**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИИ**

**ФГБОУ ВПО "Пензенский государственный университет"**

**Кафедра "Математическое обеспечение и применение ЭВМ"**

Отчет

по лабораторной работе №2

**«Заполнение базы данных»**

Выполнили:

Хамзин Х.

Суворов М.

Хусаинов И.

ст. гр. 17ВИ1

Проверил:

доцент Казакова И.А.

**2020**

**Лабораторная работа №2**

Диаграмма базы данных гостиничного комплекса

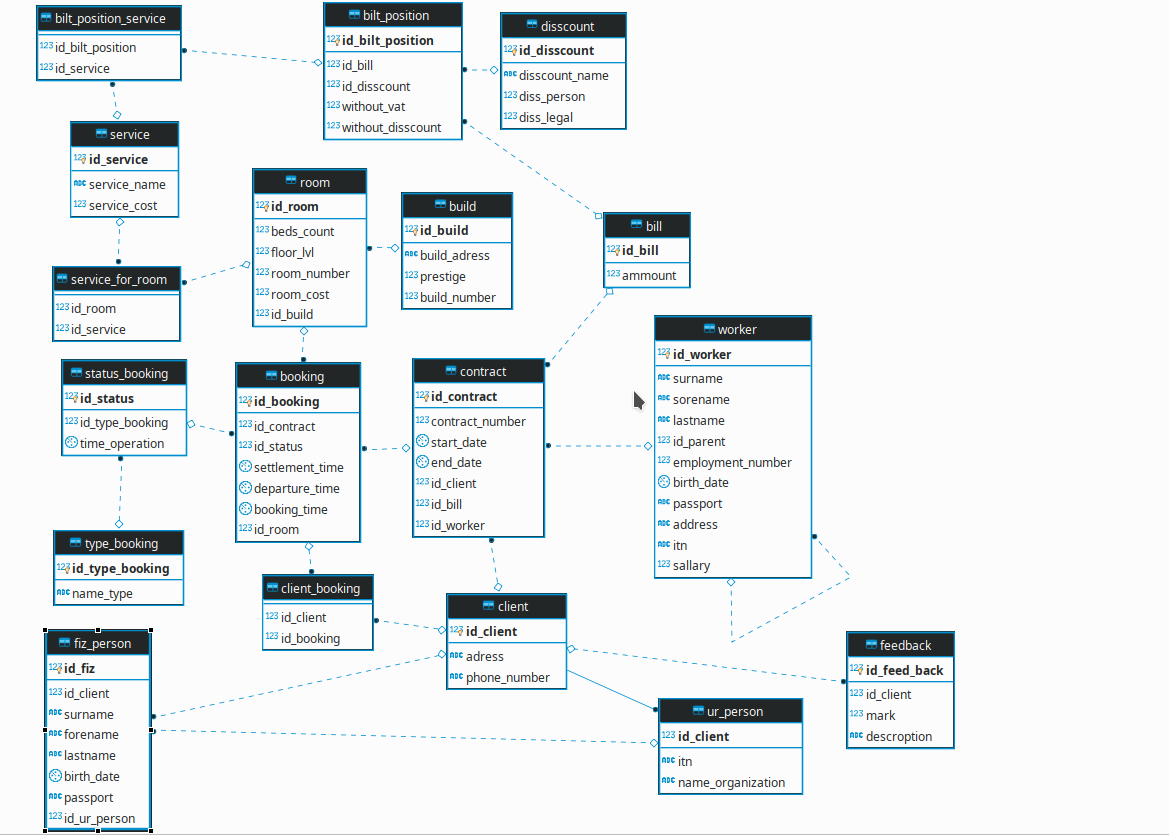
****

Таблица 1 - Сотрудник

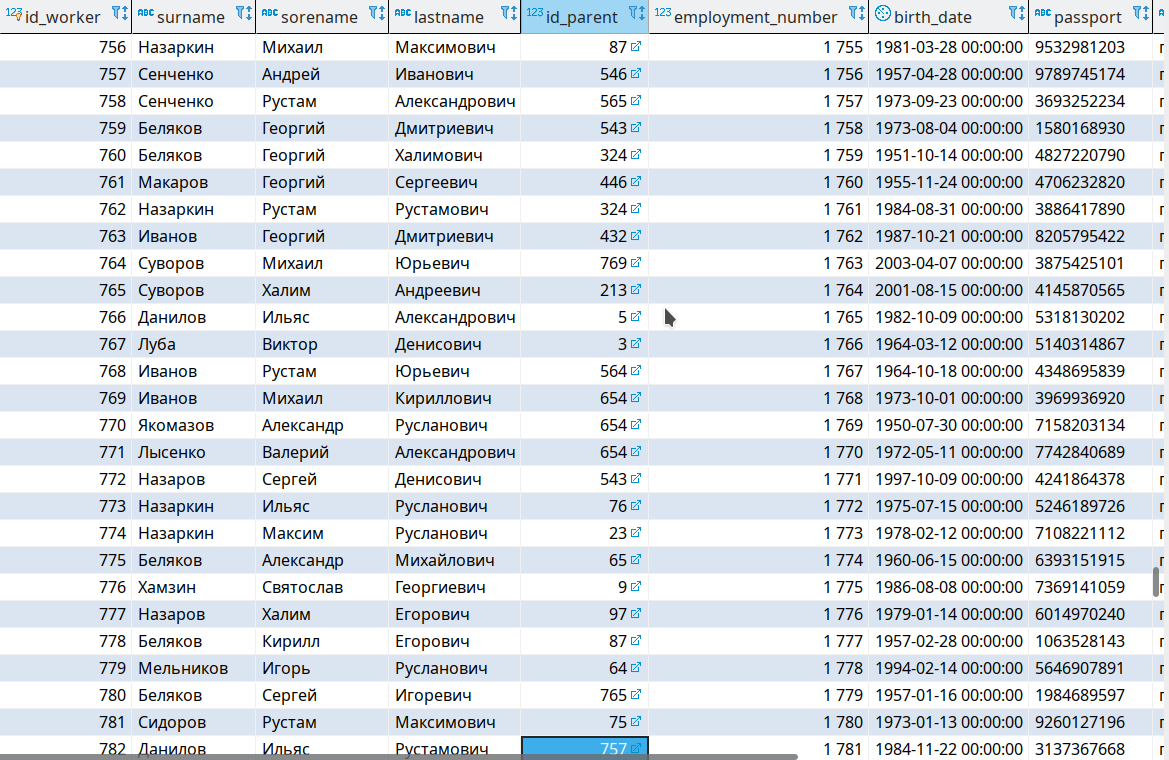


Таблица 2 — Здание

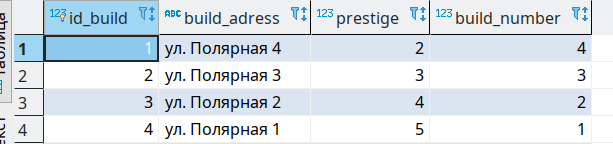


Таблица 3- Номер

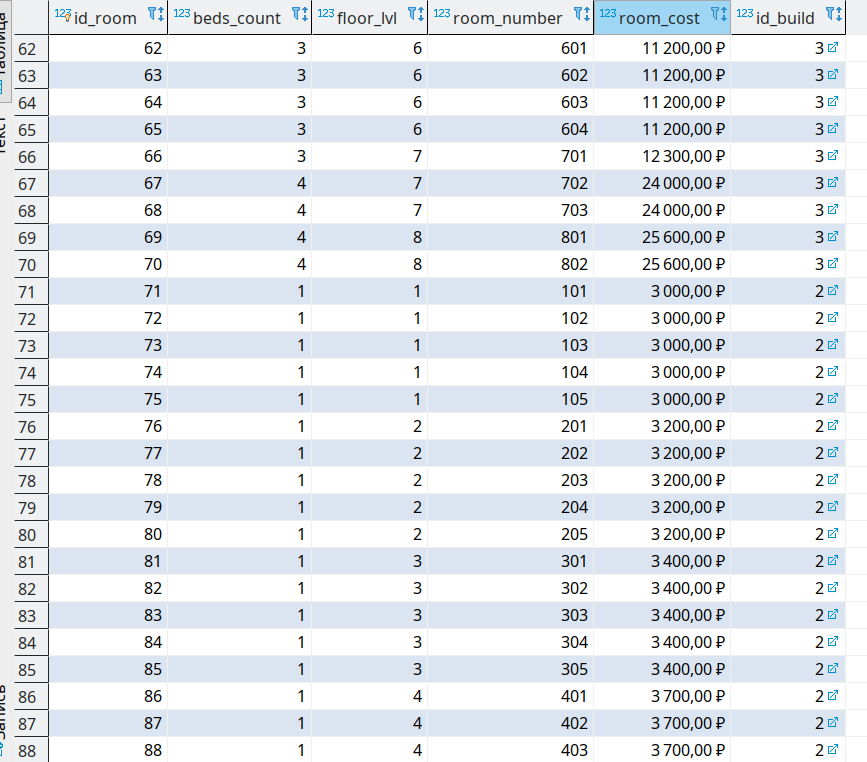


Таблица 4 — Услуга

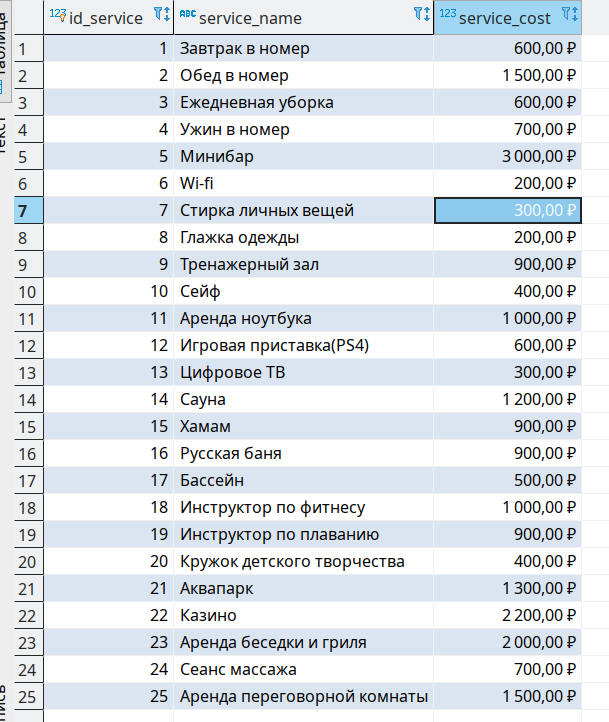


Таблица 5- Услуга-номер(Промежуточная таблица). Какие услуги уже включены в номер

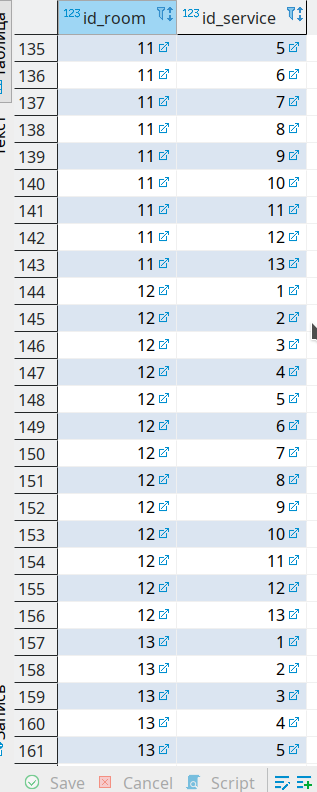


Таблица 6 — Скидка

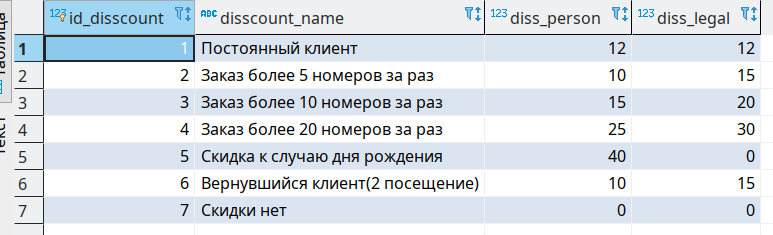


Таблица 7 — тип статуса бронирования

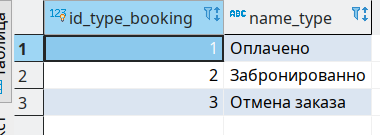


Таблица 8 — Счет

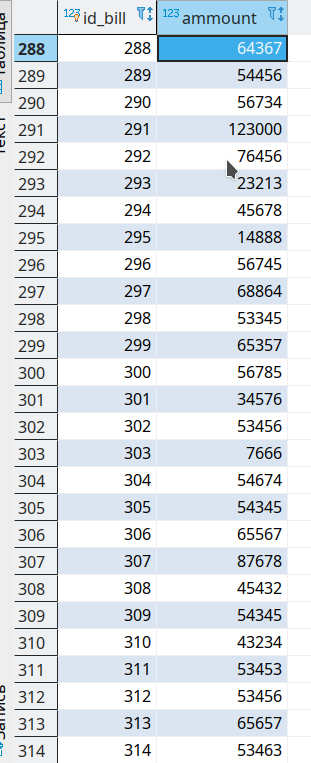


Таблица 9 — Позиция счета

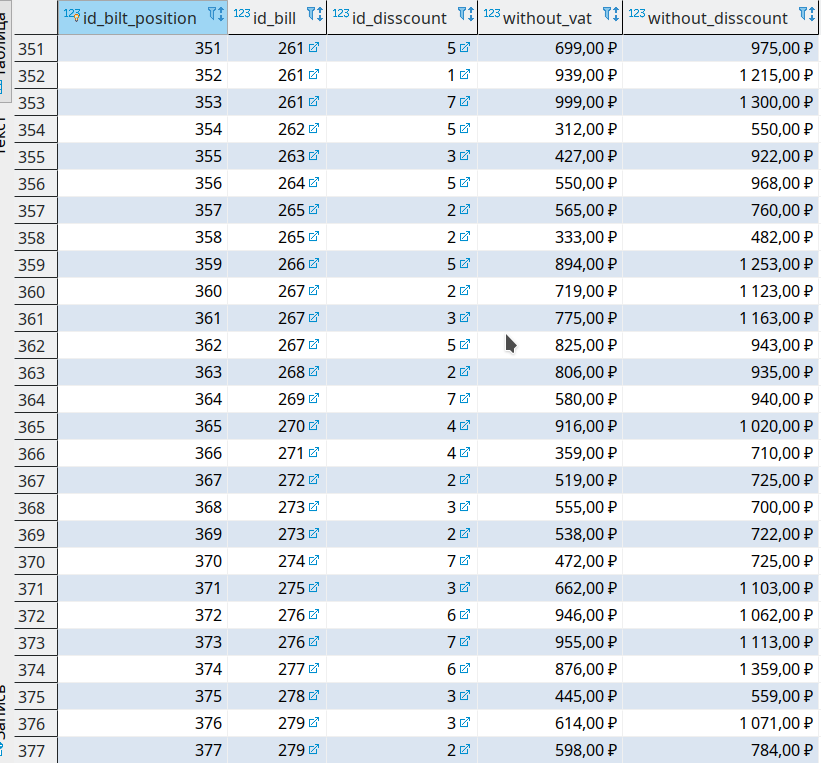


Таблица 10 — Позиция счета — услуга(Промежуточная таблица).

Услуги приобретенные во время нахождения в гостинице

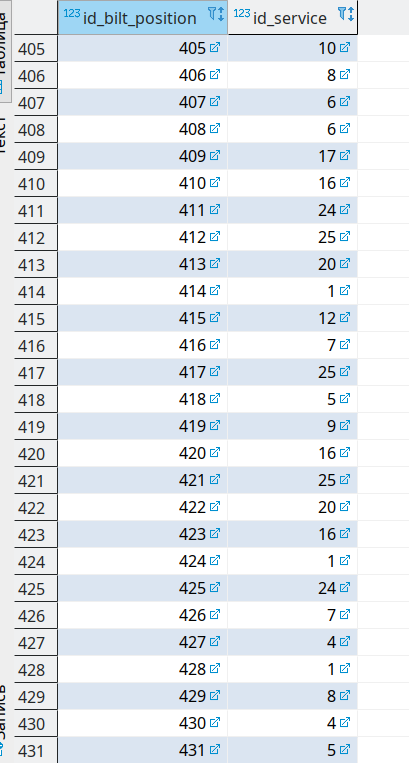


Таблица 11 — Бронирование

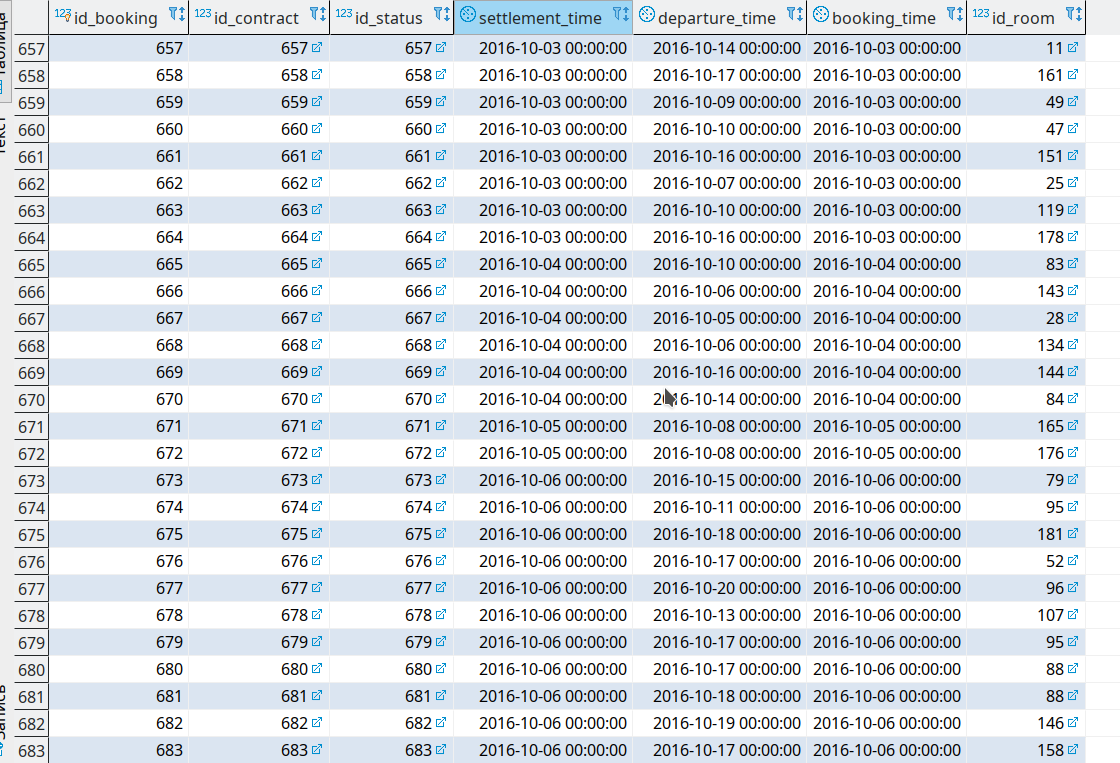


Таблица 12 — Клиент

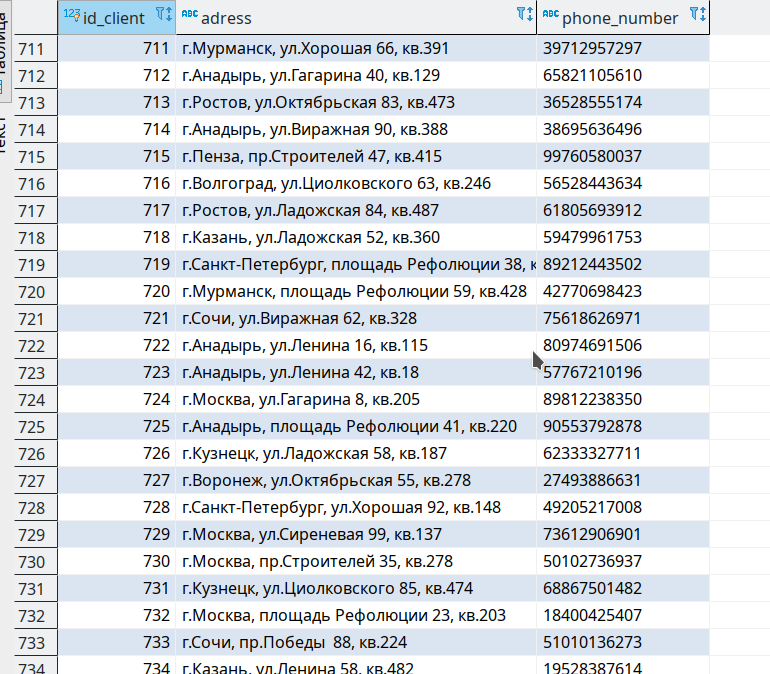


Таблица 13 — Клиент — бронирование(Промежуточная таблица). В каком номере находится каждый клиент

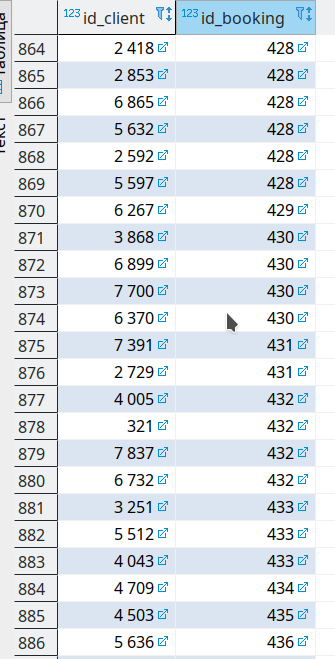


Таблица 14 - Договор

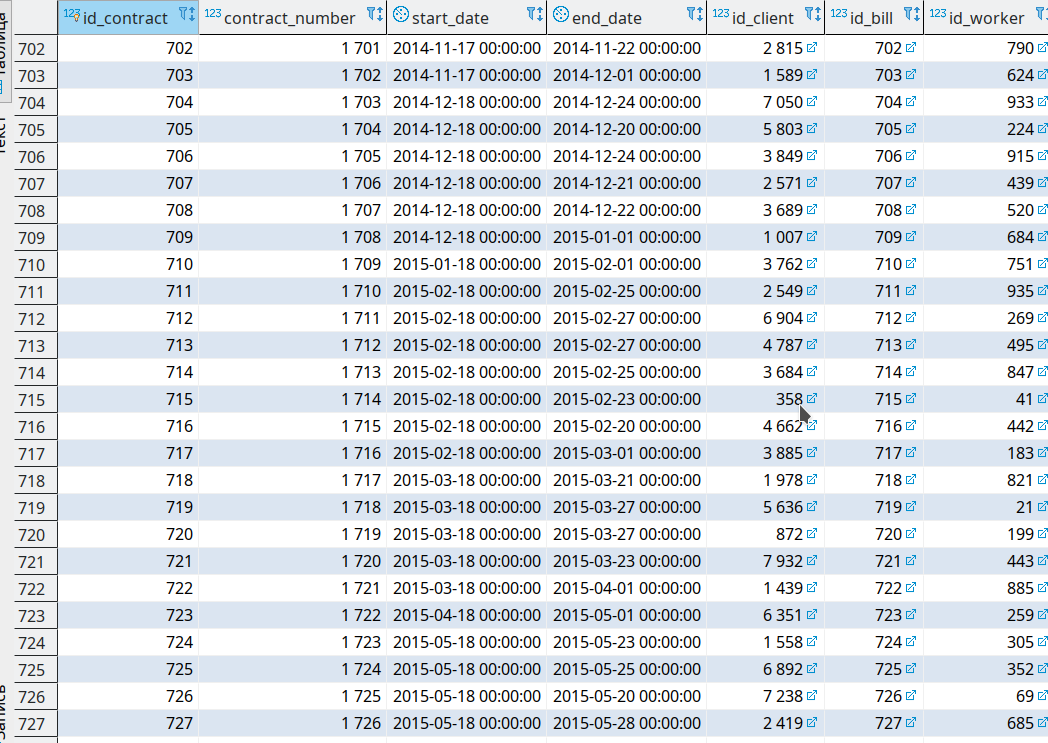


Таблица 15 — Отзыв

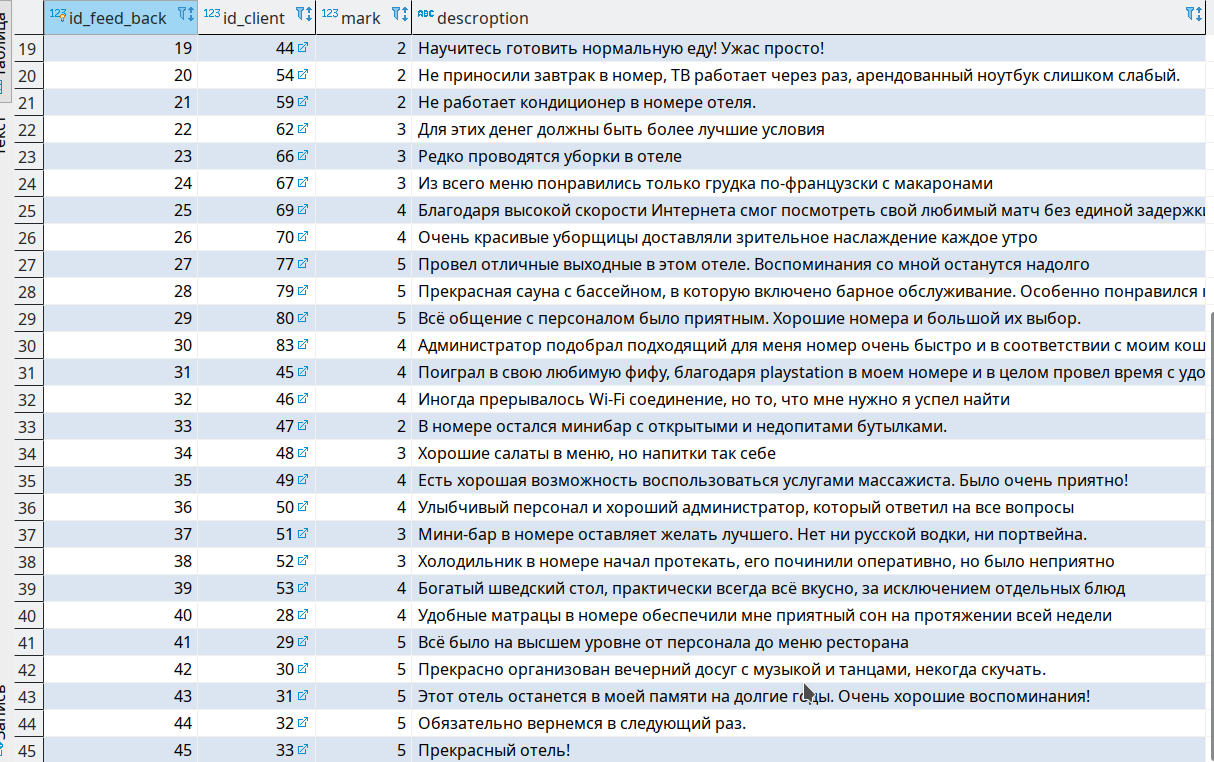


Таблица 16 — Физ. Лицо

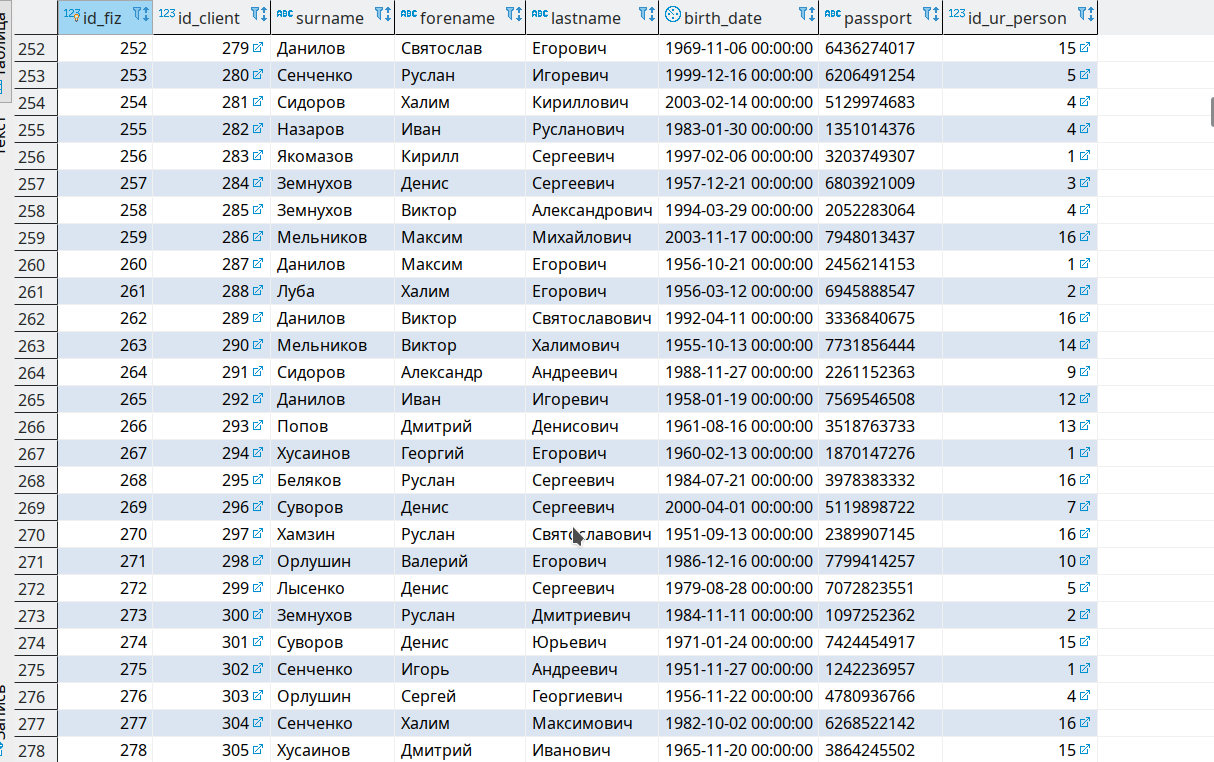


Таблица 17 — Статус бронирования

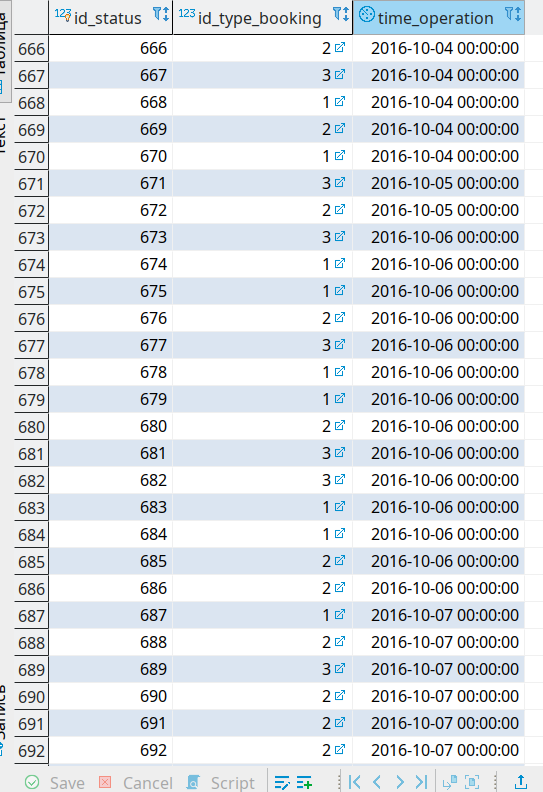
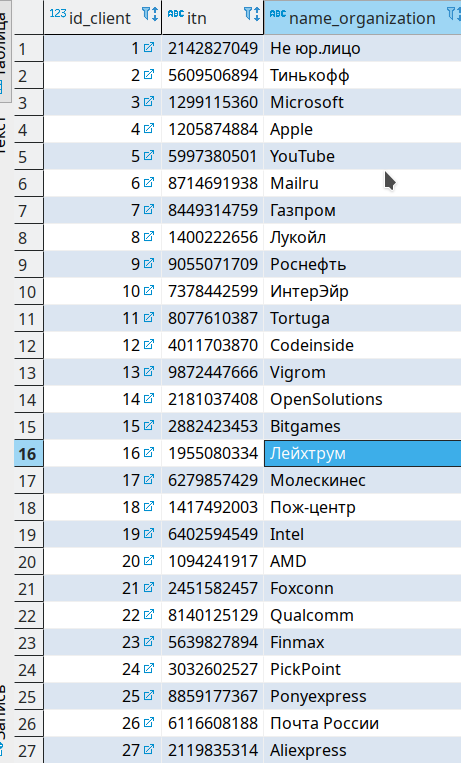


Таблица 18 — Юр.лицо



для заполнения таблиц отзыв, корпус, номер, услуга, номер-услуга(промежуточная таблица), скидка, тип бронирования использовался скпипт на SQL.

Для остальных таблиц(клиент, юр.лицо, физ. лицо, сотрудник,договор, бронирование, статус бронирования, клиент в бронировании(промежуточная таблица),позиция счета, счет, наименовании услуги в позиции счета(промежуточная таблица)) использовался следующий код на python3.6:

import random

import time

import psycopg2

import datetime

conn = psycopg2.connect(dbname='hotelaaa', user='admin',

password='samihad228', host='localhost')

cursor = conn.cursor()

def str\_time\_prop(start, end, format, prop):

stime = time.mktime(time.strptime(start, format))

etime = time.mktime(time.strptime(end, format))

ptime = stime + prop \* (etime - stime)

return time.strftime(format, time.localtime(ptime))

def random\_date(start, end, prop):

return str\_time\_prop(start, end, '%d.%m.%Y', prop)

first\_name\_man = ('Халим','Максим','Ильяс','Андрей','Святослав','Никита','Петр','Георгий','Руслан','Денис','Михаил','Рустам','Дмитрий','Егор','Кирилл','Валерий','Александр','Сергей','Иван','Виктор','Юрий','Игорь')

second\_name\_man=('Лысенко','Гаврилов','Данилов','Красовский','Дюжев','Макаров','Сидоров','Сенченко','Попов','Земнухов','Иванов','Петров','Орлушин','Чистяков','Русин','Хамзин','Суворов','Хусаинов','Пантелеев','Данилов','Назаров','Назаркин','Мельников','Макаров','Якомазов','Луба','Кулишов','Беляков')

last\_name\_man = ('Халимович','Максимович','Ильясович','Андреевич','Святославович','Никитич','Георгиевич','Русланович','Денисович','Михайлович','Рустамович','Дмитриевич','Егорович','Кириллович','Валерьевич','Александрович','Сергеевич','Иванович','Викторович','Юрьевич','Игоревич')

count\_id = 1

street = ('ул.Московская ','ул.Дмитриевская','пр.Победы ','пр.Строителей','ул.Ладожская','ул.Циолковского','ул.Водонаева ','ул.Октябрьская','ул.Ленина','ул.Сталинская','площадь Рефолюции','ул.Дачная','ул.Радужная','ул.Сиреневая','ул.Хорошая','ул.Виражная','ул.Гагарина')

city=('г.Москва','г.Пенза','г.Воронеж','г.Санкт-Петербург','г.Витебск','г.Казань','г.Сочи','г.Иваново','г.Мурманск','г.Волгоград','г.Ростов','г.Кузнецк','г.Сургут','г.Анадырь')

ur\_name=['Не юр.лицо','Тинькофф','Microsoft','Apple','YouTube','Mailru','Газпром','Лукойл','Роснефть','ИнтерЭйр','Tortuga','Codeinside','Vigrom','OpenSolutions','Bitgames','Лейхтрум','Молескинес','Пож-центр','Intel','AMD','Foxconn','Qualcomm','Finmax','PickPoint','Ponyexpress','Почта России','Aliexpress']

#Заполнение клиента

for i in range(800):

addres = f'{random.choice(city)}, {random.choice(street)} {random.randint(1,100)}, кв.{random.randint(1,500)}'

number = str(random.randint(11111111111,99999999999))

query = f'INSERT INTO Client(Adress, Phone\_number) values(\'{addres}\',\'{number}\')'

cursor.execute(query)

conn.commit()

#Заполнение юр лица

for i in range(len(ur\_name)):

query = f'INSERT INTO UR\_Person(id\_client,ITN,Name\_organization) values ({count\_id},\'{random.randint(1000000000,9999999999)}\',\'{ur\_name[i]}\')'

cursor.execute(query)

conn.commit()

count\_id += 1

#заполение физ\_лица от юрлица

for i in range(320):

query = f'INSERT INTO fiz\_person(id\_client,id\_ur\_person, Surname, forename, Lastname, Birth\_date, Passport) VALUES({count\_id},{random.randint(1,16)},\'{random.choice(second\_name\_man)}\',\'{random.choice(first\_name\_man)}\',\'{random.choice(last\_name\_man)}\',\'{random\_date("1.1.1950", "1.1.2005", random.random())}\',\'{str(random.randint(1000000000, 9999999999))}\')'

cursor.execute(query)

conn.commit()

count\_id+=1

#заполение физ\_лица

for i in range(180):

query = f'INSERT INTO fiz\_person(id\_client,id\_ur\_person, Surname, forename, Lastname, Birth\_date, Passport) VALUES({count\_id},{1},\'{random.choice(second\_name\_man)}\',\'{random.choice(first\_name\_man)}\',\'{random.choice(last\_name\_man)}\',\'{random\_date("1.1.1950", "1.1.2005", random.random())}\',\'{str(random.randint(1000000000, 9999999999))}\')'

cursor.execute(query)

conn.commit()

count\_id+=1

employment\_number = 1000

#заполение сотрудника

for i in range(100):

addres = f'{random.choice(city)}, {random.choice(street)} {random.randint(1, 100)}, кв.{random.randint(1, 500)}'

query=f'INSERT INTO Worker(Employment\_number, Surname, sorename, Lastname, Birth\_date, Sallary, Passport, Address, ITN) values({employment\_number},\'{random.choice(second\_name\_man)}\',\'{random.choice(first\_name\_man)}\',\'{random.choice(last\_name\_man)}\',\'{random\_date("1.1.1950", "1.1.2005", random.random())}\',{random.randint(15,150)\*1000},\'{str(random.randint(1000000000, 9999999999))}\',\'{addres}\',\'{str(random.randint(100000000000, 999999999999))}\')'

employment\_number=+1

cursor.execute(query)

conn.commit()

count\_id += 1

cursor.execute("SELECT COUNT(\*) FROM Worker;")

ver = cursor.fetchall()

for row in ver:

worker\_count = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM client;")

ver = cursor.fetchall()

for row in ver:

client\_count = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM ur\_person;")

ver = cursor.fetchall()

for row in ver:

ur\_person\_count = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM fiz\_person;")

ver = cursor.fetchall()

for row in ver:

fiz\_person\_count = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM type\_booking;")

ver = cursor.fetchall()

for row in ver:

type\_booking = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM room;")

ver = cursor.fetchall()

for row in ver:

room\_count = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM disscount;")

ver = cursor.fetchall()

for row in ver:

disscount\_count = int(row[0])

conn.commit()

cursor.execute("SELECT COUNT(\*) FROM service;")

ver = cursor.fetchall()

for row in ver:

service\_count = int(row[0])

conn.commit()

contract\_number = 1000

id\_bill = 1

year = 2001

month = 1

day = 1

coin = 1

#заполение договора

for i in range(1000):

query = "insert into bill(ammount) values (0);"

cursor.execute(query)

conn.commit()

start\_date = datetime.date(year, month, day)

end\_day = day + random.randint(1, 14)

if (end\_day > 28):

month += 1

end\_day = 1

if (month == 13):

year += 1

month = 1

end\_date = datetime.date(year, month,end\_day)

query = f'insert into contract(contract\_number, id\_client, id\_bill, id\_worker, start\_date, end\_date) values' \

f'({contract\_number},' \

f'{random.randint(1,client\_count)},' \

f'{id\_bill},' \

f'{random.randint(1,worker\_count)},' \

f'\'{start\_date}\',' \

f'\'{end\_date}\')'

coin = random.randint(1,10000)

if(coin > 9000):

day+=1

if(day > 28):

month += 1

day = 1

if (month == 13):

year += 1

month = 1

contract\_number+=1

id\_bill+=1

cursor.execute(query)

conn.commit()

conn.close()

cursor.execute("SELECT COUNT(\*) FROM contract;")

ver = cursor.fetchall()

for row in ver:

contract\_count = int(row[0])

conn.commit()

#заполение бронирования

i = 1

year = 2001

month = 1

day = 1

coin = 1

status = 1

while( i < contract\_count):

setlement = datetime.date(year, month, day)

booking = datetime.date(year, month, day)

end\_day = day + random.randint(1, 14)

if (end\_day > 28):

month += 1

end\_day = 1

if (month == 13):

year += 1

month = 1

departe = datetime.date(year, month,end\_day)

coin = random.randint(1, 10000)

if(coin > 9000):

day+=1

if(day > 28):

month += 1

day = 1

if (month == 13):

year += 1

month = 1

query = f"insert into status\_booking (id\_type\_booking, time\_operation) values" \

f"({random.randint(1,type\_booking)}," \

f"\'{booking}\');"

cursor.execute(query)

conn.commit()

query = f"insert into booking (id\_contract, id\_status, settlement\_time, departure\_time, booking\_time, id\_room) values" \

f"({i}," \

f"{status}," \

f"\'{setlement}\'," \

f"\'{departe}\'," \

f"\'{booking}\'," \

f"{random.randint(1,room\_count)})"

cursor.execute(query)

conn.commit()

i+=1

status+=1

#заполение клиент-бронирование

cursor.execute("SELECT COUNT(\*) FROM booking;")

ver = cursor.fetchall()

for row in ver:

booking\_count = int(row[0])

conn.commit()

i = 1

while(i < booking\_count):

query =f"insert into client\_booking(id\_client, id\_booking) values " \

f"({random.randint(1,client\_count)},{i})"

cursor.execute(query)

conn.commit()

coin = random.randint(1,6)

if(coin < 4):

i+=1

cursor.execute("SELECT COUNT(\*) FROM bill;")

ver = cursor.fetchall()

for row in ver:

bill\_count = int(row[0])

conn.commit()

i = 1

#заполение позиций счета

while(i < bill\_count):

money = random.randint(300,1000)

query = f"insert into bilt\_position(id\_bill,id\_disscount,without\_vat,without\_disscount) values" \

f"({i}," \

f"{random.randint(1,disscount\_count)}," \

f"{money}," \

f"{money+ random.randint(100,500)});"

cursor.execute(query)

conn.commit()

coin = random.randint(1, 10)

if(coin > 3):

i+=1

cursor.execute("SELECT COUNT(\*) FROM bilt\_position;")

ver = cursor.fetchall()

for row in ver:

bill\_position\_count = int(row[0])

conn.commit()

i=1

#заполение наименований услуг в позициях счета

while(i < bill\_position\_count):

query = f"insert into bilt\_position\_service(id\_bilt\_position, id\_service) values" \

f"({i}," \

f"{random.randint(1,service\_count)});"

cursor.execute(query)

conn.commit()

i+=1