Санкт-Петербургский Политехнический университет Петра Великого

Институт компьютерных наук и технологий

Кафедра Компьютерных Систем и Программных Технологий

**ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №4**

Дисциплина: **Базы данных**

# Тема: Хранимые процедуры

А.М. Кириллов

А.В. Мяснов

Выполнил студент гр. № 43501/1

Преподаватель

Санкт-Петербург

2016

# Цель работы

# Ознакомиться возможностями реализации более сложной обработки данных на стороне сервера с помощью хранимых процедур.

# Программа работы

1. Изучить возможности языка PSQL
2. Создать две хранимые процедуры в соответствии с **индивидуальным заданием**, полученным у преподавателя
3. Выложить скрипт с созданными сущностями в svn
4. Продемонстрировать результаты преподавателю

# Ход работы

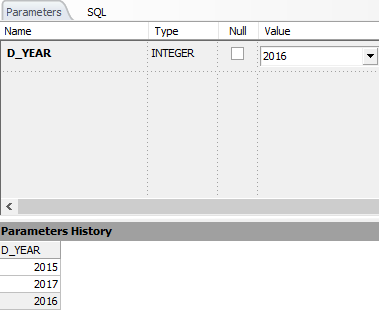
# Индивидуальное задание:

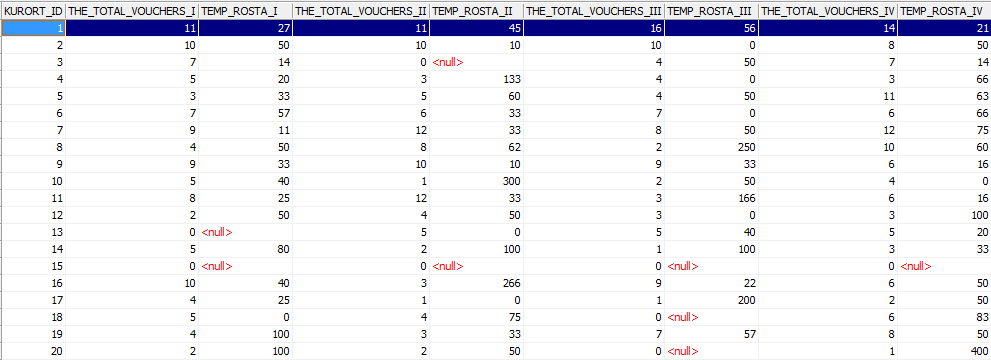
1. Для каждого курорта вывести для заданного года суммарный объем купленных путевок поквартально и отличие от аналогичного периода годом ранее в %.

|  |
| --- |
| 1. begin 2. for select kurort\_id from kurort order by kurort\_id into :kurort\_id do begin 3. kol\_buy\_this\_year\_I=0; 4. kol\_buy\_last\_year\_I=0; 5. kol\_buy\_this\_year\_II=0; 6. kol\_buy\_last\_year\_II=0; 7. kol\_buy\_this\_year\_III=0; 8. kol\_buy\_last\_year\_III=0; 9. kol\_buy\_this\_year\_IV=0; 10. kol\_buy\_last\_year\_IV=0; 11. KOL\_YEAR\_1=0; 12. KOL\_YEAR\_2=0; 13. KOL\_YEAR\_3=0; 14. KOL\_YEAR\_4=0; 15. for select id from tour where(id\_kurort=:kurort\_id) order by id into :tourID 16. do 17. begin 18. for select start\_time from trip 19. where (id\_tour=:tourID)and((extract(year from start\_time)=:d\_year)or(extract(year from start\_time)=:d\_year-1)) 20. into :buy\_date do 21. if (extract(year from buy\_date)=:d\_year) then begin 22. if ((extract(month from buy\_date)>=1)and(extract(month from buy\_date)<=3)) 23. then 24. begin kol\_buy\_this\_year\_I=kol\_buy\_this\_year\_I+1; 25. end 26. if ((extract(month from buy\_date)>=4)and(extract(month from buy\_date)<=6)) 27. then 28. begin kol\_buy\_this\_year\_II=kol\_buy\_this\_year\_II+1; 29. end 30. if ((extract(month from buy\_date)>=7)and(extract(month from buy\_date)<=9)) 31. then 32. begin kol\_buy\_this\_year\_III=kol\_buy\_this\_year\_III+1; 33. end 34. if ((extract(month from buy\_date)>=10)and(extract(month from buy\_date)<=12)) then 35. begin kol\_buy\_this\_year\_IV=kol\_buy\_this\_year\_IV+1; 36. end end else begin 37. if ((extract(month from buy\_date)>=1)and(extract(month from buy\_date)<=3)) 38. then 39. begin kol\_buy\_last\_year\_I=kol\_buy\_last\_year\_I+1; 40. end 41. if ((extract(month from buy\_date)>=4)and(extract(month from buy\_date)<=6)) 42. then 43. begin kol\_buy\_last\_year\_II=kol\_buy\_last\_year\_II+1; 44. end 45. if ((extract(month from buy\_date)>=7)and(extract(month from buy\_date)<=9)) 46. then 47. begin kol\_buy\_last\_year\_III=kol\_buy\_last\_year\_III+1; 48. end 49. if ((extract(month from buy\_date)>=10)and(extract(month from buy\_date)<=12)) then 50. begin kol\_buy\_last\_year\_IV=kol\_buy\_last\_year\_IV+1; 51. end 52. end 53. end 54. the\_total\_vouchers\_I=kol\_buy\_this\_year\_I; 55. the\_total\_vouchers\_II=kol\_buy\_this\_year\_II; 56. the\_total\_vouchers\_III=kol\_buy\_this\_year\_III; 57. the\_total\_vouchers\_IV=kol\_buy\_this\_year\_IV; 58. IF(kol\_buy\_this\_year\_i>kol\_buy\_last\_year\_i) THEN 59. BEGIN 60. KOL\_YEAR\_1=kol\_buy\_this\_year\_i-kol\_buy\_last\_year\_i; end 61. ELSE begin KOL\_YEAR\_1=kol\_buy\_last\_year\_i-kol\_buy\_this\_year\_i; end 62. IF(kol\_buy\_this\_year\_ii>kol\_buy\_last\_year\_ii) THEN 63. BEGIN 64. KOL\_YEAR\_2=kol\_buy\_this\_year\_ii-kol\_buy\_last\_year\_ii; end 65. ELSE begin KOL\_YEAR\_2=kol\_buy\_last\_year\_ii-kol\_buy\_this\_year\_ii; end 66. IF(kol\_buy\_this\_year\_iii>kol\_buy\_last\_year\_iii) THEN 67. BEGIN 68. KOL\_YEAR\_3=kol\_buy\_this\_year\_iii-kol\_buy\_last\_year\_iii; end 69. ELSE begin KOL\_YEAR\_3=kol\_buy\_last\_year\_iii-kol\_buy\_this\_year\_iii; end 70. IF(kol\_buy\_this\_year\_iv>kol\_buy\_last\_year\_iv) THEN 71. BEGIN 72. KOL\_YEAR\_4=kol\_buy\_this\_year\_iv-kol\_buy\_last\_year\_iv; end 73. ELSE begin KOL\_YEAR\_4=kol\_buy\_last\_year\_iV-kol\_buy\_this\_year\_iv; end 74. if(kol\_buy\_this\_year\_I<>0) then 75. begin temp\_rosta\_I=kol\_year\_1\*100/kol\_buy\_this\_year\_I; 76. end else begin 77. temp\_rosta\_I=NULL; end 78. if(kol\_buy\_this\_year\_II<>0) then begin 79. temp\_rosta\_II=kol\_year\_2\*100/kol\_buy\_this\_year\_II; end 80. else begin 81. temp\_rosta\_II=NULL; end 82. if(kol\_buy\_this\_year\_III<>0) then begin 83. temp\_rosta\_III=kol\_year\_3\*100/kol\_buy\_this\_year\_III; end 84. else begin 85. temp\_rosta\_III=NULL; end 86. if(kol\_buy\_this\_year\_IV<>0) then begin 87. temp\_rosta\_IV=kol\_year\_4\*100/kol\_buy\_this\_year\_IV; end 88. else begin 89. temp\_rosta\_IV=NULL; 90. end 91. suspend; 92. end 93. end |

Результат работы:

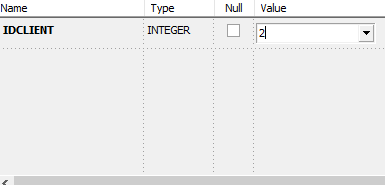
Выбираем год, который нужно проверить:



Результат работы:  


2.Вывести клиентов, у которых с каждой следующей поездкой увеличивалось количество достопримечательностей в курортах.

|  |
| --- |
| create or alter procedure CLIENTPLACE (  IDCLIENT integer)  returns (  CLIENTNAME varchar(10),  CLIENTID integer,  KOL\_TRIP integer,  KOL\_KURORT integer)  as  declare variable CURRENTV integer;  declare variable currtrip integer;  begin  kol\_kurort = 0;  for select id\_client, name from client, trip where (id\_client=:idclient)and  trip.id\_client=client.pasport\_id order by name into:clientid,:clientname do begin  for select count(\*) from trip where (id\_client=:idclient) order by count(\*) into: currtrip do begin  for select place\_of\_interest.id\_kurort from place\_of\_interest , kurort, tour, trip  where (place\_of\_interest.id\_kurort = kurort.kurort\_id)  and(kurort.kurort\_id=tour.id\_kurort)  and(tour.id=trip.id\_tour)and(id\_client=:idclient)  order by id\_kurort into:currentv do begin  while(:currentv > kol\_kurort) do  begin  kol\_kurort = :currentv;  kol\_trip =:currtrip;  end end end end suspend; end |





**Выводы**

В ходе выполнения данной лабораторной работы мы ознакомились с возможностями реализации обработки данных на стороне сервера с помощью хранимых процедур.

Отличие хранимых процедур от представлений состоит в том, что ХП представляют собой части программы, находящиеся и исполняющиеся непосредственно на сервере базы данных. Они хранятся вместе с базой на сервере в откомпилированном виде и позволяют перенести часто повторяющиеся длительные операции с клиентской машины на SQL-сервер.