

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт кибернетики и информационных технологий

Кафедра “Кибербезопасность, обработка и хранение информации”

Амангелді Бакдәулет Бауыржанұлы

Разработка и администрирование базы данных MS SQL Server

ДИПЛОМНЫЙ ПРОЕКТ


Специальность 5В100200 – Системы информационной безопасности

Алматы 2021

СӘТБАЕВ
УНИВЕРСИТЕТІ



КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени К.И. САТПАЕВА
ИНСТИТУТ КИБЕРНЕТИКИ И
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА КИБЕРБЕЗОПАСНОСТЬ,
ОБРАБОТКА И ХРАНЕНИЕ ИНФОРМАЦИИ

«Допущен к защите»
Заведующий кафедрой КОиХИ
канд. техн. наук, доцент
 Н.А.Сейлова
“ 25 ” 05 2021 г.

ДИПЛОМНЫЙ ПРОЕКТ

на тему: «Разработка и администрирование базы данных MS SQL Server»

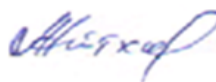
по образовательной программе 5В100200 – «Системы информационной безопасности»

Выполнил

Амангелді Б.Б.

Научный руководитель

к.т.н., ассоц. проф. Айтхожаева Е.Ж.



10.05.2021 г.

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева


Институт кибернетики и информационных технологий

Кафедра “Кибербезопасность, обработка и хранение информации”

5В100200 - Системы информационной безопасности

УТВЕРЖДАЮ

Заведующий кафедрой КОиХИ
канд. техн. наук, доцент

 Н.А.Сейлова
“ 25 ” 05 2021 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Амангелді Бақдәулет Бауыржанұлы

Тема: Разработка и администрирование базы данных MS SQL Server

Утверждена приказом Ректора Университета №2131-б от “24” 11.2020г.

Срок сдачи законченной работы “04” 05 2021г.

Исходные данные к дипломному проекту:

предметная область (Чемпионаты по футболу), сервер баз данных MS SQL Server, CASE-средство проектирования баз данных CA ERwin Modeling Suite, механизмы администрирования в MS SQL Server.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) обзор механизмов администрирования в MS SQL Server;

б) проектирование базы данных MS SQL Server в CASE-средстве проектирования ERwin;

в) организация защиты и безопасности базы данных;

г) приложения.

Перечень графического материала (с точным указанием обязательных чертежей):

механизмы администрирования в MS SQL Server – 1А3; проектирование логической и физической модели данных в Erwin – 1А3; схема БД в MS SQL Server – 1А3; представления и триггеры – 1А3; предоставление привилегий и создание резервной копии базы данных – 1А3.

Рекомендуемая основная литература: из 10 наименований


ГРАФИК

подготовки дипломного проекта

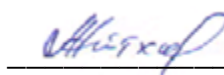
Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Администрирование серверных баз данных	01.03.2021 г.	Выполнено
Разработка серверной базы данных MS SQL Server	26.03.2021 г.	Выполнено
Администрирование базы данных Чемпионаты по футболу	26.04.2021 г.	Выполнено

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

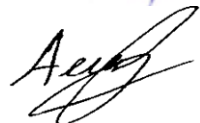
Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	подпись
Нормоконтроль	магистр техн.наук, ассистент Кабдуллин М.А.	15.05.2021	

Научный руководитель



Айтхожаева Е.Ж.

Задание принял к исполнению обучающийся



Амангелді Б.Б.

Дата

“24” 11. 2020г.

АНДАТПА

Бұл дипломдық жоба MS SQL Server мәліметтер базасын әзірлеуге және басқаруға арналған.

Әкімшінің негізгі міндеттері анықталды, ол мәліметтер базасының жұмысын сүйемелдеу мен оңтайландырудан бастап, бақылау мен тексеруден басталды. Сақтық көшірме жасау стратегиясы, импорттау мен экспорттау, қауіпсіздік саясаты жүзеге асырылды.

«Футбол чемпионаттары» тақырыбына талдау жүргізілді. Осы талдау негізінде объектілер, атрибуттар және олардың арасындағы қатынастар анықталды, ER диаграммалары логикалық және физикалық деңгейде Erwin CASE мәліметтер базасын жобалау құралында құрылды. MS SQL Server ДҚБЖ-да мәліметтер қорын енгізу үшін SQL сценарийлері құрылды. Деректер базасы MS SQL Server 2016-да енгізілген.

Деректер қорының қауіпсіздігін қамтамасыз ету үшін триггерлер мен көріністер құрылды. Көріністер пайдаланушының жайлы тәжірибесін қамтамасыз етуге және физикалық кестелерді қорғауға арналған. Триггерлер мәліметтер базасындағы өзгерістерді бақылау үшін жасалған және енгізілген.

Артықшылық беру механизмі қарастырылған, оны қолдану көрсетілген. Сондай-ақ мәліметтер базасының нысандарын резервтік көшіру және мәліметтер базасын қалпына келтіру жұмыстары жүргізілді.

АННОТАЦИЯ

Данный дипломный проект посвящен разработке и администрированию базы данных MS SQL Server.

Были определены основные задачи администратора, начиная с обслуживания и оптимизации работы баз данных, заканчивая мониторингом и аудитом. Также определены стратегия резервного копирования, импорт и экспорт данных, была реализована политика безопасности.

Был проведен анализ предметной области «Чемпионаты по футболу». На основе этого анализа были определены сущности, атрибуты и связи между ними, построены ER-диаграммы в CASE-средстве проектирования баз данных Erwin на логическом и физическом уровне. Сгенерированы SQL скрипты для реализации базы данных в СУБД MS SQL Server. База данных была реализована в MS SQL Server 2016.

Для обеспечения безопасности БД были созданы триггеры и представления. Представления созданы с целью обеспечения комфортности работы пользователей и защиты физических таблиц. Проектируются и реализуются триггеры для мониторинга изменений в базе данных.

Рассмотрен механизм предоставления привилегий, показано его применение. А также было выполнено резервное копирование объектов БД и восстановление БД.

ANNOTATION

This diploma project is dedicated to the development and administration of the MS SQL Server database.

The main tasks of the administrator were identified, starting with maintenance and optimization of the operation of databases, ending with monitoring and auditing. Also defined backup strategy, data import and export, security policy was implemented.

The analysis of the subject area "Football Championships" was carried out. Based on this analysis, entities, attributes and relationships between them were identified, ER diagrams were built in the Erwin CASE database design tool at the logical and physical levels. Generated SQL scripts for database implementation in MS SQL Server DBMS. The database was implemented in MS SQL Server 2016.

To ensure the security of the database, triggers and views were created. Views are designed to provide a comfortable user experience and protect physical tables. Triggers are designed and implemented to monitor changes in the database.

The mechanism of granting privileges is considered, its application is shown. And also backup copying of database objects and database restoration was performed.

СОДЕРЖАНИЕ

Введение	8
1 Администрирование серверных баз данных	9
1.1 Обеспечение защиты и безопасности	9
1.2 Механизмы администрирования	11
2 Разработка серверной базы данных MS SQL Server	15
2.1 ER-проектирование баз данных	15
2.2 Проектирование логической модели данных	15
2.3 Проектирование физической модели данных	17
2.4 Реализация базы данных в MS SQL Server	20
2.5 Разработка представлений	22
3 Администрирование базы данных для предметной области «Чемпионаты по футболу»	24
3.1 Создание триггеров администрирования	24
3.2 Управление привилегиями	27
3.3 Резервное копирование и восстановление данных	29
Заключение	32
Список использованной литературы	33
Приложение А	34
Приложение Б	38

ВВЕДЕНИЕ

В современном мире базы данных являются многофункциональными программными системами. В настоящее время они доступны для использования в деловой сфере, в которых хранятся сведения, необходимые для работы предприятия. Такие базы управляются администраторами баз данных с помощью специальных систем управления базами данных (СУБД). СУБД - это программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных. Она также обеспечивает защиту данных, хранение и взаимодействие пользователя с БД. SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире.

SQL Server - мощная система управления базами данных. В дополнение к обычным функциям СУБД SQL Server включает ряд встроенных служб интеллектуального анализа данных. Все приложения, разработанные в .Net и Visual Studio, а также приложения в пакете Microsoft Office, могут получать доступ к данным, хранящимся в SQL Server. SQL Server обеспечивает высокую масштабируемость, производительность и безопасность [1].

Администрирование осуществляется с использованием функций, обеспечивающих безопасную и надежную работу системы баз данных, соответствие содержимого базы данных информационным потребностям пользователя и фактическое указание состояния предметной области в базе данных. Исходя из этого, вам потребуются кадры для управления базой данных. MS SQL SERVER предоставляет администраторам баз данных единую консоль управления для мониторинга, управления и настройки всех баз данных и связанных служб на предприятии.

Целью дипломной работы является разработка и администрирование базы данных MS SQL SERVER, решение задач администратора, начиная с этапа разработки и корректного проектирования и включая использование механизмов обеспечения безопасности сервера MS SQL SERVER.

1 Администрирование серверных баз данных

1.1 Обеспечение защиты и безопасности

Главным требованием для администратора баз данных является обеспечение защиты и безопасности, потому что надежная защита данных имеет очень важное значение в каждом приложении. Система безопасности баз данных бывают на уровне сервера и на уровне баз данных. На уровне сервера разрешается либо отклоняется доступ пользователей к самому серверу. На уровне базы данных пользователи, которые имеют доступ на уровне сервера, могут иметь доступ к объектам базы данных. С таким подходом пользователи могут более гибко управлять доступом к базам данных[2].

Система безопасности на уровне сервера использует следующие понятия:

- аутентификация (authentication);
- учетная запись (login);
- встроенные роли сервера (fixed server roles).

На уровне базы данных используются понятия:

- пользователь базы данных (database user);
- встроенная роль базы данных (fixed database role);
- пользовательская роль базы данных (users database role);
- роль приложения (application role).

В сервере SQL Server имеются две группы ролей:

- роли сервера (server role);
- роли базы данных (database role).

Реализация безопасности для SQL Server обычно начинается на уровне сервера, где пользователи проходят проверку подлинности в базе данных входа в систему и настраиваются с ролями уровня сервера для помощи в управлении разрешениями.

После создания имени входа необходимо убедиться, что хотя бы один логин может получить доступ к базе данных. Доступ к базе данных для входа в систему предоставляется путем создания пользователя базы данных.

После создания пользователя базы данных, который гарантирует доступ к базе данных и ролям, вы уже можете использовать разрешения для управления доступом пользователей к данным для устранения неполадок в базе данных. Большинство баз данных имеют более детальные требования к безопасности, чем фиксированные роли базы данных.

Роль администратора базы данных включает ряд основных задач, которые необходимо решить, чтобы эти базы данных оптимально хранились и постоянно поддерживались в согласованном состоянии, обеспечивали работу с высокой производительностью.

Ниже представлены основные задачи администрирования баз данных.

Подготовка базы данных и сервера базы данных. Это может включать установку и настройку экземпляра SQL Server на физическом или виртуальном сервере или создание современной виртуальной машины на основе стандартов образов, создание базы данных и распространение ее информации и файлов журнала на устройство хранения.

Хранение файлов и объектов базы данных. После того, как база данных построена и заполнена данными, она требует постоянного обслуживания и оптимизации для обеспечения оптимальной производительности. Это означает уменьшение фрагментации, которая возникает при добавлении и удалении записей, и сохранение файла данных правильного размера. Он обеспечивает альтернативную структуру для логических и физических данных.

Управление восстановлением в случае сбоя базы данных. Базы данных имеют решающее значение для деловых операций, поэтому главной задачей администратора является планирование соответствующей стратегии резервного копирования и восстановления для каждой базы данных, что позволит осуществить восстановление базы данных в случае сбоя, преднамеренного искажения данных, несанкционированного шифрования данных и других нарушений целостности и доступности. Администратор должен разбираться в методах резервирования и восстановления, знать плюсы и минусы каждого из них.

Импорт и экспорт данных. Данные часто передаются между системами, поэтому администраторам баз данных необходимо выполнять экспорт или же импорт данных.

Реализация политик безопасности. Серверы баз данных организации содержат данные, необходимые для работы бизнеса. Устранение нарушений безопасности может потребовать больших затрат и времени, а также потерять доверие клиентов. Чтобы защитить активы и снизить риски безопасности, администратор баз данных должны внедрить политики безопасности, которые гарантируют пользователям доступ к нужным им данным, соблюдая при этом правовые нормы бизнеса.

Мониторинг и устранение неполадок систем баз данных. Администратор баз данных должны уметь определять узкие места в своих системах, которые ограничивают их производительность, и должны обладать необходимыми знаниями для устранения проблем с оптимизацией производительности баз данных. Он должен уметь отслеживать подозрительные события, чтобы решить проблему.

Для решения задач администрирования администратор должен уметь пользоваться встроенными механизмами администрирования и обеспечения безопасности БД. Любой сервер БД, в том числе и MS SQL Server, имеет встроенные механизмы администрирования, которые включают в себя различные средства. MS SQL Server 2016 имеет утилиту SQL Server Management Studio, которая может быть использована для выполнения большей части основных задач администрирования SQL Server.

1.2 Механизмы администрирования

Для решения задач **обслуживания и оптимизации работы** баз данных, администратор баз данных работающий с Microsoft SQL Server, обязан управлять базами данных и хранением данных. По этой причине необходимо понимать, как формировать базы данных, как хранятся сведения в базах данных, как регулировать физические файлы базы данных.

В MS SQL Server выделяют три типа файлов базы данных:

- первичный файл данных (primary file), является основным, в нем хранится служебная информация и данные, является отправной точкой баз данных с расширением mdf (только один файл);

- вторичные файлы данных (secondary file). Это файлы, к которым относятся все файлы данных, за исключением первичного файла. В них хранятся только данные пользователей. Файлы secondary file имеют расширение ndf;

- файлы журнала транзакций (transaction log file), содержат всю информацию о транзакциях, с расширением ldf.

Файлы данных имеют страничную организацию. На уровне файла базы данных основной единицей хранения данных является страница. Каждая страница файла базы данных составляет 8 КБ. Страницы объединяются в экстенты. Объем составляет 8 страниц (64 КБ) [3].

Страница имеет служебную область – заголовок (96 байт), в котором хранится системная информация: тип страницы, свободное место на странице и идентификационный номер таблицы.

Резервное копирование и восстановление баз данных.

Администратор баз данных разрабатывает стратегию резервного копирования баз данных для обеспечения надежности БД с использованием полной копии, разностной копии или копий журнала транзакций. Наличие резервных копий БД позволяет восстановить БД после случайных или преднамеренных сбоев, а также после атак вирусов-шифровальщиков.

Полная резервная копия позволяет полностью восстановить БД на момент завершения резервного копирования. Также при полной копии система может отслеживать изменения. Изменения отслеживаются на уровне страниц. В каждой странице есть флаг архивирования, который сбрасывается при создании полной копии и устанавливается, если данные на странице были изменены.

Разностное, или дифференциальное резервное копирование предназначено для архивации данных, которые изменились со времени последнего резервного копирования. При резервном копировании сохраняются только изменения, поэтому архивация занимает меньше времени.

Резервное копирование журнала транзакций позволяет сохранить информацию обо всех транзакциях, которые были выполнены в базе данных. В отличие от полных или разностных архивов, в архиве журнала транзакций записано состояние журнала на момент начала операции резервного копирования.

Импорт и экспорт данных. Большой объем данных в системе Microsoft SQL Server вводится пользователем непосредственно в приложение, но часто данные необходимо импортировать или экспортировать. SQL Server поддерживает массовый экспорт данных из таблиц SQL Server и массовый импорт данных в таблицу или представление.

Для выполнения этих задач SQL Server предоставляет набор инструментов. Ниже представлены основные средства для выполнения операций импорта/экспорта данных.

Программа Bcp является программой командной строки (Bcp.exe), которая импортирует и экспортирует данные, а также создает файлы форматирования.

Инструкции BULK INSERT – это инструкции T-SQL, которые импортируют данные из файла данных в таблицу базы данных.

Функция OPENROWSET – это табличная функция, для массового импорта данных в таблицу. Эту функцию можно использовать для подключения и извлечения данных из источников данных OLE DB. Полная информация о том, как подключиться к источнику данных, указывается в параметрах функции. SQL Server предлагает специальный поставщик OLE DB, именуемый BULK, который можно использовать с помощью функции OPENROWSET.

Мастер импорта и экспорта. Мастер создает простой пакет для импорта и экспорта данных в различных форматах, включая базы данных, электронные таблицы и текстовые файлы.

Реализация политики безопасности. В Microsoft SQL Server существует два типа безопасности: безопасность SQL Server и надежная проверка подлинности. Безопасность SQL Server - это стандартная комбинация имени пользователя для имени входа и пароля, доверенная проверка подлинности предполагает, что устройство, пытающееся подключиться к SQL Server, авторизовано процедурой проверки подлинности имени домена, и это подтверждение передается в сервер SQL. И предполагается что домен, на котором расположен экземпляр SQL Server, доверяет учетной записи пользователя, в котором проверка уже выполнялась ранее. [4, 5]

Пользователи и приложения, пытающиеся подключиться к SQL Server через механизм безопасности SQL Server, должны предоставить имя пользователя и пароль. При подключении с использованием доверенной проверки подлинности просто установите флаг в строке подключения, указывающий на использование доверенной проверки подлинности. В этом случае учетные данные конечного пользователя или учетной записи службы приложения передаются в качестве маркера безопасности. Доступ предоставляется, если у SQL Server есть соответствующее имя пользователя.

В SQL Server применяется разграничение прав безопасности на базе учетных записей и ролей. Роль - это определенный набор прав, который можно назначить определенному пользователю либо группе пользователей. Кроме этого можно создавать свои роли, для которых администратор может самостоятельно определить перечень разрешений и запретов. Созданные по умолчанию роли уровня сервера и роли уровня базы данных имеют предопределенный набор разрешений, и они не могут быть изменены.

Шифрование баз данных. Под безопасностью обычно понимают средства защиты от злоумышленников. При этом часто упускается из виду риск кражи съёмных носителей. Поэтому соблюдение политики безопасности требует использовать шифрование, как способ защита данных с помощью ключа или пароля. SQL Server может шифровать данные, процедуры и подключения к серверу. Шифрование возможно с использованием сертификатов, асимметричных или симметричных ключей. SQL Server поддерживает все известные криптографические алгоритмы. Windows Crypto API используется для реализации криптографических алгоритмов.

Наиболее распространенными типами шифрования являются прозрачное шифрование данных (TDE) и постоянное шифрование. Прозрачное шифрование данных полностью шифрует всю базу данных. Если физические носители или файлы украдены, злоумышленники не смогут получить доступ к информации в базе данных [5].

Always Encrypted позволяет хранить зашифрованные данные в SQL Server без отправки ключа шифрования на сам SQL Server. Always Encrypted шифруется на уровне столбца, а не на уровне базы данных, точно так же, как TDE шифрует данные в базе данных.

Мониторинг и аудит. Все серверы БД имеют специальные механизмы для выполнения аудита и мониторинга, чтобы узнать, насколько сервер баз данных в данный момент загружается, какие ресурсы доступны. Мониторинг вводится для наблюдения за работой ОС, ее служб, времени нахождения пользователя в системе, времени выполнения хранимых процедур и т.д. Все это можно узнать с помощью специальных средств мониторинга. К средствам

мониторинга относятся специальные встроенные утилиты, системные процедуры сервера баз данных. Самым распространенным из них является SQL Profiler.

SQL Profiler – это инструмент с графическим интерфейсом, который предназначен для мониторинга сервера и баз данных. SQL Server Profiler отслеживает события обработки ядра, например, начало пакета или транзакции и записывает данные об этих событиях в таблице SQL Server или в файле. При мониторинге SQL Profiler разделяет информацию на категории по типу событий. Создается шаблон трассировки, и на его основе создается файл трассировки. Набор зарегистрированных событий называется профилем трассировки (trace).

Помимо мониторинга SQL Server предоставляет возможность вести аудит любой пользовательской активности в экземпляре сервера. Аудит - это мощный инструмент, который позволяет полностью контролировать действия пользователей в базе данных. Часто используемым инструментом является SQL Audit, который позволяет производить аудит групп событий или отдельных событий на уровне сервера или баз данных.

Механизмы администрирования БД приведены в Приложении Б на листе 1.

2 Разработка серверной базы данных MS SQL Server

2.1 ER-проектирование баз данных

При проектировании реляционных БД наибольшее распространение получил ER-метод, метод «сущность-связь» (Entity - Relationship), разработанный в 1976 г. Питером Ченом, в котором для представления информационной модели используются ER-диаграммы. ER-диаграммы – это представление информационных моделей. Они наглядны, понятны всем пользователям процесса проектирования. Также они содержат всю информацию о предметной области, достаточную для дальнейшего проектирования. В ER-методе основными терминами являются сущность, связь, атрибут [6].

Сущность определяется как некоторый объект, представляющий интерес для пользователя. Примерами сущностей могут быть книги, машины, банковские счета, колледжи, сотрудники или студенты, магазины и т.д.

Атрибуты предназначены для описания конкретной сущности. Каждая сущность имеет определенный набор атрибутов. Например, атрибутами, которые могут быть свойствами сущности Студент, являются номер зачетной книжки, имя, фамилия, дата рождения, специальность и т. д.

Связь представляет собой соединение между двумя или более экземплярами сущностей. Связь обычно выражается глаголом. Типичные примеры связей между двумя сущностями: команды ИГРАЮТ в матчах, игроки ЧИСЛЯТСЯ в командах, команды ИМЕЮТ тренеров. Важной характеристикой связи между двумя (или несколькими) сущностями является тип связи, которая может быть типа: 1:1 (один-к-одному), 1:М (один-к-многим), М:1 (многие – к – одному), М:М (многие – к – многим).

Для моделирования и проектирования баз данных используются CASE-средства проектирования БД, лидером среди которых является AllFusion Erwin Data Modeler r7.

2.2 Проектирование логической модели данных

После проведения анализа предметной области «Чемпионаты по футболу», была спроектирована ER-диаграмма базы данных, которая представлена на рисунке 2.1. При анализе предметной области были определены сущности с атрибутами, также связи между сущностями. В ER-диаграмме сущности представляются в виде прямоугольников, связи – в виде

ромбов, атрибуты – в виде овалов. Для поддержания ссылочной целостности связи между сущностями были определены с помощью первичных ключей (РК - Primary Key) и внешних ключей (FK - Foreign Key).

Ниже перечислены сущности и их атрибуты.

Сущность КОМАНДА (Код команды - РК, Название, Страна, Город, Год основания, Бюджет).

Сущность ИГРОКИ (Id игрока - РК, ФИО, Год рождения, Позиция, Гражданство, Зарплата, Код команды - FK).

Сущность ТРЕНЕР (Id тренера - РК, ФИО, Год рождения, Телефон, Гражданство, Зарплата, Код команды - FK).

Сущность ЧЕМПИОНАТ (Код чемпионата - РК, Вид чемпионата, Дата начала, Бюджет).

Сущность МАТЧИ (Код матча - РК, Очко 1, Очко 2, Время, Код чемпионата – FK, Код команды 1 – FK1, Код команды 2 – FK2).

Связи между сущностями: Команда СОДЕРЖИТ игроков (связь Один ко многим), команда ИМЕЕТ тренера (связь Один ко многим), команда ИГРАЕТ матчи (связь Один ко многим), матчи ПРОХОДЯТ в чемпионате (связь Многие к одному).

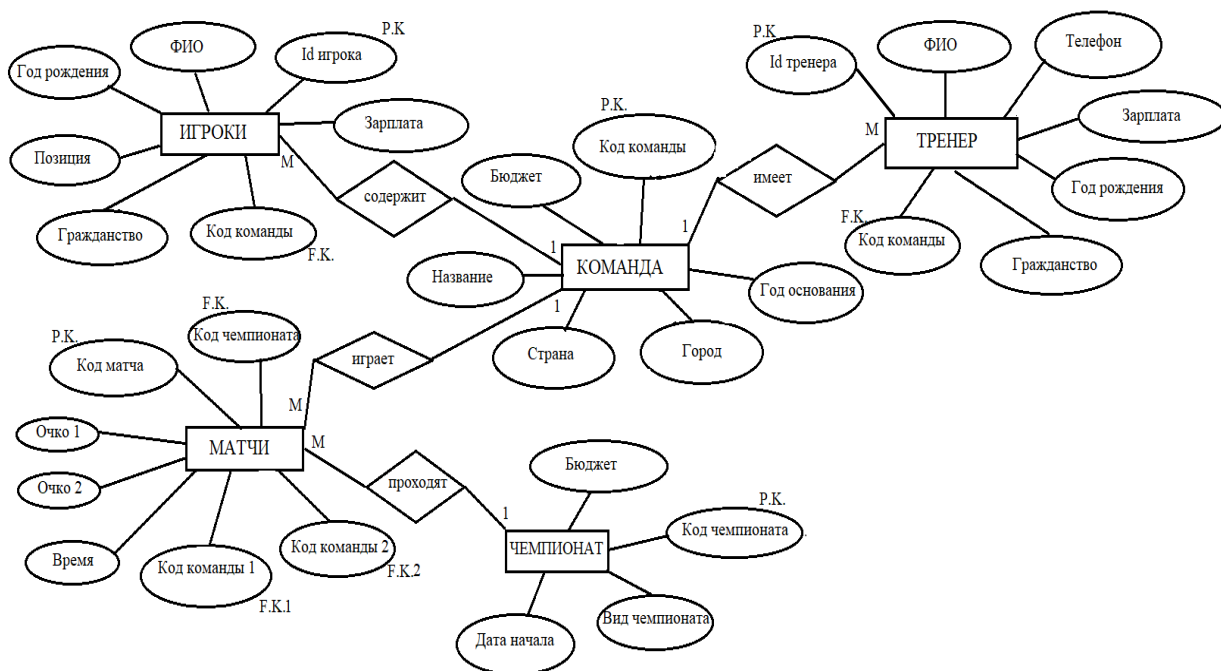


Рисунок 2.1 – ER-диаграмма

С использованием CASE-средства проектирования БД AllFusion Erwin Data Modeler r7 (Erwin) были созданы модели данных. Erwin создает визуальное представление – модель данных для решаемой проблемы. Сочетает в себе графический интерфейс Windows, инструмент построения диаграмм ER и редактор для создания логических и физических описаний моделей данных [7].

Логический уровень означает явное отображение фактов из реальной жизни. Логическая модель данных никак не связана с конкретной реализацией СУБД. На рисунке 2.2 представлена логическая модель баз данных, в котором были определены сущности и их атрибуты, связи между ними, типы атрибутов, а также первичные и внешние ключи.

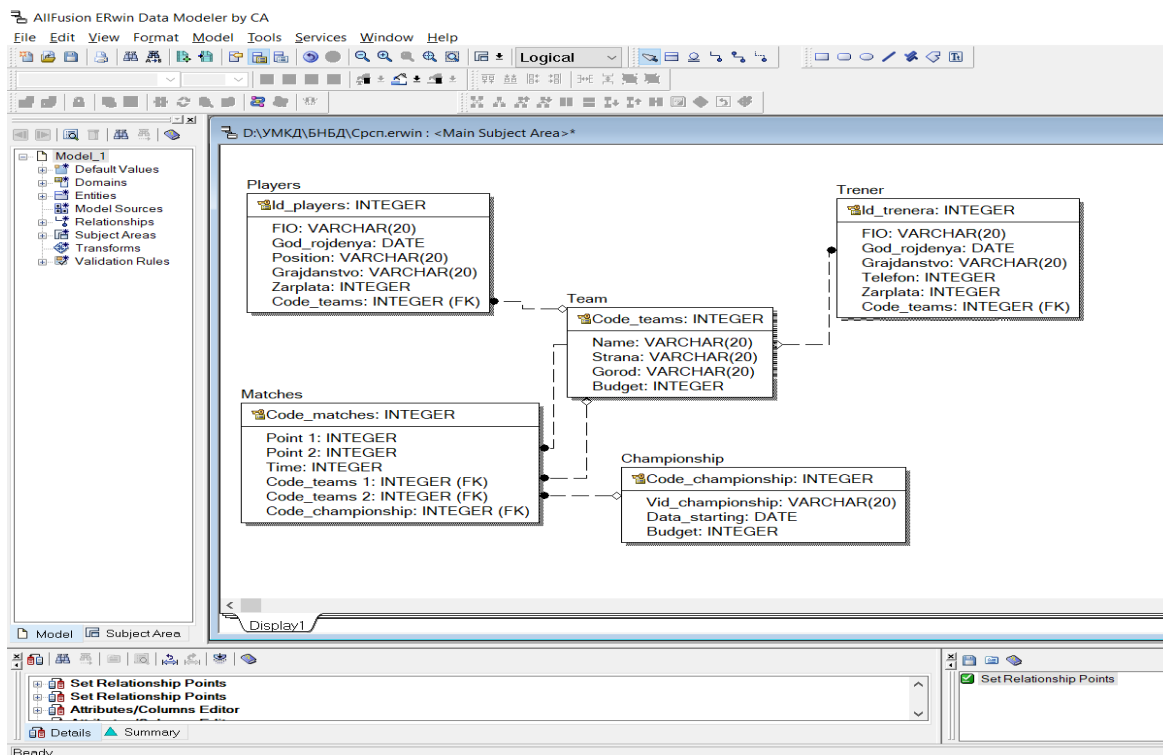


Рисунок 2.2 – Логическая модель баз данных

2.3 Проектирование физической модели данных

Физический уровень соответствует описанию схемы данных в конкретной СУБД. В физической модели содержится информация о всех объектах БД, также представлена вся информация о физических объектах - таблицах, колонках, индексах, процедурах и т. д. На рисунке 2.3 представлена физическая модель базы данных, в которой были определены типы данных и другие ограничения целостности данных.

Для поддержания целостности БД необходимо создать дополнительные ограничения целостности – CONSTRAINTS [8]. Ограничения задают правила допустимости определенных значений в столбцах и представляют собой стандартный механизм обеспечения целостности. Они могут быть заданы только на физическом уровне выбором вкладки CONSTRAINTS. На рисунке 2.4 представлено окно задания правила проверки, которое вызывается выбором вкладки Constraints в окне Columns при выделенном столбце, для которого необходимо задать ограничения целостности, т.е. ограничения на

значения по определенным правилам. А именно в таблице ИГРОКИ в столбце Позиция показано правило ограничения типа Valid Values List (Список допустимых значений), в котором определено то, что в позиции могут быть нападающий, полузащитник, защитник, вратарь. Ввод других значений запрещен (будет выдаваться сообщение об ошибке).

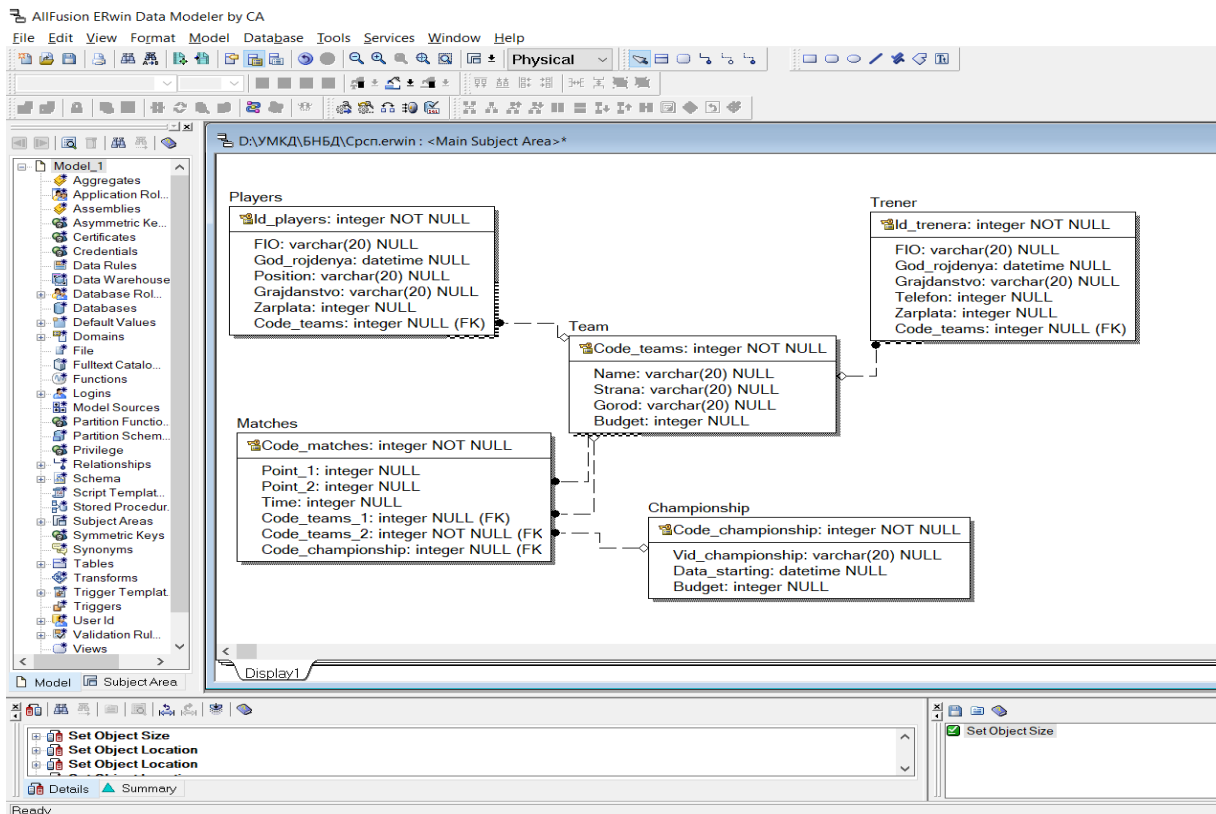


Рисунок 2.3 – Физическая модель баз данных

The screenshot shows the 'Validation Rules' dialog box. It contains a table of validation rules and configuration options.

Validation Name	Validation Rule
Position	@col IN ('Нападающий', 'Полузащитник', 'Защитник')
Time	@col BETWEEN 0 AND 90
Zarp	@col BETWEEN 100000 AND 1000000
Zarplata	@col BETWEEN 100000 AND 500000

Configuration for the 'Position' rule:

- Type**: Valid Values List (selected)
- Valid Value List**:

Valid Value	Display Value	Definition
Нападающий		
Полузащитник		
Защитник		
Вратарь		
- Physical Only**:
- Schema**: [Dropdown menu]

Рисунок 2.4 – Задание ограничения Valid Values List

Ниже на рисунке 2.5 представлено правило ограничения минимального и максимального значения (min/max) для столбца Зарплата в таблице ТРЕНЕР.

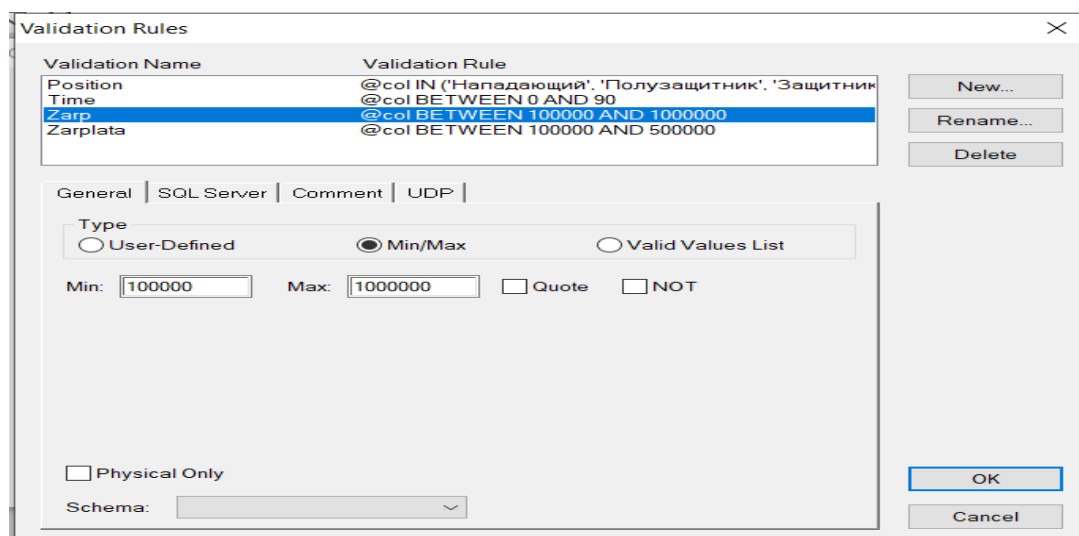


Рисунок 2.5 - Задание ограничения min/max

Ниже на рисунке 2.6 также представлено ограничение, но только по умолчанию. А именно была использована функция GETDATE (), которая определяет текущую дату и время для столбца Дата начало в таблице Чемпионат.

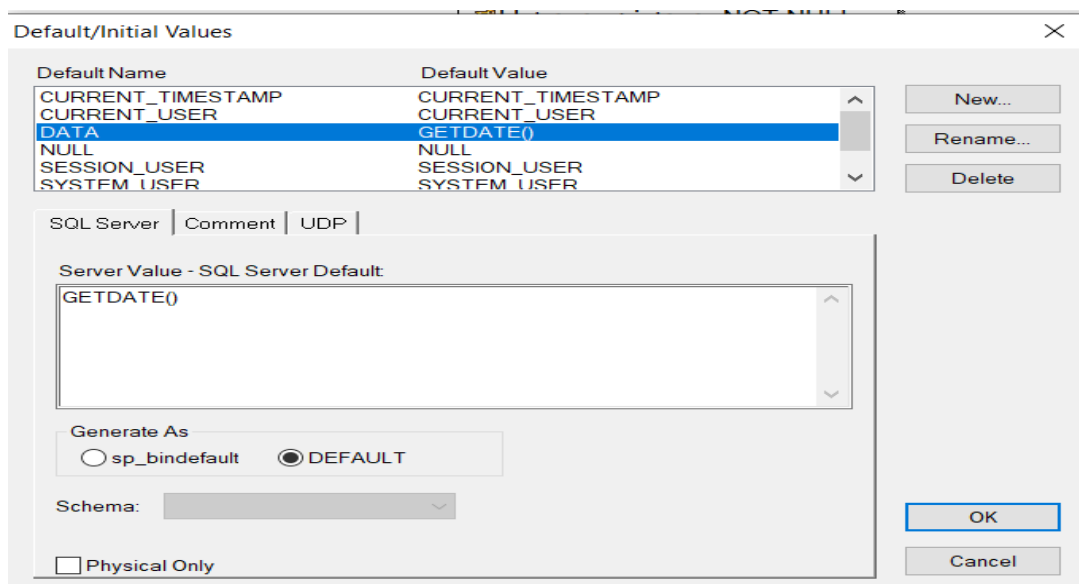


Рисунок 2.6 – Задание значения по умолчанию

Модели БД логического и физического уровня Erwin приведены в Приложении Б на листе 2.

2.4 Реализация базы данных в MS SQL Server

С использованием SQL скриптов, полученных в результате проектирования БД в Erwin, были созданы в БД Чемпионаты по футболу таблицы (Championship, Matches, Players, Team, Trener). SQL скрипты создания таблиц с ограничениями целостности приведены в Приложении А. Была создана схема БД, наглядно показывающая связи между таблицами БД, приведённая на рисунке 2.7 и в Приложении Б на листе 3.

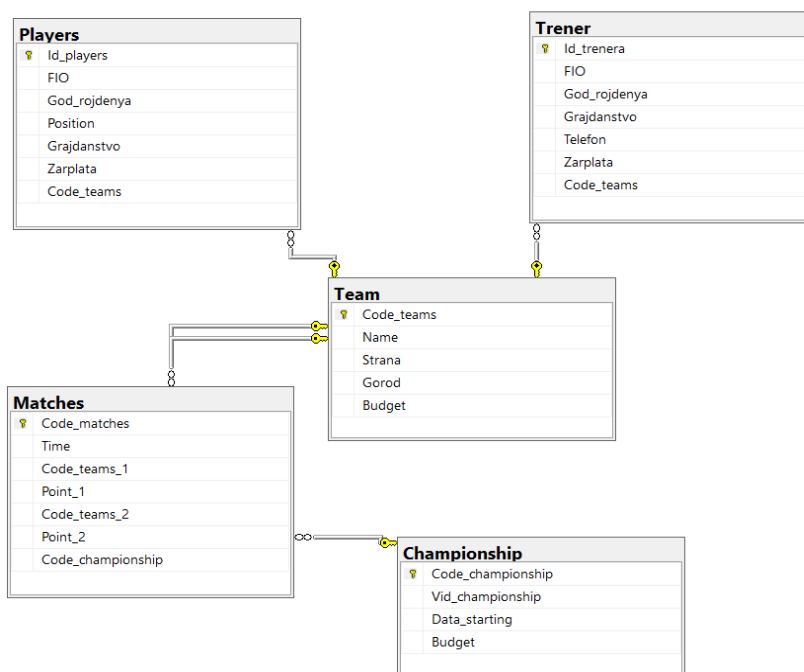


Рисунок 2.7 – Схема БД

Структуры таблиц с их свойствами приведены на рисунках 2.8-2.12 и в Приложении Б на листе 3.


Column Name	Data Type	Allow Nulls
 Code_championship	int	<input type="checkbox"/>
Vid_championship	varchar(20)	<input checked="" type="checkbox"/>
Data_starting	datetime	<input checked="" type="checkbox"/>
Budget	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Рисунок 2.8 – Структура таблицы Championship

Column Name	Data Type	Allow Nulls
Code_matches	int	<input type="checkbox"/>
Time	int	<input checked="" type="checkbox"/>
Code_teams_1	int	<input checked="" type="checkbox"/>
Point_1	int	<input checked="" type="checkbox"/>
Code_teams_2	int	<input type="checkbox"/>
Point_2	int	<input checked="" type="checkbox"/>
Code_championship	int	<input checked="" type="checkbox"/>

Рисунок 2.9 – Структура таблицы Matches

Column Name	Data Type	Allow Nulls
Id_players	int	<input type="checkbox"/>
FIO	varchar(20)	<input checked="" type="checkbox"/>
God_rojdenya	datetime	<input checked="" type="checkbox"/>
Position	varchar(20)	<input checked="" type="checkbox"/>
Grajdanstvo	varchar(20)	<input checked="" type="checkbox"/>
Zarplata	int	<input checked="" type="checkbox"/>
Code_teams	int	<input checked="" type="checkbox"/>

Рисунок 2.10 – Структура таблицы Players

Column Name	Data Type	Allow Nulls
Code_teams	int	<input type="checkbox"/>
Name	varchar(20)	<input checked="" type="checkbox"/>
Strana	varchar(20)	<input checked="" type="checkbox"/>
Gorod	varchar(20)	<input checked="" type="checkbox"/>
Budget	int	<input checked="" type="checkbox"/>

Рисунок 2.11 – Структура таблицы Team

Column Name	Data Type	Allow Nulls
Id_trenera	int	<input type="checkbox"/>
FIO	varchar(20)	<input checked="" type="checkbox"/>
God_rojdenya	datetime	<input checked="" type="checkbox"/>
Grajdanstvo	varchar(20)	<input checked="" type="checkbox"/>
Telefon	int	<input checked="" type="checkbox"/>
Zarplata	int	<input checked="" type="checkbox"/>
Code_teams	int	<input checked="" type="checkbox"/>

Рисунок 2.12 – Структура таблицы Trener

2.5 Разработка представлений

Представления данных (VIEW) представляют собой временные таблицы, в которых информация не хранится постоянно как в обычных стандартных таблицах, а формируется динамически при обращении к ним [9]. Они упрощают комплексные SQL-запросы. Также они защищают данные от пользователей, так как представления могут дать доступ к части таблицы, а не ко всей таблице. Представления также позволяют возвращать отформатированные значения из физических таблиц в подходящей и удобной форме. На рисунке 2.13 представлен скрипт создания представления Полузащитники и результат его работы, в котором был использован многотабличный запрос. Данные выбираются из таблиц Players и Teams, которые объединены через первичный и внешний ключи и выполнена фильтрация данных по позиции (только полузащитники).

```
CREATE VIEW Полузащитники
AS SELECT Players.FIO AS 'ФИО игрока',
Team.Name AS 'Название команды',
Team.Gorod AS 'Город',
Players.Position AS 'Позиция'
FROM Players, Team
WHERE Players.Code_teams = Team.Code_teams AND Players.Position = 'Полузащитник'
SELECT * FROM Полузащитники
```

ФИО игрока	Название команды	Город	Позиция
Henrih Mkhitaryan	Roma	Rome	Полузащитник
Jordan Henderson	Liverpool	Liverpool	Полузащитник
Mason Mount	Chelsea	London	Полузащитник
Bruno Fernandes	Manchester United	Manchester	Полузащитник
Ivan Rakitic	Sevilla	Sevilla	Полузащитник

Рисунок 2.13 – Скрипт создания представления Полузащитники

На рисунке 2.14 представлен скрипт создания представления Тренеры и результат его работы, в котором был использован запрос с подзапросом для формирования списка тренеров, у которых зарплата выше средней.

```
CREATE VIEW Тренеры
AS SELECT F1.FIO AS 'ФИО тренера',
F1.Grajdanstvo AS 'Гранжданство',
F1.Zarplata AS 'Зарплата'
FROM Trener F1
WHERE F1.Zarplata > (SELECT avg(F2.Zarplata) FROM Trener F2)
SELECT * FROM Тренеры
```

ФИО тренера	Гранжданство	Зарплата
Hans-Dieter Flick	Germany	870000
Thomas Tuchel	Germany	570000
Jurgen Klopp	Germany	750000
Ole Gunnar Solskjaer	Norway	650000
Diego Simeone	Argentina	550000
Andrea Pirlo	Italy	600000
Zinedine Zidane	France	700000
Ronald Koeman	Netherlands	800000

Рисунок 2.14 – Скрипт создания представления Тренеры

На рисунке 2.15 представлен скрипт создания представления Матчи и результат его работы. В этом представлении был использован многотабличный запрос, содержащий группировку данных, для получения информации о количестве матчей в каждом чемпионате.

```

Create view Матчи
As Select Championship.Vid_championship As 'Вид чемпионата',
Count(Matches.Code_championship) As 'Количество матч'
From Championship, Matches
Where Championship.Code_championship = Matches.Code_championship
Group by Vid_championship
Select*From Матчи
    
```

Вид чемпионата	Количество матч
1 Bundesliga	2
2 La liga	2
3 Premier League	2
4 Seria A	2

Рисунок 2.15 – Скрипт создания представления Матчи

Для удобства пользователей и защиты физических таблиц от пользователей было создано представление Результаты, которое представлено на рисунке 2.16. Представление создано на основе трех таблиц (Чемпионаты, Матчи, Команды) и позволяет просмотреть результаты игр всех чемпионатов с указанием очков команд, исключая доступ к реальным таблицам.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
Vid_champion...	[Вид че...	Champion...	<input checked="" type="checkbox"/>						
Code_matches	[Код м...	Matches	<input checked="" type="checkbox"/>						
Time	Время	Matches	<input checked="" type="checkbox"/>						
Name	Коман...	Team	<input checked="" type="checkbox"/>						
Point_1	Очко_1	Matches	<input checked="" type="checkbox"/>						

```

SELECT dbo.Championship.Vid_championship AS [Вид чемпионата], dbo.Matches.Code_matches AS [Код матча], dbo.Matches.Time AS Время, dbo.Team.Name AS Команда_1,
dbo.Matches.Point_1 AS Очко_1, Team_1.Name AS Команда_2, dbo.Matches.Point_2 AS Очко_2 FROM  dbo.Championship INNER JOIN
dbo.Matches ON dbo.Championship.Code_championship = dbo.Matches.Code_championship INNER JOIN
dbo.Team ON dbo.Matches.Code_teams_1 = dbo.Team.Code_teams INNER JOIN
dbo.Team AS Team_1 ON dbo.Matches.Code_teams_2 = Team_1.Code_teams
    
```

Вид чемпи...	Код матча	Время	Команда_1	Очко_1	Команда_2	Очко_2
La liga	103	18:00:00	Real Madrid	2	Sevilla	1
La liga	105	20:30:00	Atletico Ma...	0	Barcelona	1
Seria A	202	21:00:00	Juventus	1	Roma	0

Рисунок 2.16 – Создание представления Результаты

3 Администрирование базы данных для предметной области «Чемпионаты по футболу»

3.1 Создание триггеров администрирования

Триггеры представляют особый вид хранимой процедуры, которая вызывается автоматически при выполнении определенного действия на сервере баз данных. Триггеры бывают двух типов: DML и DDL триггеры. Триггеры DML срабатывают при добавлении, изменении или удалении строк в созданной таблице. А триггеры DDL срабатывают при выполнении команд, которые создают, изменяют или удаляют объекты базы данных.

Триггер DML - это особый тип хранимой процедуры, которая запускается автоматически, когда данные обрабатываются командами DML, которые влияют на таблицу или представление, определенные в триггере. Триггеры DML обрабатывают операторы INSERT, UPDATE и DELETE. Триггеры DML могут использоваться для обеспечения соблюдения бизнес-правил и правил целостности данных, запросов к другим таблицам и активации сложных операторов Transact-SQL. Когда запрос выполняется, срабатывает триггер. Это транзакция, которую можно откатить с помощью триггера. Когда сервер обнаруживает ошибку, вся транзакция автоматически откатывается [10].

Если в таблице триггеров есть ограничения, то их проверку можно выполнить с помощью триггеров AFTER и INSTEAD OF. Триггеры AFTER выполняются после выполнения действий инструкции INSERT, UPDATE, DELETE. Этот триггер задается с помощью ключевого слова AFTER или FOR. Также их можно создавать только для базовых таблиц. Триггеры INSTEAD OF срабатывают вместо запустивших их действий. На рисунке 3.1 представлен скрипт создания триггера tr_Matches для таблицы Матчи, который будет срабатывать при выполнении операций добавления, удаления, обновления строк для контроля выполняемых действий в таблице. Результат работы триггера представлен на рисунке 3.2. Для фиксации результата работы триггера предварительно была создана таблица Football_audit:

```
Create table Football_audit (  
  Audit_ID int identity(1, 1) primary key,  
  Table_name varchar(30),  
  Code int,  
  Updated_at Datetime,  
  Operation char(6),
```

Check (operation = 'Insert' or operation = 'Delete' or operation= 'Update')
)

```
CREATE TRIGGER tr_Matches on Matches
FOR INSERT, DELETE, UPDATE
As
declare @tablename varchar(100)
SELECT @tablename = OBJECT_NAME(parent_object_id)
FROM sys.objects
WHERE sys.objects.name = OBJECT_NAME(@@PROCID)
IF EXISTS ( SELECT 0 FROM Deleted )
BEGIN
IF EXISTS ( SELECT 0 FROM Inserted )
BEGIN
INSERT INTO Football_audit(
Table_name,Code,Updated_at, Operation)
Select @tablename, D.Code_matches, Getdate(), 'Update'
From Deleted D
END
ELSE
BEGIN
INSERT INTO Football_audit(
Table_name,Code,Updated_at, Operation)
Select @tablename, D.Code_matches, GetDate(), 'Delete'
```

Рисунок 3.1 - Создание триггера tr_Matches

```
Insert Matches values(418, '20:00', 414065, 2, 433101, 2, 4)
Select*From Football_audit
Delete From Matches where Code_matches = 418
Update Matches Set Code_matches = 103 Where Code_matches = 101
Select*From Matches
```

Audit_ID	Table_name	Code	Updated_at	Operation
1	Matches	418	2021-03-26 11:25:18.960	Insert
2	Matches	418	2021-03-26 11:25:22.567	Delete
3	Matches	101	2021-03-26 11:25:25.800	Update

Рисунок 3.2 – Результат работы триггера

Аналогично был создан триггер tr_Players для таблицы Игроки (на рисунке 3.3), который будет срабатывать при выполнении операций добавления, удаления, обновления строк для контроля выполняемых действий в таблице и фиксации этих действий в таблице Football_audit. Результат работы триггера представлен на рисунке 3.4.

```
CREATE TRIGGER tr_Players on Players
FOR INSERT, DELETE, UPDATE
As
declare @tablename varchar(100)
SELECT @tablename = OBJECT_NAME(parent_object_id)
FROM sys.objects
WHERE sys.objects.name = OBJECT_NAME(@@PROCID)
IF EXISTS ( SELECT 0 FROM Deleted )
BEGIN
IF EXISTS ( SELECT 0 FROM Inserted )
BEGIN
INSERT INTO Football_audit(
Table_name,Code,Updated_at, Operation)
Select @tablename, D.Id_players, Getdate(), 'Update'
From Deleted D
END
ELSE
BEGIN
INSERT INTO Football_audit(
Table_name,Code,Updated_at, Operation)
Select @tablename, D.Id_players, GetDate(), 'Delete'
```

Рисунок 3.3 – Создание триггера tr_Players

```

Insert Players values(999330154, 'Luka Modric', '1985.07.20',
'Полузащитник', 'Croatia', 270000, 123789)
Select*From Football_audit
Delete From Players where Id_players = 999330154
Update Players Set Zarplata = 130000 Where Id_players = 399431308
Select*From Players|

```

Audit_ID	Table_name	Code	Updated_at	Operation
1	Matches	418	2021-03-26 11:25:18.960	Insert
2	Matches	418	2021-03-26 11:25:22.567	Delete
3	Matches	101	2021-03-26 11:25:25.800	Update
4	Players	999330154	2021-03-26 11:38:26.977	Insert
5	Players	999330154	2021-03-26 11:38:38.230	Delete
6	Players	399431308	2021-03-26 11:39:02.920	Update

Рисунок 3.4 – Результат работы триггера

Триггеры DDL срабатывают в ответ на различные события DDL. Большинство этих событий соответствуют операторам Transact-SQL, которые начинаются с запроса CREATE, ALTER или DROP. Системные хранимые процедуры, которые выполняют операции, подобные DDL, также могут запускать триггеры DDL. Триггеры DDL срабатывают только после выполнения запроса DDL. Кроме того, его нельзя использовать в качестве триггера INSTEAD OF. И они не создают специальных таблиц вставки и удаления. Триггеры DDL срабатывают в ответ на события Transact-SQL, обрабатываемые текущей базой данных или текущим сервером. Информацию о событии, которое привело к срабатыванию триггера DDL, и следующих изменениях, которые были выполнены триггером, можно получить с помощью функции EVENTDATA [11]. На рисунке 3.5 представлен скрипт создания триггера tr_football и результат его работы. Триггер будет срабатывать при добавления таблиц в БД и фиксировать выполненные действия в таблице Audit.

```

CREATE TABLE Audit (PostTime datetime, Command varchar(100), Nazvaniya nvarchar(200));
CREATE TRIGGER tr_football
ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS
AS
DECLARE @data XML
SET @data = EVENTDATA()
INSERT Audit
(PostTime,Command,Nazvaniya)
VALUES
(GETDATE(),
@data.value('/EVENT_INSTANCE/EventType)[1]', 'varchar(100)'),
@data.value('/EVENT_INSTANCE/ObjectName)[1]', 'varchar(200)' ) ;
CREATE TABLE Table_Champion(a int)
DROP TABLE Table_Champion
SELECT*FROM Audit ; |

```

PostTime	Command	Nazvaniya
2021-03-17 12:45:21.930	CREATE_TABLE	Table_Champion
2021-03-17 12:45:30.730	DROP_TABLE	Table_Champion

Рисунок 3.5 – Создание триггера tr_football

3.2 Управление привилегиями

Администратор должен обеспечить безопасный многопользовательский доступ к БД, представляя разные привилегии различным пользователям. При этом используется встроенный механизм учетных записей и ролей, который и был использован для защиты конкретной базы данных.

В базе данных Championship football отражена информация, необходимая для учета команд, игроков, тренеров и сыгранных матчей в чемпионатах по футболу. Исходя из анализа предметной области, были определены группы пользователей и их полномочия, реализованы роли для разных групп пользователей, выданы ролям привилегии. Так как всех пользователей базы данных можно разбить на 3 группы, были определены 3 роли. Привилегии для ролей были назначены с ориентацией на функциональные обязанности членов ролей. Были созданы роли для директора (director), судей (sudya) и для зрителей (zritelj).

При создании роли директора (director) были выбраны защищаемые объекты и выданы привилегии для действий с этими объектами:

- таблица Команда (разрешение на выборки, обновления, удаления, вставки, просмотр отслеживания изменений);
- таблица Игроки (разрешение выборки, удаления, вставки, обновления, запрет изменения);
- таблица Тренер (разрешение выборки, удаления, вставки, обновления, запрет изменения).

На рисунке 3.6 показана выдача привилегий роли Director на таблицу Team.

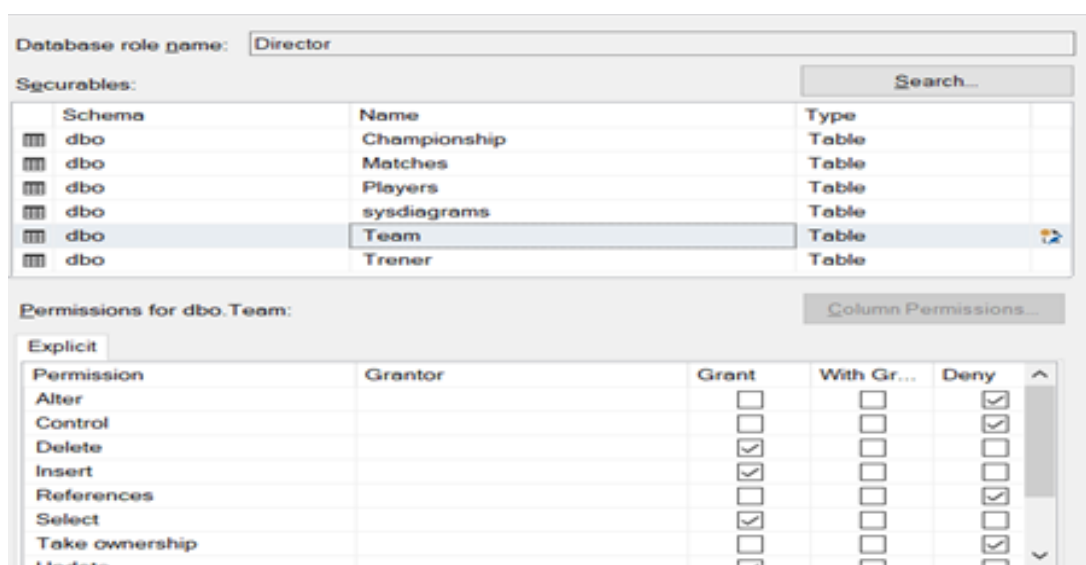


Рисунок 3.6 – Привилегии роли Director

Для членов роли Sudya были предоставлены привилегии на таблицу Матчи (разрешение на контроль, владения, выборку и просмотр отслеживания изменений).

Ниже на рисунке 3.7 показана выдача привилегий роли Sudya на таблицу Matches.

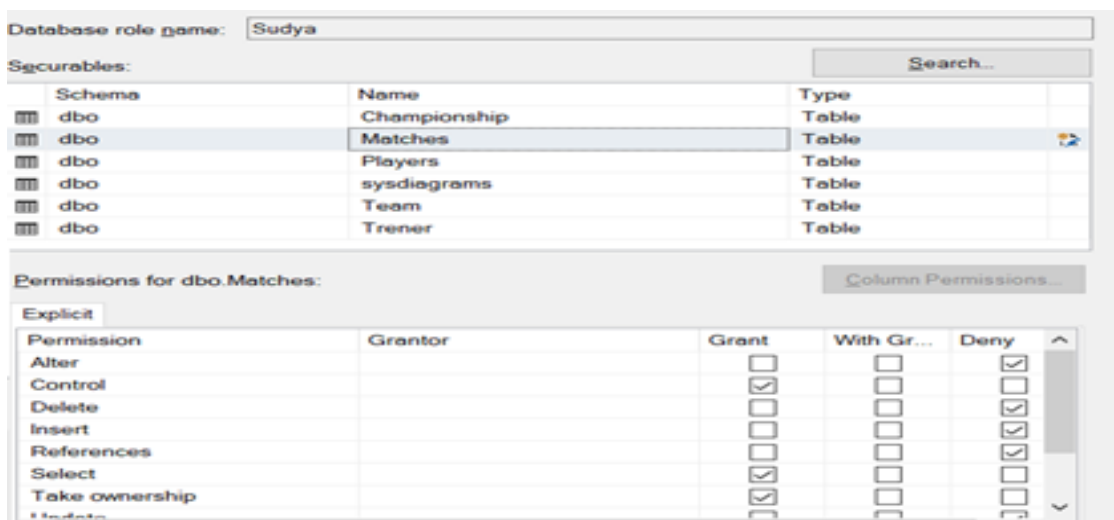


Рисунок 3.7 – Привилегии роли Sudya

Для роли зрителей (Zriteli) разрешается только просмотр всех таблиц. На рисунке 3.8 показана выдача привилегий роли Zriteli для таблицы Championship. И на рисунке 3.9 показана выдача привилегий на представление Результаты.

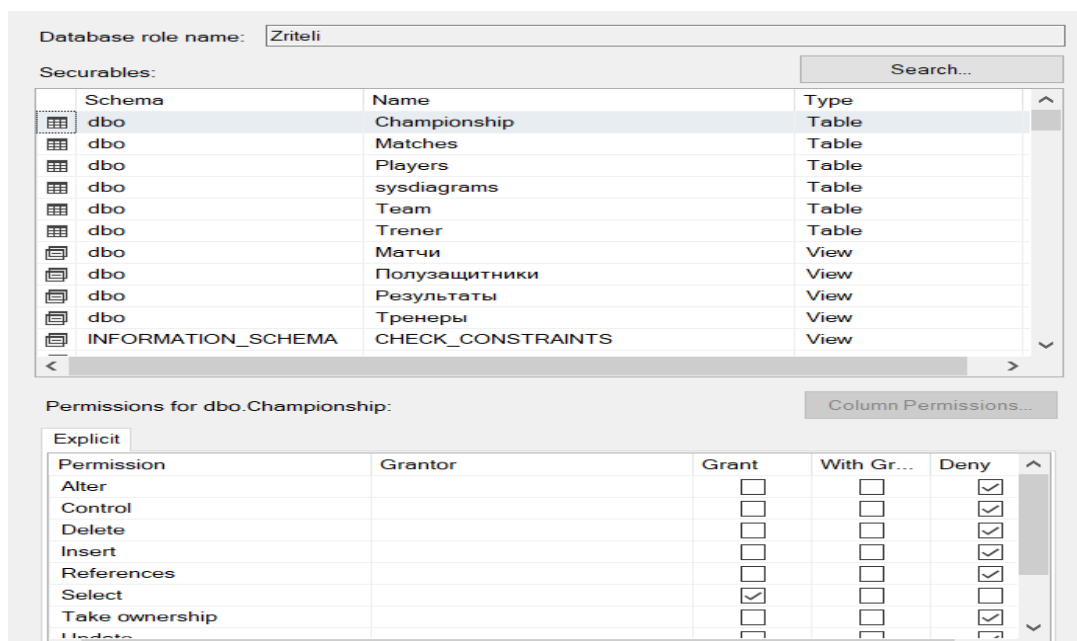


Рисунок 3.8 – Привилегии роли Zriteli на таблицу Championship

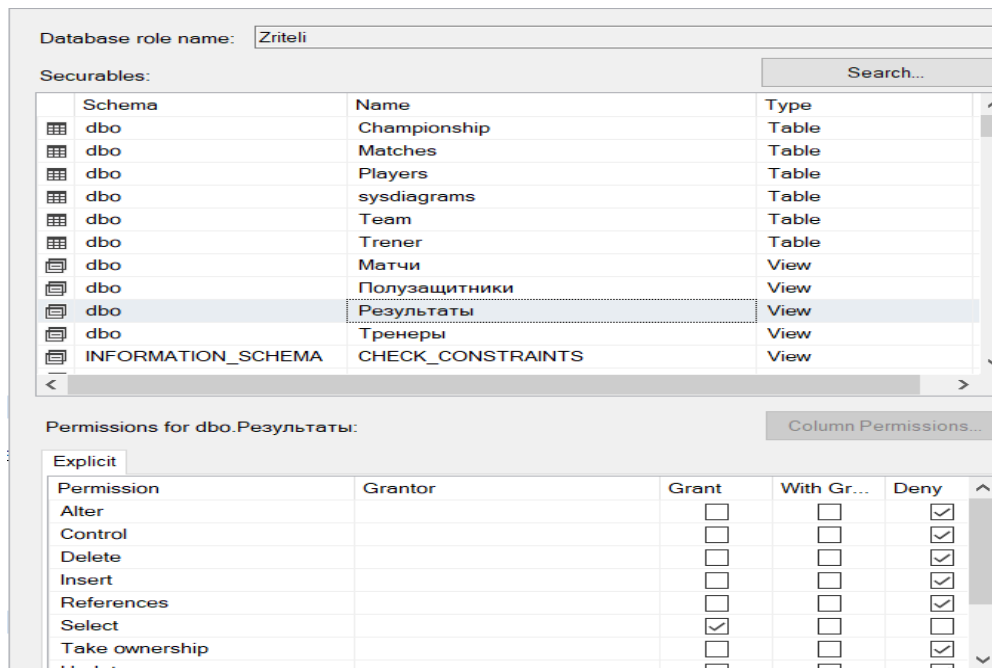


Рисунок 3.9 – Привилегии роли Zritel на представление Результаты

3.3 Резервное копирование и восстановление данных

Для того чтобы в случае сбоя можно было восстановить БД, задачей администратора является обеспечение регулярного копирования данных.

Резервная копия – это копия данных, используемая для восстановления данных после сбоя. Необходимо резервировать две компоненты: базу данных, журнал транзакций.

Для начала надо выбрать конкретную базу данных, подлежащую резервированию, потом в меню выбираем команду Tasks (задачи), затем команду Back Up (создать резервную копию). Там можно определить: тип резервирования, расположение резервной копии БД. На рисунке 3.10 представлено окно резервного копирования, в котором была создана полная (Full) копия БД.

На вкладке Параметры резервного копирования имеется возможность определить параметры резервного копирования. Можно определить необходимые параметры, такие как, «Имя», «Описание», «Срок действия резервного набора данных», также «Сжатие», «Шифрование». Все остальное представлено на рисунке 3.11.

Затем была создана разностная (Differential) копия данных, которая представлена на рисунке 3.12.

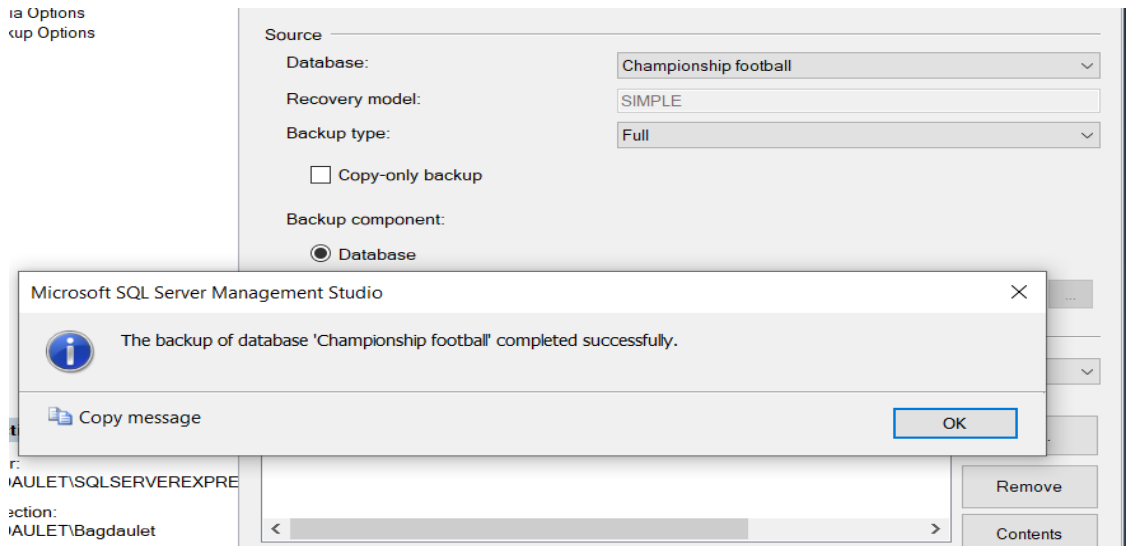


Рисунок 3.10 – Создание полной резервной копии

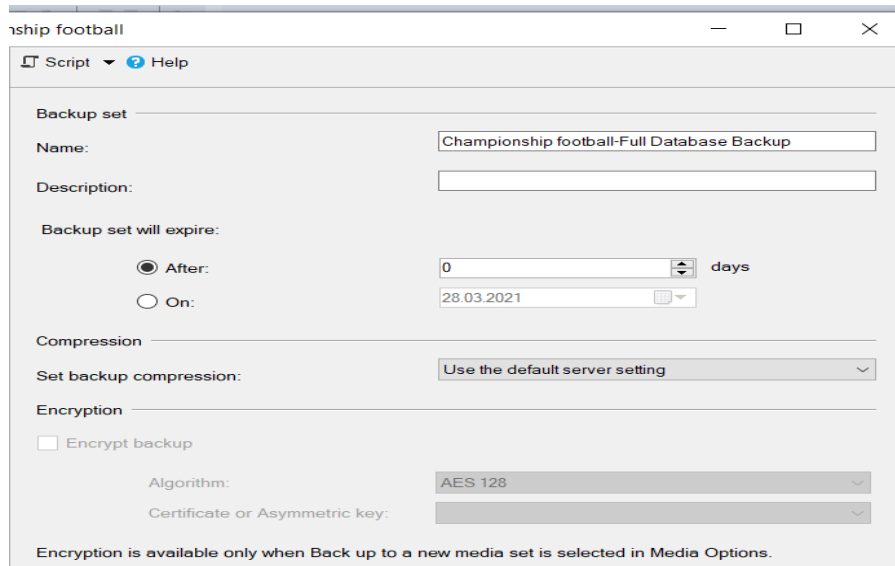


Рисунок 3.11 - Параметры резервного копирования

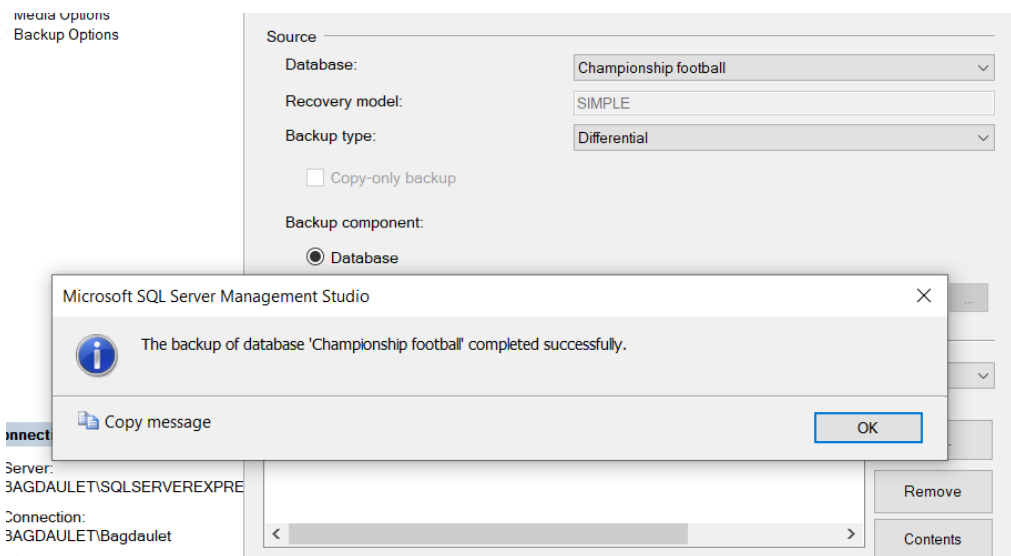


Рисунок 3.12 – Создание разностной резервной копии

Далее было выполнено резервирование журнала транзакции (Transaction Log), которое представлено на рисунке 3.13.

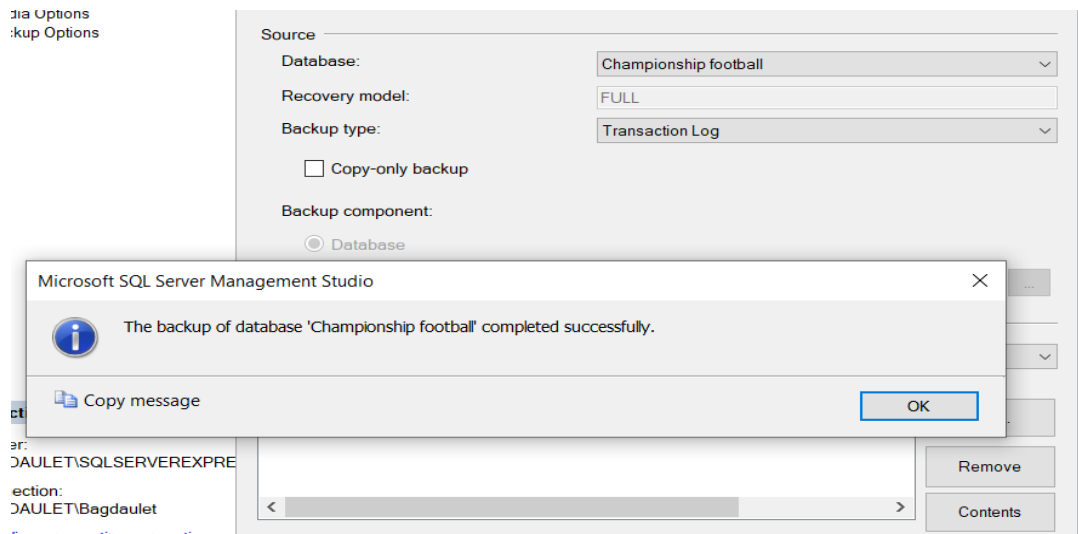


Рисунок 3.13 – Создание копии журнала транзакции

Для восстановления БД также в пункте меню выбираем Tasks, затем команду Restore Database (Восстановить базу данных). Все остальное представлено на рисунке 3.14.

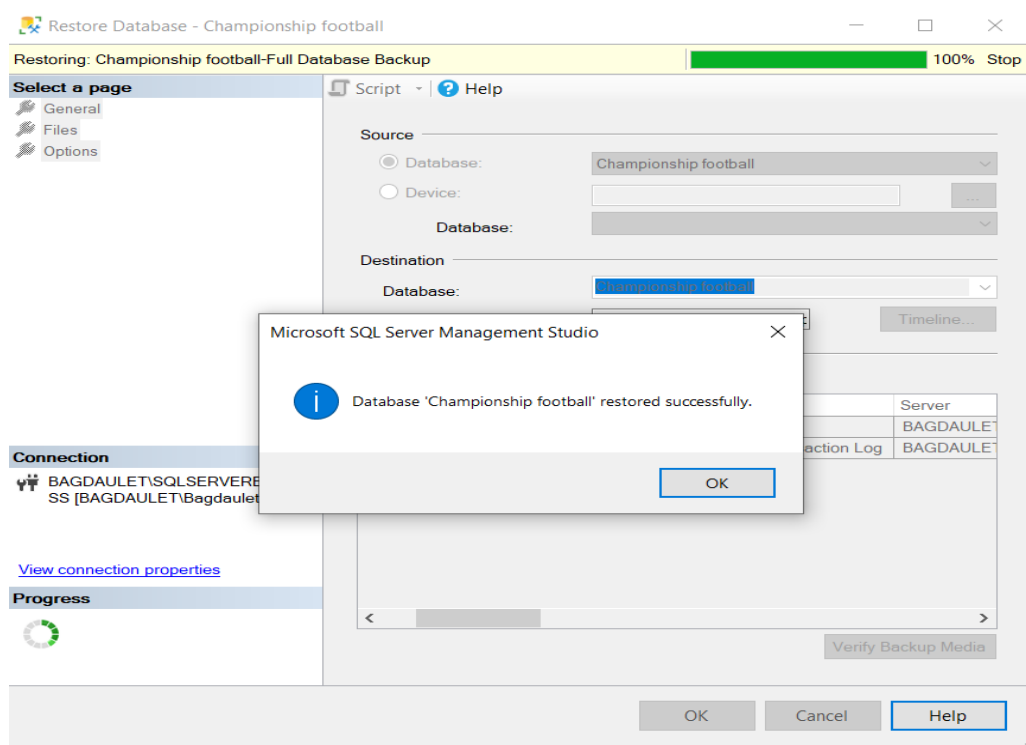


Рисунок 3.14 – Восстановление базы данных

ЗАКЛЮЧЕНИЕ

В ходе дипломного проекта была выполнена разработка и администрирование баз данных MS SQL Server с ориентацией на задачи администратора баз данных для обеспечения защиты и безопасности баз данных на уровне сервера и баз данных.

Определены основные задачи администрирования баз данных. Рассмотрены механизмы администрирования, среди которых были определены оптимизация и обслуживания работы баз данных, стратегия резервного копирования, и основные средства для выполнения импорта/экспорта данных. Также была рассмотрена реализация политики безопасности и шифрование баз данных. Были определены специальные механизмы для выполнения аудита и мониторинга. Для решения задач администратора БД были рассмотрены встроенные механизмы администрирования MS SQL Server.

Был проведен анализ предметной области «Чемпионаты по футболу». На основе этого анализа были определены сущности, атрибуты и связи между ними, выполнено ER-проектирование базы данных в CASE-средстве проектирования баз данных Erwin (AllFusion Erwin Data Modeler r7) на логическом и физическом уровне. Для обеспечения информационной безопасности, а именно целостности базы данных, в физической модели были созданы ограничения целостности, такие как ограничения на значение, значения по умолчанию, допустимость пустого значения, первичные и внешние ключи, что предотвращает внесение случайных ошибок.

Были сгенерированы SQL скрипты для реализации базы данных в СУБД MS SQL Server. База данных была реализована в MS SQL Server 2016. А также созданы представления для обеспечения удобства пользователей и защиты физических таблиц.

Триггеры обеспечивают поддержку целостности данных, запрет несанкционированного срабатывания. Были созданы триггеры DML и DDL для слежения за выполняемыми действиями в БД.

Было рассмотрена встроенный механизм предоставления привилегий, созданы роли для директора, судей и для зрителей, а также выданы привилегии ролям для защиты конкретной баз данных. Выполнено резервное копирования и восстановления баз данных.

В этой работе был использован комплексный подход для обеспечения БД, начиная с этапа проектирования целостной и корректной базы данных и включая использование механизмов обеспечения безопасности.

По результатам работы был сделан доклад на Международной конференции Сатпаевские Чтения – 2021 «Безопасность базы данных Чемпионаты по футболу» [12].

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Microsoft SQL Server // Электронная версия на сайте https://ru.wikipedia.org/wiki/Microsoft_SQL_Server
2. В. В. Скакун. Защита информации в базах данных и экспертных системах: пособие для студентов фак. радиофизики и комп. технологий /. – Минск: БГУ, 2015. – 140 с.: ил.
3. И.С. Осетрова. Администрирование MS SQL Server 2014 - СПб: Университет ИТМО, 2016. – 90 с.
4. Основы безопасности SQL Server // Электронная версия на сайте <https://www.osp.ru/winitpro/2016/12/13051089>
5. Основные средства обеспечения безопасности в SQL Server // Электронная версия на сайте <https://winitpro.ru/index.php/2020/02/07/bezopasnost-sql-server/>
6. ER-модель // Электронная версия на сайте <https://ru.wikipedia.org/wiki/ER-модель>
7. Создание баз данных с использованием пакета Erwin // Электронная версия на сайте http://www.pl63.edu.ru/images/doc/study/INF/method/m_erWin.pdf
8. SQL — Ограничения // Электронная версия на сайте <https://webformyself.com/sql-ogranicheniya/>
9. Представления // Электронная версия на сайте <https://docs.microsoft.com/ru-ru/sql/relational-databases/views/views?view=sql-server-ver15>
10. DML триггеры – SQL Server // Электронная версия на сайте <https://docs.microsoft.com/ru-ru/sql/relational-databases/triggers/dml-triggers?view=sql-server-2017>
11. Функция EVENTDATA (Transact-SQL) // Электронная версия на сайте <https://docs.microsoft.com/ru-ru/sql/t-sql/functions/eventdata-transact-sql?view=sql-server-ver15>
12. Амангелді Б.Б., Айтхожаева Е.Ж. Безопасность базы данных Чемпионаты по футболу. Материалы Международной конференции «Сатпаевские Чтения – 2021».

Приложение А

Создание базы Чемпионаты по футболу:

```
CREATE TABLE Championship
(
    Code_championship integer NOT NULL ,
    Vid_championship varchar(20) NULL ,
    Data_starting datetime NULL
    CONSTRAINT DATA_942812699
        DEFAULT GETDATE(),
    Budget integer NULL
)
```

```
ALTER TABLE Championship
    ADD CONSTRAINT XPKChampionship PRIMARY KEY
    CLUSTERED (Code_championship ASC)
```

```
CREATE TABLE Matches
(
    Code_matches integer NOT NULL ,
    Point_1 integer NULL
    CONSTRAINT SESSION_USER_1209035622
        DEFAULT SESSION_USER,
    Time integer NULL ,
    Code_championship integer NULL ,
    Code_teams_1 integer NULL ,
    Code_teams_2 integer NOT NULL ,
    Point_2 integer NULL
)
```

```
ALTER TABLE Matches
    ADD CONSTRAINT XPKMatches PRIMARY KEY CLUSTERED
    (Code_matches ASC)
```

```
CREATE TABLE Players
(
    Id_players integer NOT NULL ,
    FIO varchar(20) NULL ,
    God_rojdenya datetime NULL ,
```

```

        Position varchar(20) NULL ,
        Grajdanstvo varchar(20) NULL ,
        Zarplata integer NULL ,
        Code_teams integer NULL
    )

ALTER TABLE Players
    ADD CONSTRAINT XPKPlayers PRIMARY KEY CLUSTERED
(Id_players ASC)

CREATE TABLE Team
(
    Code_teams integer NOT NULL ,
    Name varchar(20) NULL ,
    Strana varchar(20) NULL ,
    Gorod varchar(20) NULL ,
    Budget integer NULL
)

ALTER TABLE Team
    ADD CONSTRAINT XPKTeam PRIMARY KEY CLUSTERED
(Code_teams ASC)

CREATE TABLE Trener
(
    Id_trenera integer NOT NULL ,
    FIO varchar(20) NULL ,
    God_rojdenya datetime NULL ,
    Grajdanstvo varchar(20) NULL ,
    Telefon integer NULL
CONSTRAINT USER_1619818266
    DEFAULT USER,
    Zarplata integer NULL ,
    Code_teams integer NULL
)

ALTER TABLE Trener
    ADD CONSTRAINT XPKTrener PRIMARY KEY CLUSTERED
(Id_trenera ASC)

```

```
ALTER TABLE Matches
    ADD CONSTRAINT R_2 FOREIGN KEY (Code_championship)
REFERENCES Championship(Code_championship)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

```
ALTER TABLE Matches
    ADD CONSTRAINT R_3 FOREIGN KEY (Code_teams_1) REFERENCES
Team(Code_teams)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

```
ALTER TABLE Matches
    ADD CONSTRAINT R_6 FOREIGN KEY (Code_teams_2) REFERENCES
Team(Code_teams)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

```
ALTER TABLE Players
    ADD CONSTRAINT R_4 FOREIGN KEY (Code_teams) REFERENCES
Team(Code_teams)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

```
ALTER TABLE Trener
    ADD CONSTRAINT R_5 FOREIGN KEY (Code_teams) REFERENCES
Team(Code_teams)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
```

```
exec sp_bindrule 'Time', 'Matches.Time'
go
```

```
exec sp_bindrule 'Position', 'Players.Position'
go
```

```
exec sp_bindrule 'Zarplata', 'Players.Zarplata'
go
```

```
exec sp_bindrule 'Zarp', 'Trenер.Zarplata'  
go
```

```
CREATE RULE Zarplata  
AS @col BETWEEN 100000 AND 500000
```

```
CREATE RULE Zarp  
AS @col BETWEEN 100000 AND 1000000
```

```
CREATE RULE Position  
AS @col IN ('Нападающий', 'Полузащитник', 'Защитник', 'Вратарь')
```

```
CREATE RULE Time  
AS @col BETWEEN 0 AND 90
```

Приложение Б

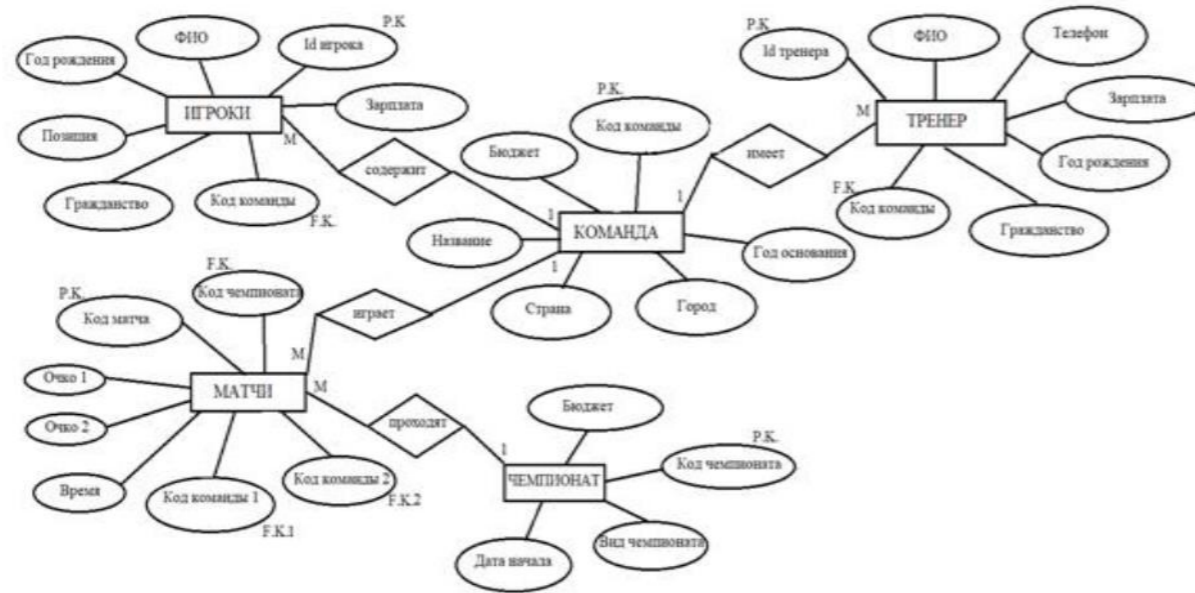
Механизмы администрирования

- **Обслуживание и оптимизация работы баз данных.** Для решения задач обслуживания и оптимизации работы баз данных, администратор баз данных работающий с Microsoft SQL Server, обязан управлять базами данных и хранением данных.
- **Резервное копирование и восстановление баз данных.** Администратор баз данных разрабатывает стратегию резервного копирования баз данных для обеспечения надежности БД с использованием полной копии, разностной копии или копий журнала транзакций. Наличие резервных копий БД позволяет восстановить БД после случайных или преднамеренных сбоев, а также после атак вирусов-шифровальщиков.
- **Импорт и экспорт данных.** SQL Server поддерживает массовый экспорт данных из таблиц SQL Server и массовый импорт данных в таблицы или представления. Для выполнения этих задач SQL Server предоставляет набор инструментов: программа Вср, инструкции BULK INSERT, функция OPENROWSET и мастер импорта и экспорта.
- **Реализация политик безопасности.** Администратор баз данных должен реализовывать такие политики безопасности, которые обеспечивают пользователям доступ к необходимым данным, но при этом соблюдают правовые нормы бизнеса по защите своих активов, а также снижают риски, связанные с нарушением безопасности.
- **Мониторинг и аудит.** Все серверы БД имеют специальные механизмы для выполнения аудита и мониторинга, чтобы узнать, насколько сервер баз данных в данный момент загружается, какие ресурсы доступны. Самым распространенным из них является SQL Profiler, SQL Audit.

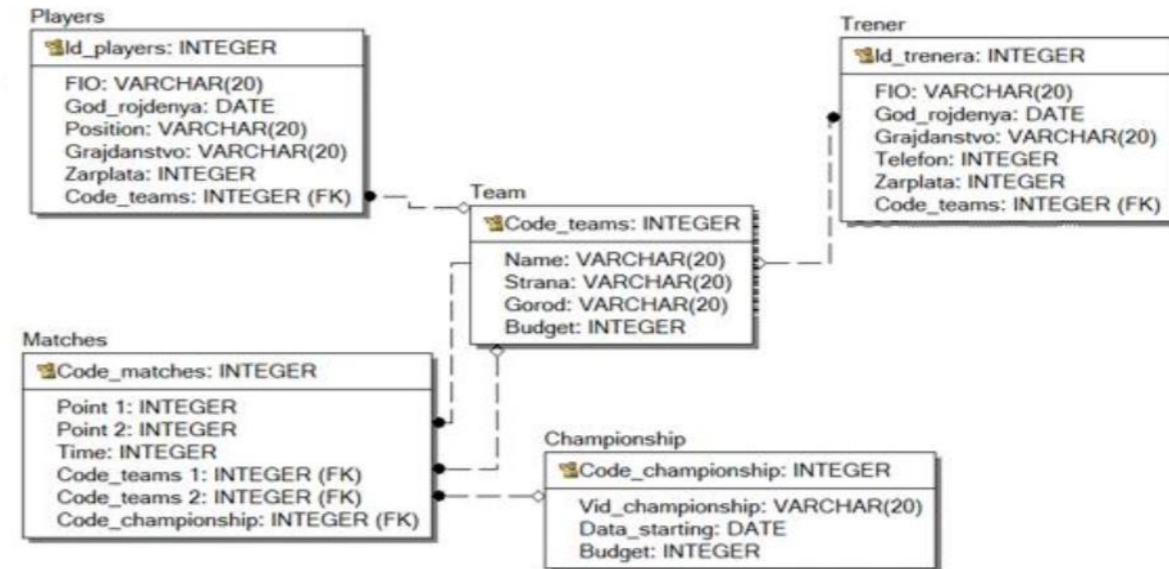
					Дипломный проект				
Изм.	Лист	Ф.И.О	Подп.	Дата	Механизмы администрирования в MS SQL Server	Лист	Масса	Масштаб	
Разраб.		Амангелді Б.							
Норм.		Кабдуллин М.							
Руков.		Айтхожаева Е							
Зав. каф.		Сейлова Н.			Тема: Разработка и администрирование базы данных MS SQL Server	Лист 1		Листов 5	
						КазНУТУ ИКИИТ СИБ 5В100200			

Приложение Б

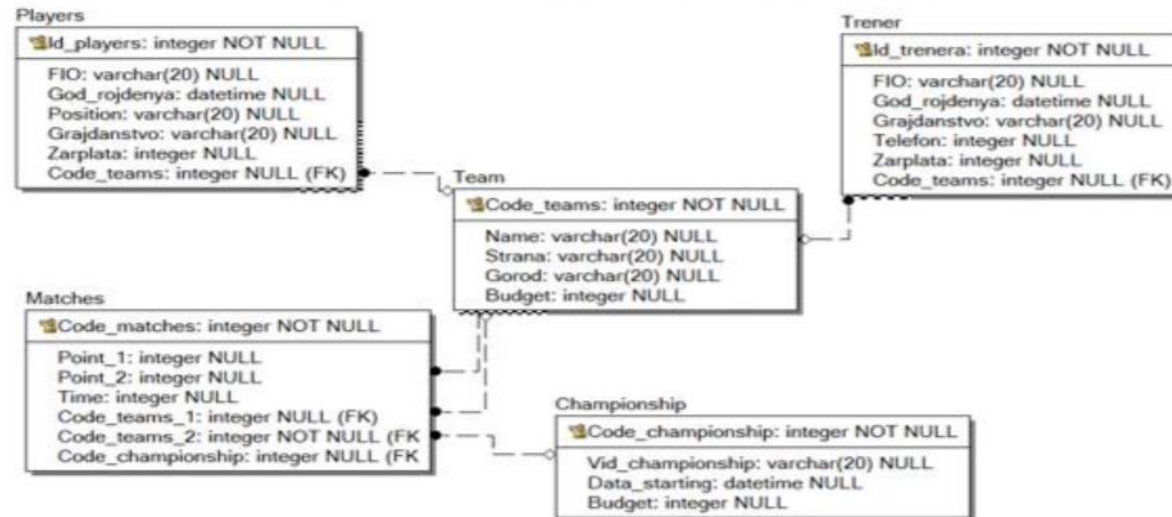
ER-диаграмма



Логическая модель баз данных



Физическая модель баз данных



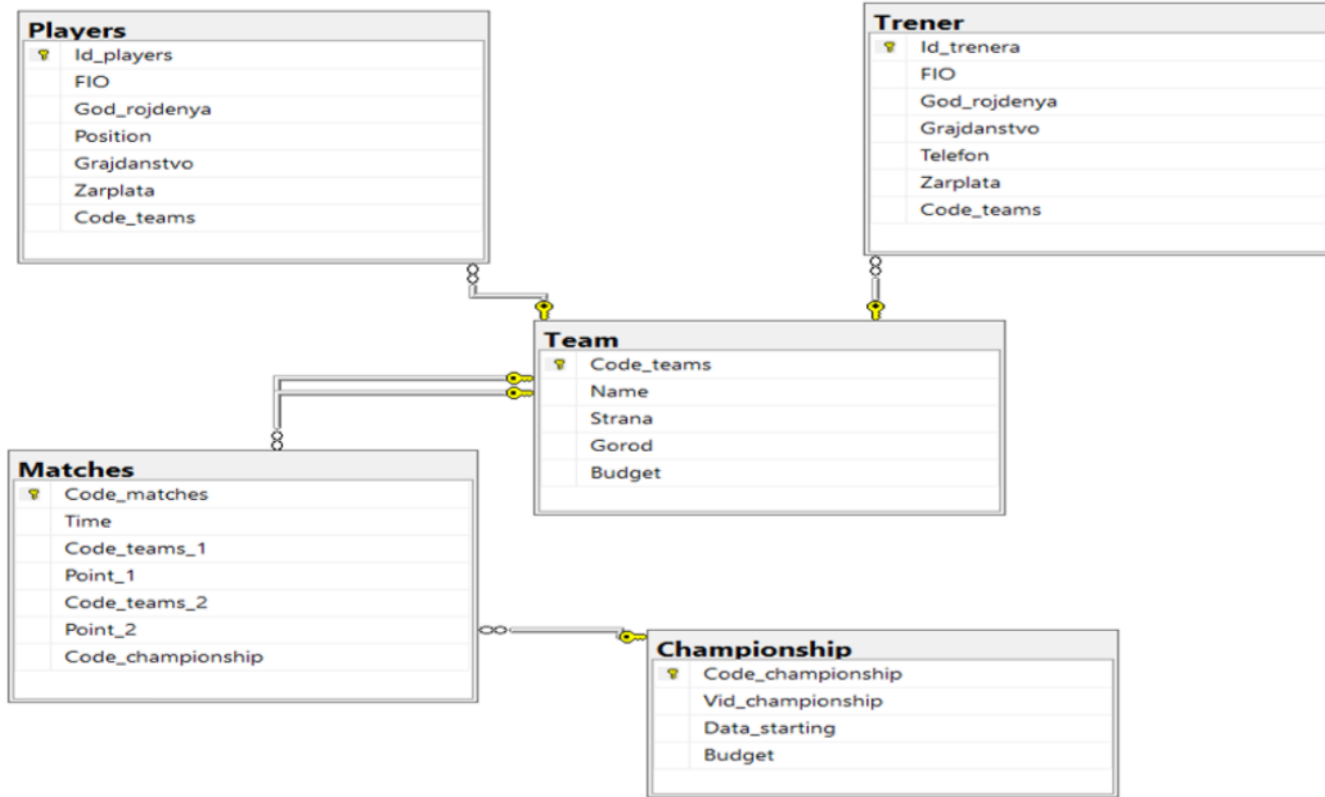
Задание ограничения Valid Values List

					Дипломный проект		
					Лист	Масса	Масштаб
Изм.	Лист	Ф.И.О	Подп.	Дата	Проектирование логической и физической модели данных в Erwin		
Разраб.		Амангелді Б.					
Норм.		Кабдуллин М.					
Руков.		Айтхожаева Е					
Зав. каф.		Сейлова Н.			Лист 2		Листов 5
					Тема: Разработка и администрирование базы данных MS SQL Server		КазНУТУ ИКиИТ СИБ 5В100200

Приложение Б

Структуры таблиц БД

Схема БД



Column Name	Data Type	Allow Nulls
Id_players	int	<input type="checkbox"/>
FIO	varchar(20)	<input checked="" type="checkbox"/>
God_rojdenya	datetime	<input checked="" type="checkbox"/>
Position	varchar(20)	<input checked="" type="checkbox"/>
Grajdanstvo	varchar(20)	<input checked="" type="checkbox"/>
Zarplata	int	<input checked="" type="checkbox"/>
Code_teams	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
Code_matches	int	<input type="checkbox"/>
Time	int	<input checked="" type="checkbox"/>
Code_teams_1	int	<input checked="" type="checkbox"/>
Point_1	int	<input checked="" type="checkbox"/>
Code_teams_2	int	<input type="checkbox"/>
Point_2	int	<input checked="" type="checkbox"/>
Code_championship	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
Id_trenera	int	<input type="checkbox"/>
FIO	varchar(20)	<input checked="" type="checkbox"/>
God_rojdenya	datetime	<input checked="" type="checkbox"/>
Grajdanstvo	varchar(20)	<input checked="" type="checkbox"/>
Telefon	int	<input checked="" type="checkbox"/>
Zarplata	int	<input checked="" type="checkbox"/>
Code_teams	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
Code_teams	int	<input type="checkbox"/>
Name	varchar(20)	<input checked="" type="checkbox"/>
Strana	varchar(20)	<input checked="" type="checkbox"/>
Gorod	varchar(20)	<input checked="" type="checkbox"/>
Budget	int	<input checked="" type="checkbox"/>

Column Name	Data Type	Allow Nulls
Code_championship	int	<input type="checkbox"/>
Vid_championship	varchar(20)	<input checked="" type="checkbox"/>
Data_starting	datetime	<input checked="" type="checkbox"/>
Budget	int	<input checked="" type="checkbox"/>

					Дипломный проект		
Изм.	Лист	Ф.И.О	Подп.	Дата	Лист	Масса	Масштаб
Разраб.		Амангелді Б.			Лист 3		Листов 5
Норм.		Кабдуллин М.					
Руков.		Айтхожаева Е					
Зав. каф.		Сейлова Н.					
					Схема БД и структуры таблиц в MS SQL Server		
					Тема: Разработка и администрирование базы данных MS SQL Server		
					КазНИТУ ИКИИТ СИБ 5B100200		

Приложение Б

Представления Результаты

Для удобства пользователей и защиты физических таблиц от пользователей было создано представление Результаты. Представление создано на основе трех таблиц (Чемпионаты, Матчи, Команды) и позволяет просмотреть результаты игр всех чемпионатов с указанием очков команд, исключая доступ к реальным таблицам.

The screenshot shows the database schema with tables Championship, Matches, and Team. Below the schema is a query window with the following SQL:

```
SELECT dbo.Championship.Vid_championship AS [Вид чемпионата], dbo.Matches.Code_matches AS [Код матча], dbo.Matches.Time AS [Время], dbo.Team.Name AS [Команда_1],
dbo.Matches.Point_1 AS [Очко_1], Team_1.Name AS [Команда_2], dbo.Matches.Point_2 AS [Очко_2] FROM dbo.Championship INNER JOIN
dbo.Matches ON dbo.Championship.Code_championship = dbo.Matches.Code_championship INNER JOIN
dbo.Team ON dbo.Matches.Code_teams_1 = dbo.Team.Code_teams INNER JOIN
dbo.Team AS Team_1 ON dbo.Matches.Code_teams_2 = Team_1.Code_teams
```

The results table shows the following data:

Вид чемпи...	Код матча	Время	Команда_1	Очко_1	Команда_2	Очко_2
La liga	103	18:00:00	Real Madrid	2	Sevilla	1
La liga	105	20:30:00	Atletico Ma...	0	Barcelona	1
Seria A	202	21:00:00	Juventus	1	Roma	0

Триггеры DML

Триггер tr_Matches для таблицы Матчи, который будет срабатывать при выполнении операций добавления, удаления, обновления строк для контроля выполняемых действий в таблице. Для фиксации результата работы триггера была создана таблица Football_audit. Результат работы триггера в таблице Football_audit.

The screenshot shows the SQL code for creating a trigger:

```
CREATE TRIGGER tr_Matches on Matches
FOR INSERT, DELETE, UPDATE
As
declare @tablename varchar(100)
SELECT @tablename = OBJECT_NAME(parent_object_id)
FROM sys.objects
WHERE sys.objects.name = OBJECT_NAME(@@PROCID)
IF EXISTS ( SELECT 0 FROM Deleted )
BEGIN
INSERT INTO Football_audit(
Table name,Code,Updated at, Operation)
Select @tablename, D.Code_matches, Getdate(), 'Update'
From Deleted D
END
ELSE
```

The execution results show the following data in the Football_audit table:

Audit_ID	Table_name	Code	Updated_at	Operation
1	Matches	418	2021-03-26 11:25:18.960	Insert
2	Matches	418	2021-03-26 11:25:22.567	Delete
3	Matches	101	2021-03-26 11:25:25.800	Update

Триггеры DDL

Триггер tr_football, который будет срабатывать при добавления таблиц в БД и фиксировать выполненные действия в таблице Audit.

The screenshot shows the SQL code for creating a DDL trigger:

```
CREATE TABLE Audit (PostTime datetime, Command varchar(100), Nazvaniya nvarchar(200));
CREATE TRIGGER tr_football
ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS
AS
DECLARE @data XML
SET @data = EVENTDATA()
INSERT Audit
(PostTime, Command, Nazvaniya)
VALUES
(GETDATE(),
@data.value('/EVENT_INSTANCE/EventType')[1], 'varchar(100)',
@data.value('/EVENT_INSTANCE/ObjectName')[1], 'nvarchar(200)') ;
CREATE TABLE Table_Champion(a int)
DROP TABLE Table_Champion
SELECT * FROM Audit ;
```

The execution results show the following data in the Audit table:

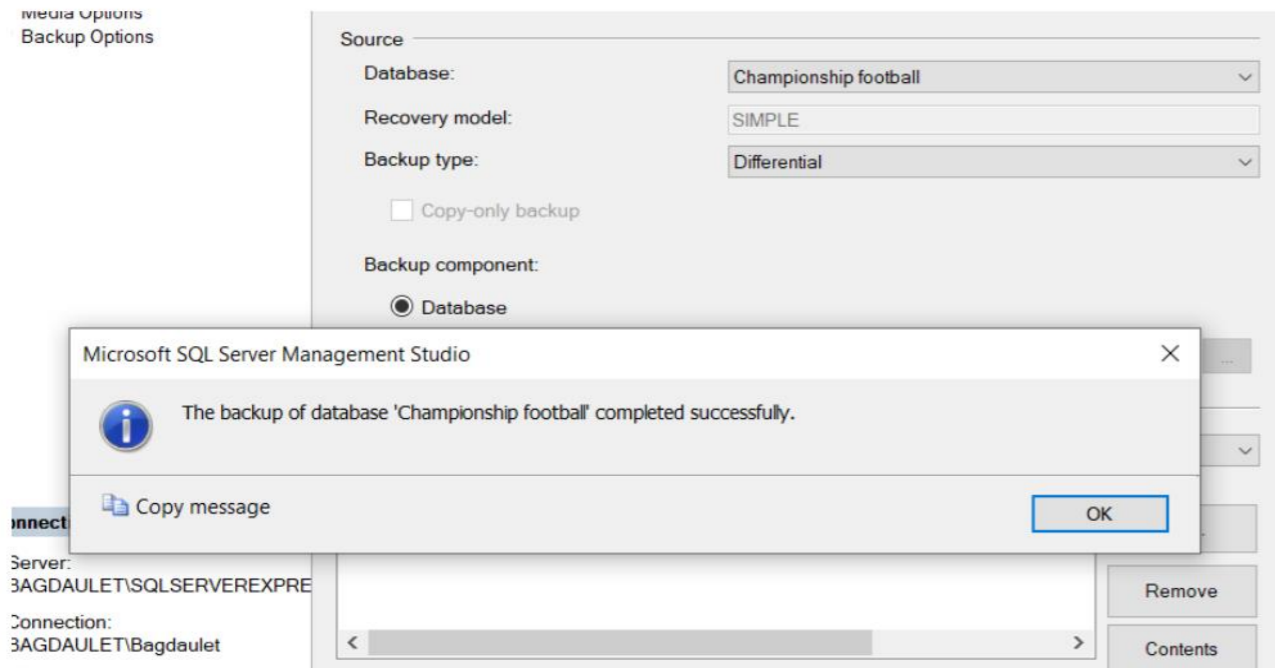
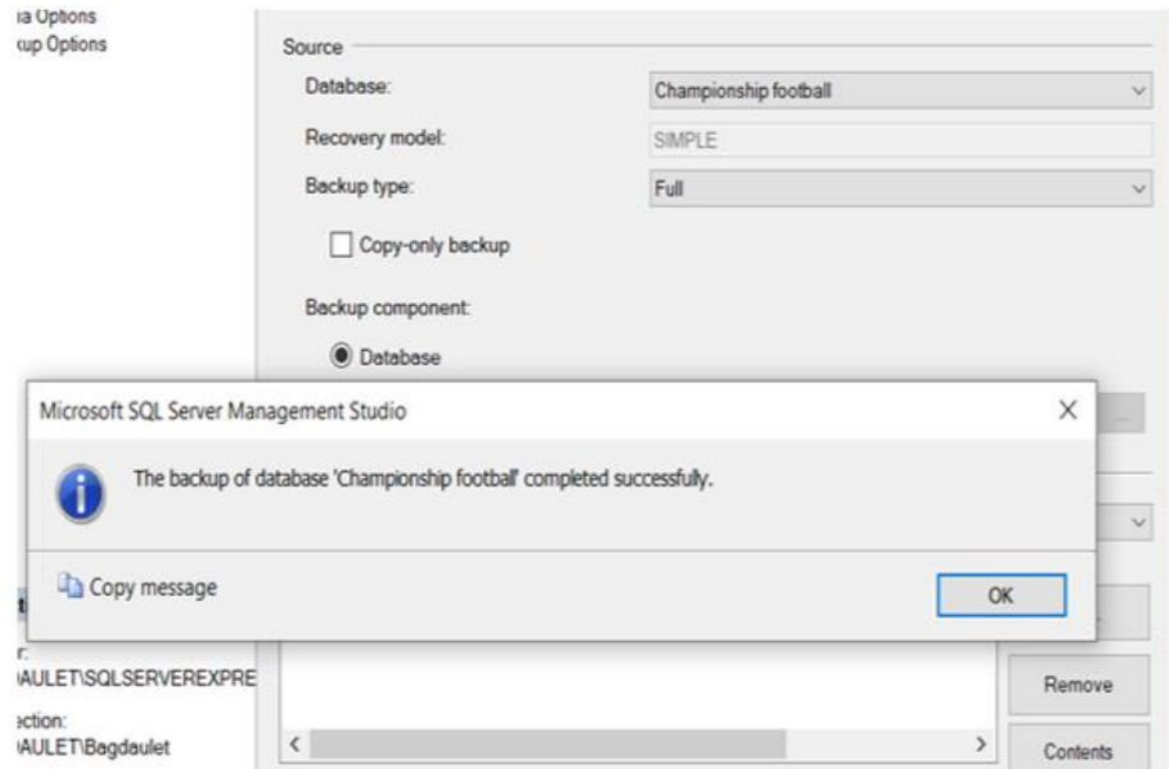
PostTime	Command	Nazvaniya
2021-03-17 12:45:21.930	CREATE_TABLE	Table_Champion
2021-03-17 12:45:30.730	DROP_TABLE	Table_Champion

Дипломный проект

Изм.	Лист	Ф.И.О	Подп.	Дата	Лист	Масса	Масштаб
Разраб.		Амангелді Б.					
Норм.		Кабдуллин М.					
Руков.		Айтхожаева Е					
Зав. каф.		Сейлова Н.					
Тема: Разработка и администрирование базы данных MS SQL Server					Лист 4	Листов 5	
					Представления и триггеры		

Приложение Б

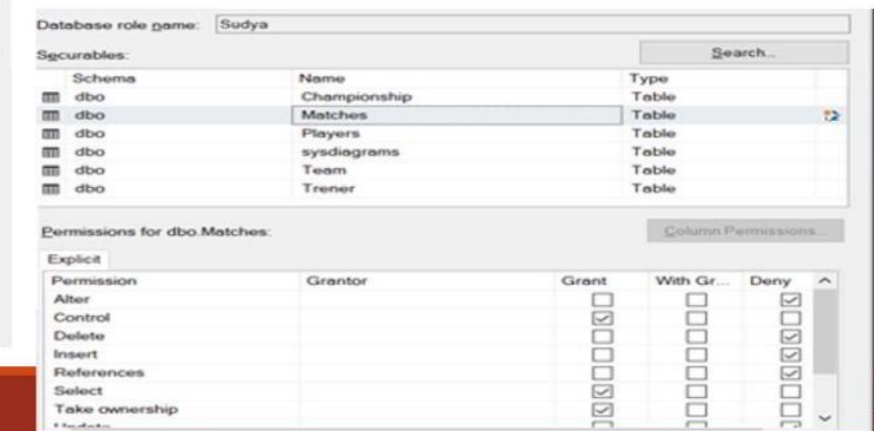
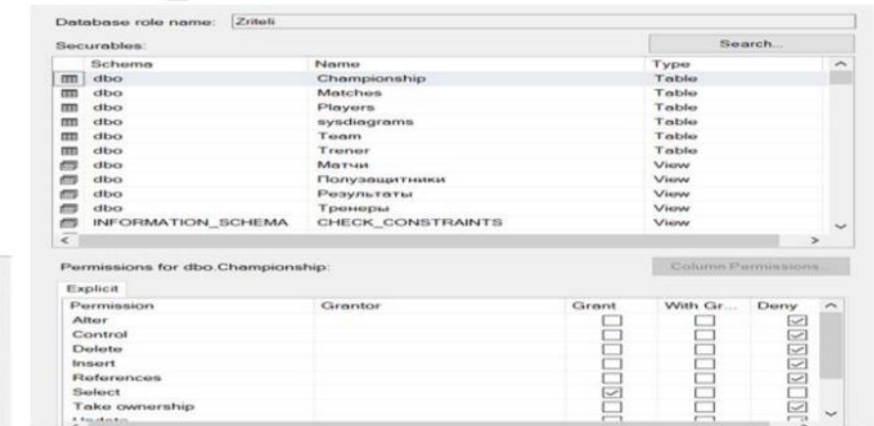
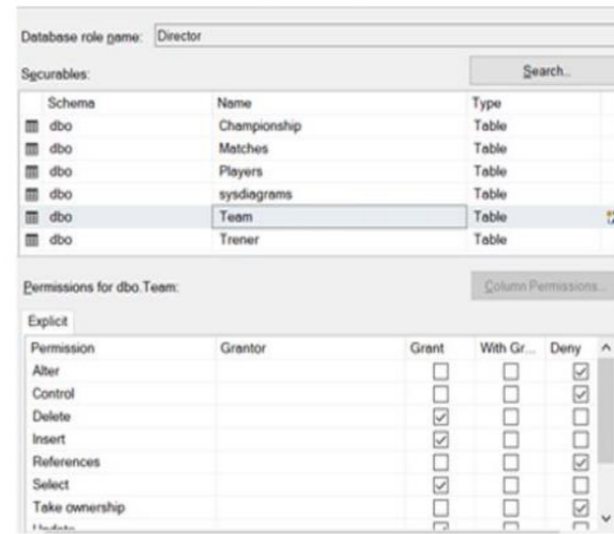
Резервное копирование БД



Системы ролей

Были созданы 3 роли:

- директор (director);
- судья (sudya);
- зрители (zriteli).



					Дипломный проект			
Изм.	Лист	Ф.И.О	Подп.	Дата	Предоставление привилегий и создание резервной копии базы данных	Лист	Масса	Масштаб
Разраб.		Амангелді Б.				Лист 5		Листов 5
Норм.		Кабдуллин М.						
Руков.		Айтхожаева Е			Тема: Проектирование и реализация безопасной базы данных			
Зав. каф.		Сейлова Н.			КазНИТУ ИКИИТ СИБ 5В100200			

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на

дипломный проект

(наименование вида работы)

Амангелді Бакдәулет Бауыржанұлы

(Ф.И.О. обучающегося)

5B100200 - Системы информационной безопасности

(шифр и наименование специальности)

Тема:

«Разработка и администрирование базы данных MS SQL Server»

Перед дипломником была поставлена задача решения ряда вопросов, связанных с разработкой и администрированием баз данных в одном из самых распространенных серверов БД MS SQL Server, с которой он успешно справился.

Амангелді Б.Б. самостоятельно выполнил всю работу по дипломному проекту. Выполнил рассмотрение механизмов администрирования БД в MS SQL Server, анализ предметной области «Чемпионаты по футболу». Для проектирования базы данных был использован ER-метод, создана информационная модель предметной области с учетом типов сущностей, определены их атрибуты и связи между ними, ограничения целостности.

Сгенерированы SQL скрипты для реализации базы данных в СУБД MS SQL Server. База данных была реализована в MS SQL Server 2016 для учета чемпионатов, футбольных матчей, команд, игроков, тренеров, фиксации результатов матчей, предоставления справочной информации болельщикам, со всеми необходимыми компонентами.

Для обеспечения безопасности БД были созданы представления и триггеры. Представления созданы с целью обеспечения комфортности работы пользователей и защиты физических таблиц, триггеры для мониторинга изменений в базе данных.

Рассмотрен механизм предоставления привилегий, созданы роли для различных групп пользователей, выданы им привилегии. Была предложена и реализована стратегия резервного копирования и восстановления БД.

В процессе дипломирования Амангелді Б.Б. показал практическое умение работать с сервером баз данных MS SQL Server и его механизмами администрирования, хорошие инженерные навыки как в области анализа, проектирования и реализации баз данных, так и умение работать с технической литературой.

Работа выполнена с использованием современных IT-технологий: CASE-средство проектирования баз данных AllFusion Erwin Data Modeler, сервер MS SQL Server 2016, утилита SQL Server Management Studio.

Дипломный проект на тему «Разработка и администрирование базы данных MS SQL Server» выполнен Амангелді Б.Б. на хорошем уровне и может быть допущен к защите.

Научный руководитель

ассоц.профессор, к.т.н.

(должность, ученая степень, звание)

 Айтхожаева Е.Ж.

(подпись)

« 10 » 05 2021 г.

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился (-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Амангелді Б.Б.

Название: Разработка и администрирование базы данных MS SQL Server

Координатор: Айтхожаева Е.Ж.

Коэффициент подобия 1: 8.47%

Коэффициент подобия 2: 2.07%

Тревога: 0.58%

После анализа Отчета подобия констатирую следующее:

✓ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

Заимствования объясняются использованием общепринятой технической терминологии, в работе приведены стандартное описание функций используемого CASE-средства проектирования баз данных AllFusion Erwin Data Modeler (ссылки на источники имеются) и описание механизмов администрирования, взятое с официального сайта Microsoft (в списке использованных источников указаны).

«24» мая 2021 г.
Дата


Подпись Научного руководителя

Протокол анализа Отчета подобия заведующего кафедрой

Заведующий кафедрой заявляет, что ознакомился (-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Амангелді Б.Б.

Название: Разработка и администрирование базы данных MS SQL Server

Координатор: Айтхожаева Е.Ж.

Коэффициент подобия 1: 8.47%

Коэффициент подобия 2: 2.07%

Тревога: 0.58%

После анализа Отчета подобия констатирую следующее:

✓ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

□ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

□ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

Незначительные заимствования объясняются наличием в тексте работы широко используемой технической терминологии, в работе также приведено стандартное описание используемого инструментария: CASE-средства проектирования баз данных AllFusion Erwin Data Modeler и механизмов администрирования и мониторинга в MS SQL Server фирмы-производителя Microsoft (ссылки имеются)

«24» мая 2021 г.

Дата


Сейлова Н.А. 
Ф.И.О., подпись зав.кафедрой

Окончательное решение в отношении допуска к защите, включая обоснование:

Дипломный проект к защите допускается в связи с признанием заимствований добросовестными.

«24» мая 2021 г.

Дата

Сейлова Н.А. 
Ф.И.О., подпись зав.кафедрой