

pg_repack: Handling table bloat in Terabytes of data

Artur Zakirov, Adjust GmbH
Berlin, February 2024

pg_repack use cases

- reduce bloat after resharding
- reduce bloat when VACUUM and autovacuum cannot handle it
- move tables to another tablespace (instead of ALTER TABLE ... SET TABLESPACE ...)

Database size

A database size: around 16 TB, NVMe + SSD

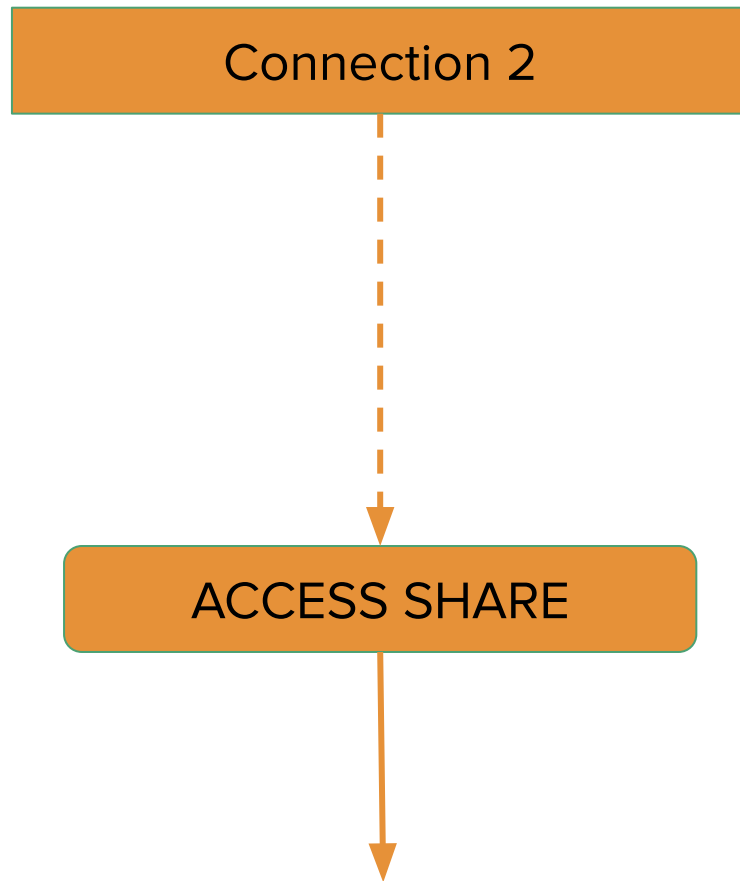
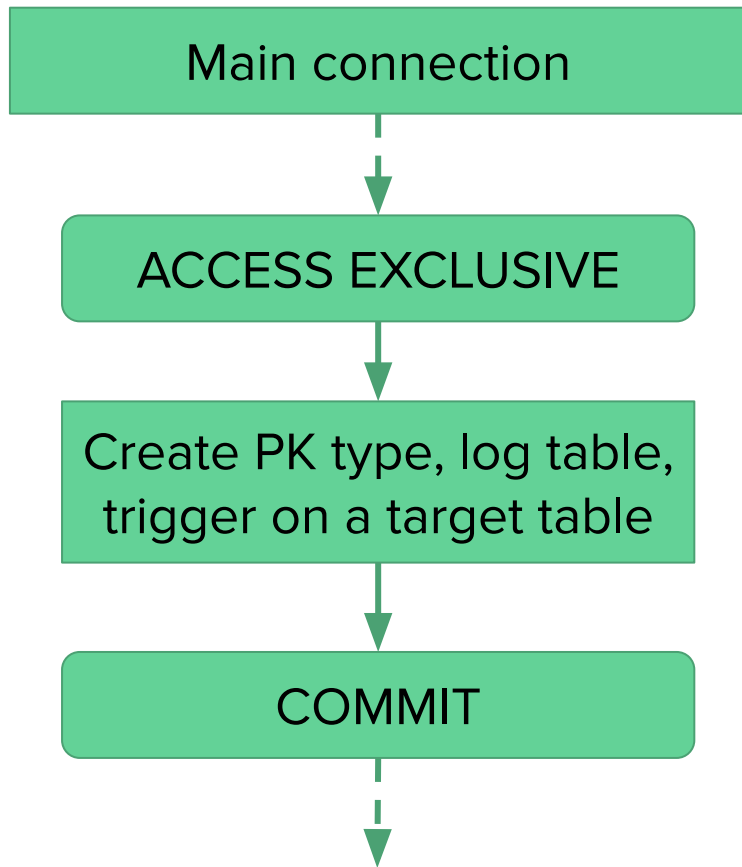
Number of partitions: around 16000

Size of a partition: up to 50 GB

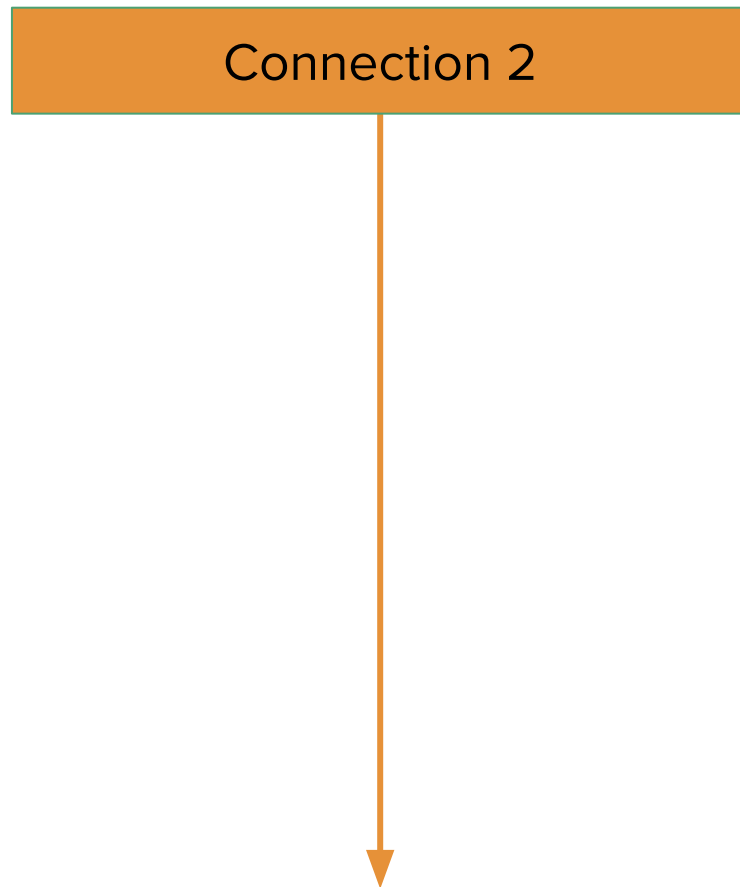
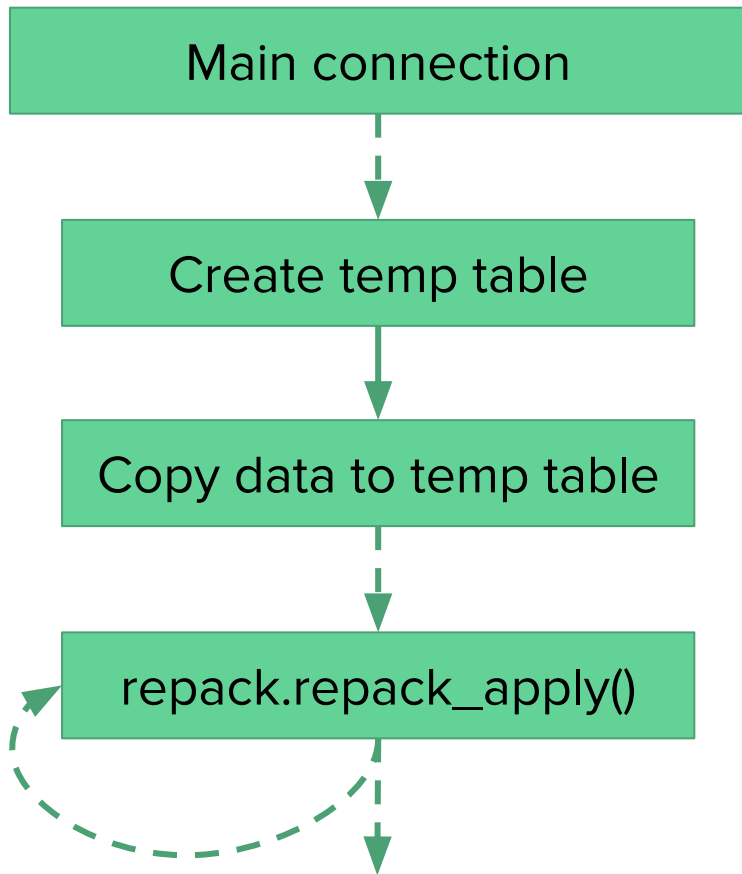
Running pg_repack: monitoring

- replication lag
- free disk space
- queries runtime
- number of locks

pg_repack execution steps



pg_repack execution steps



pg_repack execution steps

Main connection



Connection 2

ACCESS EXCLUSIVE

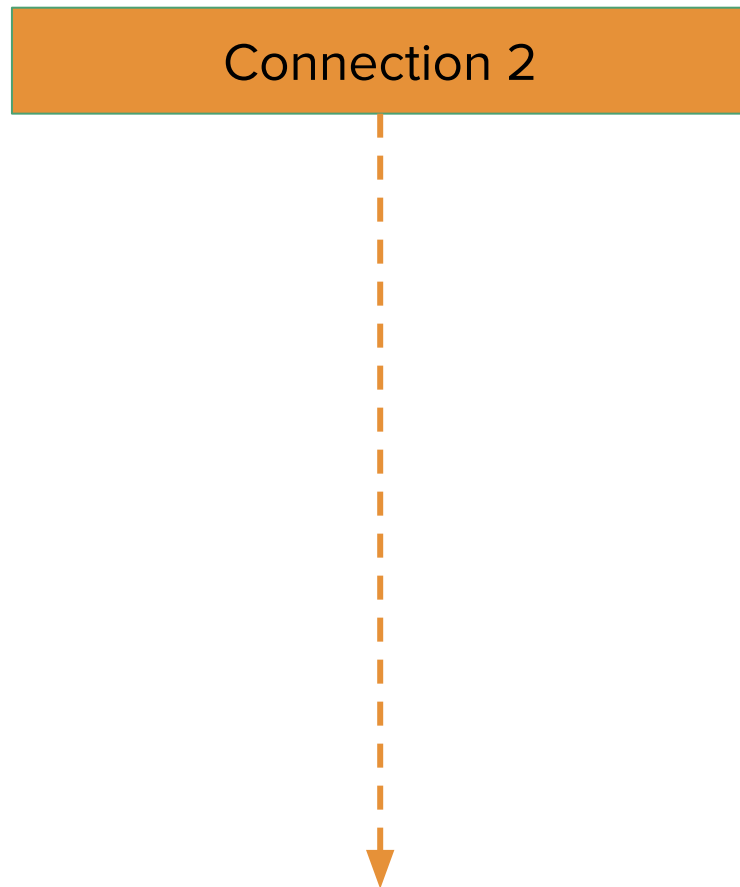
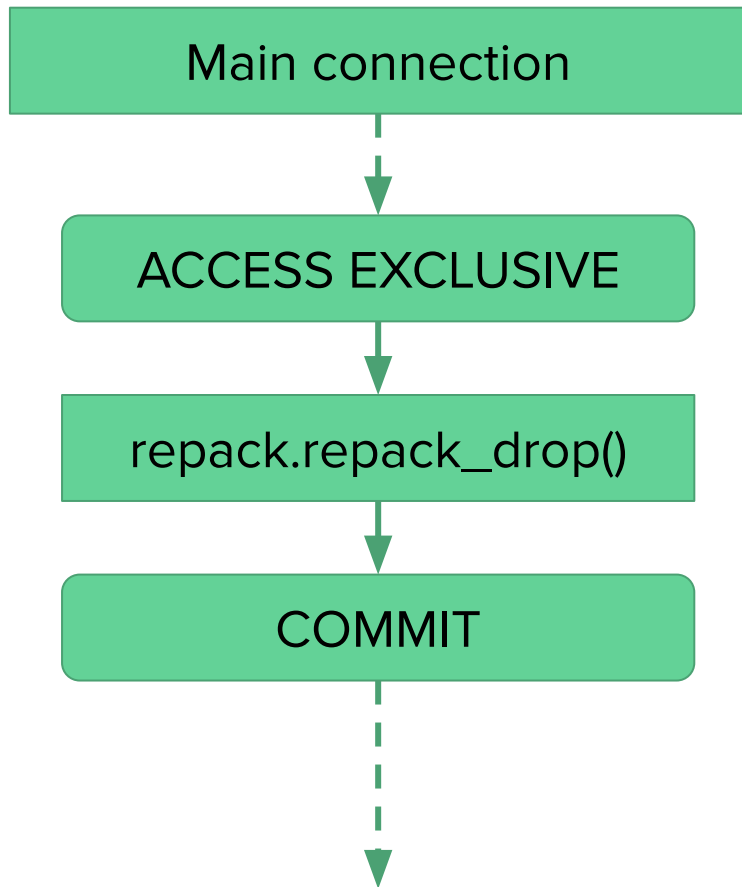
repack.repack_apply()

repack.repack_swap()

COMMIT



pg_repack execution steps



Acquiring ACCESS EXCLUSIVE in pg_repack

```
loop until timeout:  
    SET LOCAL lock_timeout = max(1sec, cur_timeout)  
    LOCK TABLE <table> IN ACCESS EXCLUSIVE MODE
```

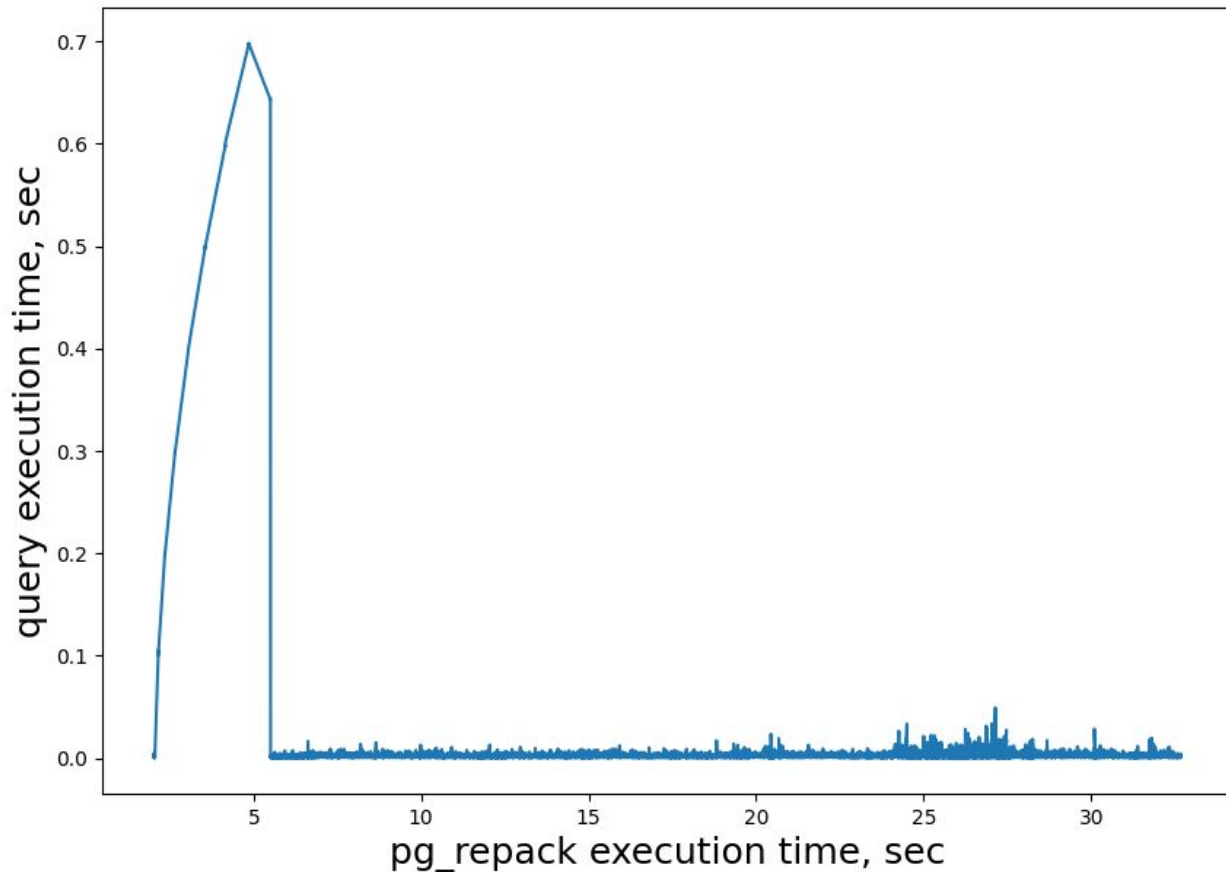
Except `repack.repack_drop()` in cleanup before pg_repack 1.5.0.

Fixed by https://github.com/reorg/pg_repack/pull/298.

Acquiring ACCESS EXCLUSIVE in pg_repack: Issue #1

Benchmark:

- Many fast SELECTs
- Long query, SELECT or UPDATE



Acquiring ACCESS EXCLUSIVE in pg_repack: Issue #1

Solutions:

- Make maximum lock_timeout configurable:

https://github.com/reorg/pg_repack/pull/183

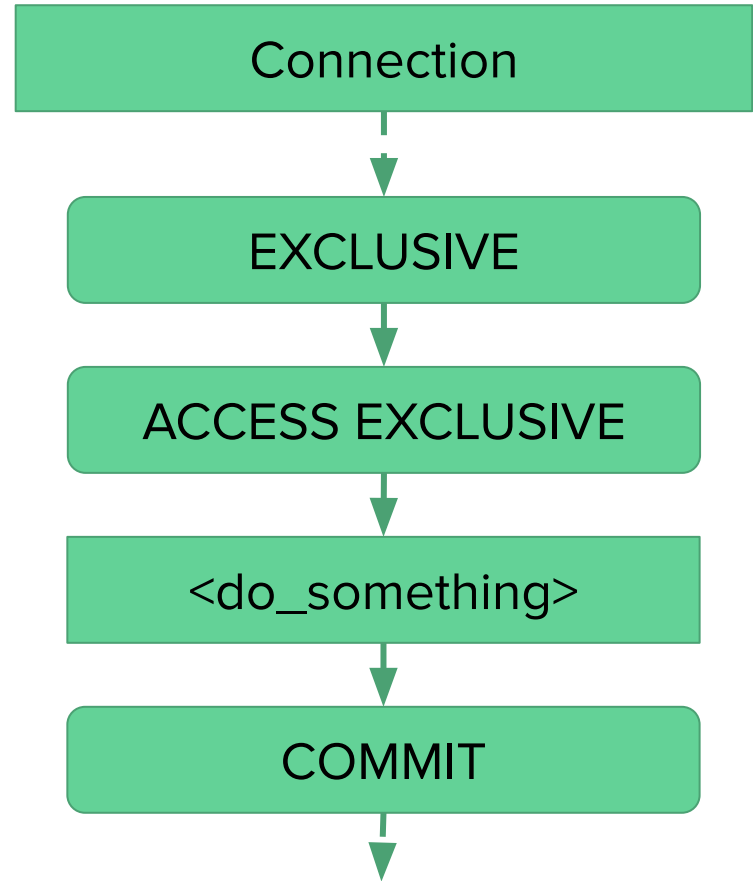
https://github.com/reorg/pg_repack/pull/275

- Try to acquire EXCLUSIVE lock first:

https://github.com/adjust/pg_repack/pull/8#issuecomment-894172672

Acquiring ACCESS EXCLUSIVE in pg_repack: Issue #1

Try to acquire EXCLUSIVE lock first



Acquiring ACCESS EXCLUSIVE in pg_repack: Issue #2

Benchmark:

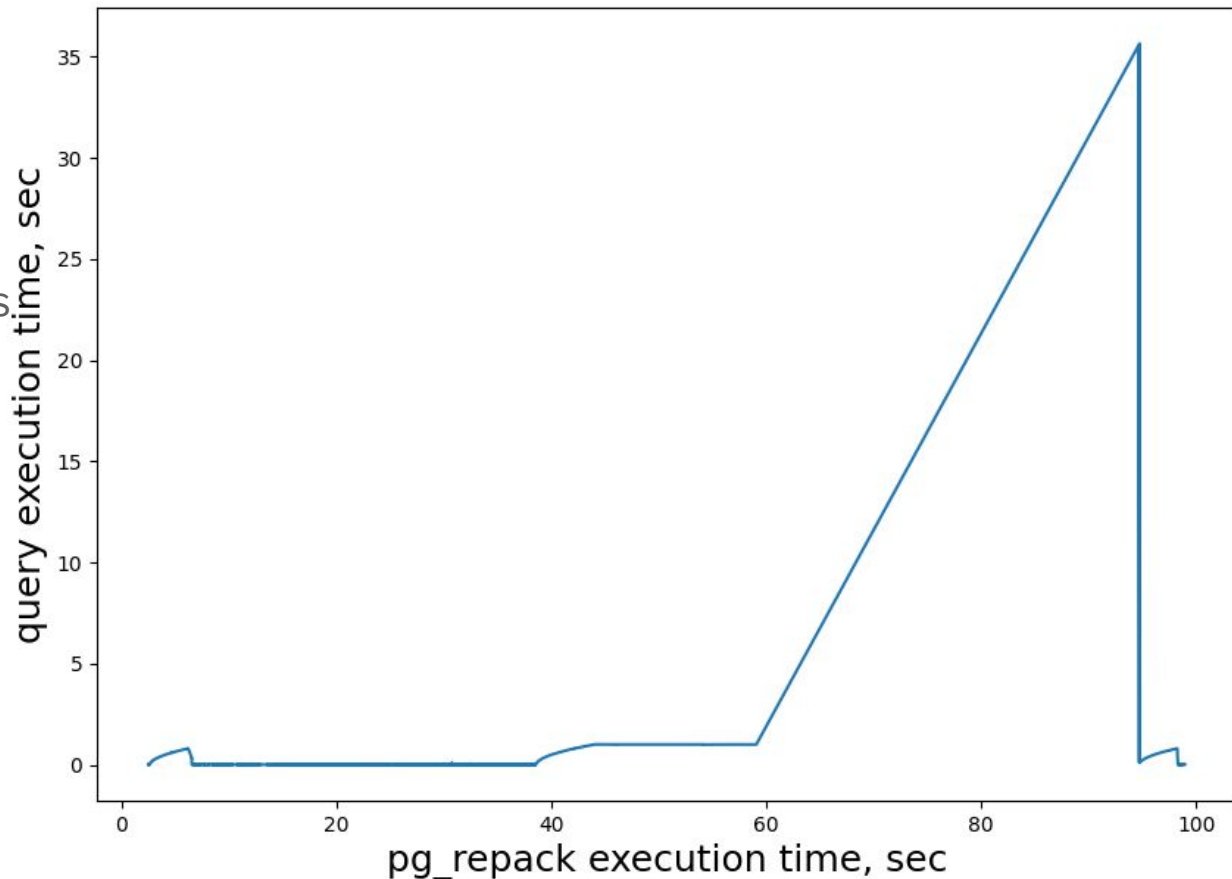
- Many fast SELECTs
- Many heavy UPDATEs

Issue:

- Log reply is too long

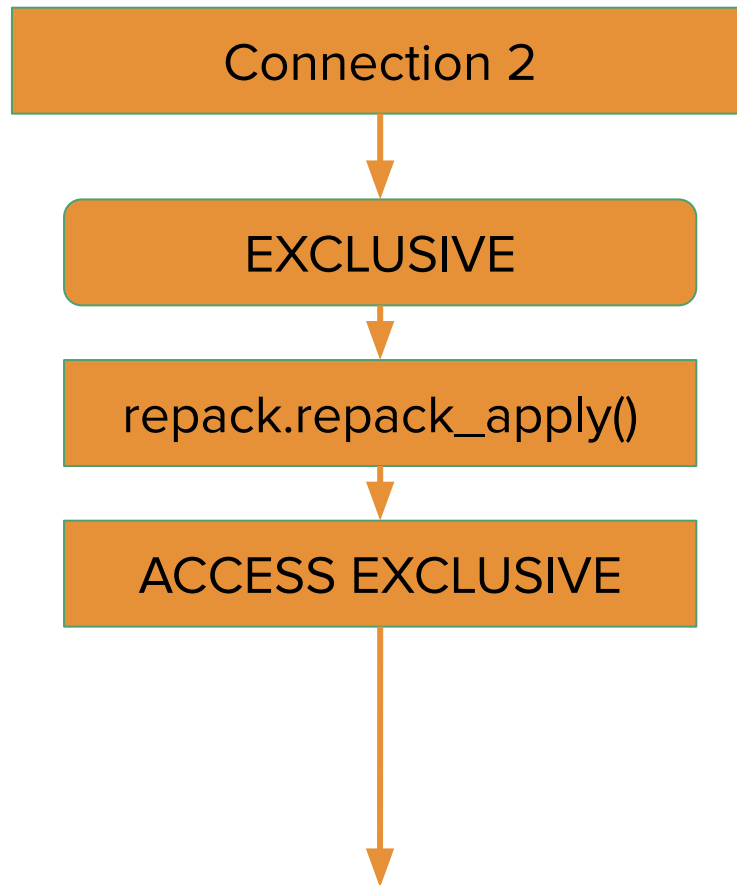
Solution:

- Acquire EXCLUSIVE lock first



Acquiring ACCESS EXCLUSIVE in pg_repack: Issue #2

Try to acquire EXCLUSIVE lock first



Thank you!
Any questions?