

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Базы данных

Отчет по лабораторной работе №5
Хранимые процедуры

Работу

выполнил:

Графов Д.И.

Группа: 33531/2

Преподаватель:

Мяснов А.В.

Санкт-Петербург
2019

Содержание

1. Цель работы	3
2. Программа работы	3
3. Теоретическая информация	3
3.1. PL/pgSQL	3
3.2. Хранимая процедура	3
4. Ход работы	4
4.1. Первое индивидуальное задание	4
4.2. Второе индивидуальное задание	5
5. Выводы	6

1. Цель работы

Познакомить студентов с возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур.

2. Программа работы

- Изучение возможностей языка PL/pgSQL.
- Создание двух хранимых процедур в соответствии с индивидуальным заданием, полученным у преподавателя.
- Выкладывание скрипта с созданными сущностями в репозиторий.
- Демонстрация результатов преподавателю.

3. Теоретическая информация

3.1. PL/pgSQL

PL/pgSQL (англ. Procedural Language/PostGres Structured Query Language) — процедурное расширение языка SQL, используемое в СУБД PostgreSQL. Этот язык предназначен для написания функций, триггеров и правил и обладает следующими особенностями:

- добавляет управляющие конструкции к стандарту SQL;
- допускает сложные вычисления;
- может использовать все объекты БД, определенные пользователем;
- прост в использовании.

3.2. Хранимая процедура

Хранимая процедура — объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. Хранимые процедуры очень похожи на обыкновенные процедуры языков высокого уровня, у них могут быть входные и выходные параметры и локальные переменные, в них могут производиться числовые вычисления и операции над символьными данными, результаты которых могут присваиваться переменным и параметрам. В хранимых процедурах могут выполняться стандартные операции с базами данных (как DDL, так и DML). Кроме того, в хранимых процедурах возможны циклы и ветвления, то есть в них могут использоваться инструкции управления процессом исполнения.

4. Ход работы

4.1. Первое индивидуальное задание

Для заданного бара и даты процедура выводит список доступных позиций на тот момент в баре. Считается, что товар, поставленный более, чем 2 недели назад, уже был распродан.

Мной была написана функция `get_available(integer, timestamp)`, принимающая 2 параметра:

- `place_id integer` - для указания заведения (PK таблицы `places`);
- `from_date timestamp` - для указания даты.

Функция возвращает таблицу, содержащую информацию о доступных позициях, а также даты поставки данных позиций. Запрос, формирующий указанную таблицу, можно разделить на 2 части. Первая часть возвращает все поставки напитков из заданного заведения и за указанный промежуток времени. Вторая часть возвращает все поставки еды с теми же критериями. Затем, с помощью оператора `union`, полученные подвыборки объединяются и возвращаются в качестве результата выполнения функции.

Листинг 1: `sp1.sql`

```
1 create or replace function get_available(place_id integer, from_date timestamp)
2     returns TABLE
3     (
4         title          varchar(50),
5         rating         double precision,
6         volume         double precision,
7         average_price  double precision,
8         date           timestamp
9     )
10 as
11 $$
12 declare
13     req_time          date = from_date - interval '2_weeks';
14     req_place_id      int  = place_id;
15 begin
16     return query (select d.title, d.rating, d.volume, d.average_price, sd.date
17                   from drinks d
18                   join supplies_drinks sd on d.drink_id = sd.drink_id
19                   where sd.date > req_time
20                   and sd.place_id = req_place_id
21                   union
22                   select f.title, f.rating, f.volume, f.average_price, sf.date
23                   from food f
24                   join supplies_food sf on f.food_id = sf.food_id
25                   where sf.date > req_time
26                   and sf.place_id = req_place_id);
27 end ;
28 $$ language plpgsql;
29
30 select *
31 from get_available(15, (now() - interval '1_year')::timestamp);
```

	title	rating	volume	average_price	date
1	mMvh5w75wm28L1ndoLVw0lq9wJ6CGAqSpUjF5WRsG	1.5140333224061493	379	984	2018-10-09 05:35:58.716014
2	mijZofJKFq2NE2cU0RzsDkvsMcs96XXkkAdGK8iDF6bgP	3.342862368748877	632	372	2018-08-18 05:49:18.648438
3	0c	1.054002017946354	637	276	2018-10-05 11:41:57.698395
4	ib0y0qJeb3BdqR7Lpf	7.018519309634206	958	182	2018-07-04 14:30:03.590140
5	PoKI3y1Z5GYFHHf5w7vSuUjvJooNHJNUwfdM5WG	7.871898747658989	604	226	2019-02-20 16:34:10.957894
6	3XgmB58bdwXacLj f6FoxAoewOCN50xHK4tJzYx53grh1svp	3.4490248491797635	434	748	2018-12-11 11:16:17.103691
7	NGfYI8CA0RW6PTjZ3A0aJ	8.702769677728073	725	302	2018-11-08 11:04:21.983794
8	kqHBxDFTfs9uXWTS7fTRRQlqX860eyILQSS	1.4227847314297328	763	114	2018-10-12 01:26:53.545878
9	hr2Bv90qMHEzMnaQso0v1ntT10Fy9N	6.430829774749123	697	352	2019-06-09 23:00:32.927827
10	WnYz8fuMSxsqsU	8.711525665938913	714	712	2018-07-18 01:04:59.045372
11	HaALY5A15	1.2748355019196898	453	987	2018-06-28 11:07:31.355443
12	VtCtJKexQ3q7R0B4FFyeel0neRdNtIZys6Zo	6.324498026418478	701	764	2018-07-24 20:19:07.478555
13	i6J2epRgFclyIovLqajLYEc5lvAirLZE6eaU5HiaTgWlp	1.4766636389049537	491	197	2019-06-16 23:23:45.148882
14	QzNSQtZTUVebF3tT4AYKIRM390BSr0FpDo4ZrmnGI0ZSwIv	5.415115330918825	571	936	2018-06-15 10:44:39.802926

Рисунок 4.1. Результат работы

4.2. Второе индивидуальное задание

Для заданного временного интервала процедура выводит перечень акций в каждом из баров в формате json:

```

1 {
2   "bar_title": [
3     {
4       "weekday": day,
5       "discount": description of discount ,
6     },
7     ...
8   ],
9   ...
10 }

```

Пояснение: на данный момент структура БД не позволяет указать временной интервал проведения акции, поэтому данная функция не имеет параметров.

Написанная функция `get_discounts()` создаёт таблицу `disc`, содержащую информацию о всех скидках во всех барах. Затем, с помощью операторов `json_object_agg` и `json_build_object`, из данных таблицы `disc` формируется результат функции в формате json.

Листинг 2: sp2.sql

```

1 create or replace function get_discounts() returns json
2 as
3 $$
4 declare
5     result json;
6 begin
7     drop table if exists disc;
8     create table if not exists disc as (select p.title , p.address , d.description
9     ↪ , d.weekday
10                                from discounts d
11                                join places p on d.place_id = p
12     ↪ .place_id);
13     result = (select json_object_agg(concat(disc.title , '_на_', disc.address),
14     ↪ json_build_object('weekday', disc.weekday,
15     ↪ 'discount', disc.description))
16     from disc);
17     return result;
18 end
19 $$ language plpgsql;
20
21 select *
22 from get_discounts();

```

```

❏ ▼ object {110}
❏ ▼ Контакт бар на пр. Просвещения, 25 {2}
❏   weekday : null
❏   discount : Цены для друзей на пиво!
❏ ▼ Контакт бар на ул. Садовая, 35 {2}
❏   weekday : null
❏   discount : Цены для друзей на пиво!
❏ ► Контакт бар на пр. Владимирский, 17 {2}
❏ ► Контакт бар на пр. Коломяжский, 15, к.2 {2}
❏ ► Контакт бар на пл. Стачек, 7, лит. А {2}
❏ ► Контакт бар на ул. Гаккелевская, 34 {2}
❏ ► Контакт бар на ул. Марата, 7 {2}
❏ ► Контакт бар на Средний пр-т. В0, 28 {2}
❏ ► Контакт бар на пр.Чернышевского 11/57 {2}
❏ ► Контакт бар на ул. Бухарестская, 74 {2}
❏ ▼ gecGMUis33 на lpkMunVbsthefuquoaе3UpGgBGTA {2}
❏   weekday : 4
❏   discount : jSYlY1Nf2k6leBz
❏ ▼ uk1P на PZDGJliiojGwv7Tdw8 {2}
❏   weekday : 2
❏   discount : hX60NTix9K9eJQwk9cu0592XAm02Qz40Lu8WY73GwjxeXq3n
❏ ▼ bafFxlUqcTydWBdqNM2uqmsprJ69vWuTMQFefz4R61 на 05qm5R6w3DSGyJJiodVDCKMALoSj7mYDSFVo7DDLiy7RVac {2}
❏   weekday : 7
❏   discount : R9JPN0DbSzzDwG0gxxLnR26YSX3zw06mCfInwk0XLHlZC
❏ ▼ 5853pFcIe7u на i12fQPCkyfQEiu50Lq6IqAVlSst74i2LKS0lttVuoQM {2}
❏   weekday : 1
❏   discount : h2GnLhfadklWzFEuBMeLPR3xEjccQUvs
❏ ▼ U2zmN9tpXAtabaRnSbJ3sArD на 6tfunfPEECeD188R3fCM {2}
❏   weekday : 7
❏   discount : kH1Epw9jv5e0zR2
❏ ▼ Uh на YwcD19k36RNP {2}
❏   weekday : 2
❏   discount : 1i4r4vC40MfQNXjn28VvdrvhLL3AxCzjpn205h0Yrm1BXoZP

```

Рисунок 4.2. Результат работы

5. Выводы

В ходе данной работы я ознакомился с процедурным языком PL/pgSQL, изучил его особенности и возможности, которые он предоставляет. Во время выполнения индивидуальных заданий я получил практические навыки по реализации обработки данных на стороне сервера с помощью хранимых процедур. Кроме того, я научился выводить результаты запросов в формате json.