PostgreSQL Concurrency control

adyen

engineered
for ambition

Perspective

User 1

Perspective

User 2

Perspective

User 3

Perspective

Derk van Veen

Database engineer

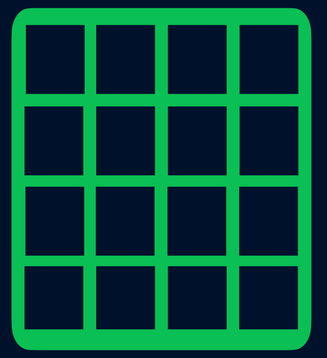
Adyen

adyen

engineered
for ambition

Adyen ♡ Postgres

100s



1000

Many > 100TB

Possibly the biggest critical Postgres shop in the world!

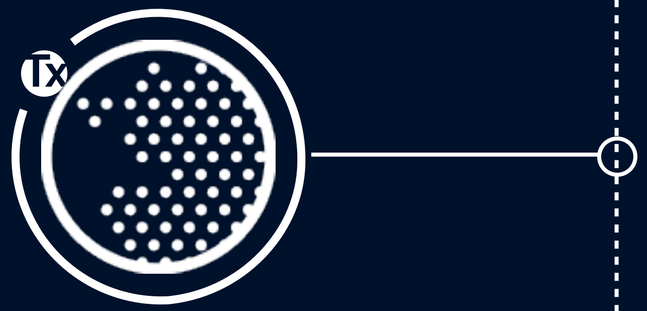
adyen

engineered
for ambition

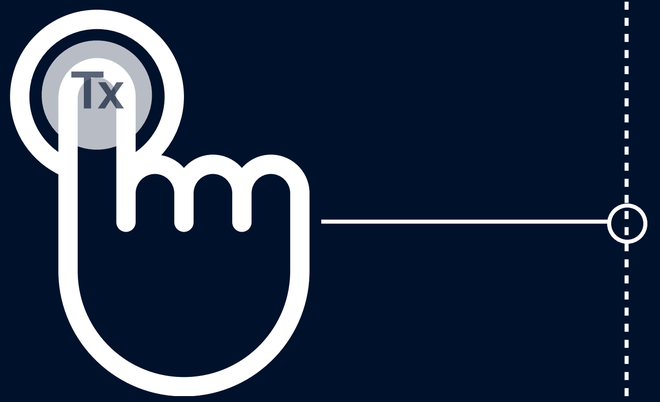
Agenda

Transaction definition

Individual transactions



Transaction space



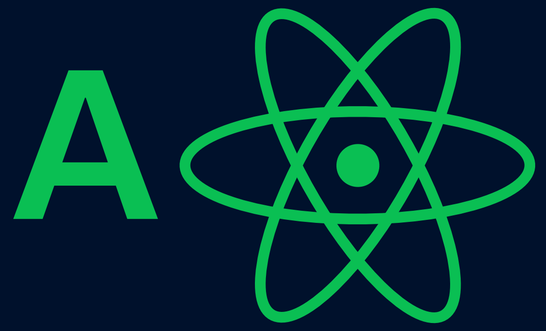
Using transactions

Locks

Isolation levels

Transaction definition

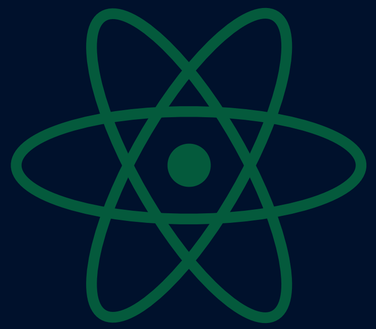
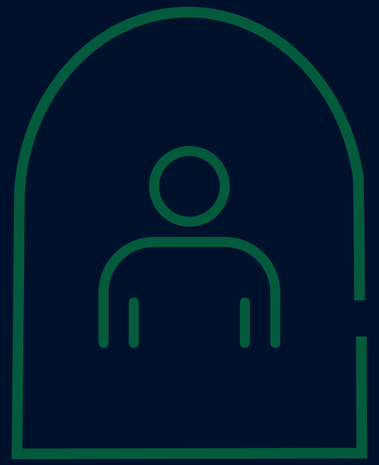
Transaction definition

A 

Transaction definition

Transaction definition

Transaction definition

A  C  I  D 

Transaction definition

Individual transactions

Insert

```
Insert into rollercoaster
values('baron')
```

t=101

Insert

t=101

SQL

Insert

t=101

SQL

Delete

```
delete from rollercoaster
('baron')
```

t=101

t=105

SQL

Delete

t=101

t=105

SQL

SQL

SQL

Update

Delete

Insert

Update

t=103

SQL

Update

t=103

SQL

Overview

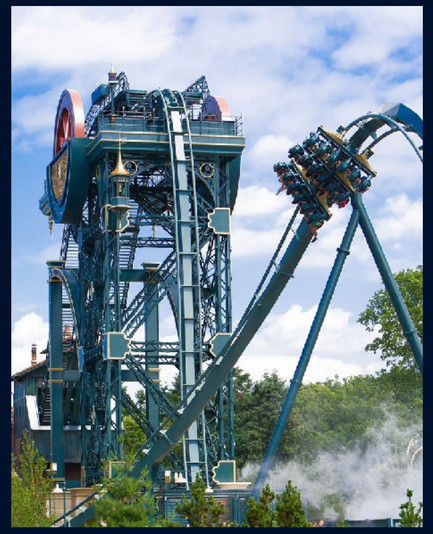
t=101 t=105

t=103

Page level

Page header

t_min=10, t_max=103



t_min=101, t_max=105

t_min=103, t_max=

Page footer

Page level

Page header

t_min=10, t_max=103

t=100

select * from rollercoaster

t_min=101, t_max=105

t_min=103, t_max=

Page footer

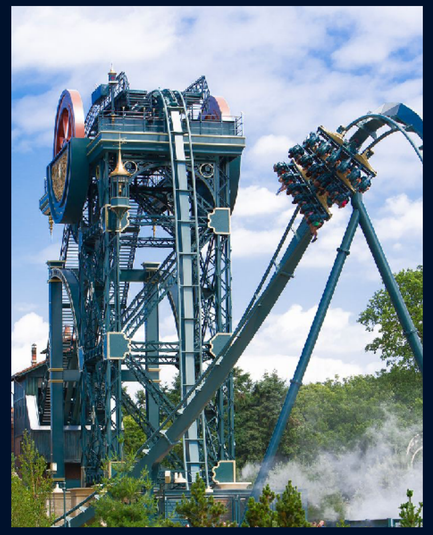
Page level

Page header

t_min=10, t_max=103

t=102

select * from rollercoaster



t_min=101, t_max=105

t_min=103, t_max=

Page footer

Page level

Page header

t_min=10, t_max=103

t_min=101, t_max=105

t_min=103, t_max=

t=106

select * from rollercoaster

Page footer

Transaction space

Transaction space

oldest not committed

first available

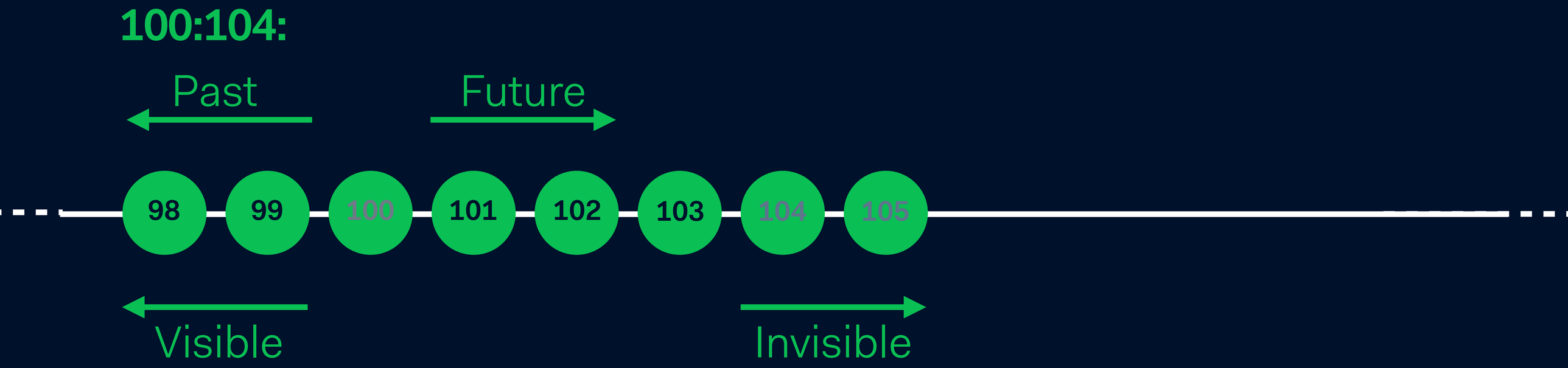
active list

Transaction snapshot

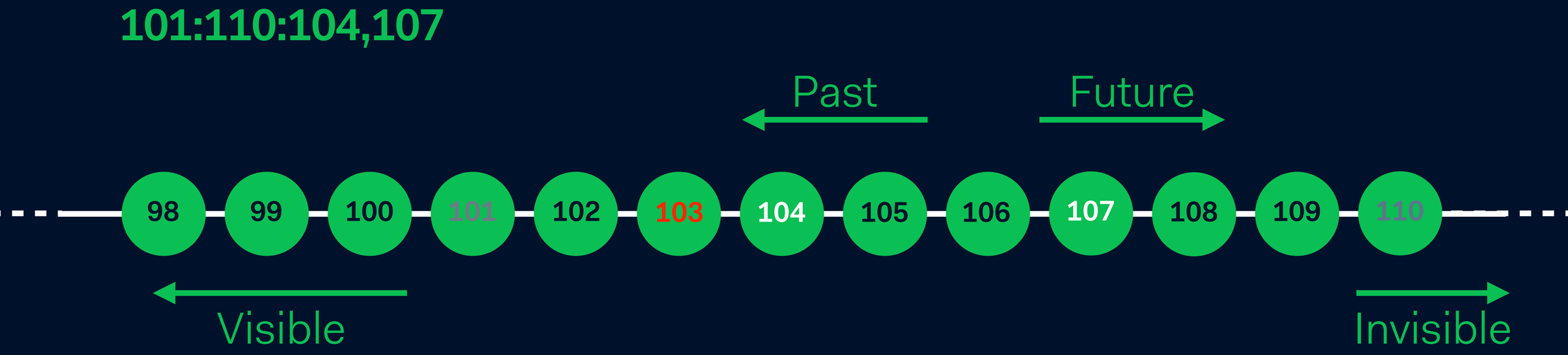
First not committed transaction

Transaction snapshot

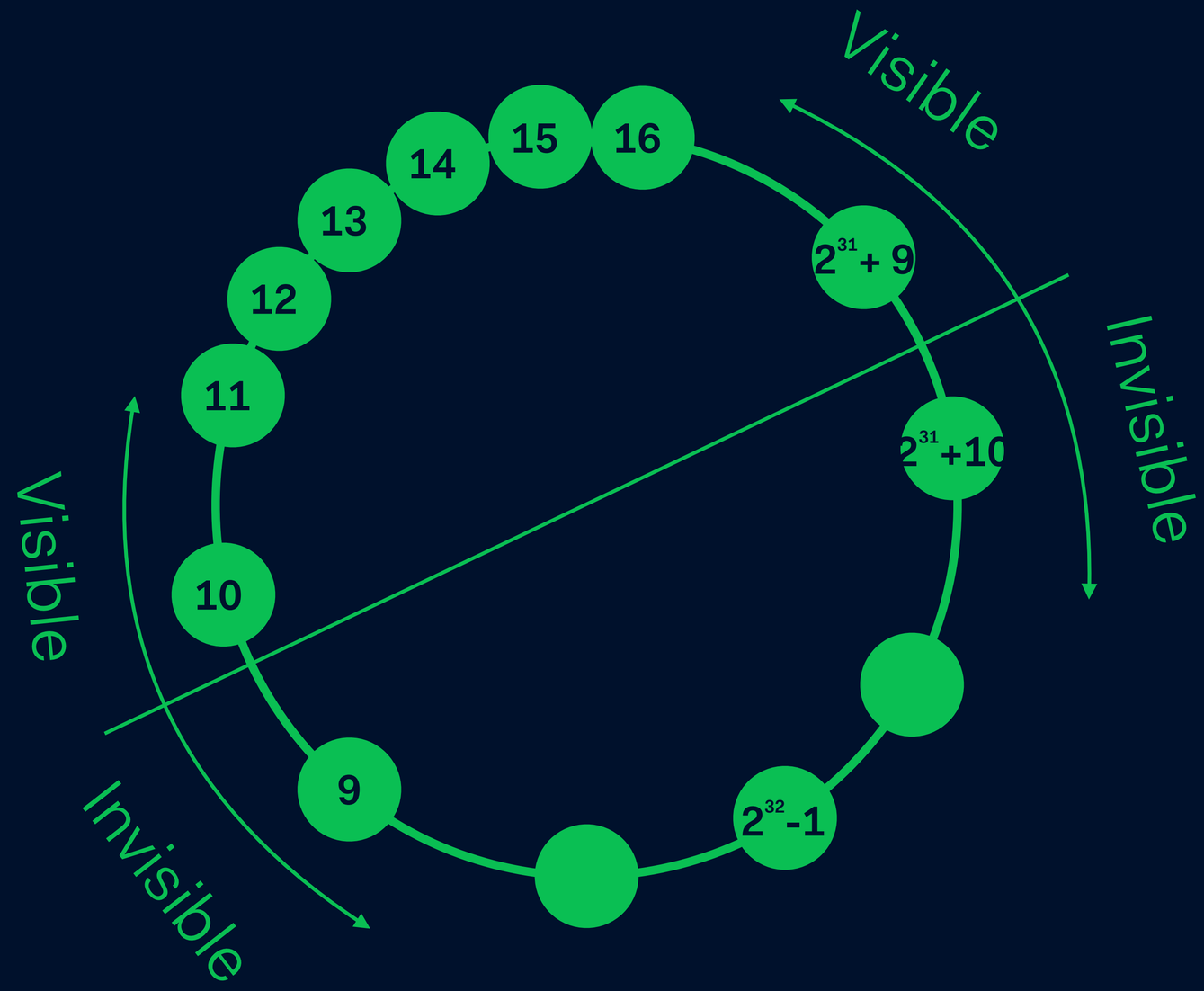
Transaction snapshot



Transaction snapshot



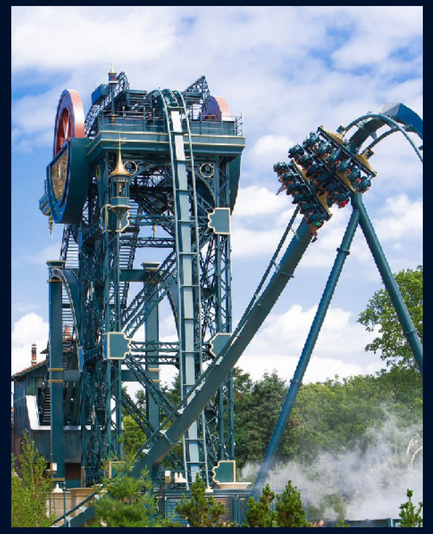
Transaction snapshot



Page level

Page header

t_min=10, t_max=102



t_min=101, t_max=105

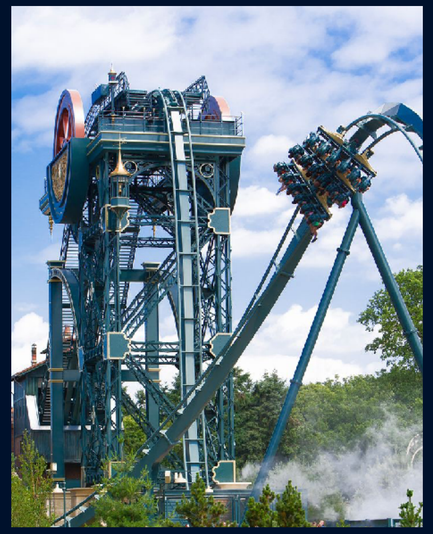
t_min=102, t_max=

Page footer

Page level

Page header

x_min=10, x_max=102



x_min=101, x_max=105

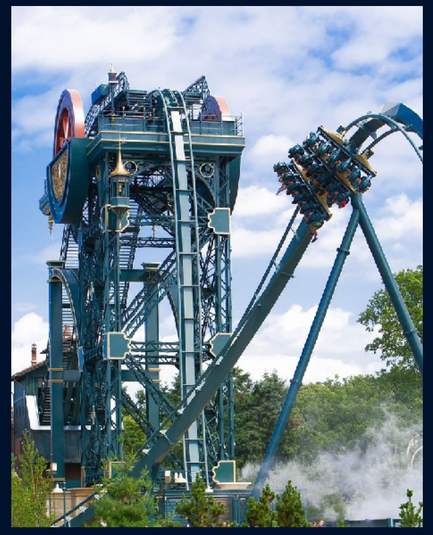
x_min=102, x_max=

Page footer

Page level

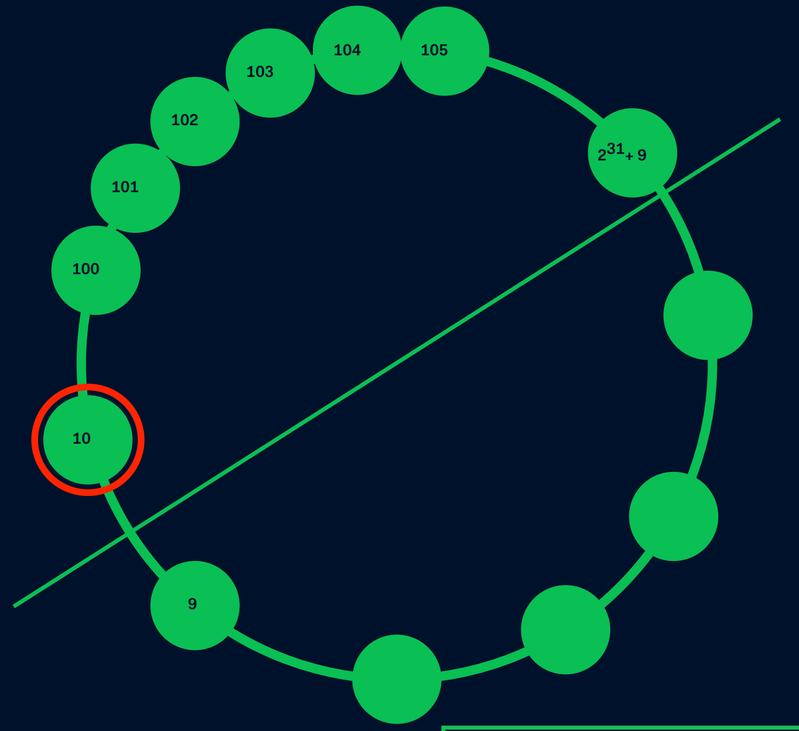
Page header

x_min=10, x_max=102



x_min=101, x_max=105

x_min=102, x_max=



Page footer

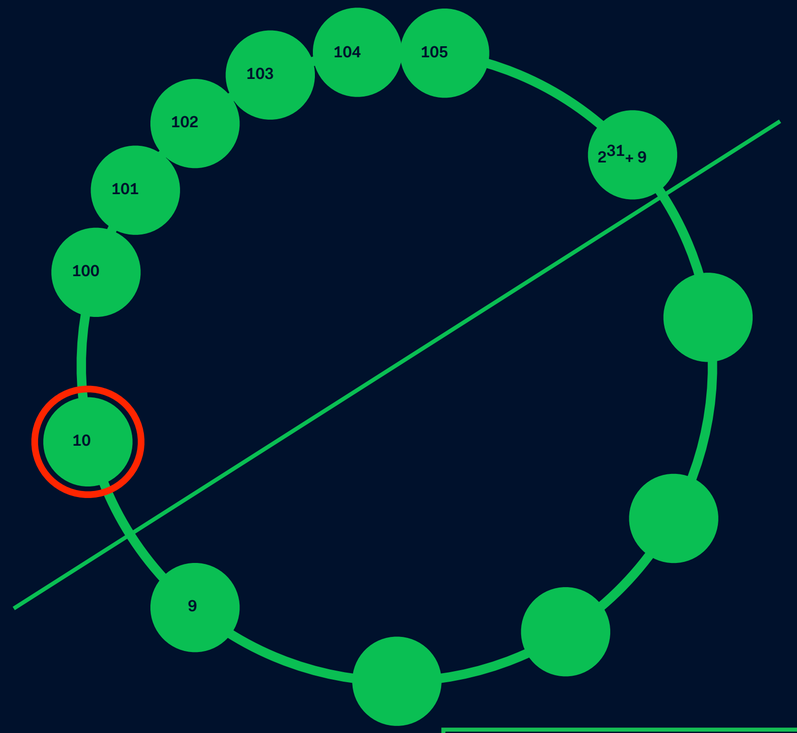
Page level

Page header

x_min=10, x_max=102

x_min=101, x_max=105

x_min=102, x_max=

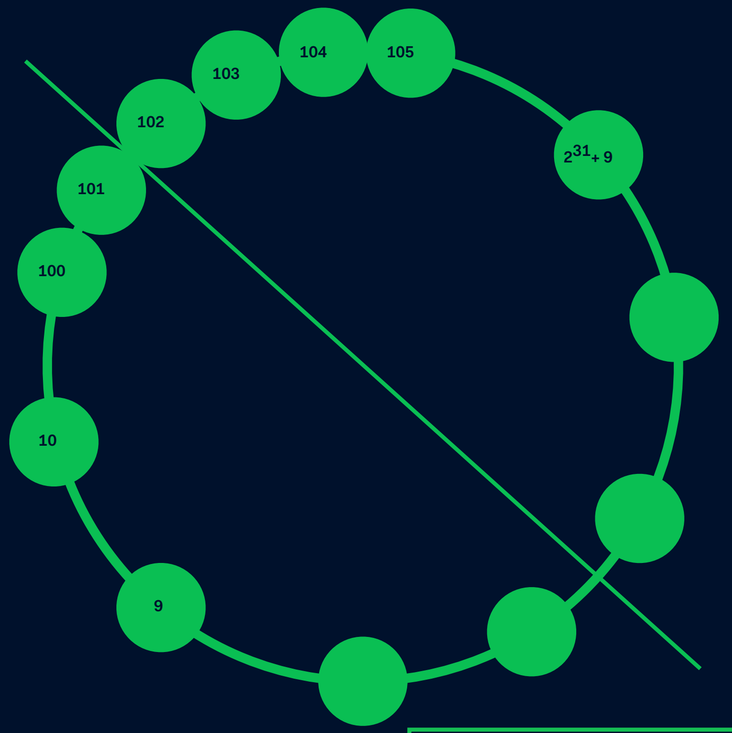


Page footer

Page level

Page header

x_min=102, x_max=

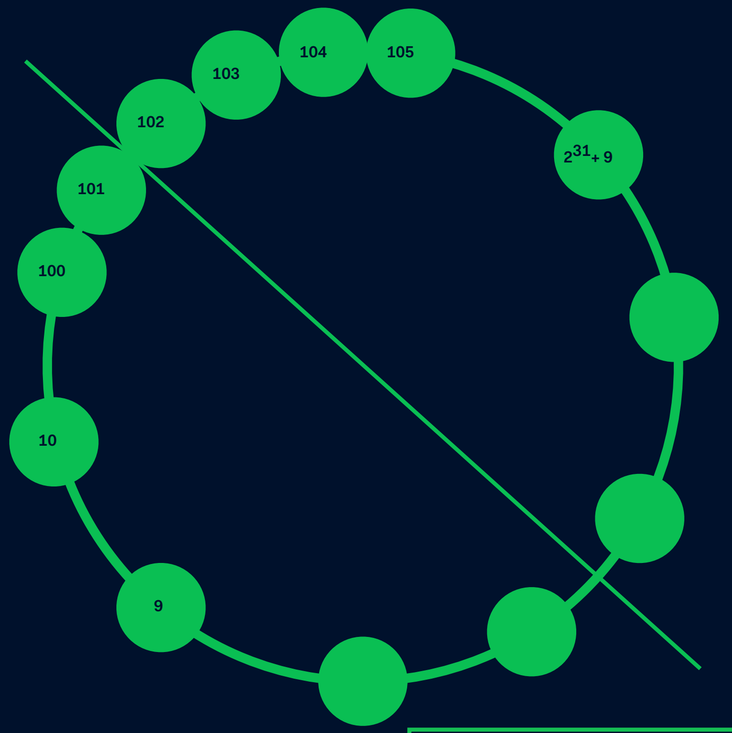


Page footer

Page level

Page header

x_min=102, x_max=frozen

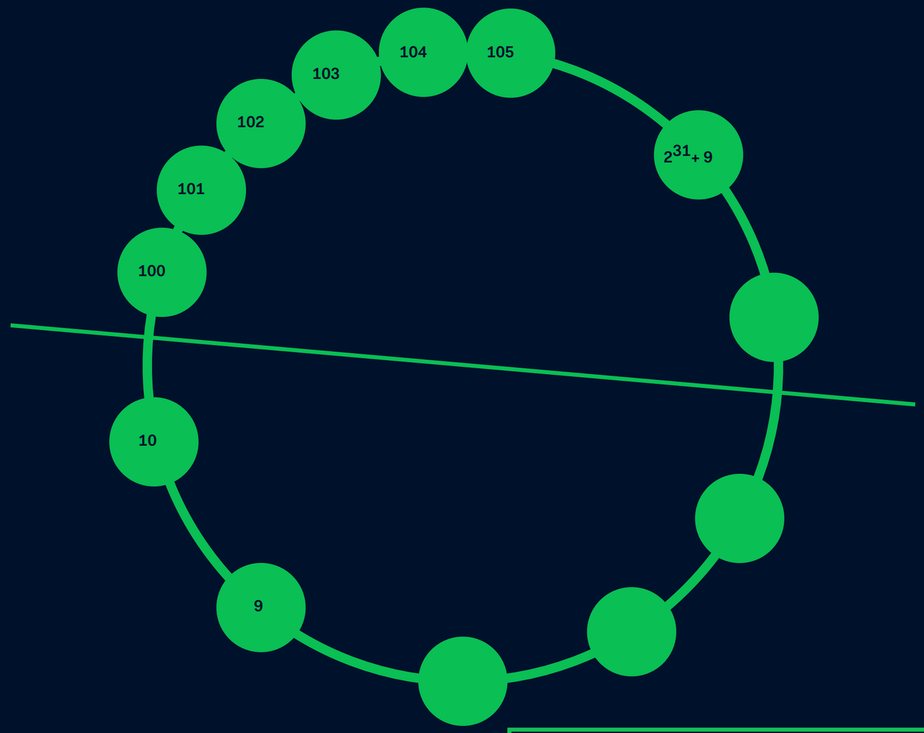


Page footer

Page level

Page header

x_min=102, x_max=frozen



Page footer

Using transactions

Using transactions

begin

Using transactions

```
CREATE OR REPLACE PROCEDURE test_tx(v_commit boolean)
FOR i IN 0..1000000 LOOP
   PERFORM * FROM t1;
   IF (v_commit) THEN
      commit;
   END IF;
END LOOP;
```

Call test_tx(false)	**00:03:068**
Call test_tx(true)	**00:05:455**

Using transactions

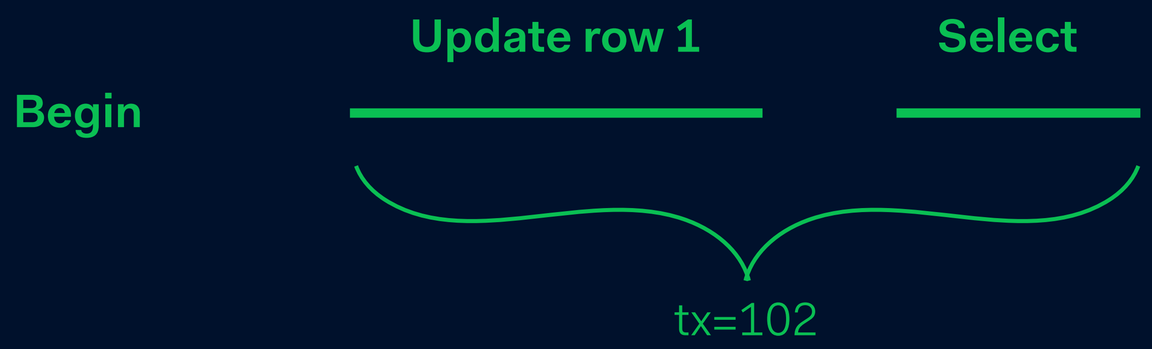
Using transactions

Using transactions

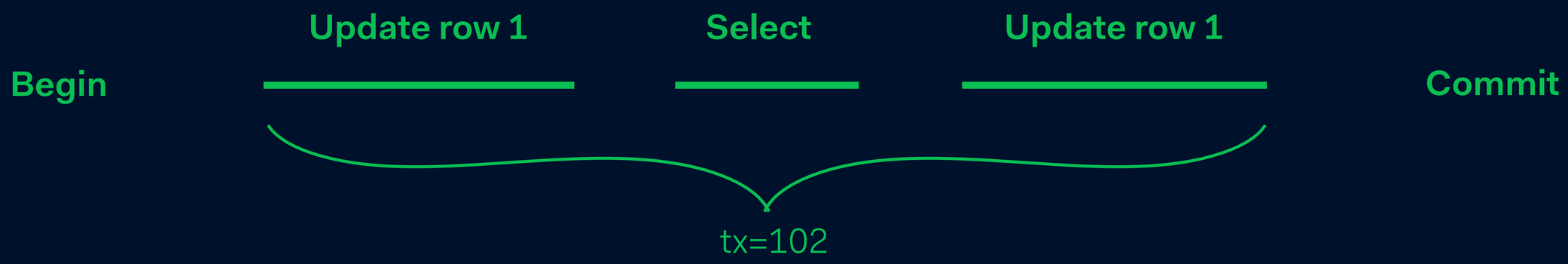
Using transactions

Using transactions

Using transactions



Using transactions



Page level

Page header

x_min=10, x_max=102

x_min=102, x_max=102

x_min=102, x_max=

Page footer

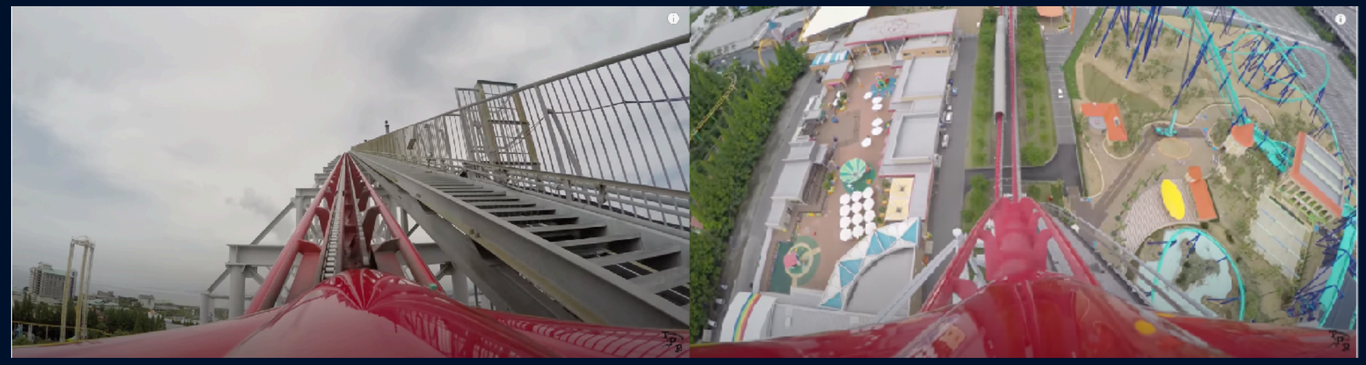
Locking

Locking

tx=100

Update

tx=102



Locking

Locking

Begin

Locking

Locking

Locking

Locking

Locking

Locking

Locking

Locking

```
Select * from table where flag is false for update;


Update table where flag is false;
```

Locking

Locking

```
Begin;
Select * from table where flag is false for update;


Update table where flag is false;
Commit;
```

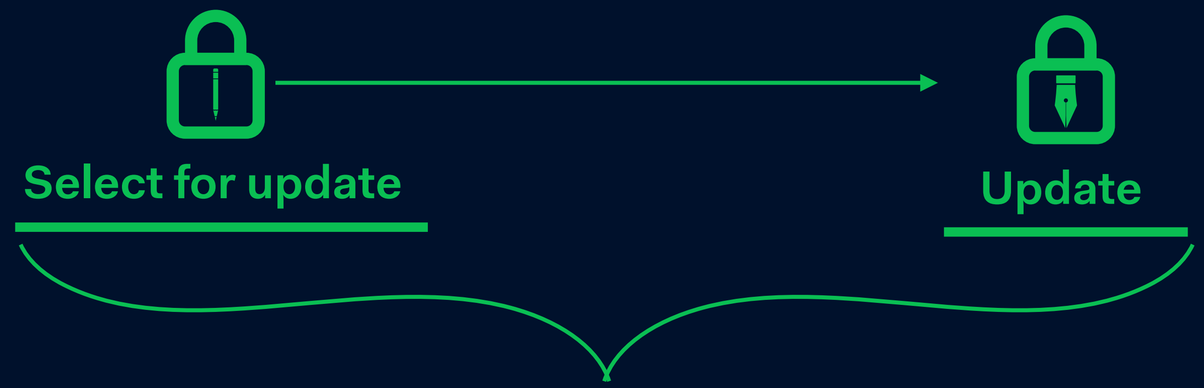
Locking

Begin

Locking

Begin;

Locking

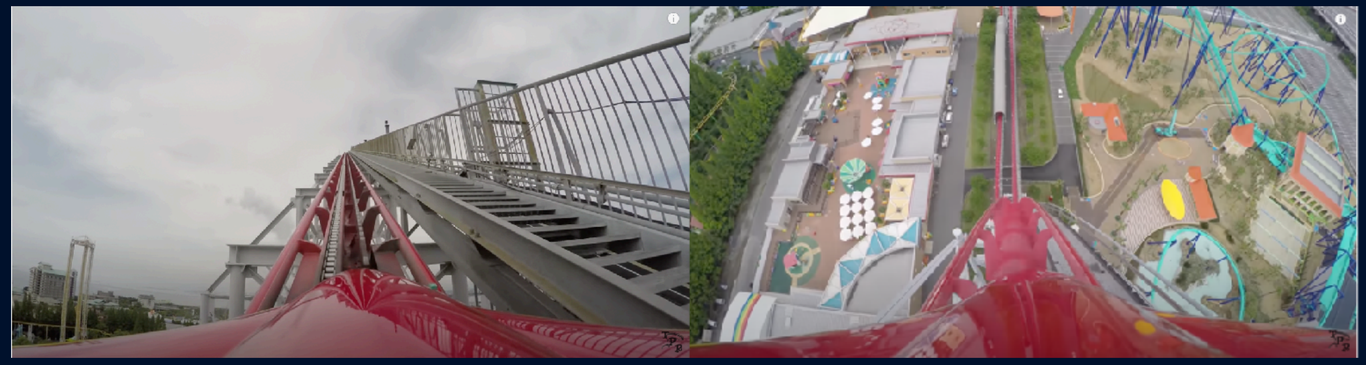


Locking

tx=100

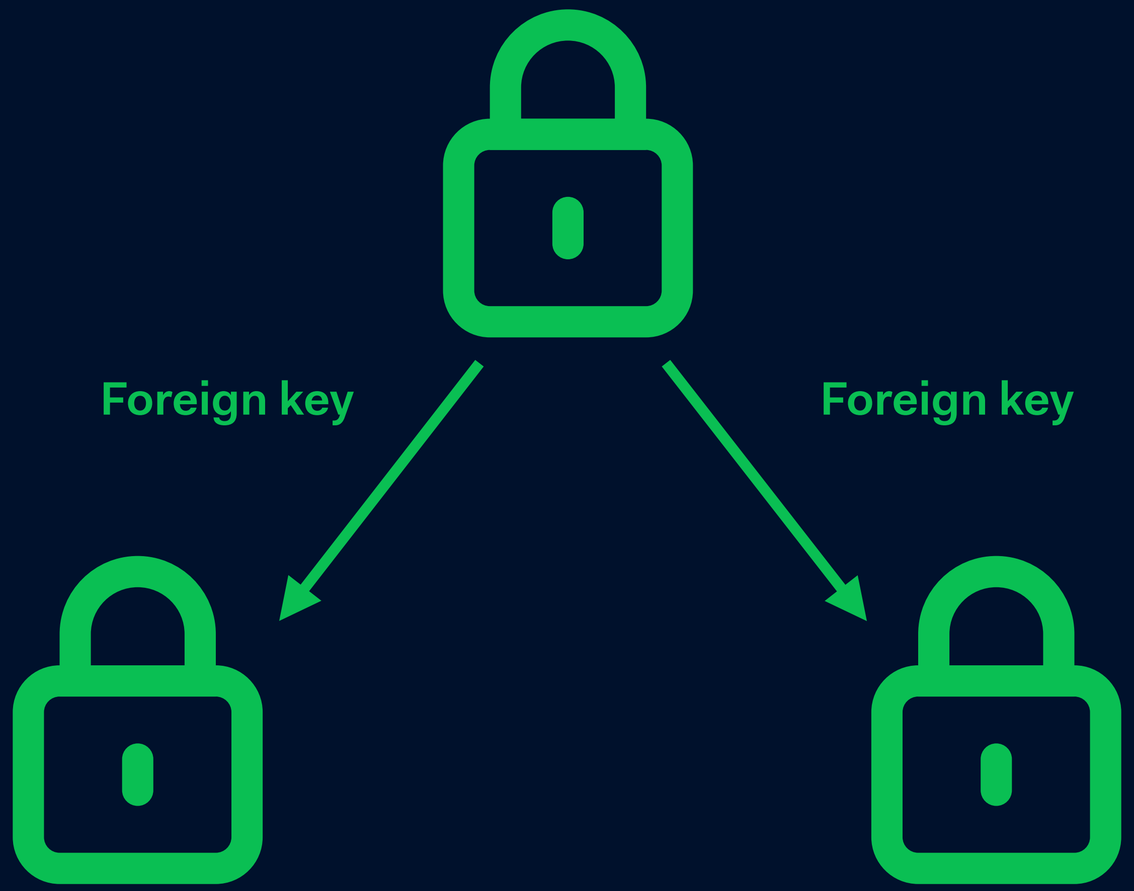
Update

tx=102

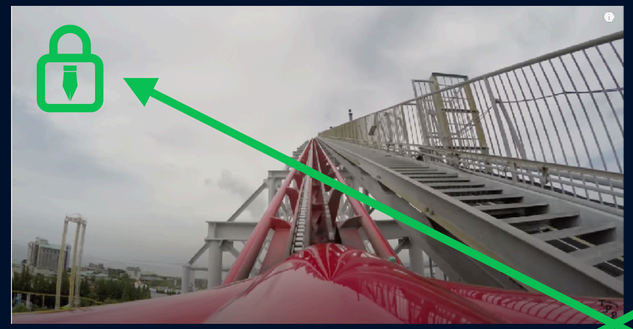


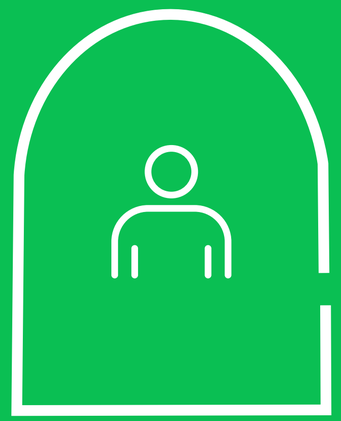
Locking

Locking



Locking

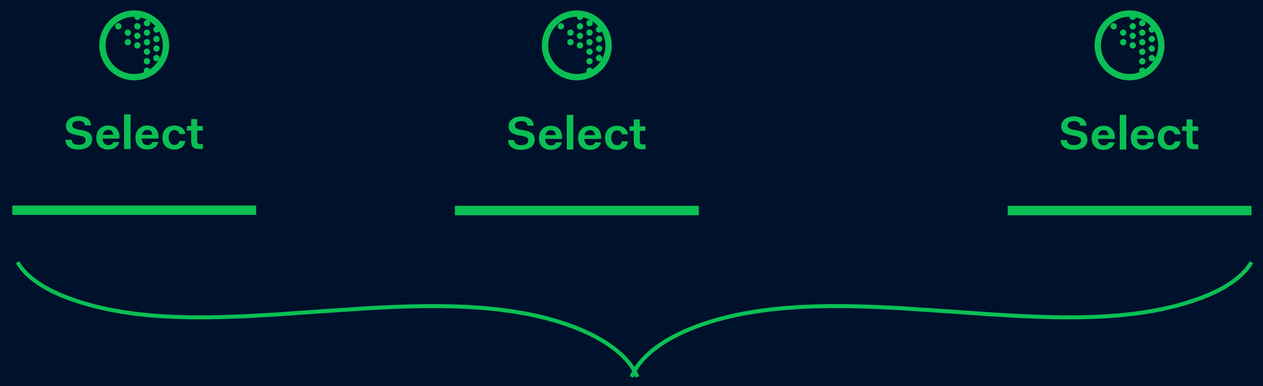




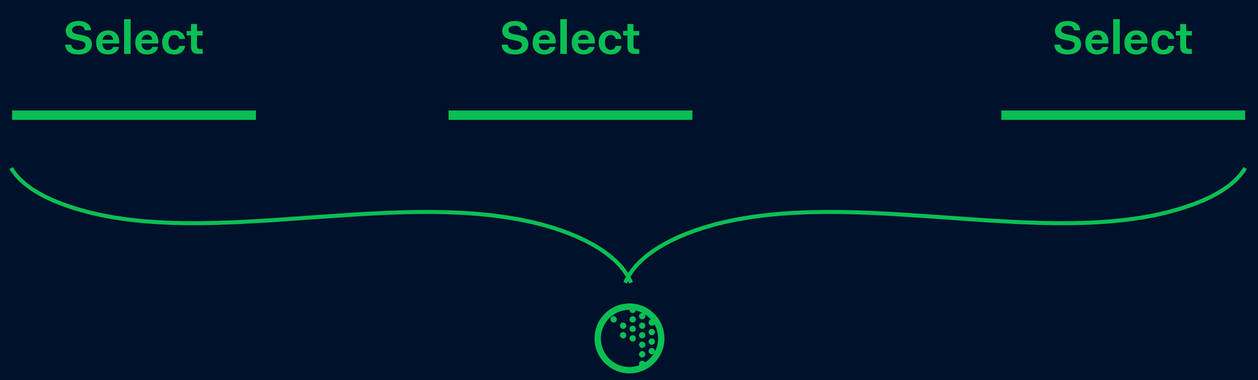
Isolation levels

Isolation levels

Isolation levels

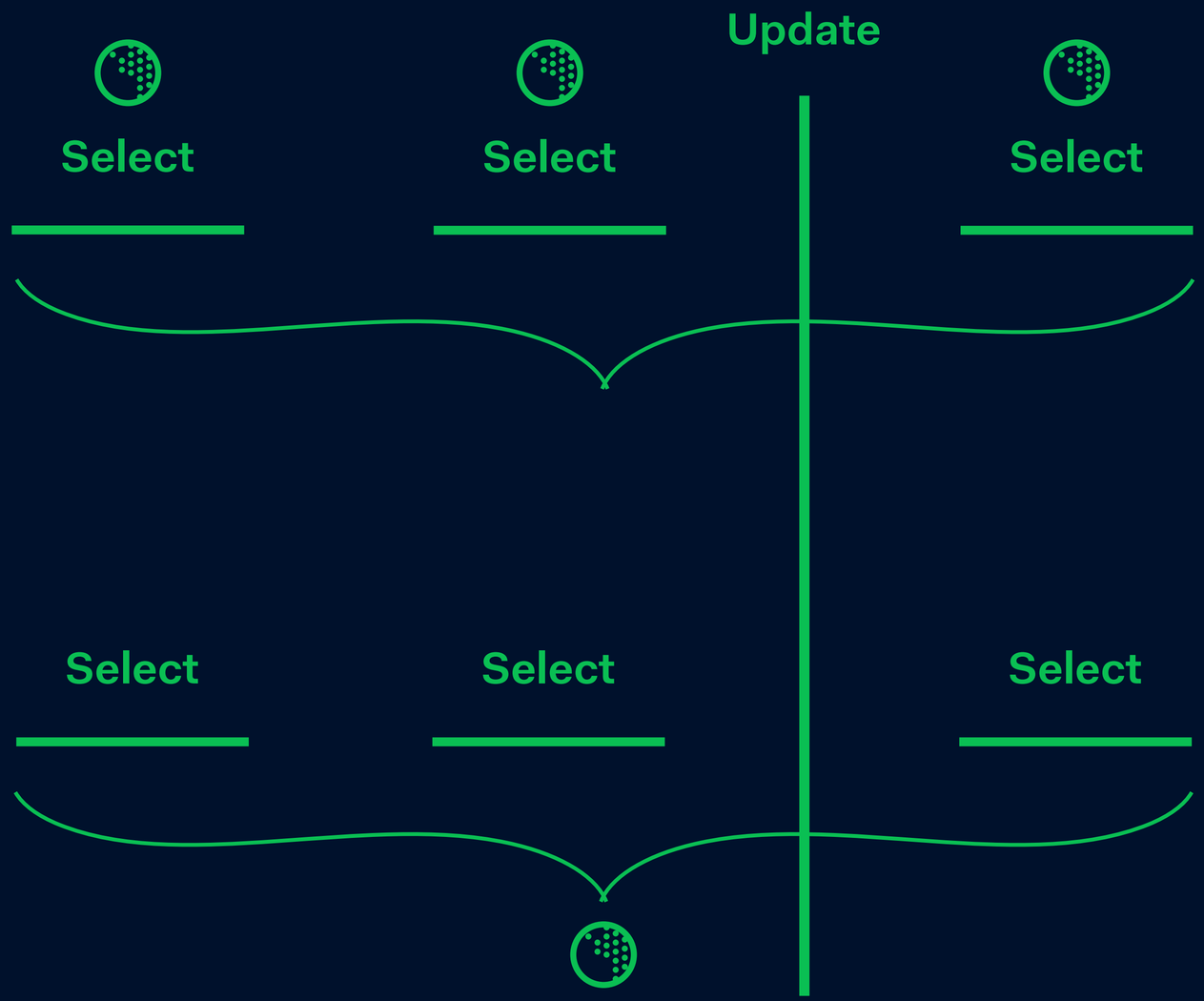


Read committed



Repeatable read

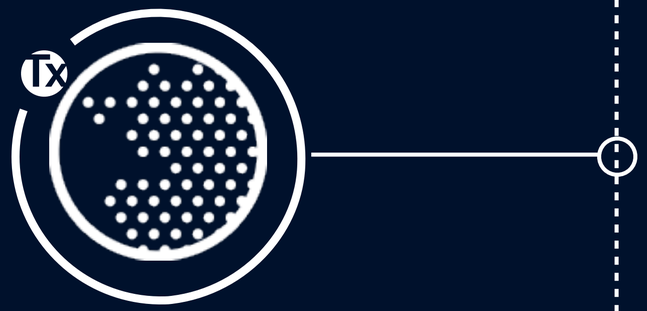
Isolation levels



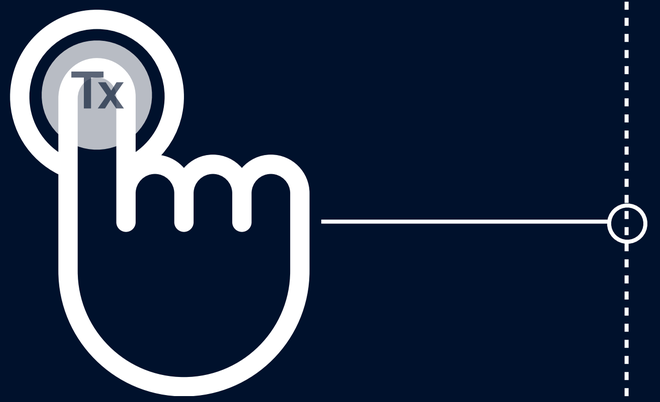
Read committed

Repeatable read

Learnings



Transaction space



Using transactions

Locks

Isolation levels

Questions

```
def withdraw(amt, user_id):
beginTxn()
    bal = readBalance(user_id)
    if (bal >= amt):
        writeBalance(bal – amt, user_id)
commit()
```