МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

Институт Автоматизированных Систем и Технологий

Кафедра информационных систем и технологий

Информационные системы и технологии

Направление подготовки

09.03.02

Разработка программного обеспечения информационных систем

Дисциплина: Базы данных

**КУРСОВАЯ РАБОТА**

Проектирование и реализация реляционной базы данных для системы ресторана

Выполнил студент:

Мусатов Эдуард Артурович

Группа: 21-ИСбо-2

Проверил: кандидат наук, доцент

Исаева Мария Владимировна

Оценка

Подпись преподавателя

Кострома

2023

Оглавление

[1. Введение 2](#_Toc154632634)

[2. Описание (сценарий) предметной области 2](#_Toc154632635)

[3. Анализ предметной области 6](#_Toc154632636)

[3.1 Цели создания БД и информационной системы в целом в данной предметной области 6](#_Toc154632637)

[3.2 Категории данных предметной области 6](#_Toc154632638)

[3.3 Пользователи системы и их задачи 10](#_Toc154632639)

[3.4 Ограничения и бизнес-правила с классификацией на декларативные (по видам) и процедурные ограничения целостности 11](#_Toc154632640)

[4. Построение ERD 18](#_Toc154632641)

[5. Построение реляционной схемы 19](#_Toc154632642)

[6. Реализация БД на сервере 21](#_Toc154632643)

[7. Реализация объектов БД 33](#_Toc154632644)

[7.1 Хранимые процедуры 34](#_Toc154632645)

[7.2 Триггеры 40](#_Toc154632646)

[7.3 Индексы 44](#_Toc154632647)

[8. Реализация системы безопасности 45](#_Toc154632648)

[9. Реализация приложения для работы с БД 47](#_Toc154632649)

[10. Заключение 49](#_Toc154632650)

[11. Список используемой литературы 50](#_Toc154632651)

# 1. Введение

Цель курсовой работы - спроектировать и реализовать реляционную

базу данных для системы ресторана с использованием СУБД

Microsoft SQL Server и с применением знаний и умений, полученных в

рамках изучения дисциплины “Базы данных”.

В соответствии с целью были поставлены следующие задачи:

1) составить описание предметной области;

2) выполнить анализ предметной области;

3) построить ERD;

4) построить реляционную схему БД;

5) реализовать БД на сервере;

6) реализовать объекты БД;

7) реализовать систему безопасности;

8) реализовать приложение для работы с БД.

# 2. Описание (сценарий) предметной области

Ресторан представляет собой заведение, в котором можно предварительно забронировать столик, а также оформлять заказы в самом ресторане и на вынос.

У ресторана есть меню, состоящее из разделов, каждое со своим названием и описанием. В разделы меню можно условно вставлять ассоциирующиеся с названием позиции меню. Раздел включает в себя множество позиций меню, у которых указывается название, описание, цена, его порция, что представляет собой определенное количество единиц измерения (например, 200 грамм). Неплохо было бы хранить дату ввода этой позиции в меню для дальнейшей аналитики (например, для оценки эффективности обновлений меню, если новые позиции меню быстро становятся популярными, это может быть индикатором успешных изменений в меню). Позиция меню может быть недоступна в данный момент времени по различным причинам (например, если это сезонное блюдо или по отсутствию конкретных ингредиентов), то есть такую позицию нельзя будет заказать, а как-то отслеживать доступность необходимо.

При оформлении заказа у клиента есть возможность изменить количество выбранных позиций меню. Позицией меню может быть как напиток, блюдо или что-то еще.

В зале ресторана находится множество столиков, каждый из которых рассчитан на определённое количество гостей. Каждый столик имеет свой номер и своё описание, в котором можно написать его месторасположение и другие данные. Также столик может быть недоступен в данный момент времени. Предполагается, что в ресторане есть столики максимум на 12 мест. Также предполагается, что к конкретному столику привязываются конкретные официанты, которые будут обрабатывать заказы с этих столиков. Сам официант может быть привязан к нескольким столикам.

Ресторан имеет свой сайт, на котором клиент может самостоятельно забронировать столик и оформить заказ на вынос.

В разделе бронирования клиент может предварительно забронировать столик, указав количество гостей и дату с временем, в которые будет начата бронь. Забронировать за раз можно только один столик. Затем бронь будет обрабатываться определенным оператором бронирования, который будет пользователем системы. Оператор бронирования сможет определить точное время начала бронирования без возникновения конфликтов, а также выберет столик с подходящим количеством гостей. Клиент также может указать особые требования (например, пожелания по месторасположению столика). Также чтобы оператору сообщить уточненные дату начала брони и столик, клиент должен указать свое имя и телефон для дальнейшей связи. Забронировать столик также может произойти, если позвонить по определенному номеру, где будет отвечать оператор бронирования. Также сеанс бронирования может быть отменен.

После предварительного бронирования столика клиент может прийти в ресторан и там оформить заказ. Это будет заказ в самом ресторане. К нему привязывается определенный столик, потому что можно не забронировав столик, сесть за свободный столик и оформить заказ. Заказ будет принимать один официант, и эту дату и время приема заказа следует хранить, также неплохо было бы отдельно хранить дату и время подачи блюда. Эту информацию может где-то фиксировать официант, чтобы в дальнейшем всю эту информацию добавить в базу данных. Заказ может состоять из нескольких позиций меню, каждая из которых может быть в определенном количестве. Общая стоимость заказа вычисляется как сумма стоимостей каждой позиции меню в своем количестве. В самом заказе можно хранить способ оплаты этого заказа: наличным или безналичным расчетом.

На самом сайте клиент может оформить заказ на вынос. И тут у клиента существует аккаунт, чтобы под этим аккаунтом оформлять заказы. В самом аккаунте можно отслеживать количество оформленных заказов, чтобы на последующие заказы предоставлять скидку на весь заказ. За каждое необходимое количество заказов предоставляется определенный процент скидки от всего заказа. В самом аккаунте будет храниться информация о ФИО клиента, его логине и пароле, телефон, почта, его пол, а также количество оформленных заказов и исходя из этого количества определенная скидка.

На сайте расположено само меню с доступными для заказа блюдами и напитками, которые можно выбрать в определенном количестве и оформить заказ на вынос. Сам заказ на вынос может иметь два варианта получения: доставка и самовывоз. В заказе на вынос можно хранить имя и телефон клиента, ведь оформить заказ на вынос можно не только через аккаунт, но и без него на самом сайте или позвонив оператору обработки заказов: информация об имени и телефоне в таком случае пригодится для дальнейшего уточнения деталей заказа. В случае с аккаунтом, информация об имени и телефоне будет храниться уже там. В заказе на вынос также следует хранить особые требования клиента и дату получения заказа. Дата оформления заказа также нужна для будущей аналитики. Если заказ будет доставляться, то следует хранить курьера, предварительный адрес доставки и затем определяемую стоимость доставки. Стоимость доставки не вычисляется автоматически, а выбирается оператором при обработке заказа. Предварительно уже есть оценка соответствия расстояния определенной стоимости, это можно для удобства также хранить в виде справочника в базе данных. Оператор смотрит адрес доставки, определяет расстояние и исходя из этого расстояния выбирает определенную стоимость доставки. Затем стоимость доставки добавляется в общую стоимость заказа. В случае с самовывозом эту информацию уже не требуется хранить. Заказ может быть отменен, обрабатываться и быть принят. Общая стоимость заказа вычисляется как стоимость всех позиций + стоимость доставки - скидка, если она присутствует. Способ оплаты: наличный и безналичный расчет, также следует хранить.

Каждая позиция меню состоит из ингредиентов, и состав каждого блюда/напитка следует хранить. У самого ингредиента можно хранить название, описание, и общее количество в определенной единице измерения. В самой позиции ингредиент может быть в определенном количестве.

Можно также хранить информацию о поставках и поставщиках. Поставщик в себя включает организацию, контактное лицо, адрес, телефон и почту. Сама поставка имеет свою общую стоимость и дату. В одной поставке могут присутствовать множество ингредиентов в определенном количестве, и это количество также будет иметь определенную стоимость. И общая стоимость поставки вычисляется вычисляется как сумма этих стоимостей. Естественно, поставка включает в себя одного поставщика.

Также необходимо хранить информацию о сотрудниках и их должностях: ФИО сотрудника, его пол, телефон и почта, дата найма и дата увольнения и его должность. В должности храним информацию о названии, описании и предположительном окладе должности. Некоторые сотрудники являются пользователями БД: например, операторы бронирования и операторы обработки заказов.

# 3. Анализ предметной области

# 3.1 Цели создания БД и информационной системы в целом в данной предметной области

1) Улучшение обслуживания клиентов

2) Сбор и анализ данных: БД позволит собрать информацию о заказах, бронированиях, блюдах, ингредиентах, клиентах, поставках и т.д. Анализ этих данных позволит принимать более обоснованные решения и улучшать бизнес-стратегию.

3) С помощью БД можно получить информацию о производительности работы персонала (например, количество обслуженных заказов у каждого официанта)

4) Анализ финансов и прибыли

5) Оптимизация процессов: автоматизация бронирования и заказов снизит нагрузку на персонал, что повысит эффективность ресторана

6) Расширение бизнеса

# 3.2 Категории данных предметной области

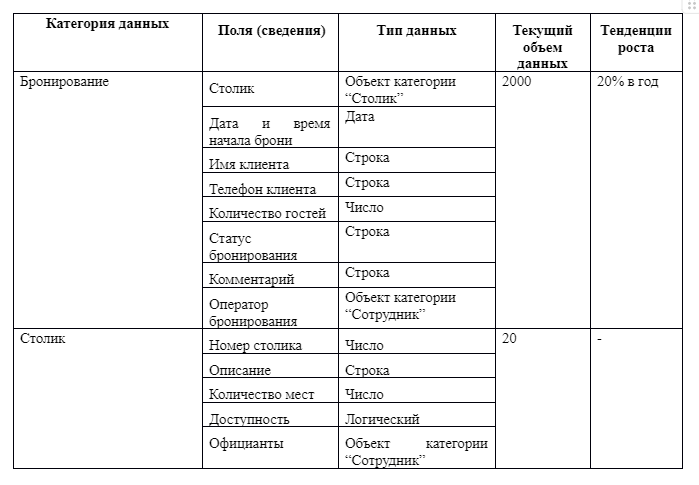


Рис.1. Категории «Бронирование» и «Столик»

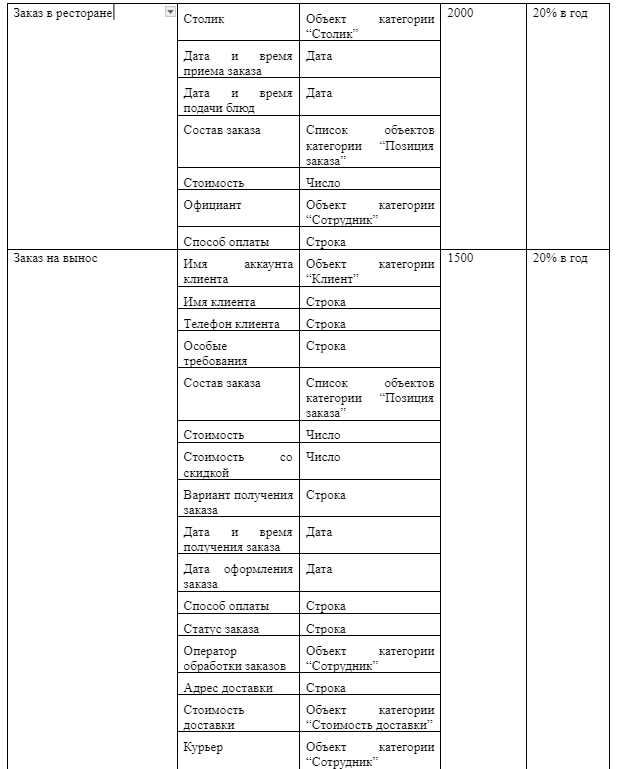


Рис. 2. Категории «Заказ в ресторане» и «Заказ на вынос»

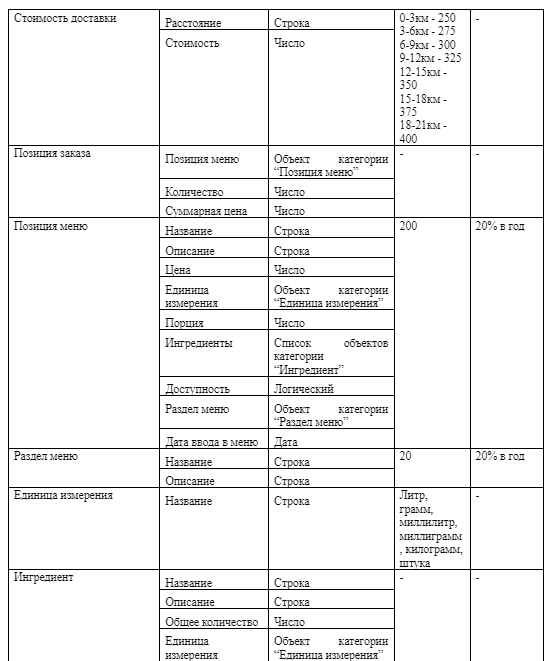


Рис. 3. Категории «Стоимость доставки», «Позиция заказа», «Позиция меню», «Раздел меню», «Единица измерения», «Ингредиент»



Рис. 4. Категории «Поставка», «Поставщик», «Сотрудник»

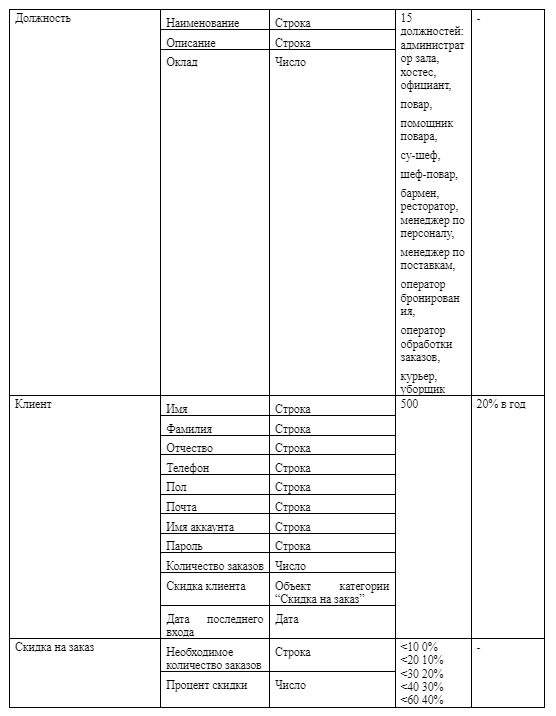


Рис. 5. Категории «Должность», «Клиент», «Скидка на заказ»

# 3.3 Пользователи системы и их задачи

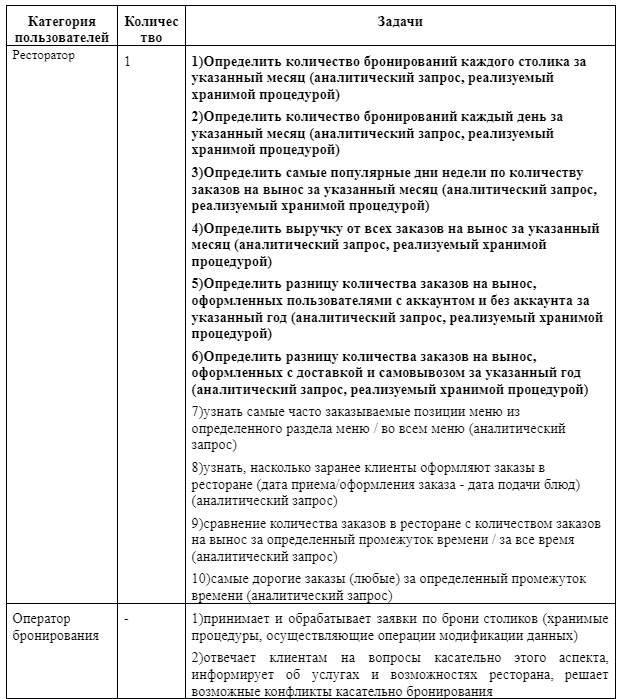


Рис. 6. Пользователи «Ресторатор» и «Оператор бронирования»

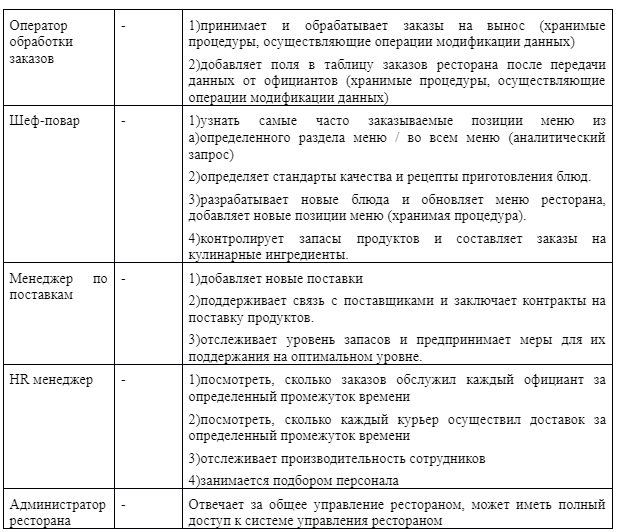


Рис. 7. Пользователи «Оператор обработки заказов», «Шеф-повар», «Менеджер по поставкам», «HR менеджер», «Администратор ресторана»

# 3.4 Ограничения и бизнес-правила с классификацией на декларативные (по видам) и процедурные ограничения целостности

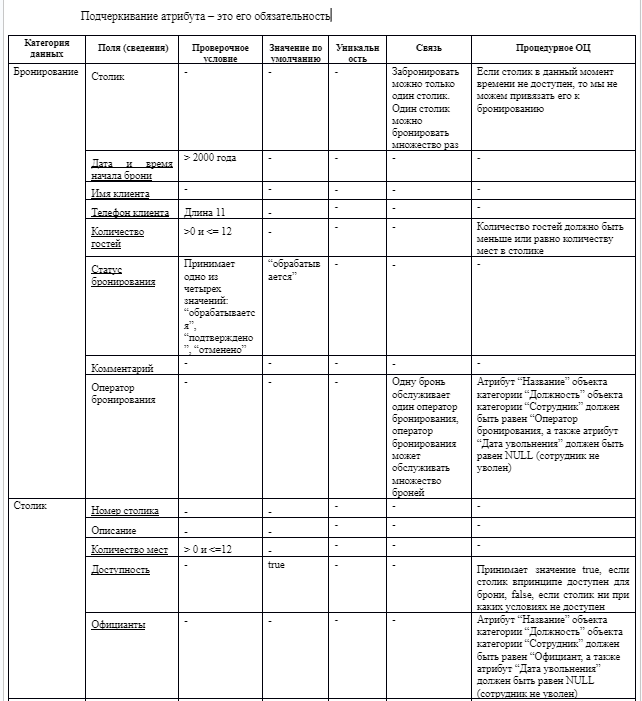


Рис. 8. ОЦ для категорий «Бронирование» и «Столик»

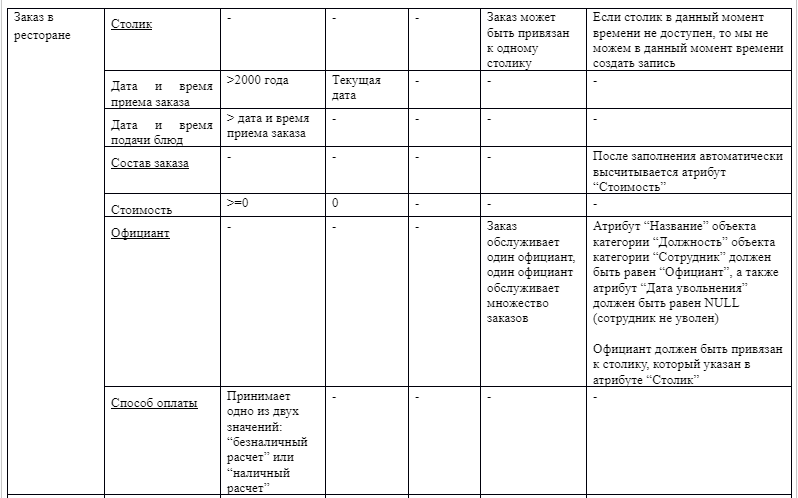
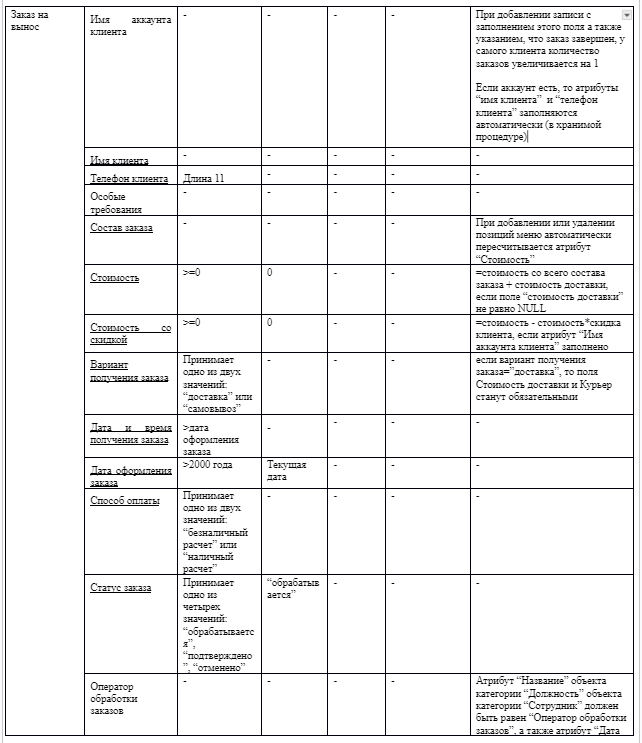


Рис. 9. ОЦ для категории «Заказ в ресторане»



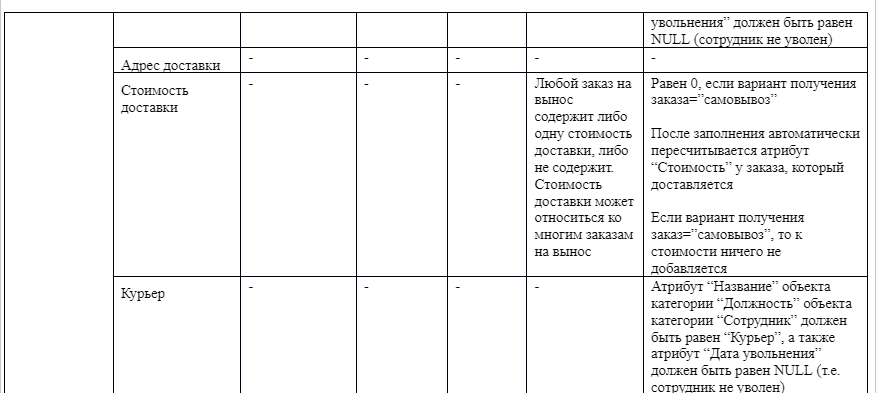


Рис. 10-11. ОЦ для категории «Заказ на вынос»

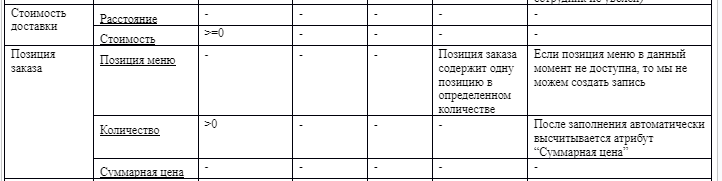


Рис. 12. ОЦ для категории «Стоимость доставки» и «Позиция заказа»

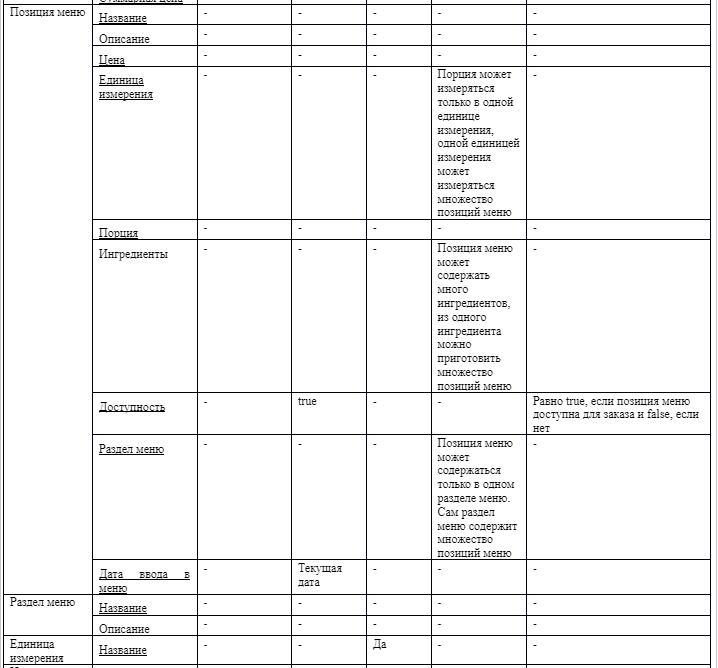


Рис. 13. ОЦ для категорий «Позиция меню», «Раздел меню» и «Единица измерения»

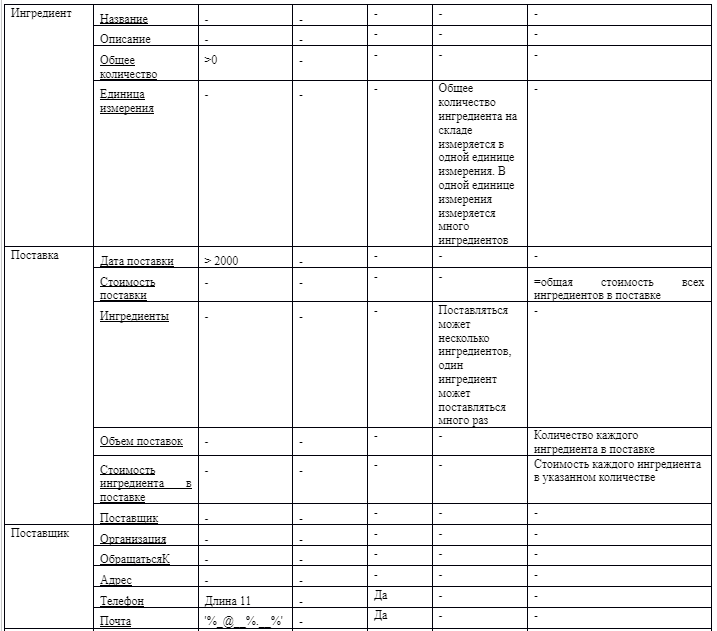


Рис. 14. ОЦ для категорий «Ингредиент», «Поставка» и «Поставщик»

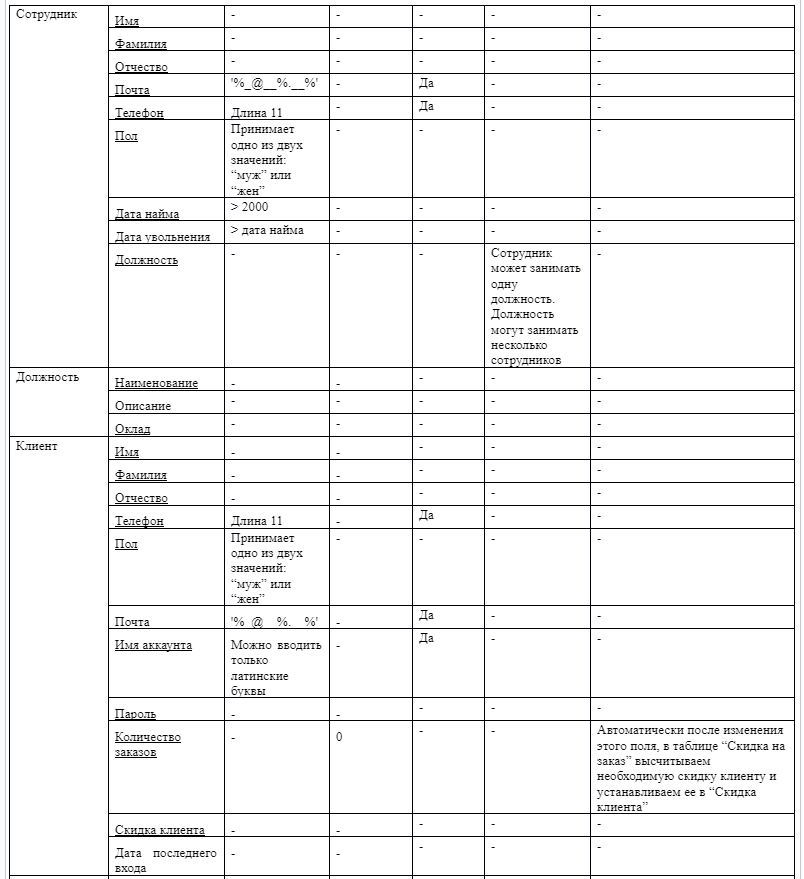


Рис. 15. ОЦ для категорий «Сотрудник», «Должность» и «Клиент»

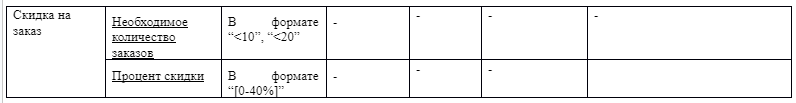
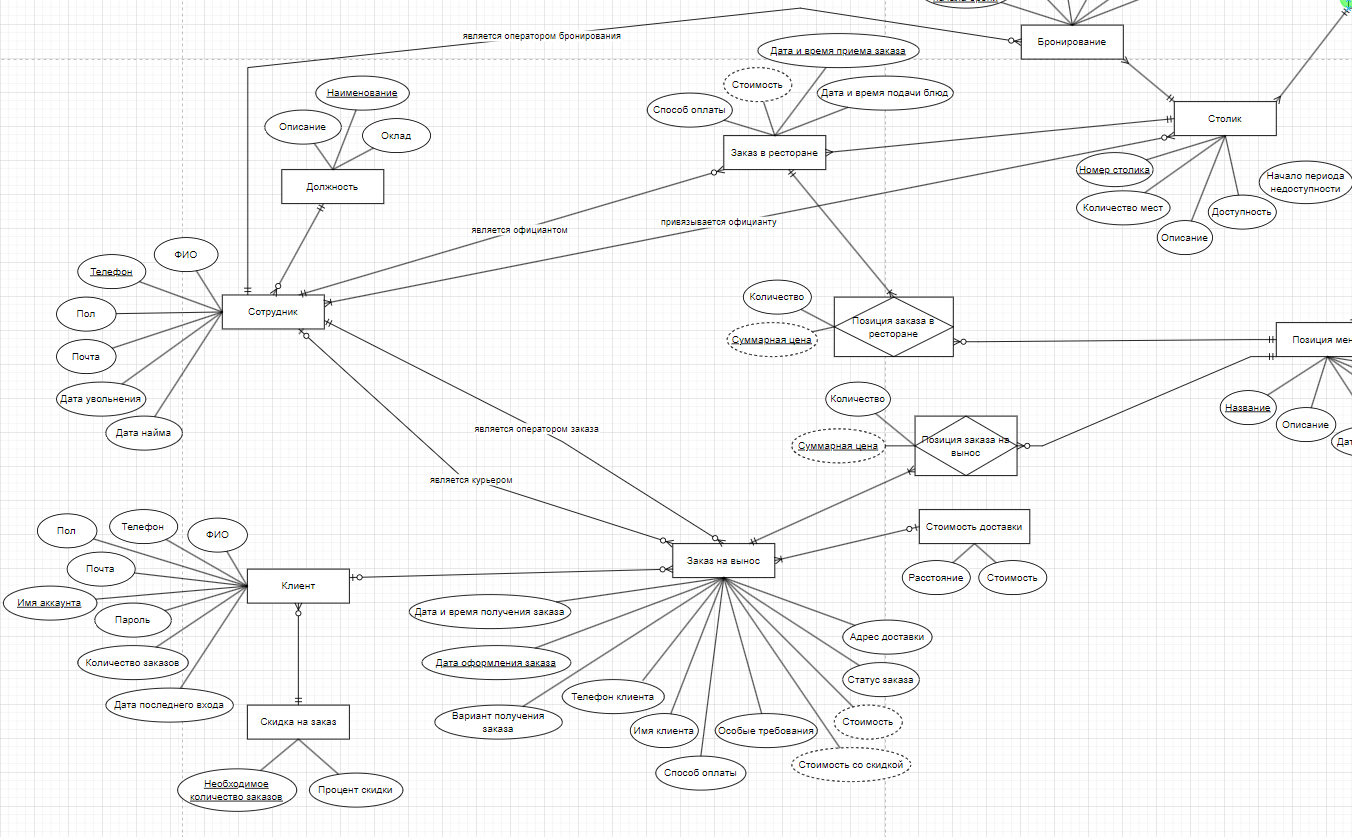


Рис. 16. ОЦ для категории «Скидка на заказ»

# 4. Построение ERD



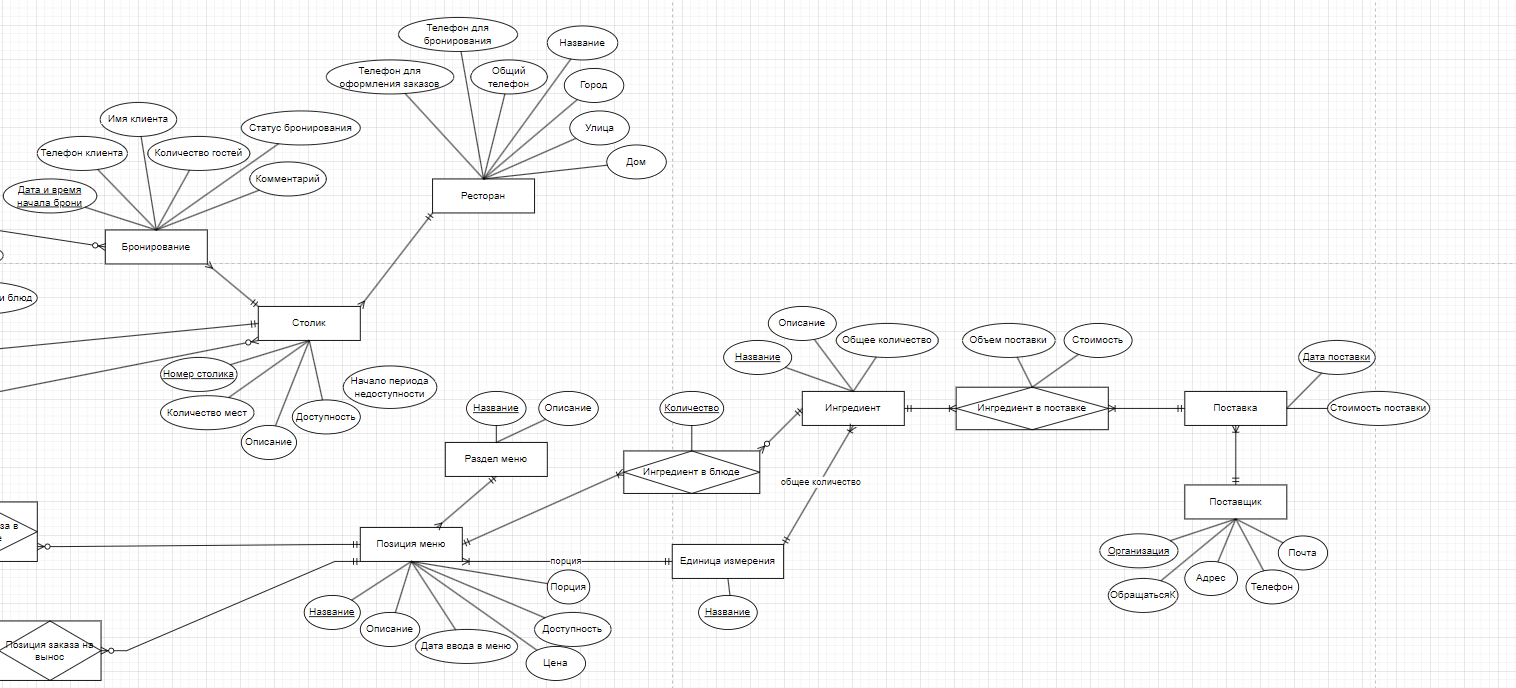


Рис. 17-18. ER-диаграмма, модель «сущность-связь»

# 5. Построение реляционной схемы

На данном этапе на основе ER-диаграммы была построена реляционная схема. Так как нам необходима реляционная модель, то и строить необходимо реляционную схему.

1) СОТРУДНИК (код сотрудника, имя, фамилия, отчество, телефон, пол, почта, дата найма, дата увольнения, код должности)

2) СОТРУДНИК-ПОЛЬЗОВАТЕЛЬ (код сотрудника, код пользователя БД)

3) ДОЛЖНОСТЬ (код должности, наименование, описание, оклад)

4) КЛИЕНТ (имя аккаунта клиента, имя, фамилия, отчество, телефон, пол, почта, пароль, дата последнего входа, количество заказов, код скидки)

5) СКИДКА НА ЗАКАЗ (код скидки, необходимое количество заказов, процент скидки)

6) СТОЛИК (номер столика, описание, количество мест, доступность, код ресторана)

7) ОФИЦИАНТЫ-СТОЛИКИ (код сотрудника, номер столика)

8) БРОНИРОВАНИЕ (код бронирования, дата и время начала брони, имя клиента, телефон клиента, комментарий, количество гостей, статус бронирования, код столика, код сотрудника (ОПЕРАТОР БРОНИРОВАНИЯ))

9) РАЗДЕЛ МЕНЮ (код раздела меню, название, описание)

10) ПОЗИЦИЯ МЕНЮ (код позиции меню, название, описание, доступность, дата ввода в меню, цена, порция, название единицы измерения, код раздела меню)

11) СОСТАВ ЗАКАЗА В РЕСТОРАНЕ (количество, суммарная цена, код позиции меню, код заказа в ресторане)

12) СОСТАВ ЗАКАЗА НА ВЫНОС (количество, суммарная цена, код позиции меню, код заказа на вынос)

13) ЗАКАЗ В РЕСТОРАНЕ (код заказа в ресторане, дата и время приема заказа, дата и время подачи блюд, стоимость, способ оплаты, код столика, код сотрудника (ОФИЦИАНТ))

14) ЗАКАЗ НА ВЫНОС (код заказа на вынос, особые требования, имя клиента, телефон клиента, стоимость, стоимость со скидкой, способ оплаты, дата оформления заказа, дата и время получения заказа, вариант получения заказа, адрес доставки, статус заказа, код стоимости доставки, код сотрудника (КУРЬЕР), имя аккаунта клиента, код сотрудника (ОПЕРАТОР ОБРАБОТКИ ЗАКАЗОВ))

15) СТОИМОСТЬ ДОСТАВКИ (код стоимости доставки, расстояние, стоимость)

16) ЕДИНИЦА ИЗМЕРЕНИЯ (название единицы измерения)

17) ИНГРЕДИЕНТ (код ингредиента, название, описание, общее количество, название единицы измерения)

18) СОСТАВ ПОЗИЦИЙ МЕНЮ (код ингредиента, код позиции меню, количество)

19) ПОСТАВКА (код поставки, дата поставки, стоимость поставки, код поставщика)

20) ПОСТАВОЧНЫЙ АССОРТИМЕНТ (код поставки, код ингредиента, объем поставки, стоимость)

21) ПОСТАВЩИК (код поставщика, организация, обращатьсяК, адрес, телефон, почта)

22) РЕСТОРАН (код ресторана, название ресторана, город, улица, дом, общий телефон, телефон для бронирования, телефон для оформления заказов)

Первичные ключи – **подчеркнуты**

Внешние ключи – **выделены красным цветом**

Приведение к 1НФ.

Эти отношения в 1НФ, так как все атрибуты атомарны.

Приведение к 2НФ.

Требование 2НФ: каждый не первичный атрибут функционально полно зависит от ключа.

Очевидно, что эти отношения находятся во 2НФ, так как в них ключ состоит

из одного атрибута.

Приведение к 3НФ.

Построим ДФЗ для проверки соответствия требованию 3НФ:

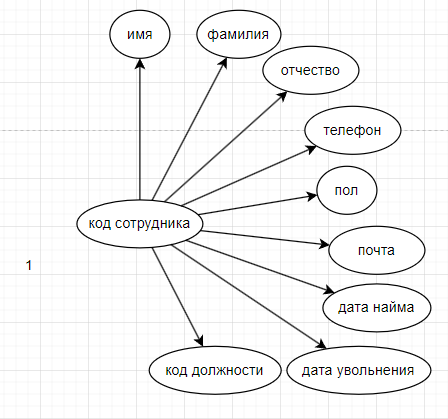


Рис. 19. ДФЗ с атрибутами отношения СОТРУДНИК

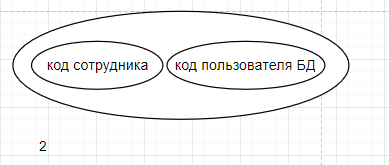


Рис. 20. ДФЗ с атрибутами отношения СОТРУДНИК-ПОЛЬЗОВАТЕЛЬ

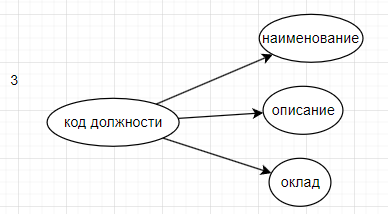


Рис. 21. ДФЗ с атрибутами отношения ДОЛЖНОСТЬ

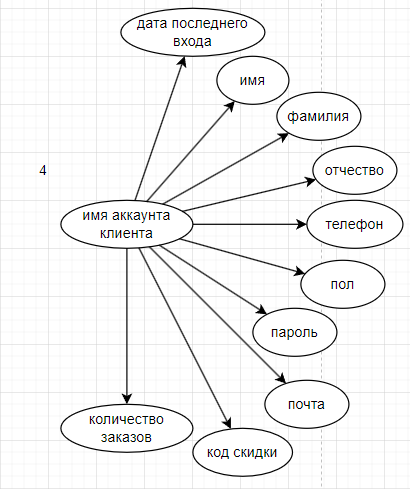


Рис. 22. ДФЗ с атрибутами отношения КЛИЕНТ

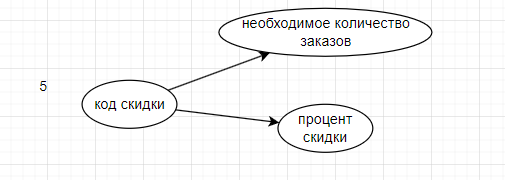


Рис. 23. ДФЗ с атрибутами отношения СКИДКА НА ЗАКАЗ

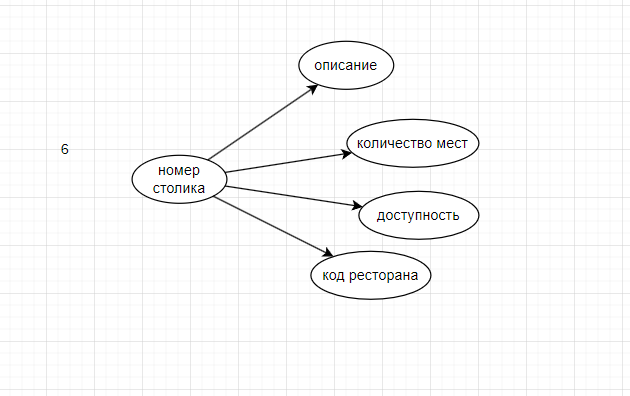


Рис. 24. ДФЗ с атрибутами отношения СТОЛИК

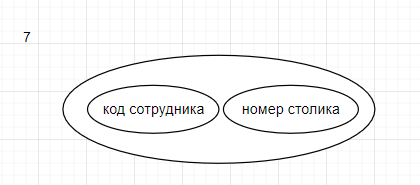


Рис. 25. ДФЗ с атрибутами отношения ОФИЦИАНТЫ-СТОЛИК

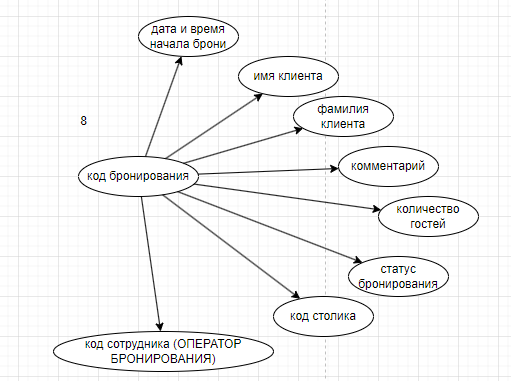


Рис. 26. ДФЗ с атрибутами отношения БРОНИРОВАНИЕ

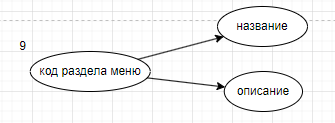


Рис. 27. ДФЗ с атрибутами отношения РАЗДЕЛ МЕНЮ

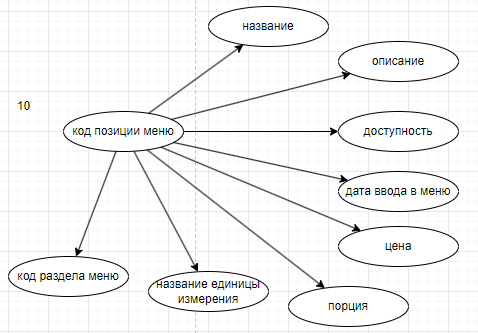


Рис. 28. ДФЗ с атрибутами отношения ПОЗИЦИЯ МЕНЮ

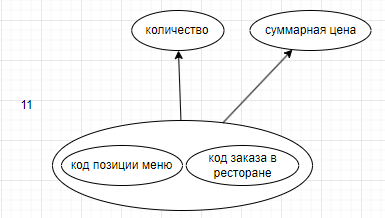


Рис. 29. ДФЗ с атрибутами отношения СОСТАВ ЗАКАЗА В РЕСТОРАНЕ

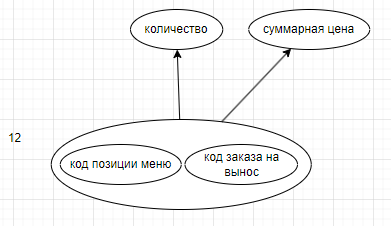


Рис. 30. ДФЗ с атрибутами отношения СОСТАВ ЗАКАЗА НА ВЫНОС

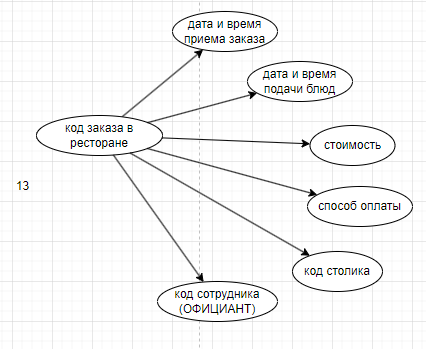


Рис. 31. ДФЗ с атрибутами отношения ЗАКАЗ В РЕСТОРАНЕ



Рис. 32. ДФЗ с атрибутами отношения ЗАКАЗ НА ВЫНОС

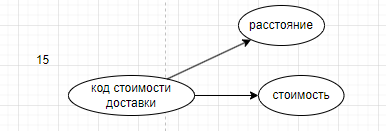


Рис. 33. ДФЗ с атрибутами отношения СТОИМОСТЬ ДОСТАВКИ

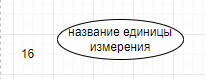


Рис. 34. ДФЗ с атрибутами отношения ЕДИНИЦА ИЗМЕРЕНИЯ

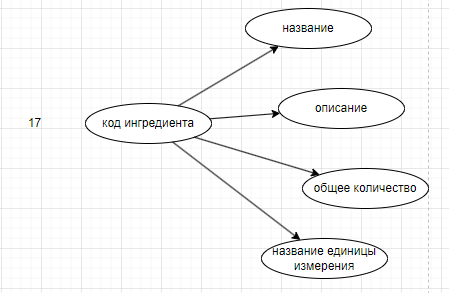


Рис. 35. ДФЗ с атрибутами отношения ИНГРЕДИЕНТ



Рис. 36. ДФЗ с атрибутами отношения СОСТАВ ПОЗИЦИЙ МЕНЮ

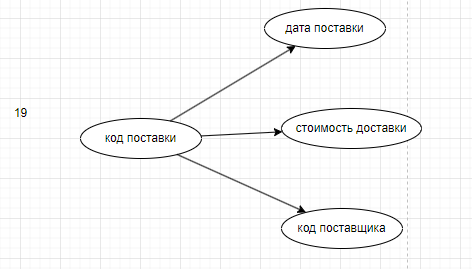


Рис. 37. ДФЗ с атрибутами отношения ПОСТАВКА



Рис. 38. ДФЗ с атрибутами отношения ПОСТАВОЧНЫЙ АССОРТИМЕНТ

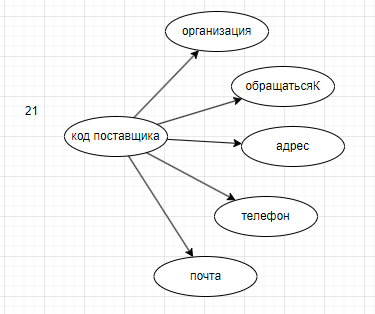


Рис. 39. ДФЗ с атрибутами отношения ПОСТАВЩИК



Рис. 40. ДФЗ с атрибутами отношения РЕСТОРАН

Так как на представленных диаграммах каждый неключевой атрибут

нетранзитивно зависит от ключа, то все отношения находятся в 3НФ.

# 6. Реализация БД на сервере

В качестве СУБД был выбран MS SQL Server, IDE – SSMS (SQL Server Management Studio). На данном этапе необходимо создать БД в СУБД, создать необходимые таблицы, установить правильные типы и декларативные ограничения целостности.

Запросами, с помощью инструкций CREATE TABLE были созданы таблицы, и их структура: поля, типы данных полей, а также декларативные ограничения в этих полях, в том числе первичные и внешние ключи. Затем таблицы были заполнены данными.

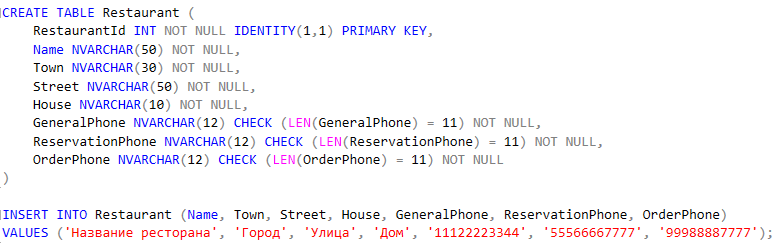


Рис. 41. Типы данных и декларативные ограничения целостности таблицы Restaurant

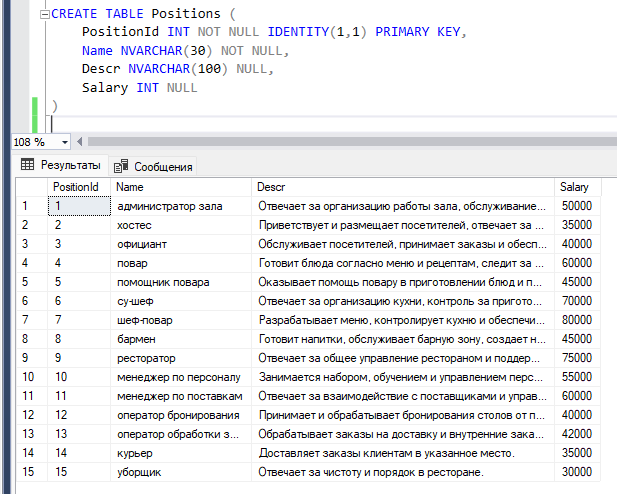


Рис. 42. Типы данных, декларативные ограничения целостности и заполнение таблицы Positions

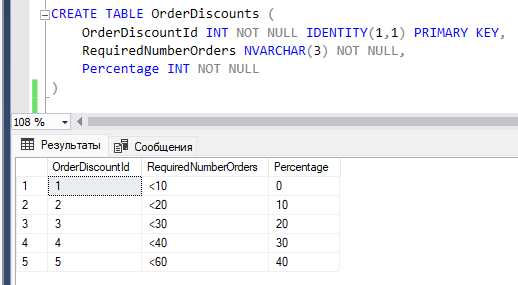


Рис. 43. Типы данных, декларативные ограничения целостности и заполнение таблицы OrderDiscounts

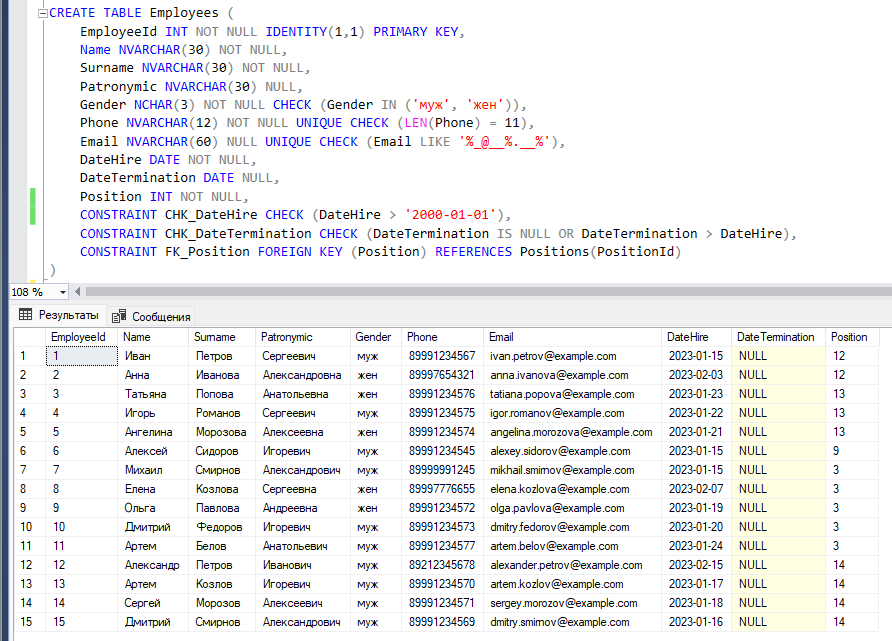


Рис. 44. Типы данных, декларативные ограничения целостности и заполнение таблицы Employees

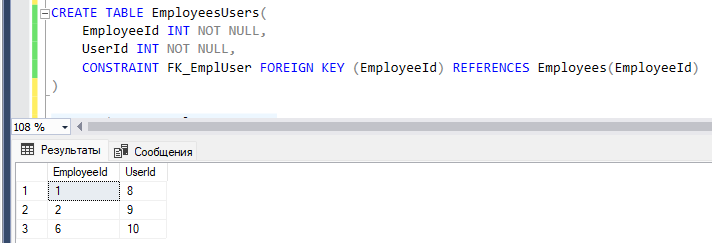


Рис. 45. Типы данных, декларативные ограничения целостности и заполнение таблицы EmployeesUsers

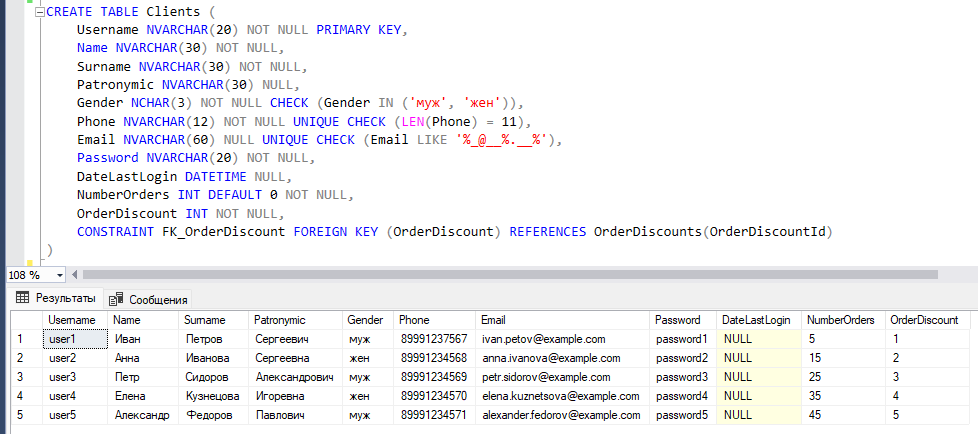


Рис. 46. Типы данных, декларативные ограничения целостности и заполнение таблицы Clients

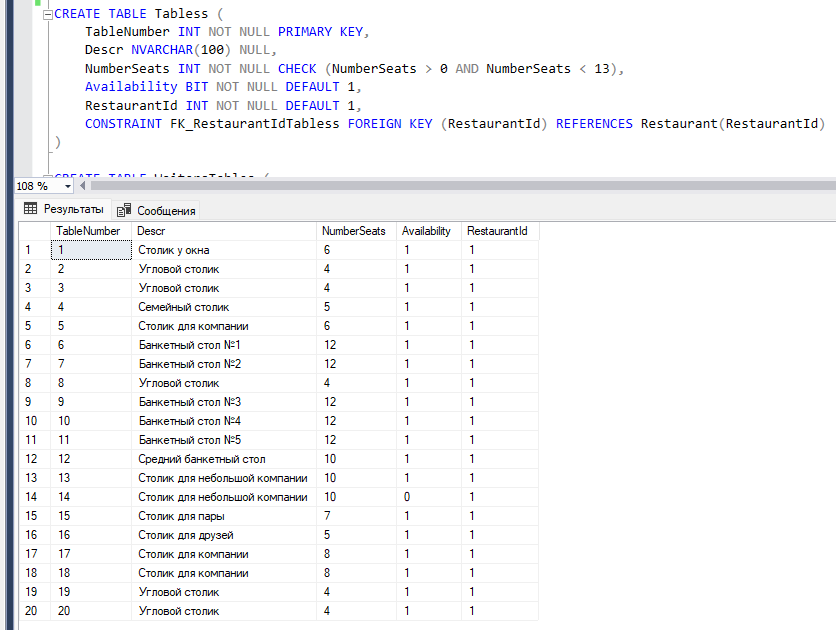


Рис. 47. Типы данных, декларативные ограничения целостности и заполнение таблицы Clients

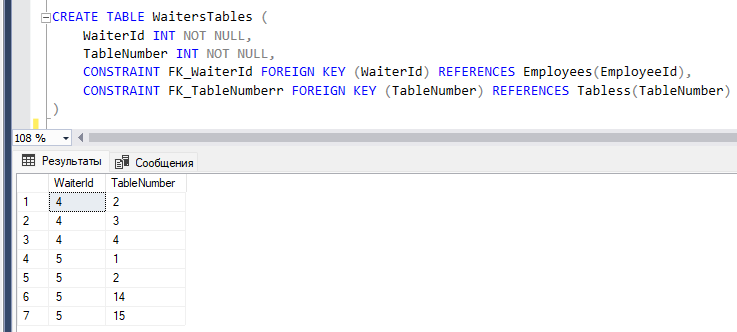


Рис. 48. Типы данных, декларативные ограничения целостности и заполнение таблицы WaitersEmployees



Рис. 49. Типы данных, декларативные ограничения целостности и заполнение таблицы Reservations

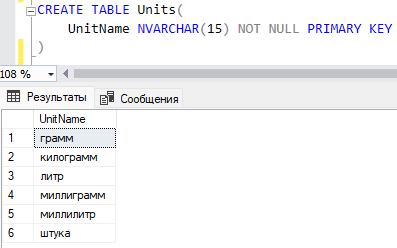


Рис. 50. Типы данных, декларативные ограничения целостности и заполнение таблицы Units

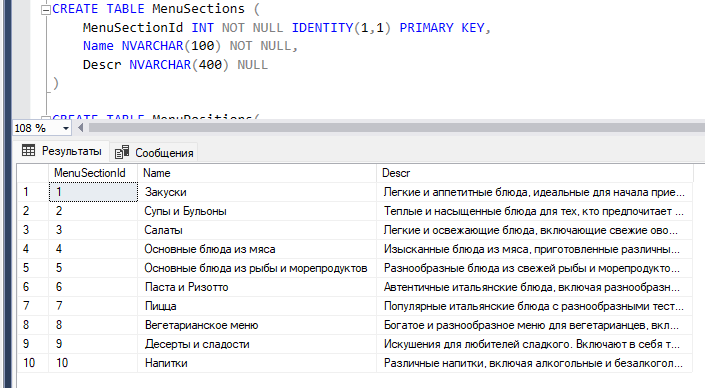


Рис. 51. Типы данных, декларативные ограничения целостности и заполнение таблицы MenuSections

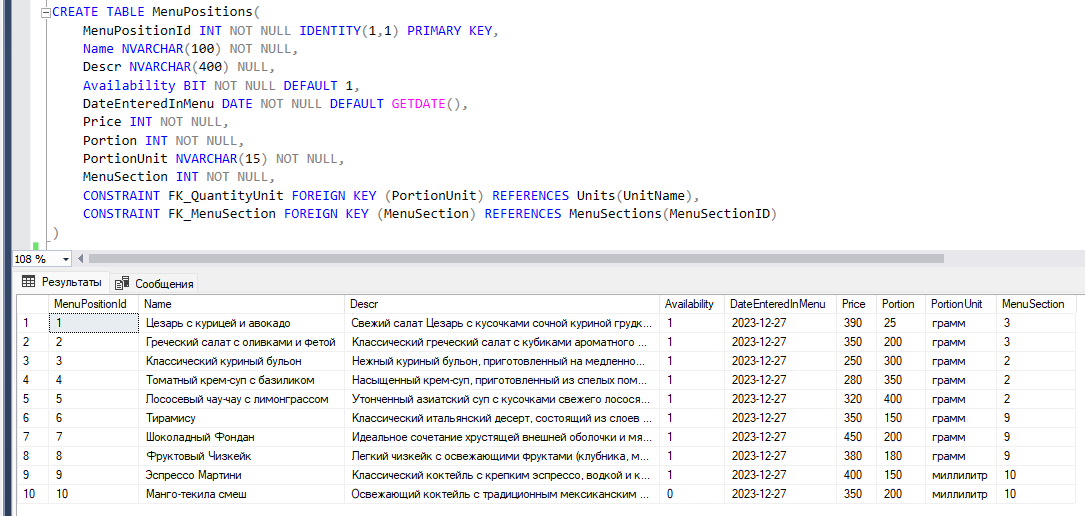


Рис. 52. Типы данных, декларативные ограничения целостности и заполнение таблицы MenuPositions

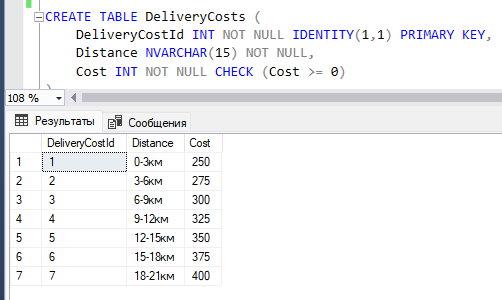


Рис. 53. Типы данных, декларативные ограничения целостности и заполнение таблицы DeliveryCosts

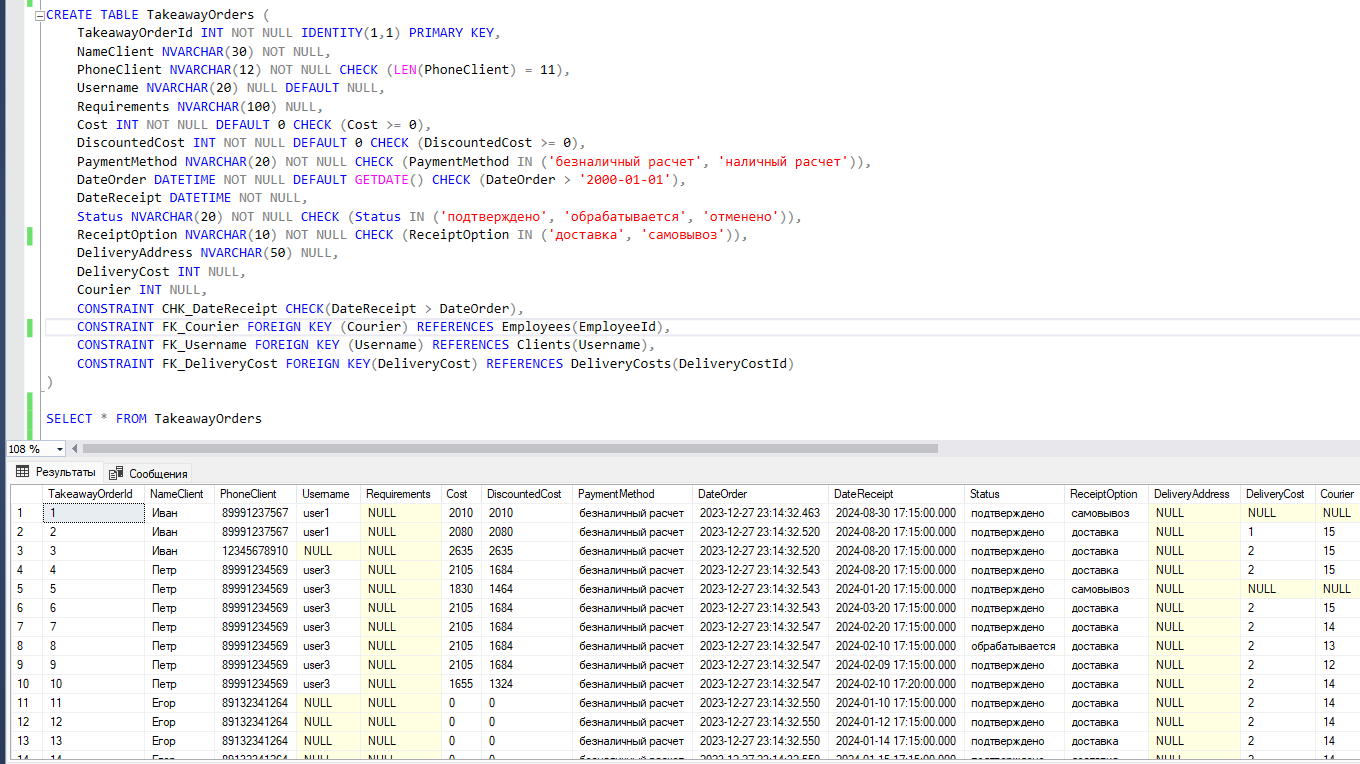


Рис. 54. Типы данных, декларативные ограничения целостности и заполнение таблицы TakeawayOrders

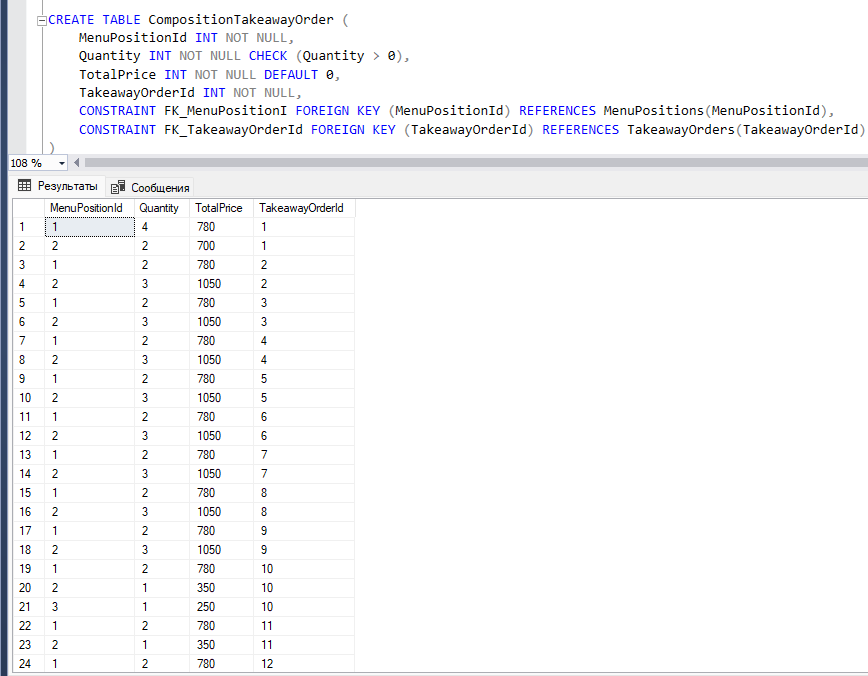


Рис. 55. Типы данных, декларативные ограничения целостности и заполнение таблицы CompositionTakeawayOrder

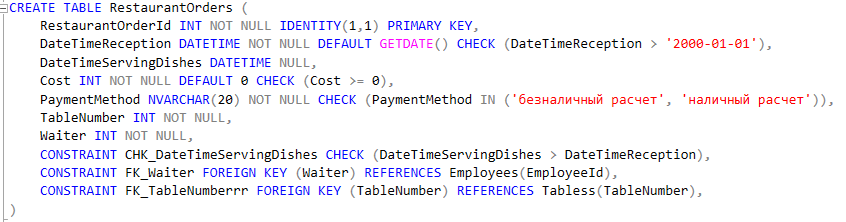


Рис. 56. Типы данных, декларативные ограничения целостности и заполнение таблицы RestaurantOrders

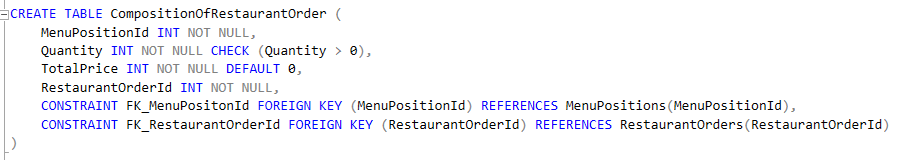


Рис. 57. Типы данных, декларативные ограничения целостности и заполнение таблицы CompositionOfRestaurantOrder

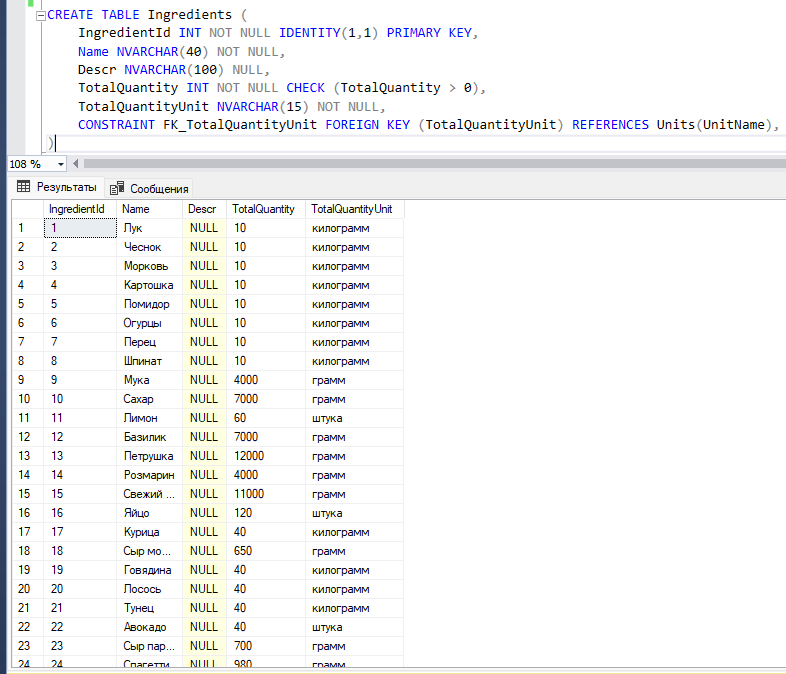


Рис. 58. Типы данных, декларативные ограничения целостности и заполнение таблицы Ingredients

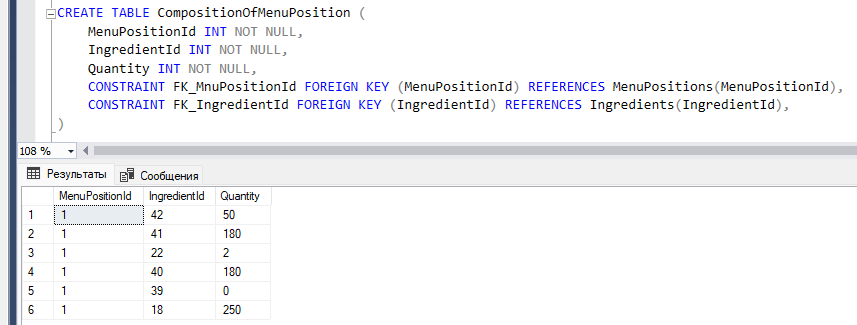


Рис. 59. Типы данных, декларативные ограничения целостности и заполнение таблицы CompositionOfMenuPosition

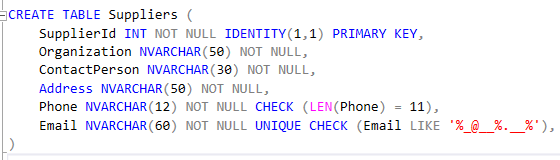


Рис. 60. Типы данных, декларативные ограничения целостности и заполнение таблицы Suppliers

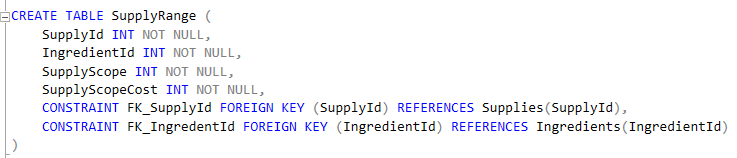


Рис. 61. Типы данных, декларативные ограничения целостности и заполнение таблицы SupplyRange

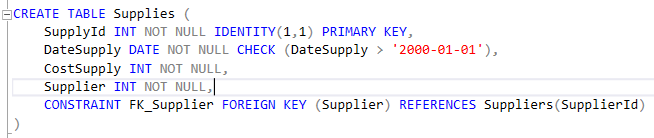
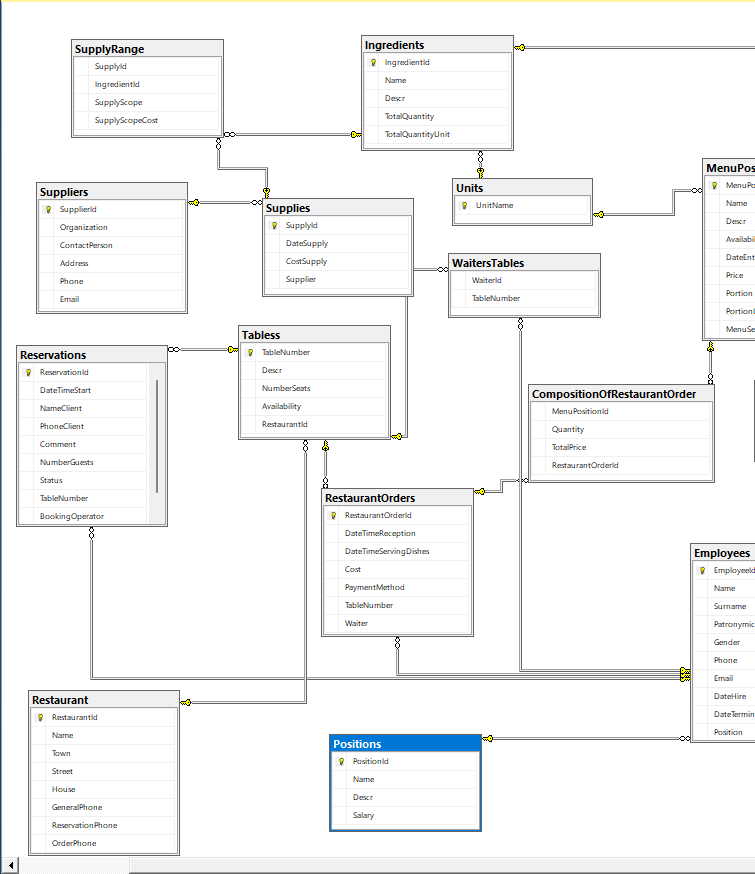


Рис. 62. Типы данных, декларативные ограничения целостности и заполнение таблицы Supplies



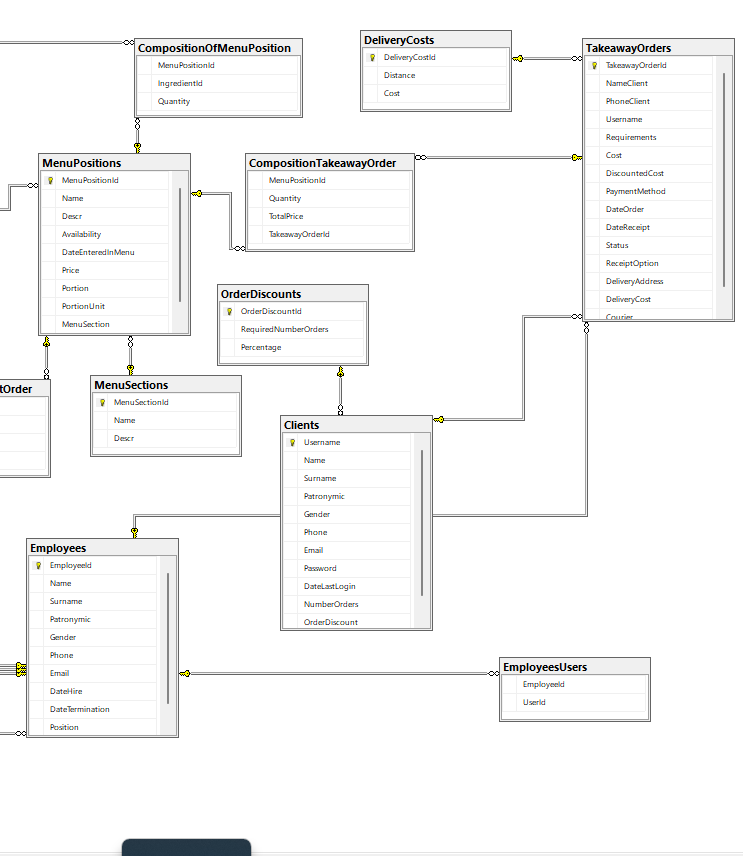


Рис. 63-64. Получившаяся диаграмма БД

# 7. Реализация объектов БД

На этом этапе необходимо дополнительно реализовать следующие объекты БД: хранимые процедуры, триггеры и индексы. Хранимые процедуры представляют собой блоки кода, которые можно вызывать непосредственно на фронтенде, что обеспечивает дополнительную безопасность. Триггеры необходимы для реализации процедурных ограничений целостности при модификации данных. Индексы необходимы для оптимизации поиска данных в таблицах.

# 7.1 Хранимые процедуры

Хранимые процедуры представляют собой предварительно скомпилированный набор SQL-операторов, который хранится и обрабатывается на сервере баз данных. Это обеспечивает дополнительный уровень безопасности. В моем случае все созданные хранимые процедуры можно разделить на 3 типа: хранимые процедуры, реализующие аналитические запросы, хранимые процедуры, осуществляющие модификацию данных и хранимые процедуры, нужные на фронтенде для выполнения дополнительных целей.

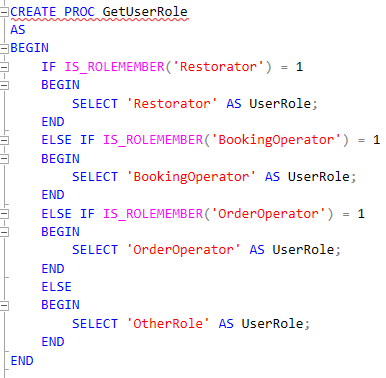


Рис. 65. Хранимая процедура GetUserRole, возвращающая роль текущего пользователя, который создал текущее соединение с БД (используется на фронтенде)

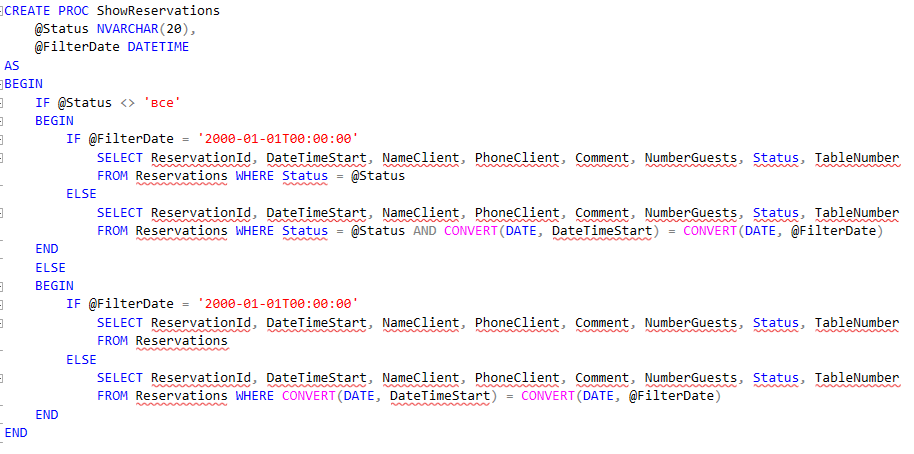


Рис. 66. Хранимая процедура ShowReservations, показывающая список всех бронирований в отфильтрованном виде (используется в фильтре на фронтенде)

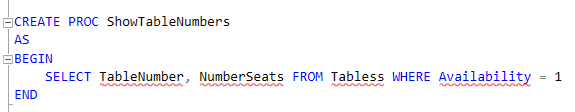


Рис. 67. Хранимая процедура ShowTableNumbers, возвращающая список всех доступных столиков (используется на фронтенде)

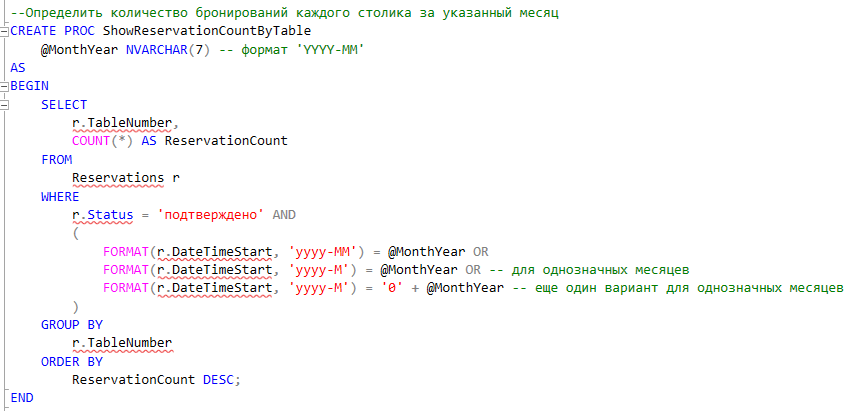


Рис. 68. Хранимая процедура ShowTableNumbers, определяющая количество бронирований каждого столика за указанный месяц (аналитический запрос)

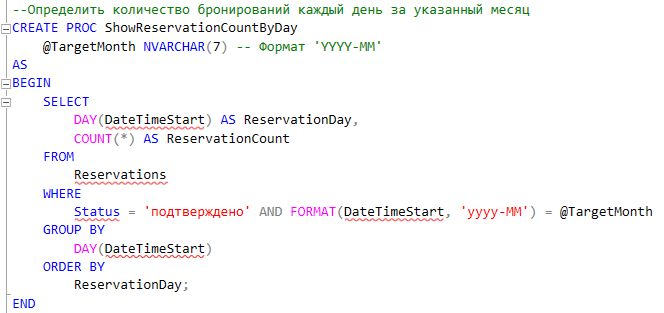


Рис. 69. Хранимая процедура ShowReservationCountByDay, определяющая количество бронирований каждый день за указанный месяц (аналитический запрос)



Рис.70. Хранимая процедура ShowMostPopularDaysOfWeekForTakeawayOrders, определяющая самые популярные дни недели по количеству заказов на вынос за указанный месяц (аналитический запрос)

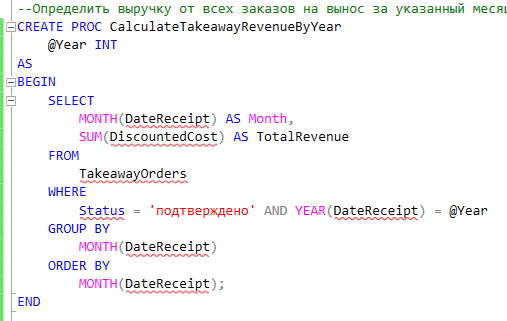


Рис. 71. Хранимая процедура CalculateTakeawayRevenueByYear, определяющая выручку от всех заказов на вынос за указанный месяц (аналитический запрос)

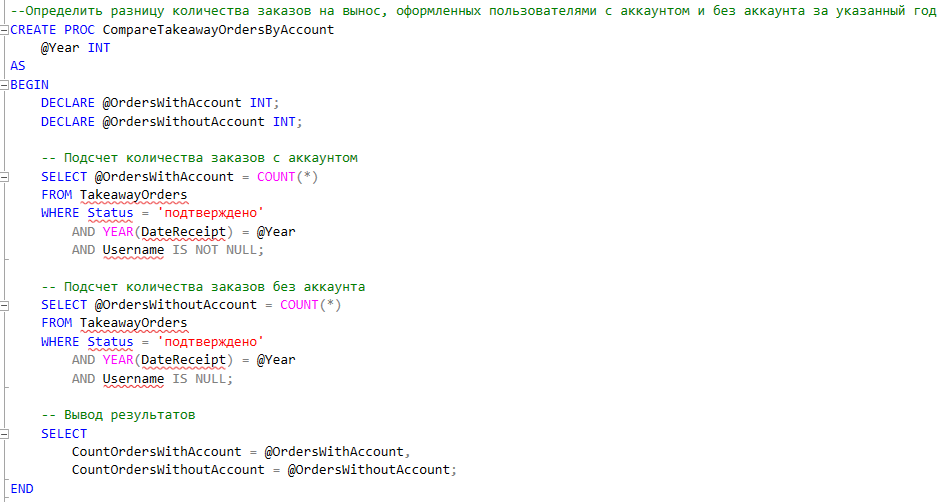


Рис. 72. Хранимая процедура CompareTakeawayOrdersByAccount, определяющая разницу количества заказов на вынос, оформленных пользователями с аккаунтом и без аккаунта за указанный год (аналитический запрос)

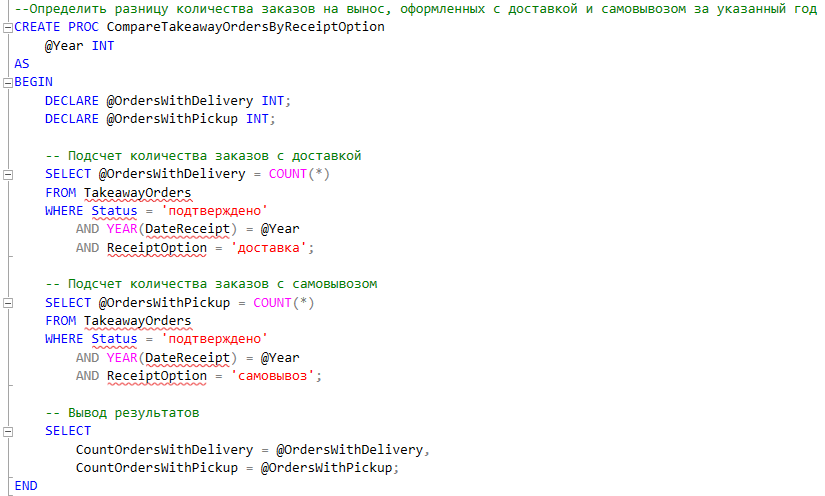


Рис. 73. Хранимая процедура CompareTakeawayOrdersByReceiptOption, определяющая разницу количества заказов на вынос, оформленных с доставкой и самовывозом за указанный год (аналитический запрос)

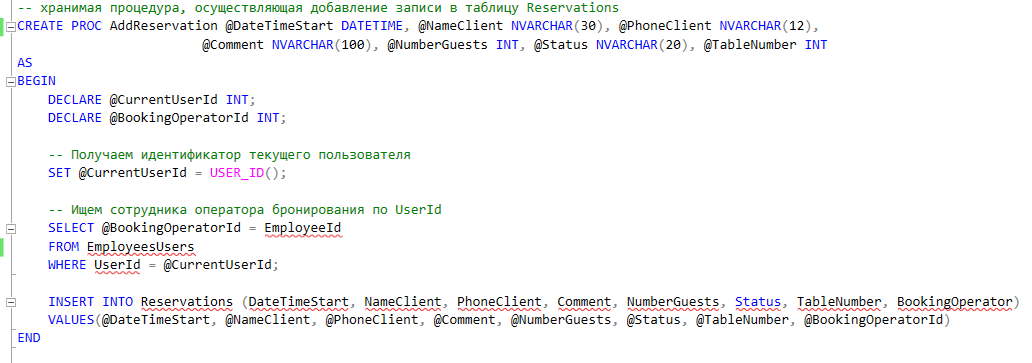


Рис. 74. Хранимая процедура AddReservation, осуществляющая добавление записи в таблицу Reservations (модификация данных)

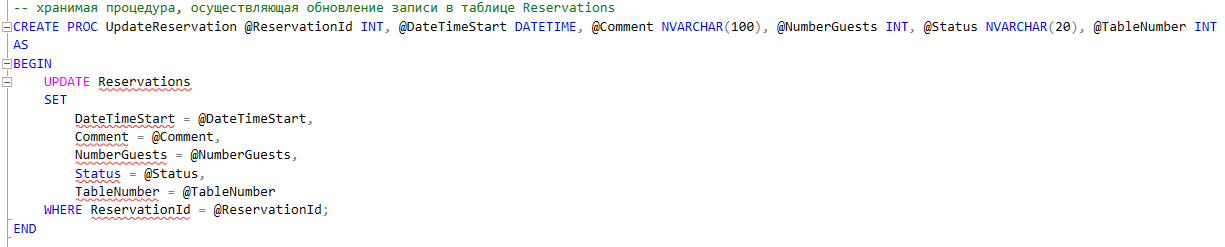


Рис. 75. Хранимая процедура UpdateReservation, осуществляющая обновление записи в таблице Reservations (модификация данных)

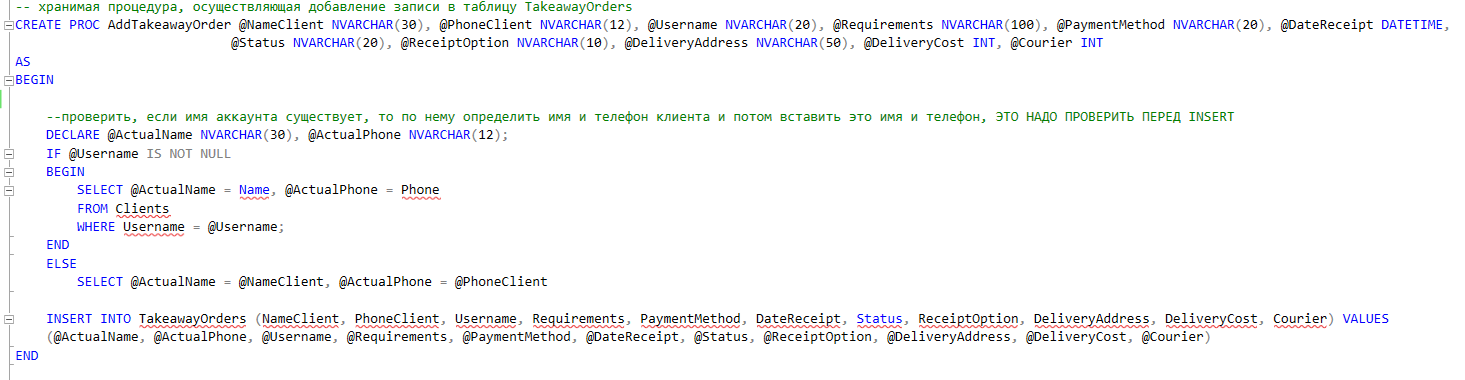


Рис. 76. Хранимая процедура AddTakeawayOrder, осуществляющая добавление записи в таблицу TakeawayOrders (модификация данных)

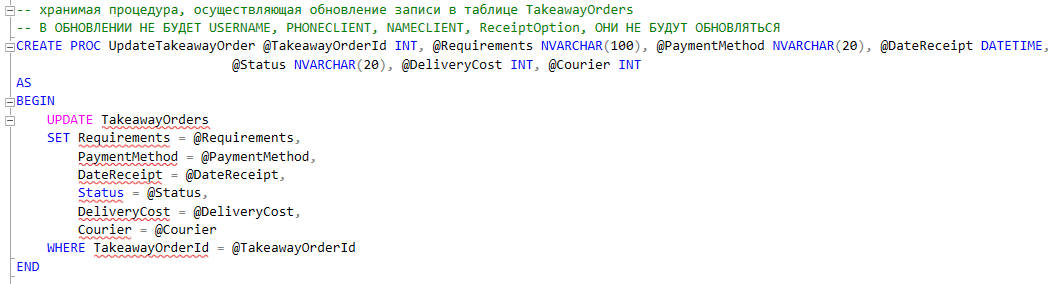


Рис. 77. Хранимая процедура UpdateTakeawayOrder, осуществляющая обновление записи в таблице TakeawayOrders (модификация данных)

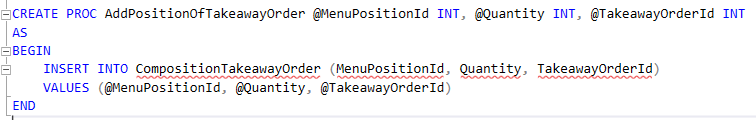


Рис. 78. Хранимая процедура AddPositionOfTakeawayOrder, осуществляющая добавление записи в таблицу CompositionTakeawayOrder (модификация данных)

# 7.2 Триггеры

Триггеры – это особый вид хранимых процедур, то есть его тело тоже может в себя включать множество операторов SQL, выполняется автоматически при возникновении событий на сервере БД. В моем случае, используются триггеры DML, которые выполняются в случае модификации данных в таблицах. С помощью триггеров проверяются процедурные ограничения целостности, для которых необходимо писать определенную логику.

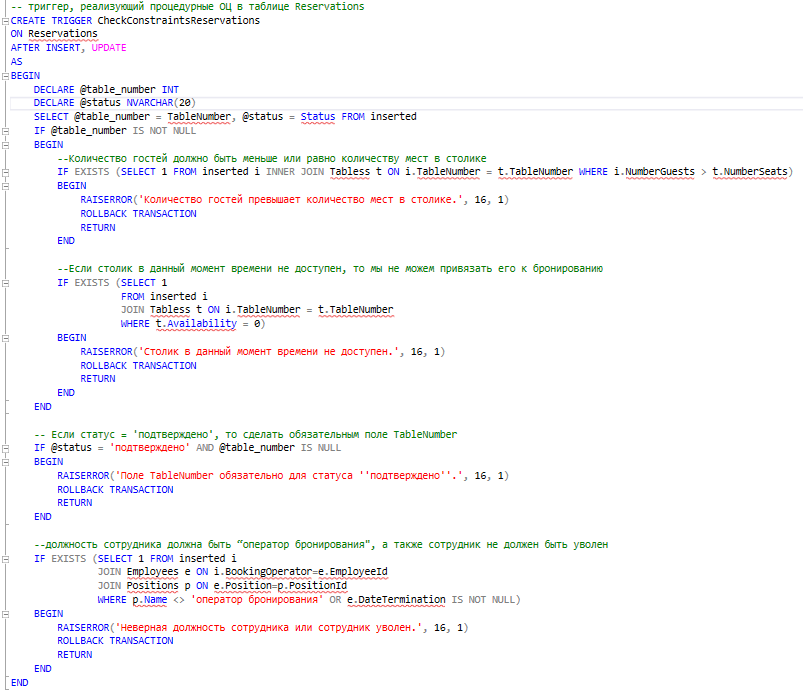
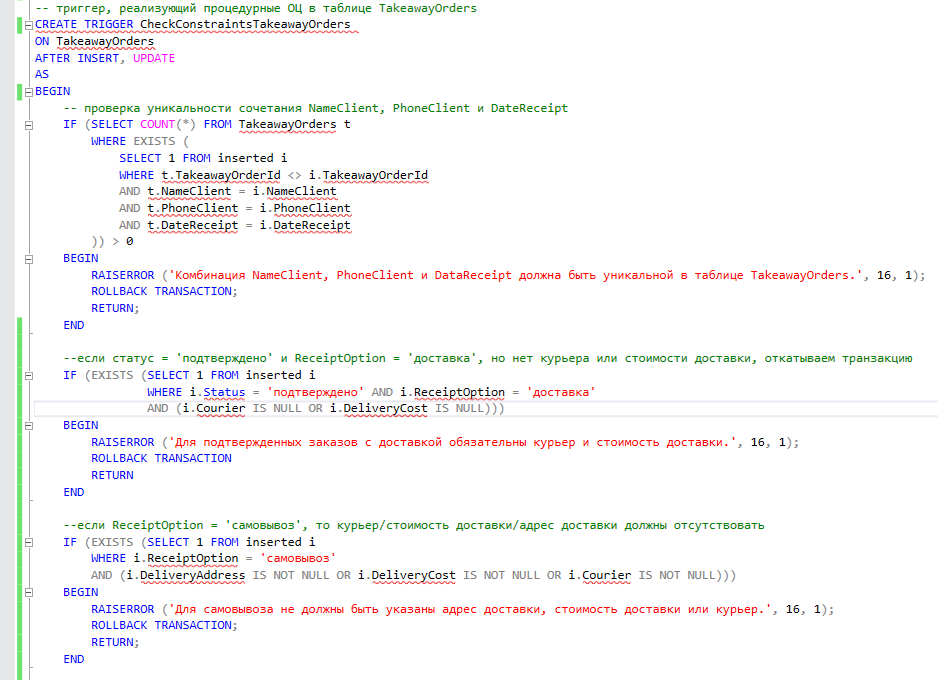
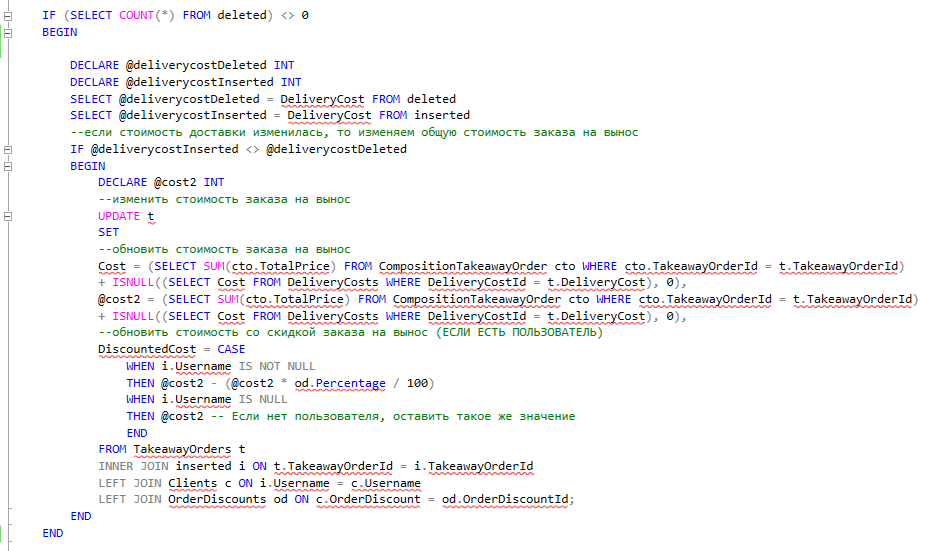


Рис. 79. Триггер CheckConstraintsReservations, реализующий процедурные ОЦ в таблице Reservations





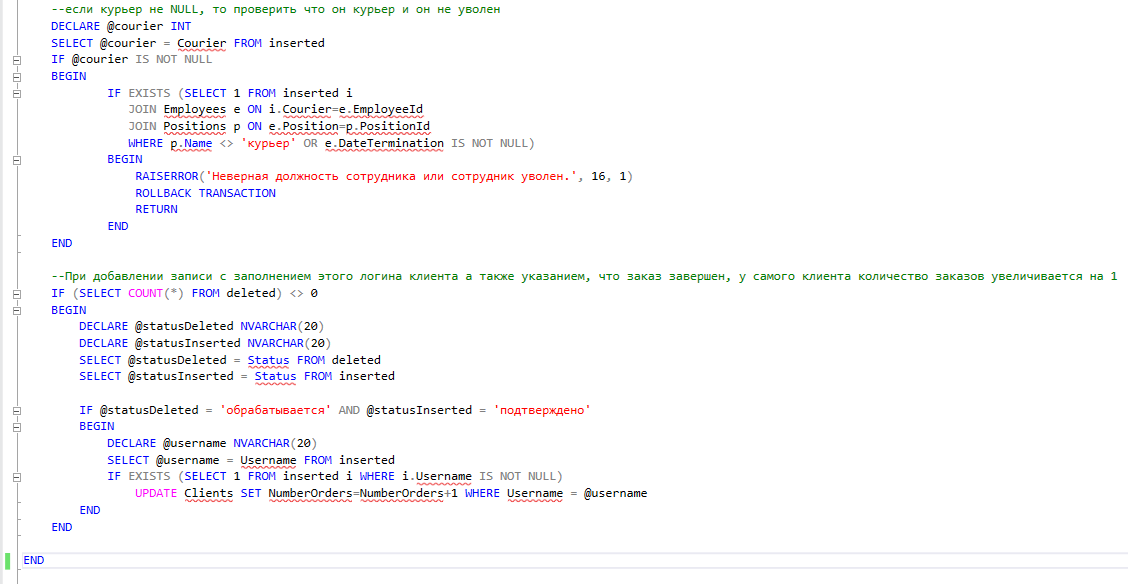
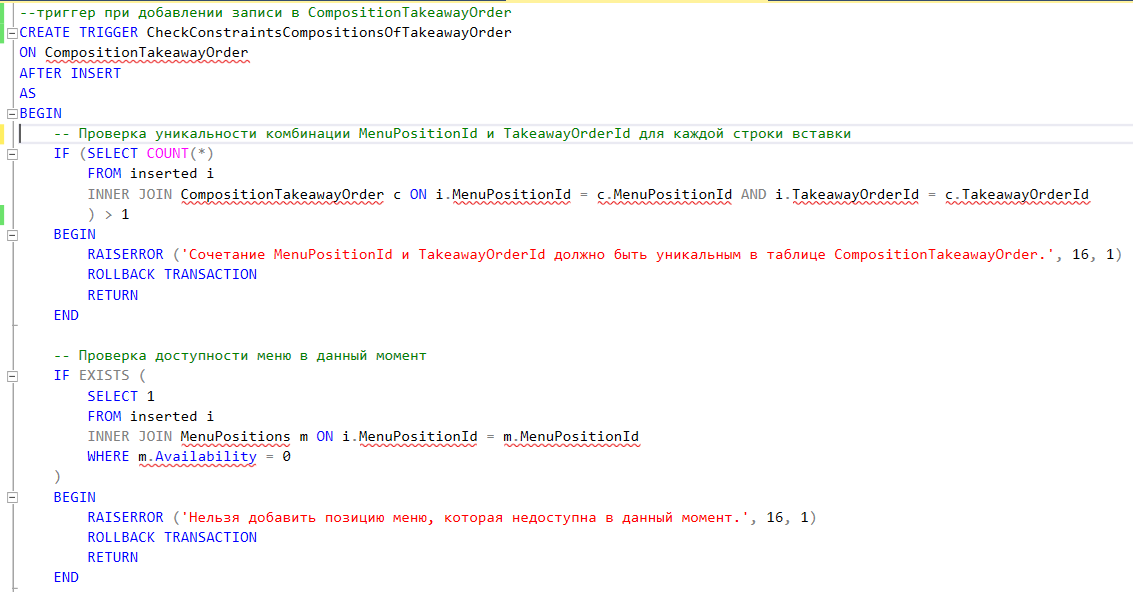


Рис. 80-82. Триггер CheckConstraintsTakeawayOrders, реализующий процедурные ОЦ в таблице TakeawayOrders



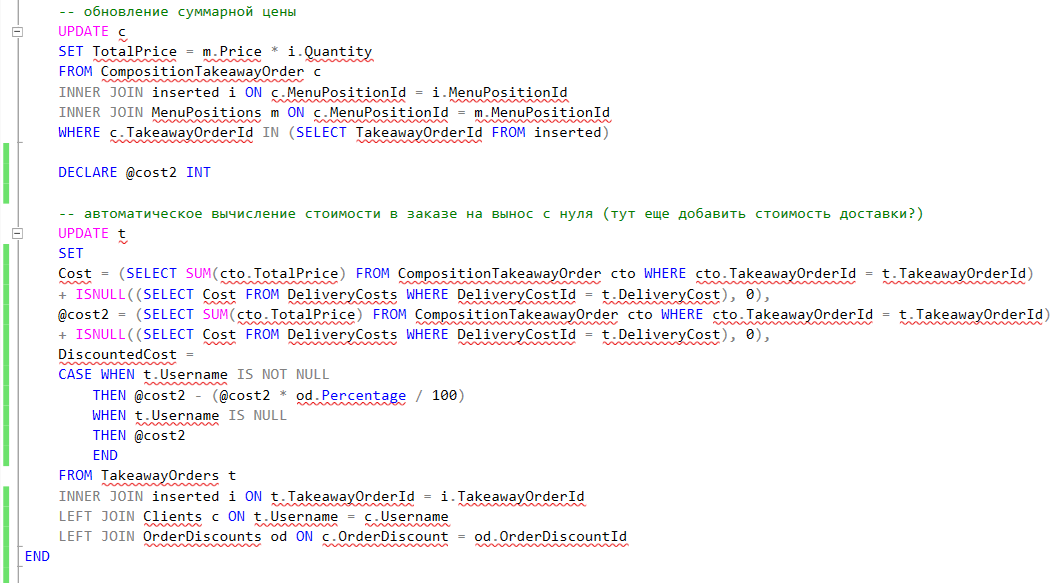


Рис. 83-84. Триггер CheckConstraintsCompositionsOfTakeawayOrder, реализующий процедурные ОЦ в таблице CompositionTakeawayOrder

# 7.3 Индексы

Индексы в базах данных представляют собой структуры данных, созданные для оптимизации производительности запросов. Они ускоряют поиск и сортировку данных в таблицах и обеспечивают быстрый доступ к данным. Большую часть индексов я создал для таблиц Reservations и TakeawayOrders, потому что большая часть select-запросов происходит к ним в хранимых процедурах.



Рис. 85. Все созданные индексы

# 8. Реализация системы безопасности

На данном этапе необходимо в соответствии с таблицей «Пользователи системы и их задачи» настроить права доступа пользователей разных категорий к объектам БД (в моем случае, это только хранимые процедуры). Создание ролей в базе данных – это важный механизм обеспечения безопасности данных. Роли позволяют группировать пользователей и предоставлять определенные права доступа к объектам базы данных. Я могу привязывать к роли те объекты, которые должны быть ему доступны.

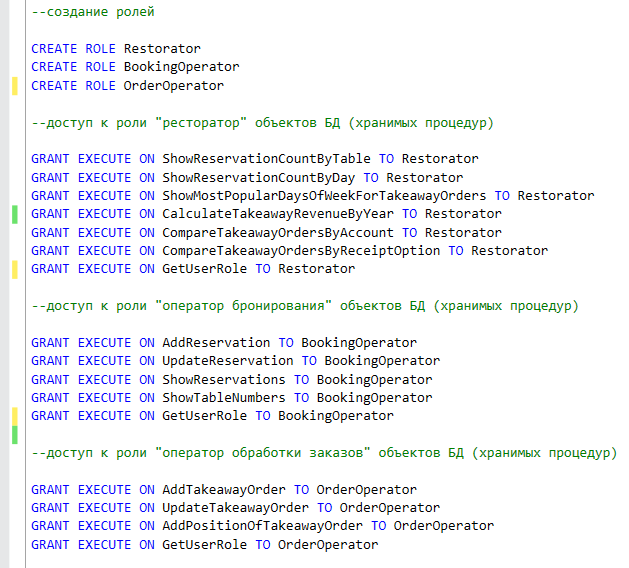


Рис. 86. Создание ролей и привязка каждой роли определенных хранимых процедур

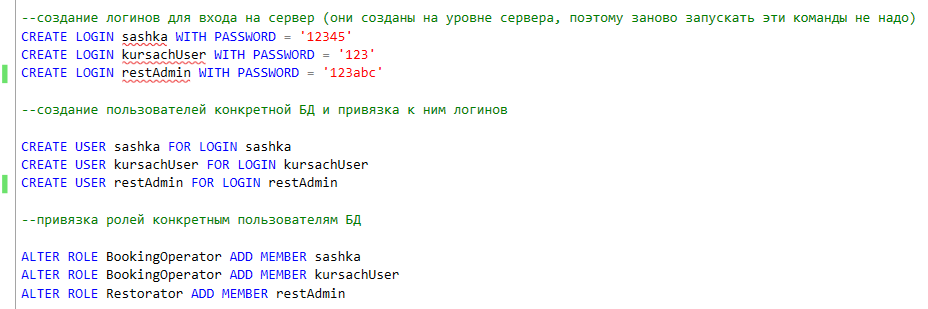


Рис. 87. Создание пользователей БД и привязка к конкретным ролям этих пользователей

# 9. Реализация приложения для работы с БД

На данном этапе необходимо создать приложение, которое обеспечивает доступ по сети к БД и используют созданные объекты БД (в моем случае, хранимые процедуры). Данное приложение является приложением для сотрудников-пользователей БД, которым предоставляются формы для просмотра и модификации данных из БД.

В приложении есть одна форма, осуществляющая добавление/обновление записи в таблице Reservations. Также есть форма, которая отображает результаты аналитических запросов в виде различных диаграмм. В приложении есть система авторизации, которая работает с пользователями, созданными в рамках 8 этапа, таким образом подключаясь к серверу под конкретным пользователем.

В качестве типа приложения выбрано десктопное приложение WinForms, язык программирования – C#, IDE – Visual Studio 2022. Для работы с СУБД ms sql server использовалась мощная технология ADO.NET, предоставляющая множество классов и компонентов для взаимодействия с базами данных в среде .NET Framework.

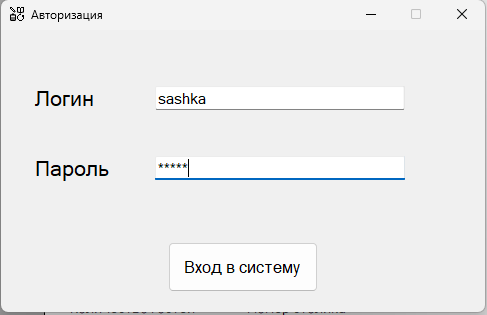


Рис. 88. Форма авторизации

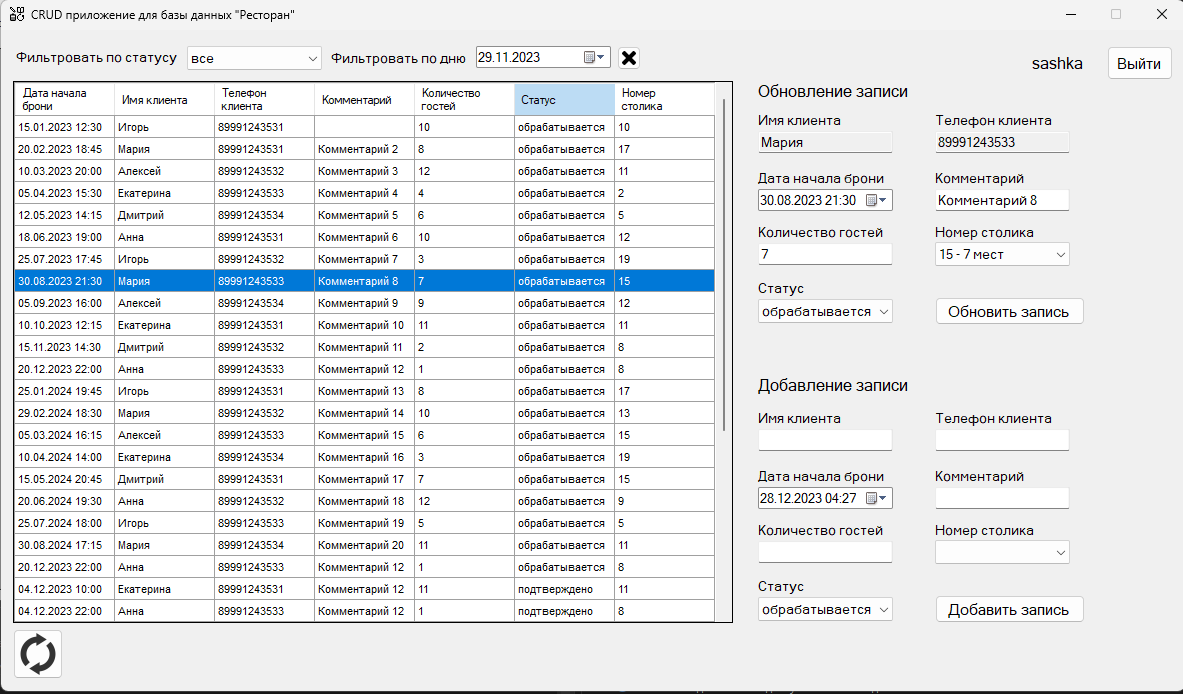


Рис. 89. Форма для модификации данных в таблице Reservations для пользователей с ролью=BookingOperator

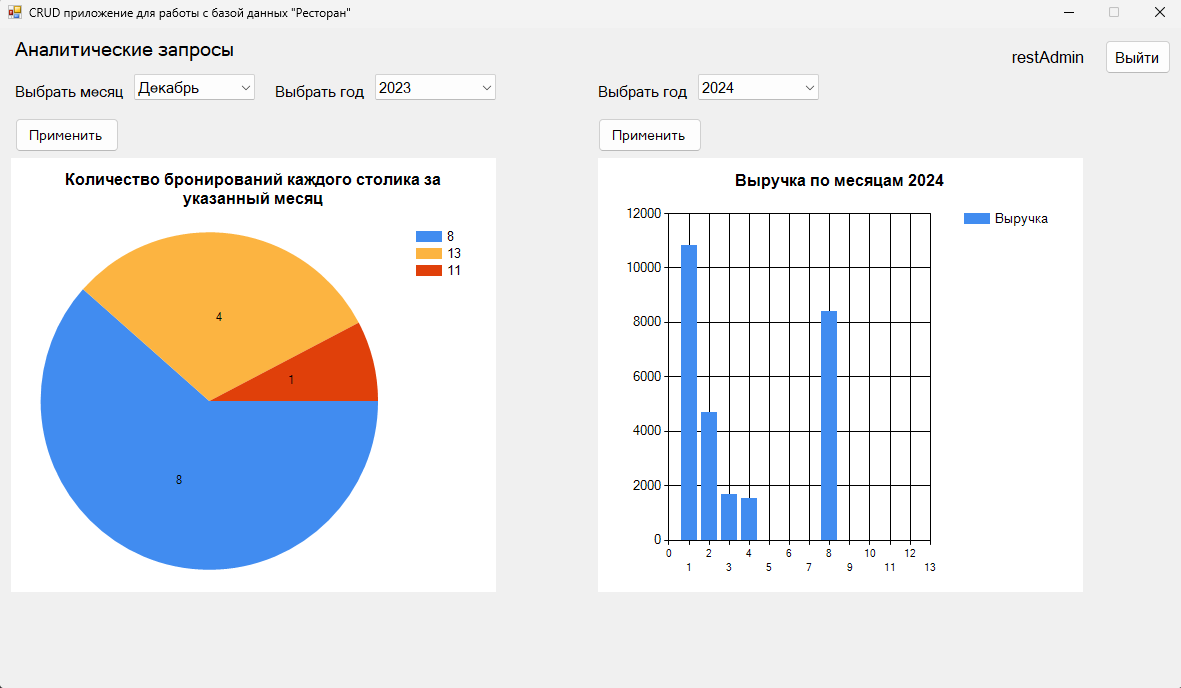


Рис. 90. Форма для отображения результатов аналитических запросов в виде различных диаграмм для пользователей с ролью=Restorator

# 10. Заключение

В ходе данного курсового проекта была создана база данных с

использованием СУБД MS SQL Server. В ней реализованы все основные

аспекты современных баз данных, в том числе необходимые объекты БД и

язык запросов SQL.

Целями работы являлись:

* получение навыков выполнения проектных процедур при создании реляционных баз данных
* получение навыков работы с БД, функционирующей в рамках архитектуры "клиент-сервер".

Цели данной работы были достигнуты и все поставленные задачи

выполнены.

Достоинства курсового проекта:

1) готовность к будущему масштабированию и разрастанию системы. База данных и структура системы спроектированы таким образом, чтобы легко внедрять изменения, добавлять новые функции и поддерживать рост объемов данных и пользователей. Например, можно добавить систему сети ресторанов: в таком случае уже есть отдельная таблица «Ресторан», а также можно будет добавить таблицу «Меню», которая будет относиться к конкретному ресторану.  
2) актуальность. Все современные приложения работают с БД.

Недостатки курсового проекта:1) не до конца реализованные объекты БД, которые были определены при проектировании, – хранимые процедуры, триггеры, отсутствие некоторых выявленных в ходе проектирования ролей, приложение, которое работает не со всеми созданными объектами БД

2) отсутствие реальных данных. Курсовой проект часто использует вымышленные или упрощенные данные. Это может не отражать сложности, связанные с реальными данными и их разнообразием, что могло бы быть важным аспектом в реальном мире.

Таким образом, было получено большое количество знаний проектирования и реализации БД.

# 11. Список используемой литературы

1. METANIT.COM: [сайт]. - 2018. - URL: https://metanit.com/sharp/adonetcore/

(дата публикации: 03.03.2021). - Текст: электронный.

2. SQL Server technical documentation : [сайт] - URL:

https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16 - Текст:

электронный

3.СДО КГУ: Курс Базы данных - URL:

https://sdo.ksu.edu.ru/course/view.php?id=2654. - Текст: электронный