МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»

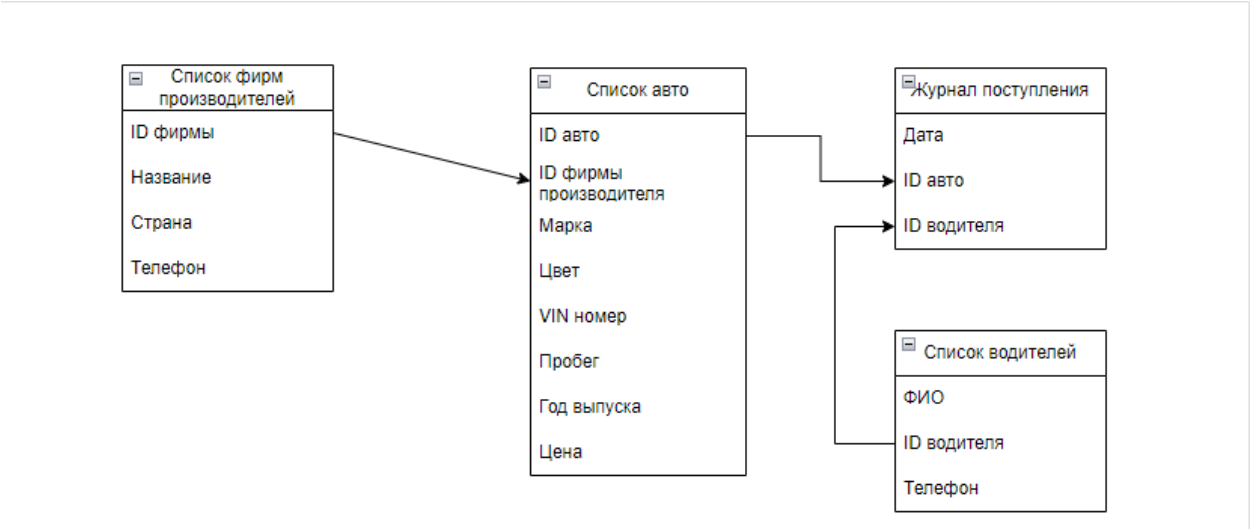
Факультет Управление и информатика в технологических системах
Кафедра Информационная безопасность
Специальность 10.05.03 «Информационная безопасность автоматизированных систем»

Создание базы данных в postgresql

Выполнил студент гр. УБ-01
Цой Кирилл Вадимович

Воронеж – 2023

Модель базы данных



Проектирование базы данных «Магазин по продаже автомобилей»

Для создания новой базы используем оператор CREATE DATABASE. Чтобы убедится в её создании используем команду \l и посмотрим список баз данных:

```
postgres-# \l
                                                  Список баз данных
      Имя       | Владелец | Кодировка |     LC_COLLATE     |     LC_CTYPE       | локаль ICU | Провайдер локали |     Права доступа
----------------+----------+-----------+--------------------+--------------------+------------+------------------+-----------------------
 CarSellingShop | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |            | libc             |
 postgres       | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |            | libc             |
 template0      | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |            | libc             | =c/postgres          +
                |          |           |                    |                    |            |                  | postgres=CTc/postgres
 template1      | postgres | UTF8      | Russian_Russia.1251 | Russian_Russia.1251 |            | libc             | =c/postgres          +
                |          |           |                    |                    |            |                  | postgres=CTc/postgres
(4 строки)


postgres-#
```

Для подключения к созданной базе используем команду \connect

```
postgres-# \connect CarSellingShop
Вы подключены к базе данных "CarSellingShop" как пользователь "postgres".
CarSellingShop-#
```

Создание таблиц

Для создания таблиц в базе данных используется оператор CREATE TABLE. Создадим следующие таблицы: company(фирмы производители), auto(автомобили), arrival_log(журнал поступления), drivers(водители, пригнавшие авто)

Основная функция этого оператора – создание новой таблицы и описание ее столбцов и типов данных. Кроме того, этот оператор позволяет определять первичные ключи, альтернативные ключи и внешние ключи с некоторыми ограничениями ссылочной целостности, а также задавать ограничения на столбцы и таблицы

Таблица company:

```
CREATE TABLE company(
id BIGSERIAL PRIMARY KEY NOT NULL,
name VARCHAR(50) NOT NULL,
country VARCHAR(50) NOT NULL,
phone bigint NOT NULL);
```

```
CarSellingShop=# \d company
                                    Таблица "public.company"
 Столбец |          Тип          | Правило сортировки | Допустимость NULL |            По умолчанию
---------+-----------------------+--------------------+-------------------+------------------------------------
 id      | bigint                |                    | not null          | nextval('company_id_seq'::regclass)
 name    | character varying(50) |                    | not null          |
 country | character varying(50) |                    | not null          |
 phone   | integer               |                    | not null          |
Индексы:
    "company_pkey" PRIMARY KEY, btree (id)
```

Таблица auto:

```
CREATE TABLE auto(
id BIGSERIAL PRIMARY KEY NOT NULL,
id_company int REFERENCES company(id) NOT NULL,
brand VARCHAR(50) NOT NULL,
color VARCHAR(50) NOT NULL,
vin VARCHAR(17) NOT NULL,
mileage int NOT NULL,
year int NOT NULL,
price bigint NOT NULL);
```

```
CarSellingShop=# \d auto
                                      Таблица "public.auto"
  Столбец   |          Тип          | Правило сортировки | Допустимость NULL |            По умолчанию
------------+-----------------------+--------------------+-------------------+------------------------------------
 id         | bigint                |                    | not null          | nextval('auto_id_seq'::regclass)
 id_company | integer               |                    | not null          |
 brand      | character varying(50) |                    | not null          |
 color      | character varying(50) |                    | not null          |
 vin        | character varying(17) |                    | not null          |
 mileage    | integer               |                    | not null          |
 year       | integer               |                    | not null          |
 price      | integer               |                    | not null          |
Индексы:
    "auto_pkey" PRIMARY KEY, btree (id)
Ограничения внешнего ключа:
    "auto_id_company_fkey" FOREIGN KEY (id_company) REFERENCES company(id)
```

Таблица drivers:

```
CREATE TABLE drivers(
id BIGSERIAL PRIMARY KEY NOT NULL,
fio VARCHAR(17) NOT NULL,
phone bigint NOT NULL);
```

```
CarSellingShop=# \d drivers
                                    Таблица "public.drivers"
 Столбец |          Тип          | Правило сортировки | Допустимость NULL |            По умолчанию
---------+-----------------------+--------------------+-------------------+------------------------------------
 id      | bigint                |                    | not null          | nextval('drivers_id_seq'::regclass)
 fio     | character varying(17) |                    | not null          |
 phone   | integer               |                    | not null          |
Индексы:
    "drivers_pkey" PRIMARY KEY, btree (id)
```

Таблица arrival_log:

```
CREATE TABLE arrival_log(
id_auto int REFERENCES auto(id) NOT NULL,
```

id_driver int REFERENCES drivers(id) NOT NULL,
date DATE NOT NULL,
time TIME NOT NULL);

```
CarSellingShop=# \d arrival_log
                          Таблица "public.arrival_log"
  Столбец  |          Тип           | Правило сортировки | Допустимость NULL | По умолчанию
-----------+------------------------+--------------------+-------------------+--------------
 id_auto   | integer                |                    | not null          |
 id_driver | integer                |                    | not null          |
 date      | date                   |                    | not null          |
 time      | time without time zone |                    | not null          |
Ограничения внешнего ключа:
    "arrival_log_id_auto_fkey" FOREIGN KEY (id_auto) REFERENCES auto(id)
    "arrival_log_id_driver_fkey" FOREIGN KEY (id_driver) REFERENCES drivers(id)
```

С помощью команды \d выведем список таблиц

```
CarSellingShop=# \d
                  Список отношений
 Схема  |      Имя       |        Тип         | Владелец
--------+----------------+--------------------+----------
 public | arrival_log    | таблица            | postgres
 public | auto           | таблица            | postgres
 public | auto_id_seq    | последовательность | postgres
 public | company        | таблица            | postgres
 public | company_id_seq | последовательность | postgres
 public | drivers        | таблица            | postgres
 public | drivers_id_seq | последовательность | postgres
(7 строк)
```

Заполнение таблиц

Используем команду INSERT INTO

Таблица company:

INSERT INTO company (name, country, phone) values ('BMW', 'Germany', '13407794456'), ('KIA', 'Korea', '2011805871'), ('Mazda', 'Japan', '10645239277 ');

Выведем таблицу используя оператора SELECT

```
CarSellingShop=# SELECT * FROM company;
 id | name  | country |    phone
----+-------+---------+-------------
  1 | BMW   | Germany | 13407794456
  2 | KIA   | Korea   |  2011805871
  3 | Mazda | Japan   | 10645239277
(3 строки)
```

Таблица auto:

INSERT INTO auto(id_company, brand, color, vin, mileage, year, price) values ('1', 'M8', 'black', '4A3AB76T68E011282', '20000', '2020', '15672002'), ('2', 'RIO', 'gray', '1GCHK23244F199207', '20000', '2010', '50000'), ('3', 'RX-8', 'red', 'JH4DA9440NS003801', '5000', '2022', '500000');

```
CarSellingShop=# SELECT * FROM auto;
 id | id_company | brand | color |        vin        | mileage | year |  price
----+------------+-------+-------+-------------------+---------+------+----------
  1 |          1 | M8    | black | 4A3AB76T68E011282 |   20000 | 2020 | 15672002
  2 |          2 | RIO   | gray  | 1GCHK23244F199207 |   20000 | 2010 |    50000
  3 |          3 | RX-8  | red   | JH4DA9440NS003801 |    5000 | 2022 |   500000
(3 строки)
```

Таблица drivers:
INSERT INTO drivers(fio, phone) values ('Chet Williamson', '8333'), ('Joe Barbaro', '8123'), ('Joel Miller', '123456');

```
CarSellingShop=# SELECT * FROM drivers;
 id |       fio       | phone
----+-----------------+--------
  1 | Chet Williamson |   8333
  2 | Joe Barbaro     |   8123
  3 | Joel Miller     | 123456
(3 строки)
```

Таблица arrival_log:
INSERT INTO arrival_log values ('1', '2', '21.10.2022', '17:14:52'), ('2', '3', '04.12.2011', '16:39:25'), ('3', '1', '07.02.2023', '08:36:43');

```
CarSellingShop=# SELECT * FROM arrival_log;
 id_auto | id_driver |    date    |   time
---------+-----------+------------+----------
       1 |         2 | 2022-10-21 | 17:14:52
       2 |         3 | 2011-12-04 | 16:39:25
       3 |         1 | 2023-02-07 | 08:36:43
(3 строки)
```

Чтение заданных строк из одиночной таблицы

Ранее рассмотренные SQL-запросы выбирали определенные столбцы всех строк таблицы. Теперь рассмотрим запросы, позволяющие выбирать столбцы определенных строк. Следующий запрос получает все столбцы из тех строк таблицы auto, в которых mileage больше 6000:
SELECT * FROM auto WHERE mileage > 6000

```
CarSellingShop=# SELECT * FROM auto WHERE mileage >  6000;
 id | id_company | brand | color |        vin         | mileage | year |  price
----+------------+-------+-------+--------------------+---------+------+----------
  1 |          1 | M8    | black | 4A3AB76T68E011282 |   20000 | 2020 | 15672002
  2 |          2 | RIO   | gray  | 1GCHK23244F199207 |   20000 | 2010 |    50000
(2 строки)
```