

While you are waiting, get started on Prerequisites
<https://vitess.io/docs/get-started/scaleway/>

Vitess

Paris Workshop

Sugu Sougoumarane

Co-creator, CTO @ PlanetScale

Morgan Tocker

Community Development Manager, Vitess Maintainer

@PlanetScale

@VITESSIO

House Keeping

Class Format is **Lecture and Exercises**

Please reach out if you get stuck

Each exercise follows on from the previous

Please help a neighbor if you finish early vs. jump ahead :-)

Restrooms

Breaks

Combined with Lab Exercises

When to ask questions

Whenever you can!

We will try and time manage, and may take longer questions offline.

Course Outline

3PM House Keeping and Introduction
3:30PM Deploy a Simple Vitess Cluster in Kubernetes (Exercise)
4PM Continued Lecture
4:30PM Vertical Split (Exercise)
4:45PM Continued Lecture
5:15PM Horizontal Shard (Exercise)

Continue last exercise until the end of the day.

What is **Vitess**?

**Sharded
MySQL**

**Massively
Scalable**

HA

Cloud-Native

Vitess Stats

Started
2010


100+
Contributors


8,000+
Stars

16,000+
 Commits

1000+
Slack Members

1000+
Forks 

Key Adopters



40% Migrated
to Vitess



Cash App fully
runs on Vitess

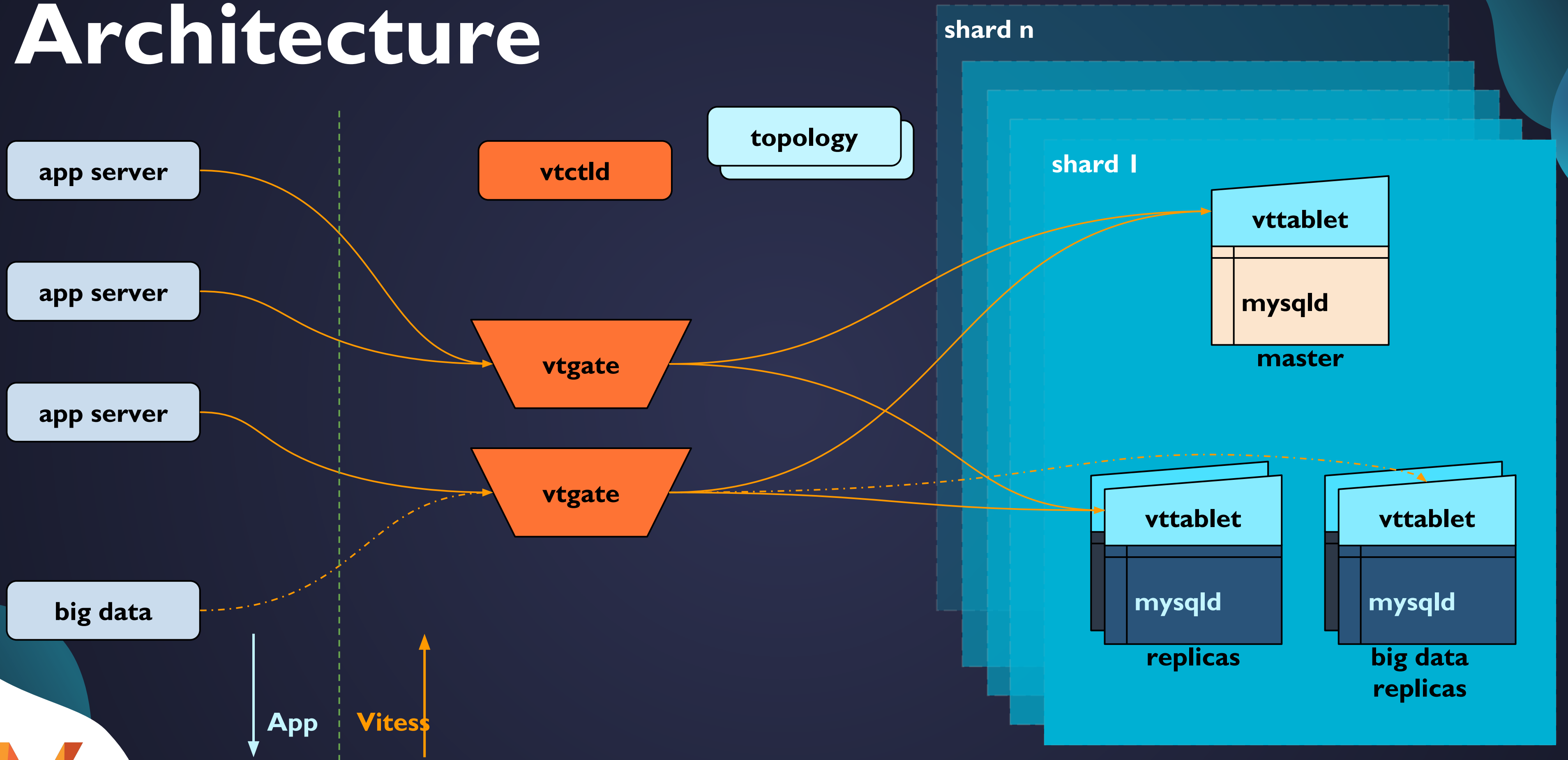


Powers all
advertising
campaign
management

Key Adopters



Architecture



Terminology

- **VTGate:** The Proxy that your applications connect to.
- **Topo Server:** The etcd server containing meta data.
- **Tablet:** A combination of a vttablet and a MySQL server.
- **Keyspace:** a logical “MySQL database” (aka schema). Keyspaces can be sharded or unsharded.
- **Cell:** A data center

For MySQL Users...

What version of MySQL is it?

VTGate currently advertises itself as MySQL 5.5.

What versions of MySQL does it support?

MySQL 5.6+ and MariaDB 10.0+

Requires RBR + GTIDs enabled

Strongly recommends semi-sync

What Queries are Supported?

Depends on if keyspace is sharded

Major Limitations:

SET [SESSION] var = x;

GROUP BY key ORDER BY different_key;

Consistency Model

- READ-COMMITTED for reads
- Atomic within a shard on update
- 2PC also available but not recommended
- Has elements of opt-in eventual consistency (VReplication provides materialized views).

(Translucent) Sharding

- Still designing a VSchema
- Recommend modeling updates to be single shard
- vs. transparent:
 - Distributes data everywhere
 - Possible latency penalty; typically a new engine

The “Best Use Case”

Signs you might be a fit include:

- You currently have schema-per-tenant
- You have a multi-tenant Application
- You need an upgrade path from MySQL to sharded MySQL
- Your monolithic databases are blocking your Kubernetes adoption

To Be Aware Of

Still MySQL underneath

- Including the MySQL Optimizer (good for OLTP)
- In some cases may improve analytics (VReplication + parallel scatter gather). Not All.

Born as an Internal Company Project

- Ease of use not Day 1 Priority
- It is a Priority for PlanetScale

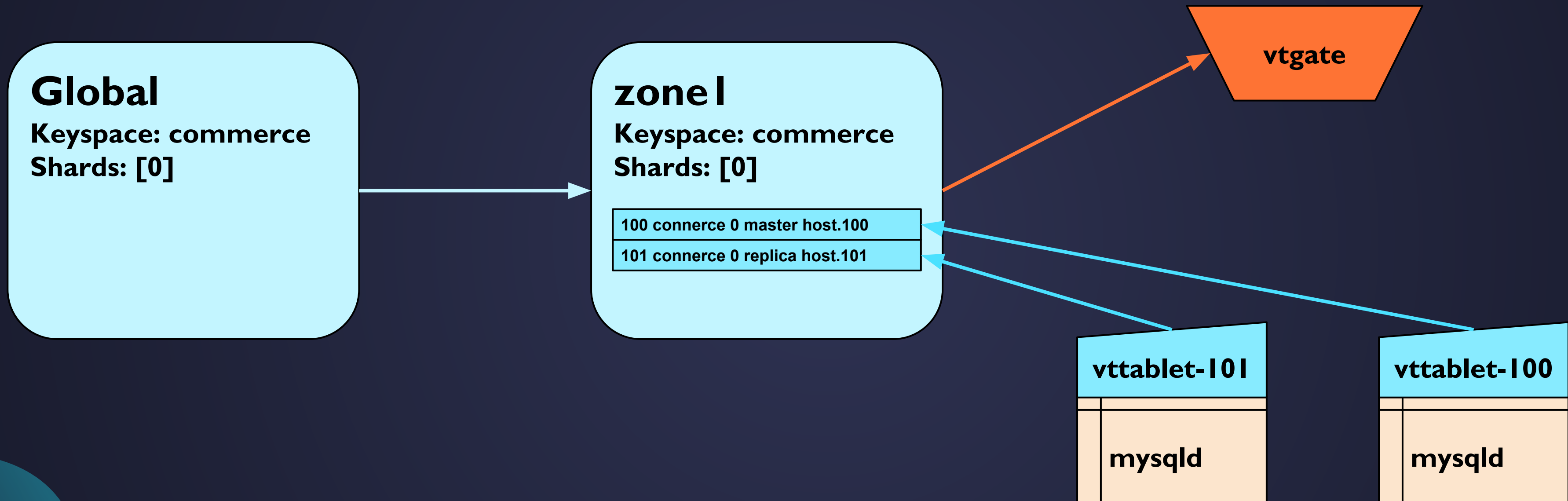
Exercise: Single Keyspace Cluster

Go to <https://vitess.io/docs/get-started/scaleway/>

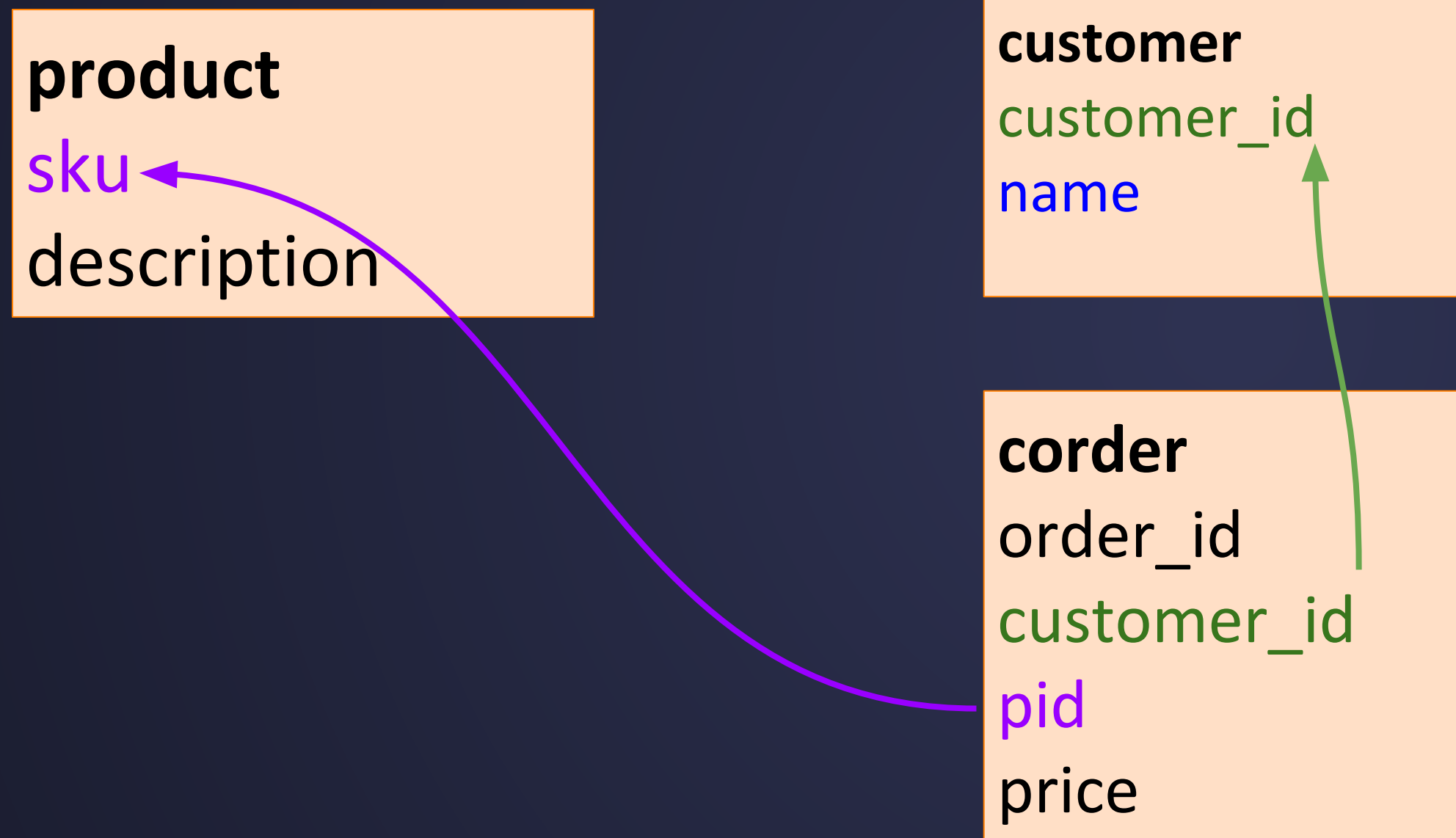
Stop **before** you reach “Topology”.

Class will continue at **4PM**

Topology



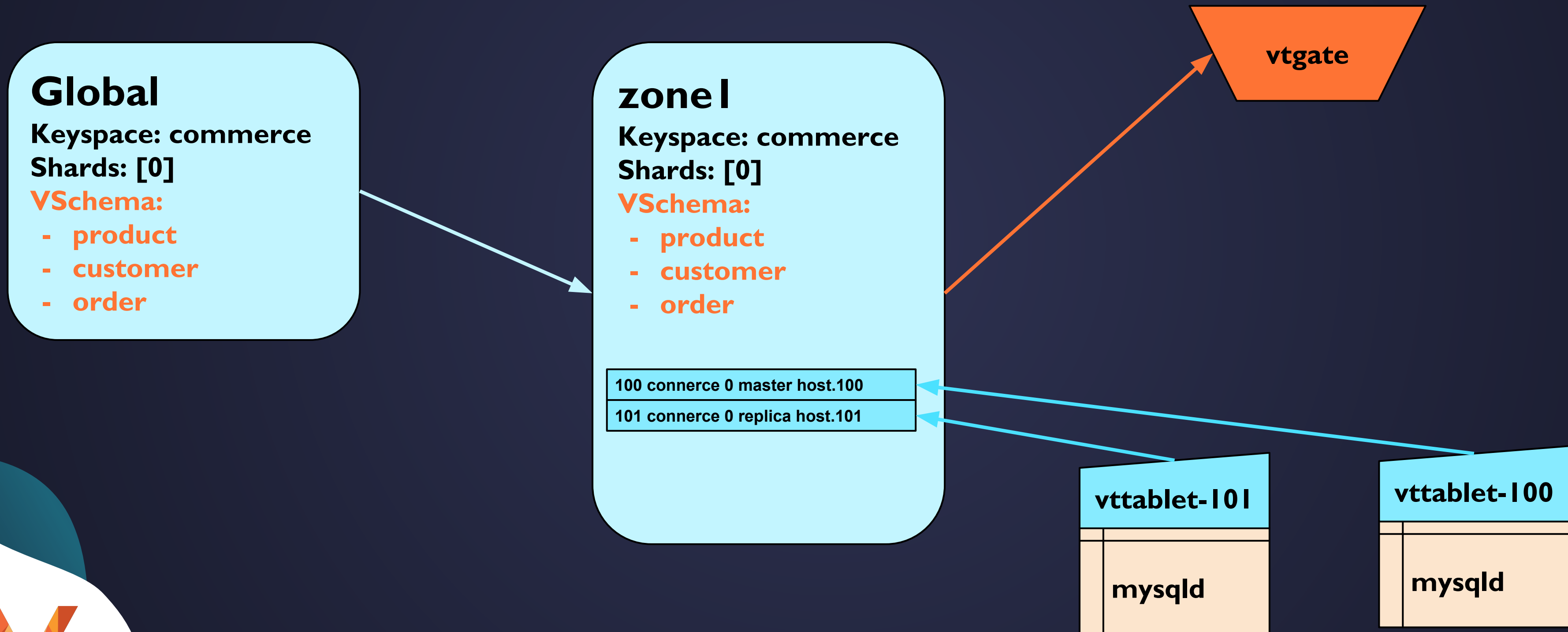
Schema



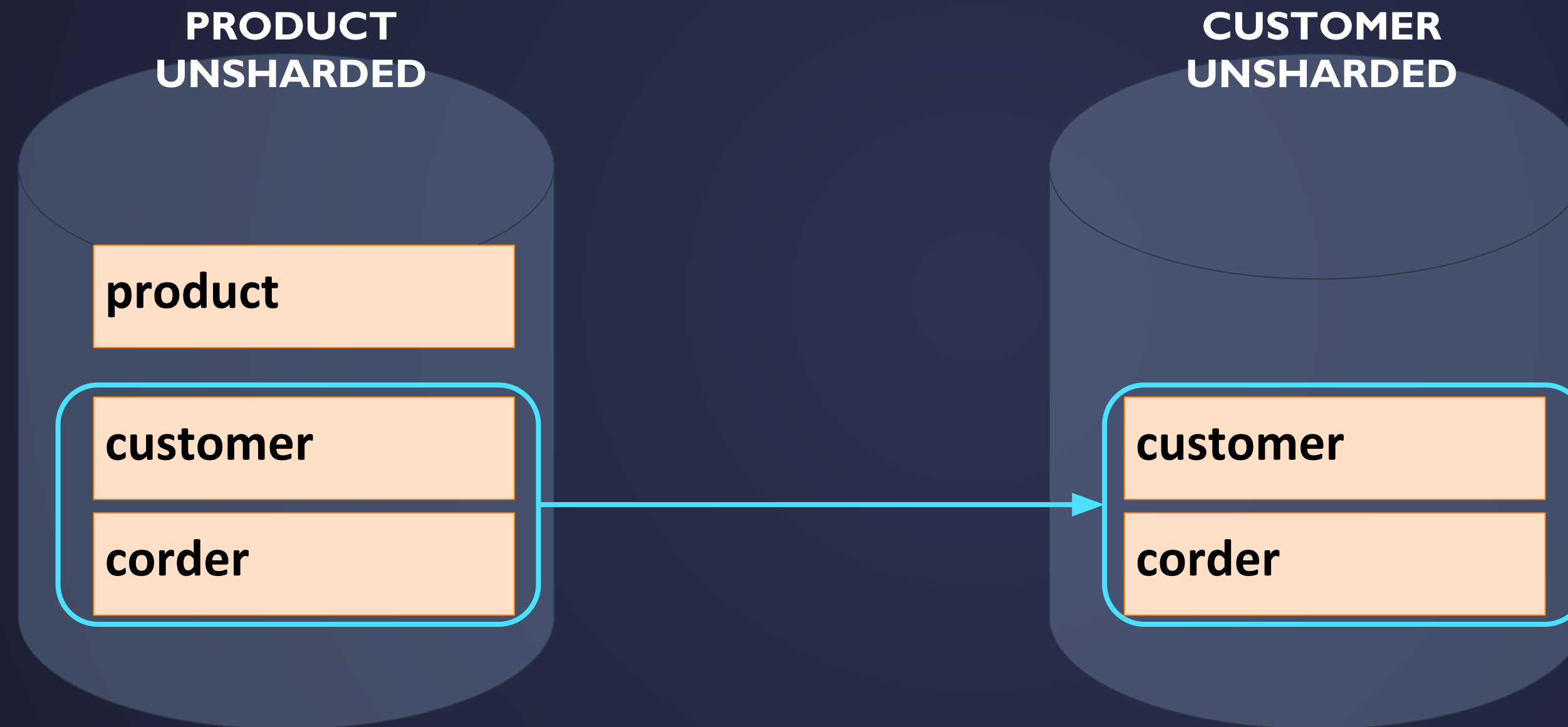
VSchema

**Everything needed
to route a query**

VSchema



Vertical Split



Exercise: Vertical Split

Go to <https://vitess.io/docs/get-started/scaleway/>

Stop before you reach “Horizontal Sharding”.

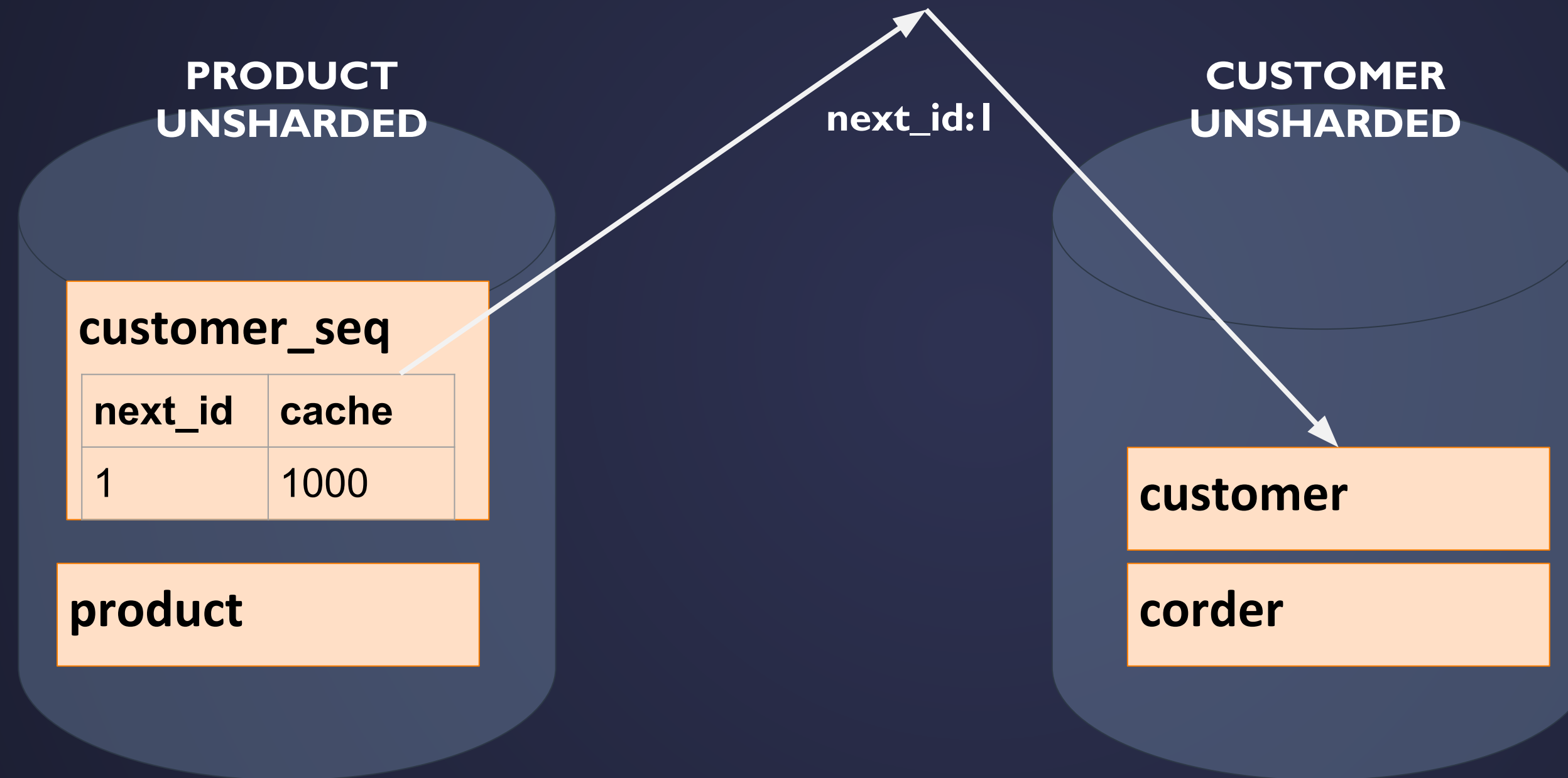
Class will continue at **4:45PM**

Sequences

Cross-shard autoinc

Sequences

`INSERT INTO cusotmer(customer_id,..) VALUES(NULL,...)`



Vindexes

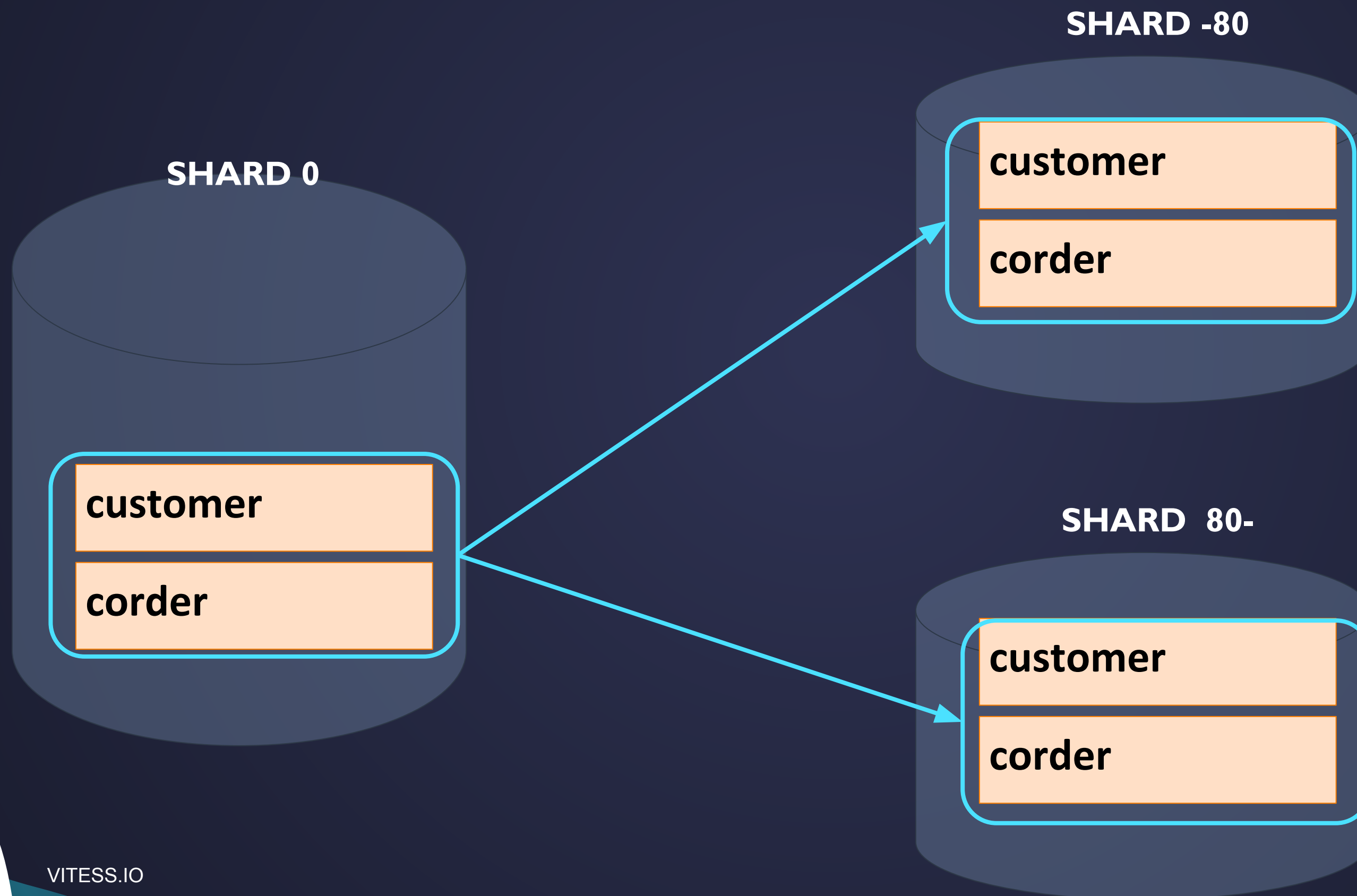
**How is a table
sharded?**

Vindexes

```
{  
  "tables": {  
    "customer_seq": {  
      "type": "sequence"  
    },  
    "order_seq": {  
      "type": "sequence"  
    },  
    "product": {}  
  }  
}
```

```
{  
  "sharded": true,  
  "vindexes": {  
    "hash": {  
      "type": "hash"  
    }  
  },  
  "tables": {  
    "customer": {  
      "column_vindexes": [  
        {  
          "column": "customer_id",  
          "name": "hash"  
        }  
      ],  
      "auto_increment": {  
        "column": "customer_id",  
        "sequence": "customer_seq"  
      }  
    },  
    "corder": {  
      "column_vindexes": [  
        {  
          "column": "customer_id",  
          "name": "hash"  
        }  
      ],  
      "auto_increment": {  
        "column": "order_id",  
        "sequence": "order_seq"  
      }  
    }  
  }  
}
```

New Shards



Shard naming key ranges



Exercise: Horizontal Sharding

Go to <https://vitess.io/docs/get-started/scaleway/>

Finish the remainder of the exercises.

Thank you for attending!