Московский государственный технический университет им.Н.Э.Баумана кафедра "Системы обработки информации и управления"



Постреляционные базы данных

к лабораторной работе №1

Лабораторная работа «Объектно-реляционная модель данных» по дисциплине «Постреляционные базы данных»

Инструктор: Мария Валерьевна

Email:2623859464@qq.com

Студент: Ван Чаочао

группа ИУ5И-22М

2022/03/01

Цель работы:

- 1. Изучить различные технологии работы с ОР СУБД.
- 2. Освоить языки и технологии SQL\PSM.
- 3. Получить навыки программирования на стороне сервера и на стороне клиента.

Средства выполнения:

```
ПаСУБД PostgreSQL
🖺 🗊 PgAdmin.
```

Литература:

Документация PostgreSQL - postgrespro.ru - Документация - PostgeSQL 9.4 -Серверное программирование

Время выполнения:

Время выполнения лабораторной работы 4 часа.

Пункты задания для выполнения:

Задание 1. Создание таблиц и ограничений (базовая часть)

Мая тема «Веб-сервис по продажам автомобилей », Поэтому необходимо создать следующие таблицы: Users Account orrder

(1): Create Table 创建表格

```
Query Editor Query History
```

```
1 CREATE TYPE address AS(
       city varchar(20),
3
       town varchar(20),
4
       street varchar(20),
5
       house varchar(20)
6
       );
7
8
9 CREATE TABLE users(
       userID SERIAL PRIMARY KEY,
10
       Firstname CHARACTER VARYING(40) NOT NULL,
11
       Lastname CHARACTER VARYING(40) NOT NULL,
12
13
       Email CHARACTER VARYING(40) NOT NULL,
       Telephone varchar(15)[],
14
       addr address,
15
16
       Birthday DATE,
17
       dat DATE[],
       UNIQUE (Email)
18
19);
20
```

```
21
    CREATE TABLE Account(
22
        accountID SERIAL PRIMARY KEY,
23
        userID INTEGER,
        Login CHARACTER VARYING(20) CHECK (login !=''),
25
        Pass CHARACTER VARYING(20) CHECK (login !=''),
        FOREIGN KEY (userID) REFERENCES users(userID) ON UPDATE CASCADE ON DELETE SET NULL,
26
27
        CONSTRAINT check_login UNIQUE(Login)
28 );
29
        create type OrderShippingType as enum ('MerchantDelivery','PersonalPickup');
30
31
        create type Vehicle_Brands as enum ('Benz', 'BMW', 'Geely');
32
    CREATE TABLE Orders(
        orderID SERIAL PRIMARY KEY,
33
34
        accountID INTEGER,
35
        shippingType OrderShippingType,
36
        brands Vehicle_Brands,
        FOREIGN KEY (accountID) REFERENCES Account(accountID) ON UPDATE CASCADE ON DELETE SET NULL,
        Ammount_items INTEGER DEFAULT 1 CHECK (Ammount_items>0 AND Ammount_items<3)
38
39 );
```

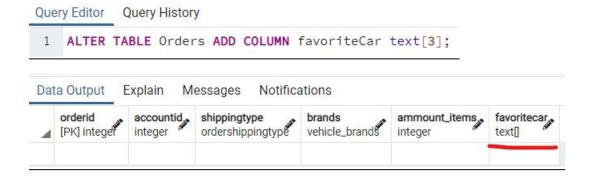
(2)Формат таблицы следующий



Задание 2. Сложные типы данных (хорошо)

(1) Добавить в таблицу поле типа массив (например, список преподавателей) и продемонстрировать:

Добавить поле типа массива 增加一个数组类型字段

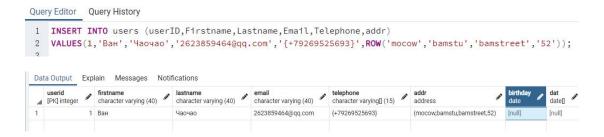


вставить данные в массив 向数组中插入数据

```
Query Editor Query History

1 insert into Orders(favoriteCar)
2 values ('{BMW}');
```

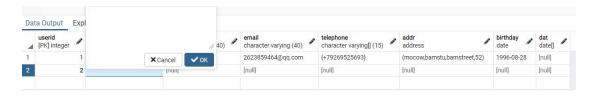
Входные данные 插入表格数据



обновить данные 更新某一行数据



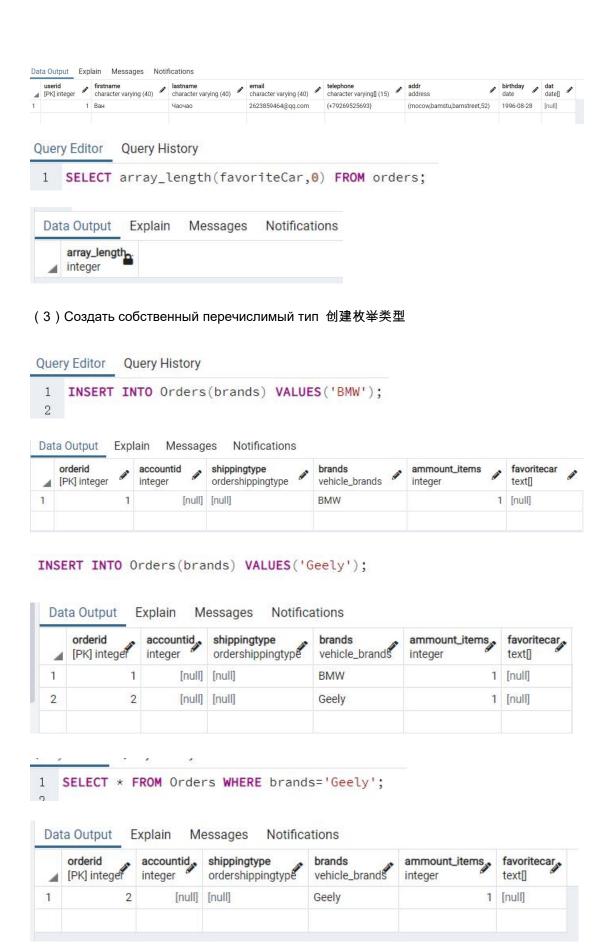
Добавить новую строку с помощью интерфейса



вывод строк таблицы, содержащих указанный элемент или набор элементов

按条件输出数据





Задание 3: Наследование таблиц (хорошо)

Создайте две таблицы

```
Query Editor Query History
    CREATE TABLE students(
1
2
        name text,
3
        age int,
        studentID int,
4
5
        profession text DEFAULT 'wy5'
6
    );
7
8
   CREATE TABLE foreignStudent(
9
        nation text
   ) INHERITS (students);
10
```

Введите информацию

Query Editor Query History

```
INSERT INTO students(name,age,studentID)

VALUES('BaH YaoYao',18,1),

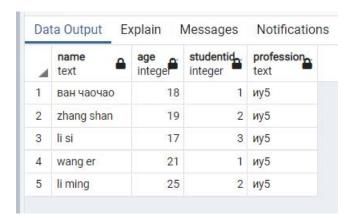
('zhang shan',19,2),

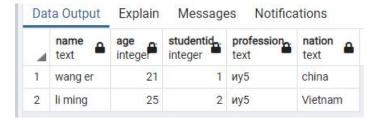
('li si',17,3);

INSERT INTO foreignStudent(name,age,studentID,nation)

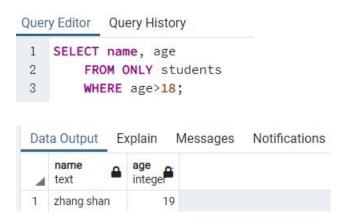
VALUES('wang er',21,1, 'china'),

('li ming ',25,2, 'Vietnam');
```





Запрос по условию





Дополнительное задание (отлично):

4 .Пользовательский тип (PostgreSQL)

Создать собственный скалярный тип, например, Параметры_вуза — Uparams(кол.НИИ, кол.факультетов, кол. студентов), определить для него функции сортировки, ввода и вывода. Добавить как поле в таблицу.

продемонстрировать работу с типом.

Учебная литература

1. Документация PostgreSQL [электронный ресурс]. – Режим доступа: https://postgrespro.ru/docs/postgresql/, свободный.