**Этапы выполнения работы:**

1. Файлы скриптов:
   1. Файл создания хранимых процедур и распределения прав доступа;



* 1. Файл заполнение структуры данными;



* 1. Файл манипулирования над данными;



* 1. Файл создания пересоздания структуры БД;



* 1. Файл удаления структуры.



1. Приложение 1, создание хранимых процедур и распределение прав доступа;
2. Приложение 2, заполнение структуры базы данных;
3. Приложение 3, выборка данных и манипулирование данными;
4. Создание структуры базы данных;

Таблица 1 – Хронология создания структуры БД

| № ПП | Файл | Скрипт |
| --- | --- | --- |
|  | Create.sql | create or replace procedure Structure\_Create ()  language plpgsql  as $$  begin  -- Таблица Класс (Class)  CREATE TABLE Class (  ID\_Class SERIAL NOT NULL CONSTRAINT PK\_Class PRIMARY KEY,  Name\_Class VARCHAR(12) NOT NULL,  Image\_Class VARCHAR(100) NOT NULL  );  -- Таблица Статус номера (Status\_room)  CREATE TABLE Status\_Room (  ID\_Status\_Room SERIAL NOT NULL CONSTRAINT PK\_Status\_Room PRIMARY KEY,  Name\_Status\_room VARCHAR(12) NOT NULL  );  -- Таблица Номер (Room)  CREATE TABLE Room (  ID\_Room SERIAL NOT NULL CONSTRAINT PK\_Room PRIMARY KEY,  Number\_Room VARCHAR(3) NOT NULL,  Daily\_Cost\_Room INT NOT NULL,  Class\_ID INT NOT NULL REFERENCES Class(ID\_Class),  Status\_Room\_ID INT NOT NULL REFERENCES Status\_Room(ID\_Status\_Room)  );  -- Таблица Комплектация (Configuration)  CREATE TABLE Configuration (  ID\_Configuration SERIAL NOT NULL CONSTRAINT PK\_Configuration PRIMARY KEY,  Furniture\_Configuration VARCHAR(14) NOT NULL  );  -- Таблица Комплектация номера (Room\_Configuration)  CREATE TABLE Room\_Configuration (  ID\_Room\_Configuration SERIAL NOT NULL CONSTRAINT PK\_Room\_Configuration PRIMARY KEY,  Quantity\_Room\_Configuration INT NOT NULL,  Configuration\_ID INT NOT NULL REFERENCES Configuration(ID\_Configuration),  Room\_ID INT NOT NULL REFERENCES Room(ID\_Room)  );  -- Таблица Опции (Options)  CREATE TABLE Options (  ID\_Options SERIAL NOT NULL CONSTRAINT PK\_Options PRIMARY KEY,  Name\_Options VARCHAR(13) NOT NULL  );  -- Таблица Опции номера (Room\_Options)  CREATE TABLE Room\_Options (  ID\_Room\_Options SERIAL NOT NULL CONSTRAINT PK\_Room\_Options PRIMARY KEY,  Quantity\_Room\_Options INT NOT NULL,  Option\_ID INT NOT NULL REFERENCES Options(ID\_Options),  Room\_ID INT NOT NULL REFERENCES Room(ID\_Room)  );  -- Таблица Бронирование (Reservation)  CREATE TABLE Reservation (  ID\_Reservation SERIAL NOT NULL CONSTRAINT PK\_Reservation PRIMARY KEY,  Number\_Reservation VARCHAR(16) NOT NULL,  Creation\_Date\_Reservation DATE NOT NULL,  Checkin\_Date\_Reservation DATE NOT NULL,  Checkin\_Time\_Reservation TIME NOT NULL,  Checkout\_Date\_Reservation DATE NOT NULL,  Checkout\_Time\_Reservation TIME NOT NULL,  Total\_Cost\_Reservation INT NOT NULL,  Room\_ID INT NOT NULL REFERENCES Room(ID\_Room)  );  -- Таблица Клиент (Client)  CREATE TABLE Client (  ID\_Client SERIAL NOT NULL CONSTRAINT PK\_Client PRIMARY KEY,  Surname\_Client VARCHAR(50) NOT NULL,  Name\_Client VARCHAR(50) NOT NULL,  Patronymic\_Client VARCHAR(50),  Document\_Series\_Client VARCHAR(5) NOT NULL,  Document\_Number\_Client INT NOT NULL,  Date\_of\_Birth\_Client DATE NOT NULL,  Gender\_Client VARCHAR(3) NOT NULL,  Issued\_By\_Client VARCHAR(70) NOT NULL,  Issued\_Date\_Client DATE NOT NULL,  Code\_Department\_Client VARCHAR(7) NOT NULL,  Phone\_Client VARCHAR(16) NOT NULL,  Email\_Address\_Client VARCHAR(70) NOT NULL,  Login\_Client VARCHAR(50) NOT NULL,  Password\_Client VARCHAR(50) NOT NULL  );  -- Таблица Бронирование клиента (Client\_Reservation)  CREATE TABLE Client\_Reservation (  ID\_Client\_Reservation SERIAL NOT NULL CONSTRAINT PK\_Client\_Reservation PRIMARY KEY,  Client\_ID INT NOT NULL REFERENCES Client(ID\_Client),  Reservation\_ID INT NOT NULL REFERENCES Reservation(ID\_Reservation)  );  -- Таблица Дополнительные услуги (Add\_Services)  CREATE TABLE Add\_Services (  ID\_Add\_Services SERIAL NOT NULL CONSTRAINT PK\_Add\_Services PRIMARY KEY,  Name\_Add\_Services VARCHAR(8) NOT NULL,  Cost\_Add\_Services INT NOT NULL  );  -- Таблица Дополнительные услуги бронирования (Add\_Services\_Reservation)  CREATE TABLE Add\_Services\_Reservation (  ID\_Add\_Services\_Reservation SERIAL NOT NULL CONSTRAINT PK\_Add\_Services\_Reservation PRIMARY KEY,  Add\_Services\_ID INT NOT NULL REFERENCES Add\_Services(ID\_Add\_Services),  Reservation\_ID INT NOT NULL REFERENCES Reservation(ID\_Reservation)  );  -- Таблица Сотрудник (Employee)  CREATE TABLE Employee (  ID\_Employee SERIAL NOT NULL CONSTRAINT PK\_Employee PRIMARY KEY,  Surname\_Employee VARCHAR(50) NOT NULL,  Name\_Employee VARCHAR(50) NOT NULL,  Patronymic\_Employee VARCHAR(50),  Login\_Employee VARCHAR(50) NOT NULL,  Password\_Employee VARCHAR(50) NOT NULL  );  -- Таблица Договор (Contract)  CREATE TABLE Contract (  ID\_Contract SERIAL NOT NULL CONSTRAINT PK\_Contract PRIMARY KEY,  Number\_Contract VARCHAR(12) NOT NULL,  Creation\_Date\_Contract DATE NOT NULL,  Creation\_Time\_Contract TIME NOT NULL,  Reservation\_ID INT NOT NULL REFERENCES Reservation(ID\_Reservation),  Employee\_ID INT NOT NULL REFERENCES Employee(ID\_Employee)  );  -- Создание индексов  -- Индексы для таблицы Класс (Class)  CREATE INDEX IF NOT EXISTS Index\_ID\_Class ON Class(ID\_Class);  -- Индексы для таблицы Статус номера (Status\_Room)  CREATE INDEX IF NOT EXISTS Index\_ID\_Status\_Room ON Status\_Room(ID\_Status\_Room);  CREATE UNIQUE INDEX Index\_Name\_Status\_Room ON Status\_Room(Name\_Status\_Room);  -- Индексы для таблицы Номер (Room)  CREATE INDEX IF NOT EXISTS Index\_ID\_Room ON Room(ID\_Room);  CREATE UNIQUE INDEX Index\_Number\_Room ON Room(Number\_Room);  -- Индексы для таблицы Комплектация (Configuration)  CREATE INDEX IF NOT EXISTS Index\_ID\_Configuration ON Configuration(ID\_Configuration);  -- Индексы для таблицы Комплектация номера (Room\_Configuration)  CREATE INDEX IF NOT EXISTS Index\_ID\_Room\_Configuration ON Room\_Configuration(ID\_Room\_Configuration);  -- Индексы для таблицы Опции (Options)  CREATE INDEX IF NOT EXISTS Index\_ID\_Options ON Options(ID\_Options);  -- Индексы для таблицы Опции номера (Room\_Options)  CREATE INDEX IF NOT EXISTS Index\_ID\_Room\_Options ON Room\_Options(ID\_Room\_Options);  -- Индексы для таблицы Бронирование (Reservation)  CREATE INDEX IF NOT EXISTS Index\_ID\_Reservation ON Reservation(ID\_Reservation);  CREATE UNIQUE INDEX Index\_Number\_Reservation ON Reservation(Number\_Reservation);  -- Индексы для таблицы Клиент (Client)  CREATE INDEX IF NOT EXISTS Index\_ID\_Client ON Client(ID\_Client);  CREATE INDEX index\_surname\_client ON Client (Surname\_Client);  CREATE INDEX index\_name\_client ON Client (Name\_Client);  CREATE INDEX index\_patronymic\_client ON Client (Patronymic\_Client);  CREATE INDEX index\_document\_series\_client ON Client (Document\_Series\_Client);  CREATE INDEX index\_document\_number\_client ON Client (Document\_Number\_Client);  CREATE INDEX index\_date\_of\_birth\_client ON Client (Date\_of\_Birth\_Client);  CREATE INDEX index\_gender\_client ON Client (Gender\_Client);  CREATE INDEX index\_issued\_by\_client ON Client (Issued\_By\_Client);  CREATE INDEX index\_issued\_date\_client ON Client (Issued\_Date\_Client);  CREATE INDEX index\_code\_department\_client ON Client (Code\_Department\_Client);  CREATE UNIQUE INDEX Index\_Phone\_Client ON Client(Phone\_Client);  CREATE UNIQUE INDEX Index\_Email\_Client ON Client(Email\_Address\_Client);  CREATE UNIQUE INDEX Index\_Login\_Client ON Client(Login\_Client);  -- Индексы для таблицы Бронирование клиента (Client\_Reservation)  CREATE INDEX IF NOT EXISTS Index\_ID\_Client\_Reservation ON Client\_Reservation(ID\_Client\_Reservation);  -- Индексы для таблицы Дополнительные услуги (Add\_Services)  CREATE INDEX IF NOT EXISTS Index\_ID\_Add\_Services ON Add\_Services(ID\_Add\_Services);  -- Индексы для таблицы Дополнительные услуги бронирования (Add\_Services\_Reservation)  CREATE INDEX IF NOT EXISTS Index\_ID\_Add\_Services\_Reservation ON Add\_Services\_Reservation(ID\_Add\_Services\_Reservation);  -- Индексы для таблицы Сотрудник (Employee)  CREATE INDEX IF NOT EXISTS Index\_ID\_Employee ON Employee(ID\_Employee);  CREATE UNIQUE INDEX Index\_Login\_Employee ON Employee(Login\_Employee);  -- Индексы для таблицы Договор (Contract)  CREATE INDEX IF NOT EXISTS Index\_ID\_Contract ON Contract(ID\_Contract);  CREATE UNIQUE INDEX Index\_Number\_Contract ON Contract(Number\_Contract);  -- Выдача прав доступа  -- Предоставление прав для роли rl\_Client  GRANT SELECT ON Class TO rl\_Client;  GRANT SELECT ON Status\_Room TO rl\_Client;  GRANT SELECT ON Room TO rl\_Client;  GRANT SELECT ON Configuration TO rl\_Client;  GRANT SELECT ON Room\_Configuration TO rl\_Client;  GRANT SELECT ON Options TO rl\_Client;  GRANT SELECT ON Room\_Options TO rl\_Client;  GRANT SELECT ON Reservation TO rl\_Client;  GRANT SELECT ON Client\_Reservation TO rl\_Client;  GRANT SELECT ON Add\_Services TO rl\_Client;  GRANT SELECT ON Add\_Services\_Reservation TO rl\_Client;  GRANT SELECT ON Contract TO rl\_Client;  -- Предоставление прав для роли rl\_Employee  GRANT SELECT, INSERT, UPDATE ON Room\_Configuration TO rl\_Employee;  GRANT SELECT, INSERT, UPDATE ON Room\_Options TO rl\_Employee;  GRANT SELECT, INSERT, UPDATE ON Reservation TO rl\_Employee;  GRANT SELECT, INSERT, UPDATE ON Client TO rl\_Employee;  GRANT SELECT, INSERT, UPDATE ON Client\_Reservation TO rl\_Employee;  GRANT SELECT, INSERT, UPDATE ON Add\_Services\_Reservation TO rl\_Employee;  GRANT SELECT, INSERT, UPDATE ON Contract TO rl\_Employee;  GRANT SELECT ON Class TO rl\_Employee;  GRANT SELECT ON Status\_room TO rl\_Employee;  GRANT SELECT ON Room TO rl\_Employee;  GRANT SELECT ON Configuration TO rl\_Employee;  GRANT SELECT ON Options TO rl\_Employee;  GRANT SELECT ON Add\_Services TO rl\_Employee;  GRANT SELECT ON Employee TO rl\_Employee;  -- Предоставление прав для роли rl\_Administrator  GRANT SELECT, INSERT, UPDATE, DELETE ON Class TO rl\_Administrator;  GRANT SELECT, INSERT, UPDATE, DELETE ON Status\_room TO rl\_Administrator;  GRANT SELECT, INSERT, UPDATE, DELETE ON Room TO rl\_Administrator;  GRANT SELECT, INSERT, UPDATE, DELETE ON Configuration TO rl\_Administrator;  GRANT SELECT, INSERT, UPDATE, DELETE ON Options TO rl\_Administrator;  GRANT SELECT, INSERT, UPDATE, DELETE ON Add\_Services TO rl\_Administrator;  GRANT SELECT, INSERT, UPDATE, DELETE ON Employee TO rl\_Administrator;  GRANT SELECT, DELETE ON Room\_Configuration TO rl\_Administrator;  GRANT SELECT, DELETE ON Room\_Options TO rl\_Administrator;  GRANT SELECT, DELETE ON Reservation TO rl\_Administrator;  GRANT SELECT, DELETE ON Client TO rl\_Administrator;  GRANT SELECT, DELETE ON Client\_Reservation TO rl\_Administrator;  GRANT SELECT, DELETE ON Add\_Services\_Reservation TO rl\_Administrator;  GRANT SELECT, DELETE ON Contract TO rl\_Administrator;  end;  $$; |
|  | Proceudre.sql | -- Таблица Класс (Class)  -- Добавление  CREATE OR REPLACE PROCEDURE Class\_Insert (  p\_Name\_Class VARCHAR(12),  p\_Image\_Class VARCHAR(100)  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Class (Name\_Class, Image\_Class)  VALUES (p\_Name\_Class, p\_Image\_Class);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Class\_Update (  p\_ID\_Class INT,  p\_Name\_Class VARCHAR(12),  p\_Image\_Class VARCHAR(100)  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Class  SET Name\_Class = p\_Name\_Class,  Image\_Class = p\_Image\_Class  WHERE ID\_Class = p\_ID\_Class;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Class\_Delete (  p\_ID\_Class INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Class  WHERE ID\_Class = p\_ID\_Class;  END;  $$;  -- Таблица Статус номера (Status\_room)  -- Добавление  CREATE OR REPLACE PROCEDURE Status\_room\_Insert (  p\_Name\_Status\_room VARCHAR(12)  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Status\_room (Name\_Status\_room)  VALUES (p\_Name\_Status\_room);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Status\_room\_Update (  p\_ID\_Status\_room INT,  p\_Name\_Status\_room VARCHAR(12)  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Status\_room  SET Name\_Status\_room = p\_Name\_Status\_room  WHERE ID\_Status\_room = p\_ID\_Status\_room;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Status\_room\_Delete (  p\_ID\_Status\_room INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Status\_room  WHERE ID\_Status\_room = p\_ID\_Status\_room;  END;  $$;  -- Таблица Номер (Room)  -- Добавление  CREATE OR REPLACE PROCEDURE Room\_Insert (  p\_Number\_Room VARCHAR(3),  p\_Class\_ID INT,  p\_Daily\_Cost\_Room INT,  p\_Status\_Room\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Room (Number\_Room, Class\_ID, Daily\_Cost\_Room, Status\_Room\_ID)  VALUES (p\_Number\_Room, p\_Class\_ID, p\_Daily\_Cost\_Room, p\_Status\_Room\_ID);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Room\_Update (  p\_ID\_Room INT,  p\_Number\_Room VARCHAR(3),  p\_Class\_ID INT,  p\_Daily\_Cost\_Room INT,  p\_Status\_Room\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Room  SET Number\_Room = p\_Number\_Room,  Class\_ID = p\_Class\_ID,  Daily\_Cost\_Room = p\_Daily\_Cost\_Room,  Status\_Room\_ID = p\_Status\_Room\_ID  WHERE ID\_Room = p\_ID\_Room;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Room\_Delete (  p\_ID\_Room INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Room  WHERE ID\_Room = p\_ID\_Room;  END;  $$;  -- Таблица Комплектация (Configuration)  -- Добавление  CREATE OR REPLACE PROCEDURE Configuration\_Insert (  p\_Furniture\_Configuration VARCHAR(14)  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Configuration (Furniture\_Configuration)  VALUES (p\_Furniture\_Configuration);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Configuration\_Update (  p\_ID\_Configuration INT,  p\_Furniture\_Configuration VARCHAR(14)  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Configuration  SET Furniture\_Configuration = p\_Furniture\_Configuration  WHERE ID\_Configuration = p\_ID\_Configuration;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Configuration\_Delete (  p\_ID\_Configuration INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Configuration  WHERE ID\_Configuration = p\_ID\_Configuration;  END;  $$;  -- Таблица Комплектация номера (Room\_Configuration)  -- Добавление  CREATE OR REPLACE PROCEDURE Room\_Configuration\_Insert (  p\_Quantity\_Room\_Configuration INT,  p\_Configuration\_ID INT,  p\_Room\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Room\_Configuration (Quantity\_Room\_Configuration, Configuration\_ID, Room\_ID)  VALUES (p\_Quantity\_Room\_Configuration, p\_Configuration\_ID, p\_Room\_ID);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Room\_Configuration\_Update (  p\_ID\_Room\_Configuration INT,  p\_Quantity\_Room\_Configuration INT,  p\_Configuration\_ID INT,  p\_Room\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Room\_Configuration  SET Quantity\_Room\_Configuration = p\_Quantity\_Room\_Configuration,  Configuration\_ID = p\_Configuration\_ID,  Room\_ID = p\_Room\_ID  WHERE ID\_Room\_Configuration = p\_ID\_Room\_Configuration;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Room\_Configuration\_Delete (  p\_ID\_Room\_Configuration INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Room\_Configuration  WHERE ID\_Room\_Configuration = p\_ID\_Room\_Configuration;  END;  $$;  -- Таблица Опции (Options)  -- Добавление  CREATE OR REPLACE PROCEDURE Options\_Insert (  p\_Name\_Options VARCHAR(13)  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Options (Name\_Options)  VALUES (p\_Name\_Options);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Options\_Update (  p\_ID\_Options INT,  p\_Name\_Options VARCHAR(13)  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Options  SET Name\_Options = p\_Name\_Options  WHERE ID\_Options = p\_ID\_Options;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Options\_Delete (  p\_ID\_Options INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Options  WHERE ID\_Options = p\_ID\_Options;  END;  $$;  -- Таблица Опции номера (Room\_Options)  -- Добавление  CREATE OR REPLACE PROCEDURE Room\_Options\_Insert (  p\_Quantity\_Room\_Options INT,  p\_Option\_ID INT,  p\_Room\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Room\_Options (Quantity\_Room\_Options, Option\_ID, Room\_ID)  VALUES (p\_Quantity\_Room\_Options, p\_Option\_ID, p\_Room\_ID);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Room\_Options\_Update (  p\_ID\_Room\_Options INT,  p\_Quantity\_Room\_Options INT,  p\_Option\_ID INT,  p\_Room\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Room\_Options  SET Quantity\_Room\_Options = p\_Quantity\_Room\_Options,  Option\_ID = p\_Option\_ID,  Room\_ID = p\_Room\_ID  WHERE ID\_Room\_Options = p\_ID\_Room\_Options;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Room\_Options\_Delete (  p\_ID\_Room\_Options INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Room\_Options  WHERE ID\_Room\_Options = p\_ID\_Room\_Options;  END;  $$;  -- Таблица Бронирование (Reservation)  -- Добавление  CREATE OR REPLACE PROCEDURE Reservation\_Insert (  p\_Number\_Reservation VARCHAR(16),  p\_Creation\_Date\_Reservation DATE,  p\_Checkin\_Date\_Reservation DATE,  p\_Checkin\_Time\_Reservation TIME,  p\_Checkout\_Date\_Reservation DATE,  p\_Checkout\_Time\_Reservation TIME,  p\_Room\_ID INT,  p\_Total\_Cost\_Reservation INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Reservation (Number\_Reservation, Creation\_Date\_Reservation, Checkin\_Date\_Reservation, Checkin\_Time\_Reservation, Checkout\_Date\_Reservation, Checkout\_Time\_Reservation, Room\_ID, Total\_Cost\_Reservation)  VALUES (p\_Number\_Reservation, p\_Creation\_Date\_Reservation, p\_Checkin\_Date\_Reservation, p\_Checkin\_Time\_Reservation, p\_Checkout\_Date\_Reservation, p\_Checkout\_Time\_Reservation, p\_Room\_ID, p\_Total\_Cost\_Reservation);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Reservation\_Update (  p\_ID\_Reservation INT,  p\_Number\_Reservation VARCHAR(16),  p\_Creation\_Date\_Reservation DATE,  p\_Checkin\_Date\_Reservation DATE,  p\_Checkin\_Time\_Reservation TIME,  p\_Checkout\_Date\_Reservation DATE,  p\_Checkout\_Time\_Reservation TIME,  p\_Room\_ID INT,  p\_Total\_Cost\_Reservation INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Reservation  SET Number\_Reservation = p\_Number\_Reservation,  Creation\_Date\_Reservation = p\_Creation\_Date\_Reservation,  Checkin\_Date\_Reservation = p\_Checkin\_Date\_Reservation,  Checkin\_Time\_Reservation = p\_Checkin\_Time\_Reservation,  Checkout\_Date\_Reservation = p\_Checkout\_Date\_Reservation,  Checkout\_Time\_Reservation = p\_Checkout\_Time\_Reservation,  Room\_ID = p\_Room\_ID,  Total\_Cost\_Reservation = p\_Total\_Cost\_Reservation  WHERE ID\_Reservation = p\_ID\_Reservation;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Reservation\_Delete (  p\_ID\_Reservation INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Reservation  WHERE ID\_Reservation = p\_ID\_Reservation;  END;  $$;  -- Таблица Клиент (Client)  -- Добавление  CREATE OR REPLACE PROCEDURE Client\_Insert (  p\_Surname\_Client VARCHAR(50),  p\_Name\_Client VARCHAR(50),  p\_Patronymic\_Client VARCHAR(50),  p\_Document\_Series\_Client VARCHAR(5),  p\_Document\_Number\_Client INT,  p\_Date\_of\_Birth\_Client DATE,  p\_Gender\_Client VARCHAR(3),  p\_Issued\_By\_Client VARCHAR(70),  p\_Issued\_Date\_Client DATE,  p\_Code\_Department\_Client VARCHAR(7),  p\_Phone\_Client VARCHAR(16),  p\_Email\_Address\_Client VARCHAR(70),  p\_Login\_Client VARCHAR(50),  p\_Password\_Client VARCHAR(50)  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Client (Surname\_Client, Name\_Client, Patronymic\_Client, Document\_Series\_Client, Document\_Number\_Client, Date\_of\_Birth\_Client, Gender\_Client, Issued\_By\_Client, Issued\_Date\_Client, Code\_Department\_Client, Phone\_Client, Email\_Address\_Client, Login\_Client, Password\_Client)  VALUES (p\_Surname\_Client, p\_Name\_Client, p\_Patronymic\_Client, p\_Document\_Series\_Client, p\_Document\_Number\_Client, p\_Date\_of\_Birth\_Client, p\_Gender\_Client, p\_Issued\_By\_Client, p\_Issued\_Date\_Client, p\_Code\_Department\_Client, p\_Phone\_Client, p\_Email\_Address\_Client, p\_Login\_Client, p\_Password\_Client);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Client\_Update (  p\_ID\_Client INT,  p\_Surname\_Client VARCHAR(50),  p\_Name\_Client VARCHAR(50),  p\_Patronymic\_Client VARCHAR(50),  p\_Document\_Series\_Client VARCHAR(5),  p\_Document\_Number\_Client INT,  p\_Date\_of\_Birth\_Client DATE,  p\_Gender\_Client VARCHAR(3),  p\_Issued\_By\_Client VARCHAR(70),  p\_Issued\_Date\_Client DATE,  p\_Code\_Department\_Client VARCHAR(7),  p\_Phone\_Client VARCHAR(16),  p\_Email\_Address\_Client VARCHAR(70),  p\_Login\_Client VARCHAR(50),  p\_Password\_Client VARCHAR(50)  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Client  SET Surname\_Client = p\_Surname\_Client,  Name\_Client = p\_Name\_Client,  Patronymic\_Client = p\_Patronymic\_Client,  Document\_Series\_Client = p\_Document\_Series\_Client,  Document\_Number\_Client = p\_Document\_Number\_Client,  Date\_of\_Birth\_Client = p\_Date\_of\_Birth\_Client,  Gender\_Client = p\_Gender\_Client,  Issued\_By\_Client = p\_Issued\_By\_Client,  Issued\_Date\_Client = p\_Issued\_Date\_Client,  Code\_Department\_Client = p\_Code\_Department\_Client,  Phone\_Client = p\_Phone\_Client,  Email\_Address\_Client = p\_Email\_Address\_Client,  Login\_Client = p\_Login\_Client,  Password\_Client = p\_Password\_Client  WHERE ID\_Client = p\_ID\_Client;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Client\_Delete (  p\_ID\_Client INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Client  WHERE ID\_Client = p\_ID\_Client;  END;  $$;  -- Таблица Бронирование клиента (Client\_Reservation)  -- Добавление  CREATE OR REPLACE PROCEDURE Client\_Reservation\_Insert (  p\_Client\_ID INT,  p\_Reservation\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Client\_Reservation (Client\_ID, Reservation\_ID)  VALUES (p\_Client\_ID, p\_Reservation\_ID);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Client\_Reservation\_Update (  p\_ID\_Client\_Reservation INT,  p\_Client\_ID INT,  p\_Reservation\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Client\_Reservation  SET Client\_ID = p\_Client\_ID,  Reservation\_ID = p\_Reservation\_ID  WHERE ID\_Client\_Reservation = p\_ID\_Client\_Reservation;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Client\_Reservation\_Delete (  p\_ID\_Client\_Reservation INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Client\_Reservation  WHERE ID\_Client\_Reservation = p\_ID\_Client\_Reservation;  END;  $$;  -- Таблица Дополнительные услуги (Add\_Services)  -- Добавление  CREATE OR REPLACE PROCEDURE Add\_Services\_Insert (  p\_Name\_Add\_Services VARCHAR(8),  p\_Cost\_Add\_Services INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Add\_Services (Name\_Add\_Services, Cost\_Add\_Services)  VALUES (p\_Name\_Add\_Services, p\_Cost\_Add\_Services);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Add\_Services\_Update (  p\_ID\_Add\_Services INT,  p\_Name\_Add\_Services VARCHAR(8),  p\_Cost\_Add\_Services INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Add\_Services  SET Name\_Add\_Services = p\_Name\_Add\_Services,  Cost\_Add\_Services = p\_Cost\_Add\_Services  WHERE ID\_Add\_Services = p\_ID\_Add\_Services;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Add\_Services\_Delete (  p\_ID\_Add\_Services INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Add\_Services  WHERE ID\_Add\_Services = p\_ID\_Add\_Services;  END;  $$;  -- Дополнительные услуги бронирования (Add\_Services\_Reservation)  -- Добавление  CREATE OR REPLACE PROCEDURE Add\_Services\_Reservation\_Insert (  p\_Add\_Services\_ID INT,  p\_Reservation\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Add\_Services\_Reservation (Add\_Services\_ID, Reservation\_ID)  VALUES (p\_Add\_Services\_ID, p\_Reservation\_ID);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Add\_Services\_Reservation\_Update (  p\_ID\_Add\_Services\_Reservation INT,  p\_Add\_Services\_ID INT,  p\_Reservation\_ID INT  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Add\_Services\_Reservation  SET Add\_Services\_ID = p\_Add\_Services\_ID,  Reservation\_ID = p\_Reservation\_ID  WHERE ID\_Add\_Services\_Reservation = p\_ID\_Add\_Services\_Reservation;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Add\_Services\_Reservation\_Delete (  p\_ID\_Add\_Services\_Reservation INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Add\_Services\_Reservation  WHERE ID\_Add\_Services\_Reservation = p\_ID\_Add\_Services\_Reservation;  END;  $$;  -- Сотрудник (Employee)  -- Добавление  CREATE OR REPLACE PROCEDURE Employee\_Insert (  p\_Surname\_Employee VARCHAR(50),  p\_Name\_Employee VARCHAR(50),  p\_Patronymic\_Employee VARCHAR(50),  p\_Login\_Employee VARCHAR(50),  p\_Password\_Employee VARCHAR(50)  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Employee (Surname\_Employee, Name\_Employee, Patronymic\_Employee, Login\_Employee, Password\_Employee)  VALUES (p\_Surname\_Employee, p\_Name\_Employee, p\_Patronymic\_Employee, p\_Login\_Employee, p\_Password\_Employee);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Employee\_Update (  p\_ID\_Employee INT,  p\_Surname\_Employee VARCHAR(50),  p\_Name\_Employee VARCHAR(50),  p\_Patronymic\_Employee VARCHAR(50),  p\_Login\_Employee VARCHAR(50),  p\_Password\_Employee VARCHAR(50)  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Employee  SET Surname\_Employee = p\_Surname\_Employee,  Name\_Employee = p\_Name\_Employee,  Patronymic\_Employee = p\_Patronymic\_Employee,  Login\_Employee = p\_Login\_Employee,  Password\_Employee = p\_Password\_Employee  WHERE ID\_Employee = p\_ID\_Employee;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Employee\_Delete (  p\_ID\_Employee INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Employee  WHERE ID\_Employee = p\_ID\_Employee;  END;  $$;  -- Договор (Contract)  -- Добавление  CREATE OR REPLACE PROCEDURE Contract\_Insert (  p\_Number\_Contract VARCHAR(12),  p\_Reservation\_ID INT,  p\_Employee\_ID INT,  p\_Creation\_Date\_Contract DATE,  p\_Creation\_Time\_Contract TIME  )  LANGUAGE plpgsql  AS $$  BEGIN  INSERT INTO Contract (Number\_Contract, Reservation\_ID, Employee\_ID, Creation\_Date\_Contract, Creation\_Time\_Contract)  VALUES (p\_Number\_Contract, p\_Reservation\_ID, p\_Employee\_ID, p\_Creation\_Date\_Contract, p\_Creation\_Time\_Contract);  END;  $$;  -- Изменение  CREATE OR REPLACE PROCEDURE Contract\_Update (  p\_ID\_Contract INT,  p\_Number\_Contract VARCHAR(12),  p\_Reservation\_ID INT,  p\_Employee\_ID INT,  p\_Creation\_Date\_Contract DATE,  p\_Creation\_Time\_Contract TIME  )  LANGUAGE plpgsql  AS $$  BEGIN  UPDATE Contract  SET Number\_Contract = p\_Number\_Contract,  Reservation\_ID = p\_Reservation\_ID,  Employee\_ID = p\_Employee\_ID,  Creation\_Date\_Contract = p\_Creation\_Date\_Contract,  Creation\_Time\_Contract = p\_Creation\_Time\_Contract  WHERE ID\_Contract = p\_ID\_Contract;  END;  $$;  -- Удаление  CREATE OR REPLACE PROCEDURE Contract\_Delete (  p\_ID\_Contract INT  )  LANGUAGE plpgsql  AS $$  BEGIN  DELETE FROM Contract  WHERE ID\_Contract = p\_ID\_Contract;  END;  $$;  GRANT EXECUTE ON PROCEDURE Room\_Configuration\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Room\_Configuration\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Room\_Configuration\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Room\_Options\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Room\_Options\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Room\_Options\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Reservation\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Reservation\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Reservation\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Client\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Client\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Client\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Client\_Reservation\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Client\_Reservation\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Client\_Reservation\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Contract\_Insert TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Contract\_Update TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Contract\_Delete TO rl\_employee;  GRANT EXECUTE ON PROCEDURE Class\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Class\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Class\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Status\_room\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Status\_room\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Status\_room\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Configuration\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Configuration\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Configuration\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Options\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Options\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Options\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Employee\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Employee\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Employee\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Configuration\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Configuration\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Configuration\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Options\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Options\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Room\_Options\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Reservation\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Reservation\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Reservation\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Client\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Client\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Client\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Client\_Reservation\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Client\_Reservation\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Client\_Reservation\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Delete TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Contract\_Insert TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Contract\_Update TO rl\_administrator;  GRANT EXECUTE ON PROCEDURE Contract\_Delete TO rl\_administrator; |
|  | Insert.sql | INSERT INTO Class (Name\_Class, Image\_Class) VALUES  ('Стандартный', 'C:\\'),  ('Стандартный', 'C:\\'),  ('Бизнес-класс', 'C:\\');  CALL Class\_Insert('Де-люкс', 'C:\\');  CALL Class\_Insert('Де-люкс', 'C:\\');  CALL Class\_Insert('Люкс', 'C:\\');  INSERT INTO status\_room (name\_status\_room) VALUES  ('Свободен'),  ('Забронирован');  CALL Status\_room\_Insert('Недоступен');  CALL Status\_room\_Insert('Занят');  INSERT INTO Room (Number\_Room, Class\_ID, Daily\_Cost\_Room, Status\_Room\_ID) VALUES  ('201', 1, 1500, 2),  ('202', 2, 1600, 2),  ('203', 3, 2500, 4);  CALL Room\_Insert('302', 4, 3500, 1);  CALL Room\_Insert('303', 5, 3750, 4);  CALL Room\_Insert('304', 6, 5500, 4);  INSERT INTO Configuration (Furniture\_Configuration) VALUES  ('кровать'),  ('спальное место');  CALL Configuration\_Insert('стул');  CALL Configuration\_Insert('стол');  INSERT INTO Room\_Configuration (Quantity\_Room\_Configuration, Configuration\_ID, Room\_ID) VALUES  (2, 1, 1),  (4, 2, 1),  (4, 3, 1),  (2, 4, 1),  (2, 1, 2),  (4, 2, 2),  (6, 3, 2),  (2, 4, 2);  CALL Room\_Configuration\_Insert(1, 1, 3);  CALL Room\_Configuration\_Insert(3, 2, 3);  CALL Room\_Configuration\_Insert(3, 3, 3);  CALL Room\_Configuration\_Insert(1, 4, 3);  CALL Room\_Configuration\_Insert(2, 1, 4);  CALL Room\_Configuration\_Insert(4, 2, 4);  CALL Room\_Configuration\_Insert(3, 3, 4);  CALL Room\_Configuration\_Insert(1, 4, 4);  CALL Room\_Configuration\_Insert(1, 1, 5);  CALL Room\_Configuration\_Insert(1, 2, 5);  CALL Room\_Configuration\_Insert(3, 3, 5);  CALL Room\_Configuration\_Insert(1, 4, 5);  CALL Room\_Configuration\_Insert(2, 1, 6);  CALL Room\_Configuration\_Insert(3, 2, 6);  CALL Room\_Configuration\_Insert(4, 3, 6);  CALL Room\_Configuration\_Insert(1, 4, 6);  INSERT INTO Options (Name\_Options) VALUES  ('телевизор'),  ('wi-fi'),  ('полотенца'),  ('халат');  CALL Options\_Insert('тапочки');  CALL Options\_Insert('гель для душа');  CALL Options\_Insert('мини-бар');  CALL Options\_Insert('шампунь');  INSERT INTO Room\_Options (Quantity\_Room\_Options, Option\_ID, Room\_ID) VALUES  (1, 1, 1),  (1, 2, 1),  (3, 3, 1),  (1, 6, 1),  (2, 8, 1),  (1, 1, 2),  (1, 2, 2),  (6, 3, 2),  (2, 6, 2),  (4, 8, 2),  (1, 1, 3),  (1, 2, 3),  (8, 3, 3),  (4, 4, 3),  (4, 5, 3),  (8, 6, 3),  (12, 8, 3),  (1, 1, 4),  (1, 2, 4),  (1, 7, 4),  (8, 3, 4),  (3, 4, 4),  (3, 5, 4),  (6, 6, 4),  (12, 8, 4);  CALL Room\_Options\_Insert(1, 1, 5);  CALL Room\_Options\_Insert(1, 2, 5);  CALL Room\_Options\_Insert(1, 7, 5);  CALL Room\_Options\_Insert(3, 3, 5);  CALL Room\_Options\_Insert(3, 4, 5);  CALL Room\_Options\_Insert(3, 5, 5);  CALL Room\_Options\_Insert(6, 6, 5);  CALL Room\_Options\_Insert(12, 8, 5);  CALL Room\_Options\_Insert(1, 1, 6);  CALL Room\_Options\_Insert(1, 2, 6);  CALL Room\_Options\_Insert(1, 7, 6);  CALL Room\_Options\_Insert(3, 3, 6);  CALL Room\_Options\_Insert(4, 4, 6);  CALL Room\_Options\_Insert(4, 5, 6);  CALL Room\_Options\_Insert(8, 6, 6);  CALL Room\_Options\_Insert(14, 8, 6);  INSERT INTO Reservation (Number\_Reservation, Creation\_Date\_Reservation, Checkin\_Date\_Reservation, Checkin\_Time\_Reservation, Checkout\_Date\_Reservation, Checkout\_Time\_Reservation, Room\_ID, Total\_Cost\_Reservation) VALUES  ('БРН-000000001-23', '2023-08-10', '2023-09-05', '14:00:00', '2023-09-15', '12:00:00', 1, 38960),  ('БРН-000000002-23', '2023-08-27', '2023-09-03', '14:00:00', '2023-09-10', '12:00:00', 2, 36830),  ('БРН-000000003-23', '2023-08-16', '2023-09-10', '14:00:00', '2023-09-15', '12:00:00', 3, 12100);  CALL Reservation\_Insert('БРН-000000004-23', '2023-08-24', '2023-09-03', '14:00:00', '2023-09-23', '12:00:00', 4, 141184);  CALL Reservation\_Insert('БРН-000000005-23', '2023-09-01', '2023-09-11', '14:00:00', '2023-09-21', '12:00:00', 5, 28160);  INSERT INTO Client (Surname\_Client, Name\_Client, Patronymic\_Client, Document\_Series\_Client, Document\_Number\_Client, Date\_of\_Birth\_Client, Gender\_Client, Issued\_By\_Client, Issued\_Date\_Client, Code\_Department\_Client, Phone\_Client, Email\_Address\_Client, Login\_Client, Password\_Client) VALUES  ('Максимова', 'Дарья', 'Андреева', '45 26', 878214, '2000-02-23', 'Жен', 'Отделение УФМС России по городу Москве по Району ЮАО', '2020-03-24', '770-023', '+7(963)732-80-15', 'maksimova\_d\_s@ya.ru', 'MaksimovaDA', 'Pa$$w0rd'),  ('Романов', 'Владимир', 'Павлович', '43 72', 662899, '1999-07-10', 'Муж', 'Отделение УФМС России по городу Москве по Району СВАО', '2019-08-11', '770-225', '+7(983)328-52-94', 'Romanov\_v\_p@gmail.com', 'RomanovVP', 'Pa$$w0rd'),  ('Сергеева', 'Надежда', 'Николаевна', '45 76', 889212, '1999-10-08', 'Жен', 'Отделение УФМС России по городу Москве по Району ЮАО', '2019-11-09', '770-023', '+7(926)842-88-14', 'sergeeva\_n\_n@gmail.com', 'SergeevaNN', 'Pa$$w0rd');  CALL Client\_Insert('Павлов', 'Олег', 'Михайлович', '47 55', 123246, '1998-09-17', 'Муж', 'Отделение УФМС России по городу Москве по Району ЦАО', '2018-10-18', '770-115', '+7(978)628-39-03', 'pavlov\_o\_m@ya.ru', 'PavlovOM', 'Pa$$w0rd');  CALL Client\_Insert('Андреева', 'Екатерина', 'Владимировна', '47 10', 632788, '2001-01-26', 'Жен', 'Отделение УФМС России по городу Москве по Району ВАО', '2021-02-27', '770-116', '+7(962)882-77-16', 'andreeva\_e\_v@ya.ru', 'AndreevaEV', 'Pa$$w0rd');  INSERT INTO Client\_Reservation (Client\_ID, Reservation\_ID) VALUES  (1, 1);  CALL Client\_Reservation\_Insert(4, 2);  INSERT INTO Add\_Services (Name\_Add\_Services, Cost\_Add\_Services) VALUES  ('Завтрак', 300),  ('Обед', 350);  CALL Add\_Services\_Insert('Ужин', 320);  CALL Add\_Services\_Insert('Парковка', 500);  INSERT INTO Add\_Services\_Reservation (Add\_Services\_ID, Reservation\_ID) VALUES  (1, 1),  (4, 1),  (1, 2),  (2, 2),  (4, 3);  CALL Add\_Services\_Reservation\_Insert(3, 4);  CALL Add\_Services\_Reservation\_Insert(4, 4);  CALL Add\_Services\_Reservation\_Insert(1, 5);  CALL Add\_Services\_Reservation\_Insert(4, 5);  INSERT INTO Employee (Surname\_Employee, Name\_Employee, Patronymic\_Employee, Login\_Employee, Password\_Employee) VALUES  ('Иванов', 'Иван', 'Иванович', 'wk\_IvanovII', 'Pa$$w0rd'),  ('Петров', 'Пётр', 'Петрович', 'wk\_PetrovPP', 'Pa$$w0rd');  CALL Employee\_Insert('Андреев', 'Андрей', 'Андреевич', 'wk\_AndreevAA', 'Pa$$w0rd');  INSERT INTO Contract (Number\_Contract, Reservation\_ID, Employee\_ID, Creation\_Date\_Contract, Creation\_Time\_Contract) VALUES  ('ДГН-00000001', 1, 1, '2023-09-05', '13:40:12'),  ('ДГН-00000002', 2, 2, '2023-09-03', '13:32:45');  CALL Contract\_Insert('ДГН-00000003', 3, 2, '2023-09-03', '13:50:43'); |

1. Перезапись структуры;

Таблица 2 – Файл перезаписи структуры БД

|  |  |
| --- | --- |
| Название файла | Скрипт |
| Re-Create.sql | -- Отмена разрешений для rl\_employee  REVOKE EXECUTE ON PROCEDURE Room\_Configuration\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Room\_Configuration\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Room\_Configuration\_Delete FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Room\_Options\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Room\_Options\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Room\_Options\_Delete FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Reservation\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Reservation\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Reservation\_Delete FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Client\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Client\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Client\_Delete FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Client\_Reservation\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Client\_Reservation\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Client\_Reservation\_Delete FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Delete FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Contract\_Insert FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Contract\_Update FROM rl\_employee;  REVOKE EXECUTE ON PROCEDURE Contract\_Delete FROM rl\_employee;  -- Отмена разрешений для rl\_administrator  REVOKE EXECUTE ON PROCEDURE Class\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Class\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Class\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Status\_room\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Status\_room\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Status\_room\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Configuration\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Configuration\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Configuration\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Options\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Options\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Options\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Employee\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Employee\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Employee\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Configuration\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Configuration\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Configuration\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Options\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Options\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Room\_Options\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Reservation\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Reservation\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Reservation\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Client\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Client\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Client\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Client\_Reservation\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Client\_Reservation\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Client\_Reservation\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Add\_Services\_Reservation\_Delete FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Contract\_Insert FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Contract\_Update FROM rl\_administrator;  REVOKE EXECUTE ON PROCEDURE Contract\_Delete FROM rl\_administrator;  -- Отзыв прав для роли rl\_Client  REVOKE SELECT ON Class FROM rl\_Client;  REVOKE SELECT ON Status\_Room FROM rl\_Client;  REVOKE SELECT ON Room FROM rl\_Client;  REVOKE SELECT ON Configuration FROM rl\_Client;  REVOKE SELECT ON Room\_Configuration FROM rl\_Client;  REVOKE SELECT ON Options FROM rl\_Client;  REVOKE SELECT ON Room\_Options FROM rl\_Client;  REVOKE SELECT ON Reservation FROM rl\_Client;  REVOKE SELECT ON Client\_Reservation FROM rl\_Client;  REVOKE SELECT ON Add\_Services FROM rl\_Client;  REVOKE SELECT ON Add\_Services\_Reservation FROM rl\_Client;  REVOKE SELECT ON Contract FROM rl\_Client;  -- Отзыв прав для роли rl\_Employee  REVOKE SELECT, INSERT, UPDATE ON Room\_Configuration FROM rl\_Employee;  REVOKE SELECT, INSERT, UPDATE ON Room\_Options FROM rl\_Employee;  REVOKE SELECT, INSERT, UPDATE ON Reservation FROM rl\_Employee;  REVOKE SELECT, INSERT, UPDATE ON Client FROM rl\_Employee;  REVOKE SELECT, INSERT, UPDATE ON Client\_Reservation FROM rl\_Employee;  REVOKE SELECT, INSERT, UPDATE ON Add\_Services\_Reservation FROM rl\_Employee;  REVOKE SELECT, INSERT, UPDATE ON Contract FROM rl\_Employee;  REVOKE SELECT ON Class FROM rl\_Employee;  REVOKE SELECT ON Status\_room FROM rl\_Employee;  REVOKE SELECT ON Room FROM rl\_Employee;  REVOKE SELECT ON Configuration FROM rl\_Employee;  REVOKE SELECT ON Options FROM rl\_Employee;  REVOKE SELECT ON Add\_Services FROM rl\_Employee;  REVOKE SELECT ON Employee FROM rl\_Employee;  -- Отзыв прав для роли rl\_Administrator  REVOKE SELECT, INSERT, UPDATE, DELETE ON Class FROM rl\_Administrator;  REVOKE SELECT, INSERT, UPDATE, DELETE ON Status\_room FROM rl\_Administrator;  REVOKE SELECT, INSERT, UPDATE, DELETE ON Room FROM rl\_Administrator;  REVOKE SELECT, INSERT, UPDATE, DELETE ON Configuration FROM rl\_Administrator;  REVOKE SELECT, INSERT, UPDATE, DELETE ON Options FROM rl\_Administrator;  REVOKE SELECT, INSERT, UPDATE, DELETE ON Add\_Services FROM rl\_Administrator;  REVOKE SELECT, INSERT, UPDATE, DELETE ON Employee FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Room\_Configuration FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Room\_Options FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Reservation FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Client FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Client\_Reservation FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Add\_Services\_Reservation FROM rl\_Administrator;  REVOKE SELECT, DELETE ON Contract FROM rl\_Administrator;  -- Удаление индексов  DROP INDEX IF EXISTS Index\_ID\_Class;  DROP INDEX IF EXISTS Index\_Name\_Status\_Room;  DROP INDEX IF EXISTS Index\_ID\_Status\_Room;  DROP INDEX IF EXISTS Index\_ID\_Room;  DROP INDEX IF EXISTS Index\_Number\_Room;  DROP INDEX IF EXISTS Index\_ID\_Configuration;  DROP INDEX IF EXISTS Index\_ID\_Room\_Configuration;  DROP INDEX IF EXISTS Index\_ID\_Options;  DROP INDEX IF EXISTS Index\_ID\_Room\_Options;  DROP INDEX IF EXISTS Index\_ID\_Reservation;  DROP INDEX IF EXISTS Index\_Number\_Reservation;  DROP INDEX IF EXISTS Index\_ID\_Client;  DROP INDEX IF EXISTS index\_surname\_client;  DROP INDEX IF EXISTS index\_name\_client;  DROP INDEX IF EXISTS index\_patronymic\_client;  DROP INDEX IF EXISTS index\_document\_series\_client;  DROP INDEX IF EXISTS index\_document\_number\_client;  DROP INDEX IF EXISTS index\_date\_of\_birth\_client;  DROP INDEX IF EXISTS index\_gender\_client;  DROP INDEX IF EXISTS index\_issued\_by\_client;  DROP INDEX IF EXISTS index\_issued\_date\_client;  DROP INDEX IF EXISTS index\_code\_department\_client;  DROP INDEX IF EXISTS Index\_Phone\_Client;  DROP INDEX IF EXISTS Index\_Email\_Client;  DROP INDEX IF EXISTS Index\_Login\_Client;  DROP INDEX IF EXISTS Index\_ID\_Client\_Reservation;  DROP INDEX IF EXISTS Index\_ID\_Add\_Services;  DROP INDEX IF EXISTS Index\_ID\_Add\_Services\_Reservation;  DROP INDEX IF EXISTS Index\_ID\_Employee;  DROP INDEX IF EXISTS Index\_Login\_Employee;  DROP INDEX IF EXISTS Index\_ID\_Contract;  DROP INDEX IF EXISTS Index\_Number\_Contract;  -- Удаление таблиц  DROP TABLE IF EXISTS Contract;  DROP TABLE IF EXISTS Add\_Services\_Reservation;  DROP TABLE IF EXISTS Client\_Reservation;  DROP TABLE IF EXISTS Reservation;  DROP TABLE IF EXISTS Room\_Options;  DROP TABLE IF EXISTS Options;  DROP TABLE IF EXISTS Room\_Configuration;  DROP TABLE IF EXISTS Configuration;  DROP TABLE IF EXISTS Room;  DROP TABLE IF EXISTS Status\_Room;  DROP TABLE IF EXISTS Class;  DROP TABLE IF EXISTS Client;  DROP TABLE IF EXISTS Add\_Services;  DROP TABLE IF EXISTS Employee; |

1. Создание резервной копии;



1. Версия базы данных:
   1. Отчёт о созданных объектах и записях;

Таблица 3 – Перечень созданных объектов

|  | Информация по объектам |
| --- | --- |
| Запрос | select  information\_schema.tables.table\_name as "Название таблиц",  string\_agg(information\_schema.routines.routine\_name, ', ') as "Список процедур",  n\_live\_tup as "Кол-во записей в таблицах"  from information\_schema.tables  inner join information\_schema.routines  on information\_schema.tables.table\_name = substring(information\_schema.routines.routine\_name, 1, length(information\_schema.tables.table\_name))  inner join pg\_stat\_user\_tables  on information\_schema.tables.table\_name = pg\_stat\_user\_tables.relname  where  information\_schema.tables.table\_schema = 'public'and  routine\_type = 'PROCEDURE'  group by  table\_name,  n\_live\_tup  union all  select  'Количество процедур: ',  count(information\_schema.routines.routine\_name)::text,  (select  sum(n\_live\_tup)  from pg\_stat\_user\_tables)  from information\_schema.routines  where  routine\_type = 'PROCEDURE' and  routine\_name not in ('structure\_create','structure\_re\_create'); |
| Результат локальной БД |  |
| Результат удалённой БД |  |

* 1. Версия БД.

Таблица 4 – Версия файла БД

|  |  |
| --- | --- |
| Параметры | PostgreSQL |
| Номер версии | 2.2.0.1 |
| Что сделано | * Созданы 42 хранимые процедуры; * Добавлено 124 строк во всю структуру БД; * Произведено распределение доступа ролей к хранимым процедурам; * Сопровождены процедуры, для быстрого создания и перезаписи всей структуры таблиц, индексов, прав доступа, заполнения данных, создания хранимых процедур; * Создан Backup файл. |