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

Отчёт по лабораторной работе

Дисциплина: Базы данных

Тема: SQL программирование, ХП

Выполнил студент гр. 43501/3 Крылов И.С.

Преподаватель Мяснов А.В

«_____» _____2018 г.

Санкт-Петербург
2018

1 Цель работы.

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

2 Программа работы

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

3 Ход работы

В соответствии с индивидуальным заданием были разработаны две хранимые процедуры.

База была заполненна тестовыми данными с помощью генератора.

3.1 Индивидальное задание

Реализовать хранимые процедуры:

1. Для входных параметров: игра-источник, игра-назначение, повышающий коэффициент подобрать системные требования (минимальные и рекомендуемые) для игры-назначения. При этом требования для CPU и GPU вычислять по количеству ядер и/или обьему памяти.
2. Выдать оценку динамики сложности игр одного производителя. Метрика сложности должна учитывать аппаратные и программные требования для игр.

3.2 Системные требования

Был разработан скрипт, содержащий функцию, которая заполнет системные требования для игры-назначения соответсвенно игре-источнику с соблюдением заданного коэффициентного отношения.

```
1 CREATE OR REPLACE FUNCTION gen_requirements(game_src_id INTEGER,
    game_dst_id INTEGER, coeff INTEGER)
2     RETURNS void
3 LANGUAGE SQL
```

```
4  AS $$
5
6  -- minimal reqs -
7  WITH src_game_min as (select id_game, cpu.manufacturer as c_man,
       cpu.model as c_mod, cpu.cores as c_cores, cpu.frequency as
       c_freq, cpu.RAM as ram FROM game join system_requirements
       using (id_system_requirements) join minimal_requirements using
        (id_minimal_requirements) join cpu using (id_cpu) where
       id_game = game_src_id)
8  INSERT INTO cpu VALUES ((SELECT MAX(id_cpu) FROM cpu) + 1, (
       SELECT c_man FROM src_game_min ), (SELECT c_mod FROM
       src_game_min), (SELECT c_cores FROM src_game_min) * coeff, (
       SELECT c_freq FROM src_game_min), (SELECT ram FROM
       src_game_min) * coeff);
9
10 WITH src_game_min as (select id_game, gpu.manufacturer as g_man,
       gpu.model as g_mod, gpu.memory as g_mem FROM game join
       system_requirements using (id_system_requirements) join
       minimal_requirements using (id_minimal_requirements) join gpu
       using (id_gpu) where id_game = game_src_id)
11 INSERT INTO gpu VALUES ((SELECT MAX(id_gpu) FROM gpu) + 1, (
       SELECT g_man FROM src_game_min ), (SELECT g_mod FROM
       src_game_min), (SELECT g_mem FROM src_game_min) * coeff);
12
13 WITH src_game_min as (select id_game, OS.manufacturer as os_man,
       OS.version as os_v, OS.bits as os_b FROM game join
       system_requirements using (id_system_requirements) join
       minimal_requirements using (id_minimal_requirements) join OS
       using (id_OS) where id_game = game_src_id)
14 INSERT INTO OS VALUES ( (SELECT MAX(id_OS) FROM OS) + 1, (SELECT
       os_v FROM src_game_min), (SELECT os_b FROM src_game_min), (
       SELECT os_man FROM src_game_min) );
15
16 INSERT INTO minimal_requirements VALUES ( (SELECT MAX(
       id_minimal_requirements) FROM minimal_requirements)+1, (SELECT
        id_OS FROM OS ORDER BY id_OS DESC LIMIT 1), (SELECT id_cpu
       FROM cpu ORDER BY id_cpu DESC LIMIT 1), (SELECT id_gpu FROM
       gpu ORDER BY id_gpu DESC LIMIT 1) );
17
18 -- optimal requirements -
19 WITH src_game_opt as (select id_game, cpu.manufacturer as c_man,
       cpu.model as c_mod, cpu.cores as c_cores, cpu.frequency as
       c_freq, cpu.RAM as ram FROM game join system_requirements
       using (id_system_requirements) join optimal_requirements using
        (id_optimal_requirements) join cpu using (id_cpu) where
       id_game = game_src_id)
20 INSERT INTO cpu VALUES ((SELECT MAX(id_cpu) FROM cpu) + 1, (
       SELECT c_man FROM src_game_opt ), (SELECT c_mod FROM
       src_game_opt), (SELECT c_cores FROM src_game_opt) * coeff, (
       SELECT c_freq FROM src_game_opt), (SELECT ram FROM
       src_game_opt) * coeff);
21
```

```
22 WITH src_game_opt as (select id_game,  gpu.manufacturer as g_man,
       gpu.model as g_mod, gpu.memory as g_mem FROM game join
   system_requirements using (id_system_requirements) join
   optimal_requirements using (id_optimal_requirements) join gpu
   using (id_gpu) where id_game = game_src_id)
23 INSERT INTO gpu VALUES ((SELECT MAX(id_gpu) FROM gpu) + 1, (
   SELECT g_man FROM src_game_opt), (SELECT g_mod FROM
   src_game_opt), (SELECT g_mem FROM src_game_opt) * coeff);
24
25 WITH src_game_opt as (select id_game, OS.manufacturer as os_man,
   OS.version as os_v, OS.bits as os_b FROM game join
   system_requirements using (id_system_requirements) join
   optimal_requirements using (id_optimal_requirements) join OS
   using (id_OS) where id_game = game_src_id)
26 INSERT INTO OS VALUES ( (SELECT MAX(id_OS) FROM OS) + 1, (SELECT
   os_v FROM src_game_opt),  (SELECT os_b FROM src_game_opt), (
   SELECT os_man FROM src_game_opt) );
27
28 INSERT INTO optimal_requirements VALUES ( (SELECT MAX(
   id_optimal_requirements) FROM optimal_requirements) + 1, (
   SELECT id_OS FROM OS ORDER BY id_OS DESC LIMIT 1), (SELECT
   id_cpu FROM cpu ORDER BY id_cpu DESC LIMIT 1), (SELECT id_gpu
   FROM gpu ORDER BY id_gpu DESC LIMIT 1) );
29
30 -- system requirements -
31 WITH sys_req AS (SELECT id_game, system_requirements.OS as os,
   system_requirements.memory_space as mem FROM game join
   system_requirements using (id_system_requirements) where
   id_game = game_src_id )
32 INSERT INTO system_requirements VALUES ( (SELECT MAX(
   id_system_requirements) FROM system_requirements) + 1, (SELECT
    os FROM sys_req), (SELECT mem FROM sys_req), (SELECT
   id_minimal_requirements FROM minimal_requirements ORDER BY
   id_minimal_requirements DESC LIMIT 1), 1, (SELECT
   id_optimal_requirements FROM optimal_requirements ORDER BY
   id_optimal_requirements DESC LIMIT 1) );
33
34 -- update game reqs -
35 UPDATE game SET id_system_requirements = (SELECT
   id_system_requirements FROM system_requirements ORDER BY
   id_system_requirements DESC LIMIT 1) where id_game =
   game_dst_id;
36 $$;
37
38 SELECT gen_requirements(5, 12, 2);
```

Поочередно заполнялись соответсвующие поля таблиц сначала для минимальных, затем для оптимальных требований. Конечной записью было добавление внешенего ключа на систменые требования для игры-назначения.

3.3 Динамика сложности

Был разработан скрипт, содержащий функцию выполняющую поставленную задачу.

```
CREATE OR REPLACE FUNCTION dev_dif_dynamics(dev_id integer)
RETURNS TABLE (game_name varchar, rel_date date, id_dev integer,
   difficulty bigint)
LANGUAGE SQL
AS $$

SELECT name, release_date, id_developer, cpu.cores * cpu.
   frequency * cpu.ram * gpu.memory * (SELECT COUNT(*) FROM game
   join system_requirements using (id_system_requirements) join
   preinstalled_software using (id_system_requirements) WHERE
   id_developer = dev_id ) as difficulty FROM game join
   system_requirements using (id_system_requirements) join
   minimal_requirements using (id_minimal_requirements) join cpu
   using (id_cpu) join gpu using (id_gpu) WHERE id_developer =
   dev_id ORDER BY release_date;

$$;

SELECT * FROM dev_dif_dynamics(3);
```

По заданному id разработчика формируется таблица с выводом суммарной сложности в хронологическом порядке выпуска игр.

4 Вывод

В ходе лабораторной работы были улучшены навыки работы с sql и получены навыки разработки хранимых функций. Хранимые функции не могу изменять данные и должны возвращать значение.

Вызов функции осуществляется там, где требуется выражение, формирующее значение. В связи с этим, функции могут непосредственно использоваться в выражениях. Эти качества позволяют в значительной степени расширить функциональные возможности языка SQL, как средства разработки приложений.