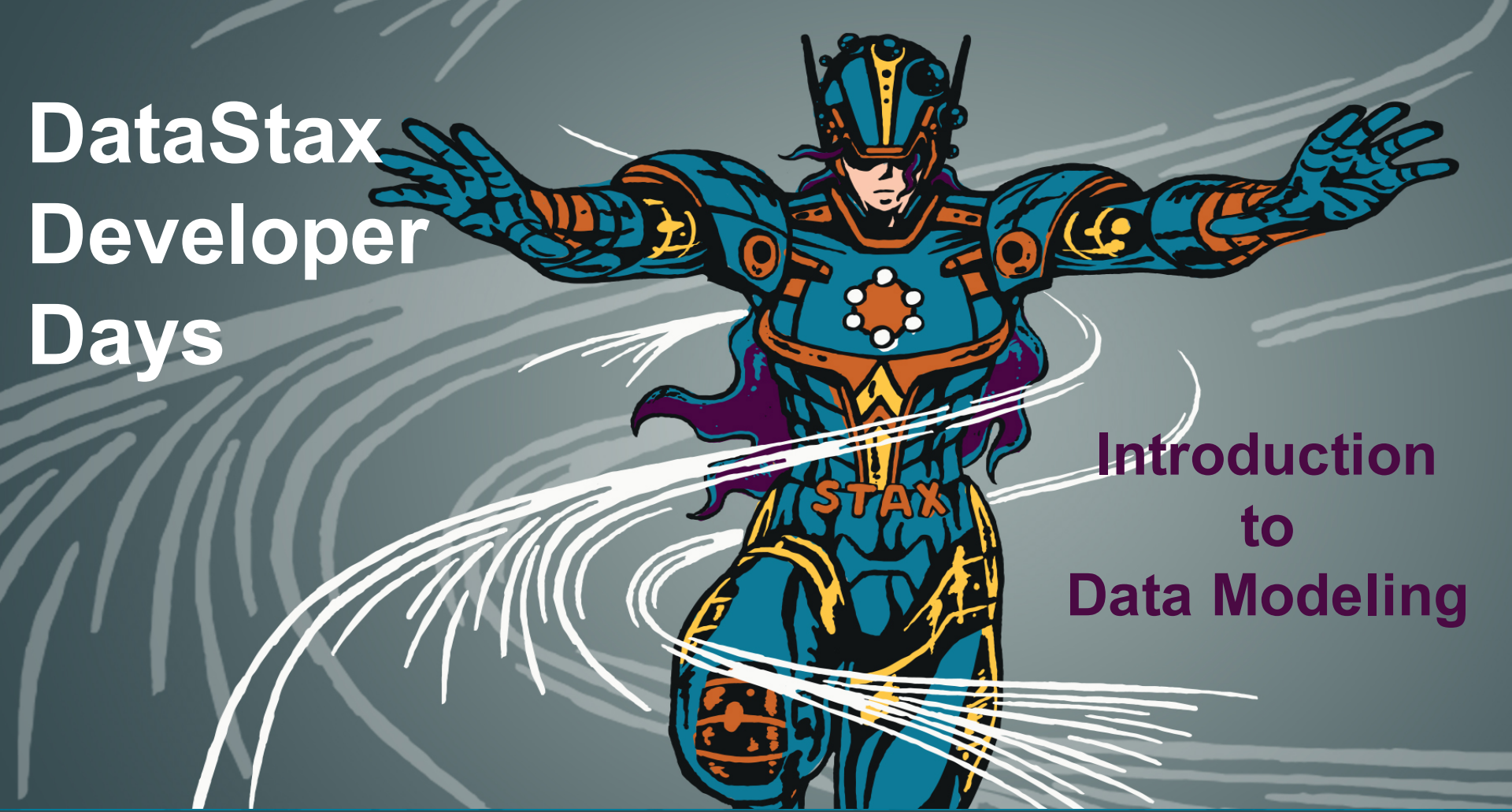
DataStax Developer Days



Introduction to Data Modeling

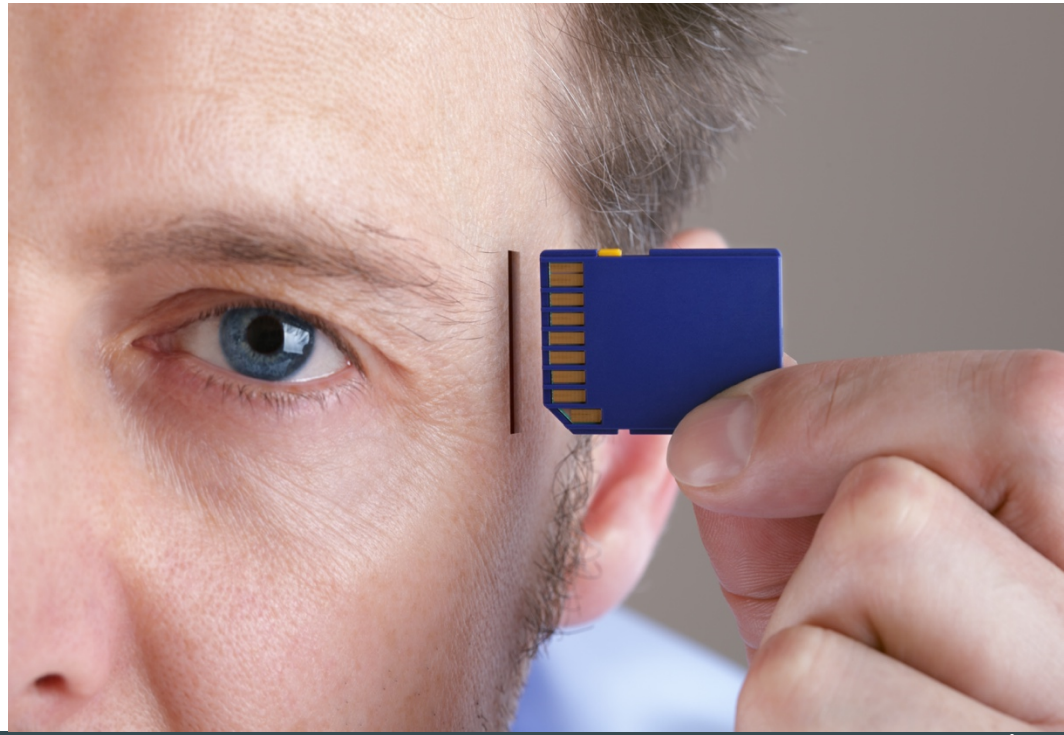
DATASTAX

DATASTAX ACADEMY

The Cassandra Mind-shift

Change the way you think about data modeling

- Two Main concepts for successful modeling:
  - Begin with the query in mind
  - Denormalization is good



Cassandra Data Modeling

Partition – the fundamental unit of access

- A Partition has
  - A partition key
  - Associated rows with
    - Clustering columns
    - Data columns

Last Name	First Name	Address	Email
Flintstone	Dino	3 Stone St	dino@gmail.com
Flintstone	Fred	3 Stone St	fred@gmail.com
Flintstone	Wilma	3 Stone St	wilm@gmail.com
Rubble	Barney	4 Rock Cir	brub@gmail.com
Rubble	Betty	4 Rock Cir	betr@gmail.com

Partition Key:
"Bedrock"

Hash Function

42

Cassandra Data Modeling

Tables hold many partitions

"Frost" Partition

Last Name	First Name	Address	Email
Moose	Bullwinkle	1 Tree St	moo@gmail.com
Squirrel	Rocky	3 Sky Blvd	fly@gmail.com
Jetson	Judy	2 StarSt	judy@gmail.com
Rubble	Barney	4 Rock Cir	brub@gmail.com
Rubble	Betty	4 Rock Cir	betr@gmail.com

Cartoon Characters by City Table

42

Last Name	First Name	Address	Email
----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----

17

----	----	----	----
----	----	----	----
----	----	----	----

83

----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----

92

----	----	----	----
----	----	----	----

Cassandra Data Modeling

Tables hold many partitions

"Bedrock" Partition

Last Name	First Name	Address	Email
Flintstone	Fred	3 Stone St	fred@gmail.com
Flintstone	Dino	3 Stone St	dino@gmail.com
Flintstone	Wilma	3 Stone St	wilm@gmail.com
Rubble	Barney	4 Rock Cir	brub@gmail.com
Rubble	Betty	4 Rock Cir	betr@gmail.com

Cassandra Table

42

Last Name	First Name	Address	Email
----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----
----	----	----	----

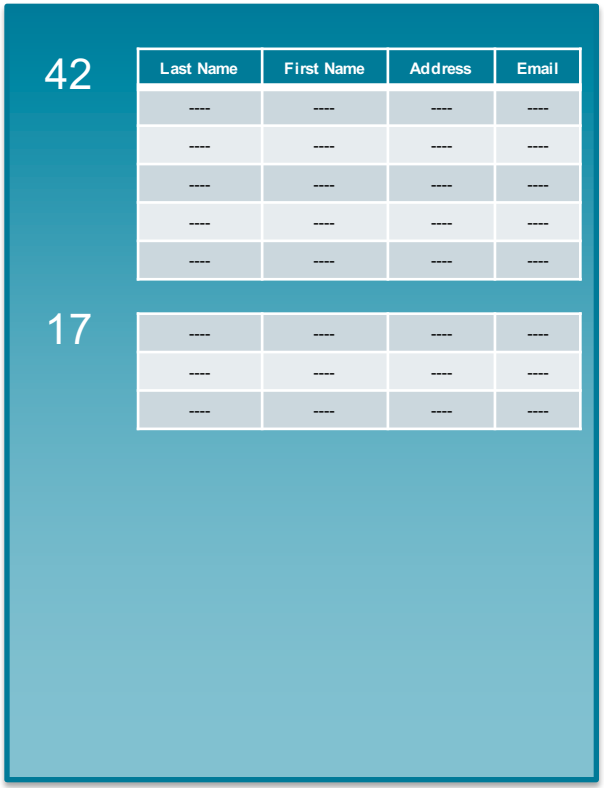
Cassandra Data Modeling

Tables hold many partitions

"Orbit City" Partition

Last Name	First Name	Address	Email
Jetson	George	2 Star St	gjet@gmail.com
Jetson	Jane	2 Star St	jane@gmail.com
Jetson	Judy	2 StarSt	judy@gmail.com

Cassandra Table



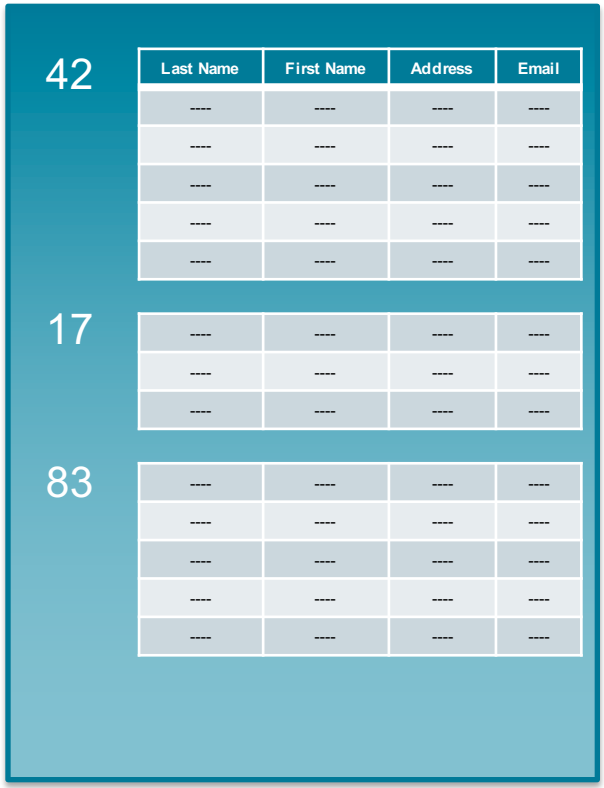
Cassandra Data Modeling

Tables hold many partitions

"Smurfville" Partition

Last Name	First Name	Address	Email
Azrael	Cat	5 Blue St	acat@gmail.com
Smurf	Brainy	7 Azure ln	brain@gmail.com
Smurf	Clumsy	4 Teal Cir	clu@gmail.com
Smurf	Greedy	9 Royal St	gime@gmail.com
Smurf	Papa	2 Navy Ct	papa@gmail.com

Cassandra Table



Cassandra Data Modeling

Tables hold many partitions

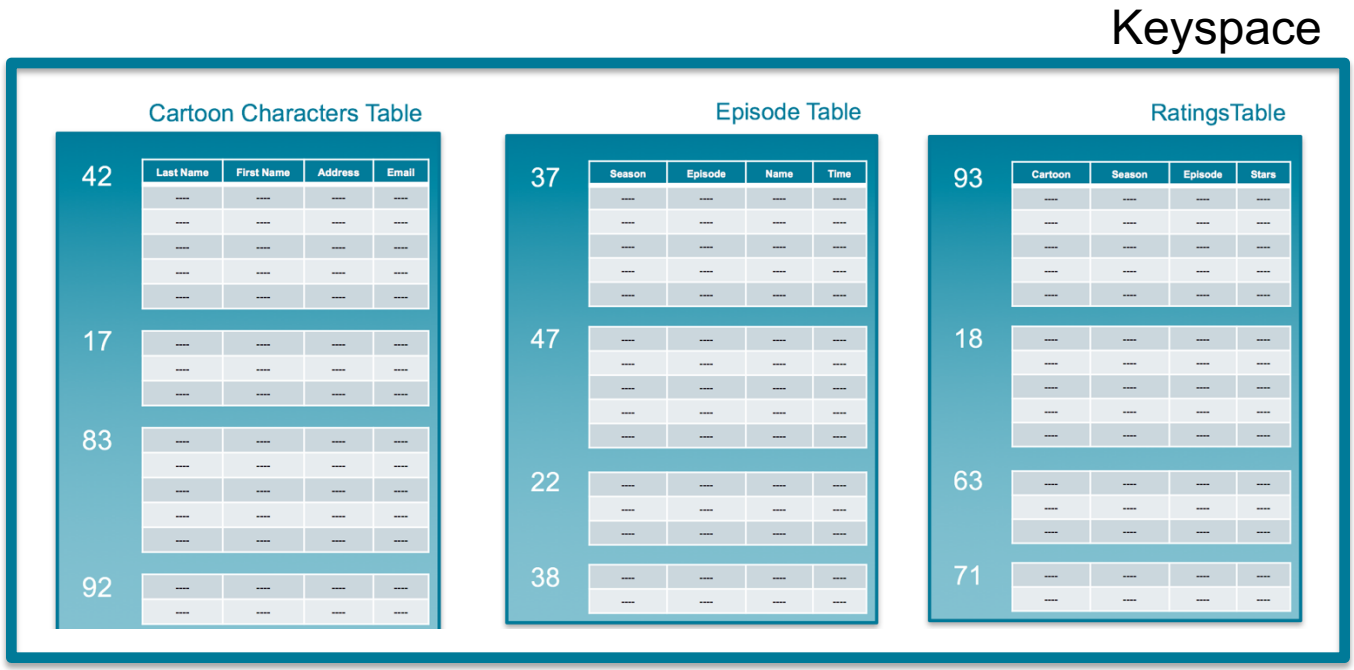
"Frostbite Falls" Partition

Last Name	First Name	Address	Email
Moose	Bullwinkle	1 Tree St	moo@gmail.com
Squirrel	Rocky	3 Sky Blvd	fly@gmail.com

Cassandra Table

Cassandra Data Modeling

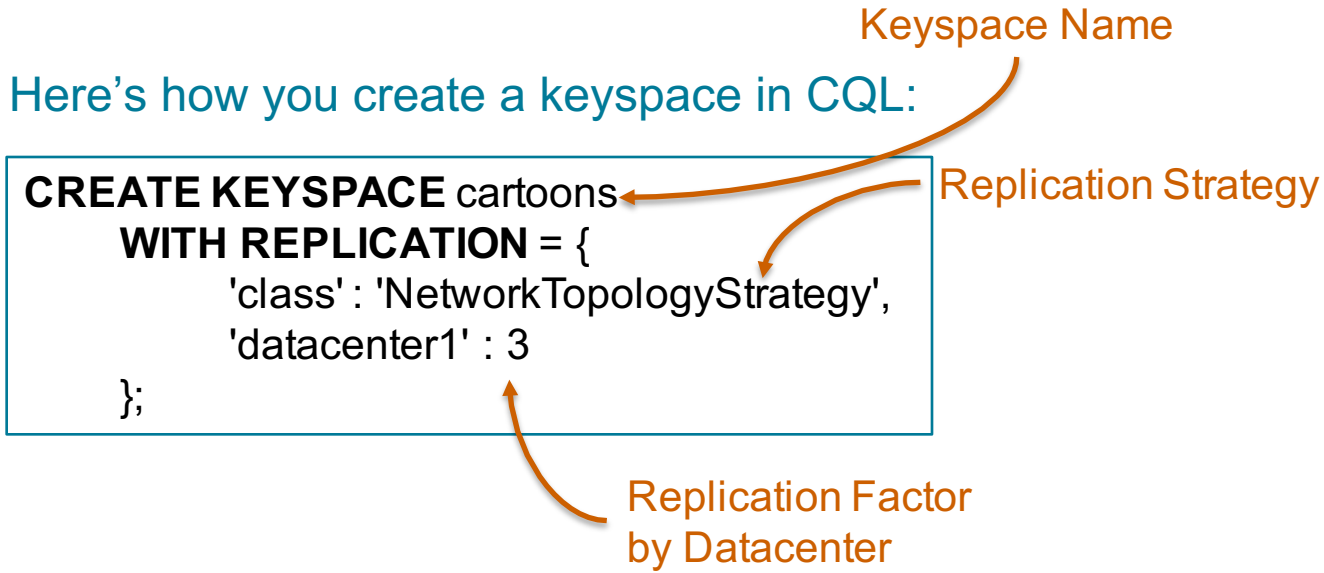
Keyspaces contain many tables



Cassandra Data Modeling

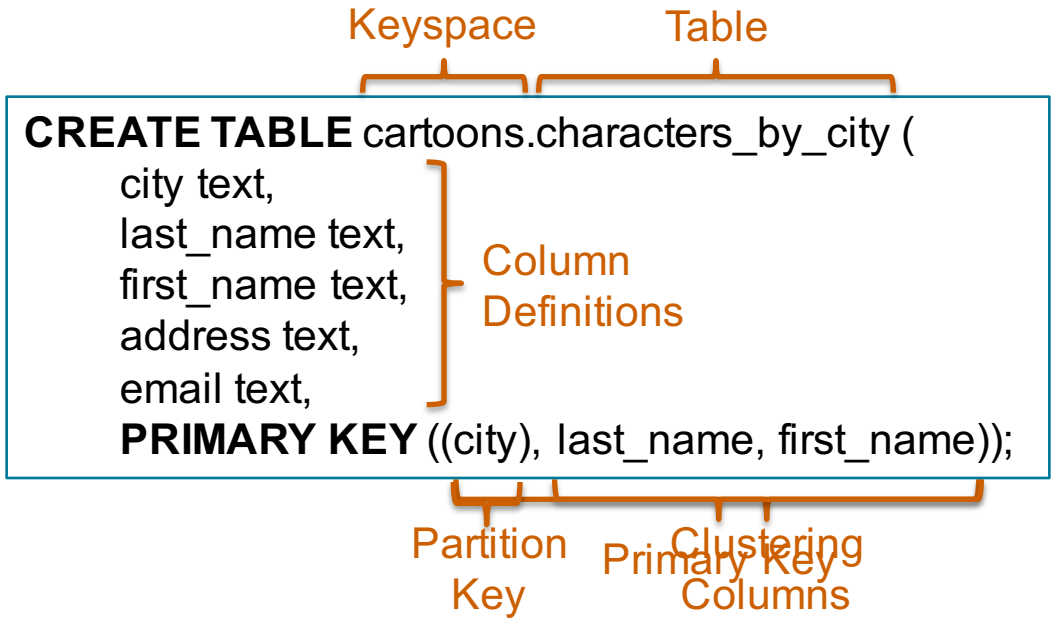
CQL – Cassandra's Query Language

Here's how you create a keyspace in CQL:



Cassandra Data Modeling

CQL – Create table example



Cassandra Data Modeling

CQLSH – Investigating our schema

To list your keyspaces:

```
DESCRIBE KEYSPACES;
```

To look at a specific keyspace definition:

```
DESCRIBE KEYSPACE cartoon;
```

To look at a table definition:

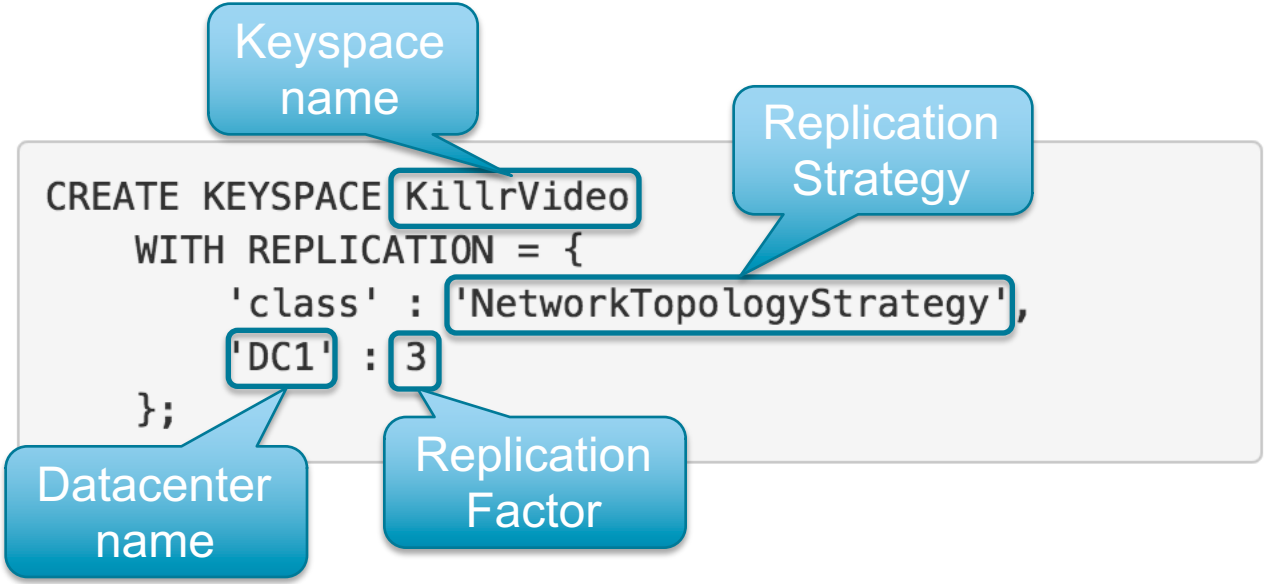
```
DESCRIBE TABLE cartoon.characters_by_city;
```

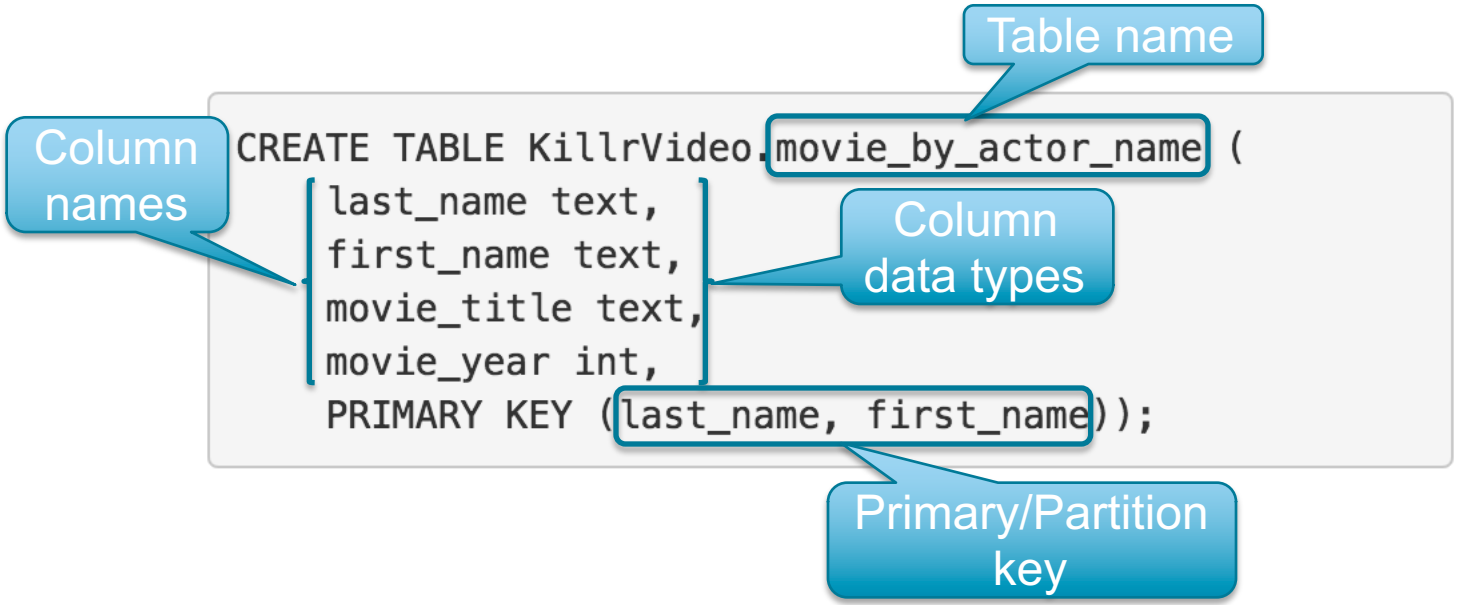

DATASTAX ACADEMY

Enough Talk Already!!!!

It's time to play!

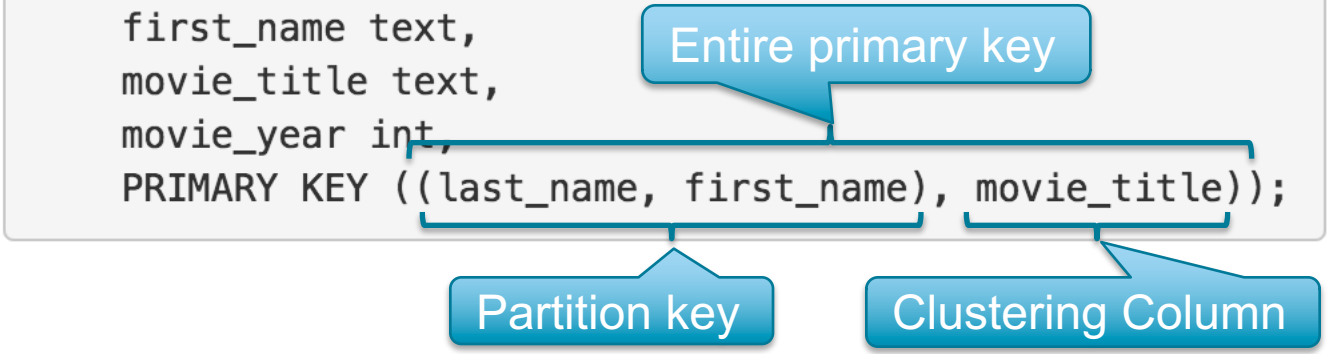
- You each have a cluster
- You will access your cluster using DSE Studio
- Open a browser and go to <Your node's IP Address>:9091
- Click on notebook "**Data Modeling: Data Modeling Intro**"
- Work your way through the steps of this Intro





```
INSERT INTO KillrVideo.movie_by_actor_name (
    last_name,
    first_name,
    movie_title,
    movie_year)
VALUES(
    'Stone',
    'Emma',
    'Battle of the Sexes',
    2017);
```


```
CREATE TABLE KillrVideo.movie_by_actor_name (
    last_name text,
    first_name text,
    movie_title text,
    movie_year int,
    PRIMARY KEY ((last_name, first_name), movie_title));
```



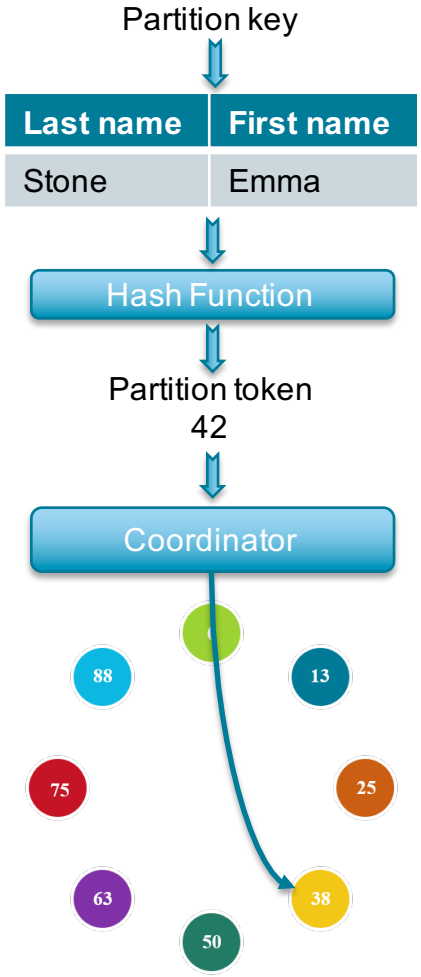
Last name	First name	Movie title	Movie year
Depp	Johnny	Lone Ranger	2013
Depp	Johnny	Tourist	2010
Jolie	Angelina	Tourist	2010
Jolie	Angelina	By the Sea	2015
Pitt	Brad	Big Short	2015
Pitt	Brad	Moneyball	2011

Input

Last name	First name
Stone	Emma

Output

Movie title	Movie year
La la land	2016



Data Modeling Intro

Quick review

- Key concepts:
  - Keyspace – contains tables
  - Table – contains partitions
  - Partition – basic unit of storage/retrieval
    - Identified by partition key embedded within primary key
    - Contains one or more rows
  - Row – has a primary key and data columns
  - Primary key – intra-table row identifier
    - Consists of partition key and clustering columns
  - Partition key – partition identifier, hashes to partition token
  - Clustering column – intra-partition key for sorting rows within partition

DATASTAX ACADEMY

Let’s do a Data Modeling Project Together!

DATASTAX

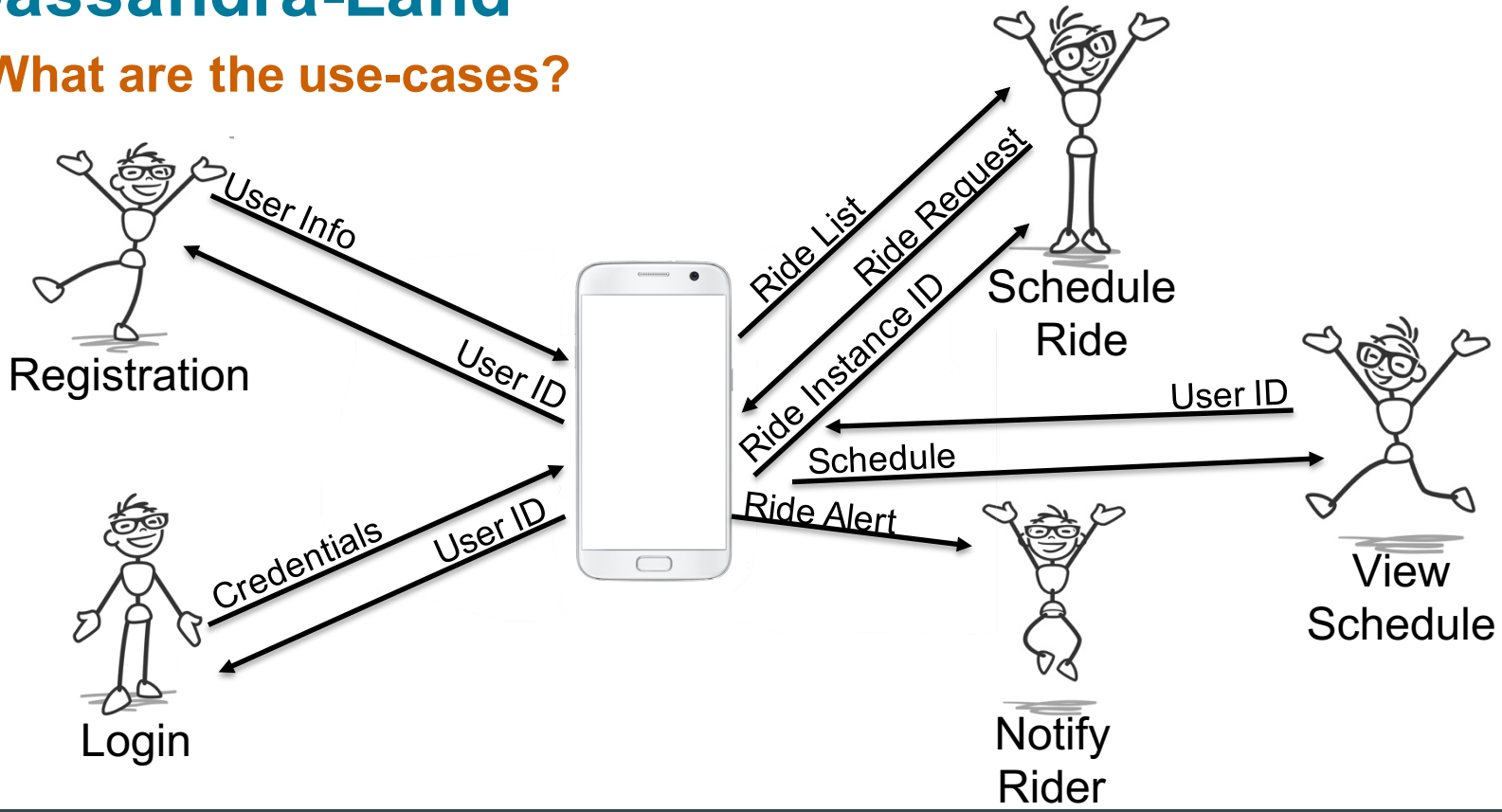
Welcome to Cassandra-Land

The Theme Park Where You Can Find…

- Distributed & Fault-Tolerant Rides
- Amazing Throughput
- And Fast Response Times

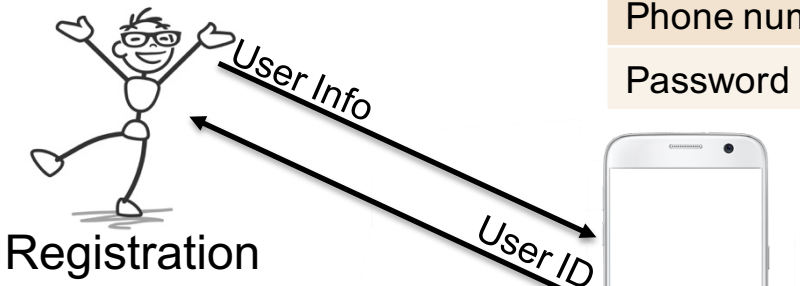
Cassandra-Land

What are the use-cases?



Cassandra-Land

Registration Use-Case



User Info

Field	Type
Phone number	text
Password	text

User ID

Field	Type
User ID	UUID

users_by_phone_number

Field	Type
phone_number	text
user_id	UUID
password	text

Registration Use-Case

What you need to know – CREATE KEYSPACE

```
CREATE KEYSPACE <keyspace name> WITH REPLICATION = {
       'class' : <replication strategy>,
      <datacenter name> : <replication factor>,
      // Specify addition datacenters/replication factors here...
};
```

For example:

```
CREATE KEYSPACE killrvideo WITH REPLICATION = {
       'class' : NetworkTopologyStrategy,
      'USWestDC': 3,
      'USEastDC': 2
};
```

Registration Use-Case

What you need to know – DESCRIBE KEYSPACE

DESCRIBE KEYSPACE <keyspace name>;

Registration Use-Case

What you need to know – CREATE TABLE

```
CREATE TABLE <keyspace name>.<table name> (
        <field name> <field type>,
        // Add additional field descriptions here
        PRIMARY KEY(<primary key descriptor>)
);
```

Registration Use-Case

What you need to know – INSERT

```
INSERT INTO <keyspace name>.<table name>
      (<column list>)
      VALUES(<column values>);
```

Registration Use-Case

What you need to know – SELECT *

SELECT * FROM <keyspace name>.<table name>;

DATASTAX ACADEMY

Registration Use-Case

What you need to know – Upserts

- Cassandra does **NOT** read before writing
- Inserting a row with the same primary key causes an update
- This behavior is called an “upsert”
- Similarly, updates to non-existent rows cause an insert
- INSERT INTO keyspace.table IF NOT EXISTS …
  - Lightweight transaction to prevent upserts

DATASTAX ACADEMY

Notebook Data Modeling: Cassandra-Land Project

Do Steps 1A and 1B

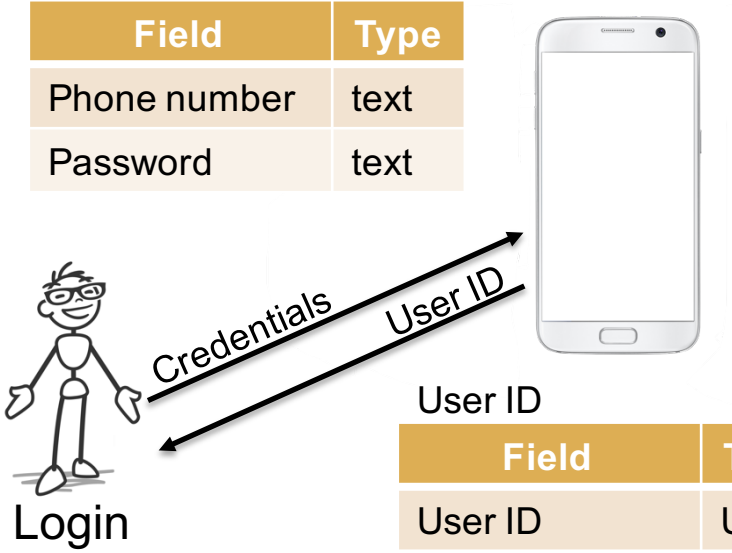
DATASTAX

Cassandra-Land

Login Use-Case

Credentials

Field	Type
Phone number	text
Password	text



User ID

Field	Type
User ID	UUID

users_by_phone_number

Field	Type
phone_number	text
user_id	UUID
password	text

Registration Use-Case

What you need to know – SELECT

```
SELECT * FROM <keyspace name>.<table name>
    WHERE <query constraints>;
```

- Must include full partition key
- Partition keys do NOT support inequalities
- Not all clustering columns need be specified, but…
- Any preceding clustering columns *MUST* be specified

DATASTAX ACADEMY

Notebook Data Modeling: Cassandra-Land Project

Do Step 2

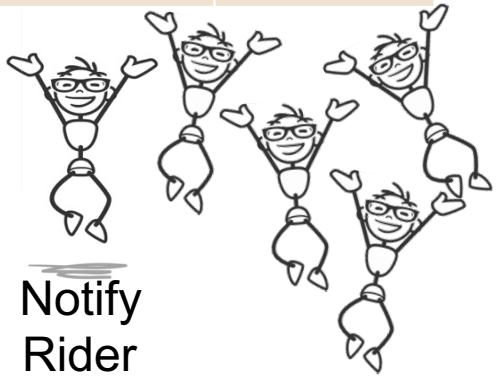
Cassandra-Land

Ride Alert Use-Case

ride_instances_by_start_time

Field	Type
start_time	timestamp
ride_id	UUID↑
ride_name	text
user_id	UUID↑
phone_number	text

Ride Alert



Ride Alert

Field	Type
phone_number	text
ride_name	text
start_time	timestamp

Cassandra-Land

View Schedule Use-Case

ride_instances_by_user_id

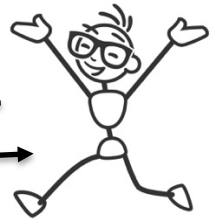
Field	Type
user_id	UUID
start_time	timestamp↑
ride_id	UUID
ride_name	text

User ID

Field	Type
User ID	UUID

User ID

Schedule



View Schedule

Schedule

Field	Type
start_time	timestamp
ride_name	text

Cassandra-Land

Schedule Ride Use-Case

ride_list_by_location

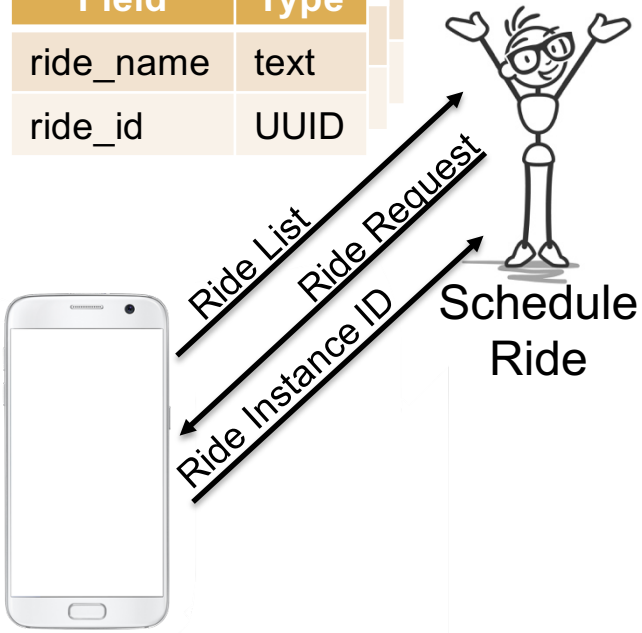
Field	Type
location	text
ride_id	UUID↑
ride_name	text
capactiy	int

rider_count_by_time_and_ride

Field	Type
start_time	timestamp
ride_id	UUID
rider_count	int

Ride List

Field	Type
ride_name	text
ride_id	UUID



Ride Request

Field	Type
user_id	UUID
ride_id	UUID
start_time	timestamp

Ride Instance ID

Field	Type
ride_instance_id	UUID

DATASTAX ACADEMY

Cassandra-Land

Table Summary

users_by_phone_number

Field	Type
phone_number	text
user_id	UUID
password	text

rider_count_by_time_and_ride

Field	Type
start_time	timestamp
ride_id	UUID
rider_count	int

ride_instances_by_start_time

Field	Type
start_time	timestamp
ride_id	UUID↑
ride_name	text
user_id	UUID↑
phone_number	text

ride_list_by_location

Field	Type
location	text
ride_id	UUID↑
ride_name	text
capactiy	int

ride_instances_by_user_id

Field	Type
user_id	UUID
start_time	timestamp↑
ride_id	UUID
ride_name	text

Last Three Use-Cases

What you need to know – PRIMARY KEY()

PRIMARY KEY((<partition key column>,…), <clustering column>,...)

- Must have one or more partition key columns
- May have zero or more clustering columns

Last Three Use-Cases

What you need to know – Timestamps

Format is ‘YYYY-MM-DDTHH:MM:SS’

- Notice the quotes

Last Three Use-Cases

What you need to know – UPDATE

```
UPDATE <keyspace name>.<table name>
      SET <assignment>
      WHERE <row specification>
IF <condition>
```

- Can have multiple <assignment>
- IF is optional – causes a lightweight transaction

Last Three Use-Cases

What you need to know – BATCH

```
BEGIN BATCH
      INSERT statement
      INSERT statement
      …
APPLY BATCH
```

- Causes all operations to complete
- Use for inserting into multiple tables

DATASTAX ACADEMY

Notebook Data Modeling: Cassandra-Land Project

Do Steps 3-5

Phone Number	Password
8017124927	JamiesPhone-NotReally
9999999999	NinesAreWild
1234567890	SeeQuential

DATASTAX ACADEMY

Cassandra-Land

What you learned!

- How to analyze use-cases to derive a data model
- How to denormalize to maintain performance
- How to use lightweight transactions
- How to leverage batch operations