

Migrating from PostgreSQL to MySQL ... without downtime

Matthew Graham
Percona Live NYC 2011

The Etsy logo is displayed in white text on an orange rectangular background with rounded corners.

Etsy

The Etsy logo is centered within a rounded rectangular orange box. The word "Etsy" is written in a white, serif font.

Global marketplace for
buying and selling handmade goods

Total

- 8 million members
- 800k active shops
- 1 billion page views / month

April 2011

- \$38.7 million goods sold
- 2 million items sold
- 1.8 million new items listed
- 390k new members joined

The Etsy logo is centered within a rounded orange rectangle. The word "Etsy" is written in a white, serif font.

Global marketplace for
buying and selling handmade goods

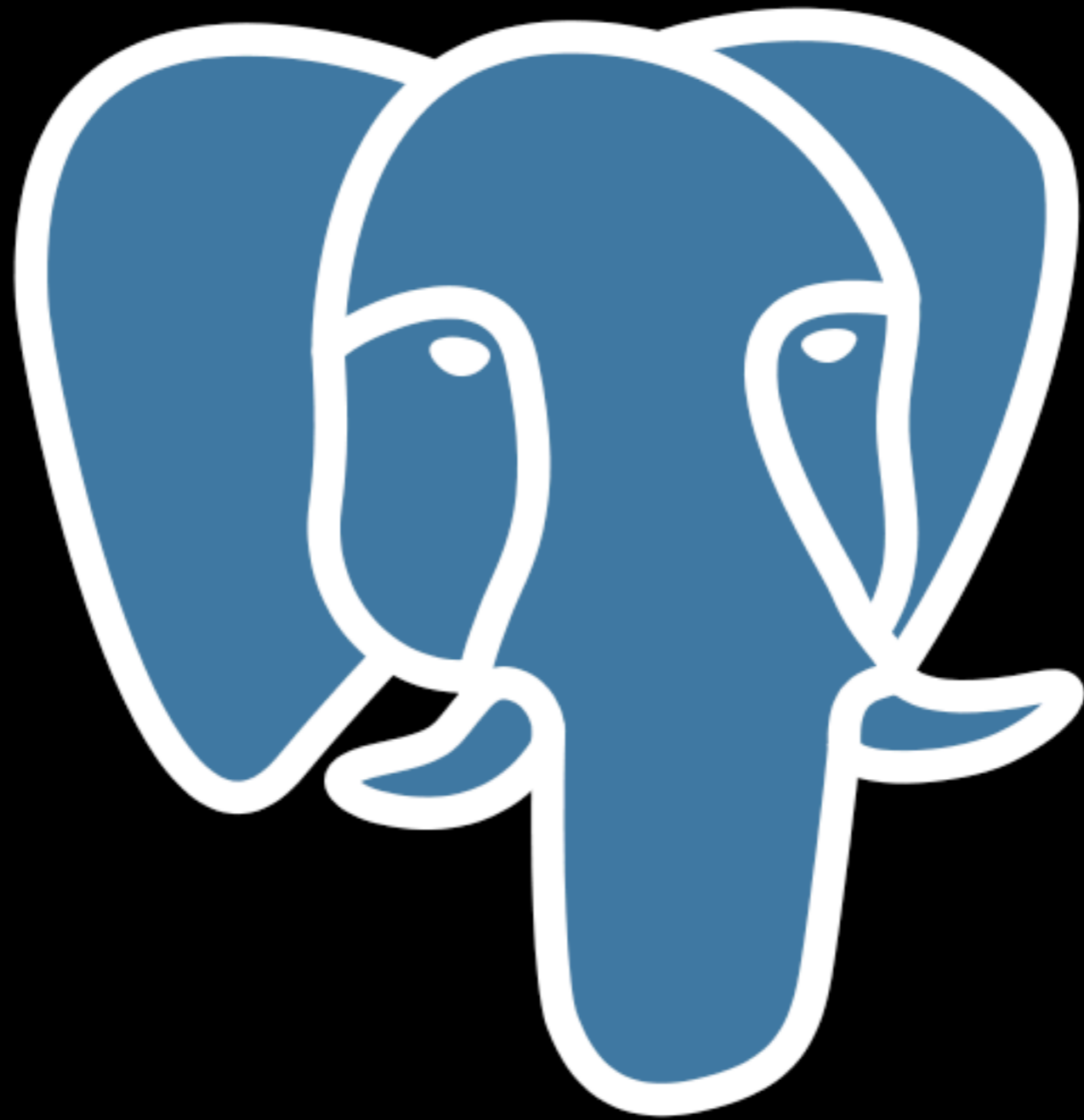
Total

- 8 million members
- 800k active shops
- 1 billion page views / month

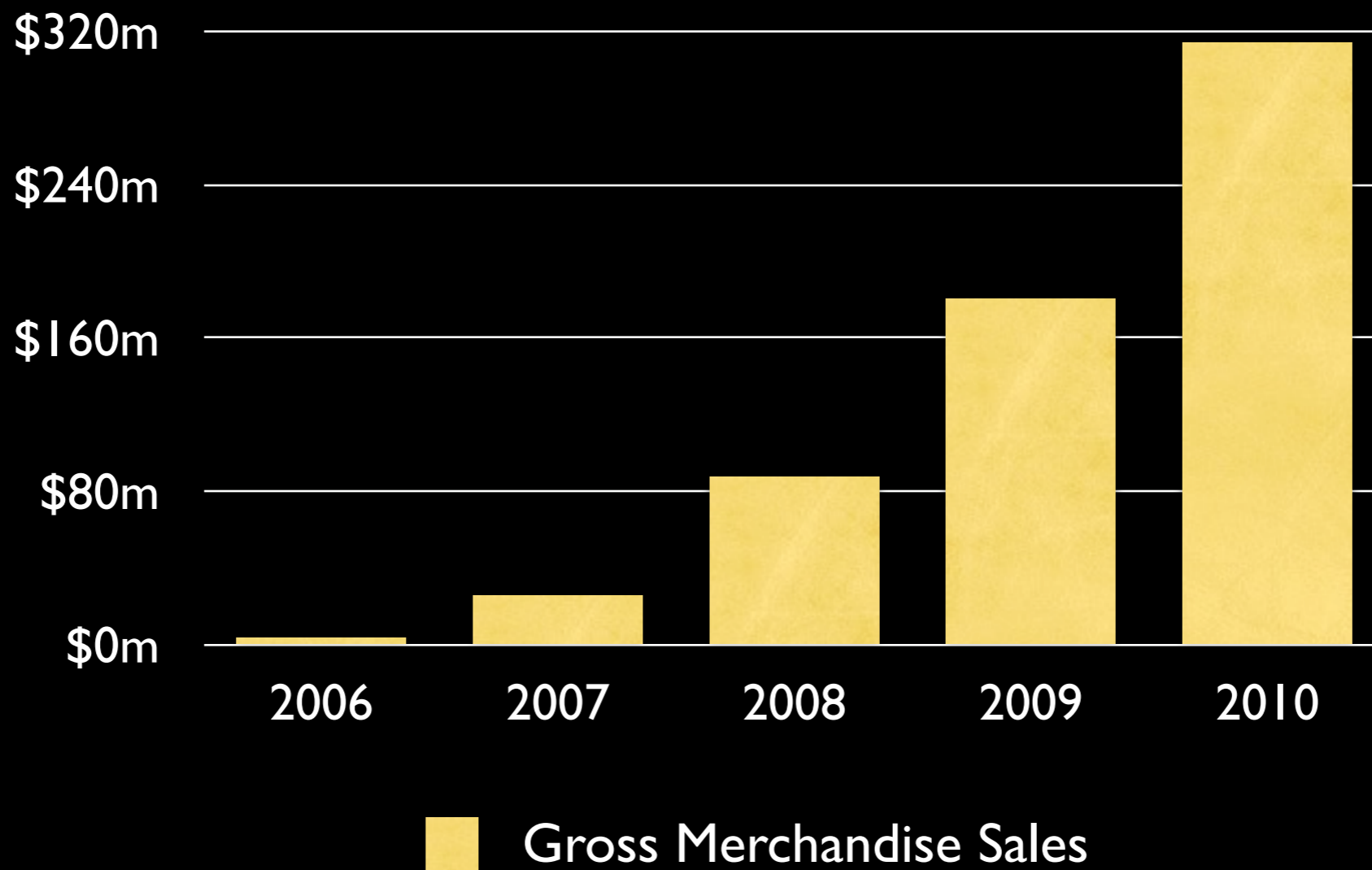
April 2011

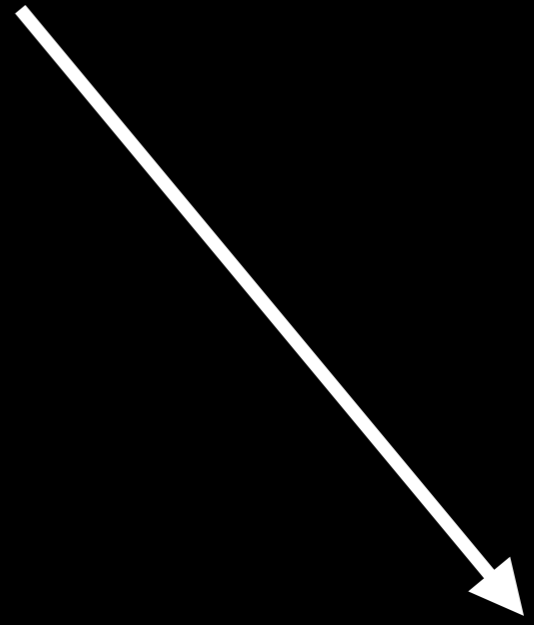
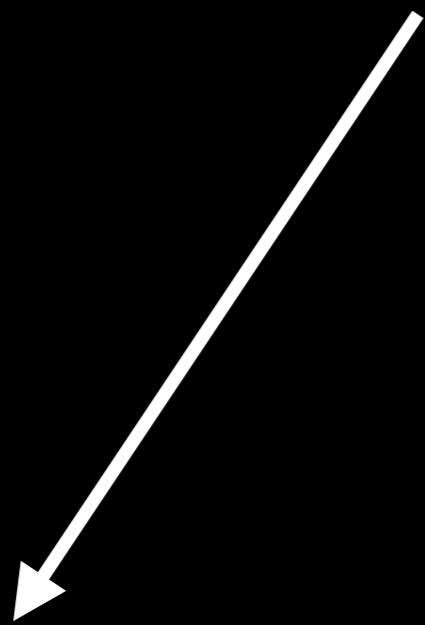
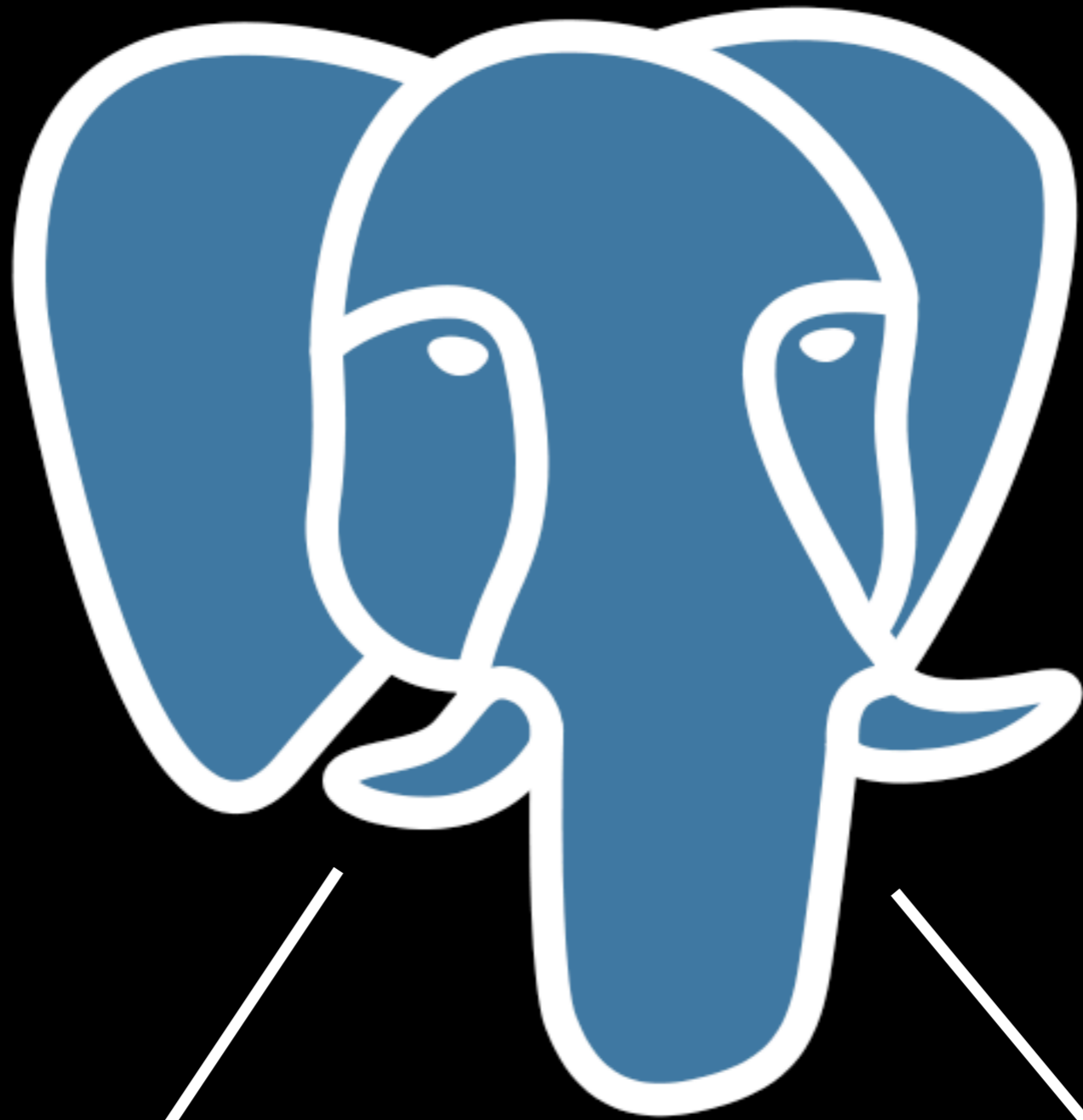
- \$38.7 million = **\$900 / min**
- 2 million items sold
- 1.8 million new items listed
- 390k new members joined

Downtime = -\$



Etsy





Reasons To Migrate

- Horizontal Scaling
- Reduce Types of Databases
- Licensing Costs
- Functional Partitioning
- Schema Refactor

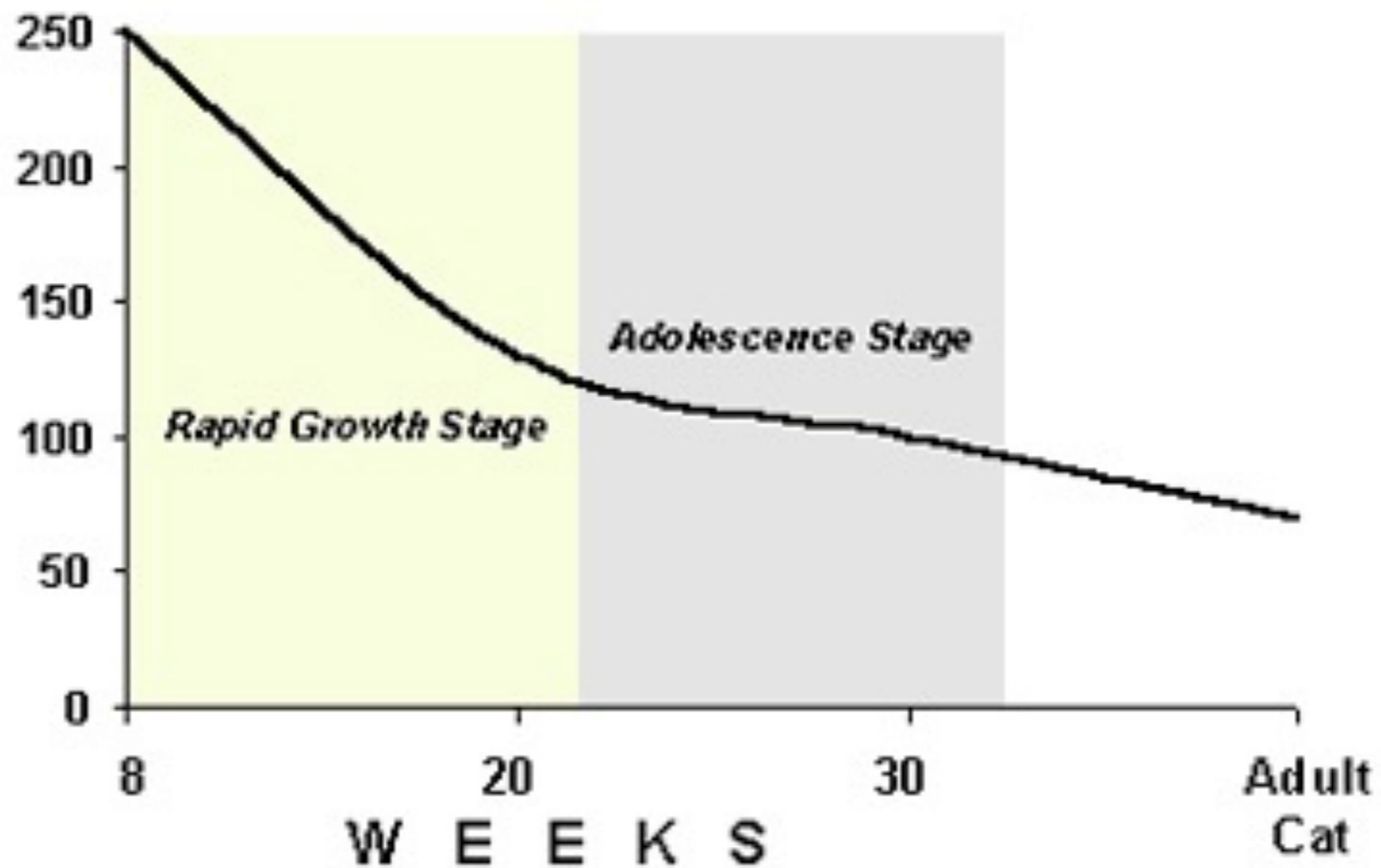
Deciding to Switch

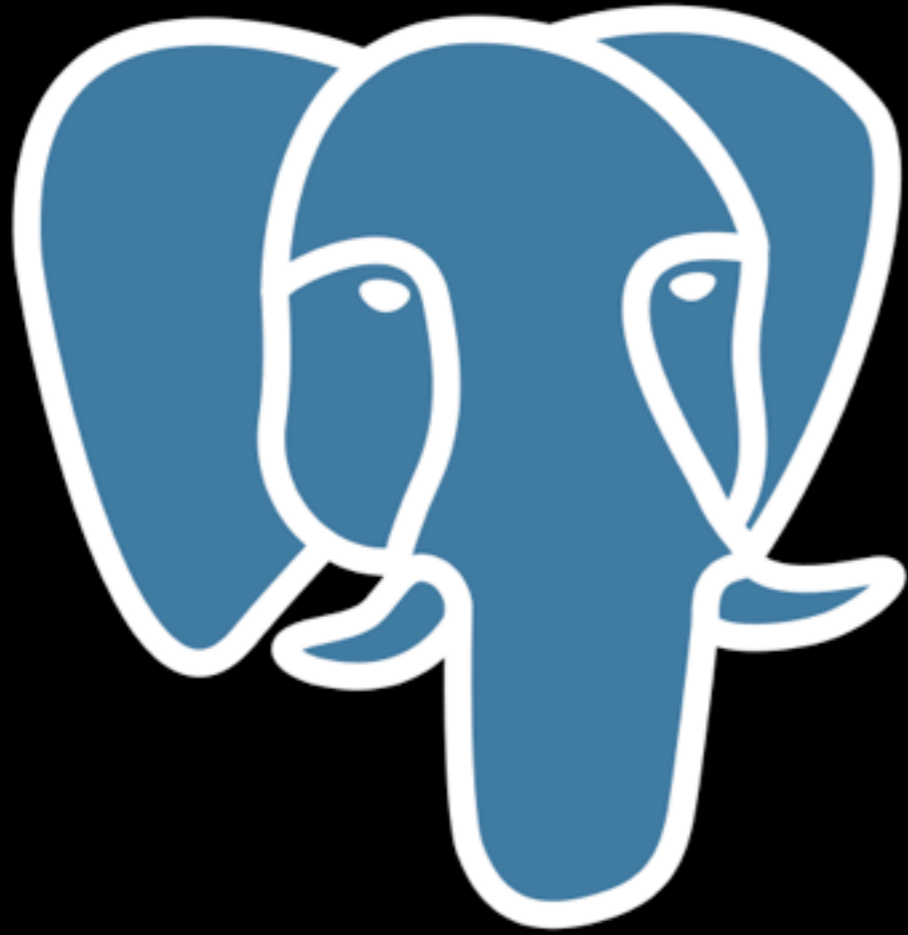
expected quality

- transition

net quality > actual quality

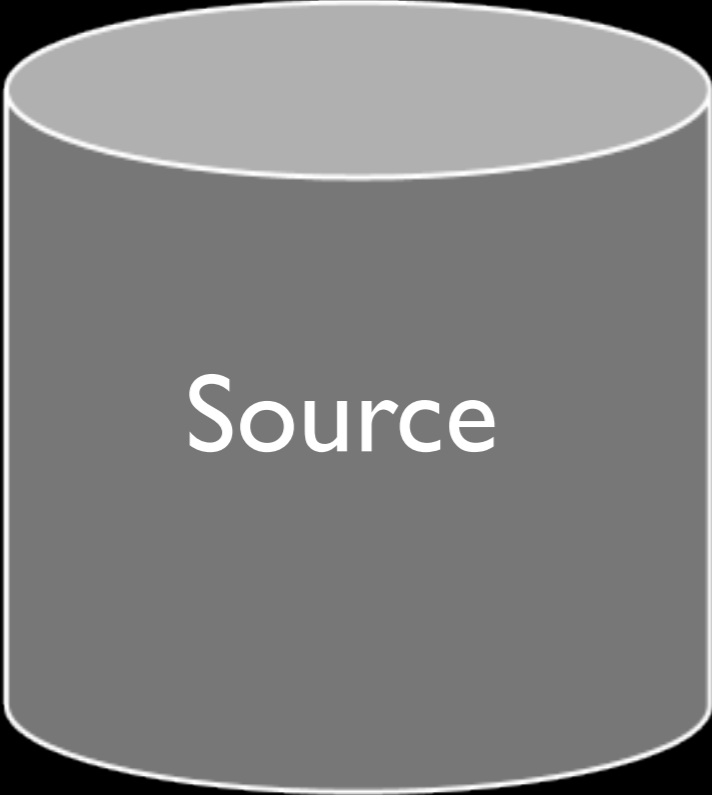
Energy Requirements of Kittens and Adult Cats





mongoDB

ORACLE®



Why MySQL®? 



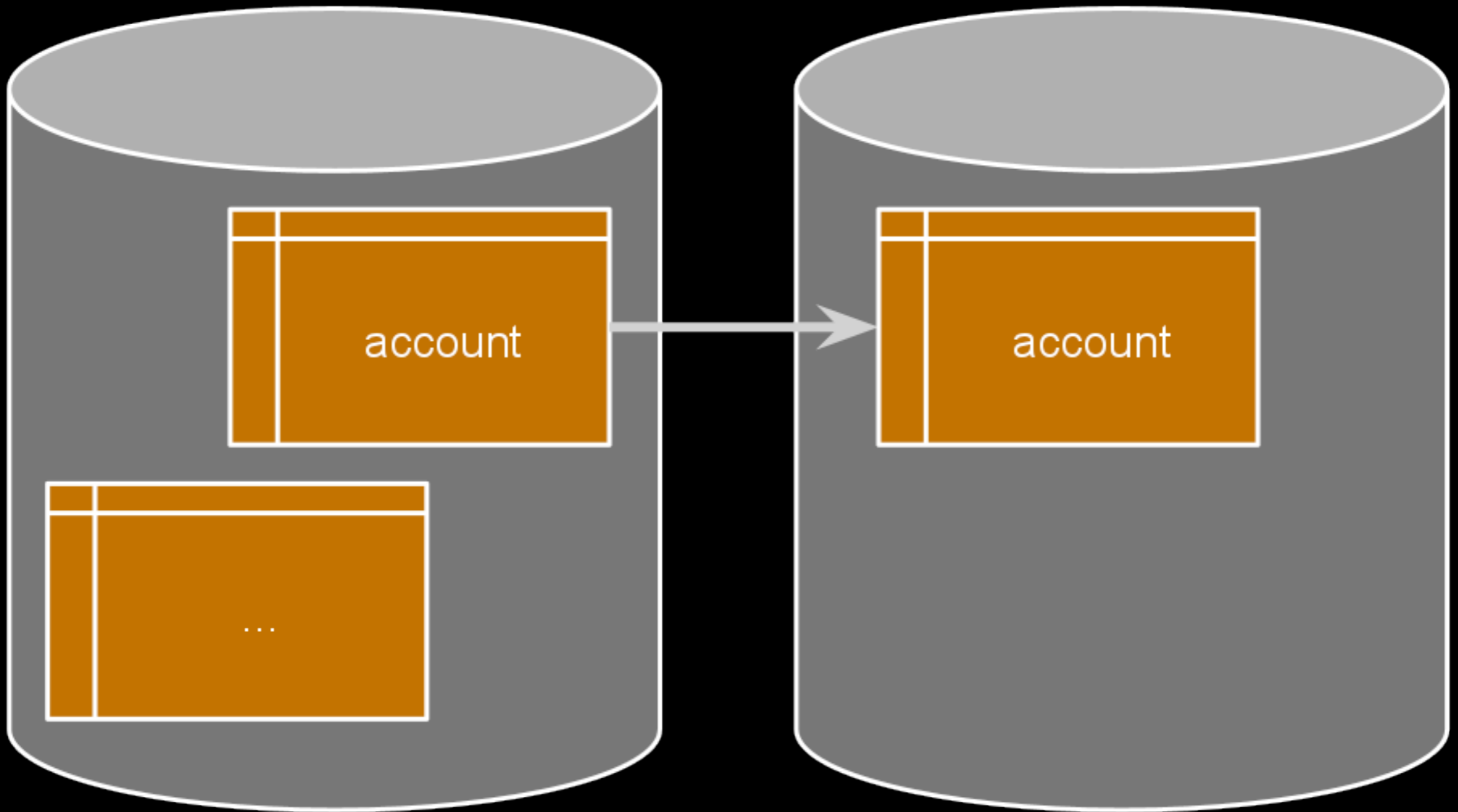
Foundational Principles

- Migrate One Table at a Time
- Progressive Ramp Up
- Data Duplication During Transition

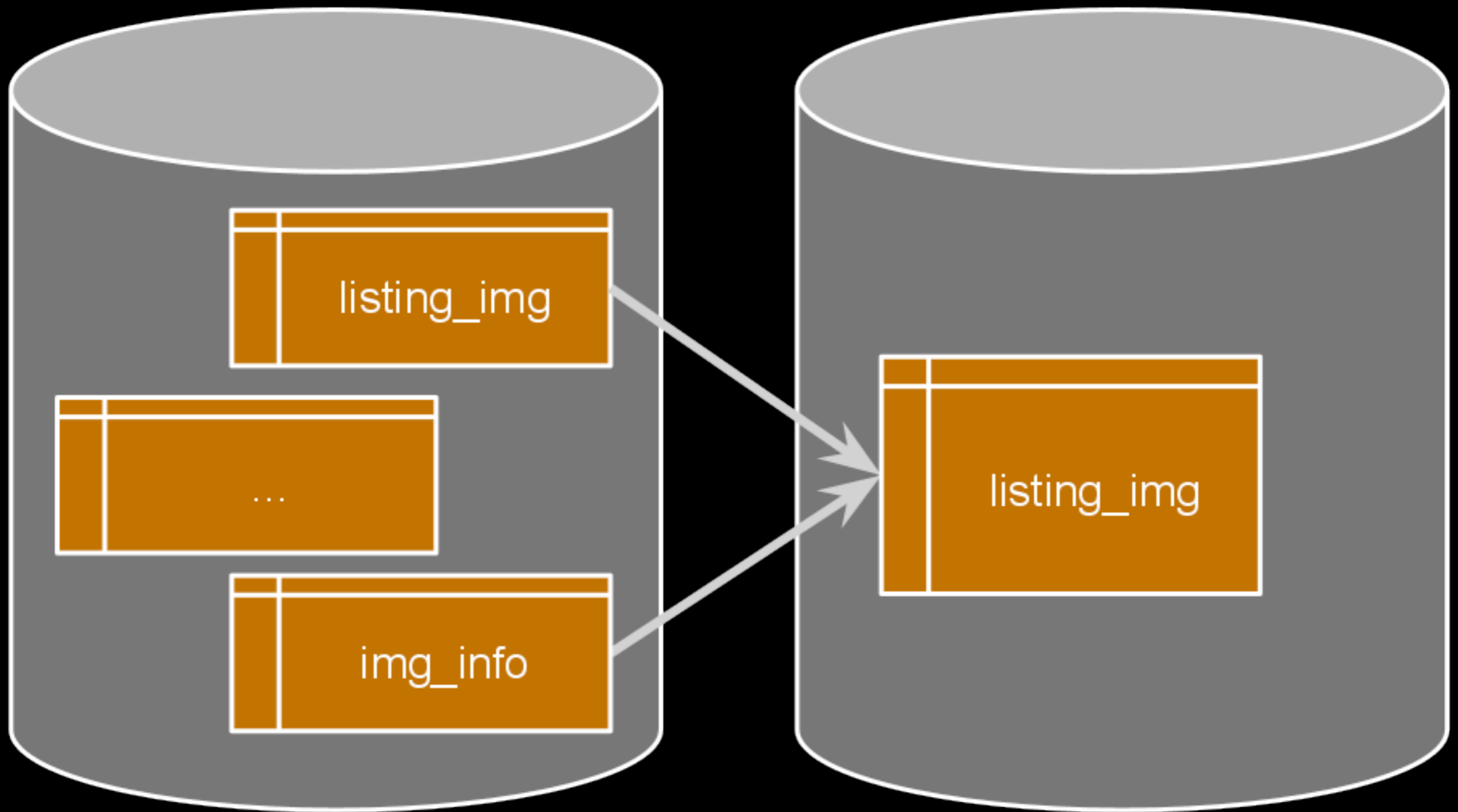
5 Steps Per Table

1. Create Target Tables
2. Tee Writes
3. Backfill
4. Read from Target Tables
5. Wrap Up

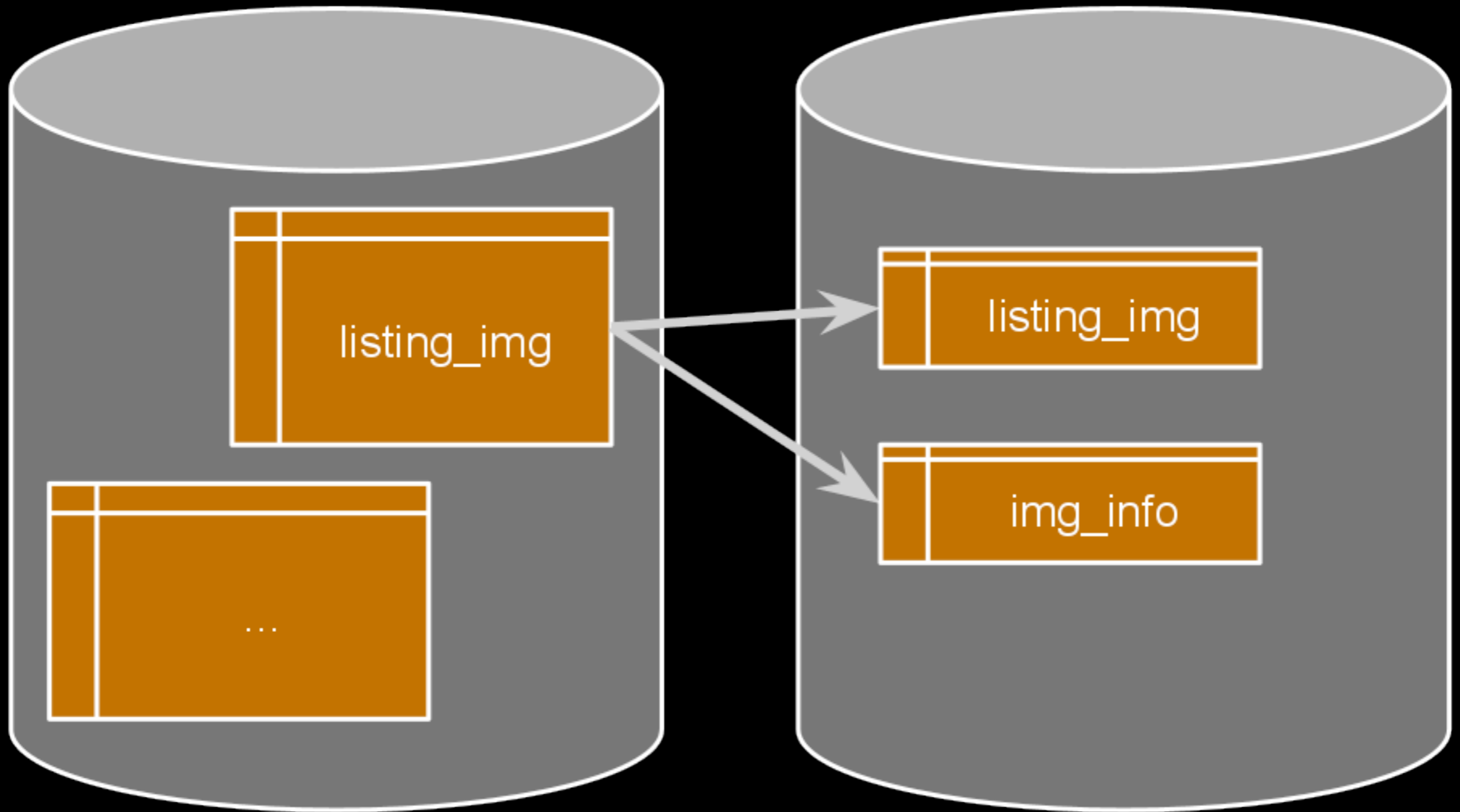
Create the Target



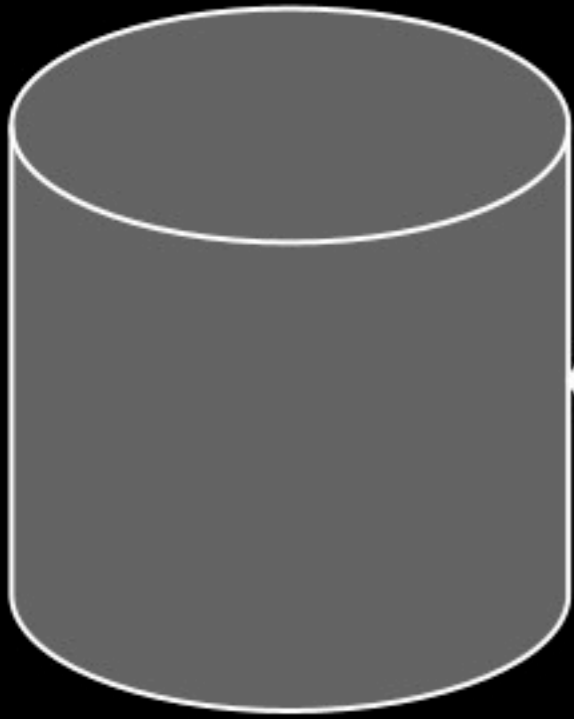
Merge into the Target



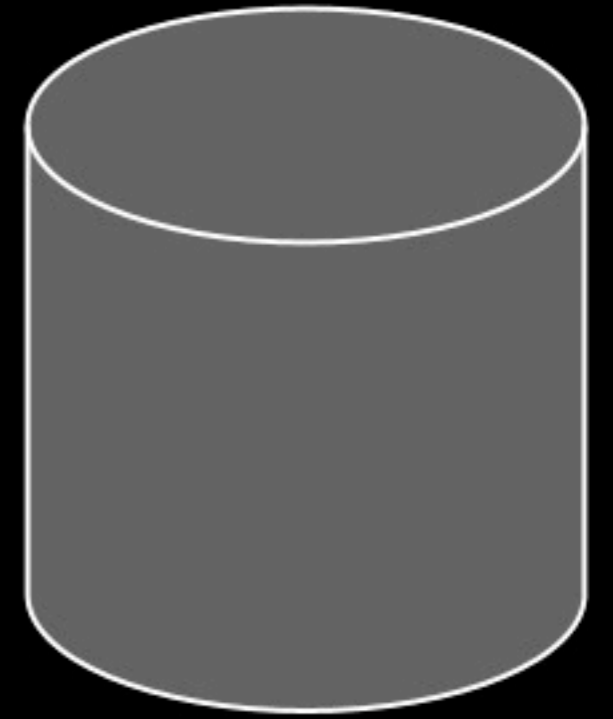
Separate the Target



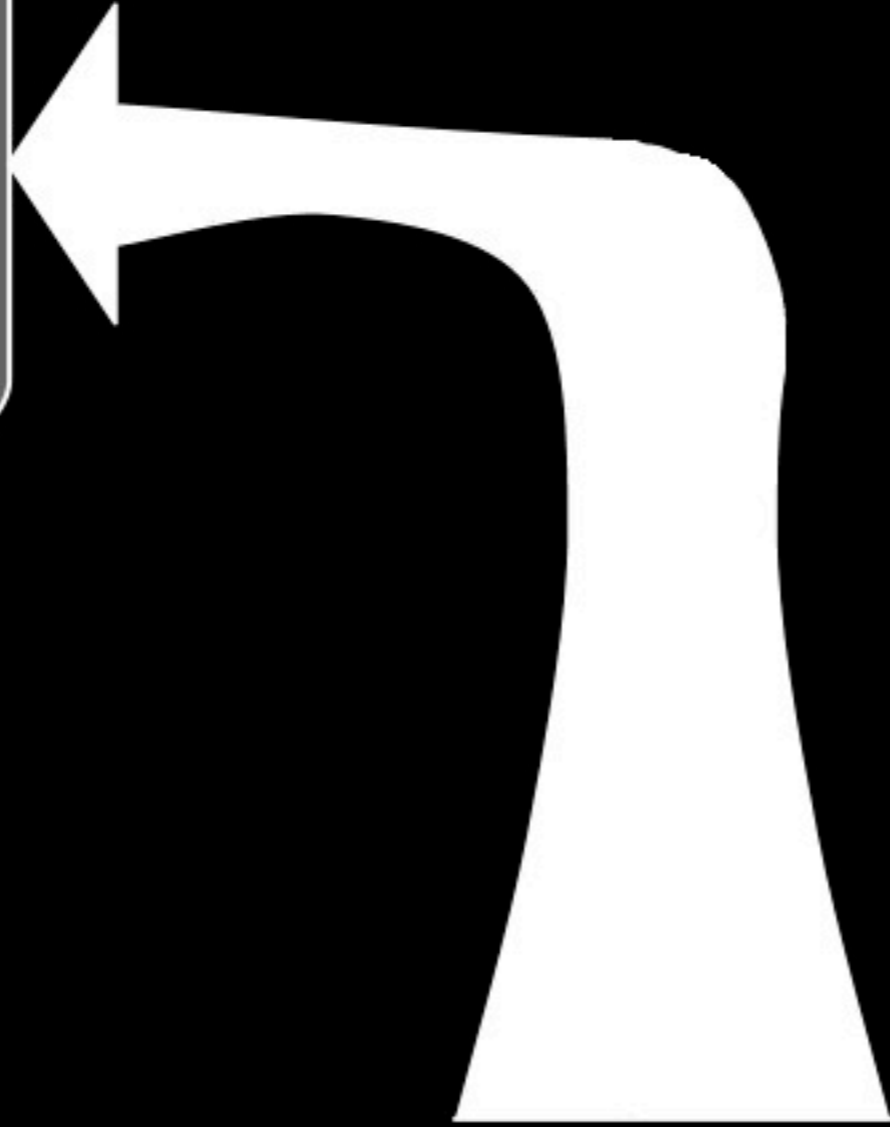
Source Only



Source

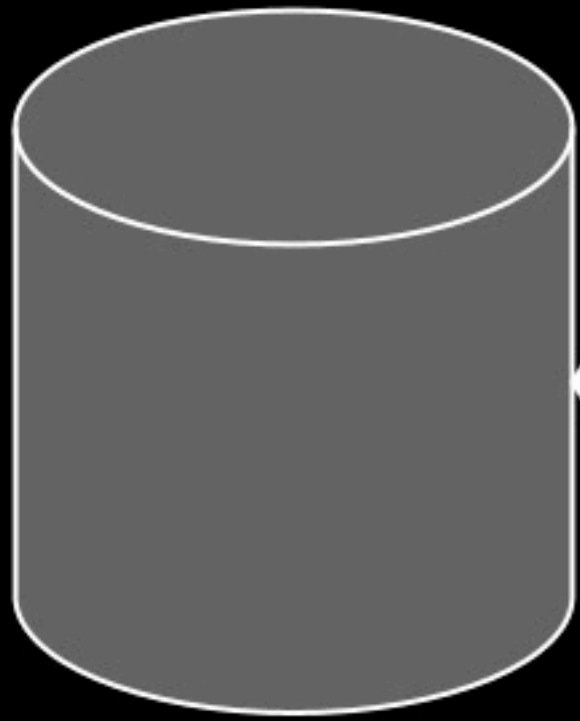


Target



Application
Writes

Teed Writes



Source

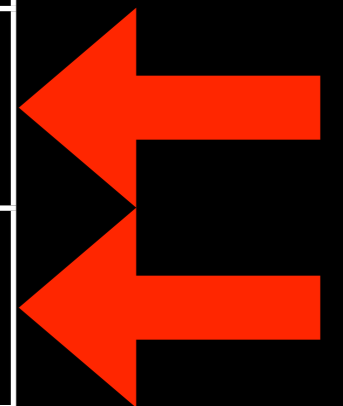
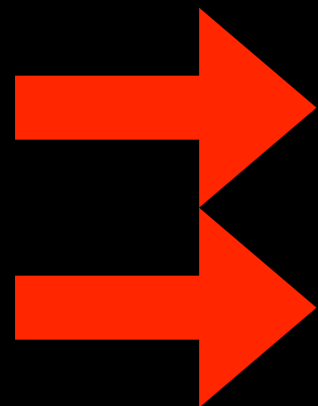


Target

Application
Writes

Generating IDs

Source		Target	
ID	Name	ID	Name
1	Adam	1	Adam
2	Bill	2	Bill
3	Charlie	3	David
4	David	4	Charlie



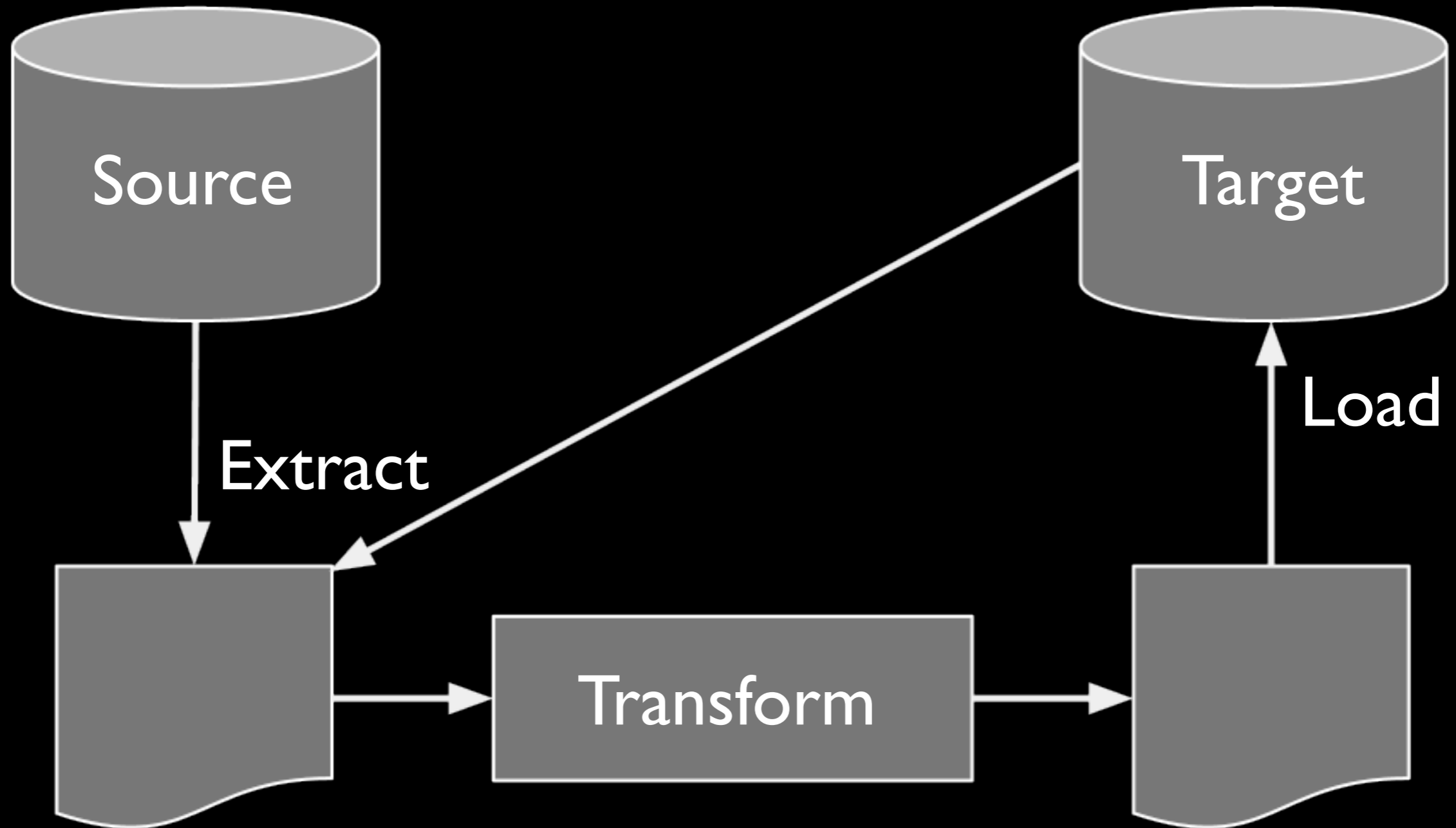
Ticket Server: <http://bit.ly/dbtickets>

Backfill

Application Code Backfill

```
function backfill_user(user_id) {  
    old_row = find_old_user(user_id);  
    new_row = create_new_user();  
    new_row.id = old_row.id;  
    ...  
    store_new_user(new_row);  
}
```

ETL Backfill



Don't Overload

- Script bulk inserts 100-2000 row batches

```
INSERT INTO account (id, firstname, lastname)
VALUES
(1, 'Alan', 'Alda'),
(2, 'Barry', 'Bonds'),
(3, 'Charlie', 'Chaplin');
```

Backfill Speed

Application Code

- Easier to Write

ETL

- Faster Run Time

Backfill Extract

Application Code

- Easier to Write
- Less Likely to Get Out Of Sync

ETL

- Faster Run Time
- Needs an Extract Source

Backfill Reruns

Application Code

- Easier to Write
- Less Likely to Get Out Of Sync
- Handles Duplicates from Multiple Executions

ETL

- Faster Run Time
- Needs an Extract Source
- REPLACE and INSERT ON DUPLICATE KEY UPDATE

Teed Writes then Backfill
or
Backfill then Teed Writes

Verification by Diff

```
$ diff_user 111222333  
target user row is missing
```

```
$ diff_user 123456789  
- source.user.address_state = 'CA'  
+ target.user.address_state = 'NY'
```

Verification by Diff

COMPARING 200 ROWS From: 111222197

User Id: 111222333

target user row is missing

User Id: 111222345

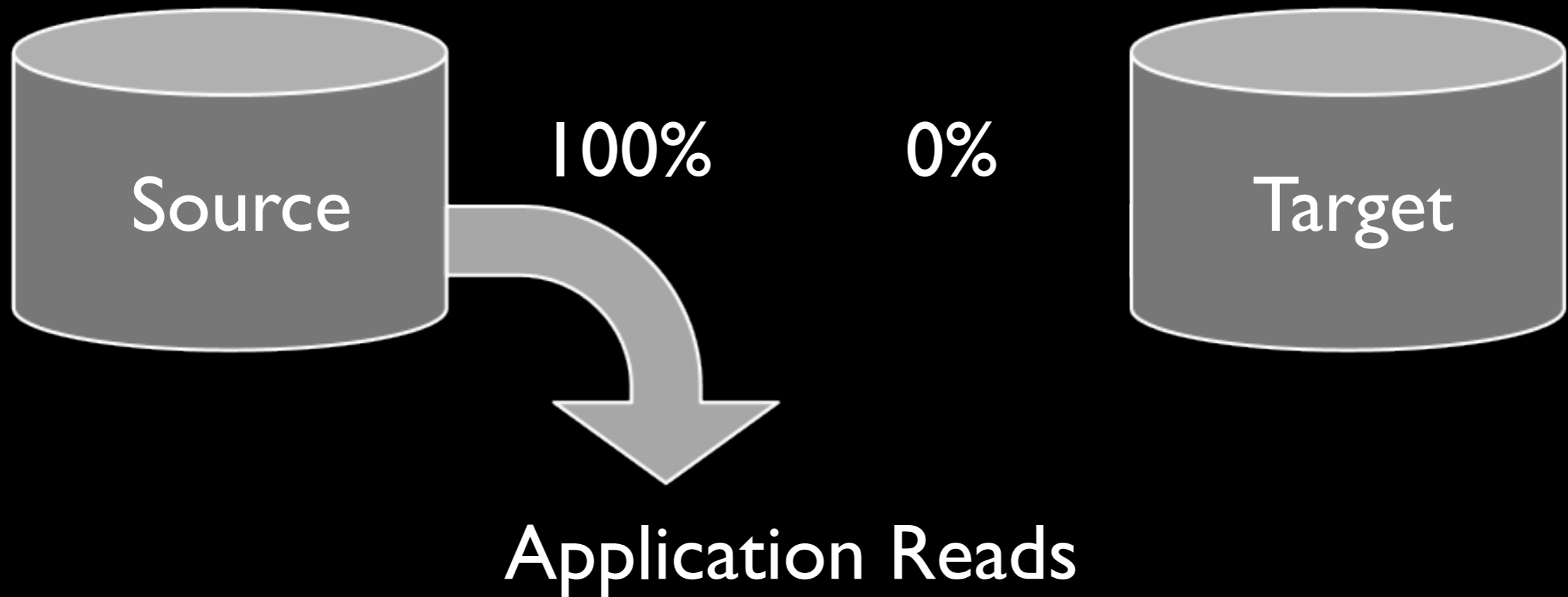
- source.user.address_state = 'CA'

+ target.user.address_state = 'NY'

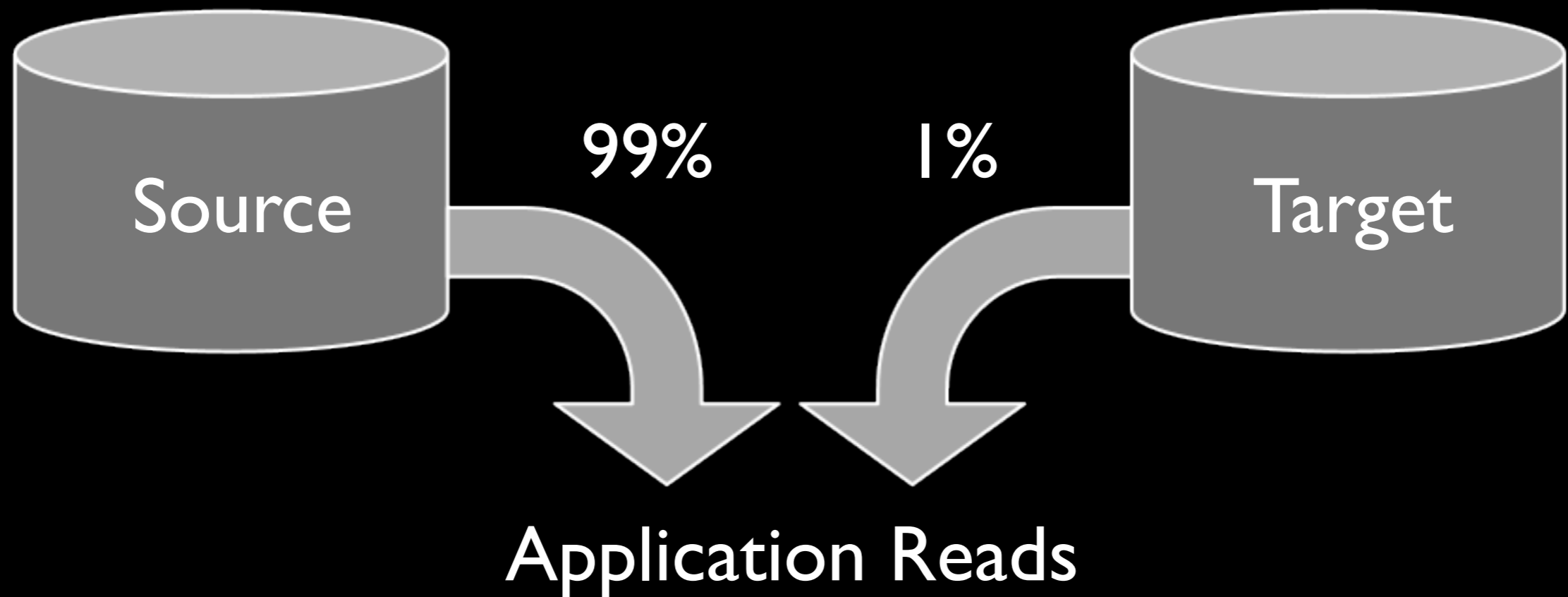
SUMMARY: total rows with errors: 2/200

Read from Target

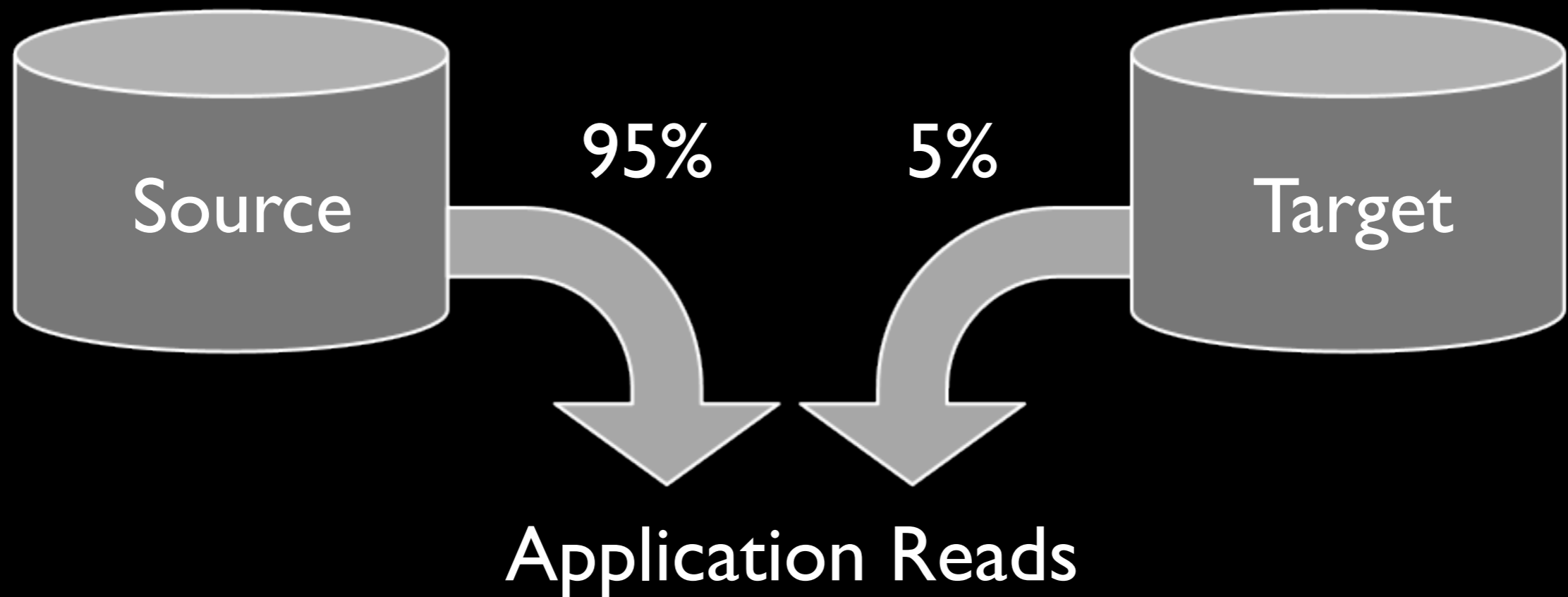
Progressive Ramp Up



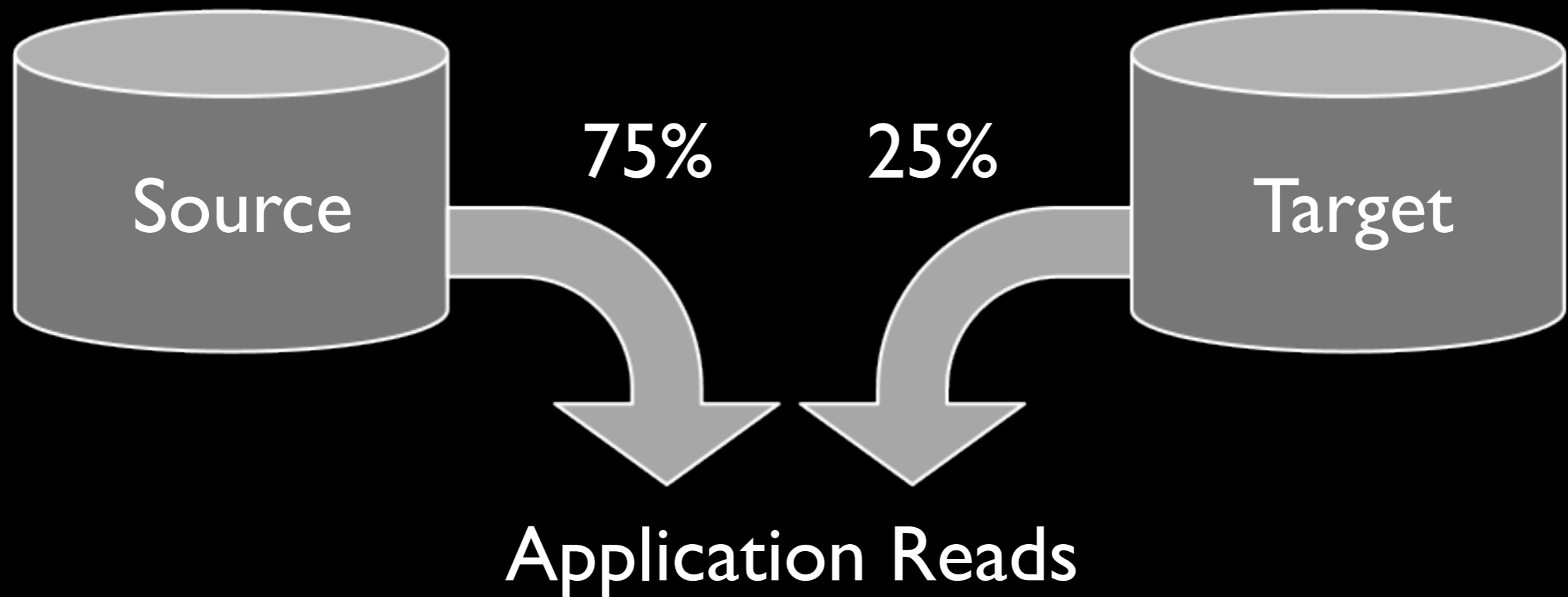
Progressive Ramp Up



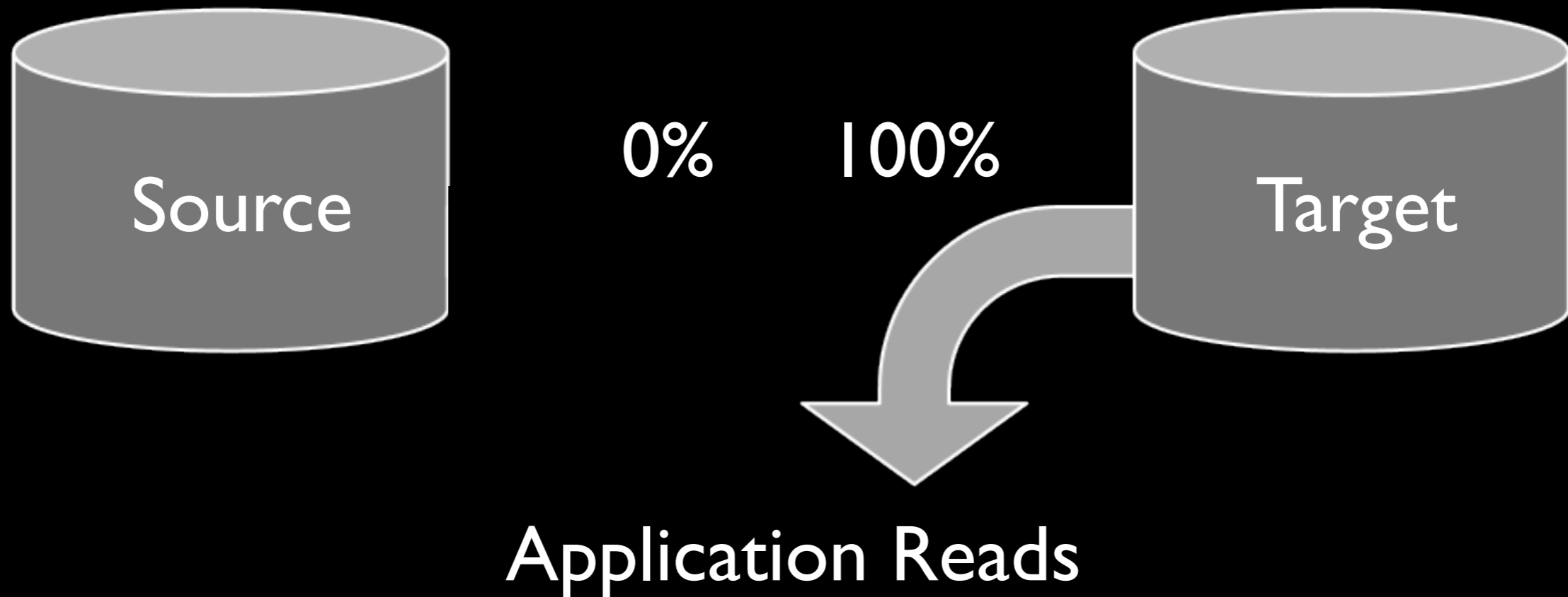
Progressive Ramp Up



Progressive Ramp Up



Progressive Ramp Up



Ramp Up Example

```
# global configuration
use_new_tables:
  employees: true
  percent: 1

# application code
if (enabled('use_new_tables')) {
  $result = read_from_target();
} else {
  $result = read_from_source();
}
```

Enabled or Disabled?

- Check cookies if user already assigned
- $\$enabled = \text{configured threshold} > \text{random \%}$
- Store $\$enabled$ in a cookie for future requests

Continuous Deployment

- Ramp Up / Ramp Down
- Backfill Fixes
- Need Code Running on Prod to Proceed
- Makes it Easier

No Foreign Keys

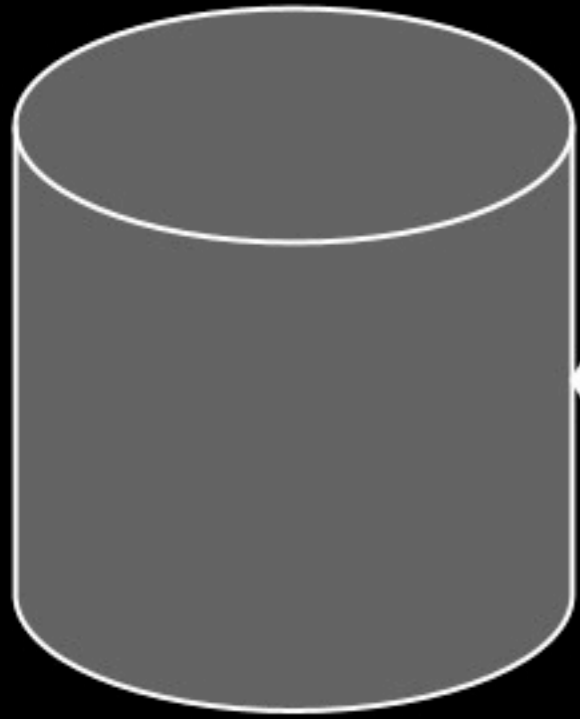


Wrapping Up

Analytical Data



Teed Writes



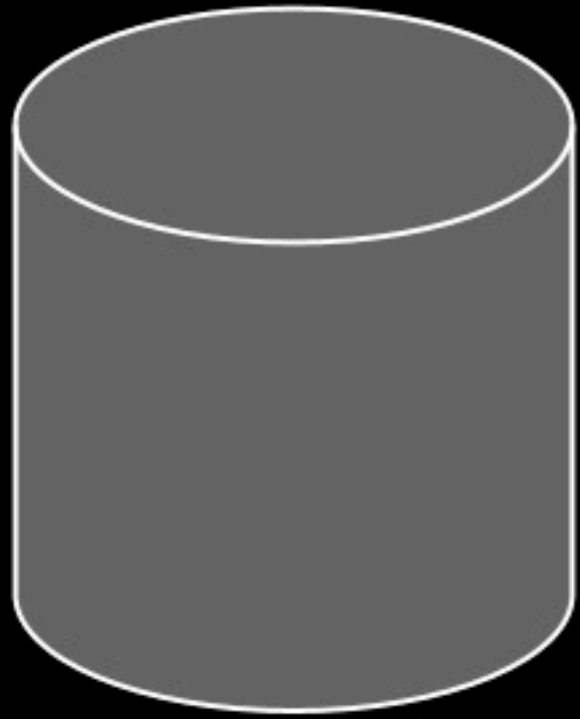
Source



Target

Application
Writes

Target Only



Source



Target



Application
Writes

Things to Remove

- Code to use old tables
- Code to switch on configuration
- Configuration
- Drop the old tables... eventually

5 Steps Per Table

1. Create Target Tables
2. Tee Writes
3. Backfill
4. Read from Target Tables
5. Wrap Up

Foundational Principles

- Migrate One Table at a Time
- Progressive Ramp Up
- Data Duplication During Transition

Questions?

- Yes, we're hiring! Specifically MySQL Ops
<http://bit.ly/etsywantsawesome>
- <http://codeascraft.etsy.com>
- <http://twitter.com/lapsu>

The Etsy logo is displayed on a solid orange rectangular background. The word "Etsy" is written in a white, serif font, centered within the rectangle.

Etsy

Matthew Graham
Percona Live NYC 2011