МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



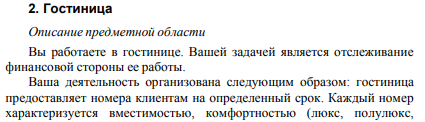


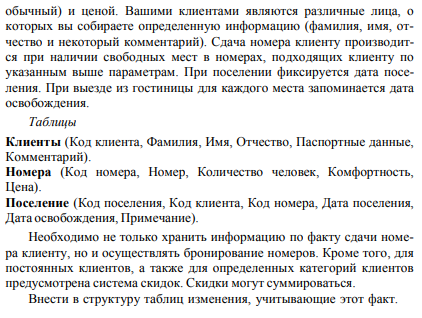
Отчет по курсовой работе

по дисциплине «Проектирование и администрирование баз данных»  
«Гостиница»

| Группа | 221-329 |
| --- | --- |
| Студент | Баденов Кирилл Дмитриевич |
| Дата | 22.06.2023 |
| Преподаватель | Евдошенко Олег Игоревич |

# ЗАДАНИЕ





**СОДЕРЖАНИЕ**

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_tyjcwt)

[1.1 Предметная область 5](#_3dy6vkm)

[1.2 Основные бизнес-процессы 5](#_1t3h5sf)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ДАННЫХ 6](#_4d34og8)

[2.1 Концептуальное проектирование базы данных 6](#_2s8eyo1)

[2.2 Логическое проектирование базы данных 8](#_17dp8vu)

[2.3 Выбор и описание СУБД 9](#_3rdcrjn)

[2.4 Физическое проектирование базы данных 10](#_26in1rg)

[3 ИСПОЛЬЗОВАНИЕ БАЗЫ ДАННЫХ 11](#_lnxbz9)

[3.1 Создание DataSet 16](#_35nkun2)

[3.2 Добавление данных в базу 20](#_1ksv4uv)

[3.3 Запросы к базе данных 28](#_44sinio)

[3.4 Тестирование функций и триггеров 33](#_2jxsxqh)

# ВВЕДЕНИЕ

## Актуальность проблемы

Модернизация структуры базы данных является неотъемлемой частью деятельности современных компаний. Это обусловлено несколькими факторами:

1. быстрое развитие технологий требует обновления и адаптации баз данных к новым возможностям и объемам информации;
2. изменения в бизнес-процессах требуют согласованной структуры баз данных для эффективного управления информационными потоками;
3. адаптация к изменяющемуся законодательству требует своевременной модернизации баз данных, чтобы соответствовать обязательствам по хранению и обработке пользовательских данных.

Учет всех этих факторов позволяет компаниям обеспечивать гибкость, эффективность и защиту интересов клиентов, делая модернизацию баз данных важной составляющей успешной бизнес-стратегии

## Цель работы:

Систематизация и закрепление полученных теоретических и практических умений по разработке баз данных с использованием СУБД PostgreSQL.

## Задачи работы:

1. изучить литературу по проектированию и разработке базы данных в СУБД PostgreSQL;
2. получить практический опыт разработки базы данных в СУБД PostgreSQL.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## Предметная область

Вы работаете в гостинице. Вашей задачей является отслеживание финансовой стороны ее работы. Ваша деятельность организована следующим образом: гостиница предоставляет номера клиентам на определенный срок. Каждый номер характеризуется вместимостью, комфортностью (люкс, полулюкс, обычный) и ценой. Вашими клиентами являются различные лица, о которых вы собираете определенную информацию (фамилия, имя, отчество и некоторый комментарий). Сдача номера клиенту производится при наличии свободных мест в номерах, подходящих клиенту по указанным выше параметрам. При поселении фиксируется дата поселения. При выезде из гостиницы для каждого места запоминается дата освобождения.

Необходимо не только хранить информацию по факту сдачи номера клиенту, но и осуществлять бронирование номеров. Кроме того, для постоянных клиентов, а также для определенных категорий клиентов предусмотрена система скидок. Скидки могут суммироваться.

Анализ предметной области показал, что необходимо будет хранить информацию о бронировании номеров, также нужно будет ввести систему категорий клиентов, при условии, что каждый клиент может принадлежать к нескольким категориям, важным аспектом данной предметной области является корректная обработка доступных номеров, при заселении нам нужно учитывать занят номер или нет, забронирован ли он, корректно обновлять информацию о бронировании и доступности комнат.

## Основные бизнес-процессы

При изучении предметной области было выявлено 5 бизнес-процессов Подробнее с ними можно ознакомиться в «Таблице 1 – Список основных бизнес-процессов».

Таблица 1 - Список основных бизнес-процессов

| Номер бизнес-процесса | Наименование бизнес процесса |
| --- | --- |
| 1 | Добавление нового клиента |
| 2 | Бронирование номера |
| 3 | Процесс заселения и выселения |
| 4 | Выдача клиенту определенных категорий |
| 5 | Управление финансами |

# 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ

## 2.1 Концептуальное проектирование базы данных

При анализе предметной области были выявлены следующие сущности: «клиенты», «номера».

Сущность «клиенты» содержит следующие атрибуты:

1. фамилия;
2. имя;
3. отчество;
4. паспортные данные;
5. комментарий;

Сущность «номера» содержит следующие атрибуты:

1. номер комнаты;
2. вместимость;
3. уровень комфорта;
4. цена;
5. статус занятости.

Связи между сущностями отображены в «Рисунке 1 – концептуальная модель данных».

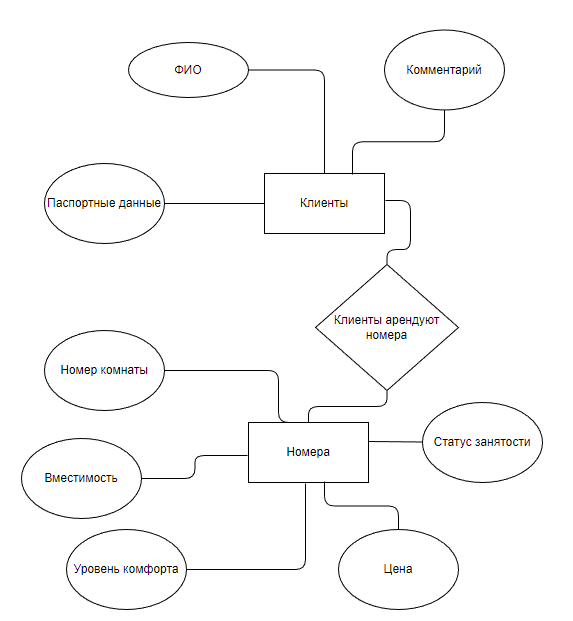


Рисунок 1 – концептуальная модель данных

Структуры сущностей приведены в последующих таблицах: «Таблица 2 – структура сущности клиент», «Таблица 3 – структура сущности номера».

Таблица 2 – структура сущности клиент

| Название параметра | Тип данных | Размер | Диапазон значений |
| --- | --- | --- | --- |
| Фамилия | Текстовый | До 30 |  |
| Имя | Текстовый | До 30 |  |
| Отчество | Текстовый | От 0 до 30 |  |
| Паспортные данные | Текстовый | 11 |  |
| Комментарий | Текстовый | от 0 до 255 |  |

Таблица 3 – структура сущности номера

| Название параметра | Тип данных | Размер | Диапазон значений |
| --- | --- | --- | --- |
| Номер комнаты | Числовой |  | От 0 |
| Вместимость | Числовой |  | От 0 |
| Уровень комфорта | Текстовый | От 4 до 8 | Обычный, Люкс, Полулюкс |
| Цена | Числовой |  | От 0 |
| Статус занятости | Числовой | 1 | false - номер свободен, true - в номере проживают |

## 2.2 Логическое проектирование базы данных

Для выявленных сущностей была создана логическая схема базы данных (Рисунок 2 – логическая схема базы данных).

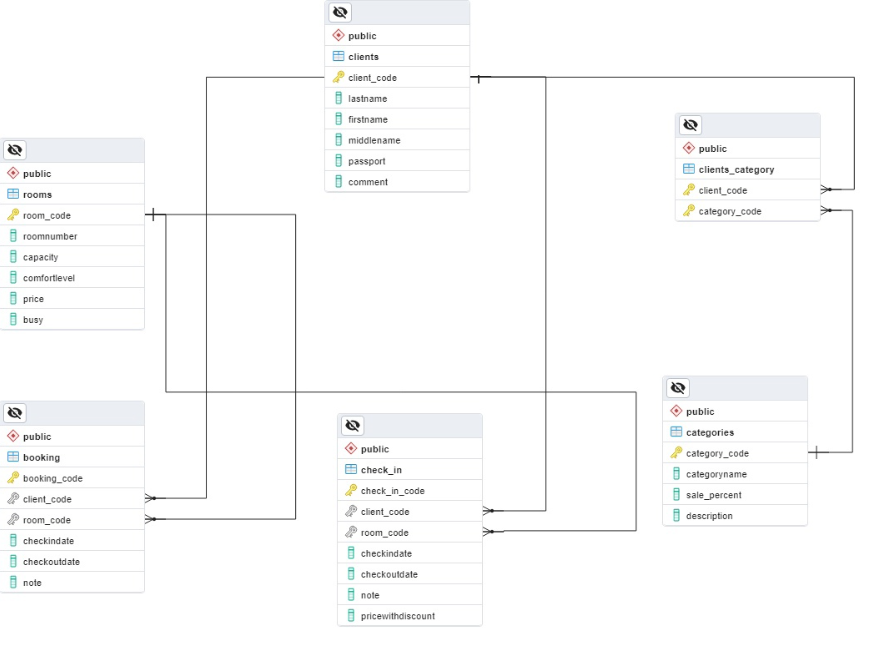


Рисунок 2 – логическая модель данных

## 2.3 Выбор и описание СУБД

Для выполнения курсовой работы по базам данных была выбрана СУБД PostgreSQL. PostgreSQL является СУБД с открытым исходным кодом, что позволяет более глубоко изучить ее внутреннее устройство. Она также обладает широким набором функций, которые обеспечивают гибкость при разработке баз данных, и поддерживает различные типы данных. Надежность и стабильность PostgreSQL являются важными преимуществами. Эта СУБД активно поддерживается и разрабатывается сообществом разработчиков, что гарантирует безопасность и исправление ошибок. Встроенные механизмы резервного копирования, восстановления и журналирования обеспечивают сохранность данных и доступность базы данных. Кроме того, PostgreSQL обладает высокой производительностью и масштабируемостью. Она эффективно обрабатывает большие объемы данных, поддерживает параллельную обработку запросов, оптимизацию запросов и кэширование. Наконец, PostgreSQL имеет широкое сообщество разработчиков и пользователей, а также обширную документацию. Это упрощает изучение и работу с PostgreSQL, а также предоставляет возможность получить поддержку и советы от опытных пользователей. Все эти факторы делают PostgreSQL отличным выбором для реализации курсовой работы.

## 2.4 Физическое проектирование базы данных

Используя функционал PostgreSQL была создана физическая схема базы данных (Рисунок 3 – физическая модель данных).

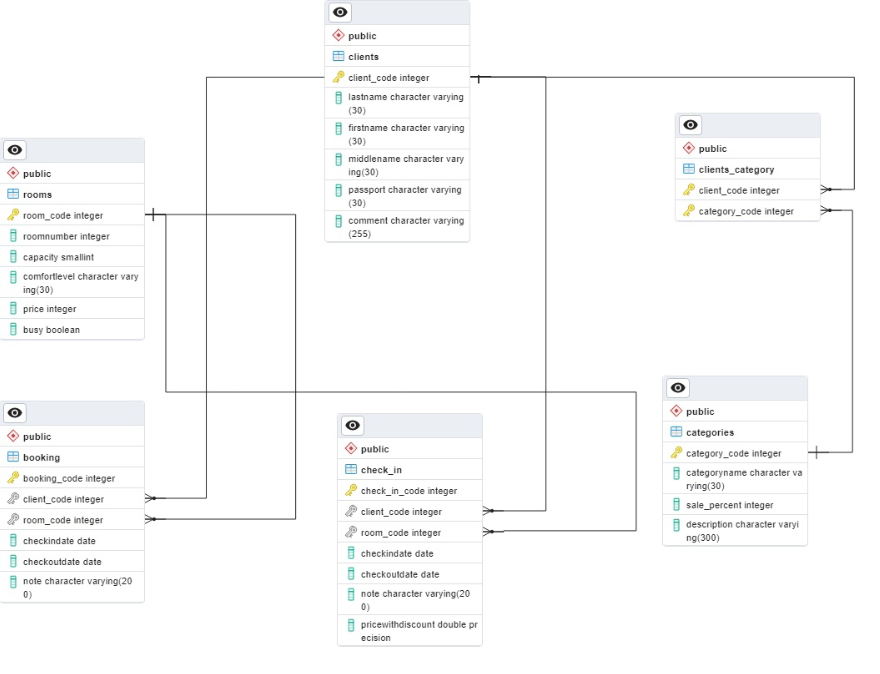
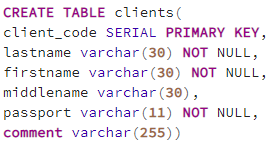


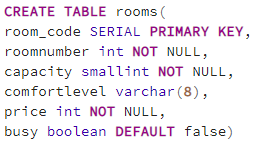
Рисунок 3 – физическая модель данных

Далее модель была реализована при помощи PostgreSQL. Для создания таблиц и связей между ними были использованы SQL-запросы, которые указаны в «Листинге 1 – инициализация базы данных».

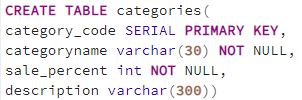
Листинг 1 – инициализация базы данных



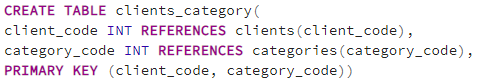
Создание таблицы clients, описывающей данные о клиентах, атрибуты middlename и comment могут быть пустыми, так как клиент может не иметь отчества и к нему не обязательно будет приписан комментарий.



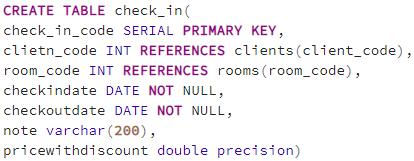
Создание таблицы rooms, описывающей данные о номерах, атрибут comfortlevel может быть пустым, так как далее будет создана триггерная функция, которая на основании стоимости номера, будет определять его уровень комфорта, значение атрибута busy по умолчанию false, так как изначально все номера не заняты.



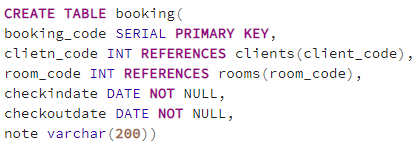
Создание таблицы categories, описывающей все возможные категории клиентов, атрибут description не должен быть пустым, так как это не простая заметка, а описание категорие, без которого не будет понятно, какие клиенты принадлежат к данной категории.



Создание таблицы clients\_category, описывающей какие категории есть у клиентов, данная таблица имеет связь многие-ко-многим, так как каждый клиент может принадлежать к нескольким категориям, а каждая категория может описывать несколько клиентов, из-за этого отношения в данной таблице используется составной ключ из двух атрибутов.



Создание таблицы check\_in, описывающей заселение клиентов в номера, данная таблица имеет два внешних ключа, для связи с таблицами clients и rooms, note может быть пустым атрибутом, так как далее сюда будет записываться слово «Выселен» при выселении клиента, pricewithdiscount может быть пустым атрибутом, так как далее будет создана триггерная функция, которая будет считать стоимость номера для клиентам, с учетом скидок от категорий, к которым принадлежит данный клиент.



Создание таблицы booking, описывающей бронирование номеров клиентами, данная таблица похоже на предыдущую таблицу check\_in, но в таблице booking отсутствует атрибут цены номера, так как оплата производится только по факту, при заселении, информация данной таблицы нужна, чтобы в желаемый клиентом номер не заселились до него в день заезда.

# 3 ИСПОЛЬЗОВАНИЕ БАЗЫ ДАННЫХ

## 3.1 Создание DataSet

Для дальнейшей работы с базой данных был разработан Dataset, указанный в «Таблице 5 – dataset таблицы clients», «Таблице 6 – dataset таблицы rooms», «Таблице 7 – dataset таблицы categories», «Таблице 8 – dataset таблицы clients\_category», «Таблице 9 – dataset таблицы check\_in», «Таблице 10 – dataset таблицы booking».

Таблица 5 – dataset таблицы clients

| client\_code | lastname | firstname | middlename | passport | comment |
| --- | --- | --- | --- | --- | --- |
| 1 | Иванов | Иван | Иванович | 2417 802112 | Буйный клиент |
| 2 | Петров | Петр | Петрович | 1506 709956 |  |
| 3 | Сидоров | Андрей | Алексеевич | 0502 432189 | Клиент с животным |
| 4 | Смирнова | Мария | Викторовна | 1208 625478 |  |
| 5 | Козлов | Алексей | Дмитриевич | 3003 890754 | VIP клиент |
| 6 | Николаев | Владимир | Игоревич | 2107 416789 | Клиент с особыми требованиями |
| 7 | Кузнецова | Анна | Сергеевна | 0310 285416 |  |
| 8 | Васильев | Дмитрий | Александрович | 0905 674910 | Регулярный клиент |
| 9 | Соколова | Елена | Павловна | 2801 753218 |  |
| 10 | Михайлов | Сергей | Иванович | 0709 986532 | Клиент-путешественник |
| 11 | Морозов | Артем | Максимович | 1804 358970 | Клиент семейного отдыха |
| 12 | Ковалев | Игорь | Валерьевич | 0212 786523 |  |
| 13 | Павлова | Ольга | Анатольевна | 0411 632109 | Клиент с детьми |
| 14 | Алексеев | Александр | Александрович | 1109 915478 |  |
| 15 | Федорова | Наталья | Викторовна | 0703 109456 | Клиент с предпочтениями в питании |
| 16 | Григорьев | Андрей | Андреевич | 1602 247890 |  |
| 17 | Исаева | Анастасия | Олеговна | 2501 785412 |  |
| 18 | Тимофеев | Евгений | Игоревич | 0912 643587 | Клиент с индивидуальным графиком |
| 19 | Савельев | Илья | Сергеевич | 0304 197856 |  |
| 20 | Андреева | Екатерина | Дмитриевна | 1406 825039 |  |

Таблица 6 – dataset таблицы rooms

| room\_code | roomnumber | capacity | comfortlevel | price | busy |
| --- | --- | --- | --- | --- | --- |
| 1 | 101 | 2 | Люкс | 9530 | FALSE |
| 2 | 102 | 2 | Полулюкс | 6100 | FALSE |
| 3 | 103 | 1 | Обычный | 1730 | FALSE |
| 4 | 104 | 3 | Полулюкс | 3760 | FALSE |
| 5 | 105 | 2 | Полулюкс | 6820 | FALSE |
| 6 | 106 | 1 | Обычный | 2170 | FALSE |
| 7 | 107 | 2 | Полулюкс | 5180 | FALSE |
| 8 | 108 | 3 | Люкс | 8370 | FALSE |
| 9 | 109 | 2 | Полулюкс | 3210 | FALSE |
| 10 | 110 | 1 | Обычный | 1050 | FALSE |
| 11 | 201 | 2 | Полулюкс | 5640 | FALSE |
| 12 | 202 | 2 | Полулюкс | 4660 | FALSE |
| 13 | 203 | 1 | Обычный | 1490 | FALSE |
| 14 | 204 | 3 | Люкс | 7320 | FALSE |
| 15 | 205 | 2 | Полулюкс | 6630 | FALSE |
| 16 | 206 | 1 | Обычный | 2280 | FALSE |
| 17 | 207 | 2 | Полулюкс | 5870 | FALSE |
| 18 | 208 | 3 | Люкс | 9150 | FALSE |
| 19 | 209 | 2 | Полулюкс | 4230 | FALSE |
| 20 | 210 | 1 | Обычный | 1220 | FALSE |
| 21 | 301 | 2 | Люкс | 8950 | FALSE |
| 22 | 302 | 2 | Полулюкс | 5370 | FALSE |
| 23 | 303 | 1 | Обычный | 1890 | FALSE |
| 24 | 304 | 3 | Полулюкс | 4050 | FALSE |
| 25 | 305 | 2 | Люкс | 7210 | FALSE |
| 26 | 306 | 1 | Обычный | 1980 | FALSE |
| 27 | 307 | 2 | Полулюкс | 5410 | FALSE |
| 28 | 308 | 3 | Люкс | 7620 | FALSE |
| 29 | 309 | 2 | Полулюкс | 3830 | FALSE |
| 30 | 310 | 1 | Обычный | 1390 | FALSE |
| 31 | 401 | 2 | Полулюкс | 6540 | FALSE |
| 32 | 402 | 2 | Полулюкс | 4720 | FALSE |
| 33 | 403 | 1 | Обычный | 1640 | FALSE |
| 34 | 404 | 3 | Полулюкс | 6840 | FALSE |
| 35 | 405 | 2 | Полулюкс | 6180 | FALSE |
| 36 | 406 | 1 | Обычный | 2560 | FALSE |
| 37 | 407 | 2 | Полулюкс | 4970 | FALSE |
| 38 | 408 | 3 | Люкс | 8790 | FALSE |
| 39 | 409 | 2 | Полулюкс | 5570 | FALSE |
| 40 | 410 | 1 | Обычный | 1870 | FALSE |
| 41 | 501 | 2 | Люкс | 9320 | FALSE |
| 42 | 502 | 2 | Полулюкс | 5560 | FALSE |
| 43 | 503 | 1 | Обычный | 2010 | FALSE |
| 44 | 504 | 3 | Люкс | 7610 | FALSE |
| 45 | 505 | 2 | Полулюкс | 6950 | FALSE |
| 46 | 506 | 1 | Обычный | 2380 | FALSE |
| 47 | 507 | 2 | Полулюкс | 5020 | FALSE |
| 48 | 508 | 3 | Люкс | 8530 | FALSE |
| 49 | 509 | 2 | Полулюкс | 6190 | FALSE |
| 50 | 510 | 1 | Обычный | 1690 | FALSE |

Таблица 7 – dataset таблицы categories

| category\_code | categoryname | sale\_percent | description |
| --- | --- | --- | --- |
| 1 | VIP | 10 | Высокопоставленные чиновники и политики, знаменитости и известные личности, бизнес-элита и корпоративные клиенты |
| 2 | Студенты | 3 | Учащиеся очной формы обучения, на бюджетных местах в государственном вузе |
| 3 | Постоянные клиенты | 15 | Проживающие в гостинице более 5 раз |
| 4 | Пенсионеры | 15 | Клиенты старше 60 лет |
| 5 | МНГ | 7 | Маломобильные группы населения |

Таблица 8 - dataset таблицы clients\_category

| client\_code | category\_code |
| --- | --- |
| 1 | 3 |
| 1 | 2 |
| 3 | 5 |
| 20 | 2 |
| 5 | 4 |
| 6 | 5 |
| 6 | 2 |
| 8 | 3 |
| 17 | 4 |
| 10 | 1 |
| 11 | 5 |
| 12 | 3 |
| 12 | 1 |
| 14 | 4 |
| 15 | 2 |

Таблица 9 – dataset таблицы check\_in

| check\_in\_code | client\_code | room\_code | checkindate | checkoutdate | note | pricewithdiscount |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 5 | 3 | 6/1/2023 | 6/5/2023 | NULL | 1470.5 |
| 2 | 12 | 7 | 6/5/2023 | 6/7/2023 | NULL | 3885 |
| 3 | 9 | 2 | 6/10/2023 | 6/15/2023 | NULL | 6100 |
| 4 | 18 | 5 | 6/12/2023 | 6/14/2023 | NULL | 6820 |
| 5 | 2 | 1 | 6/18/2023 | 6/20/2023 | NULL | 9530 |
| 6 | 11 | 6 | 6/22/2023 | 6/25/2023 | NULL | 2018.1 |
| 7 | 8 | 4 | 6/25/2023 | 6/28/2023 | NULL | 3196 |
| 8 | 19 | 9 | 6/28/2023 | 6/30/2023 | NULL | 3210 |
| 9 | 15 | 8 | 6/30/2023 | 6/30/2023 | NULL | 8118.9 |
| 10 | 3 | 12 | 6/30/2023 | 7/1/2023 | NULL | 4333.8 |

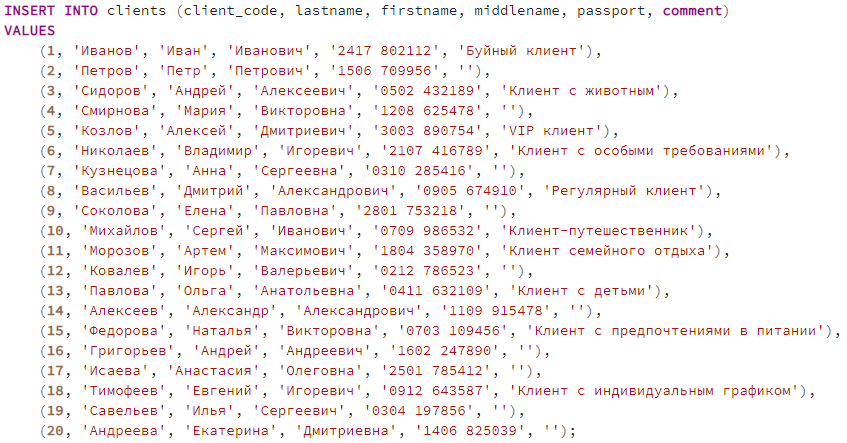
Таблица 10 – dataset таблицы booking

| booking\_code | client\_code | room\_code | checkindate | checkoutdate | note |
| --- | --- | --- | --- | --- | --- |
| 1 | 6 | 3 | 6/5/2023 | 6/10/2023 | NULL |
| 2 | 12 | 13 | 6/12/2023 | 6/15/2023 | NULL |
| 3 | 9 | 8 | 6/18/2023 | 6/22/2023 | NULL |
| 4 | 16 | 1 | 6/25/2023 | 6/28/2023 | NULL |
| 5 | 18 | 6 | 6/30/2023 | 7/2/2023 | NULL |

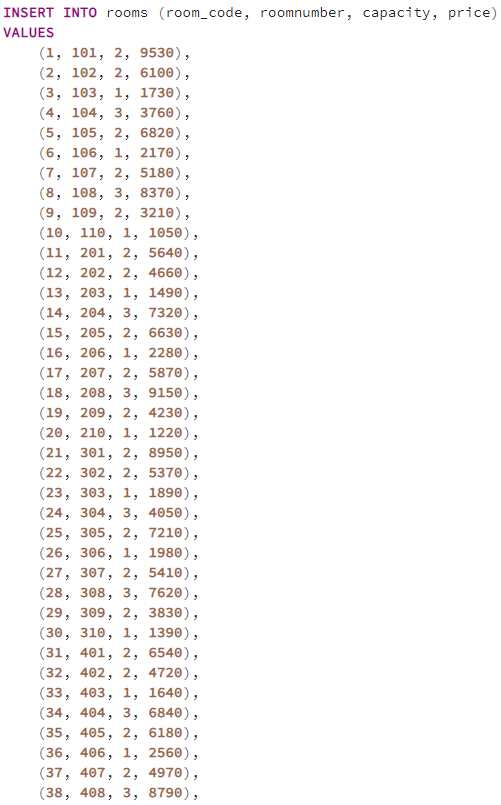
## 3.2 Добавление данных в базу

Разработанный dataset был добавлен в таблицу при помощи SQL-запросов, указанных в «Листинге 2 – внесение данных в систему»

Листинг 2 – внесение данных в систему

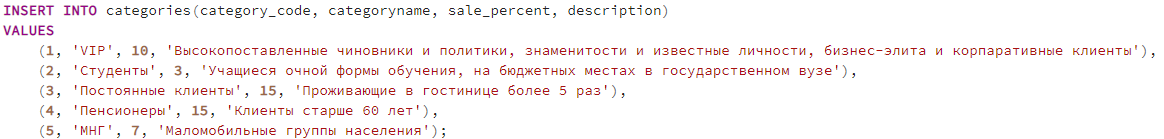


Добавление данных в таблицу clients.

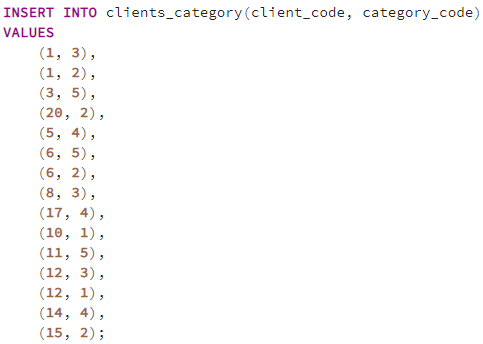




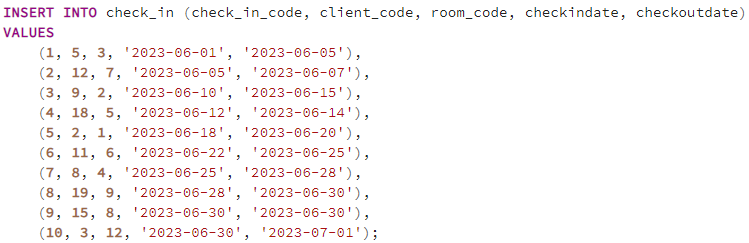
Добавление данных в таблицу rooms.



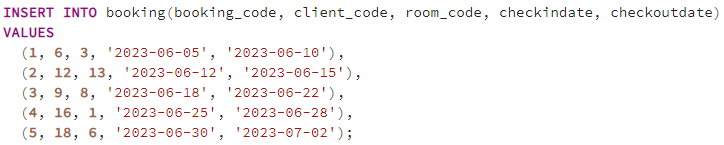
Добавление данных в таблицу categories.



Добавление данных в таблицу clients\_category.



Добавление данных в таблицу check\_in.

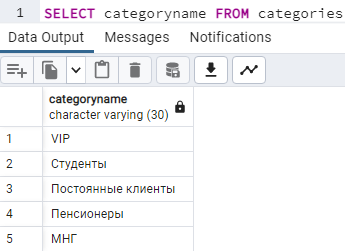


Добавление данных в таблицу booking.

## 3.3 Запросы к базе данных

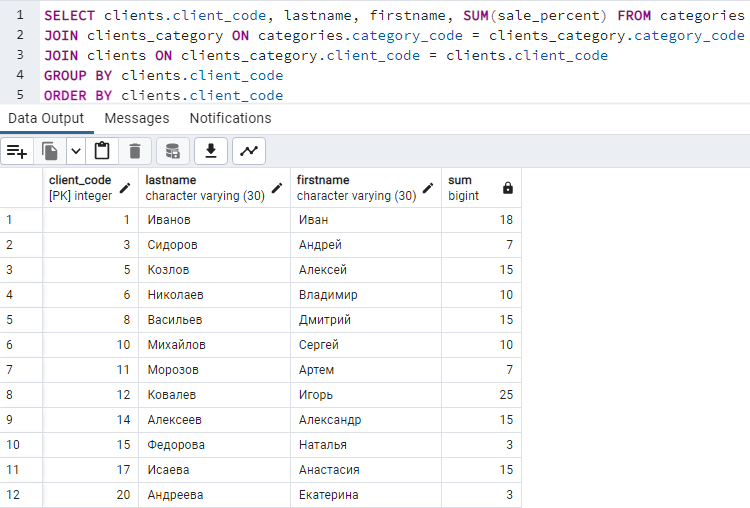
Для работы с базой данных были разработаны следующие запросы:

1. Вывод всех названий категорий. Код запроса указан в «Листинге 3 – запрос 1»;



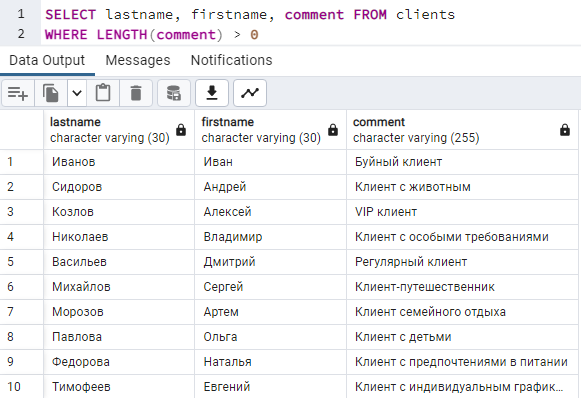
Листинг 3 – запрос 1

1. Вывод размера скидки, на которую может рассчитывать каждый клиент. Код запроса указан в «Листинге 4 – запрос 2»;



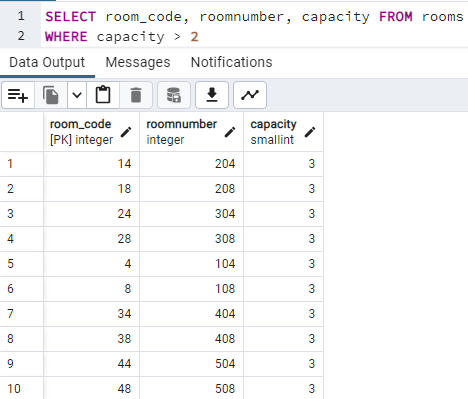
Листинг 4 – запрос 2

1. Вывод только тех клиентов, у которых есть заметка comment. Код запроса указан в «Листинге 5 – запрос 3»;



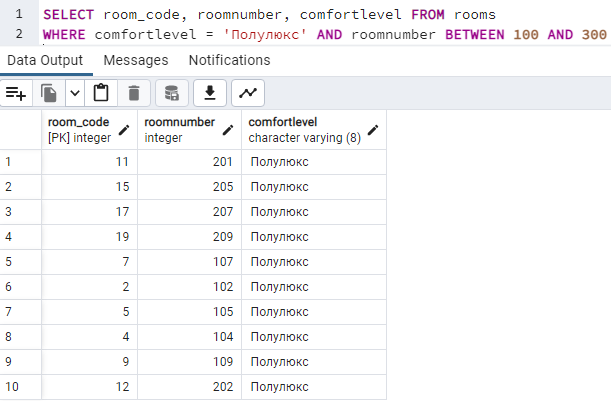
Листинг 5 – запрос 3

1. Вывод всех комнат, вместимость которых больше 2 человек. Код запроса указан в «Листинге 6 – запрос 4»;



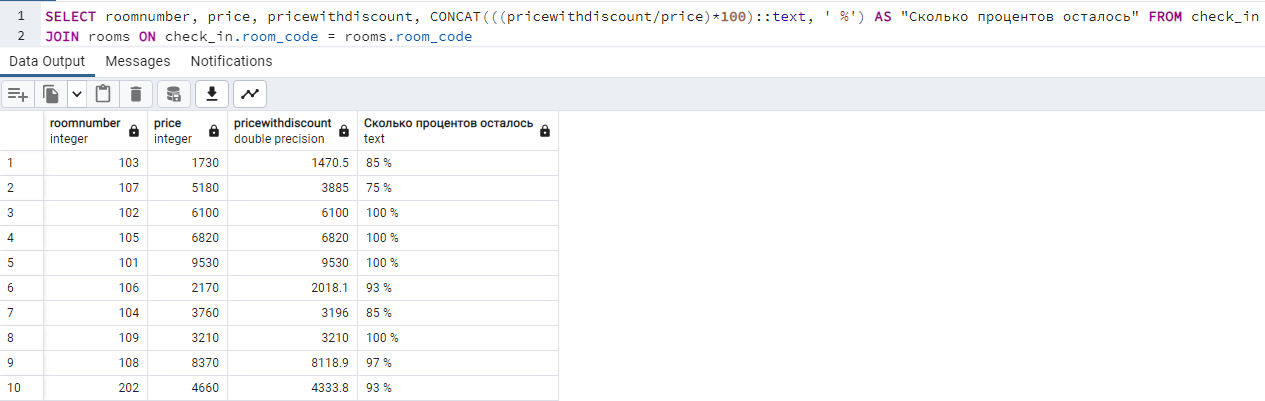
Листинг 6 – запрос 4

1. Вывод всех комнат комфорта полулюкс и с номером в пределах от 100 до 300. Код запроса указан в «Листинге 7 – запрос 5»;



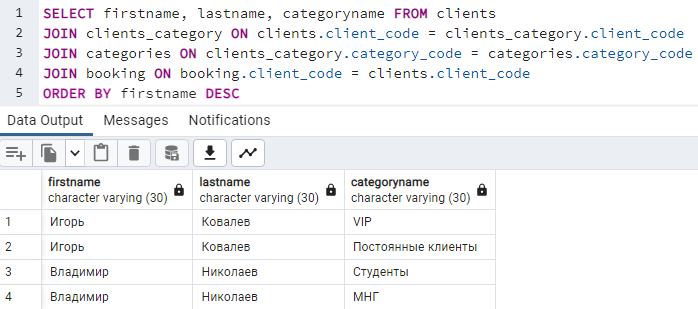
Листинг 7 – запрос 4

1. Вывод стоимости снятых номеров без скидки, со скидкой, и сколько цена со скидкой составляет от цены без скидки. Код запроса указан в «Листинге 8 – запрос 6»;



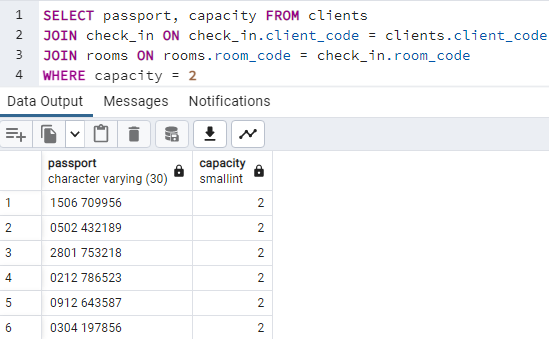
Листинг 8 – запрос 6

1. Вывод категорий клиентов, забронировавших номера, отсортировать их в обратном алфавитном порядке по Имени. Код запроса указан в «Листинге 9 – запрос 7»;



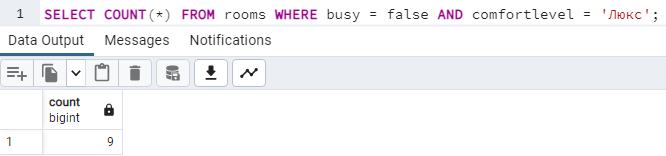
Листинг 9 – запрос 7

1. Вывод паспортных данных клиентов, заселявшихся в двухместные номера. Код запроса указан в «Листинге 10 – запрос 8»;



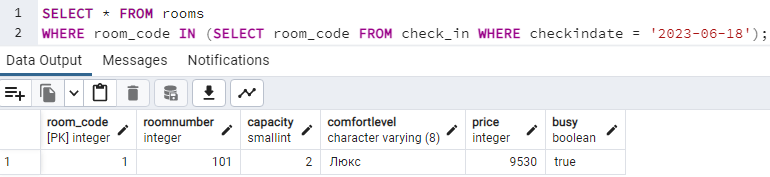
Листинг 10 – запрос 8

1. Вывод количества свободных комнат определенного уровня комфорта. Код запроса указан в «Листинге 11 – запрос 9»;



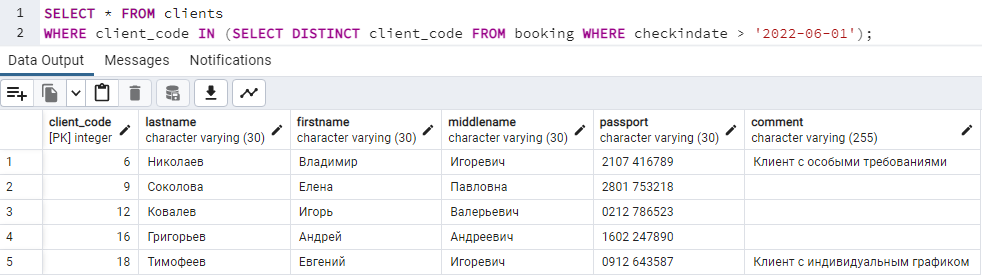
Листинг 11 – запрос 9

1. Вывод список всех комнат заселенных в определенный день. Код запроса указан в «Листинге 12 – запрос 10»;



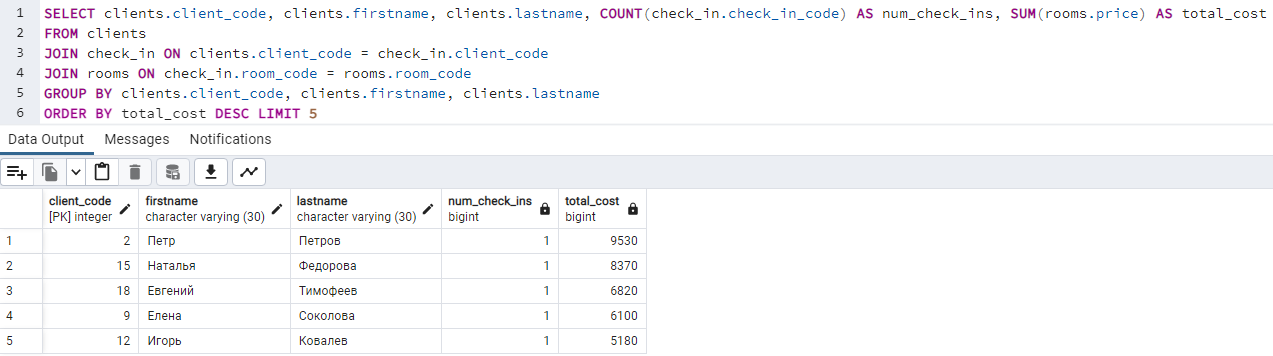
Листинг 12 – запрос 10

1. Вывод клиентов имеющих бронь с датой заезда позже определенной даты. Код запроса указан в «Листинге 13 – запрос 11»;



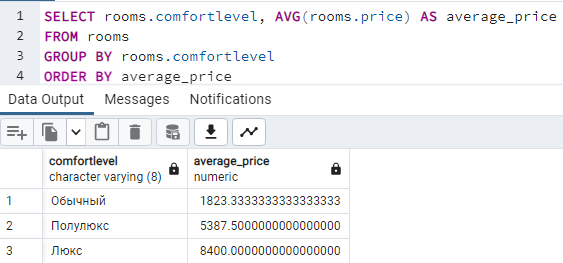
Листинг 13 – запрос 11

1. Вывод клиентов и количество их заселений в отель, вывод должен показать 5 клиентов, которые потратили больше всего денег. Код запроса указан в «Листинге 14 – запрос 12»;



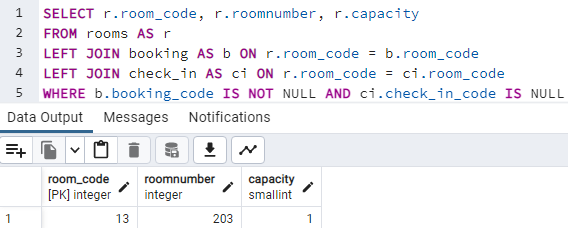
Листинг 14 – запрос 12

1. Вывод средней цены номеров каждой категории в порядке возрастания цены. Код запроса указан в «Листинге 15 – запрос 13»;



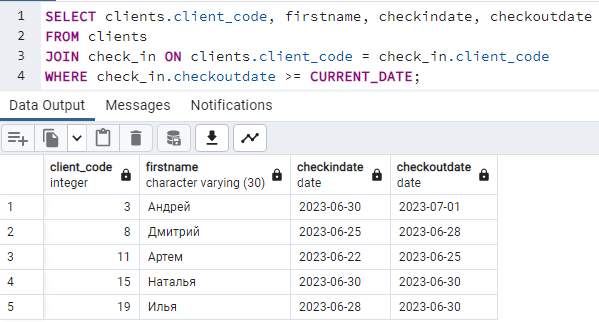
Листинг 15 – запрос 13

1. Вывод списка комнат, которые забронированы, но еще не заселены. Код запроса указан в «Листинге 16 – запрос 14»;

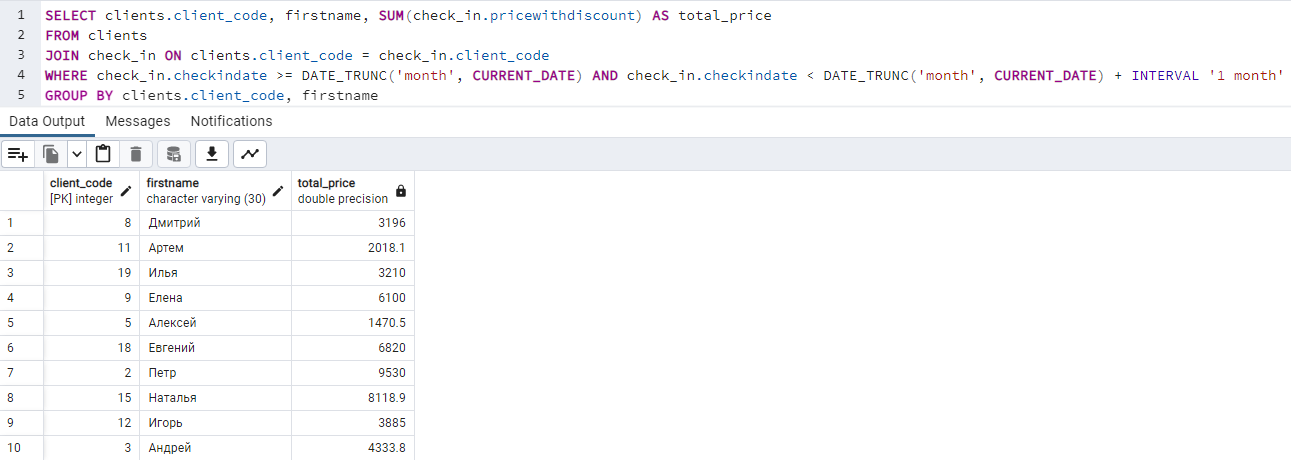


Листинг 16 – запрос 14

1. Вывод клиентов которые заселились, но еще не выселились. Код запроса указан в «Листинге 17 – запрос 15»;

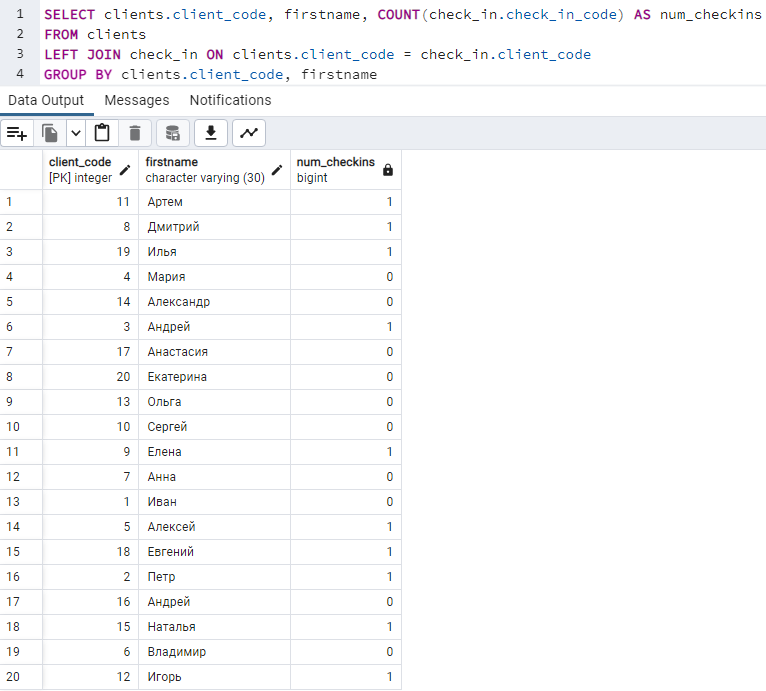


1. Вывод список клиентов и сумму которую они заплатили за последний месяц. Код запроса указан в «Листинге 18 – запрос 16»;



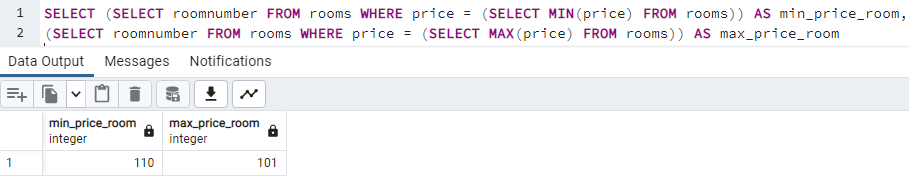
Листинг 18 – запрос 16

1. Вывод всех клиентов и количество их заселений в отель. Код запроса указан в «Листинге 19 – запрос 17»;



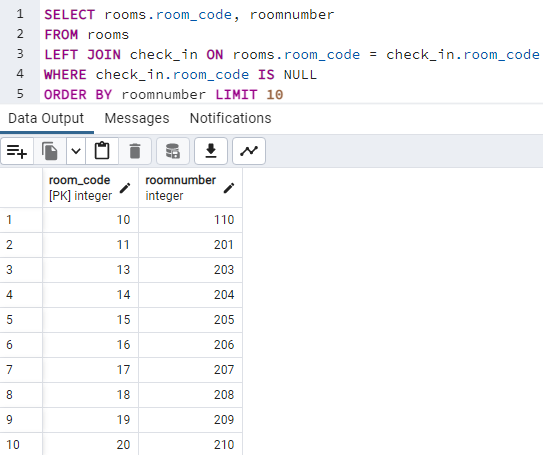
Листинг 19 – запрос 17

1. Вывод номеров комнат с самой дешевой и самой дорогой ценой. Код запроса указан в «Листинге 20 – запрос 18»;



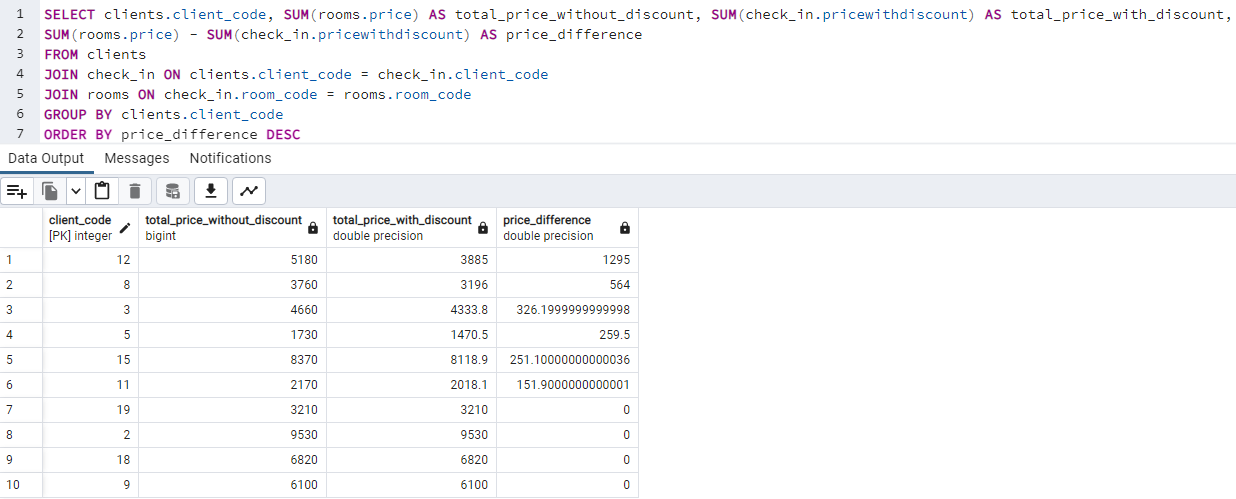
Листинг 20 – запрос 18

1. Вывод первых 10 комнат в порядке возрастания номера комнаты, которые не были еще никогда заселены. Код запроса указан в «Листинге 21 – запрос 19»;



Листинг 21 – запрос 19

1. Вывод количества сэкономленных на скидках денег в порядке убывания. Код запроса указан в «Листинге 22 – запрос 20»;

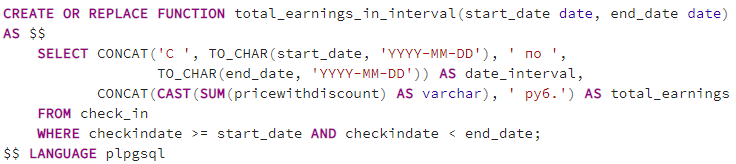


Листинг 22 – запрос 20

## 3.4 Тестирование функций и триггеров

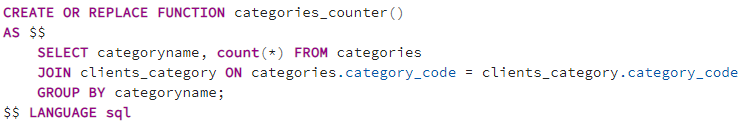
Для работы с базой данных были разработаны следующие функции и триггеры:

1. функция total\_earnings\_in\_interval(start\_date date, end\_date date), которая используется для того, чтобы посмотреть сколько заработала гостиница за определенный промежуток времени. Код функции и ее инициализация указаны в «Листинге 23 – функция total\_earnings\_in\_interval»;



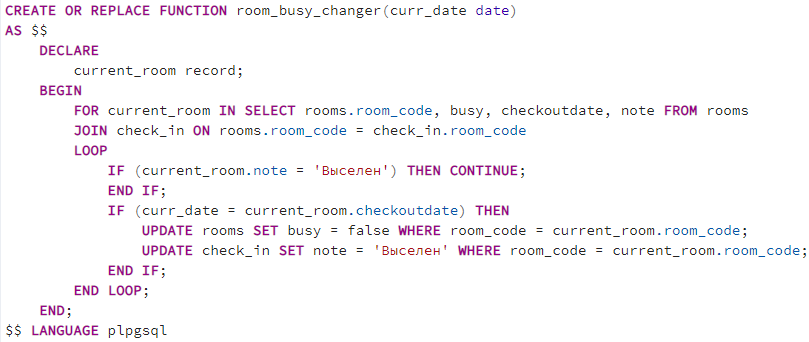
Листинг 23 – функция total\_earnings\_in\_interval

1. функция categories\_counter(), которая выводит количество клиентов в каждой категории. Код функции и ее инициализация указаны в «Листинге 24 – функция categories\_counter»;



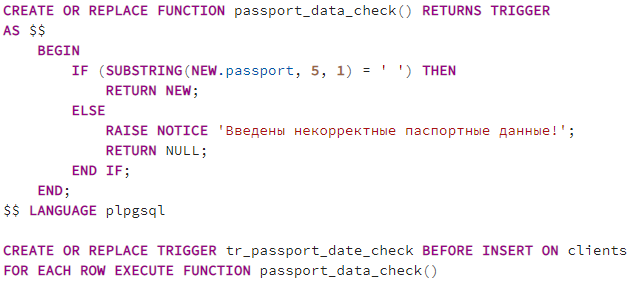
Листинг 24 – функция categories\_counter

1. функция room\_busy\_changer(curr\_date date), которая меняет статус занятости комнаты в день выселения. Данная функция должна запускаться во время выселения клиентов из номеров, например в 10:00 каждого дня, функция будет менять статус номера, которая должна быть освобождена сегодня, на свободная, при этом ставит заметку в таблицу заселений, что клиенты из выселены, чтобы не обращаться лишний раз к этой строке при проверки номеров. Код функции и ее инициализация указаны в «Листинге 25 – функция room\_busy\_changer»;



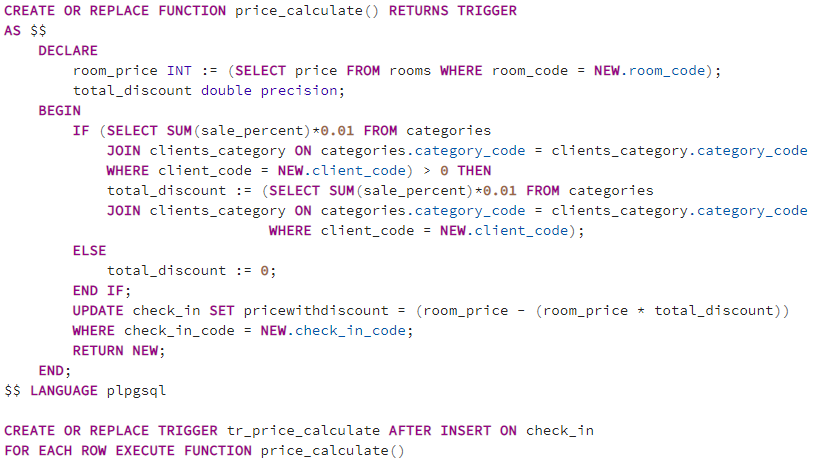
Листинг 25 – функция room\_busy\_changer

1. триггер passport\_data\_check, который проверяет корректность введенных паспортных данных при добавлении нового пользователя. Код функции и ее инициализация указаны в «Листинге 26 – триггер passport\_data\_check»;



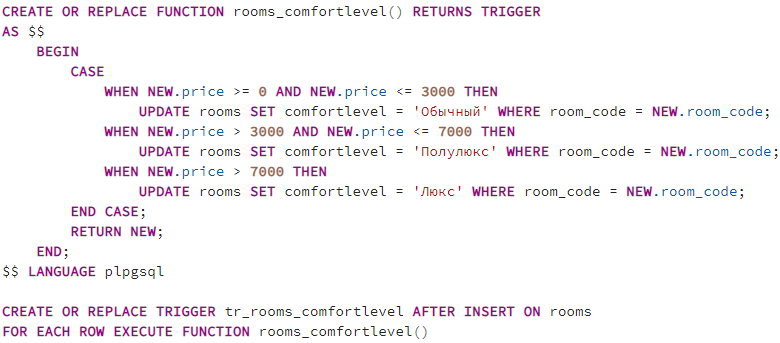
Листинг 26 – функция passport\_data\_check

1. триггер price\_calculate, который автоматически считает стоимость при заселении с учетом всех скидок клиента. Код триггера и его инициализация указаны в «Листинге 27 – триггер price\_calculate»;



Листинг 27 – триггер price\_calculate

1. триггер rooms\_comfortlevel, который автоматически определяет уровень комфорта номера исходя из его цены. Код триггера и его инициализация указаны в «Листинге 28 – триггер rooms\_comfortlevel»;



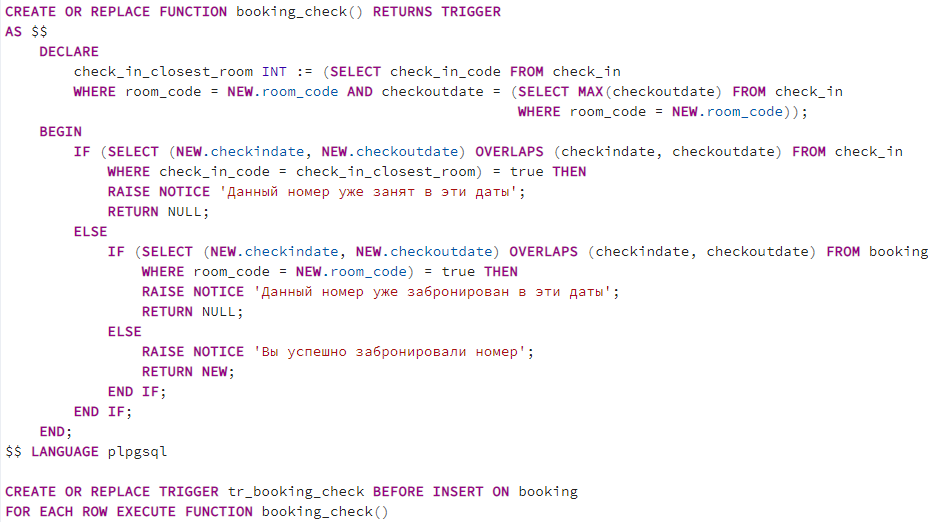
Листинг 28 – триггер rooms\_comfortlevel

1. триггер checking\_in, который проверяет наличие номера при заселении. Триггер выдаст исключения в следующих случаях: желаемый номер уже занят, желаемый номер забронирован в желаемые даты. Если даты заселения, выезда, код клиента и код комнаты совпадают с этими же атрибутами в таблице брони, то данные удаляются из таблицы бронирования и добавляются в таблицу заселения, статус номера меняется на занятый, так как пользователь только что заселился по этой брони. Во всех остальных случаях триггер заселит клиента, а статус комнаты заменит на занятую. Код триггера и его инициализация указаны в «Листинге 29 – триггер checking\_in»;



Листинг 29 – триггер checking\_in

1. триггер booking\_check, который проверяет не занят ли номер при бронировании. Триггер выдаст исключения в следующих случая: желаемый номер уже забронирован в желаемые даты, желаемый номер уже заселен в желаемые даны, во всех остальных случаях триггер добавит запись о бронировании. Код триггера и его инициализация указаны в «Листинге 30 – триггер booking\_check»;



Листинг 30 – триггер booking\_check

Далее было проведено тестирование функций на новых наборах данных, был описан ожидаемый результат и фактический результат и сделан вывод о работоспособности функции.

1. функция функция total\_earnings\_in\_interval(start\_date date, end\_date date).

Входные данные: «2023-06-01», «2023-07-01»

Ожидаемый результат: 48682.3 (именно столько было потрачено за время входных данных в нашем data set).

Фактический результат представлен на «Рисунке 4 – тестирование функции total\_earnings\_in\_interval».

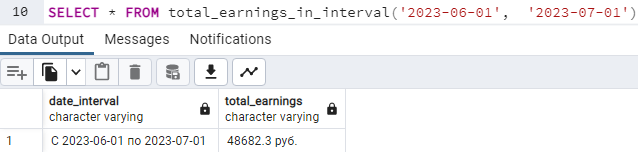


Рисунок 4 – тестирование функции total\_earnings\_in\_interval

Вывод о работе функции: функция работает исправно;

1. функция categories\_counter().

Входные данные: отсутствуют

Ожидаемый результат: Пенсионеры - 3, Студенты - 4, Постоянные клиенты - 3, МНГ - 3, VIP - 2.

Результат тестирования приведен на «Рисунке 5 – тестирование categories\_counter».

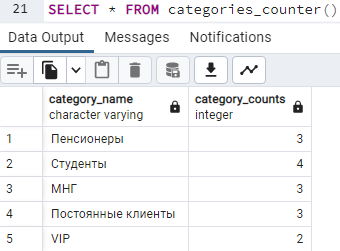


Рисунок 5 – тестирование categories\_counter

Вывод о работе функции: функция работает исправно.

1. Функция room\_busy\_changer(curr\_date date).

Входные данные: «2023-06-05»

Ожидаемый результат: выходные данные: номера, дата выселения которых совпадает с входными данными сменят статус на свободные, добавится заметка «Выселен».

Фактический результат приведен в «Рисунке 8 – тестирование room\_busy\_changer 1», «Рисунке 9 – тестирование room\_busy\_changer 2», «Рисунке 10 – тестирование room\_busy\_changer 3»

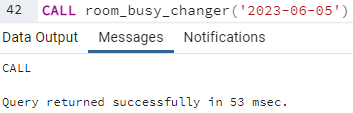


Рисунок 8 – тестирование room\_busy\_changer 1

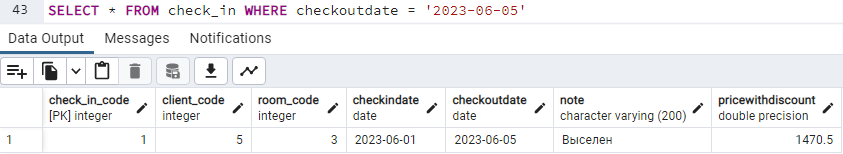


Рисунок 9 – тестирование room\_busy\_changer 2

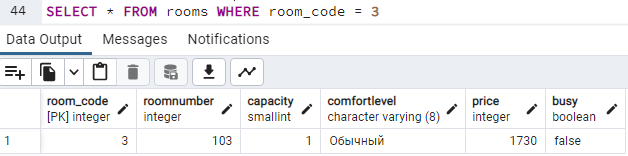


Рисунок 10 – тестирование room\_busy\_changer 3

Вывод о работе функции: функция работает исправно;

1. триггер passport\_data\_check.

Для проверки работы триггера необходимо добавить нового клиента с корректно введенными паспортными данными некорректно.

Входные данные: создание нового пользователя с полями 21, «Баденов», «Кирилл», «Дмитриевич», «2323 23332».

Ожидаемый результат: создание новой записи в таблице clients под client\_code = 21.

Фактический результат приведен в «Рисунке 11 – тестирование passport\_data\_check».

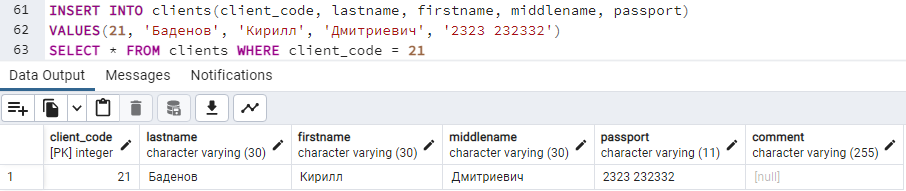


Рисунок 11 – тестирование passport\_data\_check

Входные данные: создание нового пользователя с полями 21, «Баденов», «Кирилл», «Дмитриевич», «232323332».

Ожидаемый результат: вывод сообщения «Введены некорректные паспортные данные!».

Фактический результат приведен в «Рисунке 12 – тестирование passport\_data\_check».

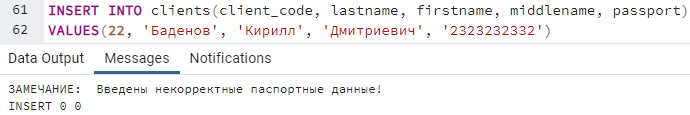


Рисунок 12 – тестирование passport\_data\_check

Вывод о работе триггера: триггер работает корректно;

1. триггер price\_calculate.

Для проверки работы триггера следует создать новую запись в таблице check\_in для клиента имеющего скидки, чтобы проверить будет ли цена посчитана с учетом скидки и клиента без скидки, чтобы проверить корректно ли цена будет указана.

Выходные данные: создание новой записи в check\_in с полями 11, 1, 15, «2023-06-23», «2023-06-25»

Ожидаемый результат: процент скидки клиента - 18, цена .комнаты с кодом 15 - 6630, значит цена с учетом скидки будет 5436,6

Фактический результат приведен в «Рисунке 13 – тестирование price\_calculate».

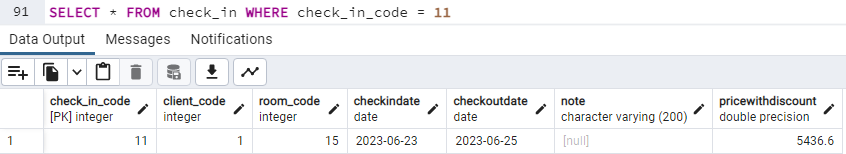


Рисунок 13 – тестирование price\_calculate

Выходные данные: создание новой записи в check\_in с полями 11, 2, 15, «2023-06-03», «2023-06-05»

Ожидаемый результат: процент скидки клиента - 0, цена .комнаты с кодом 15 - 6630, значит цена с учетом скидки будет 6630

Фактический результат приведен в «Рисунке 14 – тестирование price\_calculate».

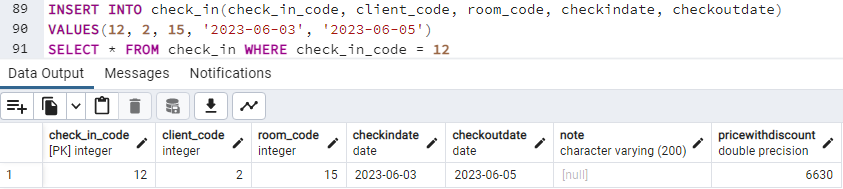


Рисунок 14 – тестирование price\_calculate

Вывод о работе триггера: триггер успешно выполнил свою работу;

1. триггер rooms\_comfortlevel

Входные данные: добавление нового номера с ценой 2500.

Ожидаемый результат: атрибут comfortlevel добавленного номера будет «Обычный».

Фактический результат приведен в таблице 15 – тестирование rooms\_comfortlevel.

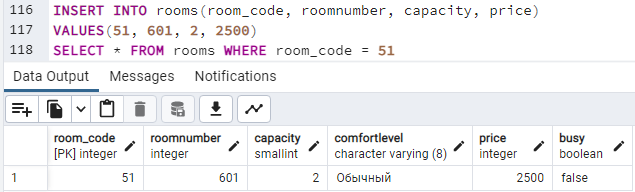


Рисунок 15 – тестирование rooms\_comfortlevel

Вывод о работе триггера: триггер работает;

1. триггер checking\_in.

Входные данные: создание нового заселения в уже занятый номер с полями 11, 19, 15, «2023-06-03», «2023-06-05».

Ожидаемый результат: «Данный номер уже занят».

Фактический результат приведен в «Рисунке 16 – тестирование checking\_in».

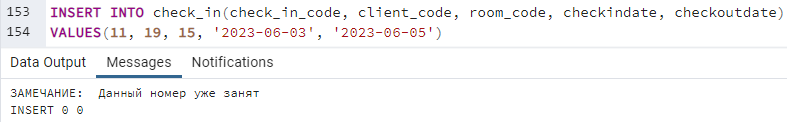


Рисунок 16 – тестирование checking\_in

Входные данные: создание нового заселения в уже занятый номер с полями 13, 6, 3, «2023-06-05», «2023-06-10».

Ожидаемый результат: вывод сообщения «Вы успешно заселились, данные о вашем бронировании удалены» и удаление соответствующих данных из таблицы бронирования, а так же добавление данных о заселении.

Фактический результат приведен в «Рисунке 17 – тестирование checking\_in 1», «Рисунке 18 – тестирование checking\_in 2», Рисунке 19 – тестирование checking\_in 3»

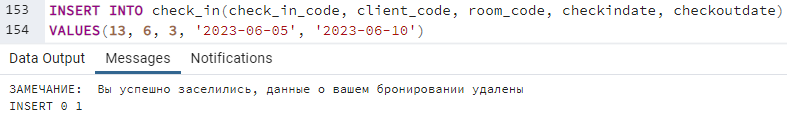


Рисунок 17 – тестирование checking\_in 1

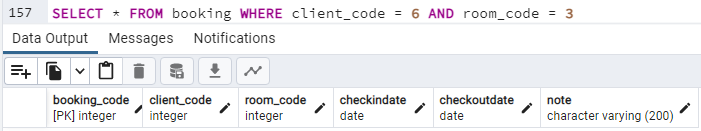


Рисунок 18 – тестирование checking\_in 2

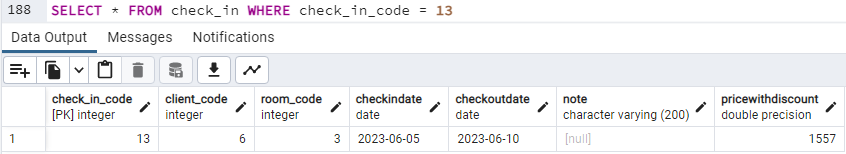


Рисунок 19 – тестирование checking\_in 3

Входные данные: создание нового заселения в свободный номер с полями 14, 6, 20, «2023-06-05», «2023-06-10».

Ожидаемый результат: добавление введенных данных в таблицу заселения.

Фактический результат приведен в «Рисунке 19 – тестирование checking\_in»

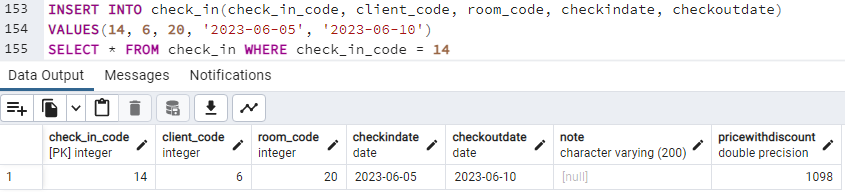


Рисунок 19 – тестирование checking\_in

Вывод о работе триггера: триггер работает корректно;

1. триггер booking\_check.

Для проверки работы триггера, нужно попытаться забронировать уже заселенный номер, забронировать забронированный номер и забронировать свободный номер.

Входные данные: создание новой записи в таблице booking с полями 11, 12, 3, «2023-06-02», «2023-06-06» о номере, который уже заселен в эти даты.

Ожидаемый результат: вывод сообщения «Данный номер уже занят в эти даты».

Фактический результат приведен в «Рисунке 20 – тестирование booking\_check».

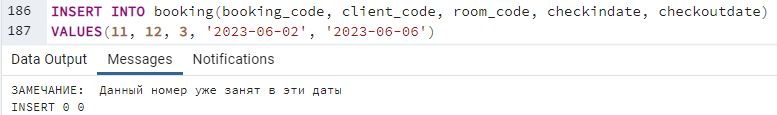


Рисунок 20 – тестирование booking\_check

Входные данные: создание новой записи в таблице booking с полями 12, 12, 8, «2023-06-10», «2023-06-13» о номере, который уже забронирован в эти даты.

Ожидаемый результат: вывод сообщения «Данный номер уже забронирован в эти даты».

Фактический результат приведен в «Рисунке 20 – тестирование booking\_check».

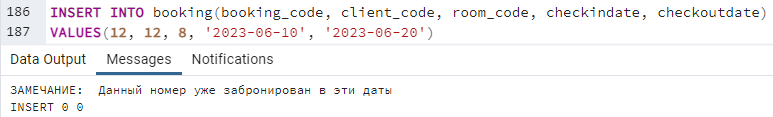


Рисунок 21 – тестирование booking\_check

Входные данные: создание новой записи в таблице booking с полями 14, 9, 20, «2023-06-10», «2023-06-20» о свободном номере.

Ожидаемый результат: вывод сообщения «Вы успешно забронировали номер» и добавление информации о бронировании в таблицу бронирования.

Фактический результат приведен в «Рисунке 22 – тестирование booking\_check 1», «Рисунке 23 – тестирование booking\_check 2».

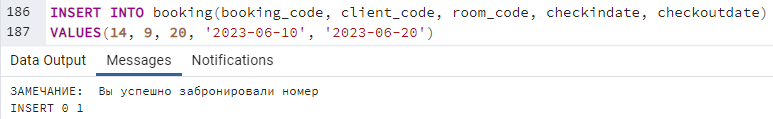


Рисунок 22 – тестирование booking\_check 1

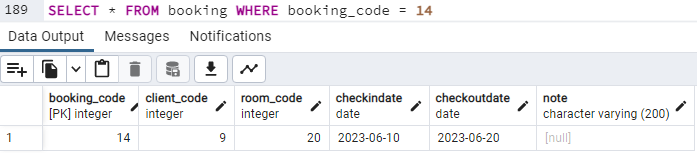


Рисунок 23 – тестирование booking\_check 2

Вывод о работе триггера: триггер работает корректно.

# ЗАКЛЮЧЕНИЕ

В процессе выполнения данной работы была разработана база данных с использованием системы управления базами данных PostgreSQL. Этот проект позволил мне получить ценный опыт в проектировании и разработке баз данных, а также в работе с SQL-запросами и PostgreSQL.

В процессе работы я изучил основные принципы моделирования данных, определил структуру базы данных, создал таблицы, установил связи между ними и заполнил их данными. Я также изучил различные возможности PostgreSQL, включая создание и использование функций, триггеров и процедур.

Создание данной базы данных помогло мне лучше понять важность эффективной организации и хранения данных для успешного функционирования информационных систем. Я осознал значение грамотного проектирования баз данных для обеспечения эффективности и масштабируемости системы, а также для обеспечения целостности и безопасности данных.

В процессе работы с PostgreSQL я столкнулся с некоторыми сложностями и проблемами, но благодаря тщательному изучению документации, я смог успешно их преодолеть и приобрести ценный опыт.

В целом, выполнение данного проекта позволило мне углубить знания и навыки в области баз данных и особенностей PostgreSQL. Я получил опыт в проектировании, разработке и оптимизации баз данных, который будет полезен для моей дальнейшей карьеры в сфере информационных технологий. PostgreSQL является мощной и гибкой системой управления базами данных, и я уверен, что приобретенные мной знания и навыки найдут применение в будущих проектах для создания надежных и эффективных баз данных.

# список использованных источников

1. ГОСТ 19.201-78. Режим доступа: https://www.swrit.ru/doc/espd/19.201-78.pdf (дата обращения: 18.04.2023).

2. Курс в ЛМС Московского политеха «Разработка технических текстов и документации». Режим доступа: https://online.mospolytech.ru/course/view.php?id=1689 (дата обращения: 18.04.2023).

3. Официальная документация разработчика PostgreSQL. Режим доступа: <https://www.postgresql.org/> (дата обращения: 29.05.2023).

4. Статья «Изучаем PostgreSQL. Часть 1. Знакомимся с архитектурой». Режим доступа: <https://habr.com/ru/companies/otus/articles/706346/> (дата обращения: 29.05.2023).

5. Информационный портал «Habr». Режим доступа: <https://habr.com/ru/all/> (дата обращения: 29.05.2023).

6 ГОСТ 7.32-2017. Режим доступа: <https://files.stroyinf.ru/Data2/1/4293742/4293742537.pdf> (дата обращения: 29.05.2023).

7. «PostgreSQL изнутри» — Е. Рогов. Режим доступа: <https://postgrespro.ru/education/books/internals> (дата обращения 29.05.2023).

8. «Postgres: первое знакомство» — П. Лузанов, Е. Рогов, И. Лёвшин. Режим доступа: <https://postgrespro.ru/education/books/introbook> (дата обращения 29.05.2023).

9. «PostgreSQL. Основы языка SQL» — Е. Моргунов. Режим доступа: <https://postgrespro.ru/education/books/sqlprimer> (дата обращения 29.05.2023).

10. Руководство по подготовке курсовых работ (проектов) и выпускных квалификационных работ. Режим доступа: <http://www.skf-mtusi.ru/files/vkr/MU_DP_KP_MTUSI.pdf> (дата обращения 29.05.2023)

11. Правила оформления курсовых и дипломных работ. Режим доступа: <https://studfile.net/preview/3009968/page:5/> (дата обращения 29.05.2023).

12. Форум по PosgreSQL Wiki.postgresql. Режим доступа: <https://wiki.postgresql.org/wiki/Main_Page/ru> (дата обращения 29.05.2023).