**Департамент образования и науки города Москвы**

**Государственное бюджетное профессиональное**

**образовательное учреждение**

**«Колледж связи № 54» имени П.М. Вострухина**

**КУРСОВОЕ ПРОЕКТИРОВАНИЕ**

Тема: Администрирование базы данных информационно-поисковой системы для ГИБДД.

Обучающийся группы: 3ИСП11-8 Потегаев Е.В.

*номер группы И.О. Фамилия*

Специальность: 09.02.07 Информационные системы и программирование

Руководитель / Шулянов К.А.

*подпись И.О. Фамилия*

Москва

2021

Содержание

[ВВЕДЕНИЕ 3](#_Toc65072715)

[ГЛАВА I. АДМИНИСТРИРОВАНИЕ БАЗЫ ДАННЫХ "ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА ДЛЯ ГИБДД". ПОСТАНОВКА ЗАДАЧ. 4](#_Toc65072716)

[1.1. Основные понятия администрирования баз данных. 4](#_Toc65072717)

[1.2. Управление целостностью данных в СУБД 6](#_Toc65072718)

[1.3. Управление безопасностью в СУБД. 8](#_Toc65072719)

[1.4. Выбор СУБД. 9](#_Toc65072720)

[ГЛАВА II. РАЗРАБОТКА СРЕДСТВ ЗАЩИТЫ И АДМИНИСТРИРОВАНИЕ БАЗЫ ДАННЫХ "ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА ДЛЯ ГИБДД" 11](#_Toc65072721)

[2.1. Выбор и создание плана резервного копирования базы данных 11](#_Toc65072722)

[2.2. Создание ролей и пользователей на сервере БД 14](#_Toc65072723)

[2.3. Автоматизирование резервного копирования БД 18](#_Toc65072724)

[2.4. Настройка прав доступа для пользователей. 24](#_Toc65072725)

[2.5 Создание триггеров авторизации. 26](#_Toc65072726)

[ЗАКЛЮЧЕНИЕ 28](#_Toc65072727)

[СПИСОК ЛИТЕРАТУРЫ 30](#_Toc65072728)

# ВВЕДЕНИЕ

Целью выполнения выпускной курсового проектирования является систематизация и закрепление полученных теоретических знаний и практических умений по общепрофессиональным и специальным дисциплинам специальности “Информационные системы и программирование” и администрирования базы данных с использованием конкретной СУБД. В ходе выполнения данного курсового проекта необходимо оптимизировать администрирование базы данных «Информационно-поисковой системы для ГИБДД.», Информационно-поисковая система занимается хранением информации об участниках дорожного движения и их нарушениях.

Актуальность обусловлена постоянным развитием информационных технологий и необходимостью автоматизировать все процессы, для облегчения рабочего труда.

Для выполнения поставленной задачи необходимо:

1. Создать пользовательские роли
2. Выдать роли всем пользователям
3. Разработать стратегию резервного копирования
4. Создать Резервную копию
5. Создать план резервного копирования
6. Создать триггер на авторизацию

Объектом исследования является MS SQL-Server, так как именно в нём будет создана база данных и выполнятся поставленные задачи.

Предметом исследования является упростить процесс работы с информационно-поисковой системой, хранение информации о водителях, так как данная тема является важной и актуальной. Она позволяет автоматизировать процесс поиска информации об участниках дорожного движения, что бы с помощью меньших усилий можно было бы получить доступ к хранимой на разных носителях информации.

# ГЛАВА I. АДМИНИСТРИРОВАНИЕ БАЗЫ ДАННЫХ "ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА ДЛЯ ГИБДД". ПОСТАНОВКА ЗАДАЧ.

## 1.1. Основные понятия администрирования баз данных.

Администратор базы данных - лицо, отвечающее за выработку требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение, включая управление учётными записями пользователей БД и защиту от несанкционированного доступа.[1]

Функции администратора базы данных:

* Анализ предметной области
* Проектирование структуры базы данных
* Задание ограничений целостности при описании структуры базы данных и процедур обработки БД
* Первоначальная загрузка и ведение базы данных
* Защита данных от несанкционированного доступа
* Защита данных от разрушений
* Обеспечение восстановления БД
* Анализ обращений пользователей к БД
* Анализ эффективности функционирования базы данных и развитие системы
* Работа с пользователями\
* Подготовка и поддержание системных программных средств
* Организационно-методическая работа

Встроенные роли SQL Server уровня сервера:

|  |  |
| --- | --- |
| sysadmin | Дает возможность выполнять любые действия на сервере. |
| serveradmin | Позволяет изменять параметры конфигурации на уровне сервера, а также выключать его. |
| securityadmin | Может управлять логинами, их правами и свойствами, предоставлять, запрещать и отменять разрешения на уровне базы данных, сбрасывать пароли для имен входа SQL Server. |
| processadmin | Позволяет завершать процессы, работающие на экземпляре SQL Server . |
| setupadmin | Может добавлять или удалять связанные серверы с помощью инструкций Transact-SQL . |
| bulkadmin | Может выполнять инструкцию BULK INSERT. |
| diskadmin | Используется для управления файлами на диске. |
| Dbcreator | Позволяет создавать, изменять, удалять и восстанавливать любые базы данных. |
| public | Содержится в каждой базе данных, включая системные базы данных. Ее нельзя удалить, а также нельзя добавлять и удалять пользователей из нее. Разрешения, предоставленные роли public, наследуются всеми остальными пользователями и ролями, поскольку они принадлежат к роли public по умолчанию. Следует предоставлять роли public только разрешения, необходимые для всех пользователей. |

Встроенные роли SQL Server уровня базы данных:

|  |  |
| --- | --- |
| db\_owner | Дает право выполнять все действия по настройке и обслуживанию базы данных, а также удалять БД. |
| db\_securityadmin | Дает возможность управлять ролями и разрешениями пользователей, в том числе своими . |
| db\_accessadmin | Позволяет добавлять или удалять права удаленного доступа к БД. |
| db\_backupoperator | Позволяет создавать резервные копии БД. |
| db\_ddladmin | Позволяет выполнять любые команды Data Definition Language (DDL) в БД. |
| db\_datawriter | Позволяет добавлять, удалять или изменять данные во всех таблицах. |
| db\_datareader | Дает право считывать все данные из всех таблиц. |
| db\_denydatawriter | Не могут добавлять, изменять или удалять данные в таблицах БД. |
| db\_denydatareader | Не могут считывать данные из таблиц базы данных. |

## 1.2. Управление целостностью данных в СУБД

Целостность базы данных - соответствие имеющейся в базе данных информации её внутренней логике, структуре и всем явно заданным правилам.[5]

Нормальная форма - свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение. На практике в большинстве случаев является необходимой и достаточной использование трех нормальных форм.[12]

* Первая нормальная форма: переменная отношения находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.
* Вторая нормальная форма: переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый неключевой атрибут неприводимо зависит от её потенциального ключа.
* Третья нормальная форма: Переменная отношения находится в третьей нормальной форме тогда и только тогда, когда она находится во второй нормальной форме, и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

Индекс - структура данных, которая помогает СУБД быстрее обнаружить отдельные записи в файле и сократить время выполнения запросов пользователей. Хотя индексы, строго говоря, не являются обязательным компонентом СУБД, они могут существенным образом повысить ее производительность. Как и в случае с предметным указателем книги, читатель может найти определение интересующего его понятия, просмотрев всю книгу, но это потребует слишком много времени. А предметный указатель, ключевые слова в котором расположены в алфавитном порядке, позволяют сразу же перейти на нужную страницу.[3]

Для ускорения доступа к данным применяется несколько типов индексов:

Первичный индекс — это такой специальный массив-указатель порядка записей, когда файл данных последовательно упорядочивается по полю ключа упорядочения, а на основе поля ключа упорядочения создается поле индексации, которое гарантированно имеет уникальное значение в каждой записи.

Индекс кластеризации - это такой специальный массив-указатель порядка записей, когда файл данных последовательно упорядочивается по неключевому полю, и на основе этого неключевого поля формируется поле индексации, поэтому в файле может быть несколько записей, соответствующих значению этого поля индексации. Неключевое поле называется атрибутом кластеризации.

Вторичный индекс — это индекс, который определен на поле файла данных, отличном от поля, по которому выполняется упорядочение.

Нормализация баз данных — это процесс удаления избыточных данных.

Избыточность данных – это когда одни и те же данные хранятся в базе в нескольких местах, что приводит к снижению производительности.[15]

## 1.3. Управление безопасностью в СУБД.

Ответственность за предоставление прав доступа к СУБД, как правило, несёт администратор базы данных, в обязанности которого входит создание учётных записей пользователей. Каждому пользователю присваивается уникальный идентификатор, с которым связывается определённый пароль, выбираемый пользователем и известный операционной системе. При регистрации пользователь должен предоставить системе пароль для аутентификации. Для обеспечения большей безопастности, необходимо задать установку на смену пароля через определённый срок.[6]

Шифрование - это способ сокрытия исходного смысла сообщения или другого документа, обеспечивающей искажение его первоначального содержимого.

В MS SQL Server можно шифровать соединения, данные и хранимые процедуры.[7]

Шифрование на уровне ячеек:

* Пароль. Это наименее надежный способ, так как для шифрования и расшифровки данных используется одна и та же парольная фраза. Если хранимые процедуры и функции не зашифрованы, то доступ к парольной фразе возможен через метаданные.
* Сертификат. Этот способ обеспечивает надежную защиту и высокое быстродействие. Сертификат можно связать с пользователем; подписать его необходимо с помощью DMK.
* Симметричный ключ. Достаточно надежен, удовлетворяет большинству требований к безопасности данных и обеспечивает достаточное быстродействие. Для шифрования и расшифровки данных используется один ключ.
* Асимметричный ключ. Обеспечивает надежную защиту, так как применяются различные ключи для шифрования и расшифровки данных. Однако это негативно влияет на быстродействие. Специалисты Microsoft не рекомендуют использовать его для шифрования крупных значений. Асимметричный ключ может быть подписан с использованием DMK или создан с помощью пароля.

Аудит в контексте СУБД в общем случае может заключаться в сохранении информации обо всем, что происходит в базе данных. Аудит, журнализация или контроль изменений структуры и данных БД обычно решают следующие задачи:

* получение информации о том, кто, когда и откуда производил изменения структуры БД или данных;
* отслеживание истории изменения структуры БД или данных;
* уведомление об изменениях структуры БД или данных.

## 1.4. Выбор СУБД.

Microsoft SQL Server представляет собой СУБД, обеспечивающую создание информационных систем с архитектурой “клиент-сервер”, в которой он играет роль сервера баз данных. SQL Server поддерживает: тиражирование данных, параллельную обработку, создание и обработку больших баз данных, отличается простотой управления и использования.

Эта СУБД обеспечивает высокую степень защиты данных, как от случайных потерь, так и от несанкционированного доступа, обладает развитыми средствами обработки данных и хорошим быстродействием.

SQL Server позволяет хранить огромные объёмы данных. SQL Server добавляет к сетевым компонентам специальные сервисы, такие как OLEDB (Object Linking and Embedding-Database - связывание и внедрение объектов базы данных) и ODBC (Open Database Connectivity - совместимость от­крытых баз данных). С их помощью обеспечивается совместимость различных кли­ентских приложений при работе с сервером. [2]

Пользователь компьютера-клиента с помощью сетевых средств своей операционной системы может устанавливать связь с компьютером-сервером, где установлен SQL Server. На компьютерах-клиентах размещаются локаль­ные базы данных, работа с которыми ведется с помощью персональных СУБД (Access, Visual FoxPro) или языков программирования (Visual Basic, Delphi, C++ Builder, Visual C++). С их помощью через ODBC осуществляется доступ к базам данных, размещенным на сервере.

В SQL Server используется понятие роли(role). Каждому пользователю может быть назначено произ­вольное число ролей. Например, пользователю может быть назначена роль Администратора.[8]

Достоинства:

1. Бесплатная.

2. Является реляционной (Реляционная база данных представляет собой набор взаимосвязанных двухмерных таблиц (отношений)), полнофункциональной.

3. Используется для создания распределенных баз данных коллективного использования.

4. Реализует многопользовательский удаленный режим типа Клиент‑Сервер доступа к данным.

Недостатки:

1. Привязка к операционным системам Windows

2. Сложный для освоения интерфейс

# ГЛАВА II. РАЗРАБОТКА СРЕДСТВ ЗАЩИТЫ И АДМИНИСТРИРОВАНИЕ БАЗЫ ДАННЫХ "ИНФОРМАЦИОННО-ПОИСКОВАЯ СИСТЕМА ДЛЯ ГИБДД"

## 2.1. Выбор и создание плана резервного копирования базы данных

Резервное копирование - процесс создания копии данных на носителе, предназначенном для восстановления данных в оригинальном или новом месте их расположения в случае их повреждения или разрушения.[10]

В MS SQL Server предоставляется несколько методов резервного копирования для удовлетворения требований всевозможных сфер бизнеса и разнообразных применений баз данных:

|  |  |
| --- | --- |
| Полная (Full backup) | Делает копию всей базы данных, включая все объекты и данные системных таблиц. Полная резервная копия не будет усекать журнал транзакций. Это основной тип резервных копий, который требуется выполнять перед другими типами резервных копий. |
| Журнал транзакций (Transaction log backup) | Копирует все транзакции, которые произошли с момента последнего резервного копирования, а затем урезает журнал транзакций для освобождения дискового пространства. |
| Разностная (Differential backup) | Копирование только тех данных, которые появились с момента последней полной резервной копии. |
| Доступная только для копирования (Copy-only) | Не нарушает текущую цепочку резервных копий и используется только в том случае, если вам нужно снять полную резервную копию, не задевая текущую цепочку бекапов. |

При осуществлении полного резервного копирования базы данных сервером SQL Server выполняются следующие действия:

* резервное копирование всех данных в базе данных;
* резервное копирование всех изменений, которые возникают во время выполнения резервного копирования;
* резервное копирование всех транзакций, не зафиксированных в журнале транзакций.

Сервером SQL Server используются части журнала транзакций, которые были записаны в файл резервной копии для обеспечения согласованности данных при восстановлении резервной копии. Восстановленная база данных совпадает с состоянием базы данных на момент завершения резервного копирования за исключением всех незафиксированных транзакций. При восстановлении базы данных производится откат незафиксированных транзакций.[9]

Для моей базы данных я выбрал план полного резервного копирования (full backup), так как полная резервная копия представляет базу данных на момент создания резервной копии и служит основным источником данных в случае сбоя системы.

Для того что создать резервную копию нам необходимо в обозревателе объектов зайти в свойства базы данных и выбрать «Задачи» => «Создать резервную копию».

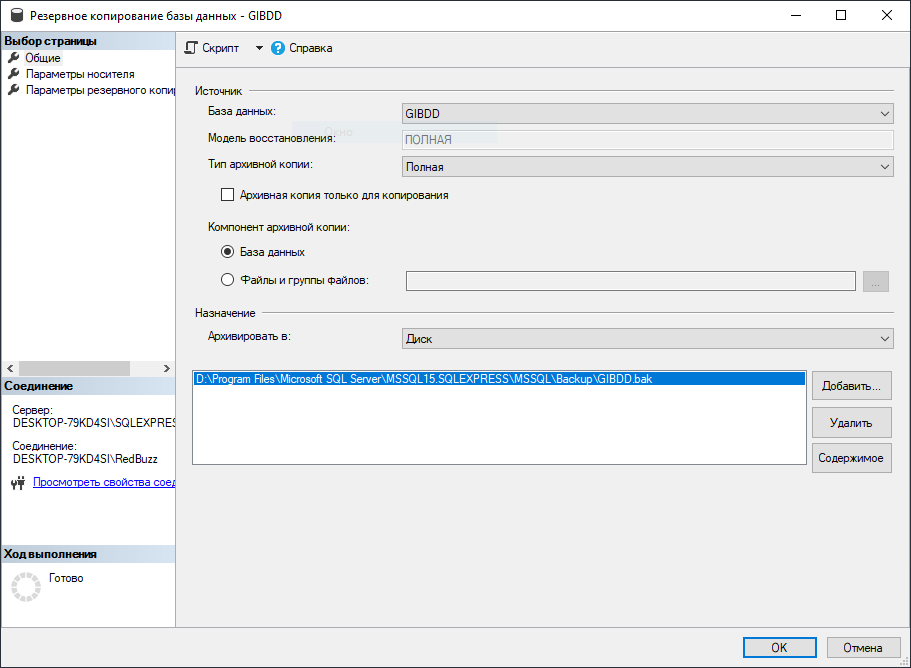


Рис. 1 «Выбор резервного копирования»

После выбора типа резервного копирования, в «Параметрах резервного копирования» можно выбрать через какой промежуток времени следует удалить резервную копию.

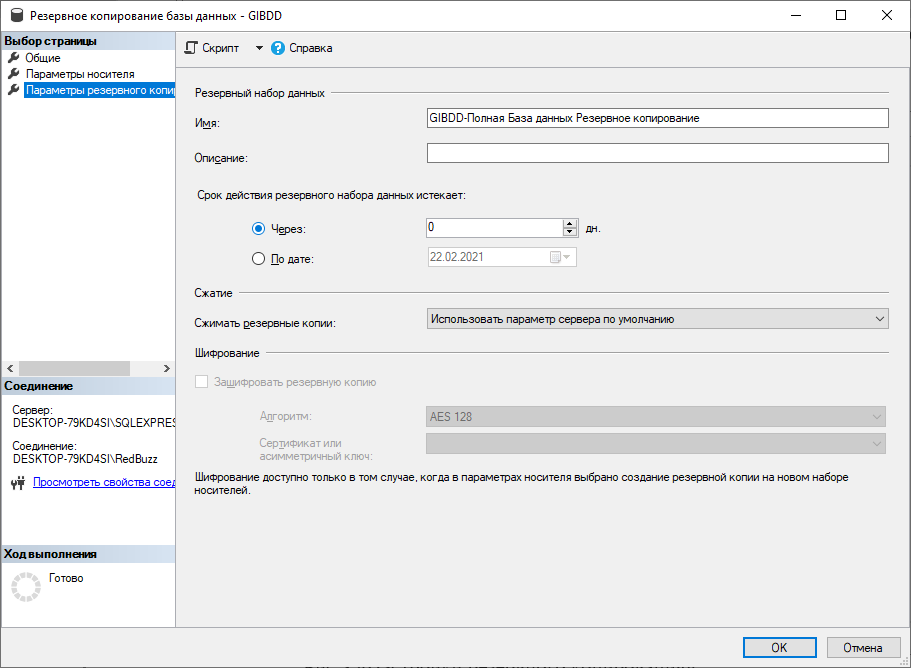


Рис. 2 «Параметры резервного копирования»

При успешном создании резервного копирования появляется диалоговое окно, оповещающее о завершении операции.

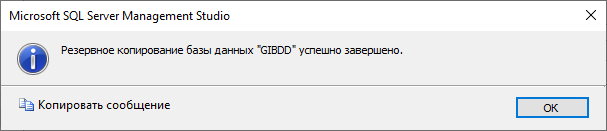


Рис. 3 «Успешное создание резервной копии»

## 2.2. Создание ролей и пользователей на сервере БД

Пользователь БД — это физическое или юридическое лицо, которое имеет доступ к БД и пользуется услугами информационной системы для получения информации. На каждом этапе развития базы данных (проектирование, реализация, эксплуатация, модернизация и развитие, полная реорганизация) с ней связаны разные категории пользователей.[4]

Существуют различные категории пользователей:

* Конечные пользователи — это основная категория пользователей, в интересах которых создается БД.
* Администратор базы данных — это лицо или группа лиц, отвечающих за выработку требований к базе данных, ее проектирование, создание, эффективное использование и сопровождение.
* Разработчики и администраторы приложений — это группа пользователей, которая функционирует во время проектирования, создания и реорганизации БД.
* Администраторы приложений — координируют работу разработчиков при разработке конкретного приложения или группы приложений, объединенных в функциональную подсистему.

В отличие от серверных ролей роли баз данных вы можете создавать самостоятельно.[11] Это можно сделать при помощи команды CREATE ROLE или через интерфейс СУБД «Безопасность» => «Роли» => «Создать роль базы данных», например:



Рис. 4 «Создание роли через запрос»

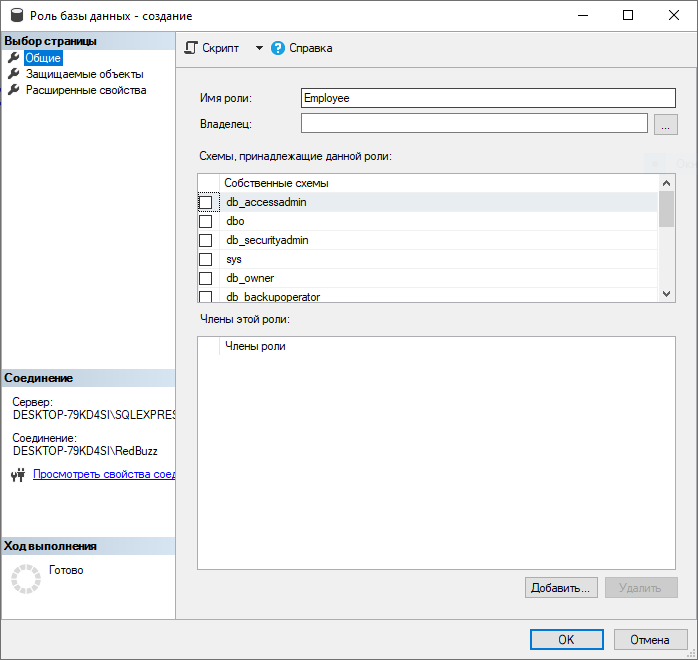


Рис. 5 «Создание роли через интерфейс»

С помощью интерфейса базы данных мы создадим пользователя. Обратимся в «Безопасность» => «Имена для входа» => «Создать имя для входа».

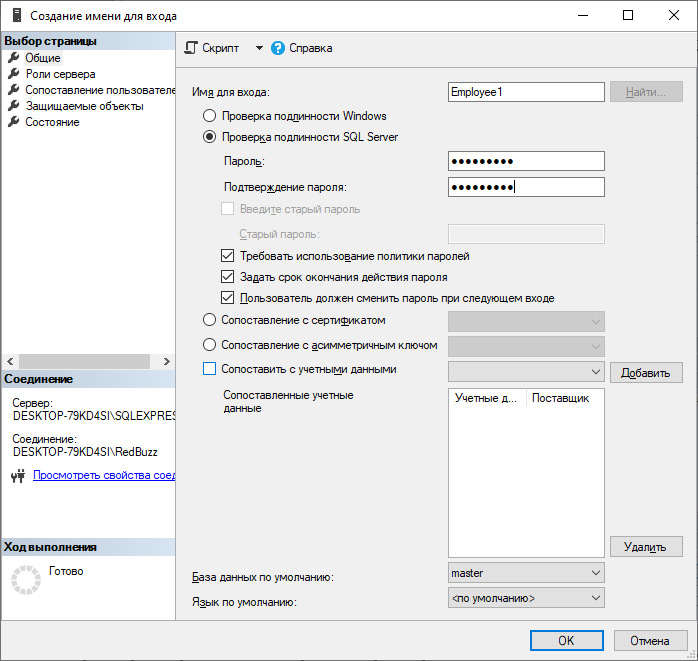


Рис. 6 «Создание имени для входа»

Устанавливаем необходимые параметры:

• Требования использование политики паролей – при использовании этого пользователя, MSSQL будет требовать пароль для входа в базу данных

• Задать срок окончания действия паролей – каждый месяц система автоматический будет просить замену пароля для этого пользователя

• Пользователь должен сменить пароль при следящем входе – MSSQL сделает запрос о сменен пароля при первой авторизации этого пользователя

В следующей вкладке «Роли сервера» проставляем роли для данного пользователя.

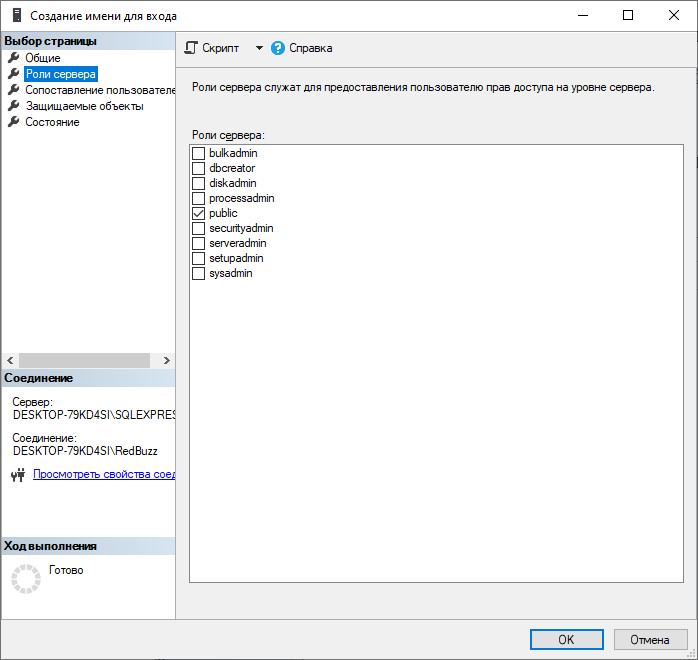


Рис. 7 «Назначение роли»

Во вкладке «сопоставление пользователей» мы указываем к каким базам данных он сможет обращаться. Также указываем ниже роли к этой базе данных.

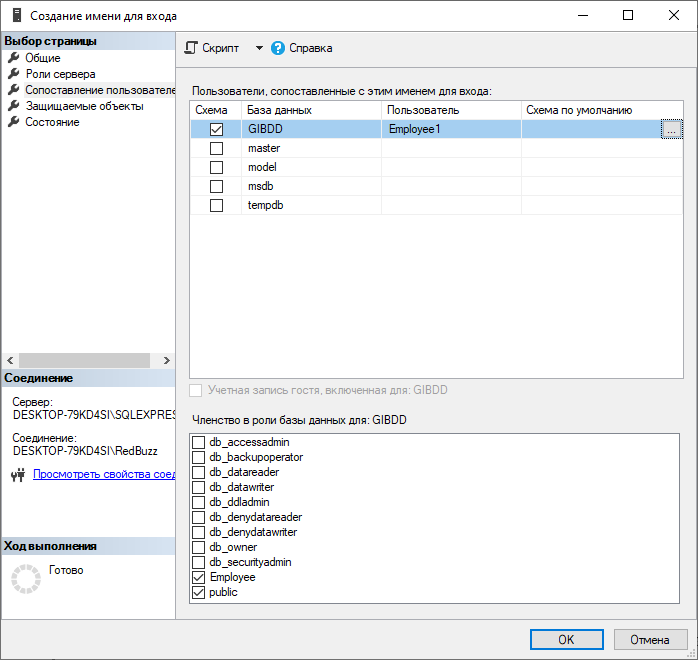


Рис. 8 «Указываем базу данных и роль в ней»

На вкладке опции «Защищаемые объекты» и «Состояние» оставляем по умолчанию.

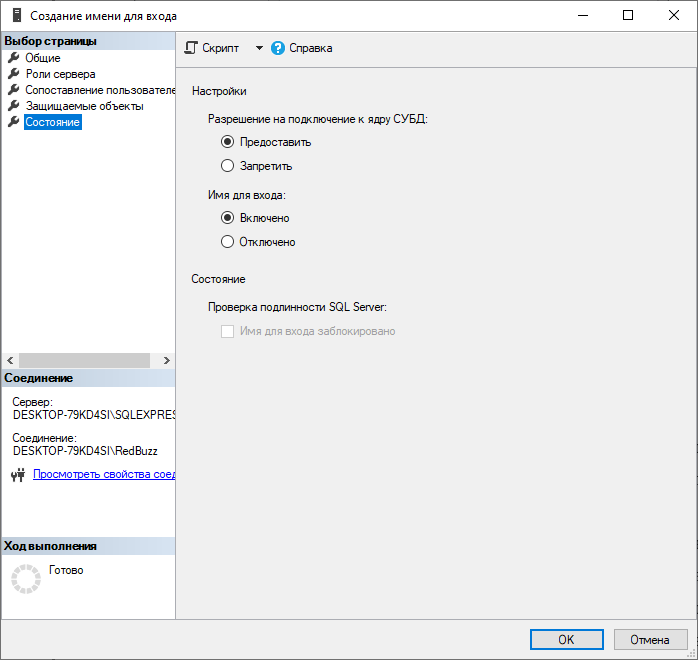


Рис. 9 «Состояние»

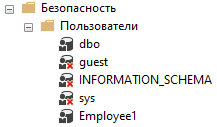


Рис. 10 «Успешное создание пользователя»

## 2.3. Автоматизирование резервного копирования БД

План обслуживания SQL Server это самый распространенный способ настройки регулярного резервного копирования.[14]

Например, зададим настройку резервного копирования базы данных на SQL Server:

* Каждые 24 часа делать полное резервное копирование
* Каждые 30 минут делать резервную копию журнала транзакций

Для начала нужно создать «Устройство резервного копирования», в «Объекты сервера» => «Устройства резервного копирования». Нужно указать имя устройства и файл, в котором будут храниться резервные копии (с расширением .bak) в дальнейшем можно посмотреть содержимое носителя, там будут перечислены все резервные копии.

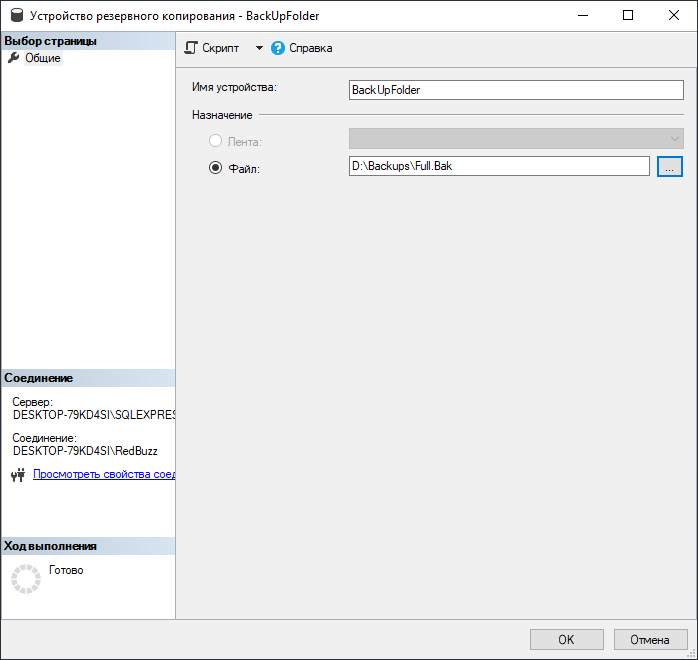


Рис. 11 «Создание устройства резервного копирования»

Далее нужно перейти в раздел «Управление» => «Планы обслуживания» => «Создать план обслуживания».

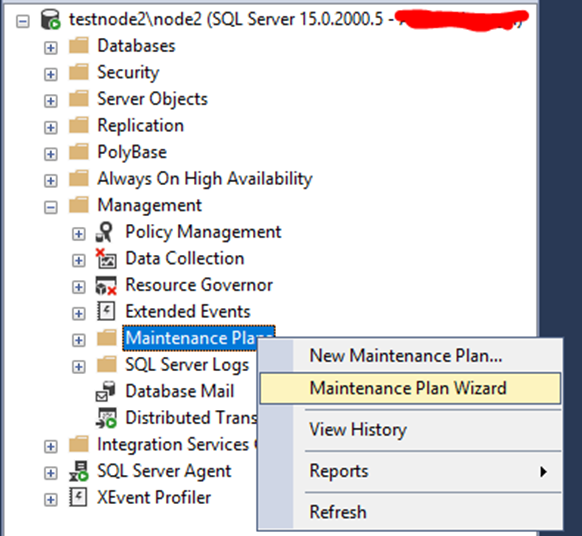


Рис. 12 «Мастер создания плана обслуживания»

Далее нужно указать название и режим работы плана.

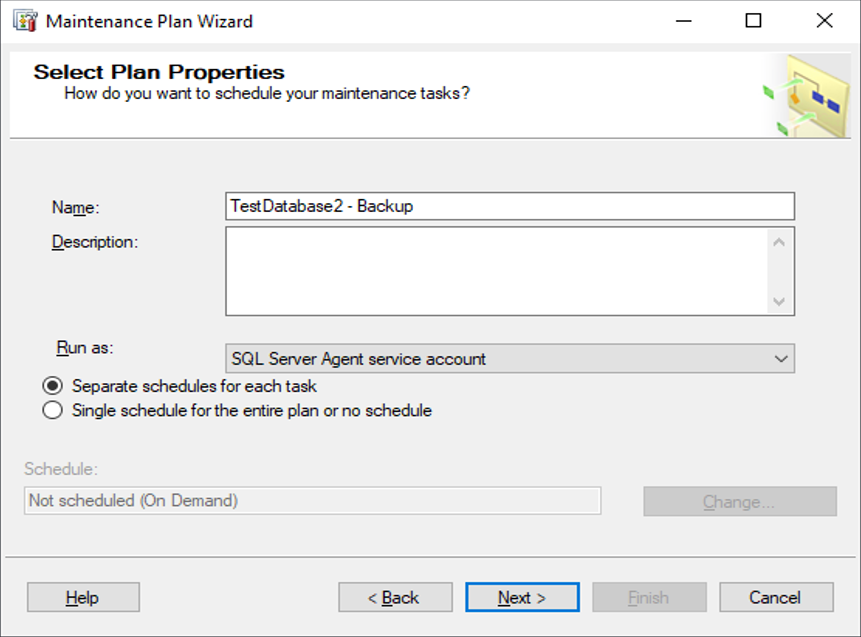


Рис.13 «Указание названия плана и режима работы»

Далее нужно выбрать операции, которые нужно сделать в этом плане обслуживания:

* Back Up Database (Full)
* Back Up Database (Transaction Log)

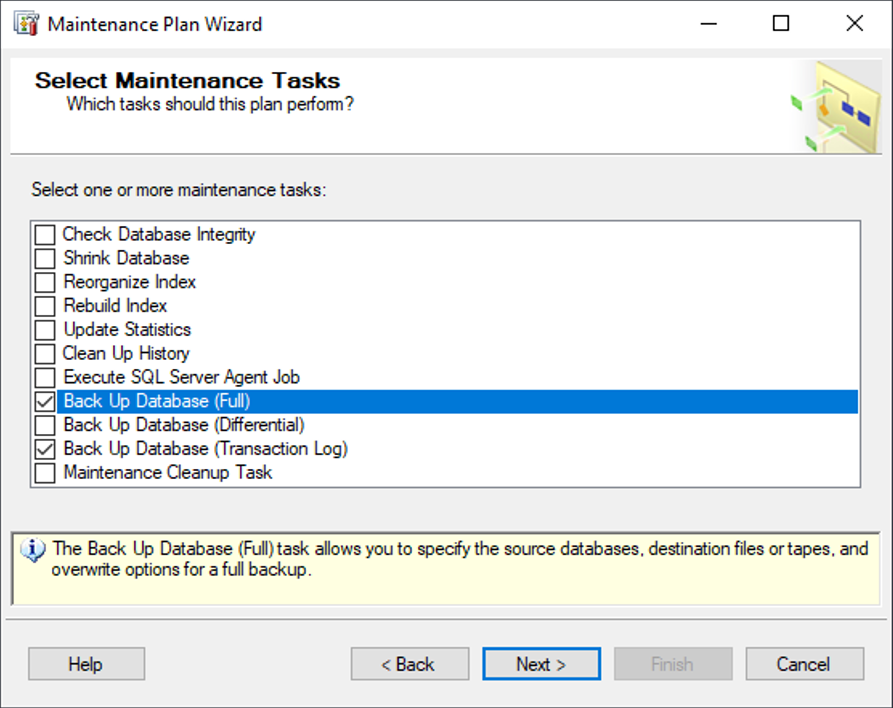


Рис. 14 «Выбор операций»

Дальше настраиваем последовательность операций в плане резервного копирования.

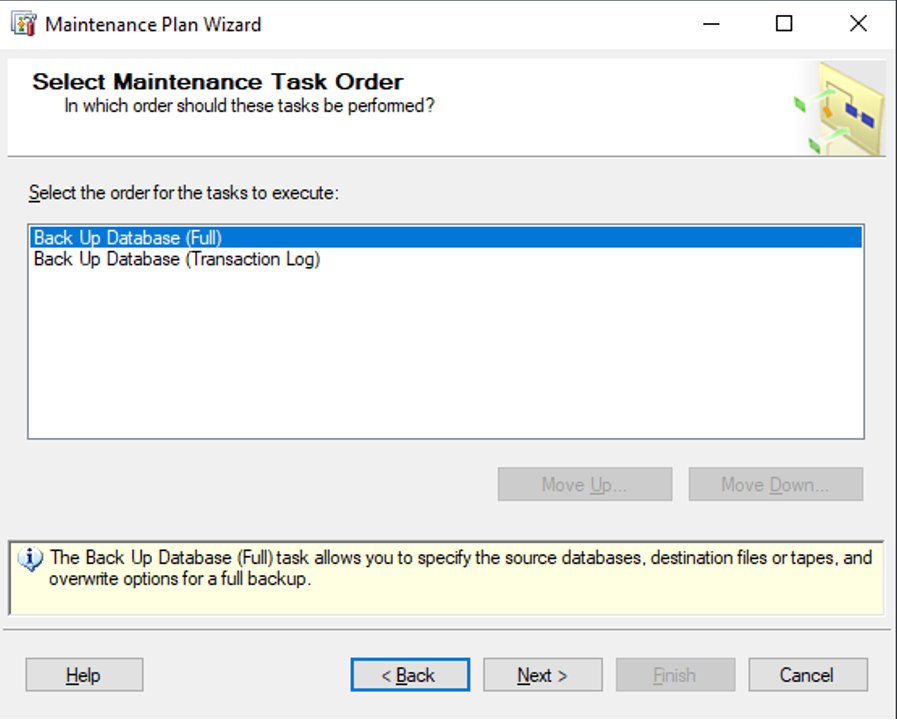


Рис. 15 «Последовательность операций в плане»

Далее выбираем базу данных для которой создаём план обслуживания

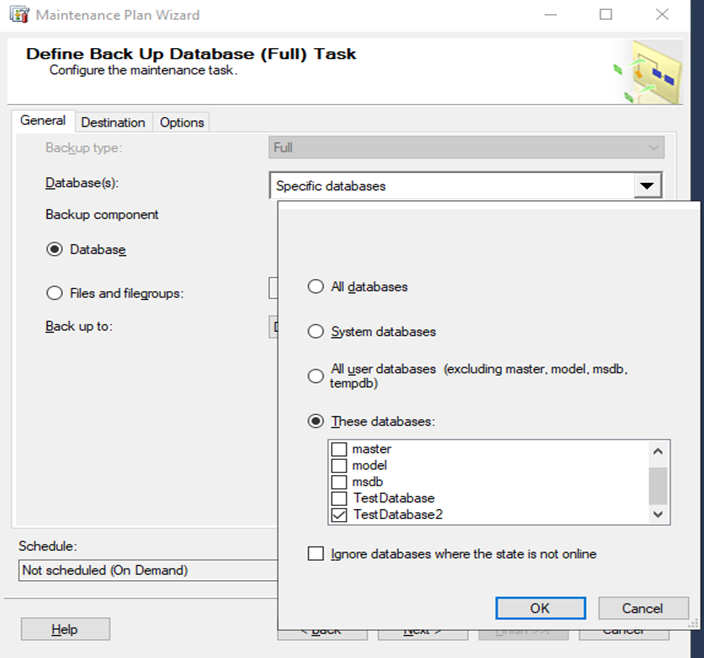


Рис. 16 «Выбор базы данных»

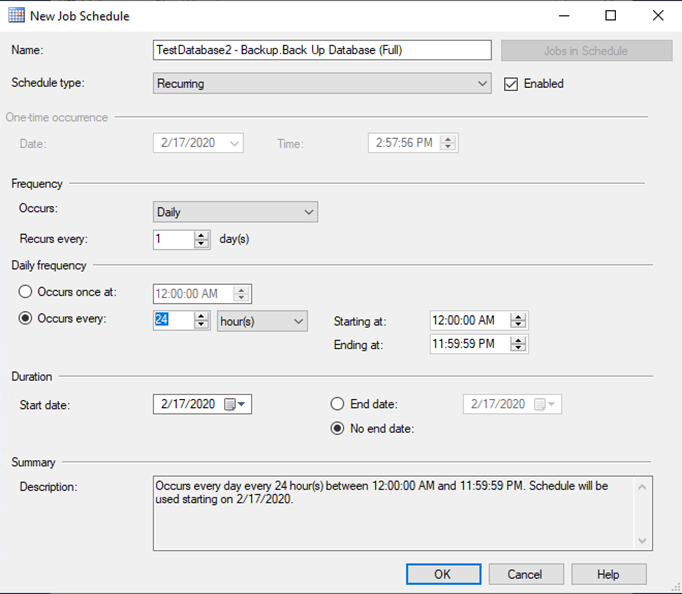


Рис. 17 «Выбор расписания»

Укажите путь к каталогу, в который нужно сохранять резервные копию ваше базы данных.

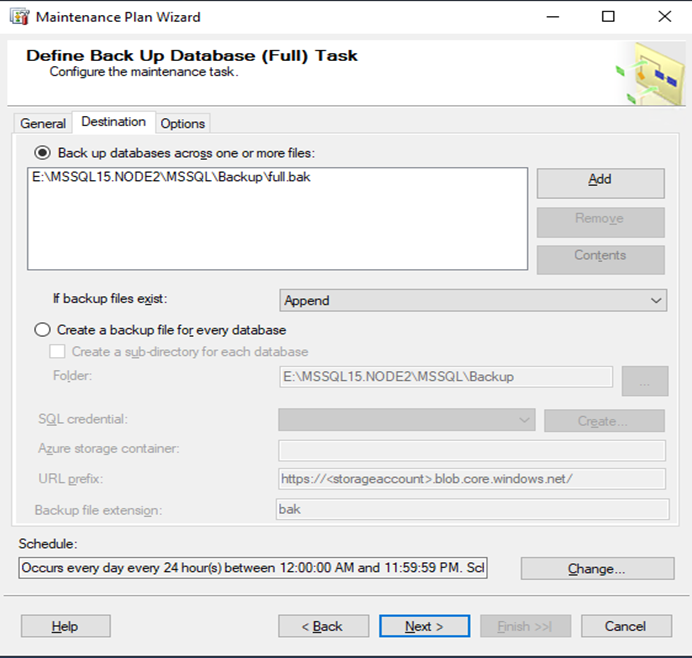


Рис. 18 «Указание пути резервного копирования»

Нажимаем «Далее» и аналогично настраиваем резервное копирование для журнала транзакций.

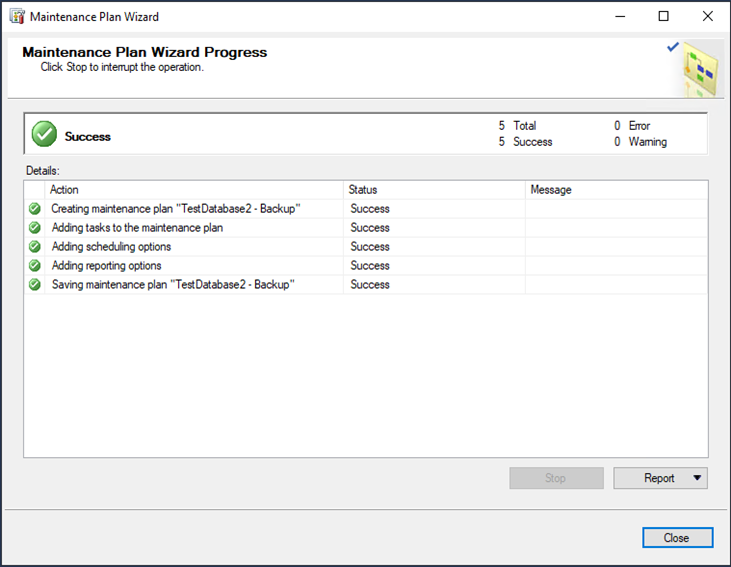


Рис. 19 «Успешное завершение создания плана обслуживания»

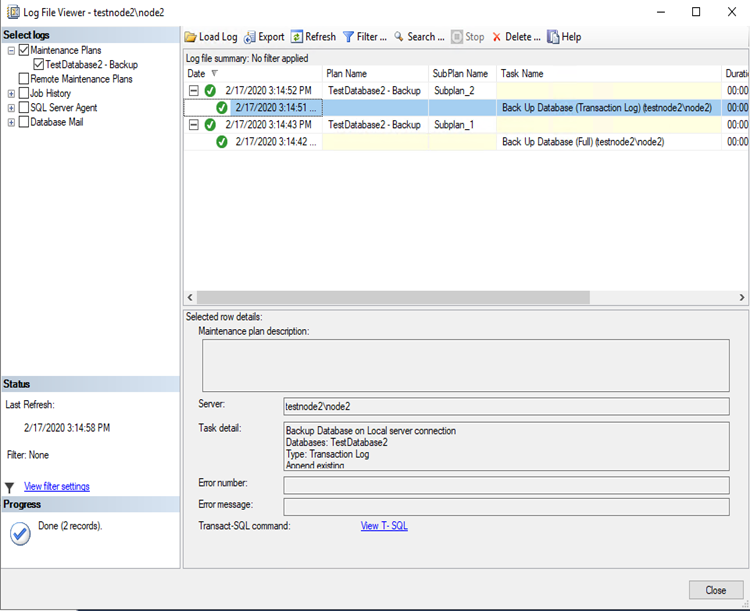


Рис. 20 «Просмотр журнала резервного копирования»

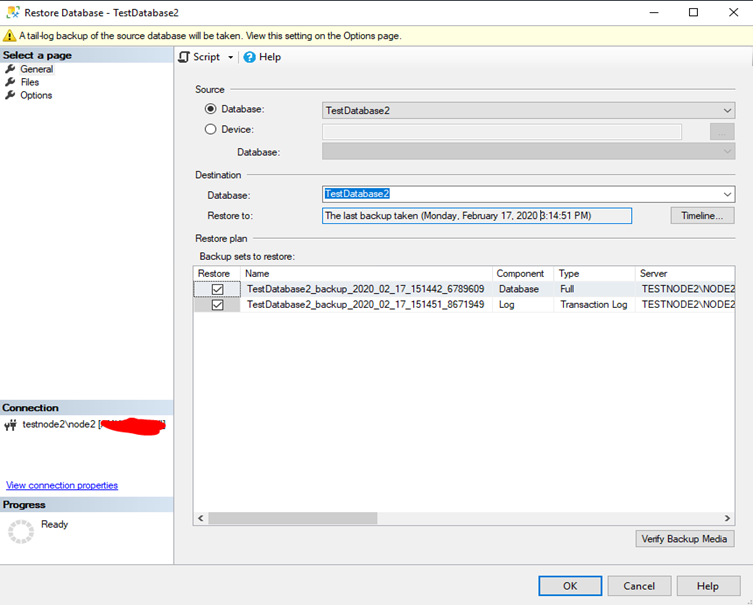


Рис. 21 «Просмотр журнала резервного копирования»

## 2.4. Настройка прав доступа для пользователей.

Для уже созданных пользователей мы можем сделать настройку этого пользователя, указать новый пароль, имя для входа, поменять права, убрать или добавить базы данных для взаимодействия. Сделать это можно во вкладке «безопасность» => «имена для входа». Выбираем существующего пользователя, и заходим в его свойства.

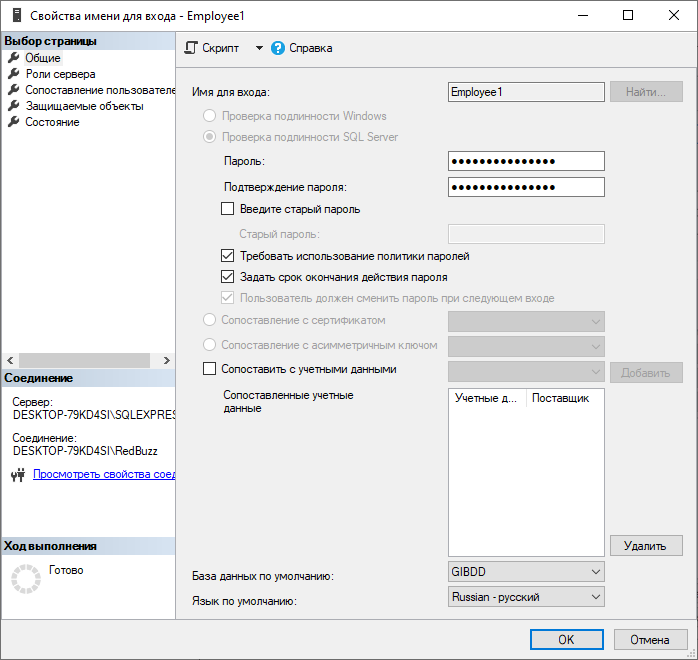


Рис. 22 «Данные о пользователе»

Добавим пользователю роль «dbcreator» для возможности взаимодействия с любыми базами данных.

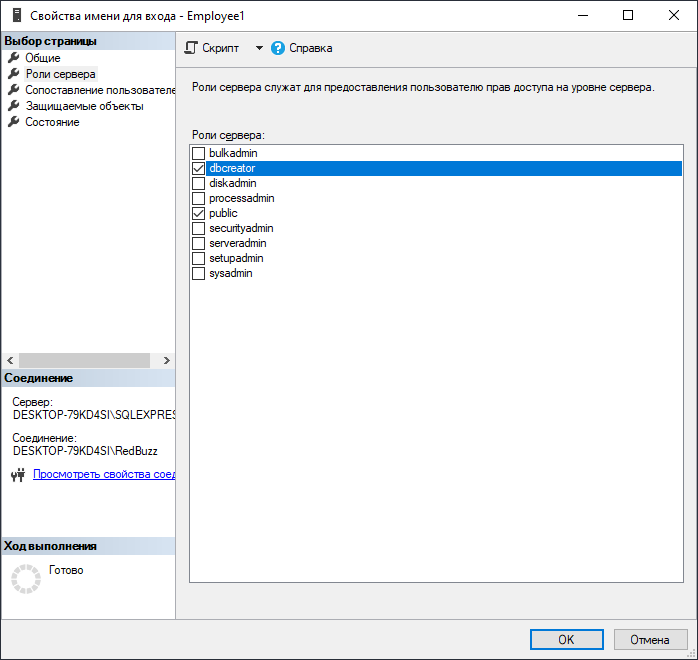


Рис. 23 «Добавление роли»

Также мы можем проставить роли для определённой базы данных. Для этого нужно нажать на вкладку «Сопоставления пользователя» и проставим ему роли для некоторых баз данных

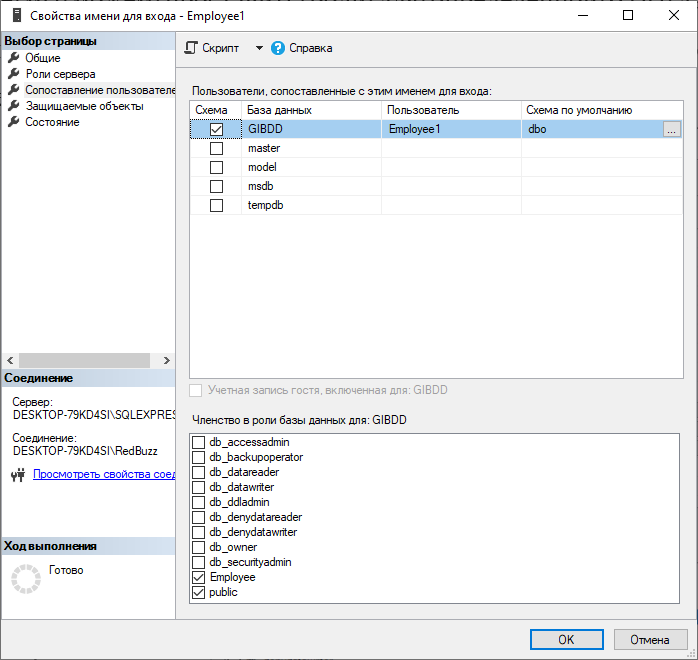


Рис. 24 «Добавление ролей в разных базах данных»

## 2.5 Создание триггеров авторизации.

Триггеры уровня сервера реагируют на серверные события. Триггер уровня сервера создается посредством использования предложения ALL SERVER в инструкции CREATE TRIGGER. В зависимости от выполняемого триггером действия, существует два разных типа триггеров уровня сервера: обычные триггеры DDL и триггеры входа. Запуск обычных триггеров DDL основан на событиях инструкций DDL, а запуск триггеров входа - на событиях входа.[13]

Создадим триггер уровня сервера, который является триггером входа:



Рис. 25 «Выдача разрешения для имени входа»

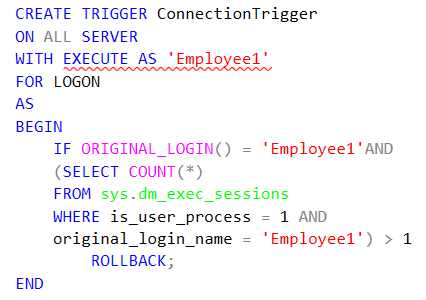


Рис. 26 «Триггер авторизации»

Для этого имени входа «Employee1» требуется разрешение VIEW SERVER STATE, которое и предоставляется ему посредством инструкции GRANT. Далее создается триггер «ConnectionTrigger», который является триггером входа, что указывается ключевым словом LOGON.

С помощью представления «sys.dm\_exec\_sessions» выполняется проверка, был ли уже установлен сеанс с использованием имени входа «Employee1». Если сеанс уже был установлен, выполняется инструкция ROLLBACK. Таким образом имя входа «Employee1» может одновременно установить только один сеанс.

# ЗАКЛЮЧЕНИЕ

Администрирование базами данных предусматривает выполнение функций, направленных на обеспечение надежного и эффективного функционирования системы баз данных, адекватности содержания базы данных информационным потребностям пользователей, отображения в базе данных актуального состояния предметной области.

Администратор БД отвечает за целостность информационных ресурсов компании. На нем лежит ответственность по созданию, обновлению и сохранности связанных между собой резервных копий файлов, исходя из задач предприятия. Этот человек должен в мельчайших подробностях знать существующие механизмы восстановления программного обеспечения БД.

Возможны ситуации, при которых администратору БД потребуется на основе логических прикладных моделей создавать элементы физической схемы, а также поддерживать связь пользователей с системой и обеспечивать соответствующий уровень информационной безопасности, следя за тем, чтобы доступ к данным имели только те люди, которые в нем нуждаются.

Администратор БД должен уметь определять узкие места системы, ограничивающие ее производительность, настраивать SQL и программное обеспечение СУБД и обладать знаниями, необходимыми для решения вопросов оптимизации быстродействия БД.

Администратор базы данных (АБД) должен координировать действия по сбору сведений, проектированию и эксплуатации базы данных, а также по обеспечению защиты данных. Он обязан учитывать текущие и перспективные информационные требования предметной области, что является одной из главных задач.

Правильная реализация функций администрирования базы данных существенно улучшает контроль и управление ресурсами данных предметной области. С этой точки зрения функции АБД являются больше управляющими, нежели техническими. Принципы работы АБД и его функции определяются подходом к данным как к ресурсам организации, поэтому решение проблем, связанных с администрированием, начинается с установления общих принципов эксплуатации СУБД.

# СПИСОК ЛИТЕРАТУРЫ

1. Грофф. Д.Р, Вайнберг. П.Н, Оппель.Э.Дж. – SQL. Полное руководство, 2017

Капрова Т. С. – Базы данных: модели, разработка, реализация: учебное пособие, 2016

1. Кузнецов. С. – Введение в реляционные базы данных. 2016
2. Кумскова И. – Базы данных, 2018
3. Мартин. Г – SQL для простых смертных. 2016
4. Мартин. Г – Введение в SQL, 2017
5. Медведкова И. Е., Бугаев Ю. В., Чикунов С. В.- Базы данных. 2018
6. Нестеров. С – Базы данных. Учебник и практикум, 2017
7. Петкович. Д. – Microsoft SQL Server. Руководство для начинающих. 2018
8. Сирант А.В, Коваленко Т.А. – Работа с базами данных. 2016
9. Шустрова Л, Воронкова О. – Базы данных учебник. 2017

**Интернет источники**

1. https://metanit.com/ (Metanit.com Сайт о программирования)
2. https://javarush.ru (javarush.ru Сайт по программированию)
3. https://npk-kaluga.ru/MakeQuery\_MSSQL.htm (Статья о sql server, запросы)
4. https://itvdn.com/ru/blog/article/m-sql (Статья о запросах sql server)
5. https://docs.microsoft.com/ru-ru/sql/t-sql/lesson-1-creating-database-objects?view=sql-server-ver15 (Официальный сайт Microsoft, урок запросов и создание баз данных)