создает резервную копию журнала транзакций SQL Server, чтобы изменения данных, хранящиеся в нем, можно было воспроизвести в процессе восстановления. Поскольку транзакции происходят в определенном порядке, бэкап журнала поддерживает этот порядок транзакций. Наличие бэкапов журналов позволяет выполнить так называемое восстановление к заданной точке во времени (point in time restore).

Tail-Log - резервные копии конечных журналов фиксируют конец журнала, даже если база данных отключена, повреждена или отсутствуют файлы данных. Это может привести к неполным метаданным команд восстановления информации и msdb. Однако неполными являются только метаданные; собранный журнал является полным и пригодным для использования.

Copy-only - этот вид резервной копии не нарушает текущую цепочку резервных копий и используется только в том случае, если вам нужно снять полную резервную копию, не задевая текущую цепочку резервной копии. За исключением этих нюансов – ничем не отличается от обычной полной копии.

Частичные резервные копии могут оказаться полезны в тех случаях, когда необходимо исключить файловые группы только для чтения. Частичная резервная копия отличается от полной тем, что содержит в себе не все файловые группы. Вместо этого (для баз данных, доступных для записи и чтения) в неё включаются все данные первичной файловой группы, всех файловых групп с режимом доступа «чтение и запись» и (дополнительно) одного или нескольких файлов только для чтения.

**ГЛАВА 2. РАЗРАБОТКА СРЕДСТВ АДМИНИСТРИРОВАНИЯ БАЗ ДАННЫХ**

2.1. Создание ролей и настройка прав доступа

Для обеспечения безопасности базы данных и сохранности данных требуется разграничить роли. Для примера будет рассмотрен процесс создания роли «administrator» для Администратора в СУБД Microsoft SQL ServerMicrosoft SQL Server.

Для этого надо раскрыть базу данных, для которой будет добавлена роль. В данном примере это «DeliveryService». Далее надо проследовать по следующему пути: безопасность, роли, роли базы данных. Дальше нужно нажать правой кнопкой мыши на «Роли базы данных» после чего откроется контекстное меню, в котором надо выбрать пункт «Создать роль базы данных» (рис. 2.1).

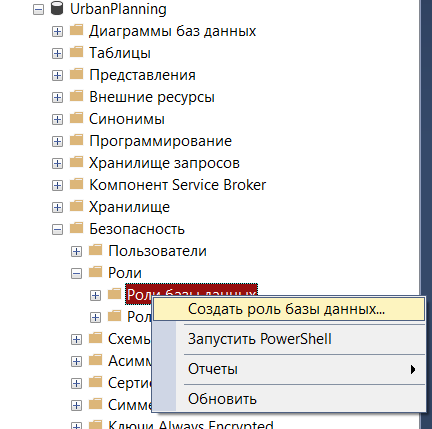


Рисунок 2.1 Создать роль базы данных

Откроется окно «Роль базы данных – создание» (рис. 2.2).

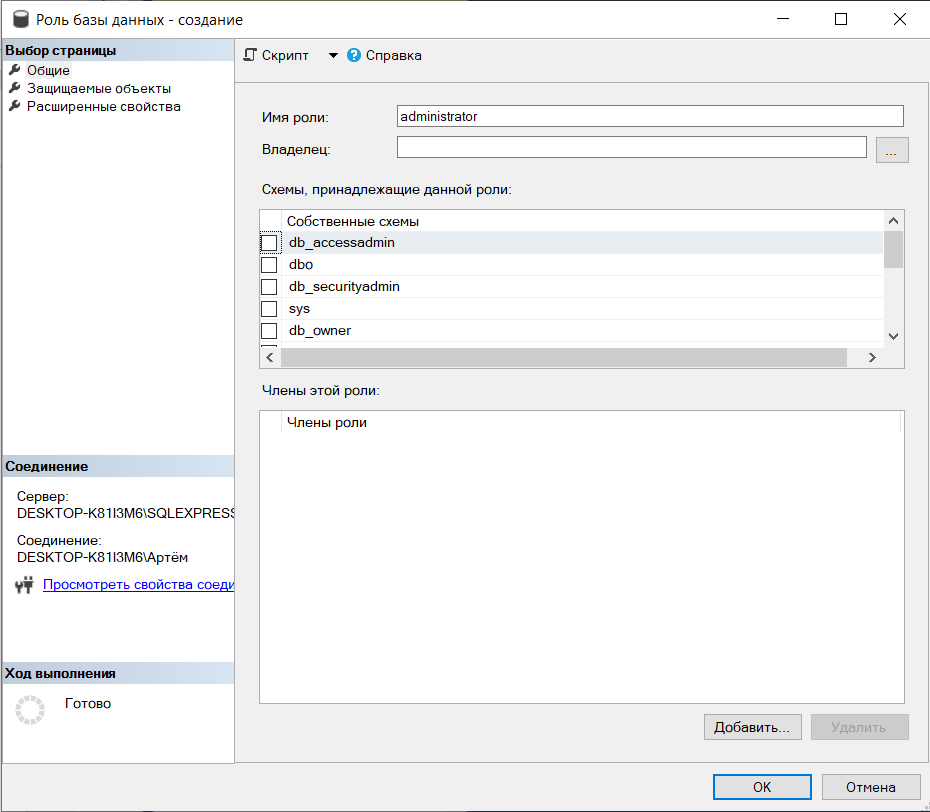


Рисунок 2.2 Создание роли

В открывшемся окне в поле «Имя роли» следует имя для новой роли, далее переходим в пункт «Защищаемые объекты».

В открывшимся окне (рис. 2.3) следует нажать кнопку «Найти», после чего выбрать «Все объекты типов».

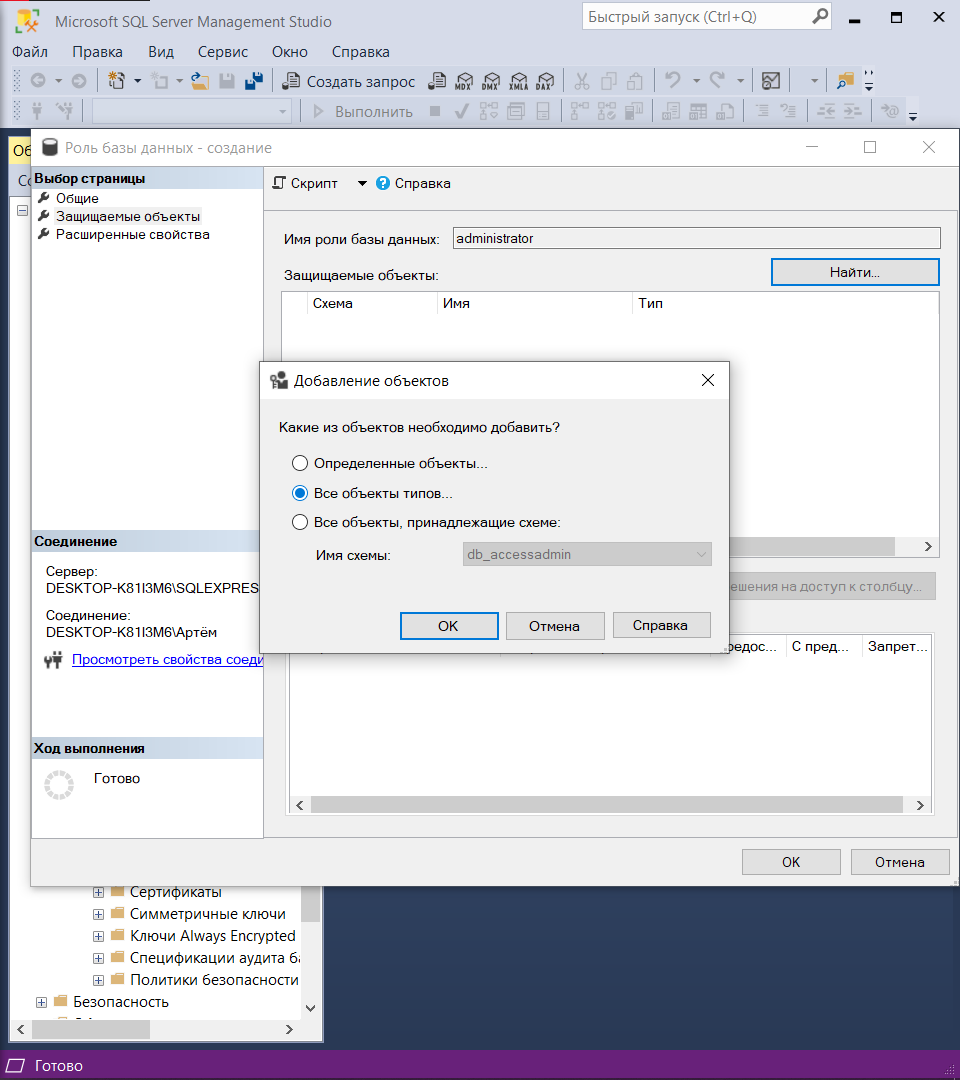


Рисунок 2.3 Добавление объектов

В следующем окне выбираем тип объектов, для которого будут выдаваться разрешения «Базы данных». (рис. 2.4).

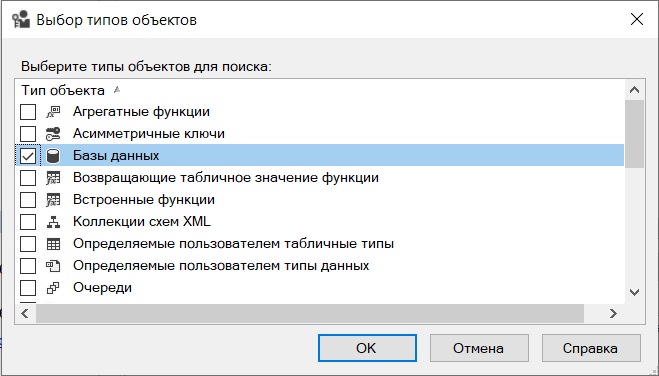


Рисунок 2.4 Выбор типов объекта

Во вкладке «Защищаемые объекты» снизу появится раздел для выбора разрешений, выставляем все галочки в столбце «Предоставить», чтобы администратор имел неограниченный доступа к базе данных (рис. 2.5)

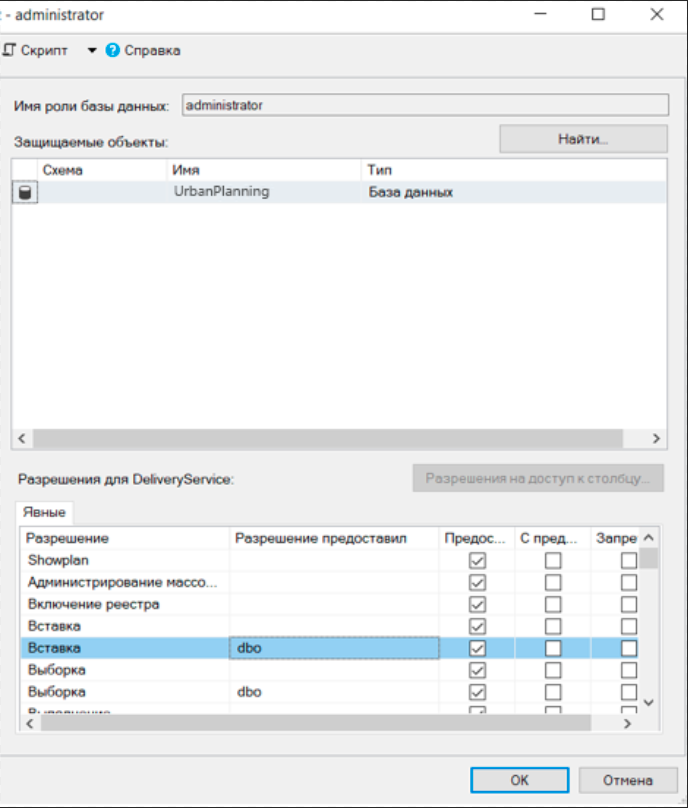


Рисунок 2.5 Разрешения для UrbanPlanning

По такому же алгоритму надо создать роль «meneger» для отдела по работе с клиентами. Эта роль будет иметь ограниченные права роли «administrator».

Также надо создать роль «courier» для отдела доставки. Эта роль будет иметь возможность только считывать данные с таблиц.

2.2. Разработка сценариев резервного копирования

Изучив и проанализировав типы резервных копий, следует спроектировать вариант резервного копирования. Два раза в сутки будет достаточно, чтобы какая-либо неполадка не повлекла за собой существенных проблем.

Для начала следует выбрать базу данных, для которой будет производиться создание резервной копии. Для этого надо будет выбрать нужную базу данных нажать на кнопку «Создать запрос» (рис. 2.6).

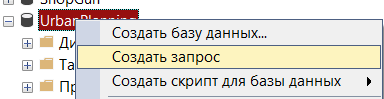


Рисунок 2.6 Создать запрос

В открывшимся окне создания запроса надо будет прописать следующий код для создания полной резервной копии с обязательным указанием куда будет сохранена резервная копия (рис. 2.7).

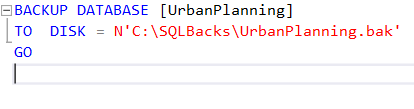


Рисунок №2.7 Код полного резервного копирования

Также для большей сохранности данных каждый час будет производиться дифференциальное резервное копирование (рис. 2.8).

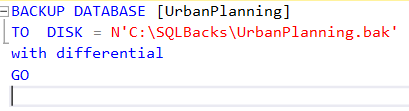


Рисунок 2.8 Код дифференциального резервного копирования

Чтобы проконтролировать успешность создания резервных копий следует написать и выполнить следующий код (рис. 2.9).

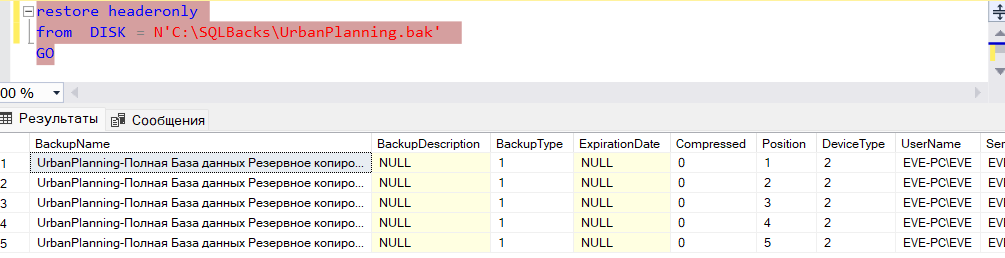


Рисунок №2.9 Проверка создания резервных копий

2.3. Разработка требований к авторизации пользователей посредством написания триггера на подключение к базе данных

Перед созданием триггера на подключение к базе данных следует создать пользователя. Для примера будет рассмотрен пример создания пользователя в СУБД Microsoft SQL Server для администратора.

Для начала нужно выбрать нужный сервер и зайти в подпапку «Имена для входа», которая находится в безопасности. После этого шага надо нажать правой кнопкой мыши на «Имена для входа». В открывшемся контекстном меню надо нажать «Создать имя для входа» (рис. 2.10).

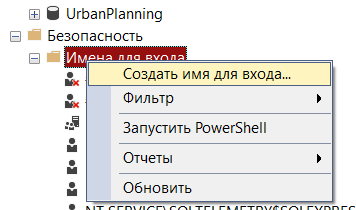


Рисунок 2.10 Создание пользователя

В открывшемся окне надо ввести «Имя для входа», выбрать «Проверка подлинности SQL Server», вписать пароль и убрать галочку с пункта «Пользователь должен сменить пароль при следующем входе» (рис. 2.11).

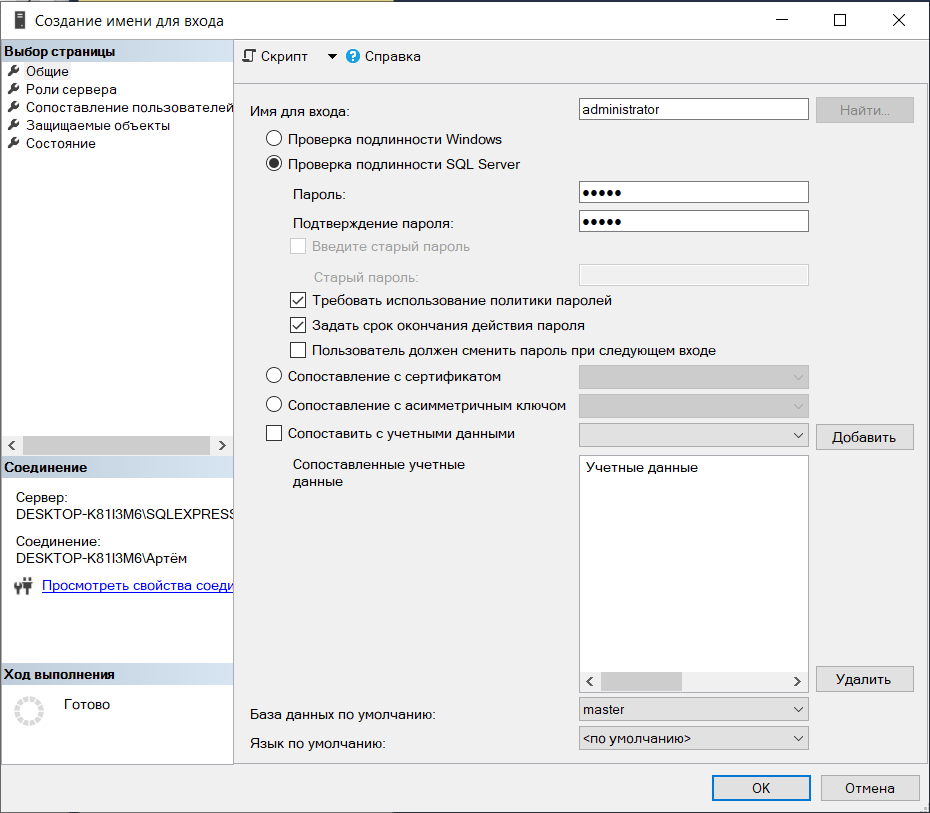


Рисунок 2.11 Создание имени для входа

Следующим шагом надо зайти в нужную базу данных и зайти в подпапку «Пользователи», в папке безопасность. Далее надо нажать правой кнопкой мыши на «Пользователи». В открывшемся контекстном меню надо нажать «Создать пользователя…» (рис 2.12).

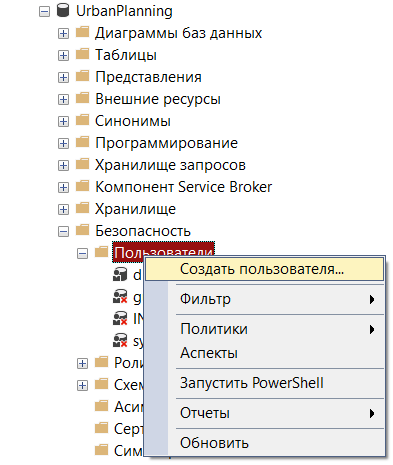


Рисунок 2.12 Создание пользователя

В открывшемся окне в поле “Имя пользователя” необходимо ввести имя нового пользователя и выбрать ранее созданное имя для входа. В” Схема по умолчанию” для пользователя нужно выбрать объект «dbo» (рис. 2.13).

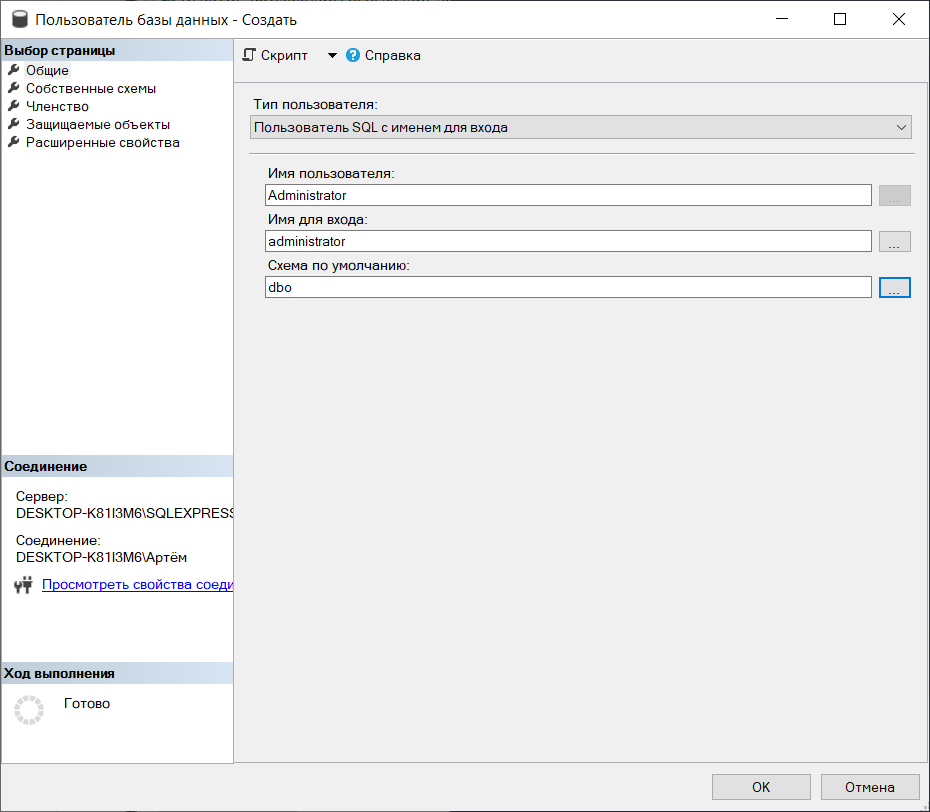


Рисунок 2.13 Создание пользователя

После этого необходимо зайти во вкладку «Членство» и выбрать ранее созданную роль «administrator» (рис. 2.14).

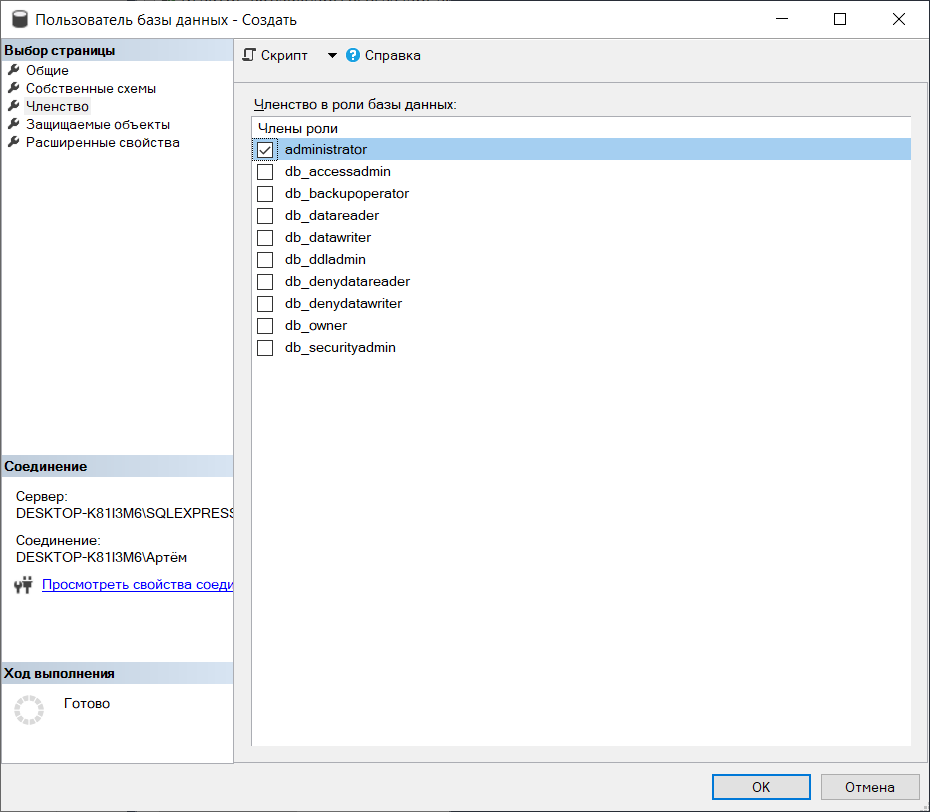


Рисунок 2.14 Выбор ролей

Следующим шагом создаём новый запрос, в котором прописываем код триггера на авторизацию, который ограничит количество входов на сервер для пользователей с определённой ролью (рис. 2.15).

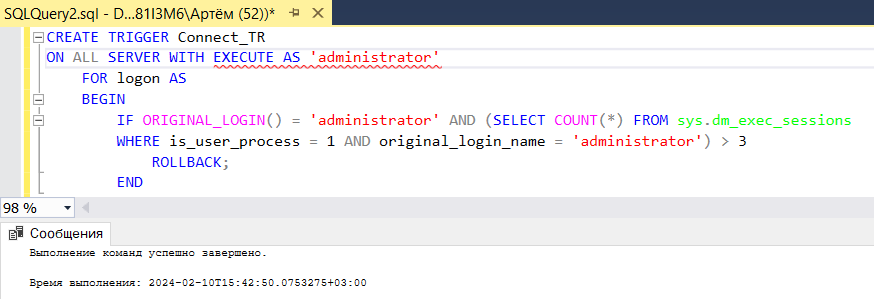


Рисунок 2.15 Код создания триггера

**ЗАКЛЮЧЕНИЕ**

Администрирование баз данных является важным аспектом разработки и поддержки программного обеспечения. Администратор баз данных отвечает за управление и поддержку баз данных, включая установку и настройку, резервное копирование и восстановление, оптимизацию производительности и управление пользователями. Для администрирования баз данных используются различные технологии и СУБД, включая Oracle Database, MySQL, Microsoft SQL Server и PostgreSQL.

В заключение курсовой работы по теме «Администрирование и автоматизация базы данных для управления процессом городского планирования и развития» можно сделать следующие выводы.

В настоящее время автоматизация процессов играет важную роль в эффективном управлении организациями, включая сервисы доставки готовой еды. Автоматизация обработки и хранения информации позволяет сократить время, улучшить качество работы, повысить эффективность и уменьшить вероятность ошибок.

В процессе исследования были изучены основные принципы работы администраторов баз данных и их инструменты. Был рассмотрен пример создания ролей для управления доступом в базу данных, процесс создания резервных копий для обеспечения безопасности данных, а также настройка пользователей и триггеров для авторизации в Microsoft SQL Server.

В заключение можно отметить, что автоматизация и администрирование баз данных являются существенной частью эффективного управления любой базой данных.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Домбровская. Г., Оптимизация запросов PostgreSQL/ Домбровская. Г.: ДМК-Пресс, 2022 г.
2. Новиков. А., Основы технологий баз данных. Учебное пособие / Новиков. А. ДМК-Пресс, 2020 г
3. Дейт. К., Введение в системы баз данных. Том 2. Классическое издание / Дейт. К. Диалектика, 2024 г.

Электронные ресурсы

1. Майкрасофт / Статья «Техническая документация по SQL Server» https://learn.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-ver16 (дата обращения 18.02.2023)
2. РСВПУ / Электронное учебное пособие «основы создания и администрирования базы данных средствами microsoft sql server»  
   <https://elar.rsvpu.ru/bitstream/123456789/29378/1/RSVPU_2019_307.pdf> (дата обращения 18.02.2023)
3. Линтер / Статья «Администрирование базы данных»  
   <https://linter.ru/ru/documentations//guiutils/administrirovanie-bazy-dannyh-administrirovanie-bazy-dannyh/> (дата обращения 18.02.2023)
4. Сфиа / Статья «Администрирование базы данных DBAD» <https://sfia-online.org/ru/sfia-8/skills/database-administration>   
   (дата обращения 18.02.2023)
5. Хабр / Статья «Администрирование баз данных \* Все об администрировании БД» https://habr.com/ru/hubs/db\_admins/articles/  
   (дата обращения 18.02.2023)