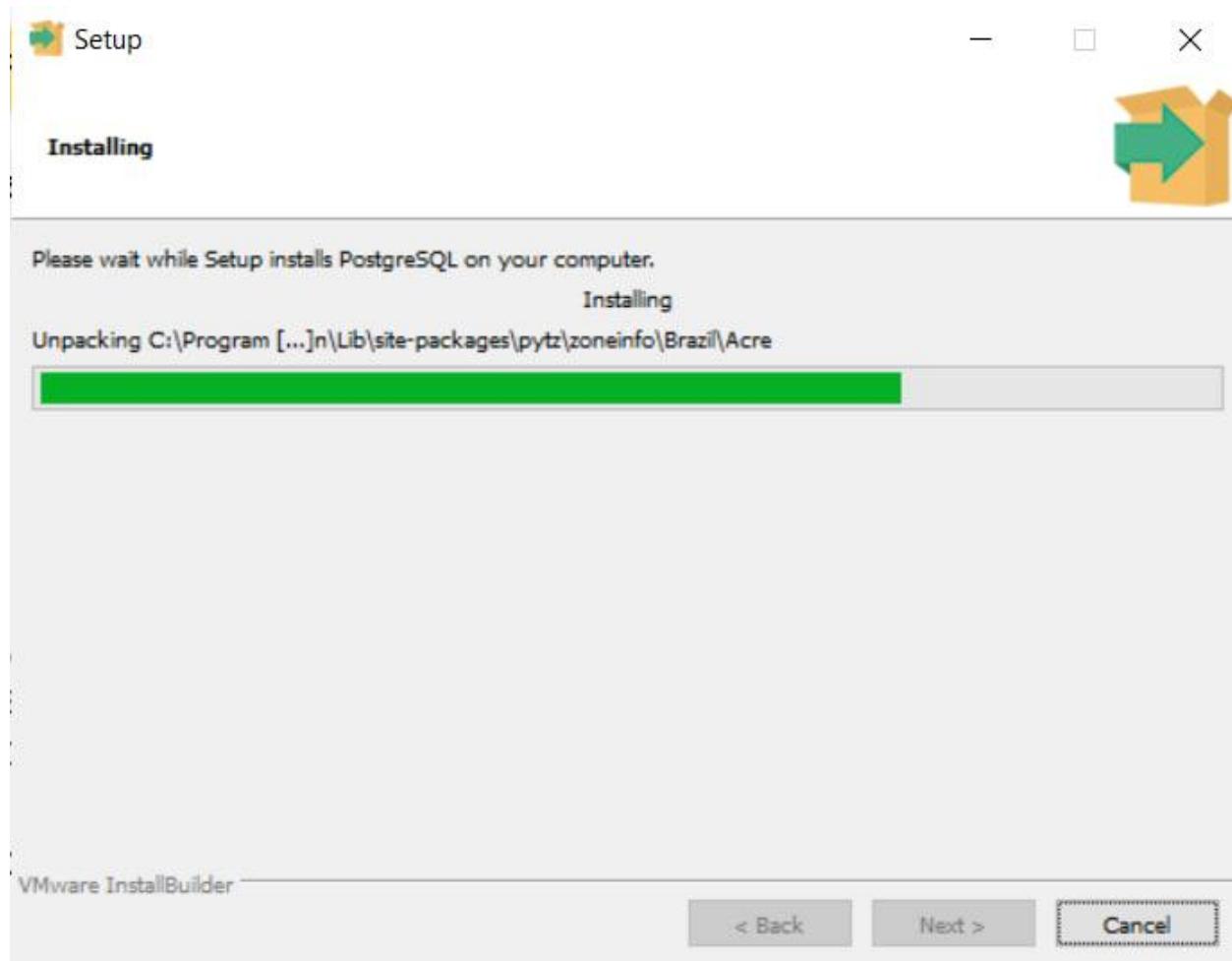


Отчет по практическим работам 1-3

1. Устанавливаем PostgreSQL и pgAdmin в OC Windows



2. Запускаем SQL Shell и создаем базу данных “hotel_reason”

```
SQL Shell (psql)
template1 | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |          | libc          | postgres=CTc/postgres
          |          |           +-----+-----+
          |          |          +-----+-----+
          |          |          postgres=CTc/postgres

(3 ёёёюш)

postgres=# CREATE DATABASE hotel_reason;
ОШИБКА: ошибка синтаксиса (примерное положение: "/")
СТРОКА 2: /1
^

postgres=# CREATE DATABASE hotel_reason;
CREATE DATABASE
postgres=# clear
postgres=# /c
postgres=# \c
Вы подключены к базе данных "postgres" как пользователь "postgres".
postgres=# \l
              Табличка срч фрээвї
   №   | Тырфхыхў |  Йошхёютър |    LC_COLLATE    |    LC_CTYPE     |  йыөрьын# ICU |  Ієотршхё  йыөрьыш |    Ієртп  фюёесяр
---+-----+-----+-----+-----+-----+-----+-----+
hotel_reason | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |          | libc          | postgres=CTc/postgres
postgres     | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |          | libc          | postgres=CTc/postgres
template0    | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |          | libc          | postgres=CTc/postgres
template1    | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |          | libc          | postgres=CTc/postgres
          |          |           +-----+-----+
          |          |          +-----+-----+
          |          |          postgres=CTc/postgres
          |          |          +-----+-----+
          |          |          postgres=CTc/postgres

(4 ёёёюш)

postgres=#

```

3. В базе данных ‘hotel_reason’ создаем таблицу ‘guest_log’. В ней содержатся текущие сведения о номере(кто проживает, кто обслуживает, дата заселения/выселения). В таблице создадим столбцы id, resident_id, room_id, check_in_date, check_out_date. Resident_id и room_id являются указателями на id в таблицах ‘resident’ и ‘rooms’

```
CREATE TABLE
hotel=# CREATE TABLE guest_log (
hotel(# id SERIAL PRIMARY KEY,
hotel(# resident_id INTEGER REFERENCES residents(id),
hotel(# room_id INTEGER REFERENCES rooms(id),
hotel(# check_in_date DATE,
hotel(# check_out_date DATE
hotel(# );
CREATE TABLE
```

4. Создаем таблицу ‘rooms’, содержащую общие сведения о номере. В данной таблице создаем столбцы: id, room_type(тип номера), num_rooms, num_beds, price_per_night(цена за одну ночь)

```
hotel=# CREATE TABLE rooms (
hotel(# id SERIAL PRIMARY KEY,
hotel(# room_type VARCHAR(50),
hotel(# num_rooms INTEGER,
hotel(# num_beds INTEGER,
hotel(# price_per_night NUMERIC(10, 2)
hotel(# );
CREATE TABLE
```

5. Создаем таблицу ‘employees’, содержащую информацию об обслуживающем персонале, в ней создадим столбцы id, first_name, last_name, position, hire_date, phone, email. Для столбцов first_name, last_name, position, phone и email тип поля укажем VARCHAR(строка ограниченной длины), столбец hire_date будет с типом DATE, id имеет такой же тип как и идентификаторы в остальных таблицах.

```

hotel=# CREATE TABLE employees (
hotel(# id SERIAL PRIMARY KEY,
hotel(# first_name VARCHAR(50),
hotel(# last_name VARCHAR(50),
hotel(# position VARCHAR(100),
hotel(# hire_date DATE,
hotel(# phone VARCHAR(20),
hotel(# email VARCHAR(100)
hotel(# );
CREATE TABLE

```

6. Создаем таблицу ‘residents’, хранящую в себе сведения о постояльце. Таблица состоит из следующих строк: id, first_name, last_name, date_of_birth, phone, email. Столбцы first_name, last_name, phone и email идентичны таблице ‘employees’, date_of_birth имеет тип DATE, id по аналогии с остальными таблицами.

```

hotel=# CREATE TABLE residents (
hotel(# id SERIAL PRIMARY KEY,
hotel(# first_name VARCHAR(50),
hotel(# last_name VARCHAR(50),
hotel(# date_of_birth DATE,
hotel(# phone VARCHAR(20),
hotel(# email VARCHAR(100)
hotel(# );
CREATE TABLE

```

7. Составим схему базы данных

