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



Постреляционные базы данных

к лабораторной работе №2

Лабораторная работа «Объектное расширение языка SQL» 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 постреляционные возможности языка SQL.
- 2. Освоить языки и технологии SQL\PSM.
- 3. Получить навыки программирования на стороне сервера.

Средства выполнения:

- 1. СУБД PostgreSQL
- 2. PgAdmin

Литература:

- 1. Документация PostgreSQL postgrespro.ru Документация PostgeSQL 9.4 Серверное программирование.
- 2. Методические рекомендации.

Время выполнения:

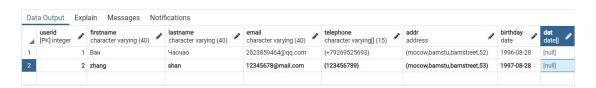
Время выполнения лабораторной работы 4 часа.

Пункты задания для выполнения:

Задание 1. Базовая часть (удовлетворительно)

1.1. Создание и заполнение таблицы (创建和填写表格)

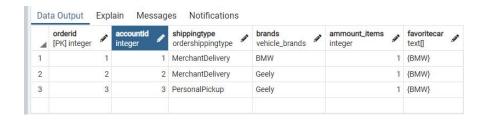
Users table(用户表)



Account table(账户表)



Oreders table(订单表)



1.2. Программирование функций с применением SQL\PSM

1) Создать скалярную функцию (创建一个标量函数)

```
CREATE OR REPLACE FUNCTION get_users_max_age(u_id integer)

RETURNS integer AS

$$

DECLARE

max_age integer;

BEGIN

SELECT age INTO max_age

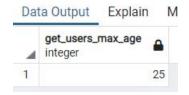
FROM users WHERE userid=u_id;

RETURN max_age;

END

$$
```

SELECT * FROM get users max age(2);



LANGUAGE plpgsql;

Select * FROM users WHERE userid=1 and

age>get_users_max_age(3);



2) Создать табличную функцию (创建一个表函数)

CREATE OR REPLACE FUNCTION users name(us id integer)

RETURNS TABLE(userid integer, firstname varchar,

lastname varchar) AS

'SELECT userid, firstname, lastname

FROM users

WHERE userid=us_id'

LANGUAGE SQL

ROWS 10;

ALTER FUNCTION users_name(us_id integer)

OWNER TO postgres;

SELECT * FROM users_name(1);



SELECT * FROM users WHERE userid IN (SELECT userid FROM users name(1)

WHERE lastname='Yaoyao');



3) Создать хранимую процедуру (创建一个存储过程)

CREATE OR REPLACE PROCEDURE increase_table(tablename varchar(100))

AS \$\$

BEGIN

EXECUTE format('

CREATE TABLE IF NOT EXISTS %I(

id integer PRIMARY KEY,

brand text,

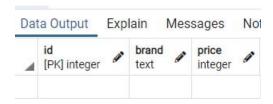
price integer)', tablename);

END

\$\$ LANGUAGE 'plpgsql';

CALL increase_table('One MoreTable');

SELECT * FROM "One MoreTable";



Задание 2. Расширенная часть (хорошо)

2.1. Извлечение части записей и результатов запросов на изменение данных

(提取数据变更请求的部分记录和结果)

SELECT * FROM users LIMIT 2;

| 4 | 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[] | age integer |
|---|------------------------|----------------------------------|---------------------------------|------------------------------|---------------------------------------|--------------|------------------|------------|----------------|
| 1 | 1 | Ван | Чаочао | 2623859464@qq.com | {+79269525693} | (mocow,ba | 1996-08-28 | [null] | 26 |
| 2 | 2 | zhang | shan | 12345678@mail.com | {123456789} | (mocow,ba | 1997-08-28 | [null] | 25 |

INSERT INTO account(accountid,userid,login,pass)

VALUES(4,4,'wcc690485','wcc690485')

RETURNING *;

| 4 | accountid [PK] integer | userid integer | login character varying (20) | pass character varying (20) | 4 |
|---|---------------------------|-------------------|---------------------------------|--------------------------------|---|
| 1 | 4 | 4 | wcc690485 | wcc690485 | |

UPDATE account

SET login='wcc690486'

WHERE userid=4

RETURNING *;

| Dat | a Output | Exp | lain M | ess | ages Notifications | | | |
|-----|---------------------------|-----|-------------------|-----|---------------------------------|---|--------------------------------|---|
| 4 | accountid [PK] integer | d' | userid integer | 4 | login character varying (20) | 4 | pass character varying (20) | 4 |
| 1 | | 4 | | 4 | wcc690486 | | wcc690485 | |
| | | | | | | | | |

2.2. Выполнение рекурсивных запросов (执行递归查询)

WITH RECURSIVE ord AS(

SELECT orderid, brands, ammount_items

FROM orders

WHERE orderid=2

UNION

SELECT dd.orderid,

dd.brands,

dd.ammount_items

FROM ord d

INNER JOIN orders dd ON dd.orderid>d.ammount_items

) select *from ord;

| Dat | a Output | Explain Mes | sag | es Notifications | 5 |
|-----|--------------------|--------------------------|---------|-----------------------|---|
| 4 | orderid integer | brands vehicle_brands | <u></u> | ammount_items integer | 2 |
| 1 | 2 | Geely | | | 1 |
| 2 | 3 | Geely | | | 1 |

2.3. Создание динамических запросов (创建动态查询)

```
CREATE OR REPLACE PROCEDURE copy_users(user_lastname varchar(40))
AS $$

DECLARE

users_id "users".userid%TYPE;

BEGIN
```

SELECT "userid" INTO users_id FROM users

WHERE "lastname"=user_lastname;

IF users_id IS NULL THEN

RAISE EXCEPTION USING errcode='E0001',

hint='no exit this user',

message='Вид спорта с title=' || \$1 || ' не найден!';

ELSE

EXECUTE format('

CREATE TABLE IF NOT EXISTS %I(

userid integer PRIMARY KEY,

lastname varchar(40)

)', user_lastname);

EXECUTE format('DELETE FROM %I', user_lastname);

EXECUTE format('INSERT INTO %I

SELECT "userid", "lastname"

FROM users WHERE "userid" = %L', \$1, users id); END IF; **END** \$\$ LANGUAGE 'plpgsql'; CALL copy_users('Yaoyao'); select * from "Чаочао"; Data Output Explain Messages userid lastname character varying (40) [PK] integer 1 1 Чаочао CALL copy_users('Y'); select * from "4"; ERROR: 错误: Вид спорта с title=Ч не найден! HINT: no exit this user CONTEXT: 在RAISE的第8行的PL/pgSQL函数copy_users(character varying) SQL state: E0001

Дополнительное задание (отлично):

3.1. Выполнение ранжирующих запросов

Продемонстрировать выполнение функций row_number(), Rank(), dense_rank(), ntile(4).

3.2. Программирование функции - агрегата

Создать функцию-агрегат, например, **MaxMin**(), которая применяется к целым числам и возвращает соотношение максимальных и минимальных значений. Проверить работоспособность функции из окна запроса

3.3. Создание DDL триггера

Создать DDL триггер (например, на удаление таблиц, начинающихся с SYS), который пишет в журнал сведения о событии и делает откат транзакции. Проверить работу триггера.