Operational hazards of managing PostgreSQL DBs over 100TB

adyen

engineered
for ambition

To backup or not To backup


RUNNING PG_BASEBACKUP ON YOUR +100TB DATABASE
imgflip.com

Some “Math”:

Some “Math”:

Backup

=

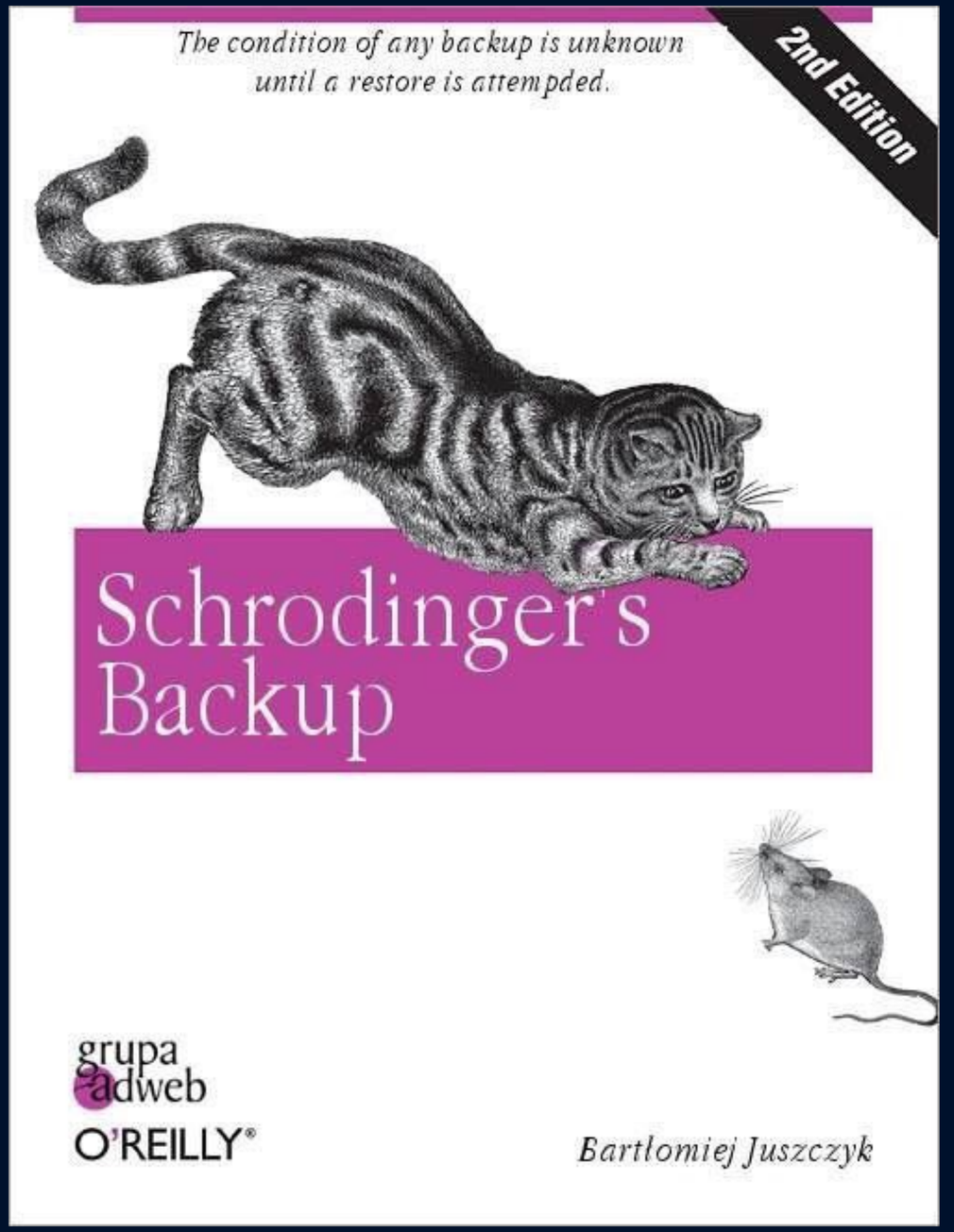
Backup + Restore

Some “Math”:

Backup

=

Backup + Restore



Source: https://orlybooks.com/

How we do it then?

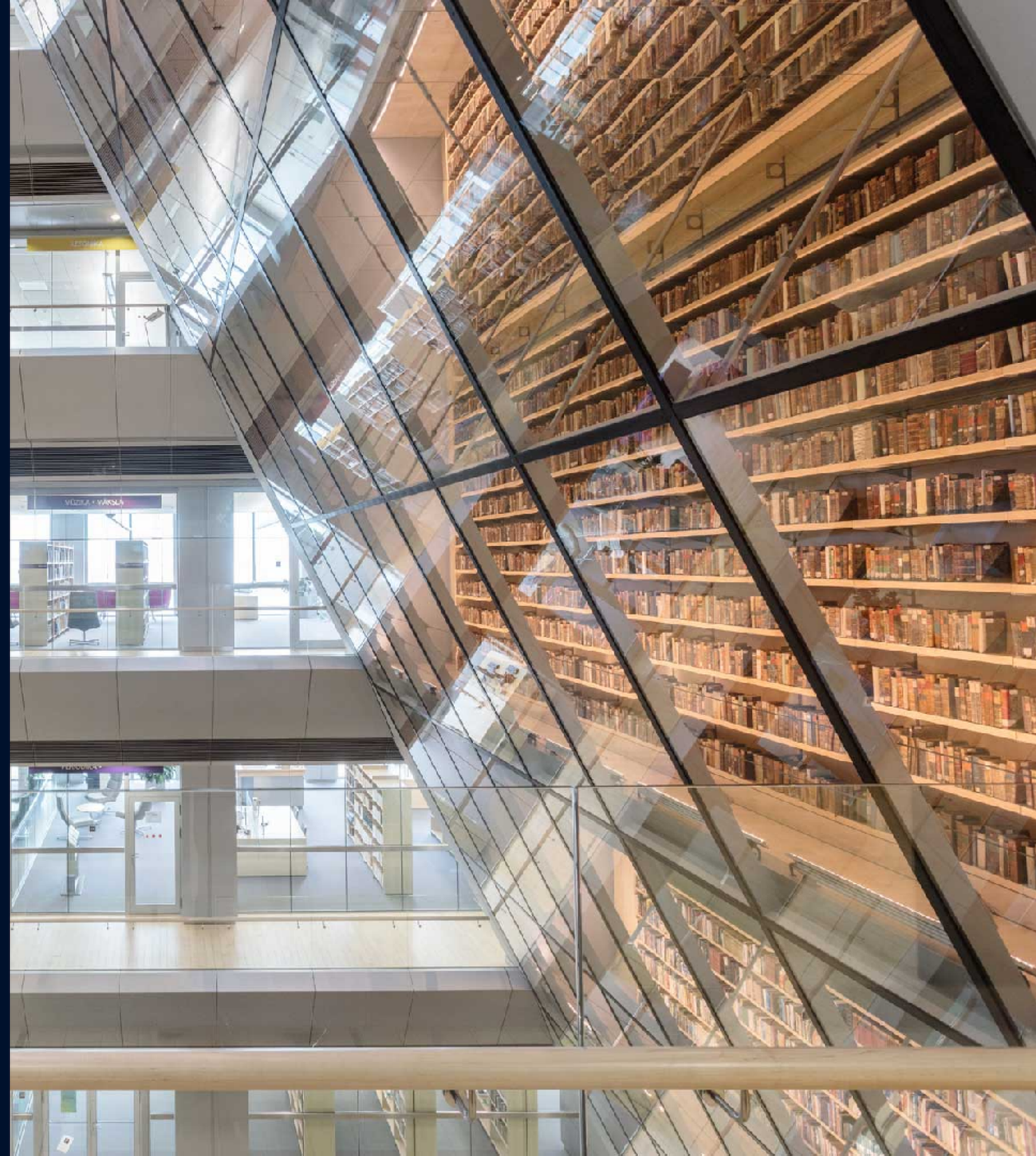
Storage
Snapshots

+

WAL
Backups

But, and the restores?!

Data, Data everywhere

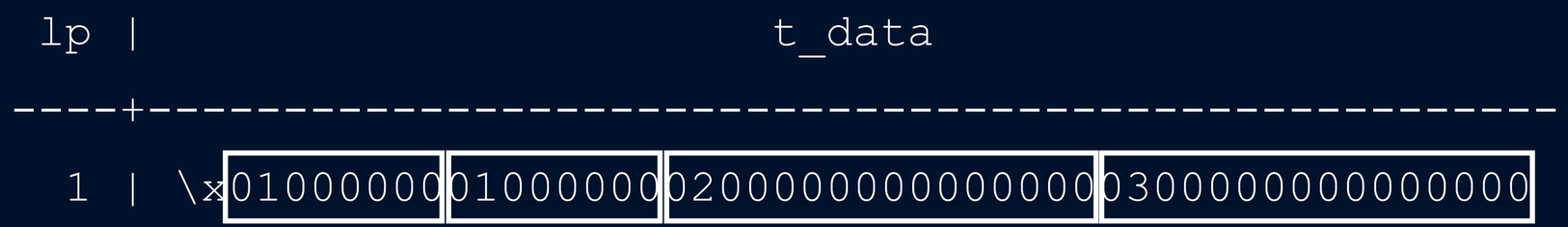


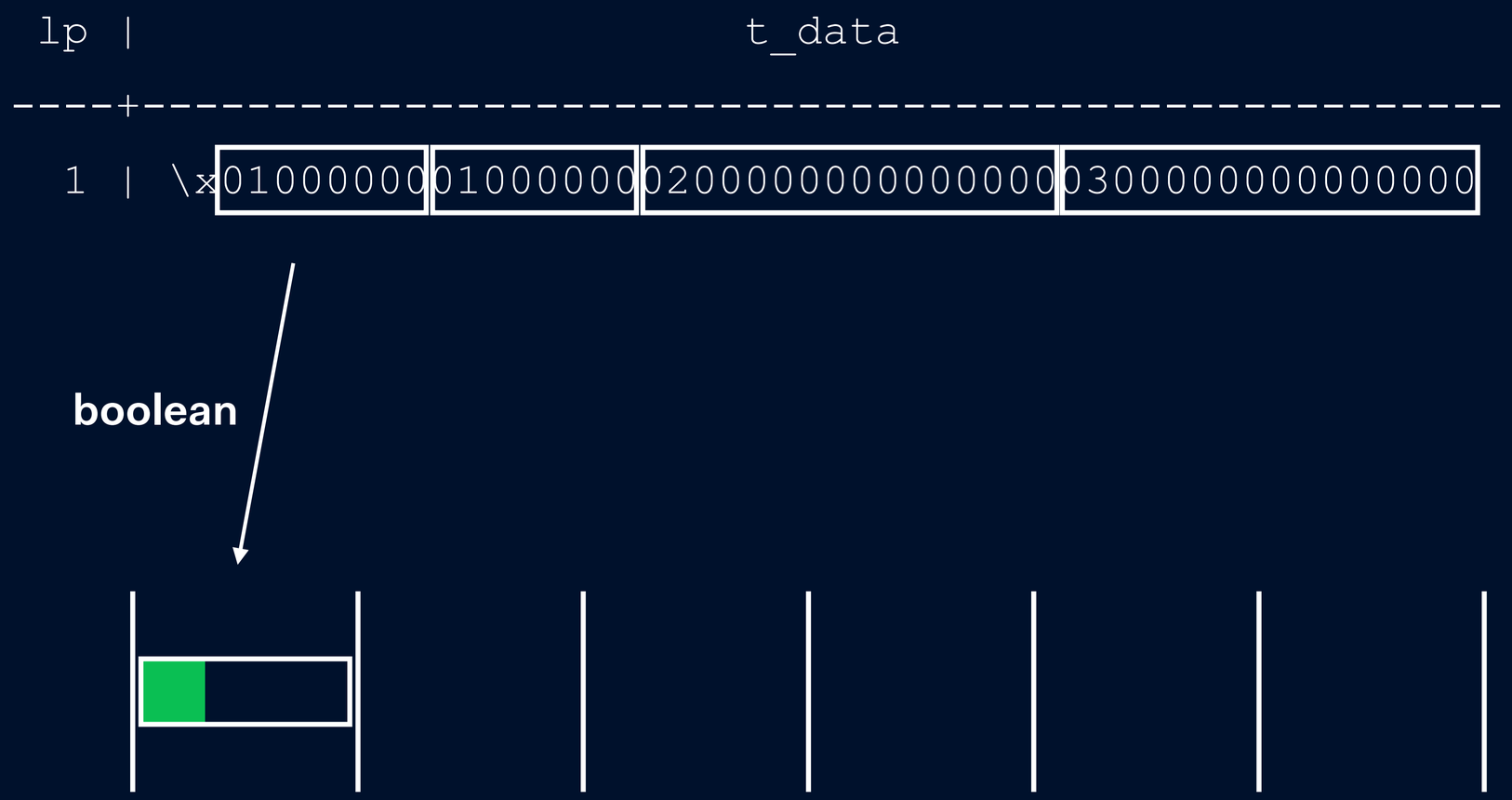

ALL OF THIS DATA...
WHY SHOULDN'T I KEEP IT?
imgflip.com

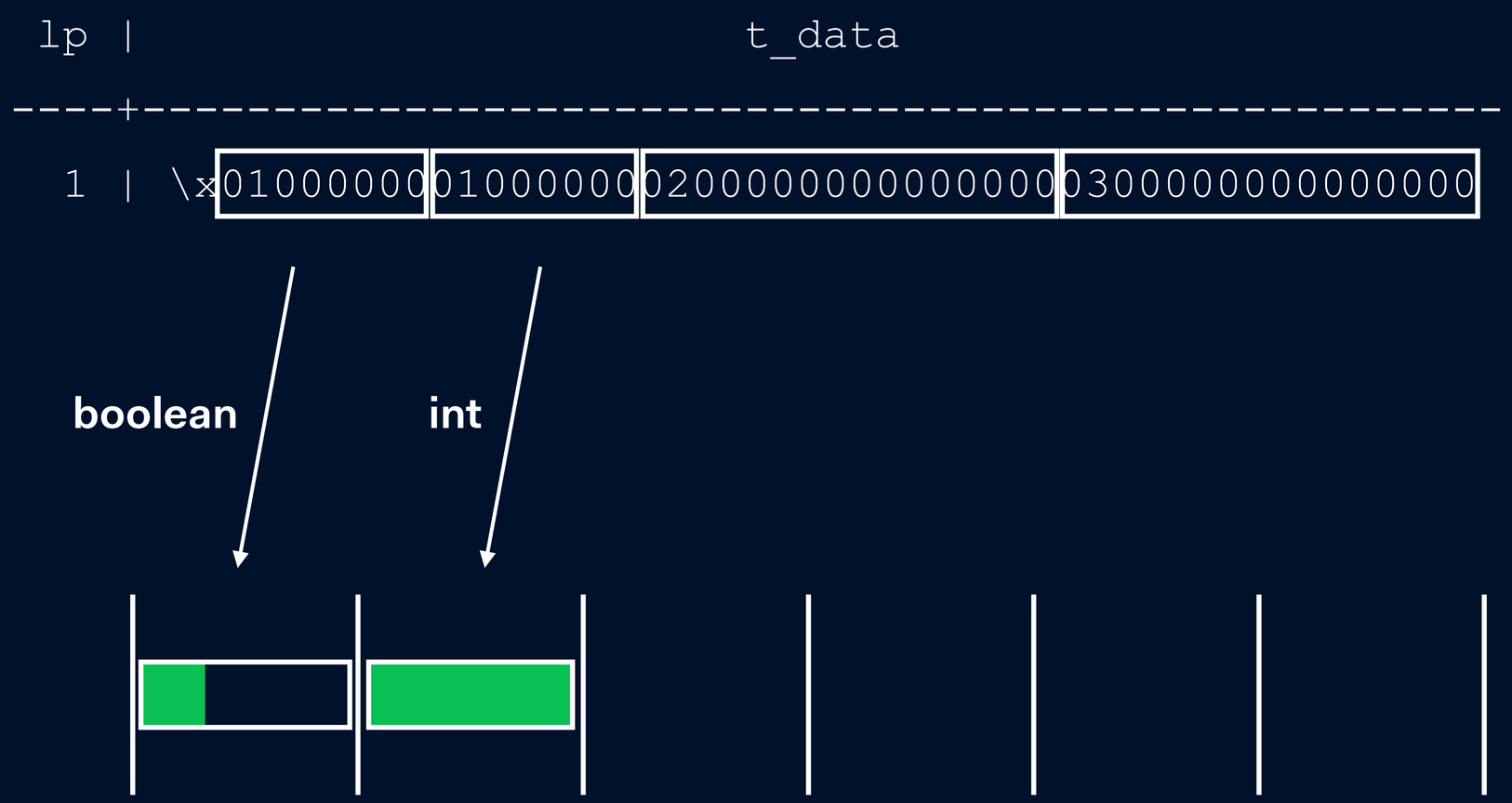
Get your columns in order

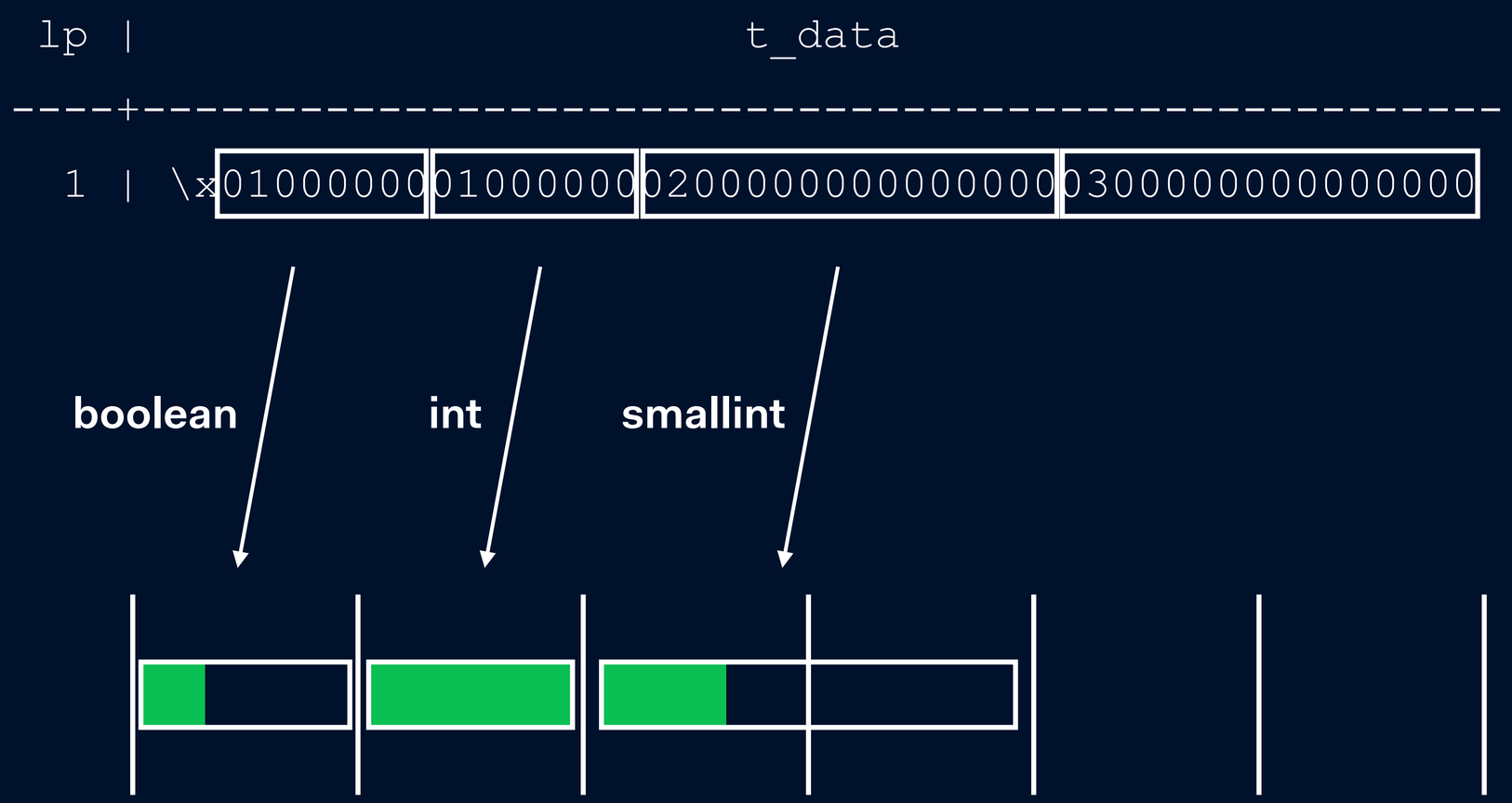
```
teresal=# \d test_order
             Table "public.test_order"
 Column |   Type   | Collation | Nullable | Default
--------+----------+-----------+----------+---------
 c1     | boolean  |           |          |
 c2     | integer  |           |          |
 c3     | smallint |           |          |
 c4     | bigint   |           |          |
```

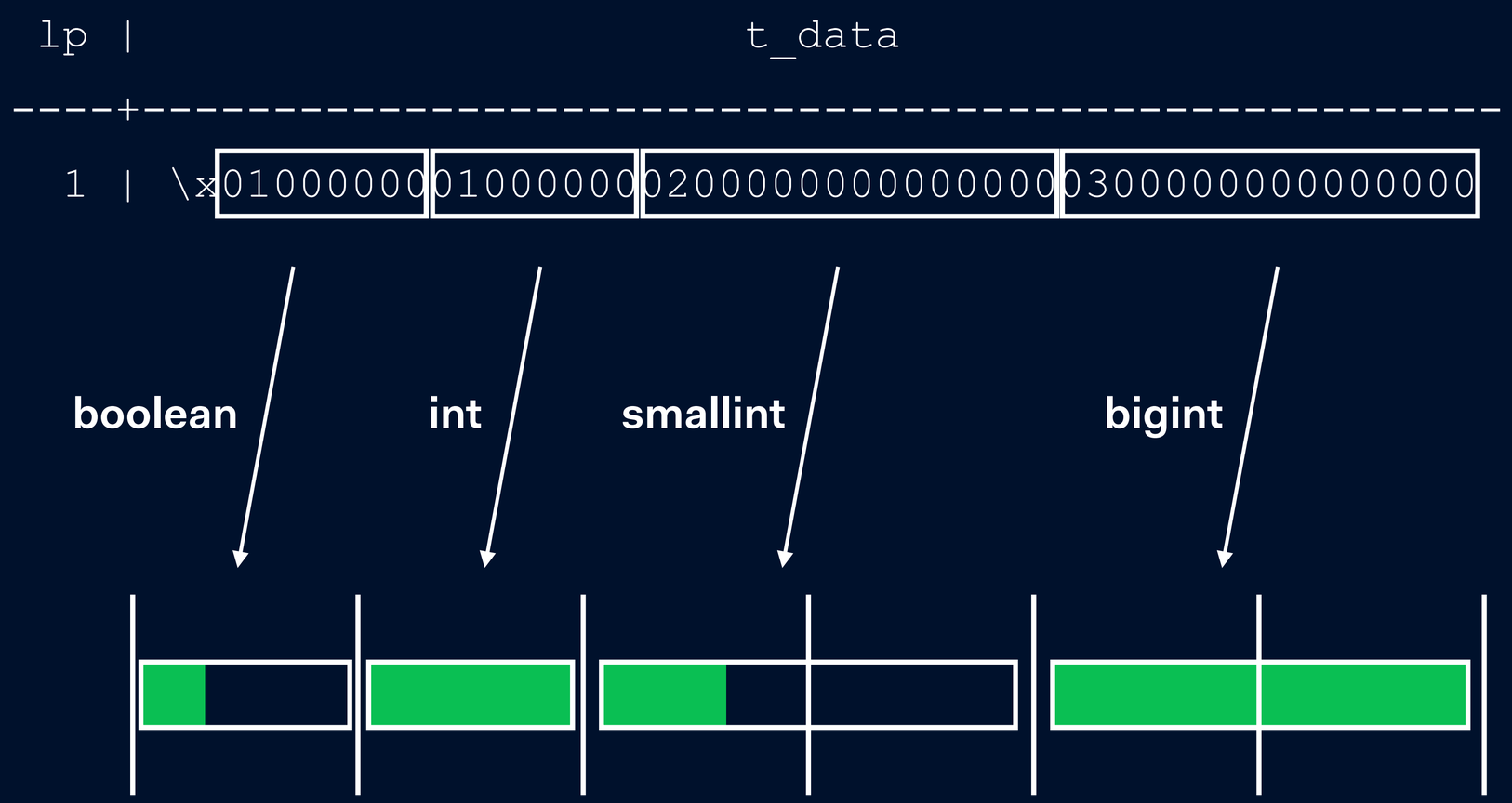
```
teresal=# \d test_order
             Table "public.test_order"
 Column |   Type   | Collation | Nullable | Default
--------+----------+-----------+----------+---------
 c1     | boolean  |           |          |
 c2     | integer  |           |          |
 c3     | smallint |           |          |
 c4     | bigint   |           |          |

teresal=# select * from test_order;
 c1 | c2 | c3 | c4
----+----+----+----
 t  |  1 |  2 |  3
```

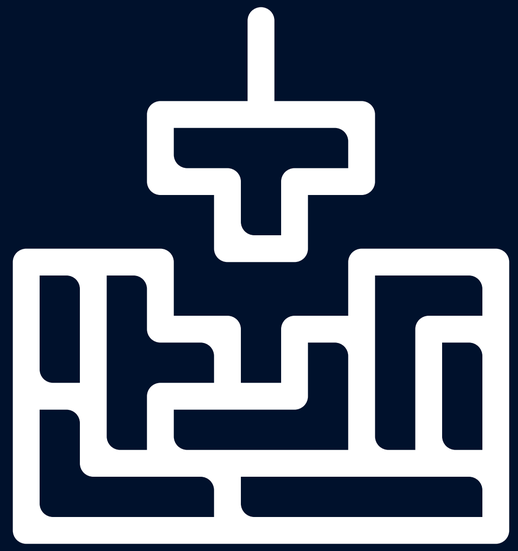

lp |
t_data
1 | \x
01000000
01000000
02000000000000000
03000000000000000


lp | t_data
1 | \x01000000010000000200000000000000300000000000000
boolean

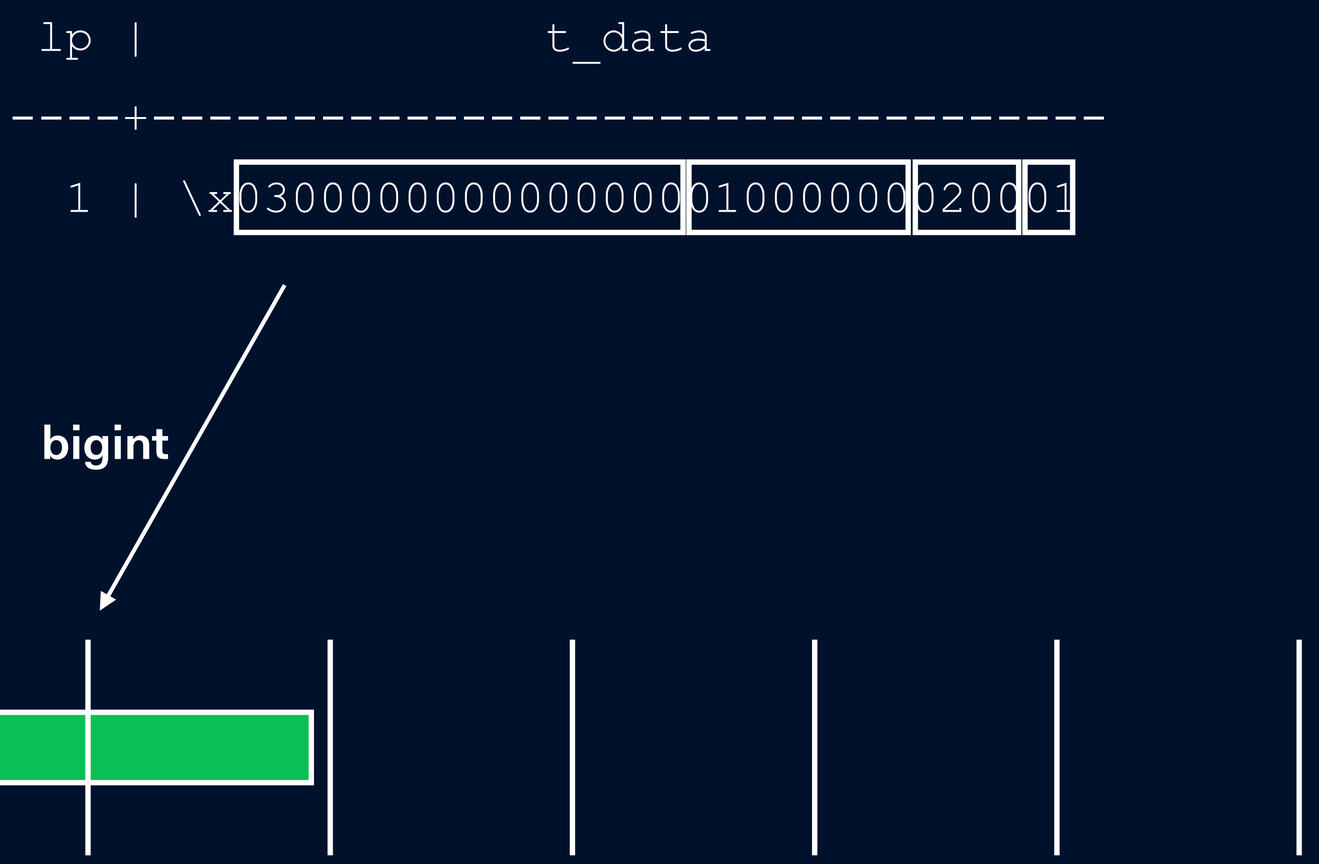

lp |
t_data
1 | \x
01000000
01000000
02000000000000000
03000000000000000
boolean
int

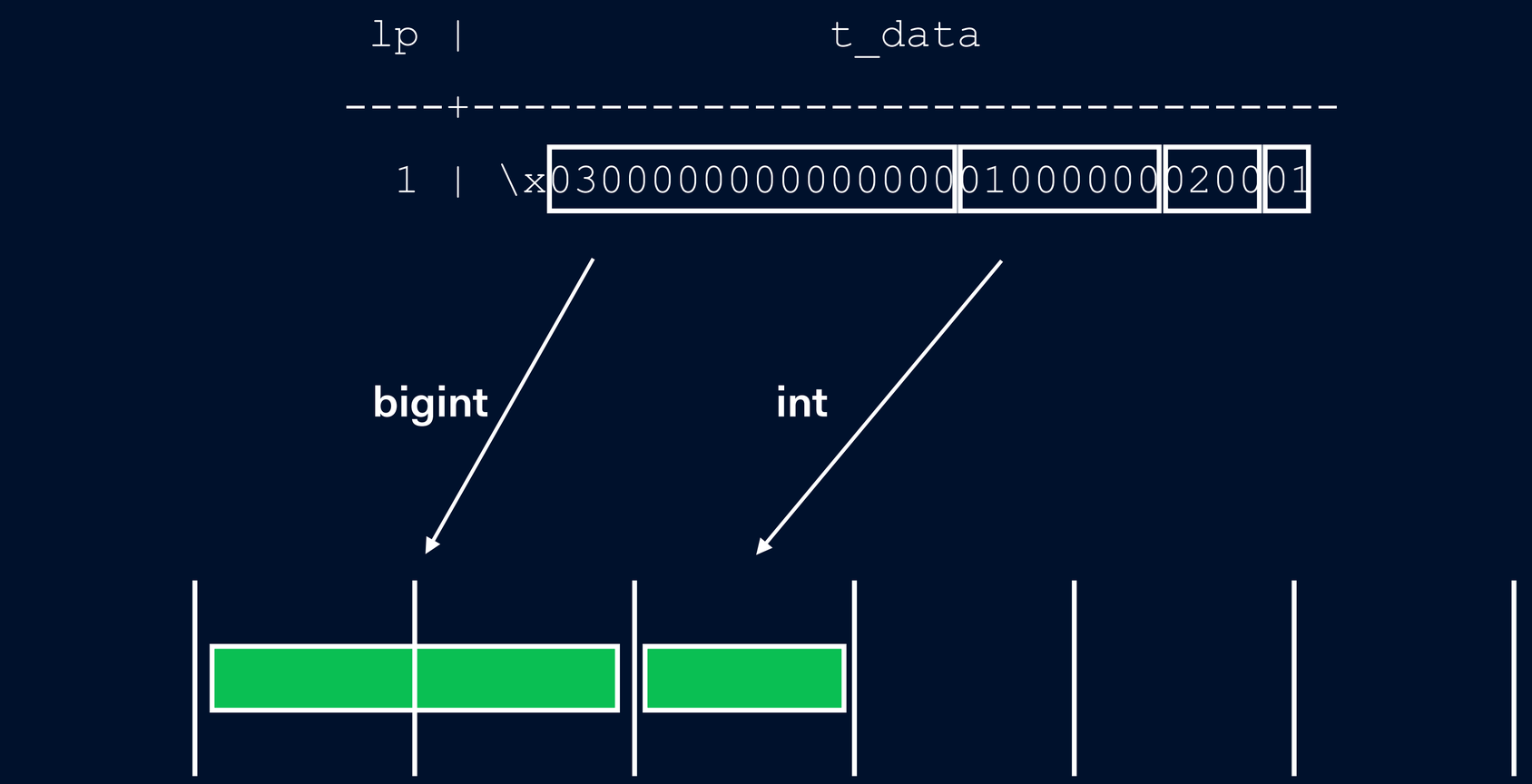

lp |
t_data
1 | \x
01000000
01000000
02000000000000000
03000000000000000
boolean
int
smallint

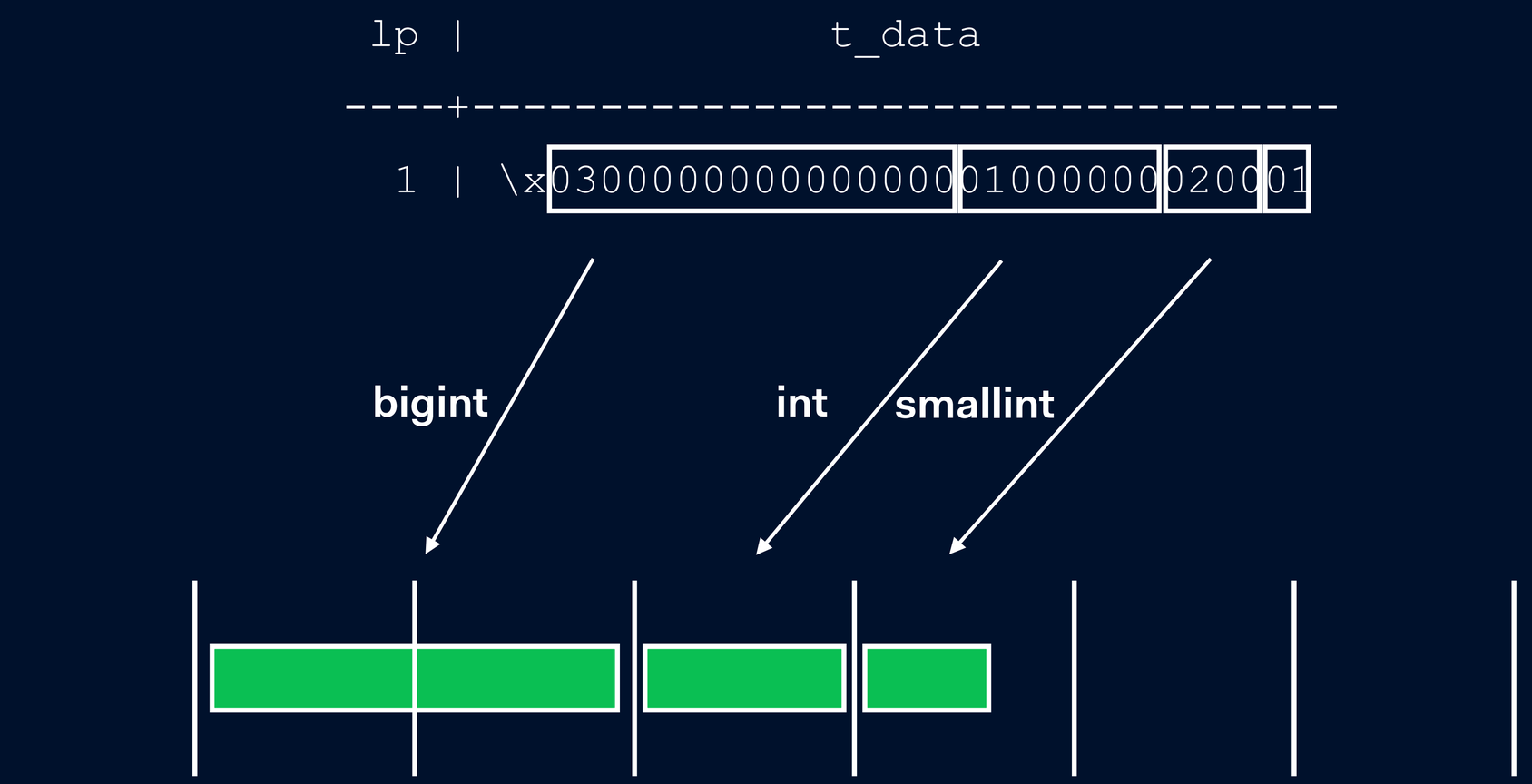

lp |
t_data
1 | \x
01000000
01000000
02000000000000000
03000000000000000
boolean
int
smallint
bigint

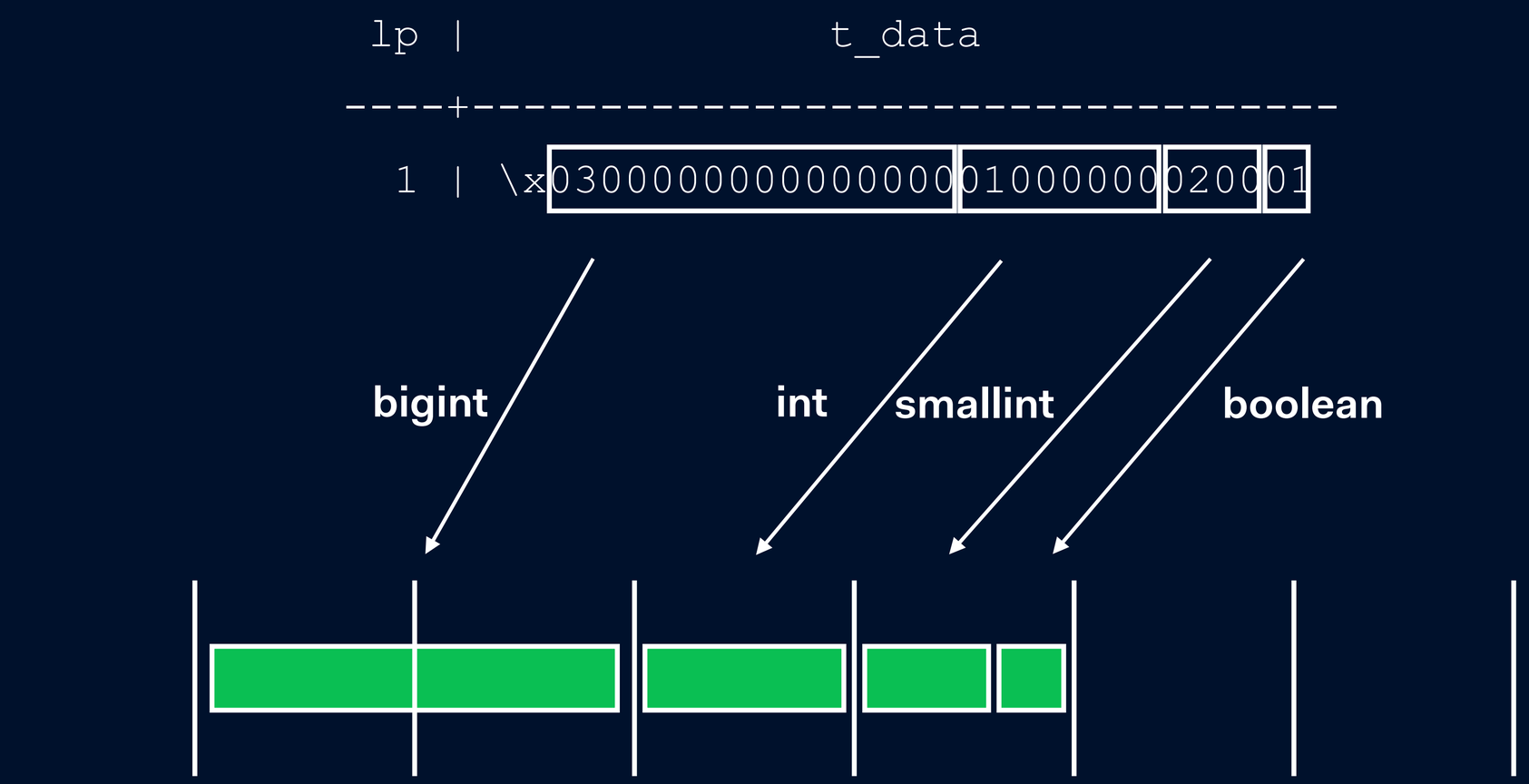


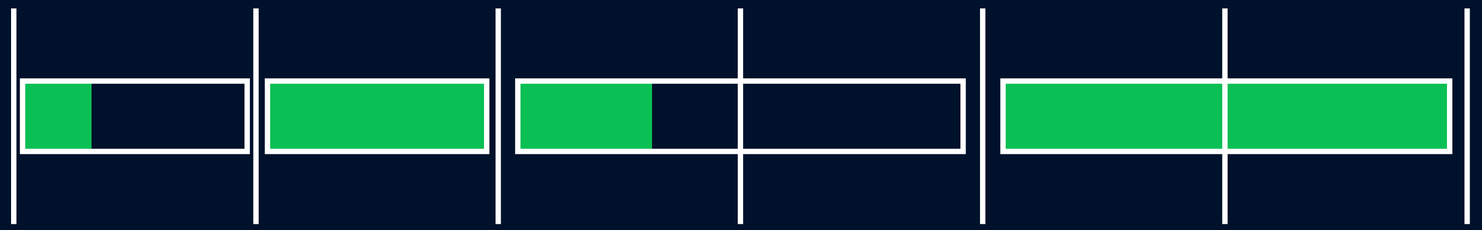

lp | t_data
1 | \x03000000000000000100000002000 1

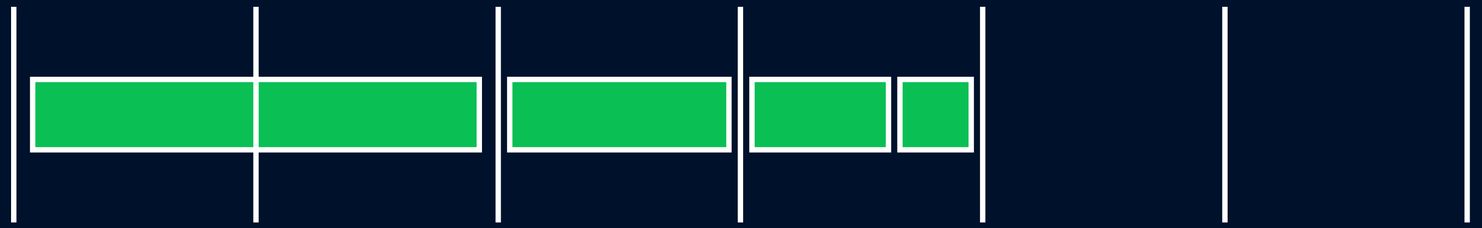

lp |
t_data
1 | \x03000000000000000100000002000 1
bigint


lp | t_data
1 | \x03000000000000000100000002000 1
bigint
int


lp |
t_data
1 | \x03000000000000000100000002 0001
bigint
int
smallint


lp |
t_data
1 | \x0300000000000000010000000200 01
bigint
int
smallint
boolean





More “Math”:

10% of +100TB

=

+10TB

More “Math”:

10% of +100TB

=

+10TB

More “Math”:

10% of +100TB

=

+10TB

DECEMBER 6, 2024 • 41 MINUTES

Column Tetris

Nikolay and Michael discuss "Column Tetris" — what it is, why it matters, how to order columns for new tables, and how to re-organise existing ones. Here are some link...

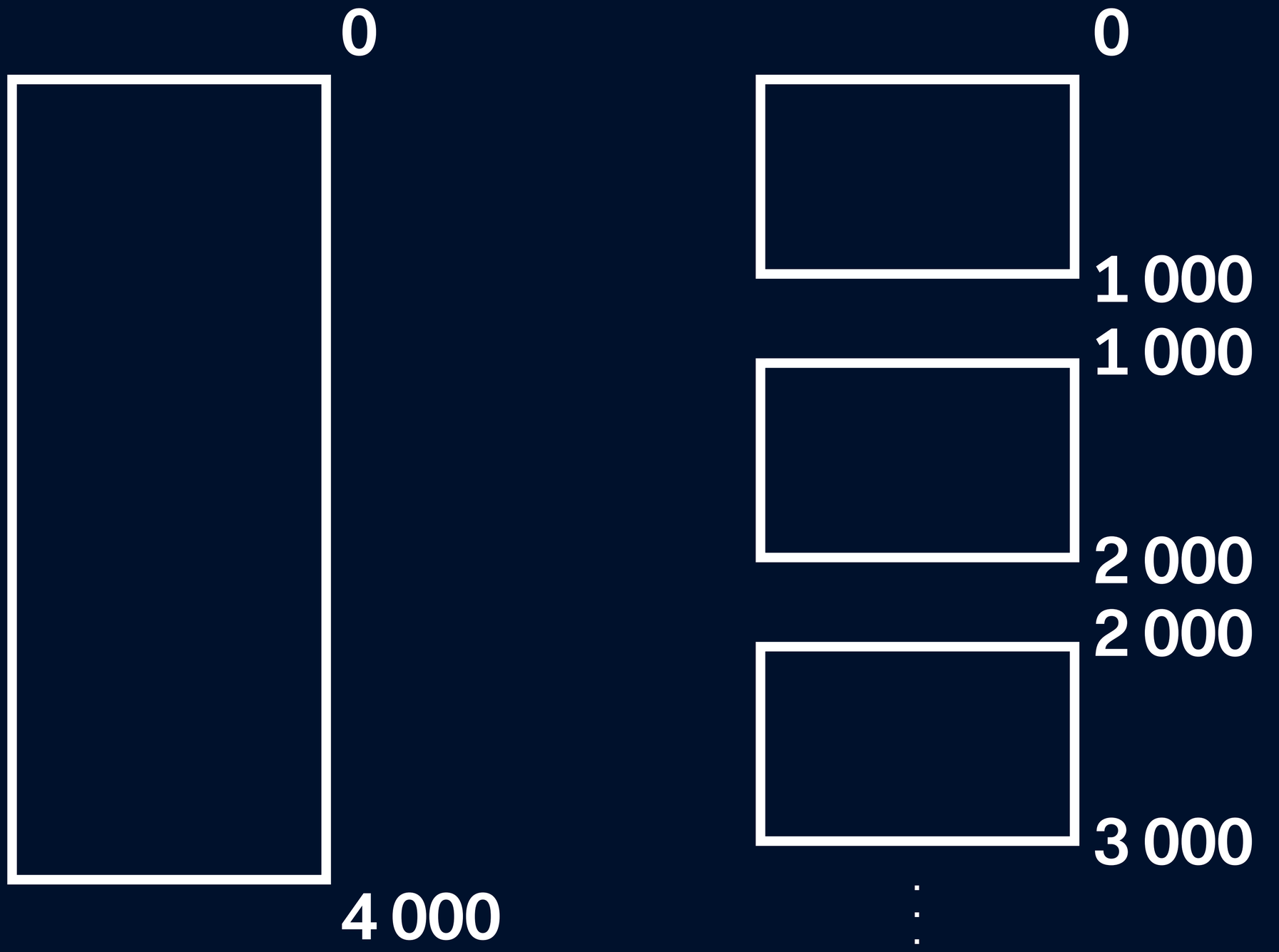
FOSDEM 2024

Reducing Costs and Improving Performance With Data Modeling in Postgres	Charly Batista

Tame the chaos,

Partition!


0
4 000


0
4 000
0
1 000
1 000
2 000
2 000
3 000

1 table ✓

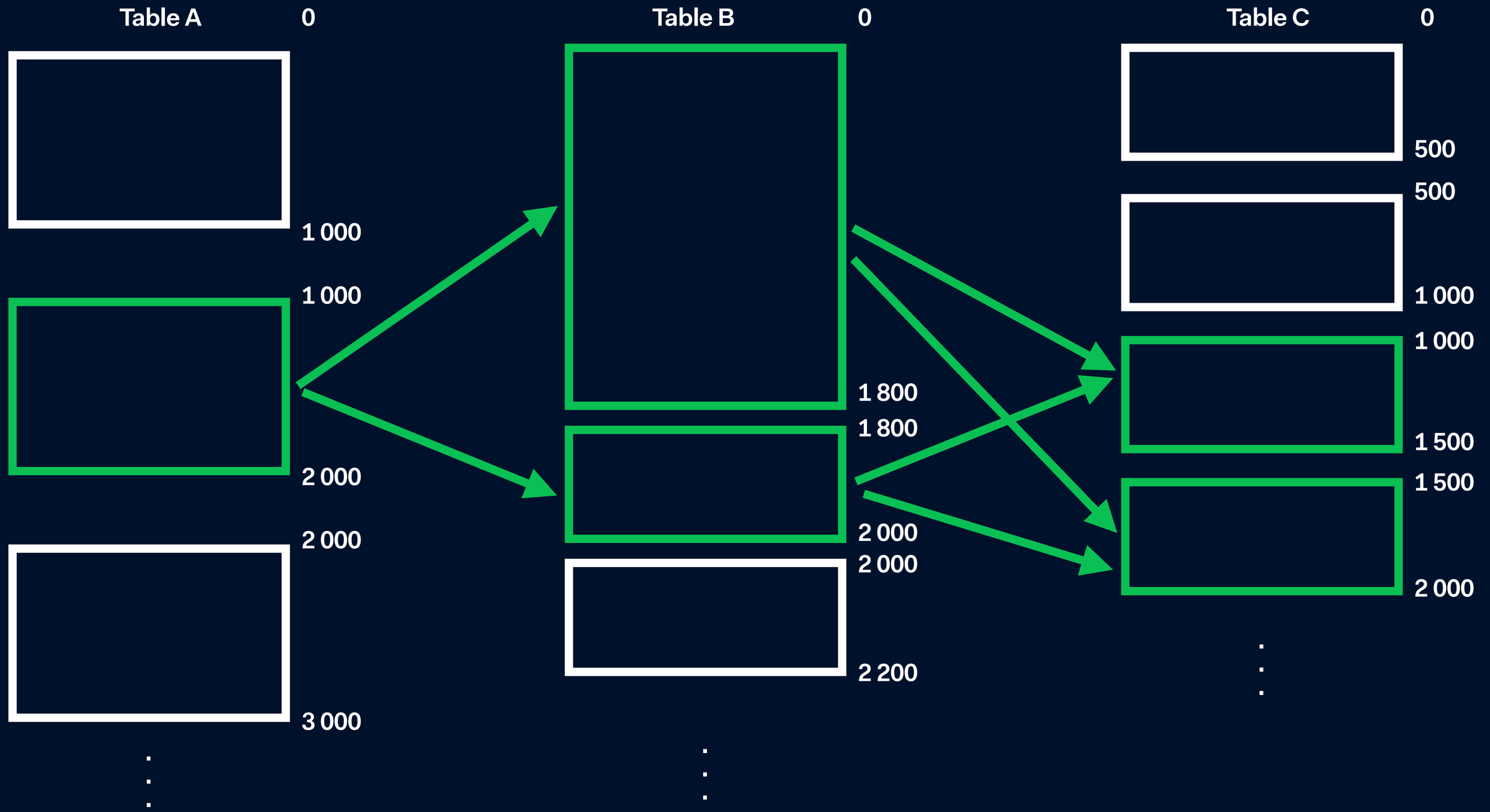
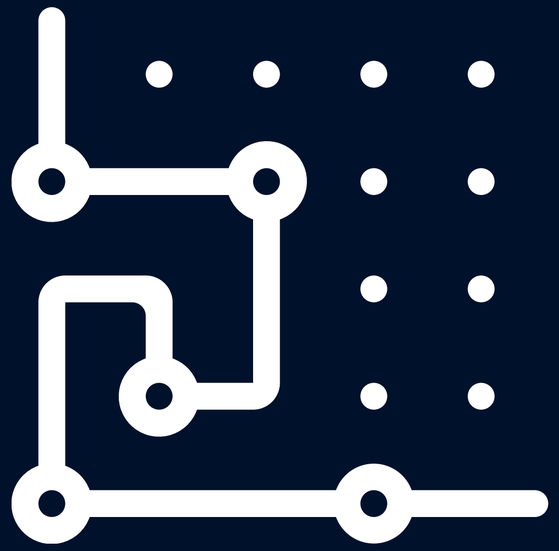


Table A
0
1 000
1 000
2 000
2 000
3 000
Table B
0
1 800
1 800
2 000
2 000
2 200
Table C
0
500
500
1 000
1 000
1 500
1 500
2 000

fast-path locks

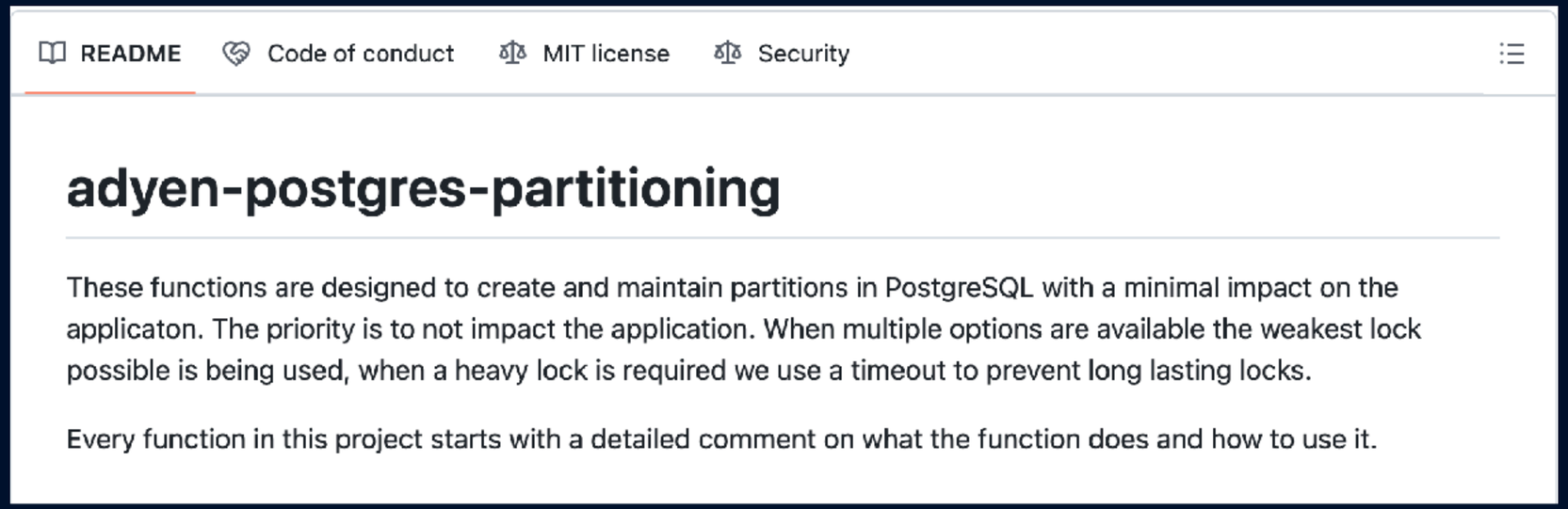
LWLock:lock_manager



Planning Time: 12.389 ms

Execution Time: 0.164 ms

Partitioning at Adyen



Partitioning at Adyen

README | Code of conduct | MIT license | Security

adyen-postgres-partitioning

These functions are designed to create and maintain partitions in PostgreSQL with a minimal impact on the applicaton. The priority is to not impact the application. When multiple options are available the weakest lock possible is being used, when a heavy lock is required we use a timeout to prevent long lasting locks.

Every function in this project starts with a detailed comment on what the function does and how to use

PGConf.EU 2025

Beyond the How: Why Table Partitioning Truly Matters

Thursday, October 23 at 13:55–14:45

Derk van Veen

Adyen

Partitioning at Adyen

Partitioning at Adyen


Tech stories

A Deep Dive into Table Partitioning part 1 🐰: Introduction to Table Partitioning

By Cosmin Octavian Pene(Java Engineer) & Derk van Veen (Database Engineer), Adyen


Tech stories

A Deep Dive into Table partitioning 🐰 Part 2: Partitioning at Adyen

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer), Adyen

Partitioning at Adyen

Tech stories

A Deep Dive into Table Partitioning part 1 🐰: Introduction to Table Partitioning

By Cosmin Octavian Pene(Java Engineer) & Derk van Veen (Database Engineer), Adyen

Tech stories

A Deep Dive into Table partitioning 🐰 Part 2: Partitioning at Adyen

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer),

Tech stories

A Deep Dive into Table partitioning 🐰 Part 3: Maintenance Under Pressure

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer), Adyen.

Partitioning at Adyen


Tech stories

A Deep Dive into Table Partitioning part 1 🐇: Introduction to Table Partitioning

By Cosmin Octavian Pene(Java Engineer) & Derk van Veen (Database Engineer), Adyen


Tech stories

A Deep Dive into Table partitioning 🐇 Part 4: How the default partition saved the day

By Derk van Veen, Database Engineer


Tech stories

A Deep Dive into Table partitioning 🐇 Part 2: Partitioning at Adyen

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer),


Tech stories

A Deep Dive into Table partitioning 🐇 Part 3: Maintenance Under Pressure

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer), Adyen.

Partitioning at Adyen

Tech stories

A Deep Dive into Table Partitioning part 1 🐰: Introduction to Table Partitioning

By Cosmin Octavian Pene(Java Engineer) & Derk van Veen (Database Engineer), Adyen

Tech stories

A Deep Dive into Table partitioning 🐰 Part 2: Partitioning at Adyen

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer),

Tech stories

A Deep Dive into Table partitioning 🐰 Part 3: Maintenance Under Pressure

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer), Adyen.

Tech stories

A Deep Dive into Table partitioning 🐰 Part 4: How the default partition saved the day

By Derk

Tech stories

Efficient Data Cleanup in Partitioned PostgreSQL Tables using Common Table Expressions

By Dwarka Rao, Database Engineer & Aakash Agarwal, Java Software Engineer.

Partitioning at Adyen

Tech stories

A Deep Dive into Table Partitioning part 1 🐇: Introduction to Table Partitioning

By Cosmin Octavian Pene(Java Engineer) & Derk van Veen (Database Engineer), Adyen

Tech stories

A Deep Dive into Table partitioning 🐇 Part 2: Partitioning at Adyen

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer),

Tech stories

A Deep Dive into Table partitioning 🐇 Part 3: Maintenance Under Pressure

By Derk van Veen (Database Engineer) & Cosmin Octavian Pene(Java Engineer), Adyen.

Tech stories

A Deep Dive into Table partitioning 🐇 Part 4: How the default partition saved the day

By Derk

Tech stories

Efficient Data Cleanup in Partitioned PostgreSQL Tables using Common Table Expressions

By Dwarka Rao, Database Engineer & Aakash Agarwal, Java Software Engineer.

Efficiently RePartitioning Large Tables in PostgreSQL

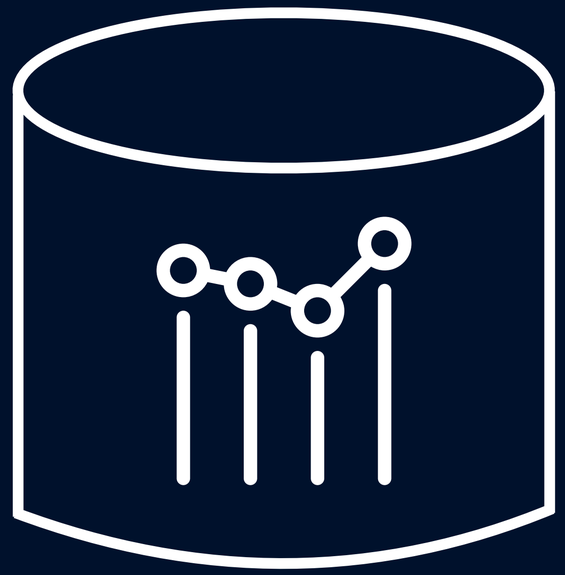
By Cagri Biroglu, Database Engineer

Data Temperature

Query Performance

Prepare your Statements

Prepare your Statements (Carefully)



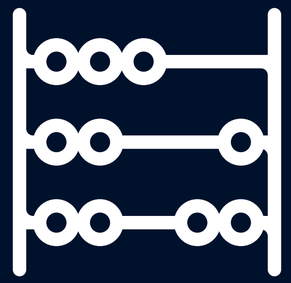
Plans

Generic

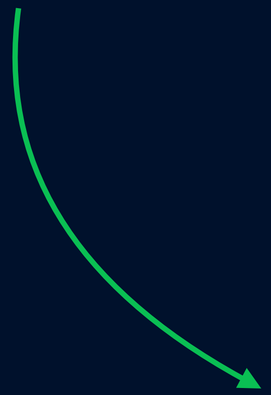
```
plan_cache_mode=force_generic_plan
```

Custom

```
plan_cache_mode=force_custom_plan
```



PostgreSQL v16



Add EXPLAIN option GENERIC_PLAN to display the generic plan for a parameterized query

(Laurenz Albe)

Wrap Up

We are hiring!

An Introduction to B-trees and LSM-trees for DEVs and DevOps

Friday, October 24 at 10:25–10:55

Dave Pitts

Adyen (Spain)

Beyond the How: Why Table Partitioning Truly Matters

Thursday, October 23 at 13:55–14:45

Derk van Veen

Adyen

The SyncRep Detective Story: Chasing Ghosts in PostgreSQL, Finding Demons in Storage

Thursday, October 23 at 11:25–12:15

Dmitry Fomin

Adyen

Wrap Up

(for real now)


OUT
OF
ORDER


OUT
OF
ORDER


OUT
OF
ORDER

What works at 100GB will fail catastrophically at +100TB.

Feedback

Operational hazards of managing PostgreSQL DBs over 100TB

adyen

engineered
for ambition