

Домашнее задание №7

Цель задания: попрактиковаться с таблицами в GreenPlum.

a) Соединитесь с GreenPlum, используя инструмент Dbeaver

b) Создайте таблицу lab8_фамилия_1 с полями

- id1 int

- id2 int

- gen1 text

- gen2 text

c) Создайте первичный ключ (PRIMARY KEY) на основании комбинации полей id1, id2, gen1. Пожалуйста укажите в вашей таблице

○ Какой будет у вас DISTRIBUTION KEY таблицы?

○ Какая компрессия может использоваться в таблице?

```
CREATE TABLE lab8_pavlov_1 (
```

```
    id1 int,
```

```
    id2 int,
```

```
    gen1 text,
```

```
    gen2 text,
```

```
    PRIMARY KEY (id1, id2, gen1)
```

```
);
```

```
DISTRIBUTED BY (id1, id2, gen1);
```

d) Создайте таблицу lab8_фамилия_2 с такими же полями, как и у предыдущей таблицы, но

- храните таблицу колоночно и сожмите таблицу с помощью ZSTD уровня 1

- распределите таблицу по полю id2

```
CREATE TABLE lab8_pavlov_2 (
```

```
    id1 int4 NULL,
```

```
    id2 int4 NULL,
```

```
    gen1 text NULL,
```

```
    gen2 text NULL
```

```
)
```

```

WITH (
    appendonly=true,
    orientation=column,
    compressstype=zstd,
    compresslevel=1
)
DISTRIBUTED BY (id2);

```

е) Сгенерируйте данные для ваших таблиц на основании следующих скриптов

```

insert into lab8_pavlov_1 select gen,gen, gen::text || 'text1', gen::text || 'text2' from
generate_series(1,200000) gen;

insert into lab8_pavlov_2 select gen,gen, gen::text || 'text1', gen::text || 'text2' from
generate_series(1,200000) gen;

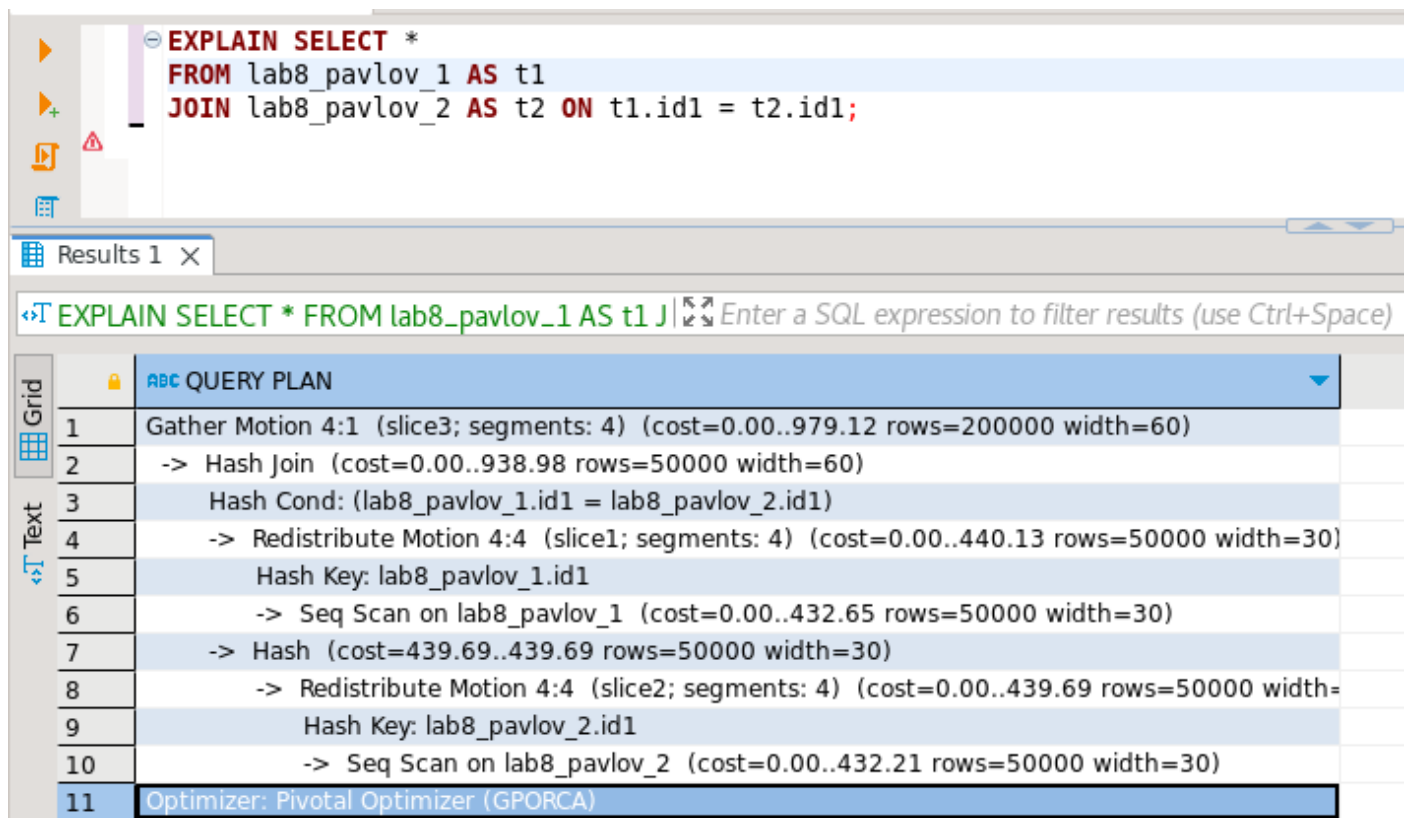
```

ф) С помощью директивы EXPLAIN просмотрите план соединения таблиц table1 и table2 по ключу id1.

```

EXPLAIN SELECT *
FROM lab8_pavlov_1 AS t1
JOIN lab8_pavlov_2 AS t2 ON t1.id1 = t2.id1;

```



The screenshot displays the SQL query and its execution plan in a database management system. The query is:

```
EXPLAIN SELECT *
FROM lab8_pavlov_1 AS t1
JOIN lab8_pavlov_2 AS t2 ON t1.id1 = t2.id1;
```

The execution plan, titled "ABC QUERY PLAN", consists of the following steps:

Step	Operation
1	Gather Motion 4:1 (slice3; segments: 4) (cost=0.00..979.12 rows=200000 width=60)
2	-> Hash Join (cost=0.00..938.98 rows=50000 width=60)
3	Hash Cond: (lab8_pavlov_1.id1 = lab8_pavlov_2.id1)
4	-> Redistribute Motion 4:4 (slice1; segments: 4) (cost=0.00..440.13 rows=50000 width=30)
5	Hash Key: lab8_pavlov_1.id1
6	-> Seq Scan on lab8_pavlov_1 (cost=0.00..432.65 rows=50000 width=30)
7	-> Hash (cost=439.69..439.69 rows=50000 width=30)
8	-> Redistribute Motion 4:4 (slice2; segments: 4) (cost=0.00..439.69 rows=50000 width=)
9	Hash Key: lab8_pavlov_2.id1
10	-> Seq Scan on lab8_pavlov_2 (cost=0.00..432.21 rows=50000 width=30)
11	Optimizer: Pivotal Optimizer (GPORCA)

g) Оптимизируйте ситуацию, попытавшись убрав REDISTRIBUTE MOTION

Распределение в таблицах по одинаковому полю id1 решило проблему

```
ALTER TABLE lab8_pavlov_1 SET DISTRIBUTED BY (id1);
```

```
ALTER TABLE lab8_pavlov_2 SET DISTRIBUTED BY (id1);
```

The screenshot shows a database query editor with the following SQL statement:

```
EXPLAIN SELECT *  
FROM lab8_pavlov_1 AS t1  
JOIN lab8_pavlov_2 AS t2 ON t1.id1 = t2.id1;
```

Below the query, the 'Results 1' tab displays the query plan:

	ABC QUERY PLAN
1	Gather Motion 4:1 (slice1; segments: 4) (cost=0.00..969.73 rows=200000 width=60)
2	-> Hash Join (cost=0.00..929.59 rows=50000 width=60)
3	Hash Cond: (lab8_pavlov_1.id1 = lab8_pavlov_2.id1)
4	-> Seq Scan on lab8_pavlov_1 (cost=0.00..432.65 rows=50000 width=30)
5	-> Hash (cost=432.21..432.21 rows=50000 width=30)
6	-> Seq Scan on lab8_pavlov_2 (cost=0.00..432.21 rows=50000 width=30)
7	Optimizer: Pivotal Optimizer (GPORCA)