КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургский колледж информационных технологий

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине «Основы проектирования баз данных»**

**Тема: «Проектирование БД для билетного оператора»**

**Специальность 09.02.07**

**Квалификация Программист**

Выполнил студент гр. 31:

Титова Елизавета

Преподаватель: Женихова Н.В.

Подпись \_\_\_\_\_\_\_\_\_\_\_

Санкт-Петербург 2024

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc169829008)

[**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ** 4](#_Toc169829009)

[**1.1.** **Анализ предметной области** 4](#_Toc169829010)

[**1.2.** **Функциональные возможности системы** 5](#_Toc169829011)

[**1.3.** **Входные и выходные данные** 6](#_Toc169829012)

[**1.4.** **Концептуально-логическая модель** 7](#_Toc169829013)

[**1.5.** **Физическая модель** 8](#_Toc169829014)

[**ПРАКТИЧЕСКАЯ ЧАСТЬ** 9](#_Toc169829015)

[**2.** **Структура базы данных** 9](#_Toc169829016)

[**2.1.** **Создание базы данных** 10](#_Toc169829017)

[**2.2.** **Создание таблиц и ограничений** 11](#_Toc169829018)

[**2.3.** **Создание связей между таблицами** 15](#_Toc169829019)

[**2.4.** **Создание процедур** 16](#_Toc169829020)

[**2.5.** **Создание триггеров** 21](#_Toc169829021)

[**2.6.** **Создание представлений** 24](#_Toc169829022)

[**2.7.** **Создание индексов** 27](#_Toc169829023)

[**2.8.** **Создание пользователей и ролей** 29](#_Toc169829024)

[**2.9.** **Стратегия резервного копирования базы данных** 32](#_Toc169829025)

[**2.10.** **Проверка на целостность БД** 33](#_Toc169829026)

[**ЗАКЛЮЧЕНИЕ** 34](#_Toc169829027)

[**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ** 35](#_Toc169829028)

# **ВВЕДЕНИЕ**

Важность систем управления базами данных трудно переоценить. Они являются главным инструментом для организации и хранения информации, которая используется в бизнесе, научных исследованиях, государственных учреждениях и многих других областях жизни. Без СУБД было бы трудно обеспечить эффективное использование данных, а также обеспечить их безопасность и надежность.

В данном проекте для разработки БД будет использоваться наиболее популярная СУБД MS Access. Программа Microsoft Access представляет собой многофункциональный редактор, который способен предоставить своим пользователям обширный набор операций при создании новых и осуществлении коррекции уже существующих информационных баз.

В целом, базы данных являются не только важным инструментом для сохранения и организации информации, но и критическим элементом современного бизнеса и науки. Они несут стержневую функцию, обеспечивая доступ к информации, помогая в принятии решений и способствуя развитию организаций в целом.

В данной курсовой работе проектирование и разработка базы данных в информационных системах будет рассмотрена на примере проектирования базы данных, содержащей информацию о билетном операторе «conserTickets».

Целью разработки базы данных является упрощение доступа к сведениям о заказах фотостудии, о ее сотрудниках, о клиентах, а также о предоставляемых услугах. Созданная база данных позволит пользователю лучше ориентироваться в вышеперечисленных разделах и в случае надобности вести их учет самостоятельно.

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **Анализ предметной области**

Билетный оператор — это посредник между организатором и аудиторией, который помогает организатору продавать билеты, беря за это комиссию.

Если раньше билеты реализовывались исключительно через кассы руками билетных агентов, то развитие интернета и электронной коммерции изменило эту ситуацию. Что это значит? Интернет в целом и интернет-реклама в частности делают маркетинг любого продукта прозрачным и прогнозируемым, однако растет и уровень необходимых компетенций для занятия этим видом деятельности.

Сейчас чтобы продавать билеты вовсе не нужно идти в кассы. На рынке много организаторов, которые успешно делают это исключительно через интернет и в огромных масштабах: десятки тысяч билетов на один концерт или фестиваль только в электронном виде.

Интернет-реклама и цифровые каналы коммуникации уравнивают возможности продажи билетов между участниками отрасли и делают этот рынок конкурентным. Монополия из-за договоренностей между конкретными организатором и билетными операторами исключена, так как все попадают в условия свободного рынка.

Камнем преткновения в этой истории становятся данные и их интерпретация. На основе данных делаются прогнозы, через них же оптимизируются рекламные кампании и выстраиваются долгосрочные стратегии развития бизнеса в целом, а не только отдельно взятого мероприятия. Данные о пользователях становятся основной и ключевой ценностью для того, кто занимается продажей билетов на событие, и значимость билетных операторов и агентов отходит на второй план. А на первый выходит грамотная маркетинговая стратегия и эквайринг.

Проведя анализ предметной области, можно выделить следующие сущности:

* Заказчики: информация о клиентах, регистрирующих концерты для продажи билетов на них;
* Концерты: информация о зарегистрированных концертах;
* Цена билетов: информация о стоимости разных классов билетов на концерты;
* Билеты: информация о купленных билетах
* Покупатели: информация о покупателях билетов.

Для каждой описанной выше сущности существуют свои атрибуты:

* Заказчики: код заказчика, фамилия, имя, отчество, название компании, название площадки, почта, номер телефона;
* Концерты: код концерта, название концерта, код заказчика, город, дата проведения, время проведения;
* Цена билетов: код концерта, цена входного, цена вип, цена супер-вип, название концерта;
* Билеты: код билета, код концерта, класс билета, цена, код покупателя;
* Покупатели: код покупателя, фамилия, имя, отчество, дата рождения, почта, номер телефона, код билета.

## **Функциональные возможности системы**

* Занесение и просмотр информации о заказчиках;
* Занесение и просмотр информации о концертах;
* Занесение и просмотр информации о цене билетов;
* Занесение и просмотр информации о билетах;
* Занесение и просмотр информации о покупателях;
* Просмотр информации о концертах до определённой даты;
* Просмотр информации о билетах определённой стоимости;
* Занесение и просмотр информации о покупателях билетов на определённый концерт;
* Занесение и просмотр информации о билетах определённого класса;
* Просмотр информации о максимальной цене разных классов билетов;
* Просмотр информации о стоимости всех билетов;
* Просмотр информации о возрасте покупателей;
* Просмотр информации о количестве дней до концерта;
* Удаление клиентов, билетов;
* Изменение номера телефона;

## **Входные и выходные данные**

Входные данные: информация о заказчиках, концертах, билетах, цене билетов и покупателях.

Выходные данные: вывод на печать отчётов о заказчиках, покупателях, вывод информации о стоимости.

## **Концептуально-логическая модель**

Концептуально-логическая модель разрабатываемой базы данных представлена ниже на рисунке 1. В ней присутствуют сущности, их атрибуты и связи между ними.



Рисунок 1–Концептуально-логическая модель

Связи между таблицами устанавливаются следующим образом:

* Таблицы “Концерты” и “Заказчики” – связь один ко многим (1:М)
* Таблицы “Концерты” и “Цена билетов” – связь один к одному (1:1)
* Таблицы “Концерты” и “Билеты” – связь один ко многим (1:М)
* Таблицы “Покупатели” и “Билеты”– связь один ко многим (1:М)

Тип связи “один ко многим” является самым распространенным.

В таком типе несколько строк из зависимой таблицы зависят от одной строки в главной таблице.

## **Физическая модель**

Построение физической модели напрямую зависит от построения концептуально-логической модели. Физическая модель базы данных “Фотостудия” представлена на рисунке 2.

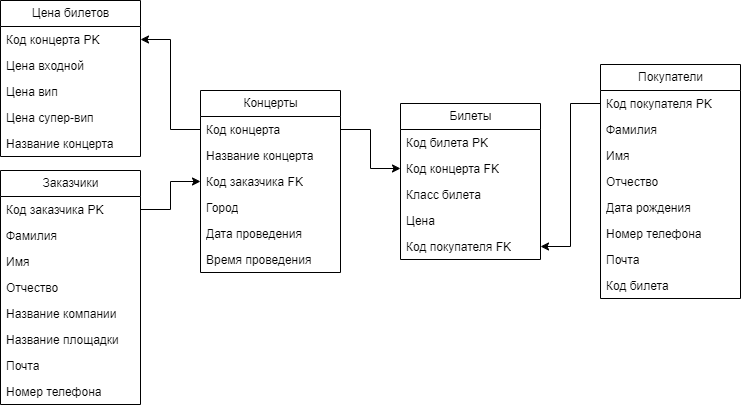


Рисунок 2– Физическая модель БД "Билетный оператор"

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

## **Структура базы данных**

Структура разработанной БД (таблицы, ключи и связи) представляется в виде схемы данных в SQL Server, представленной на рисунке 3.

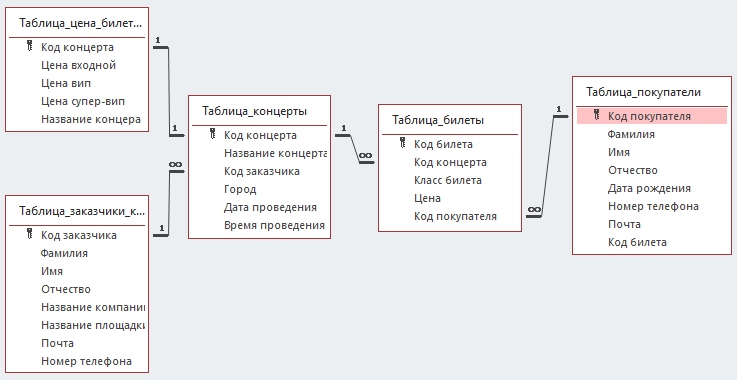


Рисунок 3– Схема данных базы "Билетный оператор"

Все таблицы связаны между собой связью 1 ко Многим кроме таблиц “Цена билетов” и “Концерты”, они связаны связью 1 к 1.

## **Создание базы данных**

Для того, чтобы создать базу данных в СУБД Microsoft SQL Management Studio, необходимо соединиться с ядром СУБД. Далее нужно создать запрос. Запросы в SQL – это наборы команд для работы с реляционными базами данных. Запросы SQL чаще всего делятся на несколько видов:

1. запросы, предназначенные для работы со структурой данных - для создания, описания и модификации БД
2. запросы, используемые непосредственно в работе с данными, с помощью которых можно добавлять, обновлять, сохранять и удалять данные
3. запросы, применяемые для предоставления или отмены прав доступа к БД

Для того чтобы создать базу данных, в запросе пишется текст с рисунка 4. Чтобы выполнить данный код, выделяем его и нажимаем на кнопку выполнить запрос или клавищу F5. После чего подслединяемся к ней для дальнейшей работы.

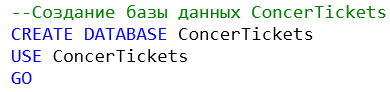


Рисунок 4 – Создание базы данных

## **Создание таблиц и ограничений**

Таблицы в SQL Server — это логически организованные хранилища данных (объектов), представленные в виде строк и столбцов.

В SQL Server предусмотрено разбиение таблиц на типы, каждый из которых предназначен для достижения определённых целей в БД. Вот некоторые типы таблиц:

1. Секционированные. Информация в них поделена секциями (блоками) горизонтально.
2. Временные. Сохраняются в базе данных tempdb и используются для хранения табличных данных как части сложного комплексного скрипта.
3. Производные. Более эффективны в плане производительности и хранятся в оперативной памяти.
4. Системные. Специальный набор таблиц, в котором SQL Server сохраняет информацию, определяющую конфигурацию сервера.
5. Широкие. В таких таблицах применяются разрежённые списки.

Ограничения в базе данных — это специальные элементы, которые устанавливают правила для приемлемых значений в столбцах таблицы.

Основные функции ограничений:

1. Гарантируют уникальность данных.
2. Обеспечивают связанность данных.
3. Гарантируют целостность значений.
4. Защищают от ошибок ввода данных.

Виды ограничений:

1. UNIQUE — гарантирует уникальность значений в столбце или группе столбцов в таблице.
2. CHECK — определяет условия, которым должны удовлетворять значения в столбце.
3. NOT NULL — указывает, что значения в определенном столбце не могут быть равны NULL.
4. FOREIGN KEY — создаёт связь между двумя таблицами.
5. DEFAULT — задаёт значения по умолчанию для столбца.

Создание таблицы «Заказчики» и её ограничений представлено на рисунке 5. Также предусмотрена сортировка на распознавание кирилицы Cyrillic\_General\_BIN.

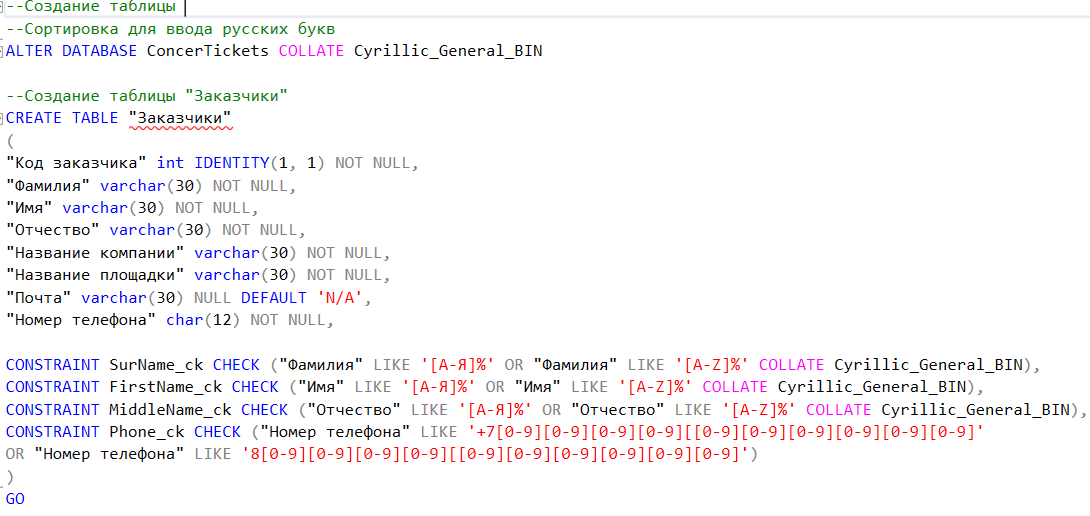


Рисунок 5 – Создание таблицы "Заказчики" и её ограничений

В таблице «Заказчики» поле «Код заказчика» – счётчик. Все поля кроме почты обязательны для заполнения. Значение «Почта» по умолчанию – «N/A», т.е. если это поле не будет заполнено, то вместо null будет это значение.

Ограничение SurName\_ck, FirstName\_ck и MiddleName\_ck устанавливают ввод только букв русского и латинского алфавита для заполнения полей «Фамилия», «Имя» и «Отчество». Ограничение Phone\_ck позволяет вводить в поле только значения формата 11 цифр, начинающихся либо с 8, либо с +7.

Создание таблицы «Цена билетов» представлено на рисунке 6.

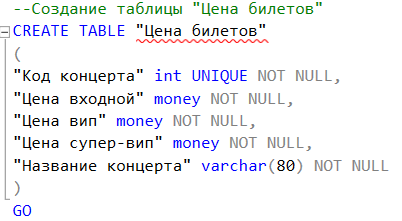


Рисунок 6 – Создание таблицы "Цена билетов"

Поле «Код концерта» – уникальное. Все поля обязательны для заполнения.

Создание таблицы «Концерты» представлено на рисунке 7.

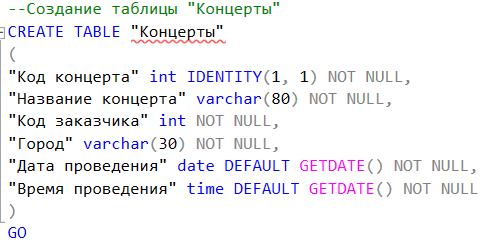


Рисунок 7 – Создание таблицы " Концерты "

В таблице «Концерты» поле «Код концерта» – счётчик. Все поля обязательны для заполнения. Для полей «Время проведения» и «Дата проведения» значение по умолчанию – текущее время или дата при заполнении поля.

Создание таблицы «Билеты» и её ограничений представлено на рисунке 8.

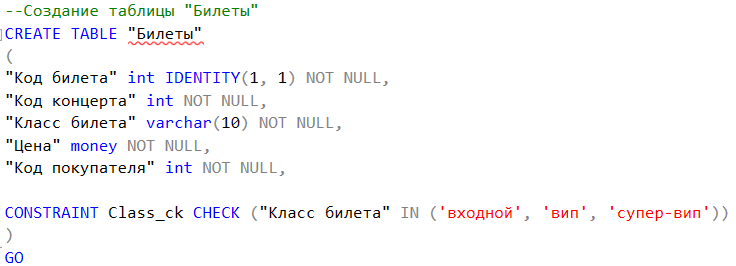


Рисунок 8 – Создание таблицы " Билеты " и её ограничений

В таблице «Билеты» поле «Код билета» – счётчик. Все поля обязательны для заполнения. Ограничение Class\_ck позволяет вводить в поле «Класс билета» значения только из указанного диапазона – «входной», «вип», «супер-вип».

Создание таблицы «Покупатели» и её ограничений представлено на рисунке 9.

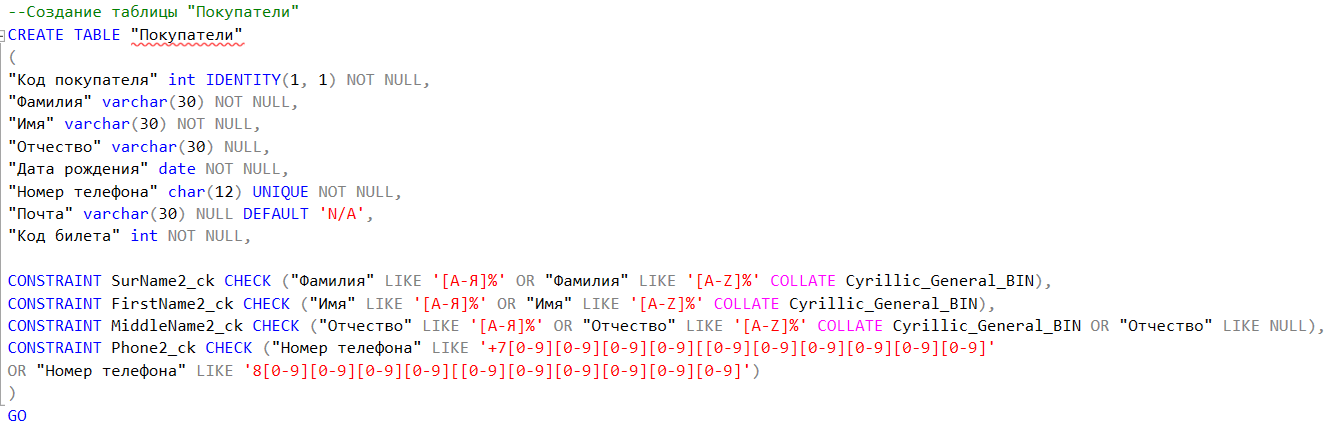


Рисунок 9 – Создание таблицы " Покупатели " и её ограничений

В таблице «Покупатели» поле «Код покупателя» – счётчик. Все поля кроме почты обязательны для заполнения. Значение «Почта» по умолчанию – «N/A», т.е. если это поле не будет заполнено, то вместо null будет это значение. Поле «Номер телефона» уникально.

Ограничение SurName2\_ck, FirstName2\_ck и MiddleName2\_ck устанавливают ввод только букв русского и латинского алфавита для заполнения полей «Фамилия», «Имя» и «Отчество». Ограничение Phone2\_ck позволяет вводить в поле только значения формата 11 цифр, начинающихся либо с 8, либо с +7.

## **Создание связей между таблицами**

Связи – это ассоциации между сущностями разных типов. При выделении связи выделяют главную таблицу и зависимую таблицу. Зависимая таблица зависит от главной. Поля, по которым идет связь между таблицами, называют ключами. Ключи были определены на этапе составления физической модели, и их создание показано на рисунке 10. Ключи тоже считаются ограничениями таблицы и их можно, при желании, определить на этапе создания таблицы.

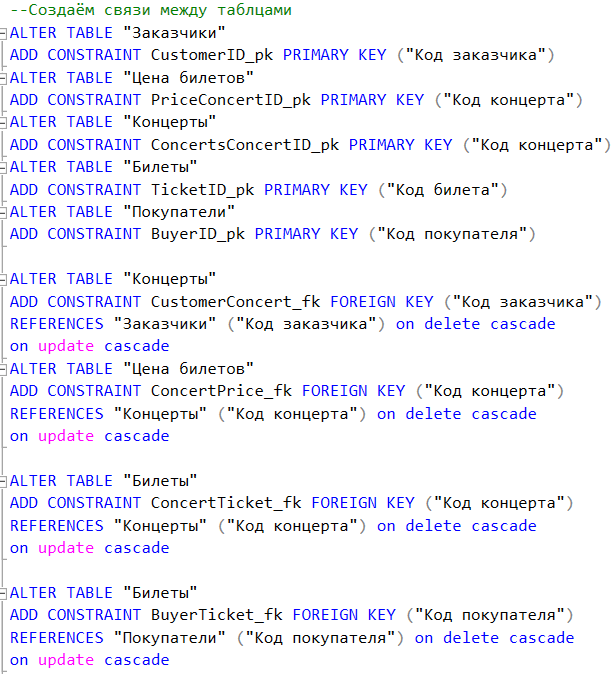


Рисунок 10 – Создание ключей (связей) базы данных

On delete cascade и on update cascade обеспечивают целостность данных в таблицах, автоматически удаляя или изменяя поля из зависимой таблицы при удалении или изменении связанных строк в главной таблице.

## **Создание процедур**

Процедуры представляют собой набор инструкций, которые выполняются как единое целое, они позволяют упростить комплексные операции и вынести их в единый объект. Также процедуры позволяют ограничить доступ к данным в таблицах и тем самым уменьшить вероятность преднамеренных или неосознанных нежелательных действий в отношении этих данных. Также процедуры обычно выполняются быстрее, чем обычные SQL-инструкции.

В отчете будут созданы процедуры на:

1. Ввод данных в базу данных
2. Обновление данных в таблицах базы данных
3. Удаление данных в таблицах базах данных
4. Поиск данных в таблицах базы данных
5. Ввод данных в базу данных

Создание процедуры на вставку новой записи о покупателе в таблицу «Заказчики» показано на рисунке 11.

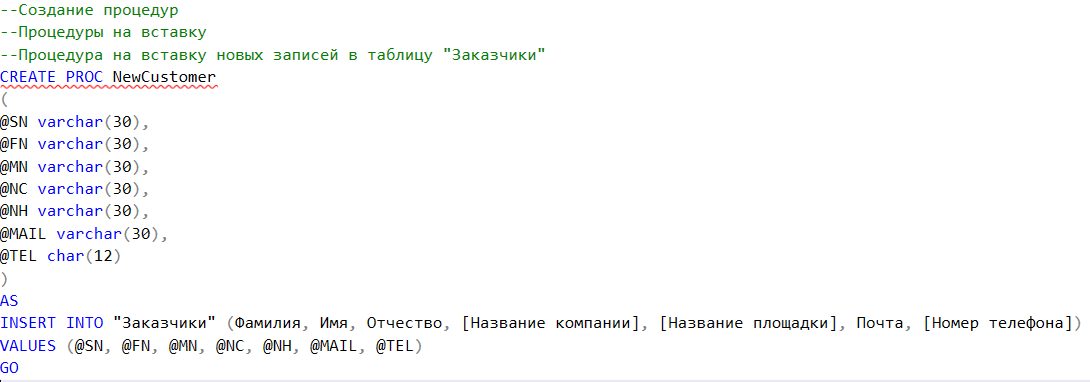


Рисунок 11 – Создание процедуры вставки новой записи в таблицу " Заказчики "

Выполнение процедуры показано на рисунке 12.

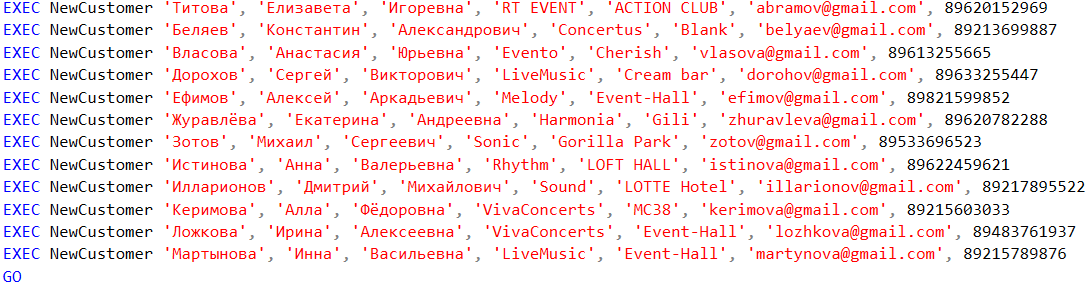


Рисунок 12 – Выполнение процедуры NewCustomer

Результат выполнения оператора показан на рисунке 13. Из полученного результата видно, что процедура была создана и выполнена корректно.

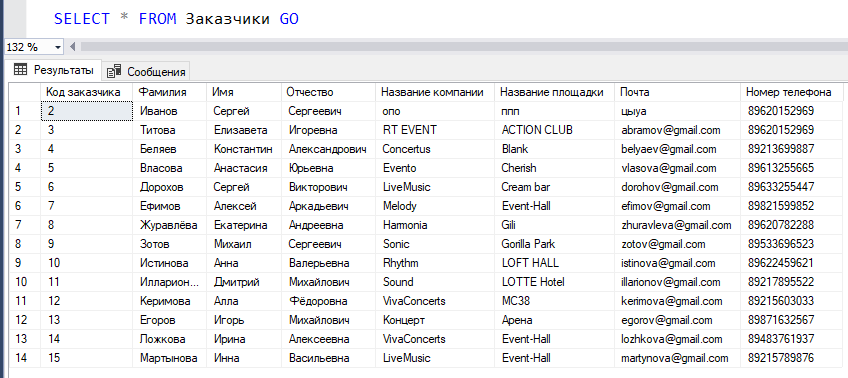


Рисунок 13 – Данные из таблицы "Заказчики"

Также были созданы процедуры на вставку новой записи в таблицу «Цена билетов», «Концерты», «Билеты» и «Покупатели».

1. Обновление данных в таблицах базы данных

Создание и выполнение процедуры на обновление данных в поле "Почта" по полю "Почта" в таблице "Заказчики" продемонстрировано на рисунке 14.

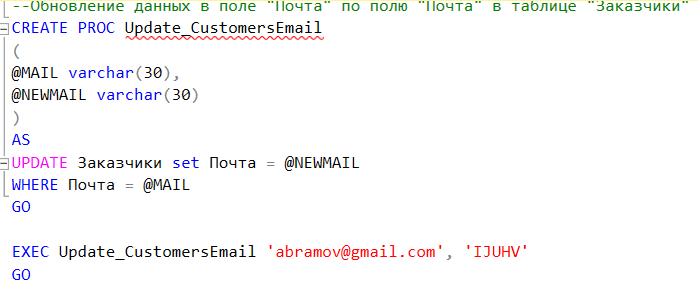


Рисунок 14 – Обновление почты покупателя

Результат выполнения оператора select показан на рисунке 15. Из полученного результата можно сделать вывод, что процедура была создана и выполнена корректно.

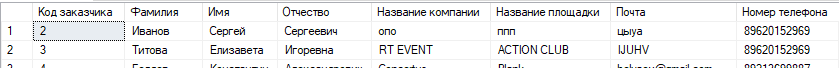


Рисунок 15 – Данные из таблицы "Заказчики"

Также были сделаны процедуры на обновление данных в поле: «Дата проведения» по полю «Код концерта» в таблице «Концерты»; "Цена супер-вип" по полю "Код концерта" в таблице "Цена билетов"; "Почта" по полю "Номер телефона" в таблице "Покупатели"; "Цена" по полю "Код покупателя" в таблице "Билеты".

1. Удаление данных в таблицах базы данных

Создание и выполнение процедуры на удаление записи из таблицы «Покупатели» по полю «Код покупателя» продемонстрировано на рисунке 16.

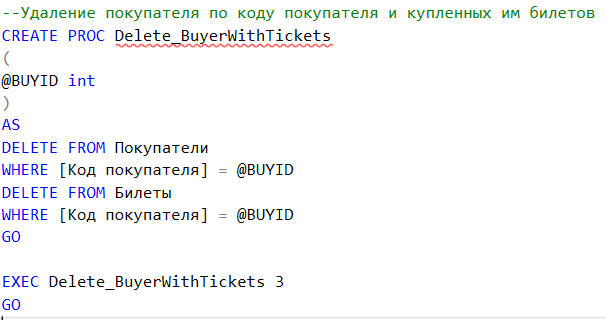


Рисунок 16 – Процедура удаления покупателя

Результат выполнения оператора select показан на рисунке 17. Из полученного результата можно сделать вывод, что процедура была создана и выполнена корректно.

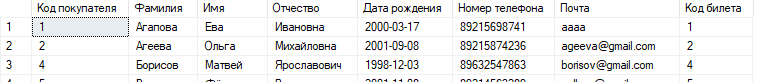


Рисунок 17 – Данные из таблицы "Покупатели"

Также были сделаны процедуры на удаление данных в таблице «Заказчики» по полю «Название компании» и в таблице «Концерты» по полю «Код концерта».

1. Поиск данных в таблицах базы данных

Создание и выполнение процедуры на поиск записи из таблицы «Цена билетов» по стоимости билетов продемонстрировано на рисунке 18.

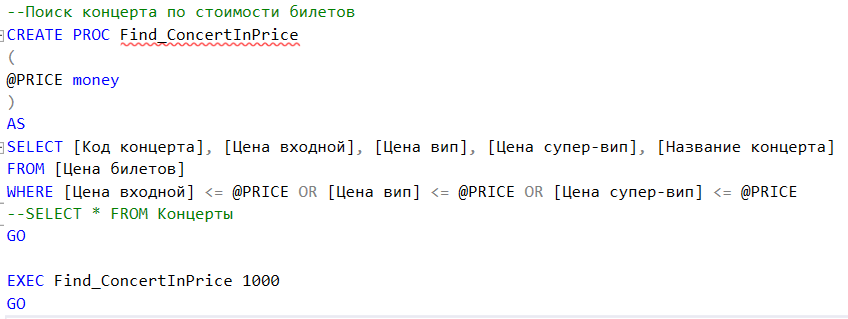


Рисунок 18 – Процедура поиска концерта по стоимости билетов

Для проверки работы данной процедуры, выполняется оператор select. Результат выполнения оператора select показан на рисунке 19. Из полученного результата можно сделать вывод, что процедура была создана и выполнена корректно.

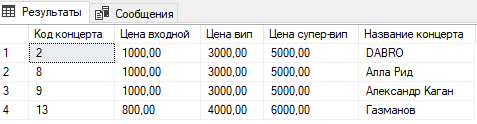


Рисунок 19 – Результат выполнение процедуры Find\_ConcertInPrice

Также были созданы процедуры на поиск поиск концерта по дате проведения, поиск цены билетов по коду концерта и на поиск ФИО покупателя по коду билета.

## **Создание триггеров**

Триггер SQL Server - это часть процедурного кода, как и хранимая процедура, но который выполняется только при наступлении определенного события. Имеются различные типы событий, которые могут вызвать срабатывание триггера. Например, вставка строки в таблицу, изменение структуры таблицы или авторизация пользователя в экземпляре SQL Server.

Для базы данных будут разработаны триггеры на:

1. Добавление
2. Обновление
3. Удаление
4. Триггер на добавление

Создание триггера, который после вставки нового покупателя печатает сообщение «Был добавлен новый покупатель». Это можно увидеть на рисунке 20.

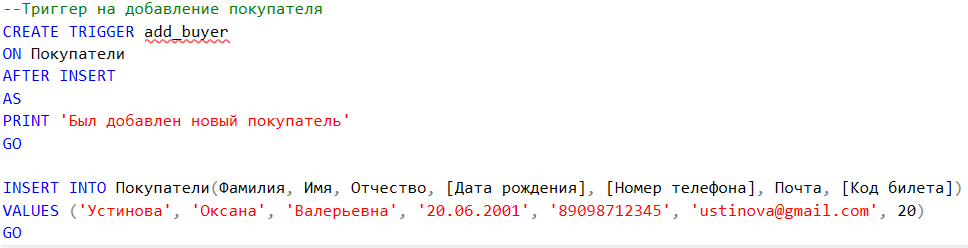


Рисунок 20 – Триггер на вставку нового покупателя

Проверка триггера путем вставки в таблицу новых значений продемонстрирована на рисунке 21.

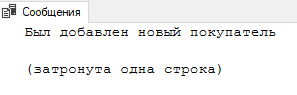


Рисунок 21 – Результат работы триггера на вставку нового покупателя

Также были созданы триггеры на добавление заказчика и концерта.

1. Триггер на обновление

Создание триггера, который после обновления в поле «Почта» таблицы "Заказчики" печатает сообщение «Была обновлена почта заказчика». Это можно увидеть на рисунке 22.

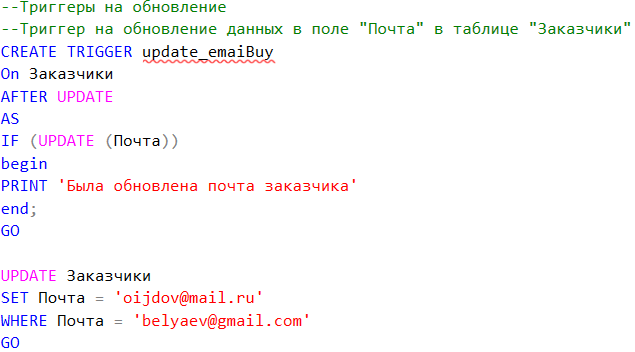


Рисунок 22 – Триггер на обновления почты заказчика

Результат проверки триггера на обновление почты заказчика показан на рисунке 23. Из полученного результата можно сделать вывод, что триггер была создан корректно.

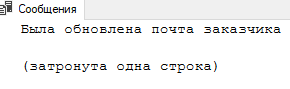


Рисунок 23 – Результат работы триггера на обновление почты заказчика

Также были созданы триггеры на обновление данных в поле «дата проведения» в таблице «Концерты» и на обновление данных в поле «Цена супер-вип» в таблице «Цена билетов».

1. Триггер на удаление

Создание триггера на удаление, который, если была попытка удалить заказчика с несуществующим названием компании, печатает сообщение «Неудалось удалить название компании», продемонстрировано на рисунке 24.

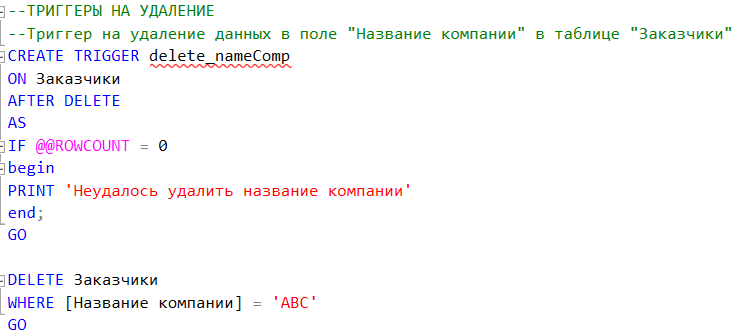


Рисунок 24 – Создание триггера на удаление заказчика

Проверка триггера путем удаление записи из таблицы «Заказчики» по полю «Название компании» продемонстрирована на рисунке 25.

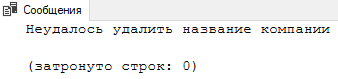


Рисунок 25 – Результат работы триггера на удаление заказчика

Также был создан триггер на удаление данных о билете по его коду.

## **Создание представлений**

Представление (англ. view) — виртуальная (логическая) таблица, представляющая собой поименованный запрос (синоним к запросу), который будет подставлен как подзапрос при использовании представления.

В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью набора данных, хранящегося в базе. Содержимое представления динамически вычисляется на основании данных, находящихся в реальных таблицах.

Изменение данных в реальной таблице базы данных немедленно отражается в содержимом всех представлений, построенных на основании этой таблицы.

Создание и вывод на экран представления рассчитывающего стоимость всех проданных билетов, максимальную, минимальную и среднюю стоимость, показано на рисунке 26.

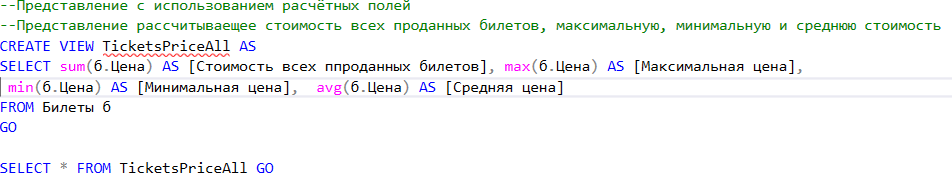


Рисунок 26 – Представление TicketsPriceAll

Полученные результаты показаны на рисунке 27.



Рисунок 27 – Стоимость всех проданных билетов, макс, минимальная и средняя стоимость

Создание и вывод на экран информации из представления, которое показывает покупателя, купившего максимальное количество билетов. Это можно увидеть на рисунке 28.

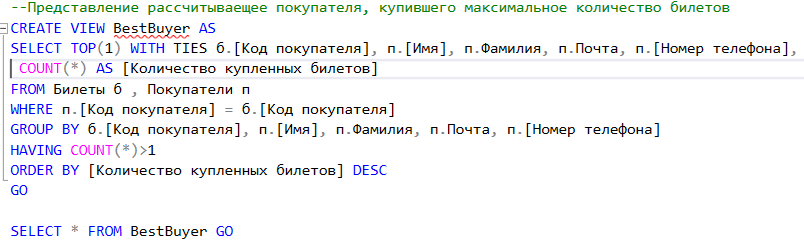


Рисунок 28 – Представление BestBuyer

Полученные результаты показаны на рисунке 29.



Рисунок 29 – Покупатель, купивший максимальное количество билетов

Создание и вывод на экран информации из представления, которое показывает количество концертов в каждом городе. Это можно увидеть на рисунке 30.

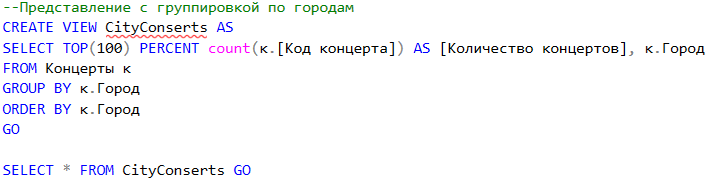


Рисунок 30 – Представление CityConserts

Полученные результаты показаны на рисунке 31.

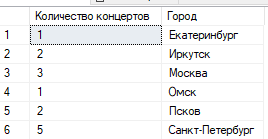


Рисунок 31 – Количество концертов в каждом городе

Также били созданы следующие представления:

* показывает иформацию объединяющую поля «Имя», «Отчество», «Номер телефона» из таблицы «Заказчики» и информацию о концерте таблицы «Концерты»;
* показывает иформацию объединяющую код покупателя, имя, номер телефона из таблицы «Покупатели» и информацию о купленных им билетах на концерт 7 из таблицы «Билеты»;
* показывает иформацию объединяющую код концерта, название, город из таблицы «Концерты» и информацию о цене входного билета >=2000 из таблицы «Цена билетов»;
* показывает иформацию объединяющую информацию о концерте из таблицы «Концерты» и купленные на него билеты в определённом городе из таблицы «Билеты»;
* показывает иформацию объединяющую информацию о концерте из таблицы «Концерты» и месте где он будет проходить из таблицы «Заказчики»;
* рассчитывает заказчика, заказавшего максимальное количество концертов;
* рассчитывает самую раннюю и самую позднюю дату концерта;
* группирует концерты по концертным площадкам;
* группирует по концертам и количеству купленных на них билетов.

## **Создание индексов**

Индекс представляет собой средство, помогающее ускорить поиск необходимых данных за счет физического или логического их упорядочивания. Индекс представляет собой набор ссылок, упорядоченных по определенному столбцу таблицы, который в данном случае будет называться индексированным столбцом. Индексы - это наборы уникальных значений для некоторой таблицы с соответствующими ссылками на данные.

Исходя из того, что для первичных ключей индексы создаются автоматически, и индексы должны присваиваться самым часто используемым в запросах полям, формируем индексы. Создание индексов для базы данных «Билетный оператор» продемонстрировано на рисунке 32.

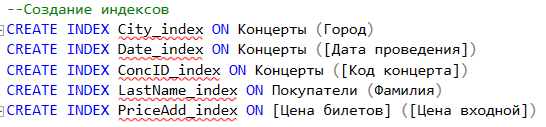


Рисунок 32 – Создание индексов для базы данных

Для того чтобы получить информацию по индексам каждой таблицы, используем процедуру sp\_helpindex, как на рисунке 33.

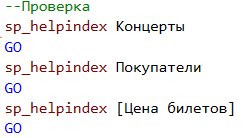


Рисунок 33 – Процедура sp\_helpindex

В результате выводятся все индексы таблиц, включая те, которые были созданы автоматически. Это видно на рисунке 34.

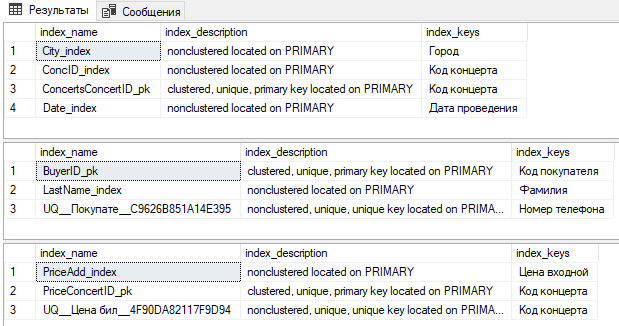


Рисунок 34 – Индексы всех таблиц

Все индексы для базы данных определены и созданы.

## **Создание пользователей и ролей**

Определим пользователей, которые будут иметь доступ к созданной базе данных.

Для начала создаём учётные записи. Учётная запись пользователя (login) позволяет ему подключиться к самому серверу, но не даёт автоматического доступа к базам данных. Создание учетных записей продемонстрировано на рисунке 35.

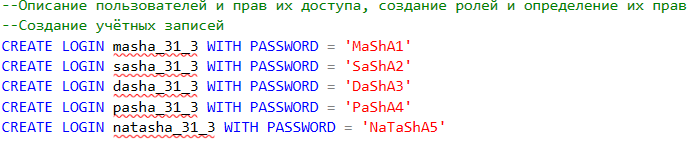


Рисунок 35 – Создание учетных записей

Далее создаём пользователей. Запись пользователя создаётся для каждой базы данных SQL-сервера на основании учётной записи. На основе прав, выданных пользователю как пользователю базы данных, его регистрационное имя (login) получает доступ к соответствующей базе данных. Создание пользователей для ранее созданных учетных записей показано на рисунке 36.

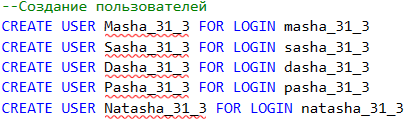


Рисунок 36 – Создание пользователей

Теперь определяем роли. Роли — это средство объединения учетных записей в группы с целью упрощения администрирования. Включив учетную запись в ту или иную роль сервера, можно предоставить ей определенный набор прав по администрированию сервера и доступа к базе данных. Создание ролей базы данных и включение в них пользователей отображено на рисунке 37.

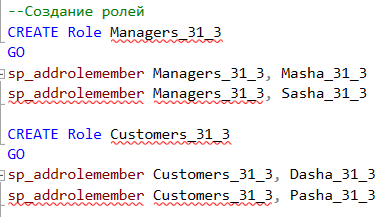


Рисунок 37 – Создание ролей

Далее можем выдавать права доступа для ролей и пользователей.

Пользователю Natasha\_31\_3 предоставим все права. Это продемонстрировано на рисунке 38.

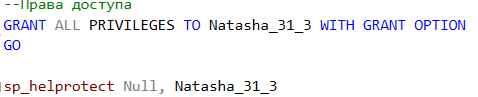


Рисунок 38 – Права доступа для пользователя Natasha\_31\_3

Роле Managers\_31\_3 предоставим права необходимые для работы менеджеров. Это продемонстрировано на рисунке 39.

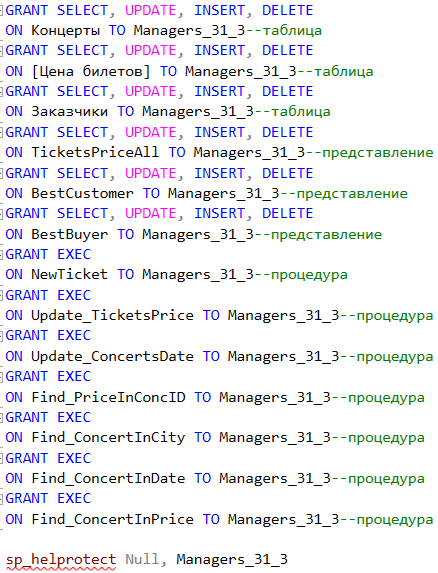


Рисунок 39 – Права доступа для роли Managers\_31\_3

Роле Customers\_31\_3 предоставим права необходимые для работы заказчиков. Это продемонстрировано на рисунке 40.

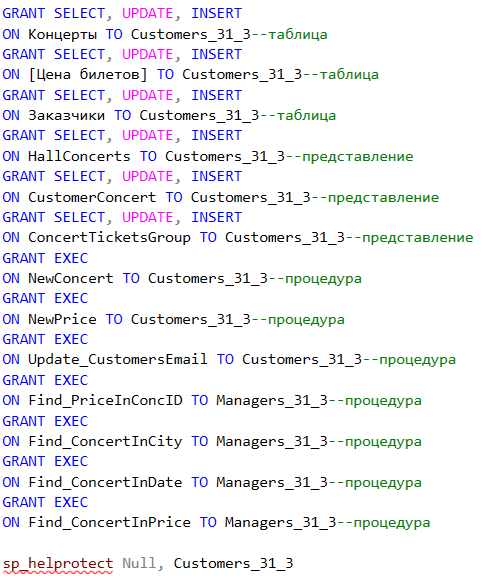


Рисунок 40 – Права доступа для роли Customers\_31\_3

Пользователям Masha\_31\_3 и Dasha\_31\_3 предоставим права продемонстрированные на рисунке 41.

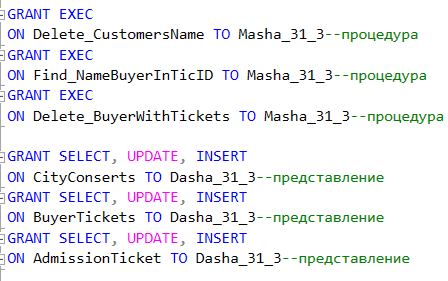


Рисунок 41 – Права доступа для пользователей Masha\_31\_3 и Dasha\_31\_3

Необходимое разграничение прав доступа к базе данных создано.

## **Стратегия резервного копирования базы данных**

База данных у меня будет сохраняться на уже существующие устройства.

Полная копия будет создаваться первого числа каждого месяца в 1:00, начиная с 21.06.2024. Как создать полную копию базы данных показано на рисунке 42.



Рисунок 42 – Создание полной резервной копии

Копия журнала транзакций будет создаваться каждый понедельник в 3:00, начиная с 24.06.2024. Как создать копию журнала транзакций базы данных показано на рисунке 43.



Рисунок 43 –Создание копии журнала транзакций

Дифференциальная копия будет создаваться каждый день в 2:00, начиная с 21.06.2024. Как создать дифференциальную копию базы данных показано на рисунке 44.



Рисунок 44 –Создание дифференциальной копии

Стратегия резервного копирования базы данных разработана.

## **Проверка на целостность БД**

Подключаемся к ядру СУБД под пользователем Sasha\_31\_3. Это показано на рисунке 45.

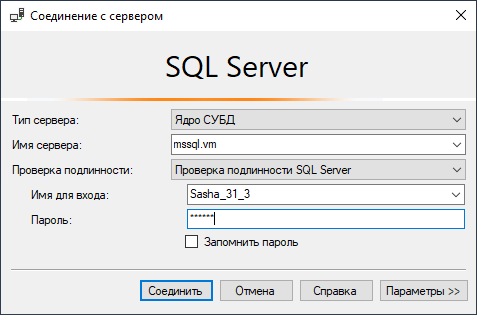


Рисунок 45 – Подключаемся к ядру СУБД под пользователем Sasha\_31\_3

Запустим процедуру, на которую у пользователя Sasha\_31\_3 есть права. Это продемонстрировано на рисунке 46.

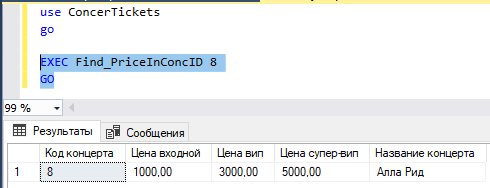


Рисунок 46 – Процедура, на которую у пользователя Sasha\_31\_3 есть права

Теперь запустим процедуру, на которую у пользователя Sasha\_31\_3 нет права. Это продемонстрировано на рисунке 47.

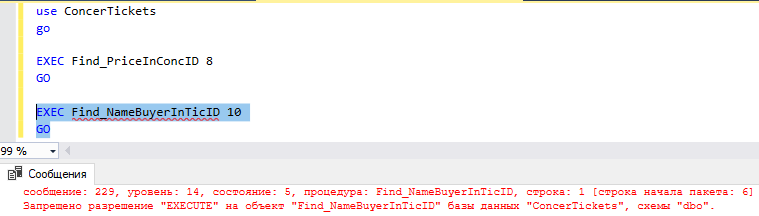


Рисунок 47 – Процедура, на которую у пользователя Sasha\_31\_3 нет права

# **ЗАКЛЮЧЕНИЕ**

При работе над курсовой работой были закреплены практические навыки и теоретические знания по дисциплине «Основы проектирования баз данных».

В процессе проектирования базы данных для билетного оператора «concerTickets», были реализованы все поставленные задачи. А именно:

1. Проведен анализ предметной области;
2. Разработана концептуально-логическая модель, описывающая объекты и связи предметной области, составлен список сущностей и атрибутов, которые их описывают;
3. Разработана физическая модель;
4. Реализована база данных, отражено ее практическое применение в виде построения таблиц, запросов, форм, отчетов, макросов и создания главной кнопочной формы.

Таблицы в разработанной базе данных позволяют хранить и вводить в них информацию о заказчиках, концертах, покупателях, билетах и о стоимости билетов. Данные могут вводиться прямо в таблицу или через экранную форму. Запросы в разработанной базе данных предназначены для быстрого поиска необходимой информации в виде выборки, подсчета, изменения существующей в таблицах информации. Формы в разработанной базе данных значительно упрощают ввод, изменение и удаление информации, переключение между объектами базы данных. Отчеты в разработанной таблице помогают очень быстро сформировать документ, который может потребоваться для использования. Макросы автоматизируют процесс управления всеми разработанными объектами.

Разработанная информационная система имеет перспективу дальнейшего совершенствования и развития, например, создание дополнительных таблиц с информацией о площадках, новых заказчиках и покупателях, купленных билетах.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. ГОСТ Р 7.0.97-2016. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов (утв. Приказом Росстандарта от 08.12.2016 N 2004-ст) (ред. от 14.05.2018)
2. Агальцов В.П. Базы данных. – М.: Издательство «ФОРУМ», 2021 г. – 352 с.
3. [Новиков Б.](https://www.combook.ru/authors/%D0%9D%D0%BE%D0%B2%D0%B8%D0%BA%D0%BE%D0%B2%20%D0%91./)А. Основы технологий баз данных/ [Горшкова Е.](https://www.combook.ru/authors/%D0%93%D0%BE%D1%80%D1%88%D0%BA%D0%BE%D0%B2%D0%B0%20%D0%95./)А., Графеева Н.Г. – М.: Издательство «ДМК Пресс», 2020 г.– 582 с.
4. Шустова Л.И. Базы данных/ Тараканов О.В. – М.: Издательство «НИЦ ИНФРА-М», 2021 г. – 304 с.
5. <https://intuit.ru> - Национальный Открытый Университет "ИНТУИТ"
6. Сайт: docs.microsoft.com – Статья: Представления. URL: https://docs.microsoft.com/ru-ru/sql/relational-databases/views/views?view=sql-server-ver16