**Task 18 (geqo)**

Папка задания в репозитории:

<https://github.com/artemgur/AdvancedPostgreSQL/tree/master/Task%2018%20(geqo)>

Для выполнения задания я использовал следующий запрос, содержащий 7 JOIN:

| SELECT \* FROM abc JOIN abc abc1 ON abc.a = abc1.b JOIN abc abc2 ON abc.b = abc2.c JOIN sb\_1000 ON abc.a = sb\_1000.c JOIN sb\_1000 s1 ON abc.c = s1.b JOIN task3 ON sb\_1000.a = task3.id JOIN sb\_1000 s2 ON abc2.b = s2.b JOIN task3 t ON abc.b = t.id LIMIT 50 |
| --- |

Этот SQL-запрос я запускал с помощью следующей команды PgBench:

| pgbench -U postgres -T 300 -nc 10 -f JoinQuery.sql |
| --- |

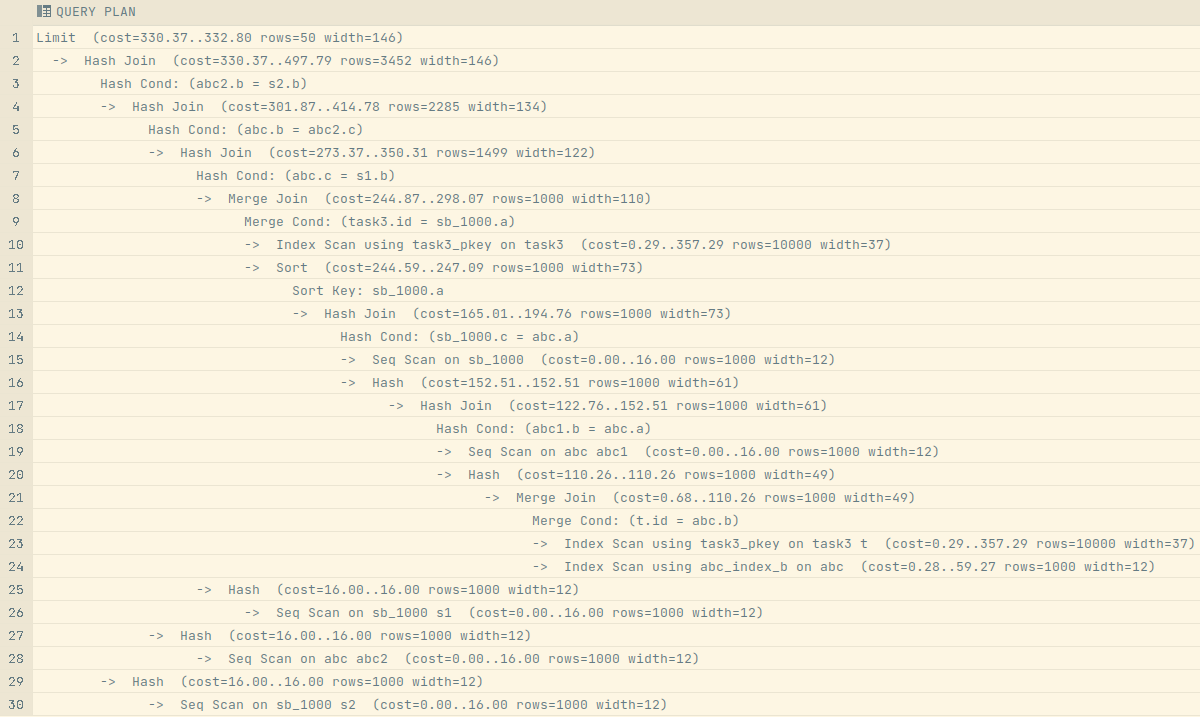
geqo по умолчанию включен.

При этом для того, чтобы geqo был использован для моего запроса, мне пришлось уменьшить значение параметра geqo\_threshold в postgresql.conf.

**Случай включенного geqo**

В случае включенного geqo я получил следующие результаты:

Дерево EXPLAIN:



Результаты PgBench:

* 211.291 ms — latency average
* 4.732819 — tps (including connections establishing)
* 4.733285 — tps (excluding connections establishing)

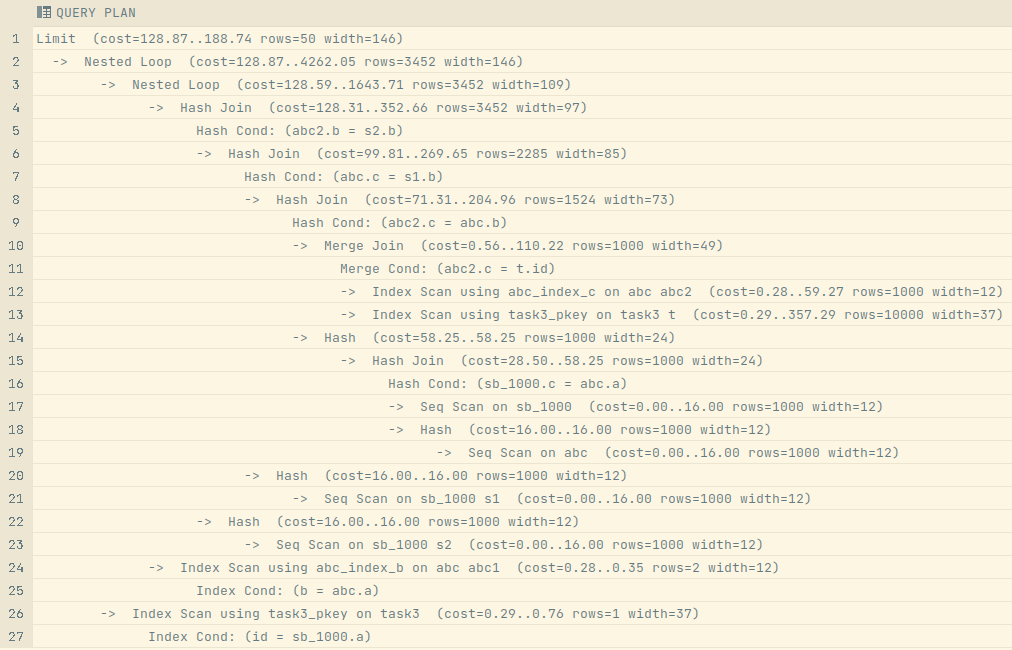
**Отключение geqo**

Чтобы отключить geqo, я изменил значение параметра geqo в файле postgresql.conf с on на off.

**Случай отключенного geqo**

В случае отключенного geqo я получил следующие результаты:

Дерево EXPLAIN:



Тот факт, что дерево EXPLAIN изменилось, показывает, что geqo действительно отключился.

Результаты PgBench:

* 545.972 ms — latency average
* 1.831594 — tps (including connections establishing)
* 1.831766 — tps (excluding connections establishing)

Таким образом, после отключения geqo запрос стал выполняться значительно медленнее.