Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет транспорта (МИИТ)» (РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

**О Т Ч Ё Т**

по лабораторной работе №3

по дисциплине:

«Информационное обеспечение систем управления»

на тему:

«Разработка запросов с помощью оконных функций»

Выполнил: ст. гр. ВУЦ-421

Лихачев С. С.

Вариант 5

Проверил: доц., к.т.н.

Васильева М. А.

Москва ­2024

**Цель работы**

Разработать запросы с помощью оконных функций на примере диалекта СУБД Postgres Pro.

**Результаты выполненной работы**

**Задание** **1**

Формулировка задания:

Запрос выдает нумерацию маршрутов в порядке убывания их длительности.

Текст скрипта:

**SELECT**

id **AS** *trail\_id*,

**name** **AS** *trail\_name*,

trail\_duration,

**ROW\_NUMBER**() **OVER** (**ORDER** **BY** trail\_duration **DESC**) **AS** *row\_num*

**FROM** trails;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 1).

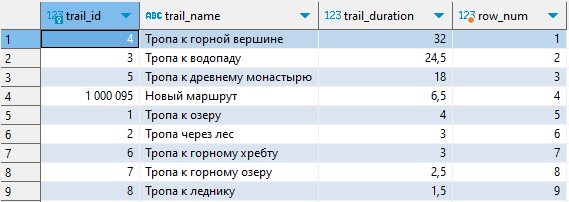


Рисунок 1 – Результат выполнения первого задания

**Задание** **2**

Формулировка задания:

Используется оконная функция для вычисления общей стоимости снаряжения в каждой категории.

Текст скрипта:

**SELECT**

et.**name** **AS** *equipment\_type*,

e.**name** **AS** *equipment\_name*,

*e*.price,

**SUM**(*e*.price) **OVER** (**PARTITION** **BY** et.**name**) **AS** total\_price\_per\_type

**FROM** equipment *e*

**JOIN** equipment\_type *et* **ON** *e*.equipment\_type\_id = *et*.id;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 2).

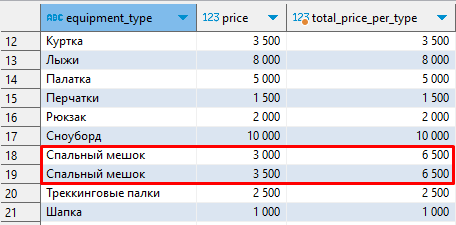


Рисунок 2 – Результат выполнения второго задания

**Задание** **3**

Формулировка задания:

Присваивает ранг маршрутам в зависимости от их перепада высот.

Текст скрипта:

**SELECT**

id **AS** *trail\_id*,

**name** **AS** *trail\_name*,

height\_difference,

**RANK**() **OVER** (**ORDER** **BY** height\_difference **DESC**) **AS** *rank\_height*

**FROM** trails;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 3).

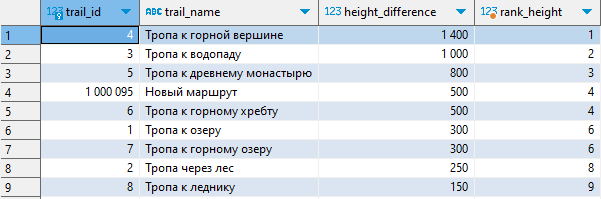


Рисунок 3 – Результат выполнения третьего задания

**Задание** **4**

Формулировка задания:

Ранжирование достижений на основе их популярности.

Текст скрипта:

**SELECT**

a.**name** **AS** *achievement\_name*,

**COUNT**(*ap*.person\_id) **AS** *person\_count*,

**DENSE\_RANK**() **OVER** (**ORDER** **BY** **COUNT**(*ap*.person\_id) **DESC**) **AS** *rank\_popularity*

**FROM** achievements *a*

**LEFT** **JOIN** achievements\_person *ap* **ON** *a*.id = *ap*.achievement\_id

**GROUP** **BY** *a*.id, a.**name**;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 4).

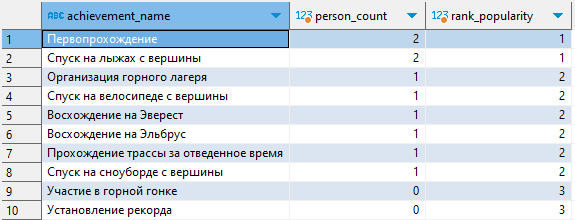


Рисунок 4 – Результат выполнения пятого задания

**Задание** **5**

Формулировка задания:

Запрос вычисляет среднюю стоимость снаряжения в пределах двух строк до и после текущей записи.

Текст скрипта:

**SELECT**

id **AS** *equipment\_id*,

**name** **AS** *equipment\_name*,

price,

**AVG**(price) **OVER** (**ORDER** **BY** id **ROWS** **BETWEEN** 2 **PRECEDING** **AND** 2 **FOLLOWING**) **AS** *avg\_price\_window*

**FROM** equipment;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 5).



Рисунок 5 – Результат выполнения пятого задания

**Задание** **6**

Формулировка задания:

Показывает предыдущее и следующее снаряжение в списке, отсортированном по стоимости.

Текст скрипта:

**SELECT**

id **AS** *equipment\_id*,

**name** **AS** *equipment\_name*,

price,

**LAG**(**name**) **OVER** (**ORDER** **BY** price) **AS** *previous\_equipment*,

**LEAD**(**name**) **OVER** (**ORDER** **BY** price) **AS** *next\_equipment*

**FROM** equipment;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 6).



Рисунок 6 – Результат выполнения шестого задания

**Задание** **7**

Формулировка задания:

Используется оконная функция для вычисления количества маршрутов, длительность которых больше и меньше текущей.

Текст скрипта:

**SELECT**

id **AS** *trail\_id*,

**name** **AS** *trail\_name*,

trail\_duration,

**COUNT**(\*) **OVER** (**ORDER** **BY** trail\_duration **ROWS** **BETWEEN** **UNBOUNDED** **PRECEDING** **AND** 1 **PRECEDING**) **AS** *trails\_shorter*,

**COUNT**(\*) **OVER** (**ORDER** **BY** trail\_duration **ROWS** **BETWEEN** 1 **FOLLOWING** **AND** **UNBOUNDED** **FOLLOWING**) **AS** *trails\_longer*

**FROM** trails;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 7).

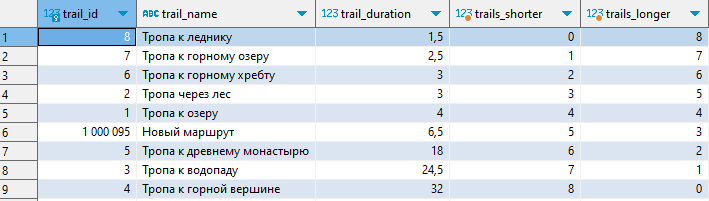


Рисунок 7 – Результат выполнения седьмого задания

**Задание** **8**

Формулировка задания:

Для каждого достижения показывает следующее достижение в алфавитном порядке.

Текст скрипта:

**SELECT**

id **AS** *achievement\_id*,

**name** **AS** *achievement\_name*,

**LEAD**(**name**) **OVER** (**ORDER** **BY** **name**) **AS** *next\_achievement*

**FROM** achievements;

Результат выполнения скрипта:

Результат выполнения скрипта представлен ниже (Рисунок 8).

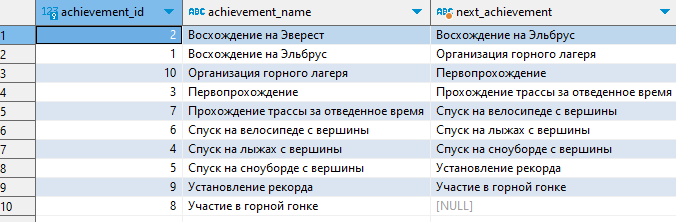


Рисунок 8 – Результат выполнения восьмого задания

Вывод: были разработаны запросы с помощью оконных функций на примере диалекта СУБД Postgres Pro.