



PROJECT #3 FINAL

CONSTRAINTS

Shimin Wang | Lei Qi | Ruirui Xiang

OUTLINE

- ▶ Project Status
- ▶ Demo
- ▶ Tests
- ▶ Code Quality Assessment
- ▶ Future Work

PROJECT STATUS

$125\% - 15\% =$

100 %

CHECK

- ▶ Multi-Column
- ▶ Comparison
- ▶ Arith&Logical Expression

100 %

NOT NULL

UNIQUE

PRIMARY KEY

100

ALTER

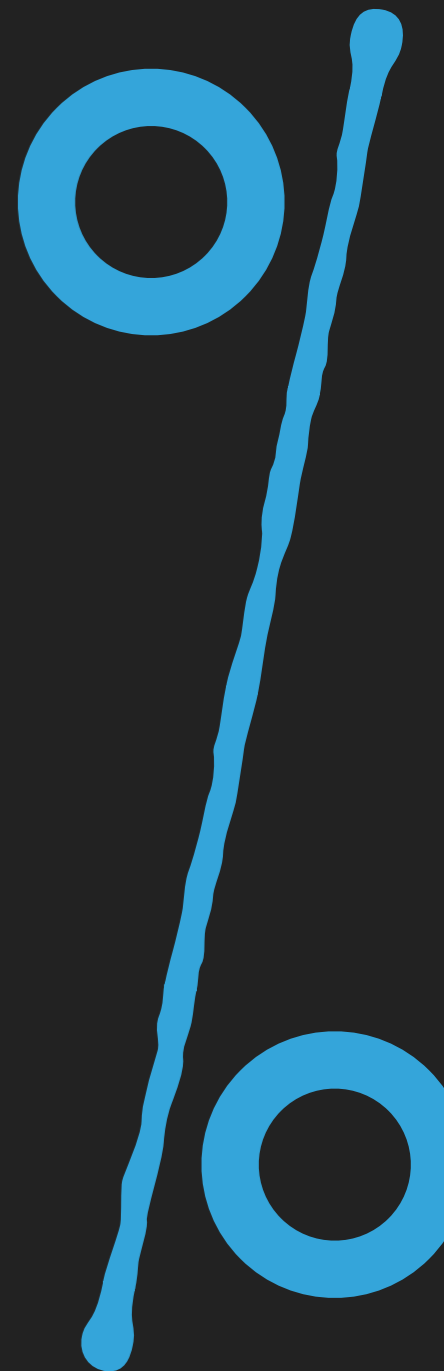
- ▶ SET NOT NULL
- ▶ DROP NOT NUL
- ▶ ADD UNIQUE
- ▶ DROP UNIQUE

90 %

FC

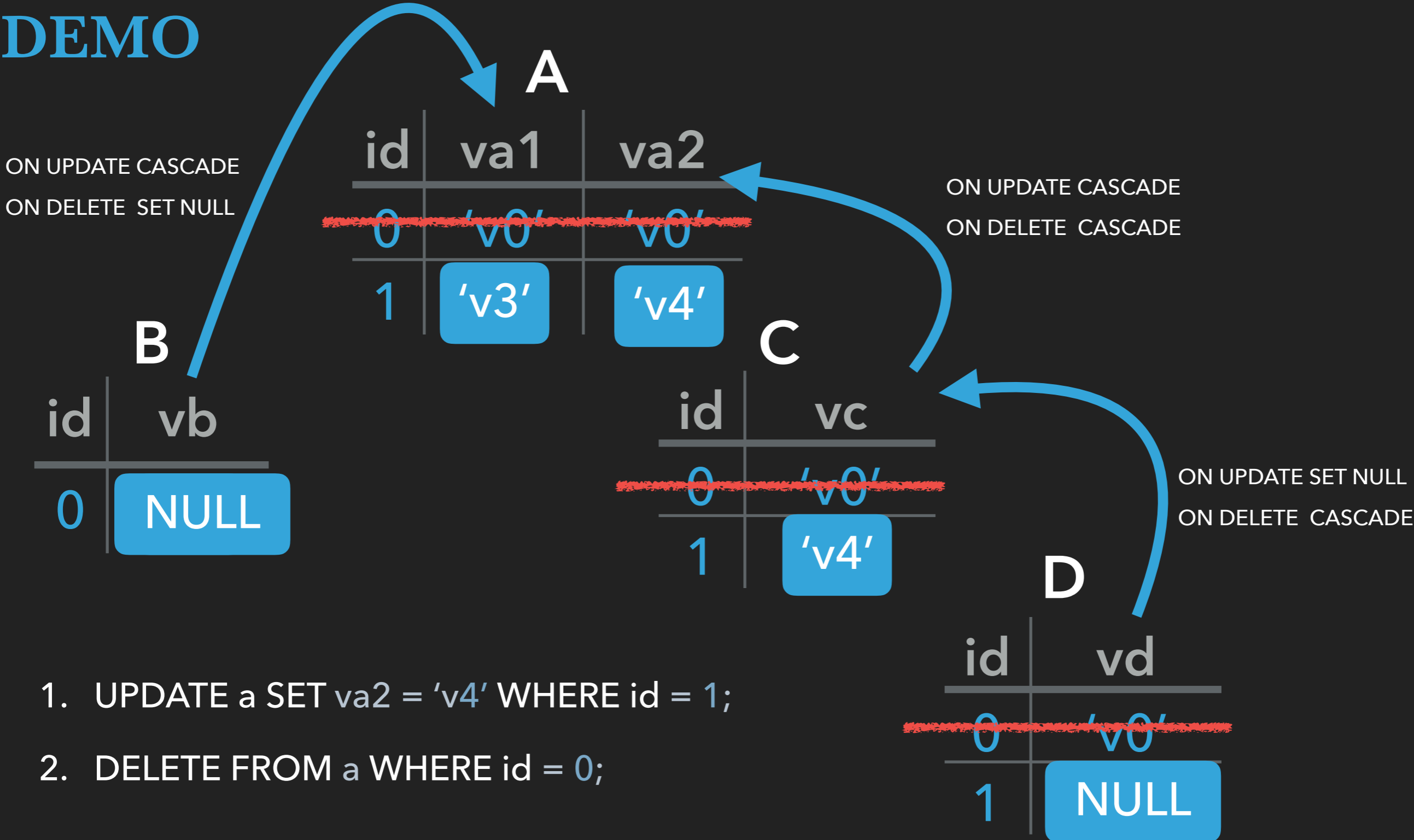
On e:

NO CT



IMPLEMENTATION OF CONSTRAINTS IN PELOTON

DEMO

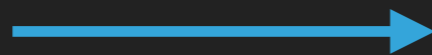


DEMO

~~1.Insert (0, 1, 2)~~

2.Insert (8, 1, 2)

~~3.Insert (1, 2, 1)~~



C1 CHECK (a * b > c + 1)

| a | b | c |
|---|---|---|
| 8 | 1 | 2 |

IMPLEMENTATION OF CONSTRAINTS IN PELOTON

DEMO

1. Insert (1, 'mavis', 100, 99)

~~2. Insert (2, 'shimin', 10, 1)~~

~~3. Insert (2, 'shimin', 100, 101)~~

4. Insert (2, 'shimin', 120, 101)

~~5. Insert (3, 'leiqi', 30, 10)~~

C2

PRIMARY KEY prod_no ,
CHECK (price > 10) ,
CONSTRAINT valid_discount CHECK
(price > discounted_price
AND discounted_price > 0)

| prod_no | name | price | discounted |
|---------|--------|-------|------------|
| 1 | mavis | 100 | 99 |
| 2 | shimin | 120 | 101 |

! PRICE > 10

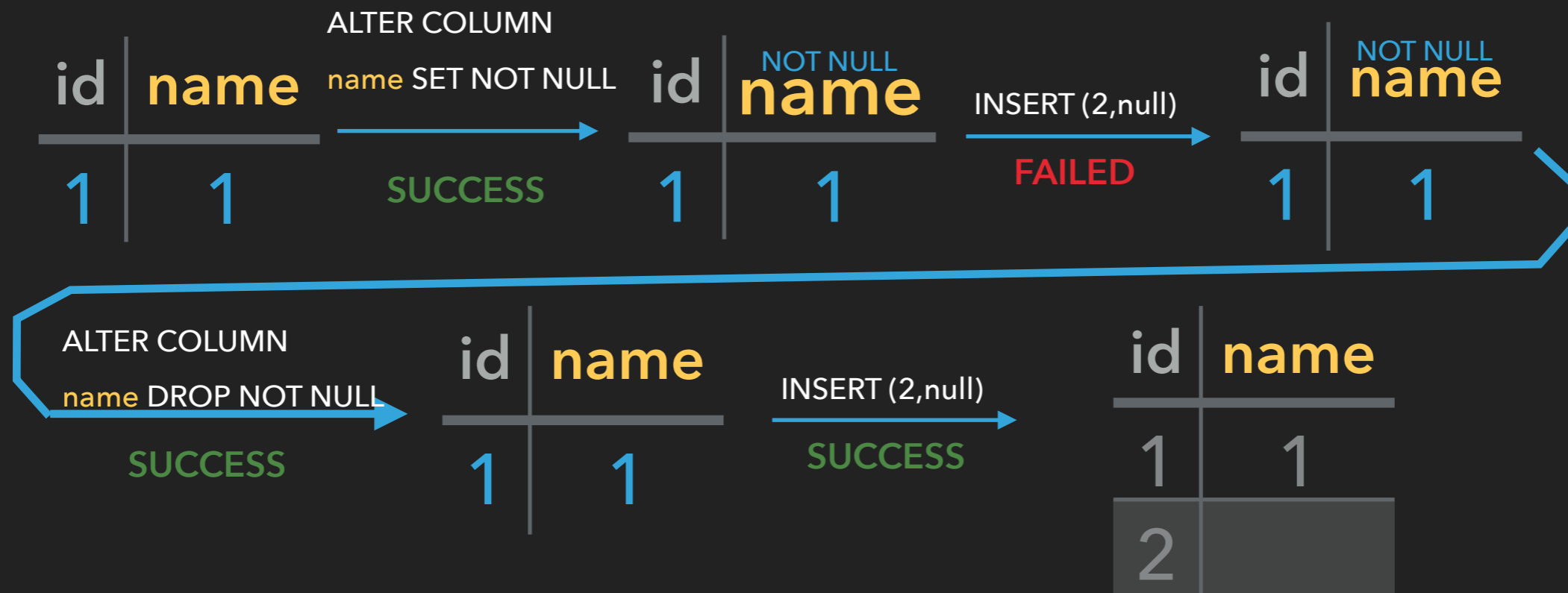
! PRICE >
DISCOUNTED

! XXX AND
DISCOUNTED > 0

IMPLEMENTATION OF CONSTRAINTS IN PELOTON

DEMO

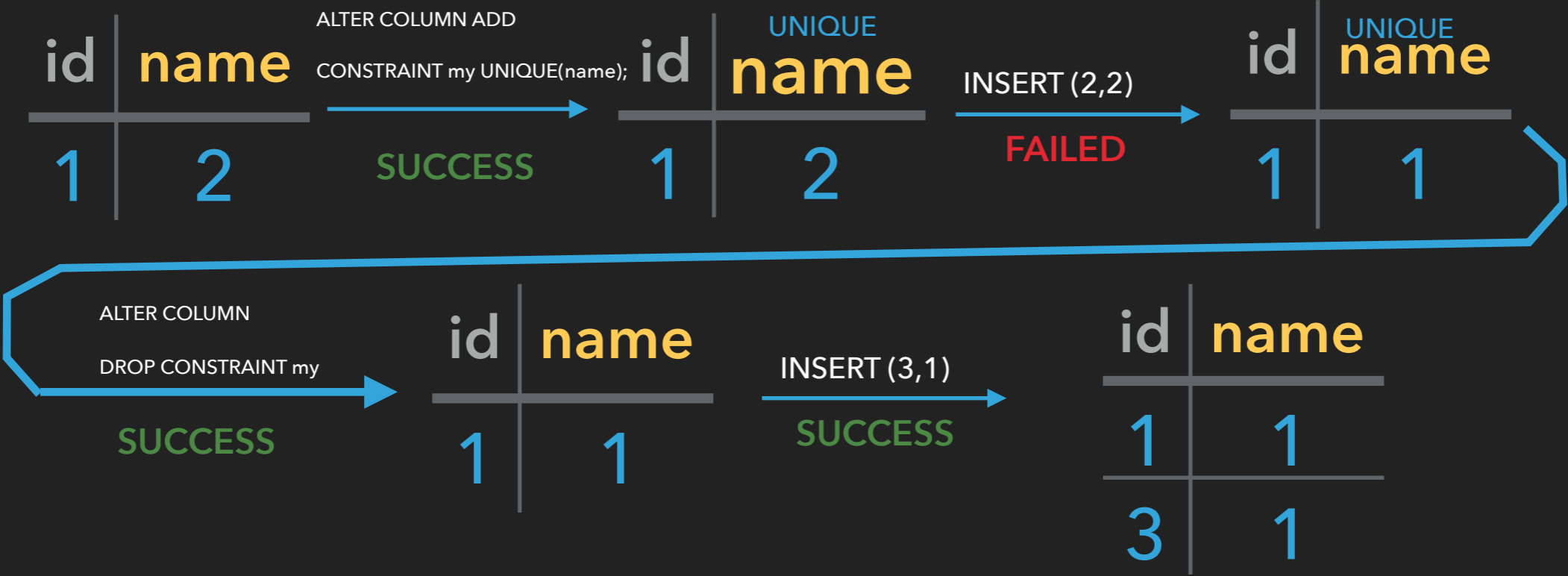
Set/Drop NOT NULL:



IMPLEMENTATION OF CONSTRAINTS IN PELOTON

DEMO

Set/Drop UNIQUE:



DESIGN

constraints declaration

POSTGRES HEAP



BRIDGE::DDLTABLE



STORAGE::DATA_TABLE
CATALOG::SCHEMA

constraints checking

EXECUTOR



TXN_MANAGER

TESTS

tests/catalog/constraints_test.cpp

```
#define NOTNULL_TEST  
  
#define PRIMARY_UNIQUEKEY_TEST  
  
#define FOREIGNN_KEY_INSERT_TEST  
#define FOREIGNN_KEY_RESTRICT_DELETE_TEST  
#define FOREIGNN_KEY_CASCADE_DELETE_TEST  
#define FOREIGNN_KEY_SETNULL_DELETE_TEST  
#define FOREIGNN_KEY_RESTRICT_UPDATE_TEST  
#define FOREIGNN_KEY_CASCADE_UPDATE_TEST  
#define FOREIGNN_KEY_SETNULL_UPDATE_TEST  
  
#define DROPSETNOTNULL_TEST  
#define DROPUNIQUE_TEST  
#define SETUNIQUE_TEST  
  
#define CHECK_CONSTRAIN_TEST
```

scripts/testing/constraints



check.sql



alterconstraint.sql



foreignkey.sql

CODE QUALITY ASSESSMENT

▶ Pros:

1. Good decoupled modular Design, full-covered Tests
2. Following good code style, like always using `unique_ptr`

▶ Cons:

1. Need Stronger Tests For Concurrent Txn
2. Not yet support concurrent ALTER TABLE

FUTURE WORK

- ▶ **Several Improvements and FIXME:**
 1. Support multicolumn foreignKey;
 2. Constraint Checking for in-place update in MVCC;
 3. Support add/drop check constraints;
 4. Support cascade behavior for alter constraints
 5. Support Default
- ▶ **Performance:**
 1. To add stress test for our constraints;
 2. Comparing our implementation with Postgres