**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

**КУРСОВОЙ ПРОЕКТ**

По дисциплине МДК.07.01 Управление и автоматизация баз данных

**Разработка базы данных «Компьютерный клуб»**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_\_\_/Жолдошев Бекбол Рысбекович /

(подпись) (Ф.И.О.)

**ОЦЕНКА:**\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ**

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_/ Егоров Данил Павлович/

(подпись) (Ф.И.О.)

Воркута

2025

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc200955115)

[ГЛАВА 1. ПРЕДПРОЕКТНАЯ ЧАСТЬ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ 4](#_Toc200955116)

[1.1 Анализ предметной области 4](#_Toc200955117)

[1.2 Разработка технического задания 4](#_Toc200955118)

[1.3 Моделирование бизнес-процессов (IDEF0 is/to be) 8](#_Toc200955119)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ И ВЫБОР ИНСТРУМЕНТАРИЯ 12](#_Toc200955120)

[2.1 Разработка диаграммы ERD 12](#_Toc200955121)

[2.2 Словарь данных 14](#_Toc200955122)

[2.3 СУБД SQL Server 18](#_Toc200955123)

[2.4 Microsoft SQL Server Management Studio 19](#_Toc200955124)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 22](#_Toc200955125)

[3.1 Разработка базы данных 22](#_Toc200955126)

[3.2 Введение в безопасность SQL Server 29](#_Toc200955127)

[3.3 Управление безопасностью уровня сервера 31](#_Toc200955128)

[3.4 Управление участниками уровня базы данных 34](#_Toc200955129)

[3.5 Управление разрешениями уровня базы данных 37](#_Toc200955130)

[3.6 Резервное копирование баз данных 40](#_Toc200955131)

[3.7 Шифрование данных баз данных 42](#_Toc200955132)

[ЗАКЛЮЧЕНИЕ 43](#_Toc200955133)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 45](#_Toc200955134)

# ВВЕДЕНИЕ

Компьютерный клуб — это специализированное заведение, предоставляющее клиентам доступ к современным компьютерным технологиям и игровым ресурсам. В таких клубах посетители могут использовать высокопроизводительные игровые станции, участвовать в турнирах, тренироваться и просто проводить свободное время за играми или работой на компьютере.

Современные компьютерные клубы отличаются широким спектром услуг, включающих не только аренду игровых мест, но и организацию мероприятий, продажу дополнительных сервисов и продуктов. Для эффективного функционирования клуба необходимо вести учёт посетителей, контролировать использование оборудования, управлять расписанием и бронированиями, а также обеспечивать качественное обслуживание клиентов.

С развитием технологий и увеличением требований к уровню сервиса автоматизация этих процессов становится ключевым элементом успешной работы клуба. Надёжная и функциональная база данных играет важную роль в обеспечении удобного и быстрого доступа к информации, необходимой для ежедневной работы персонала и удовлетворения потребностей клиентов.

**Объект**: база данных «Компьютерный клуб».

**Предмет**: анализ бизнес-процесса «Компьютерный клуб».

**Цель работы**: разработать базу данных «Компьютерный клуб».

**Задачи:**

* ﻿﻿выбрать инструментарий;
* ﻿﻿спроектировать базу данных;
* разработать базу данных.

# ГЛАВА 1. ПРЕДПРОЕКТНАЯ ЧАСТЬ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ

## 1.1 Анализ предметной области

Каждый день в компьютерных клубах проводится множество сессий, предоставляются различные услуги и осуществляется учет клиентов. Для обеспечения эффективной работы и оперативного учета информации о пользователях и услугах необходима база данных. Администраторам компьютерного клуба важно оперативно управлять данными о посетителях, сессиях, оплатах и состоянии компьютеров. Использование базы данных значительно упростит процессы учета, администрирования и предоставления услуг.

В базе данных будет храниться обширная информация о клиентах, сессиях, играх, чтобы оперативно выявлять различные сведения о посетителях, о сотрудниках.

Работать с базой данных компьютерного клуба будут следующие пользователи:

1. Сотрудники
2. Клиенты

Сотрудники компьютерного клуба должны иметь возможность добавлять клиентов в клиентскую базу, находить информацию по ним в базе, записывать клиентов на определенные компьютеры, отслеживать информацию об устройствах, отслеживать статистику посещений клуба.

Клиенты должны иметь возможность бронировать себе на определенное время компьютеры и виды комнат, просматривать актуальные цены на аренду компьютеров.

## 1.2 Разработка технического задания

**1. Наименование работы**: Разработка базы данных «Компьютерный клуб».

**2. Назначение разработки:** Разработка данной базы данных предназначена для централизованного хранения и эффективного управления информацией о деятельности компьютерного клуба. База данных позволит автоматизировать ключевые процессы, улучшить качество обслуживания клиентов и повысить эффективность управления клубом.

Пользователи базы данных:

Руководитель: Анализ отчетов, принятие решений, таких как изменение цен, внедрение новых услуг, закупка оборудования.

Администраторы: Регистрация клиентов, запись на сессии, прием оплаты, управление оборудованием, формирование отчетов, управление бронированиями.

Клиенты: Бронирование компьютеров/услуг, просмотр информации о ценах, доступности оборудования и списке игр.

**3. Требования к программе**

**3.1. Требования к функциональным характеристикам**

База данных «Компьютерный клуб» должна обеспечивать выполнение функций:

* Учет клиентов: Регистрация, хранение и обновление информации о клиентах (имя, контакты, история посещений).
* Управление тарифами: Хранение тарифных планов (цены, условия), привязка тарифов к залам или компьютерам
* Учет посещений и бронирований: Регистрация времени начала и окончания посещения, привязка посещения к конкретному клиенту и оборудованию, бронирование компьютеров на определенное время.
* Управление сеансами: Фиксация начала и окончания использования компьютера клиентом, расчет стоимости сеанса согласно тарифам
* Анализ данных: Сбор и анализ данных о посещаемости, популярности услуг и игр, создание отчетов для принятия управленческих решений.

Нормативно-справочная информация базы данных компьютерного клуба включает справочники игр, оборудования, услуг и персонала для обеспечения единообразия и целостности данных.

Первичные документы в базе данных компьютерного клуба:

* Чек: подтверждает оплату клиентом услуг.
* Журнал бронирований: фиксирует бронь компьютера/консоли.

Выходными данными являются следующие виды отчетов:

* Отчет о посещаемости: Анализ посещений (количество, время).
* Отчет о клиентах: Информация о клиентах (количество, активность).
* Отчет по бронированиям: Анализ бронирований.

**3.2. Требования к надежности**

Разрабатываемая база данных должна иметь:

■ возможность самовосстановления после сбоев (отключения электропитания, сбои в операционной системе ит. д.);

■ ограничение несанкционированного доступа к данным;

■ возможность резервного копирования информационной базы;

■ данные в базе данных должны иметь возможность восстановления.

■ разграничение пользовательских прав;

■ данные в базе данных должны иметь шифрование;

Предусмотреть контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой.

**3.3. Требования к составу и параметрам технических средств**

Системные требования для работы программного продукта должны быть следующими: тактовая частота процессора - 1800 Гц, объем оперативной памяти 4 Гб; объем свободного дискового пространства 2 Гб; разрешение монитора 1920 х 1080.

**3.4. Требования к информационной и программной совместимости**

Программа должна работать в операционных системах Windows 10. Все формируемые отчеты должны иметь возможность экспортирования в редактор электронных таблиц MS Office Excel 2016/2021 и MS Office Word 2016/2019.

**3.5. Требования к транспортированию и хранению**

Программная документация поставляется в электронном и печатном виде.

**3.6. Специальные требования**

База данных должна иметь дружественный интерфейс, рассчитанный на пользователя средней квалификации (с точки зрения компьютерной грамотности).

Ввиду объемности проекта задачи предполагается решать поэтапно. При этом модули базы данных, созданные в разное время, должны предполагать возможность наращивания системы и быть совместимы друг с другом; поэтому документация на принятую эксплуатационную базу данных должна содержать полную информацию, необходимую для работы с ней программистов. Язык запросов к базе данных определяется выбором исполнителя, при этом он должен обеспечивать возможность интеграции базы данных с пакетом MS Office 2010/2019.

**4. Требования к программной документации**

В ходе разработки базы данных для ПК-клуба должны быть подготовлены:

текст базы данных, описание структуры и функционала базы данных, программа и методика испытаний, руководство пользователя.

При выполнении операций по регистрации пользователей, бронированию компьютеров и учету времени их использования используется ручной труд, а именно ведется журнал учета пользователей и их активности. Очевидно, что использование базы данных значительно сократит время, затрачиваемое на регистрацию и учет. Кроме того, для каждого пользователя существует карточка, в которую заносятся сведения о времени использования компьютера и остатке доступных часов. Для получения этих сведений по конкретному пользователю требуется не менее 5—7 минут. С использованием базы данных затраты времени сокращаются до 1—2 минут.

В конце каждого дня ответственный работник клуба составляет отчеты об использовании компьютеров и выводит остаток доступного времени для каждого пользователя. На эту операцию уходит 1—2 часа. Формирование отчетов в базе данных займет 3—4 минуты. Кроме того, предполагается возможность получения отчетов за любой период времени. При ручном создании отчетов могут быть допущены ошибки; правильно составленный алгоритм разрабатываемой базы данных исключает эти ошибки.

Экономический эффект от внедрения базы данных для ПК-клуба ожидается за счет сокращения времени на выполнение операций администраторов, исключения ошибок при формировании отчетов, увеличения времени на анализ активности пользователей и т.д.

## 1.3 Моделирование бизнес-процессов (IDEF0 is/to be)

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ.

 Данная модель позволяет описать все основные виды процессов, как административные, так и организационные.

Стрелки могут быть:

* входящие – вводящие, которые ставят определенную задачу;
* исходящие – выводящие результат деятельности;
* контроль (сверху вниз) – механизмы управления (положения, инструкции и пр);
* механизмы (снизу вверх) – что используется для того, чтобы произвести необходимую работу.

**Модель AS IS** («Как есть») — это описание процессов в том состоянии, в котором они находятся в данный момент. Она фиксирует, как сейчас устроены процессы в организации. Чтобы создать AS IS, изучают документы и регламенты, наблюдают за процессом в реальном времени и фиксируют это в схемах, проводят интервью со всеми, кто участвует в процессе.

**Модель TO BE** («Как должно быть») — это модель для описания бизнес-процессов, которая помогает представить идеальное или улучшенное состояние организации и работы всех её подразделений. С её помощью отражают, как устранить недостатки, оптимизировать процессы и внедрить улучшения. Для построения TO BE нужны данные из AS IS — определяют «зоны роста», выбирают те, которых нужно достичь в первую очередь.

Чаще всего эти модели применяют для оптимизации бизнес-процессов в компании.

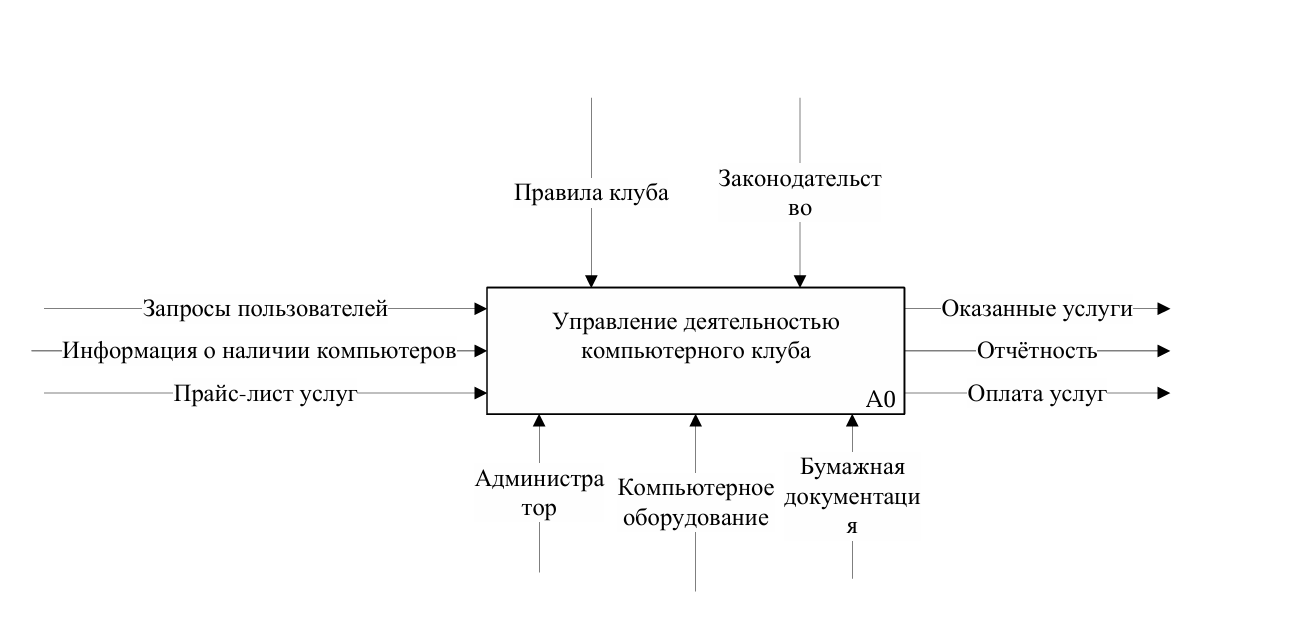


Рис.1.3.1. IDEF(0) AS IS

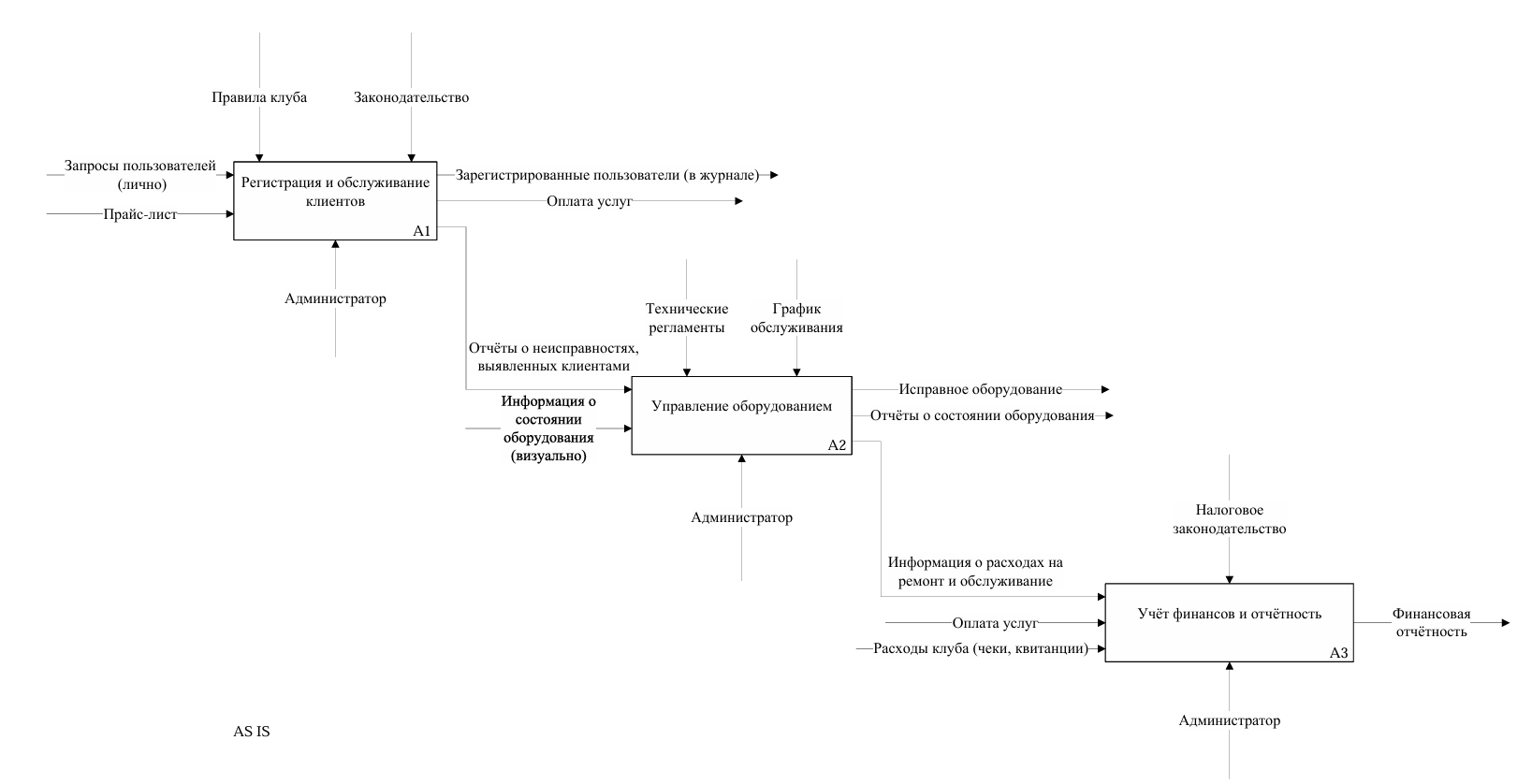


Рис.1.3.2. IDEF(0) AS IS

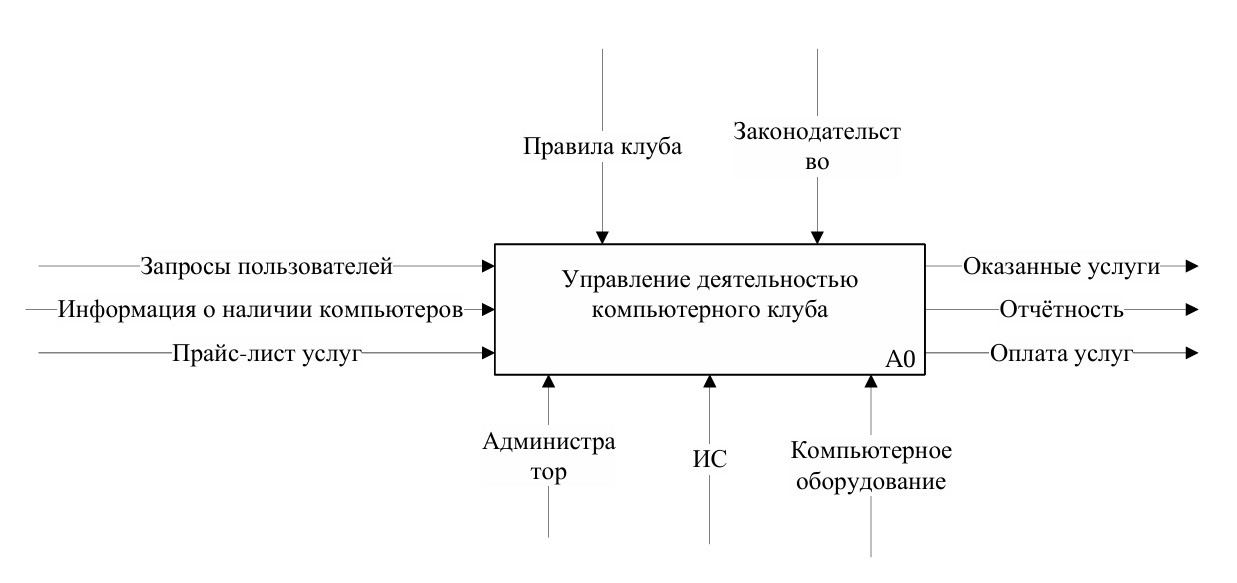


Рис.1.3.3. IDEF(0) TO BE

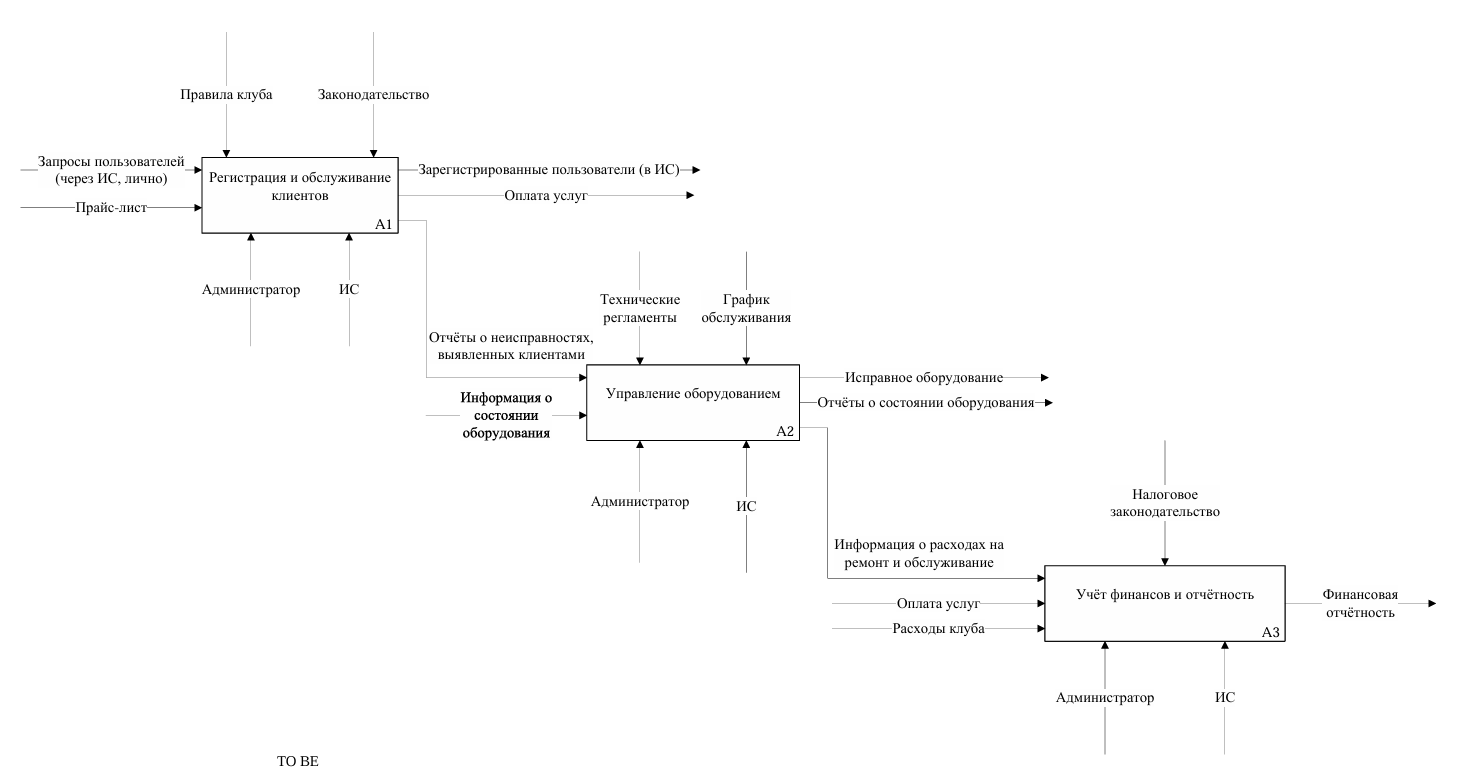


Рис.1.3.4. IDEF(0) TO BE

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ И ВЫБОР ИНСТРУМЕНТАРИЯ

## 2.1 Разработка диаграммы ERD

Характеристика диаграмм «сущность — связь». Данная диаграмма (ER-модель данных) обеспечивает стандартный способ определения данных и отношений между ними. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Диаграммы «сущность - связь» в отличие от функциональных диаграмм определяют спецификации структур данных программного обеспечения.

Базовыми понятиями ER-модели данных (ER — Entity — Relationship) являются сущность, атрибут и связь.

Сущность — это класс однотипных реальных или абстрактных объектов (людей, событий, состояний, предметов и т.п.), информация о которых имеет существенное значение для рассматриваемой предметной области. Структурой данных называют совокупность правил и ограничений, которые отражают связи, существующие между отдельными частями (элементами) данных

Каждая сущность должна иметь:

* ﻿﻿уникальное имя;
* ﻿﻿один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;
* ﻿﻿один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности.

Атрибут – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут, таким образом, представляет собой некоторый тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов. Экземпляр атрибута — определенная характеристика конкретного экземпляра сущности.

Атрибуты делятся на ключевые, т. е. входящие в состав уникального идентификатора ключа, и описательные — прочие.

Первичный ключ — это атрибут или совокупность атрибутов и (или) связей, предназначенная для уникальной идентификации каждого экземпляра сущности (совокупность признаков, позволяющих идентифицировать объект). Ключевые атрибуты помещают в начало списка и помечают символом «#».

Описательные атрибуты могут быть обязательными или необязательными.

Обязательные атрибуты для каждой сущности всегда имеют конкретное значение, необязательные могут быть не определены. Обязательные и необязательные описательные атрибуты помечают символами «\*» и «°» соответственно.

Связь – это, отношение одной сущности к другой или к самой себе. Каждая связь может иметь одну из двух модальностей связей. Если любой экземпляр одной сущности связан хотя бы с одним экземпляром другой сущности, то связь является обязательной. Необязательная cвязь представляет собой условное отношение между сущностями.



Рисунок 2.1 Диаграмма ERD

## 2.2 Словарь данных

**Пользователи**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
|  | name | VARCHAR | YES | Имя |
|  | phone | VARCHAR | NO | номер |
|  | email | VARCHAR | NO | почта |
|  | balance | DECIMAL | NO | баланс |
|  | registered\_at | DATETIME | NO | Дата регистрации |

**Сотрудники**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
|  | name | VARCHAR | YES | Имя |
|  | role | INT | NO | Роль |
|  | username | NVARCHAR | YES | Логин |
|  | password | NVARCHAR | YES | Пароль |

**Залы**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
|  | name | VARCHAR | YES | Название |

**Тарифы**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
|  | name | VARCHAR | YES | Название |
|  | price\_per\_hour | DECIMAL | YES | Цена за час |

**Зал - тариф**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
| FK | hall\_id | INT | YES | Внешний ключ |
| FK | tariff\_id | INT | YES | Внешний ключ |

**Компьютеры**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
|  | name | NVARCHAR | YES | Название ПК |
|  | specs | NVARCHAR | NO | Характеристика |
|  | status | NVARCHAR | YES | Статус |
| FK | hall\_id | INT | YES | Внешний ключ |
| FK | tariff\_id | INT | YES | Внешний ключ |

**Сеансы**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
| FK | user\_id | INT | YES | Внешний ключ |
| FK | computer\_id | INT | YES | Внешний ключ |
| FK | tariff\_id | INT | YES | Внешний ключ |
|  | start\_time | DATETIME | YES | Время начала сессии |
|  | end\_time | DATETIME | NO | Время окончания сессии |
|  | total\_price | DECIMAL | YES | Итоговая стоимость |
| FK | Employee\_id | INT | NO | Внешний ключ |

**Бронирования**

| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| --- | --- | --- | --- | --- |
| PK | id | INT | YES | Уникальный идентификатор |
| FK | user\_id | INT | YES | Внешний ключ |
| FK | computer\_id | INT | YES | Внешний ключ |
| FK | start\_time | DATETIME | YES | Внешний ключ |
| FK | end\_time | DATETIME | YES | Внешний ключ |
|  | status | NVARCHAR | YES | Статус |
| FK | employee\_id | INT | NO | Внешний ключ |

## СУБД SQL Server

SQL SERVER — это система управления базами данных, в работе с которой используется язык программирования SQL. О принципах работы и основных видах серверов баз данных SQL SERVER будет рассказано в представленной статье.

Особенности функционирования SQL SERVER

СУБД SQL SERVER используются для создания, размещения, хранения и управления реляционными (табличными) базами данных на специальных серверах или в облаке. Они работают через настольные приложения и WEB-сайты. К основным преимуществам их функционирования относятся:

* высокоскоростной доступ к данным, обеспечиваемый надежной клиент-серверной архитектурой СУБД;
* простота работы и администрирования, обусловленные понятной структурой языка программирования SQL;
* безопасность хранения информации в БД - благодаря возможности шифрования данных и резервного копирования.

Специфика работы сервера базы данных SQL SERVER заключается в транзакционной обработке данных. Это означает, что по каждому запросу от СУБД обрабатывается и сохраняется небольшое количество информации.

Применение SQL SERVER позволяет автоматизировать решение различных бизнес-задач, поддерживать проведение аналитики данных в режиме онлайн, отслеживать направление ресурсов СУБД, управлять транзакциями (операциями по обработке данных).

Одним из ключевых компонентов SQL Server является его архитектура, которая включает в себя несколько слоев. На самом нижнем уровне находится система хранения данных, которая отвечает за физическое хранение информации на диске. Над этим уровнем располагается система управления памятью, обеспечивающая эффективное использование ресурсов. Также в архитектуре SQL Server присутствует компонент обработки запросов, который интерпретирует SQL-запросы и выполняет необходимые операции над данными.

SQL Server предлагает множество инструментов для администрирования и разработки баз данных. Одним из наиболее популярных инструментов является SQL Server Management Studio (SSMS), который предоставляет пользователям удобный интерфейс для работы с базами данных. SSMS позволяет создавать и изменять структуры баз данных, выполнять запросы, а также управлять пользователями и правами доступа.

Безопасность данных является одной из главных задач любой СУБД, и SQL Server предлагает множество функций для обеспечения защиты информации. Это включает в себя шифрование данных, управление доступом на основе ролей и аудит действий пользователей. Эти механизмы помогают предотвратить несанкционированный доступ к данным и обеспечивают их целостность.

## 2.4 Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, [база данных SQL Azure, управляемый экземпляр SQL Azure](https://learn.microsoft.com/ru-ru/azure/azure-sql/database/sql-database-paas-overview)[,](https://learn.microsoft.com/ru-ru/azure/azure-sql/managed-instance/sql-managed-instance-paas-overview)[SQL Server на виртуальной машине](https://learn.microsoft.com/ru-ru/azure/azure-sql/virtual-machines/windows/sql-server-on-azure-vm-iaas-what-is-overview) Azure и [Azure Synapse Analytics](https://learn.microsoft.com/ru-ru/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-overview-what-is/). Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

Одной из ключевых особенностей SSMS является возможность работы с различными версиями SQL Server. Пользователи могут подключаться к локальным и удаленным серверам, управлять базами данных, выполнять запросы и анализировать результаты. Интерфейс SSMS интуитивно понятен и предоставляет доступ ко всем основным функциям СУБД через меню и панели инструментов.

SSMS поддерживает язык SQL, что позволяет пользователям писать и выполнять запросы к базе данных. В редакторе запросов можно использовать подсветку синтаксиса, автозаполнение и другие функции, которые делают процесс написания кода более удобным и эффективным. Кроме того, SSMS позволяет сохранять запросы в виде скриптов, что облегчает повторное использование кода в будущем.

SSMS предлагает различные механизмы для управления доступом к данным. Администраторы могут создавать роли и назначать им права доступа, что позволяет гибко настраивать уровень доступа для различных пользователей. Это помогает защитить конфиденциальную информацию и предотвратить несанкционированный доступ к данным.

Инструмент также поддерживает аудит действий пользователей, что позволяет отслеживать изменения в базе данных и выявлять потенциальные угрозы безопасности. Аудит может быть настроен на уровне серверов или отдельных баз данных, что обеспечивает дополнительный уровень защиты.

SSMS интегрируется с другими продуктами Microsoft, такими как Visual Studio и Azure Data Studio. Это позволяет разработчикам использовать знакомые инструменты для создания приложений, работающих с SQL Server. Кроме того, SSMS поддерживает различные форматы экспорта данных, что облегчает интеграцию с другими системами и приложениями.

С помощью SSMS можно также создавать резервные копии баз данных и восстанавливать их в случае необходимости. Это важная функция для обеспечения надежности хранения данных и защиты от потерь информации.

# ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

## 3.1 Разработка базы данных

Согласно техническому заданию, разработаем базу данных «Компьютерный клуб», которая содержит информацию о клиентах, сотрудниках, компьютерах, залах, тарифах, бронированиях и сеансах.

-- Создаём базу данных

CREATE DATABASE ComputerClubDB;

GO

-- Используем созданную базу

USE ComputerClubDB;

GO

-- Создаём таблицы

CREATE TABLE Users (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100) NOT NULL,

phone NVARCHAR(20),

email NVARCHAR(100),

balance DECIMAL(10, 2) DEFAULT 0,

registered\_at DATETIME DEFAULT GETDATE()

);

CREATE TABLE Employees (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100) NOT NULL,

role NVARCHAR(50),

username NVARCHAR(50) NOT NULL UNIQUE,

password\_hash NVARCHAR(255) NOT NULL

);

CREATE TABLE Halls (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100) NOT NULL,

description NVARCHAR(MAX) NULL

);

CREATE TABLE Tariffs (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100) NOT NULL,

price\_per\_hour DECIMAL(10, 2) NOT NULL

);

CREATE TABLE Hall\_Tariffs (

id INT IDENTITY(1,1) PRIMARY KEY,

hall\_id INT NOT NULL,

tariff\_id INT NOT NULL,

CONSTRAINT FK\_HallTariff\_Hall FOREIGN KEY (hall\_id) REFERENCES Halls(id) ON DELETE CASCADE,

CONSTRAINT FK\_HallTariff\_Tariff FOREIGN KEY (tariff\_id) REFERENCES Tariffs(id) ON DELETE CASCADE,

CONSTRAINT UQ\_HallTariff UNIQUE (hall\_id, tariff\_id)

);

CREATE TABLE Computers (

id INT IDENTITY(1,1) PRIMARY KEY,

name NVARCHAR(100) NOT NULL,

specs NVARCHAR(255) NULL,

status NVARCHAR(20) DEFAULT 'available' CHECK (status IN ('available', 'occupied', 'maintenance')),

hall\_id INT NOT NULL,

tariff\_id INT NOT NULL,

CONSTRAINT FK\_Computer\_Hall FOREIGN KEY (hall\_id) REFERENCES Halls(id) ON DELETE CASCADE,

CONSTRAINT FK\_Computer\_Tariff FOREIGN KEY (tariff\_id) REFERENCES Tariffs(id) ON DELETE NO ACTION

);

CREATE TABLE Sessions (

id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT NOT NULL,

computer\_id INT NOT NULL,

tariff\_id INT NOT NULL,

start\_time DATETIME NOT NULL,

end\_time DATETIME NULL,

total\_price DECIMAL(10, 2) NULL,

employee\_id INT NULL,

CONSTRAINT FK\_Session\_User FOREIGN KEY (user\_id) REFERENCES Users(id) ON DELETE CASCADE,

CONSTRAINT FK\_Session\_Computer FOREIGN KEY (computer\_id) REFERENCES Computers(id) ON DELETE CASCADE,

CONSTRAINT FK\_Session\_Tariff FOREIGN KEY (tariff\_id) REFERENCES Tariffs(id) ON DELETE NO ACTION,

CONSTRAINT FK\_Session\_Employee FOREIGN KEY (employee\_id) REFERENCES Employees(id) ON DELETE SET NULL

);

CREATE TABLE Bookings (

id INT IDENTITY(1,1) PRIMARY KEY,

user\_id INT NOT NULL,

computer\_id INT NOT NULL,

start\_time DATETIME NOT NULL,

end\_time DATETIME NOT NULL,

status NVARCHAR(20) DEFAULT 'active' CHECK (status IN ('active', 'cancelled', 'completed')),

employee\_id INT NULL,

CONSTRAINT FK\_Booking\_User FOREIGN KEY (user\_id) REFERENCES Users(id) ON DELETE CASCADE,

CONSTRAINT FK\_Booking\_Computer FOREIGN KEY (computer\_id) REFERENCES Computers(id) ON DELETE CASCADE,

CONSTRAINT FK\_Booking\_Employee FOREIGN KEY (employee\_id) REFERENCES Employees(id) ON DELETE SET NULL

);

Согласно по техническому заданию, внесли данные в таблицы:

USE ComputerClubDB;

GO

-- Вставляем пользователей

INSERT INTO Users (name, phone, email, balance, registered\_at) VALUES

('Иван Иванов', '1234567890', 'ivan@example.com', 1500.00, GETDATE()),

('Мария Петрова', '0987654321', 'maria@example.com', 1200.00, GETDATE()),

('Алексей Смирнов', '1112223333', 'aleksey@example.com', 1000.00, GETDATE()),

('Ольга Кузнецова', '2223334444', 'olga@example.com', 800.00, GETDATE()),

('Дмитрий Васильев', '3334445555', 'dmitry@example.com', 500.00, GETDATE());

-- Вставляем сотрудников

INSERT INTO Employees (name, role, username, password\_hash) VALUES

('Администратор', 'Админ', 'admin', HASHBYTES('SHA2\_256', 'adminpass')),

('Руководитель', 'Руководитель', 'manager', HASHBYTES('SHA2\_256', 'managerpass')),

('Оператор 1', 'Оператор', 'operator1', HASHBYTES('SHA2\_256', 'operator1pass')),

('Оператор 2', 'Оператор', 'operator2', HASHBYTES('SHA2\_256', 'operator2pass')),

('Оператор 3', 'Оператор', 'operator3', HASHBYTES('SHA2\_256', 'operator3pass'));

-- Вставляем залы

INSERT INTO Halls (name, description) VALUES

('Стандарт', 'ПК: Intel i3-12100, NVIDIA RTX 4060, 16GB RAM'),

('ВИП', 'ПК: Intel i7-13700K, NVIDIA RTX 4070, 32GB RAM'),

('Буткемп', 'ПК: Intel i7-13700K, NVIDIA RTX 4070, 32GB RAM');

-- Вставляем тарифы

INSERT INTO Tariffs (name, price\_per\_hour) VALUES

('Обычный', 170.00),

('ВИП и Буткемп', 220.00),

('Ночной', 500.00);

-- Вставляем связи Hall\_Tariffs

INSERT INTO Hall\_Tariffs (hall\_id, tariff\_id) VALUES

(1, 1), -- Стандарт - Обычный

(2, 2), -- ВИП - ВИП и Буткемп

(3, 2), -- Буткемп - ВИП и Буткемп

(1, 3), -- Стандарт - Ночной

(2, 3), -- ВИП - Ночной

(3, 3); -- Буткемп - Ночной

-- Вставляем компьютеры

INSERT INTO Computers (name, specs, status, hall\_id, tariff\_id) VALUES

('PC-01', 'Intel i3-12100, NVIDIA RTX 4060, 16GB RAM', 'available', 1, 1),

('PC-02', 'Intel i3-12100, NVIDIA RTX 4060, 16GB RAM', 'available', 1, 1),

('PC-03', 'Intel i7-13700K, NVIDIA RTX 4070, 32GB RAM', 'available', 2, 2),

('PC-04', 'Intel i7-13700K, NVIDIA RTX 4070, 32GB RAM', 'available', 2, 2),

('PC-05', 'Intel i7-13700K, NVIDIA RTX 4070, 32GB RAM', 'available', 3, 2);

-- Вставляем сеансы

INSERT INTO Sessions (user\_id, computer\_id, tariff\_id, start\_time, end\_time, total\_price, employee\_id) VALUES

(1, 1, 1, DATEADD(hour, -5, GETDATE()), DATEADD(hour, -4, GETDATE()), 170.00, 1),

(2, 2, 1, DATEADD(hour, -3, GETDATE()), DATEADD(hour, -2, GETDATE()), 170.00, 2),

(3, 3, 2, DATEADD(hour, -4, GETDATE()), DATEADD(hour, -3, GETDATE()), 220.00, 3),

(4, 4, 2, DATEADD(hour, -2, GETDATE()), DATEADD(hour, -1, GETDATE()), 220.00, 4),

(5, 5, 2, DATEADD(hour, -6, GETDATE()), DATEADD(hour, -5, GETDATE()), 220.00, 1);

-- Вставляем бронирования

INSERT INTO Bookings (user\_id, computer\_id, start\_time, end\_time, status, employee\_id) VALUES

(1, 1, DATEADD(hour, 1, GETDATE()), DATEADD(hour, 2, GETDATE()), 'active', 1),

(2, 2, DATEADD(hour, 3, GETDATE()), DATEADD(hour, 4, GETDATE()), 'active', 2),

(3, 3, DATEADD(hour, 5, GETDATE()), DATEADD(hour, 6, GETDATE()), 'cancelled', 3),

(4, 4, DATEADD(hour, 7, GETDATE()), DATEADD(hour, 8, GETDATE()), 'completed', 4),

(5, 5, DATEADD(hour, 9, GETDATE()), DATEADD(hour, 10, GETDATE()), 'active', 1);

Согласно техническому заданию, вывели следующие запросы:

1. **Получить список всех пользователей с их балансом:**

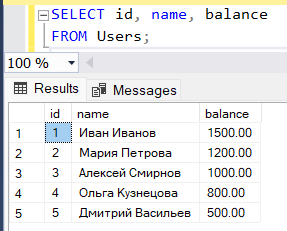


Рисунок 3.1.1 Запрос получения списка всех пользователей с их балансом

1. **Получить список компьютеров и их характеристик в определенном зале:**

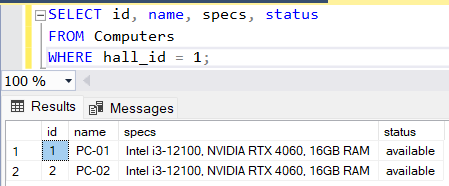


Рисунок 3.1.2 Запрос получения списка компьютеров и их характеристик в определенном зале

1. Пользователи с балансом более 1000 ₽

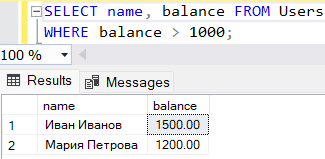


Рисунок 3.1.3 Запрос пользователей с балансом более 1000 ₽

1. Тарифы от дорогих до дешёвых:

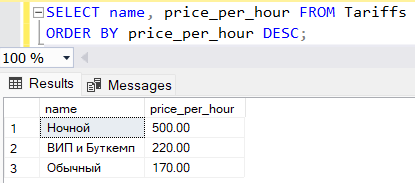


Рисунок 3.1.4 Запрос тарифов от дорогих до дешёвых:

1. Все сотрудники и их роли:

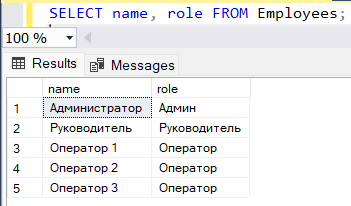


Рисунок 3.1.5 Запрос всех сотрудников и их ролей:

1. Все активные брони:

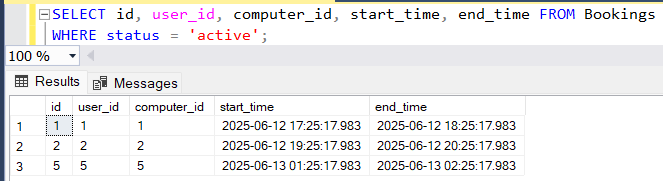


Рисунок 3.1.6 Запрос всех активных бронирований:

1. Сеансы, упорядоченные по времени начала (новые сверху)

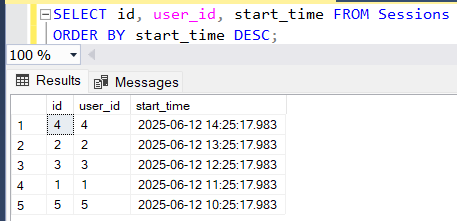


Рисунок 3.1.7 Запрос сеансов, упорядоченных по времени начала

1. Количество бронирований каждого статуса

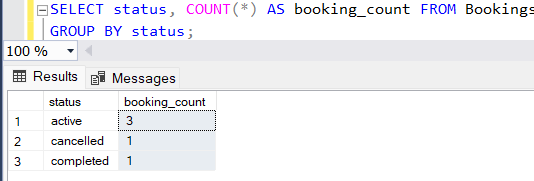


Рисунок 3.1.8 Запрос количества бронирований каждого статуса

1. Пользователи и названия залов, в которых они играли

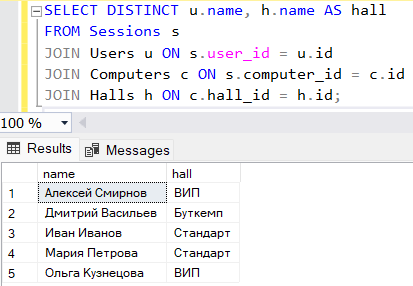


Рисунок 3.1.9 Запрос пользователей и названий залов, в которых они играли

1. Количество сеансов по каждому тарифу

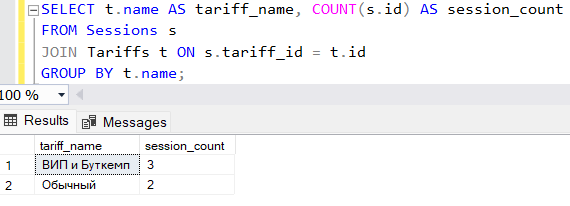


Рисунок 3.1.10 Запрос количества сеансов по каждому тарифу

## 3.2 Введение в безопасность SQL Server

Безопасность SQL Server — это комплекс мер и механизмов, направленных на защиту данных и предотвращение несанкционированного доступа к базе данных. Введение в безопасность SQL Server включает в себя несколько ключевых аспектов:

**1. Аутентификация**

Аутентификация — это процесс проверки подлинности пользователя, который пытается получить доступ к SQL Server. В SQL Server существуют два основных метода аутентификации:

• Windows Authentication: Использует учетные записи Windows для аутентификации пользователей. Это более безопасный метод, так как он интегрируется с Active Directory и использует существующие политики безопасности Windows.

• SQL Server Authentication: Пользователи создают учетные записи непосредственно в SQL Server. Этот метод менее безопасен, так как пароли хранятся в базе данных.

**2. Авторизация**

После успешной аутентификации необходимо определить, какие действия может выполнять пользователь. SQL Server использует модели безопасности на основе ролей:

• Роли: группы пользователей с определенными правами. Существует несколько встроенных ролей, таких как db\_owner, db\_datareader, db\_datawriter и другие.

• Права доступа: определяют, какие операции (например, SELECT, INSERT, UPDATE, DELETE) могут выполнять пользователи или роли на объектах базы данных.

**3. Шифрование**

Шифрование данных помогает защитить конфиденциальную информацию от несанкционированного доступа:

• Transparent Data Encryption (TDE): Позволяет шифровать файлы базы данных на уровне файловой системы.

• Column-level encryption: Позволяет шифровать отдельные столбцы таблиц.

• Always Encrypted: Обеспечивает шифрование данных в состоянии покоя и при передаче, позволяя только клиентскому приложению расшифровывать данные.

**4. Защита от SQL-инъекций**

SQL-инъекции — это распространенный вид атак, при котором злоумышленник внедряет вредоносный SQL-код в запросы к базе данных. Для защиты от таких атак рекомендуется:

• Использовать параметризованные запросы и подготовленные выражения.

• Ограничивать привилегии пользователей.

• Проверять и фильтровать входные данные.

**5. Мониторинг и аудит**

Мониторинг активности пользователей и аудирование действий в базе данных помогают выявлять подозрительное поведение и нарушения безопасности:

• SQL Server Audit: Позволяет отслеживать действия пользователей и изменения в базе данных.

• Extended Events: Инструмент для мониторинга событий и производительности SQL Server.

**6. Обновления и патчи**

Регулярное обновление SQL Server и применение патчей безопасности помогают защищать систему от известных уязвимостей.

**7. Резервное копирование и восстановление**

Создание резервных копий базы данных и регулярное тестирование восстановления данных обеспечивают защиту от потери данных в случае инцидентов или атак.

# 3.3 Управление безопасностью уровня сервера

Создание пользователя для базы данных и присвоение ему роли.

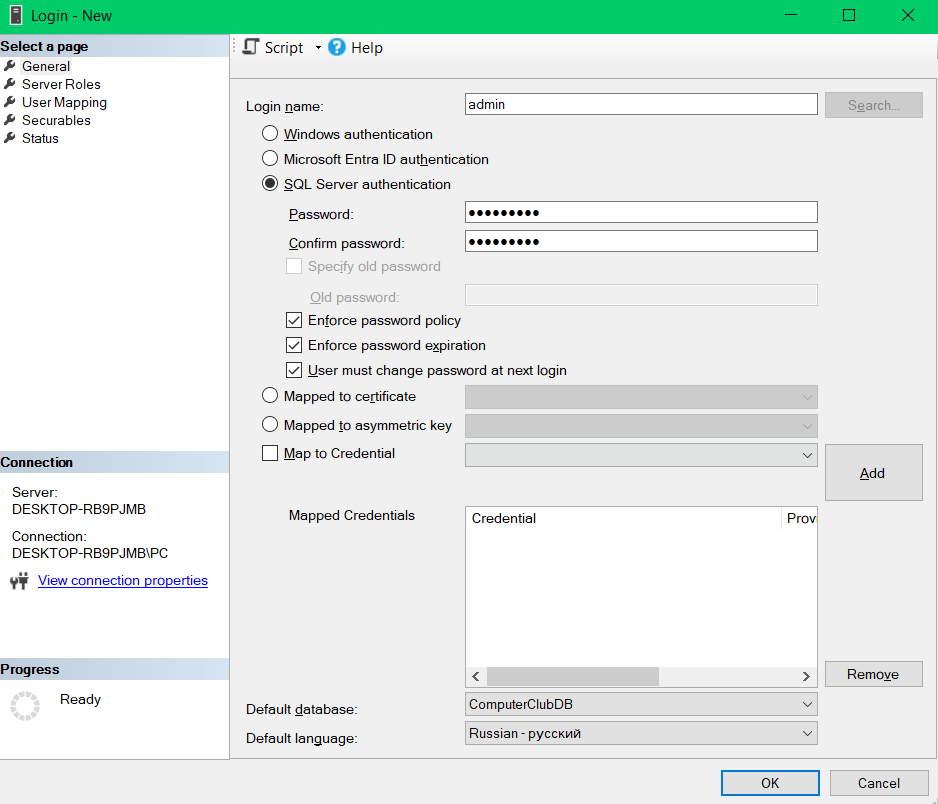


Рисунок 3.3.1 Добавление пользователя

Подключение пользователя к базе данных ComputerClubBD.

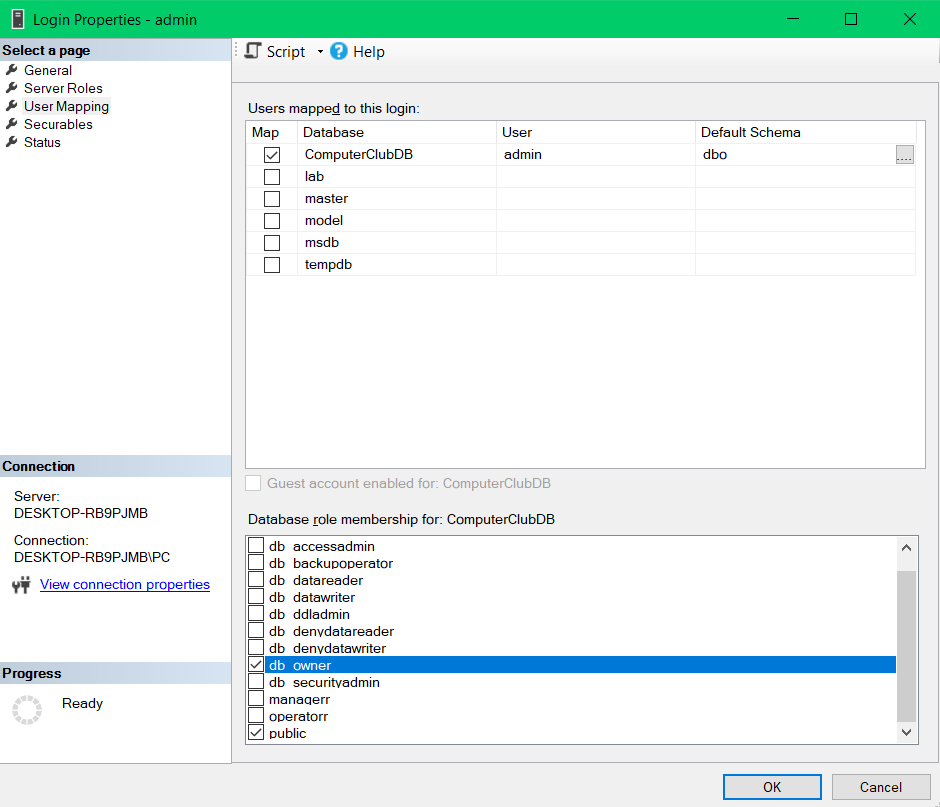


Рисунок 3.3.2 Разграничение прав доступа к базе данных для Администратора

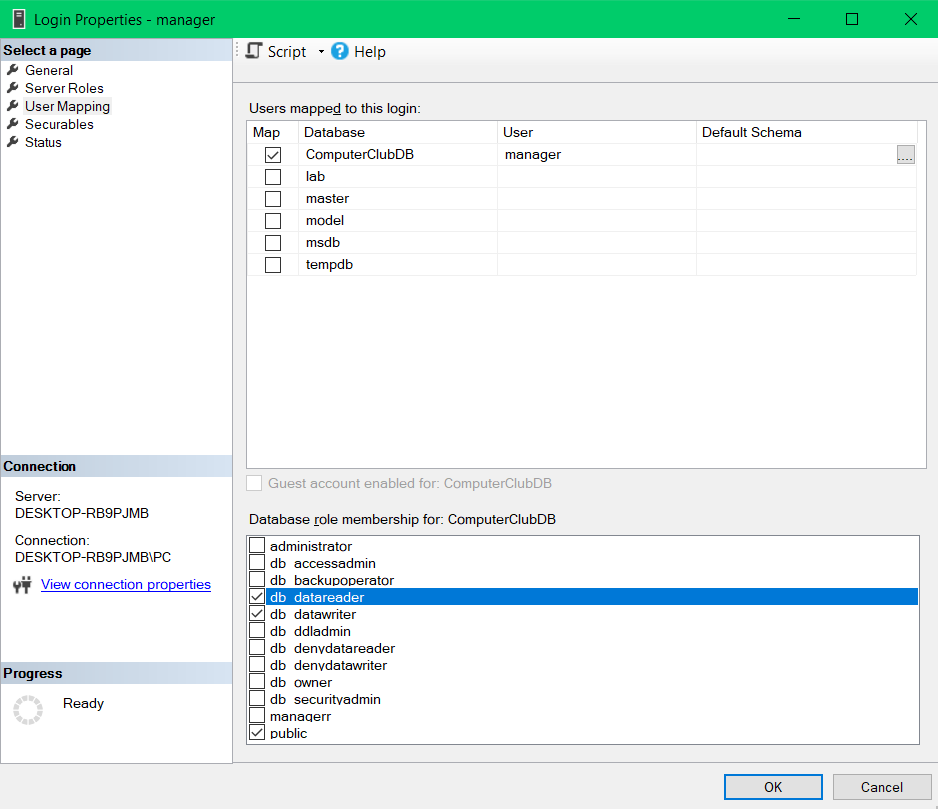


Рисунок 3.3.3 Разграничение прав доступа к базе данных для Менеджера

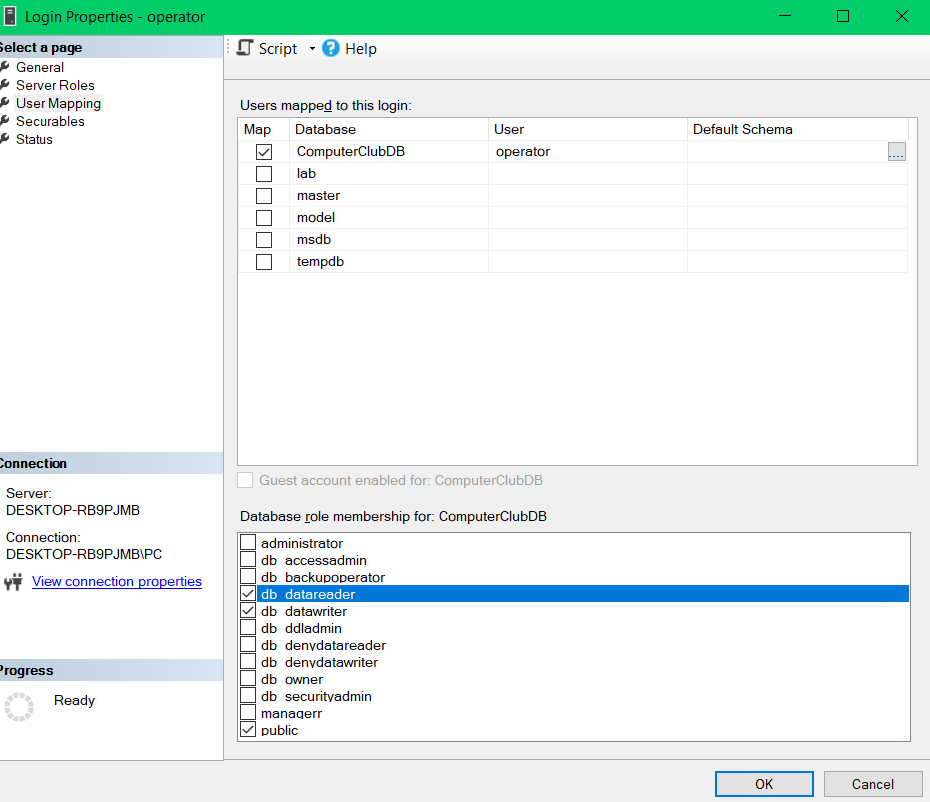


Рисунок 3.3.4 Разграничение прав доступа к базе данных для Оператора

Добавили пользователей БД.



Рисунок 3.3.5 Пользователи базы данных

# 3.4 Управление участниками уровня базы данных

Создание трёх пользователей Админ, Менеджер и Оператор на уровне сервера, для дальнейшего подключения их к базе данных.

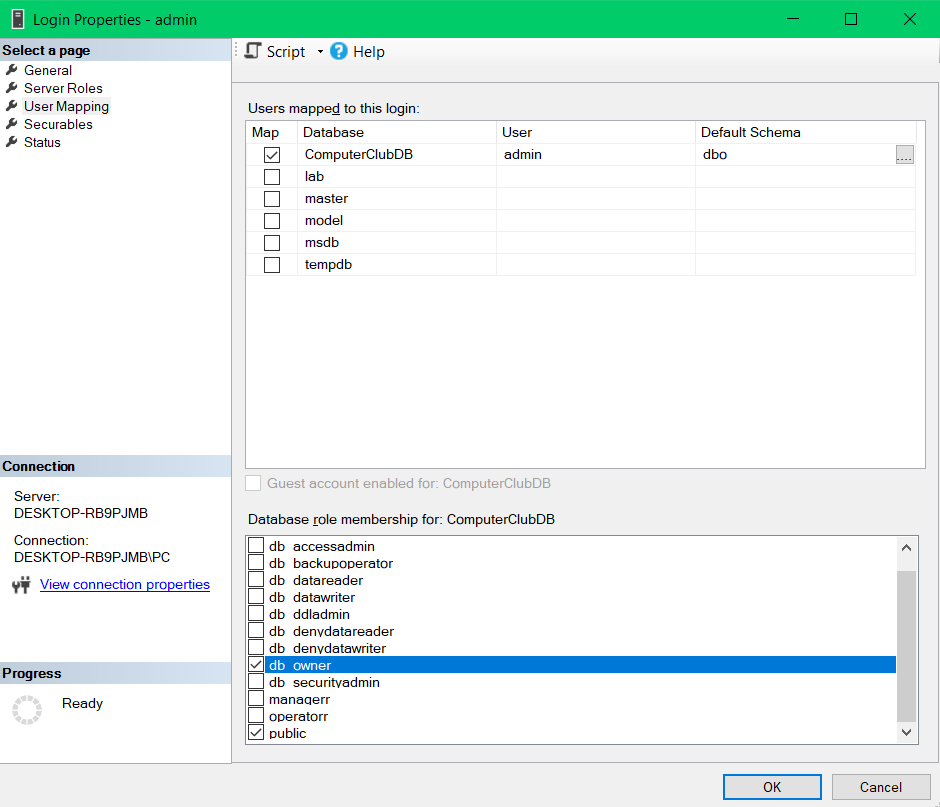


Рисунок 3.4.1 Уровень в базе данных Администратора

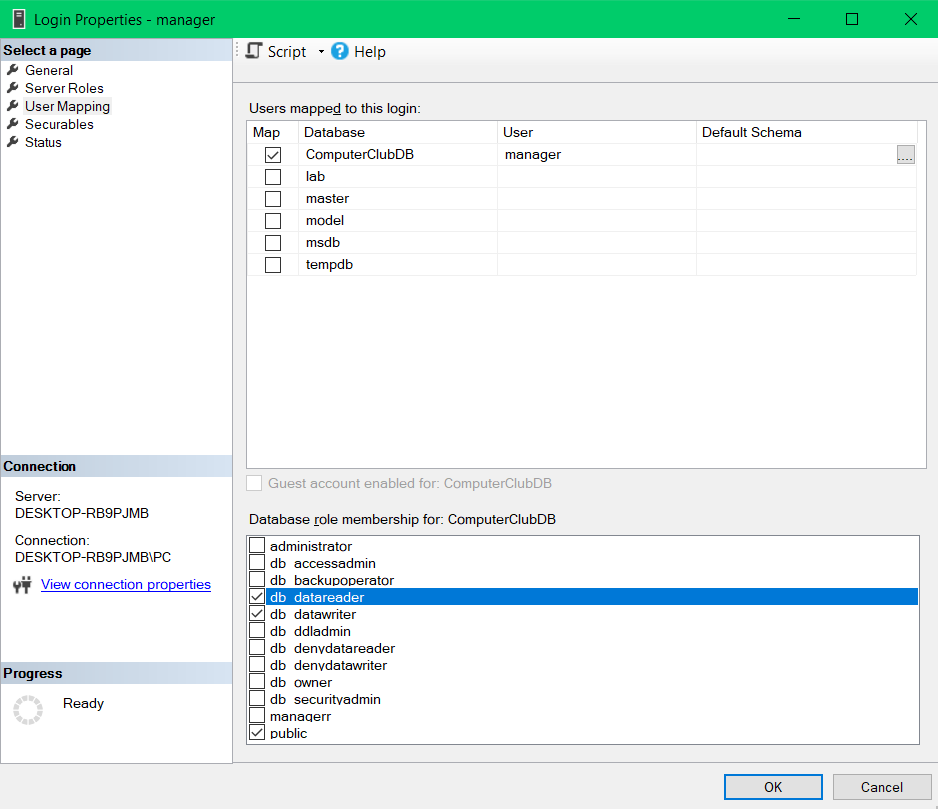


Рисунок 3.4.2 Уровень в базе данных Менеджера

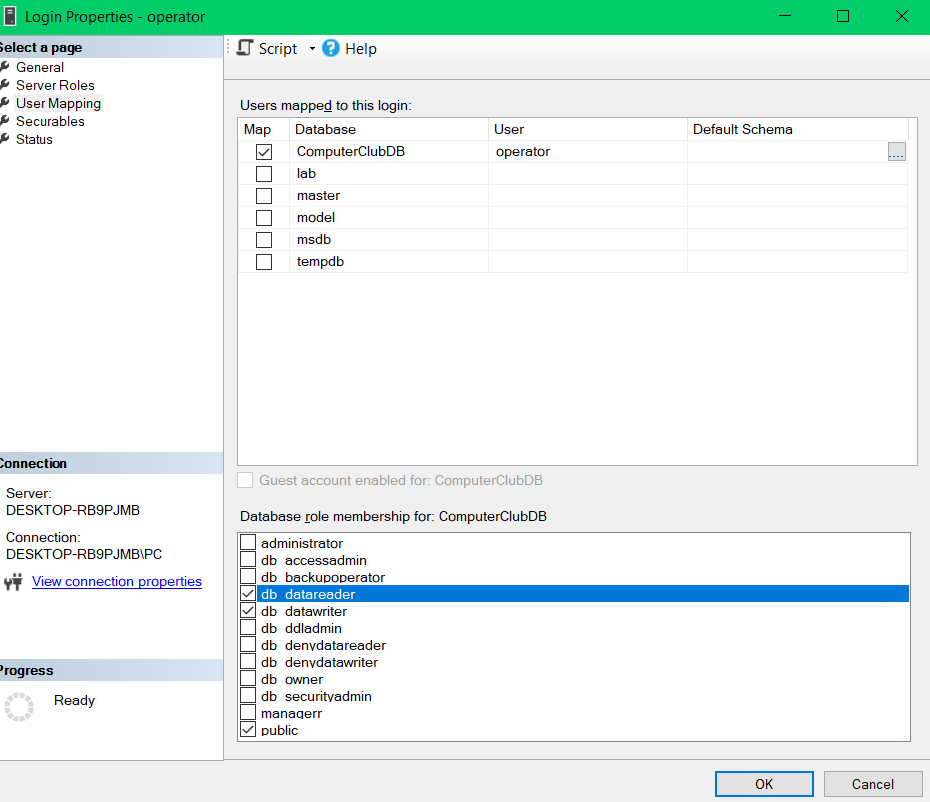


Рисунок 3.4.2 Уровень в базе данных Оператора

## 3.5 Управление разрешениями уровня базы данных

SQL Server использует системную базу данных для поддержания внутренних метаданных. Администратор базы данных сервера должен данных уметь управлять ими, а пользователь правильно ими пользоваться. В базе данных выдали права доступа Админу, Менеджеру и Механику к таблицам и права пользования функциями в данных таблицах.

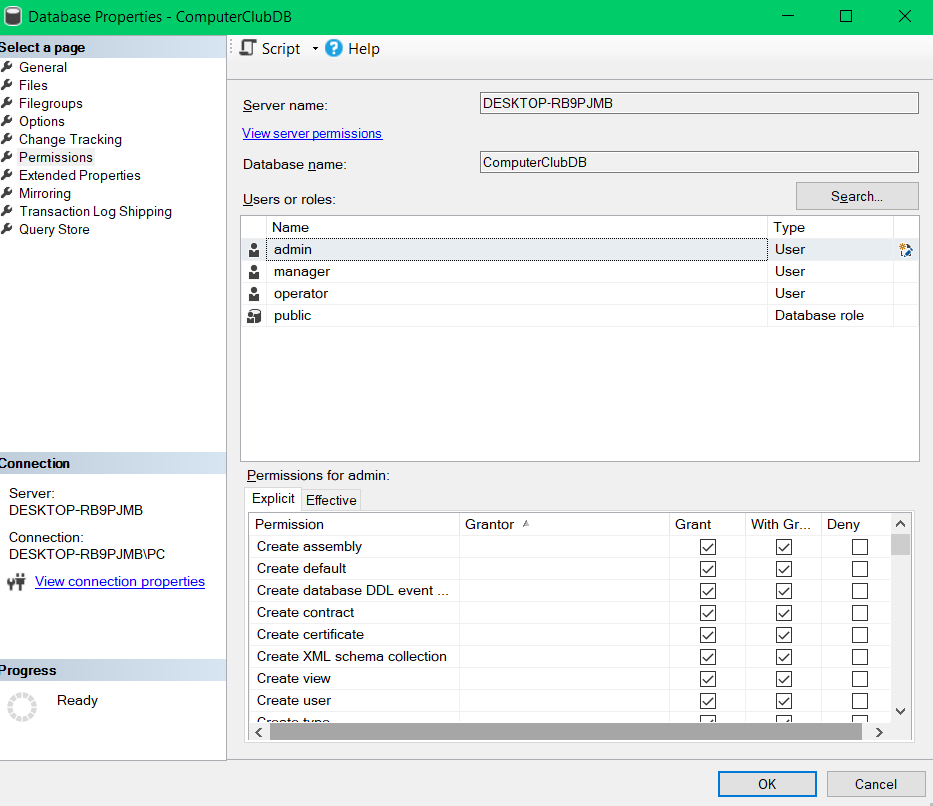


Рисунок 3.5.1 Выдача прав Администратору

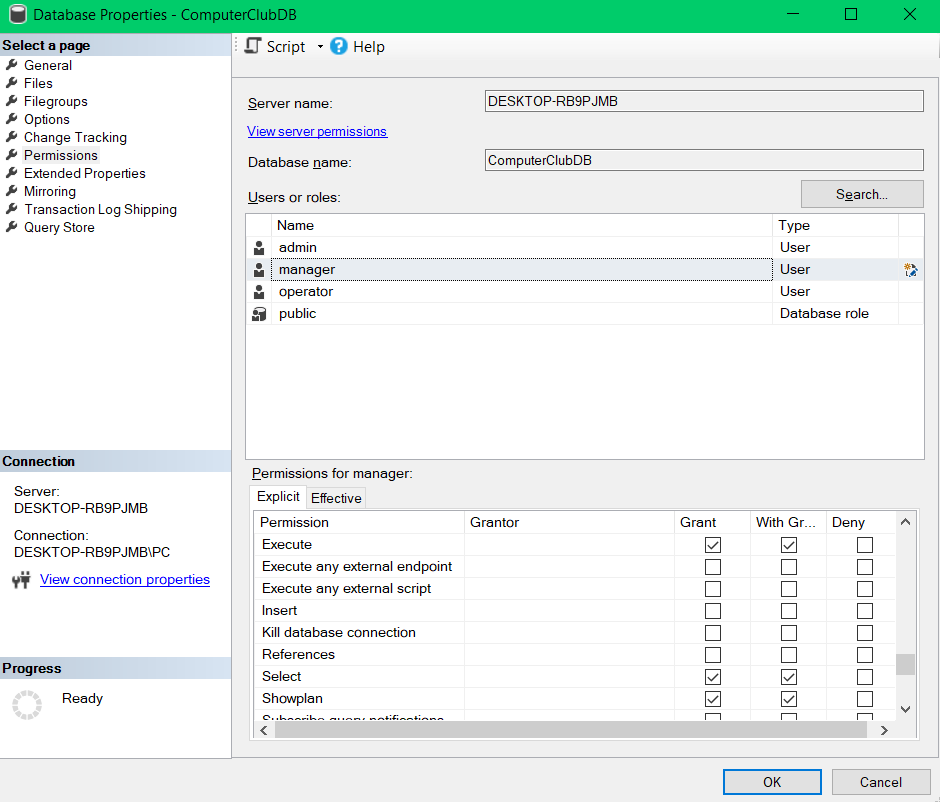


Рисунок 3.5.2 Выдача прав Менеджеру

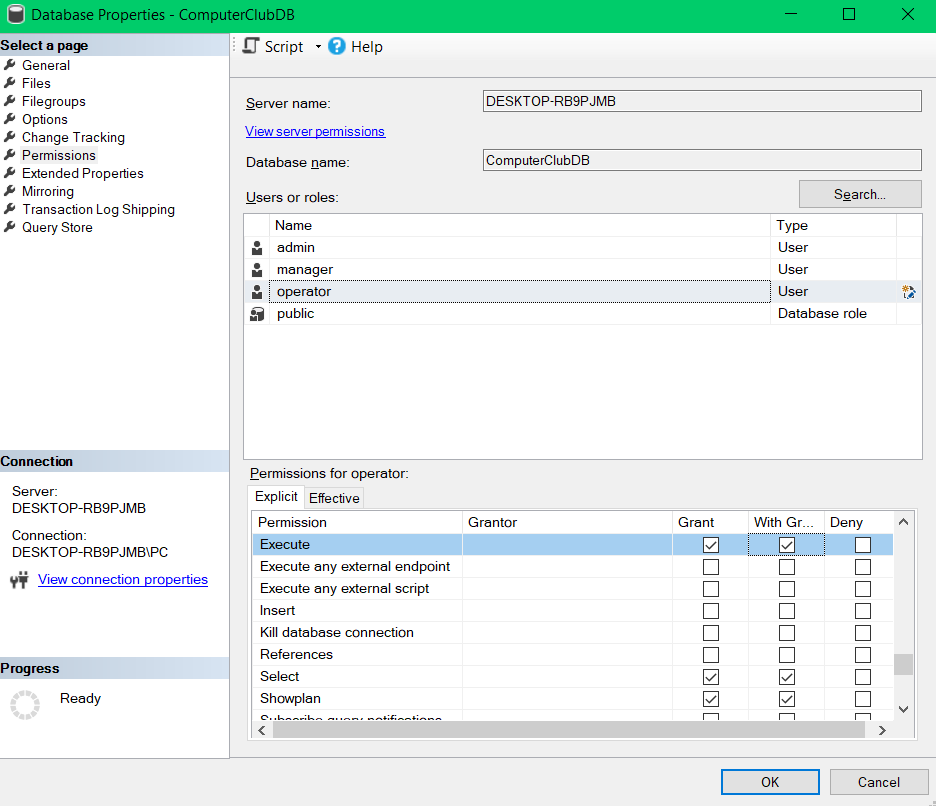


Рисунок 3.5.3 Выдача прав Оператору

## 3.6 Резервное копирование баз данных

BackUp (резервное копирование) — это процесс создания копии данных с целью их сохранения и возможного восстановления в случае их потери, повреждения или удаления. Резервное копирование может быть выполнено как на физических носителях, таких как внешние жесткие диски, USB- накопители, так и B различных облачных хранилищах. Резервное копирование играет важную роль в обеспечении безопасности данных и защите от потери информации.

Резервное копирование базы данных ComputerClub:

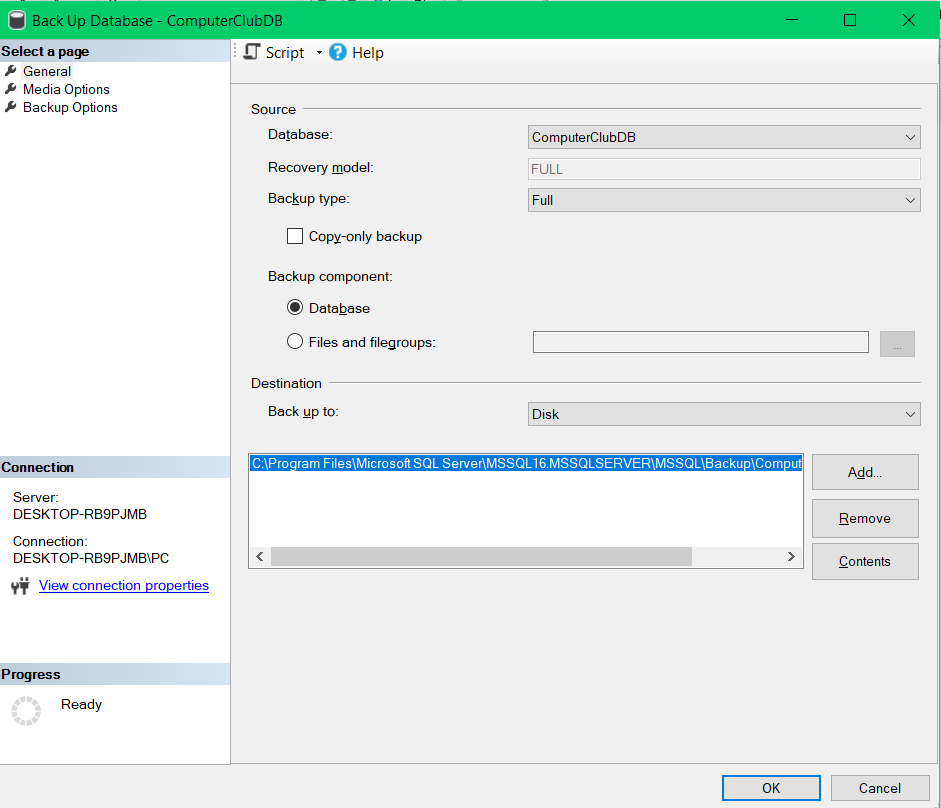


Рисунок 3.6.1 Создание резервной копии базы данных

Восстановление резервной копии базы данных ComputerClubDB:

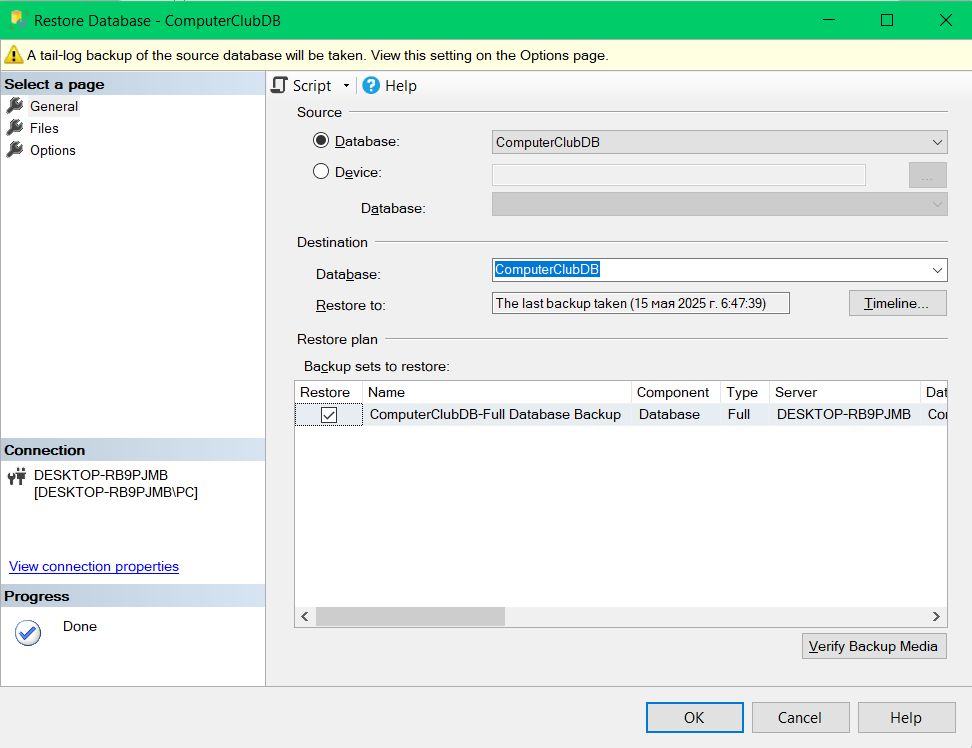


Рисунок 3.6.2 Восстановление резервной копии базы данных

## 3.7 Шифрование данных баз данных

Шифрование данных в базах данных представляет собой важный аспект безопасности, который обеспечивает защиту конфиденциальной информации от несанкционированного доступа. Этот процесс включает преобразование открытых данных в зашифрованный формат, который может быть прочитан только с использованием соответствующего ключа или пароля.

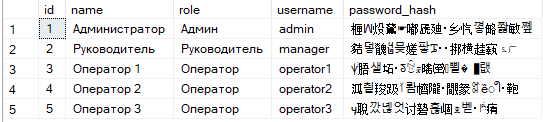


Рисунок 3.7.1. Шифрование базы данных

## ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломного проекта была разработана и спроектирована база данных для компьютерного клуба, которая обеспечивает комплексное и эффективное управление всеми ключевыми процессами клуба. В основе системы лежит продуманная структура данных, включающая основные сущности: клиенты, сотрудники с различными ролями (администраторы, менеджеры, операторы), компьютерное оборудование, залы, тарифные планы, бронирования и сеансы.

Проектируемая база данных обеспечивает надежное хранение и быстрый доступ к информации, необходимой для работы клуба. Использование нормализованных таблиц и установленных связей между ними способствует снижению избыточности данных и повышению их целостности. Внедрение системы позволит автоматизировать процессы учета клиентов, контроля за использованием компьютерного оборудования, а также управления расписанием и бронированиями, что значительно улучшит организацию работы и повысит уровень сервиса.

Особое внимание в проекте уделялось обеспечению безопасности данных, в том числе хранению паролей сотрудников в виде криптографических хешей, что соответствует современным требованиям информационной безопасности. Предусмотрены механизмы разграничения доступа, что позволит эффективно управлять правами пользователей системы и минимизировать риски несанкционированного доступа.

Гибкость и масштабируемость разработанной базы данных позволяют в дальнейшем легко расширять функционал системы, добавлять новые услуги, вводить дополнительные категории пользователей или оборудования, а также интегрировать с другими информационными системами.

Внедрение данной базы данных на практике обеспечит прозрачность и удобство учета, снизит количество ошибок и ускорит выполнение рутинных операций. Это будет способствовать повышению эффективности работы персонала, улучшению взаимодействия с клиентами и увеличению общей конкурентоспособности компьютерного клуба на рынке.

Таким образом, выполненная работа представляет собой надежную и перспективную основу для автоматизации процессов управления компьютерным клубом, что является важным вкладом в развитие современных информационных систем для сферы обслуживания.

Ссылка на репозиторий: [git@github.com:robocopchik/PC-Club-.git](mailto:git@github.com:robocopchik/PC-Club-.git)

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

Электронные ресурсы

1. Сервер баз данных SQL SERVER // Высшая школа бизнеса – Режим доступа: <https://hsbi.hse.ru/articles/server-baz-dannykh-sql-server/>
2. Что такое SQL Server Management Studio (SSMS)? // Microsoft Build – Режим доступа: <https://learn.microsoft.com/ru-ru/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>

Видео – ресурсы

1. Создание ER – диаграммы // Wordskills <https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/>
2. Администрирование SQL Server ч.1 // SQL Developer <https://youtu.be/X508KcPidyk?si=OYzfUpwM34gAuKS1>
3. BackUp/Restore баз данных. Администрирование SQL Server ч.2 // SQL Developer <https://www.youtube.com/watch?v=rGTTDC0psr8&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=2>
4. Администрирование SQL Server ч.3. Login, User. // SQL Developer
5. <https://www.youtube.com/watch?v=ETX3OXonDrg&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=3>
6. Администрирование SQL Server ч.4. Роли. // SQL Developer <https://www.youtube.com/watch?v=_JmzlbA3cp8&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=4>
7. Импорт/Экспорт данных SQL. Администрирование SQL Server ч.5 // SQL Developer <https://www.youtube.com/watch?v=i4BLUNYObP4&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=5>
8. Установка SQL Server. Администрирование SQL Server ч.6 // SQL Developer <https://www.youtube.com/watch?v=F8nG5P3ErIo&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=6>
9. SQL Profiler. Администрирование SQL Server ч.7 // SQL Developer <https://www.youtube.com/watch?v=EMooNYEf7is&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=7>