

**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:

СУБД PostgreSQL

PgAdmin.

Литература:

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

```

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

```

(2) Формат таблицы следующий

Data Output

Explain

Messages

Notifications

accountid

[PK] integer

userid

integer

login

character varying (20)

pass

character varying (20)

Data Output

Explain

Messages

Notifications

orderid

[PK] integer

accountid

integer

shippingtype

ordershippingtype

brands

vehicle_brands

ammount_items

integer

Data Output

Explain

Messages

Notifications

userid

[PK] integer

firstname

character varying (40)

lastname

character varying (40)

email

character varying (40)

telephone

character varying[] (15)

addr

address

birthday

date

dat

date[]

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

Query Editor

Query History

1

ALTER TABLE Orders ADD COLUMN favoriteCar text[3];

Data Output

Explain

Messages

Notifications

	orderid [PK] integer	accountid integer	shippingtype ordershippingtype	brands vehicle_brands	ammount_items integer	favoritecar text[]

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

Query Editor Query History

```
1 INSERT INTO users (userID,Firstname,Lastname,Email,Telephone,addr)
2 VALUES(1,'Ван','Чаочао','2623859464@qq.com','{+79269525693}',ROW('mccow','bamstu','bamstreet','52'));
```

Data Output		Explain	Messages	Notifications			
 userid [PK] integer 	firstname character varying (40)	 lastname character varying (40)	email character varying (40)	 telephone character varying[] (15) 	addr address	 birthday date 	dat date[] 
1	1 Ван	Чаочао	2623859464@qq.com	(+79269525693)	(mccow,bamstu,bamstreet,52)	[null]	[null]

обновить данные 更新某一行数据

Query Editor Query History

```
1 UPDATE users
2 SET Birthday='1996-08-28'
3 WHERE userID=1
```

Data Output		Explain	Messages	Notifications			
 userid [PK] integer 	firstname character varying (40) 	lastname character varying (40) 	email character varying (40) 	telephone character varying[] (15) 	addr address 	birthday date 	dat date[] 
1	1 Ван	Чаочао	2623859464@qq.com	{+79269525693}	(mccow,bamstu,bamstreet,52)	1996-08-28	[null]

Добавить новую строку с помощью интерфейса

Data Output		Explain						
	userid [PK] integer		40	email character varying (40)	telephone character varying[] (15)	addr address	birthday date	dat date[]
1	1			2623859464@qq.com	{+79269525693}	(mccow,bamstu,bamstreet,52)	1996-08-28	[null]
2	2			[null]	[null]	[null]	[null]	[null]

вывод строк таблицы, содержащих указанный элемент или набор элементов

按条件输出数据

Query Editor Query History

```
1 SELECT*FROM users WHERE userID=1
```

Data Output		Explain	Messages	Notifications				
 userid [PK] integer	 firstname character varying (40)	 lastname character varying (40)	 email character varying (40)	 telephone character varying (15)	 addr address	 birthday date	 dat date[]	
1	1	Ван	Чачао	2623859464@qq.com	(+79269525693)	(moco w ,bamstu,bamstreet,52)	1996-08-28	[null]

Query Editor Query History








```
1 SELECT array_length(favoriteCar,0) FROM orders;
```

Data Output	Explain	Messages	Notifications
<div><div></div><div>array_length</div><div>integer</div></div>			

(3) Создать собственный перечислимый тип 创建枚举类型

Query Editor Query History

```
1 INSERT INTO Orders(brands) VALUES('BMW');
2
```

Data Output		Explain	Messages	Notifications		
	orderid [PK] integer 	accountid integer 	shippingtype ordershippingtype 	brands vehicle_brands 	ammount_items integer 	favoritecar text[] 
1	1	[null]	[null]	BMW	1	[null]

```
INSERT INTO Orders(brands) VALUES('Geely');
```

Data Output		Explain	Messages	Notifications		
	orderid [PK] integer	accountid integer	shippingtype ordershippingtype	brands vehicle_brands	ammount_items integer	favoritecar text[]
1	1	[null]	[null]	BMW	1	[null]
2	2	[null]	[null]	Geely	1	[null]

```
1 SELECT * FROM Orders WHERE brands='Geely';
2
```

Data Output

Explain

Messages

Notifications

<div></div> <div>orderid</div> <div>[PK] integer</div>	<div></div> <div>accountid</div> <div>integer</div>	<div></div> <div>shippingtype</div> <div>ordershippingtype</div>	<div></div> <div>brands</div> <div>vehicle_brands</div>	<div></div> <div>ammount_items</div> <div>integer</div>	<div></div> <div>favoritecar</div> <div>text[]</div>
1	2	[null]	[null]	Geely	1 [null]

Задание 3: Наследование таблиц (хорошо)

Создайте две таблицы

```

Query Editor  Query History
1  CREATE TABLE students(
2      name text,
3      age int,
4      studentID int,
5      profession text DEFAULT 'иу5'
6  );
7
8  CREATE TABLE foreignStudent(
9      nation text
10 ) INHERITS (students);

```

Введите информацию

```

Query Editor  Query History
1  INSERT INTO students(name,age,studentID)
2  VALUES('ван чаочао',18,1),
3  ('zhang shan',19,2),
4  ('li si',17,3);
5
6  INSERT INTO foreignStudent(name,age,studentID,nation)
7  VALUES('wang er',21,1, 'china'),
8  ('li ming ',25,2, 'Vietnam');

```

Data Output	Explain	Messages	Notifications
name text	age integer	studentid integer	profession text
1 ван чаочао	18	1	иу5
2 zhang shan	19	2	иу5
3 li si	17	3	иу5
4 wang er	21	1	иу5
5 li ming	25	2	иу5

Data Output		Explain	Messages	Notifications	
	name text	age integer	studentid integer	profession text	nation text
1	wang er	21	1	иу5	china
2	li ming	25	2	иу5	Vietnam

Запрос по условию

Query Editor

Query History

1 SELECT name, age

2 FROM ONLY students

3 WHERE age>18;

Data Output

Explain

Messages

Notifications

	name text	age integer	
1	zhang shan	19	

Query Editor

Query History

1

2

3

```
SELECT p.relname, c.name,c.age
      FROM students c, pg_class p
      WHERE c.age>18 AND c.tableoid=p.oid;
```

Data Output

Explain

Messages

Notifications

	relname name	name text	age integer	
1	students	zhang shan	19	
2	foreignstudent	wang er	21	
3	foreignstudent	li ming	25	

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.