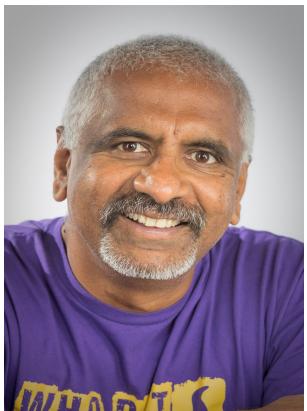


# Building an E-commerce Website on **Astra DB**

Session #1  
Product Catalog and Pricing



# Your presenters



**Raghavan Srinivas**  
Developer Advocate

*Apache Cassandra™ expert*

*All things distributed!*

*Still working on simplifying  
the developer "inner loop!"*



**Aaron Ploetz**  
Product Manager

*Author*



*Former SWE/DB Lead @  
W.W. Grainger, Target*



# DataStaxDevs: Developer Advocates Team



Cedrick  
Lunven

Aleksandr  
Volochnev

Jack  
Fryer

Kirsten  
Hunter

Stefano  
Lottini

David  
Gilardi

Ryan  
Welford

Rags  
Srinivas

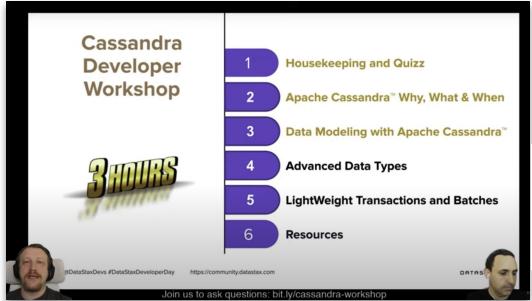
Sonia  
Siganporia

R

S

# Housekeeping #1: Attending the workshop

**Livestream:** [youtube.com/DataStaxDevs](https://youtube.com/DataStaxDevs)



**Questions:** <https://dtsx.io/discord>



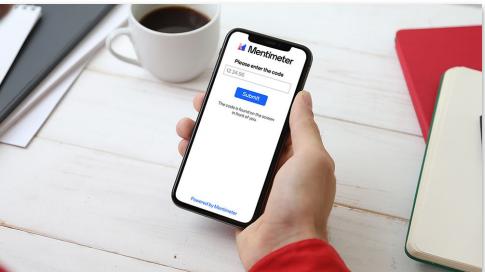
**Discord**



**YouTube**



**Games** [menti.com](https://menti.com)



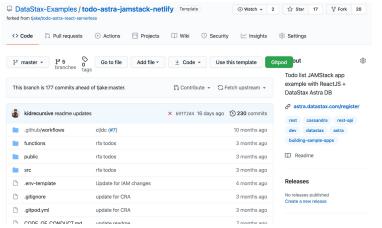
Mentimeter



Nothing to install !

# Housekeeping #2: Doing Hands-On

Source code + exercises + slides



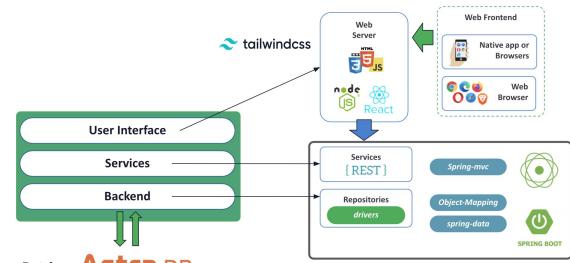
GitHub

IDE

A screenshot of an IDE (IntelliJ IDEA) showing the code for a Spring Boot application named 'StargateDemoApplication'. The code includes imports for Spring and Spring Boot annotations, and a main method that runs the application. The IDE also shows a terminal window with the command 'gitpod /workspace/workshop-spring-stargate'.

Gitpod

Website based on Astra DB



DataStax  
**Astra DB**

DataStax Developers

Welcome!

Astra DB

## Goals:

- Series on common e-tail use cases.
- Each session builds on previous.
- Show real, scalable solutions backed by:

**DataStax**



**Astra DB**

SPRING BOOT

# Agenda

DS

01



Architecture

02



Use Cases

03



Data Models

04



Service  
Layer

# Ecommerce – Technology Architecture/Choice

- Web and Mobile
- Javascript
- Angular
- Spring



SPRING BOOT



Which  
Technology ?

- Rest
- Native binary

STARGATE

CQL

Which API ?



WHY  
NoSQL ?

Not  
Only SQL

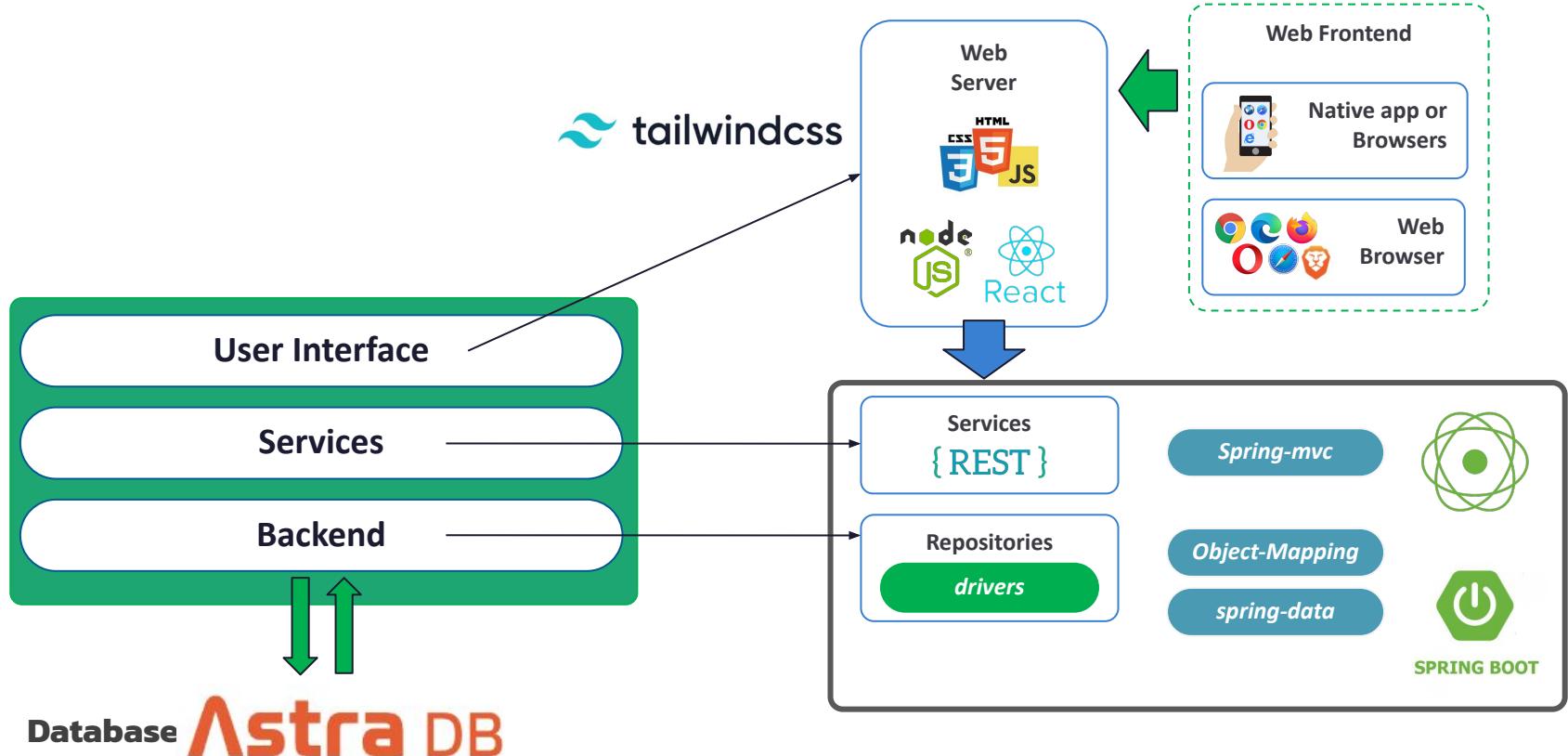
- Distributed System
- Variety of data
- High Volume
- High Throughput

Astra DB

WHY  
Astra DB ?

- Geographically Aware
- Vendor Choice
- Scalable
- Data Distribution

# Ecommerce – Solution and Opportunity



Database **Astra** DB

**Why Astra DB?**

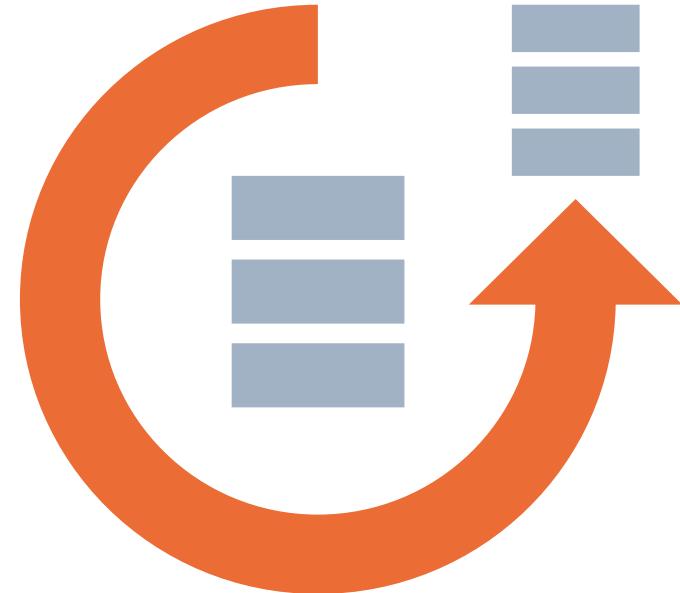
**Can other DBs handle this?**

## Database: Why not a RDBMS?

Astra DB

### RDBMS:

- Scaling (H/V) is limited.
- Data distribution is an add-on, rather than a core component.
- Geographic distribution is problematic.

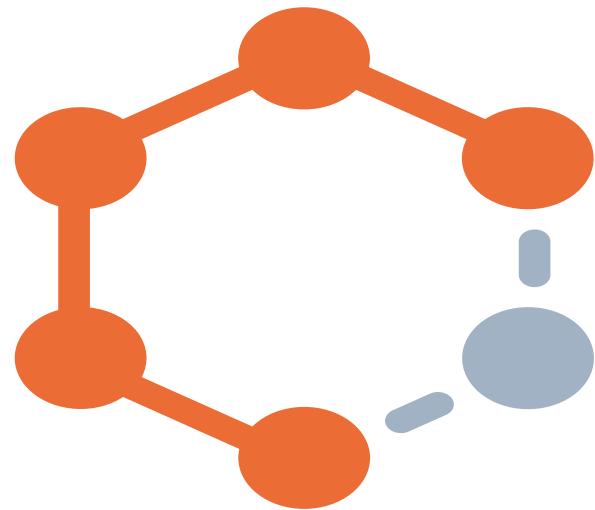


## Database: Why not a Document DB?

Astra DB

### Document Store:

- Strong consistency limits throughput.
- Reads become CPU-bound.
- Schema is actually important!
- Heterogeneous node types complicate DevOps.\*



\*Not an issue for “serverless” implementations.

## Database: Why Astra DB?

Astra DB

### Astra DB:

- No operational overhead
- Geographic Awareness
- Not cloud vendor bound
- “No touch” Scalability



# Jump to README step #2

<https://github.com/datastaxdevs/workshop-ecommerce-app#2-create-astra-db-instance>

# Agenda

DS

01



Architecture

02



Use Cases

03



Data Models

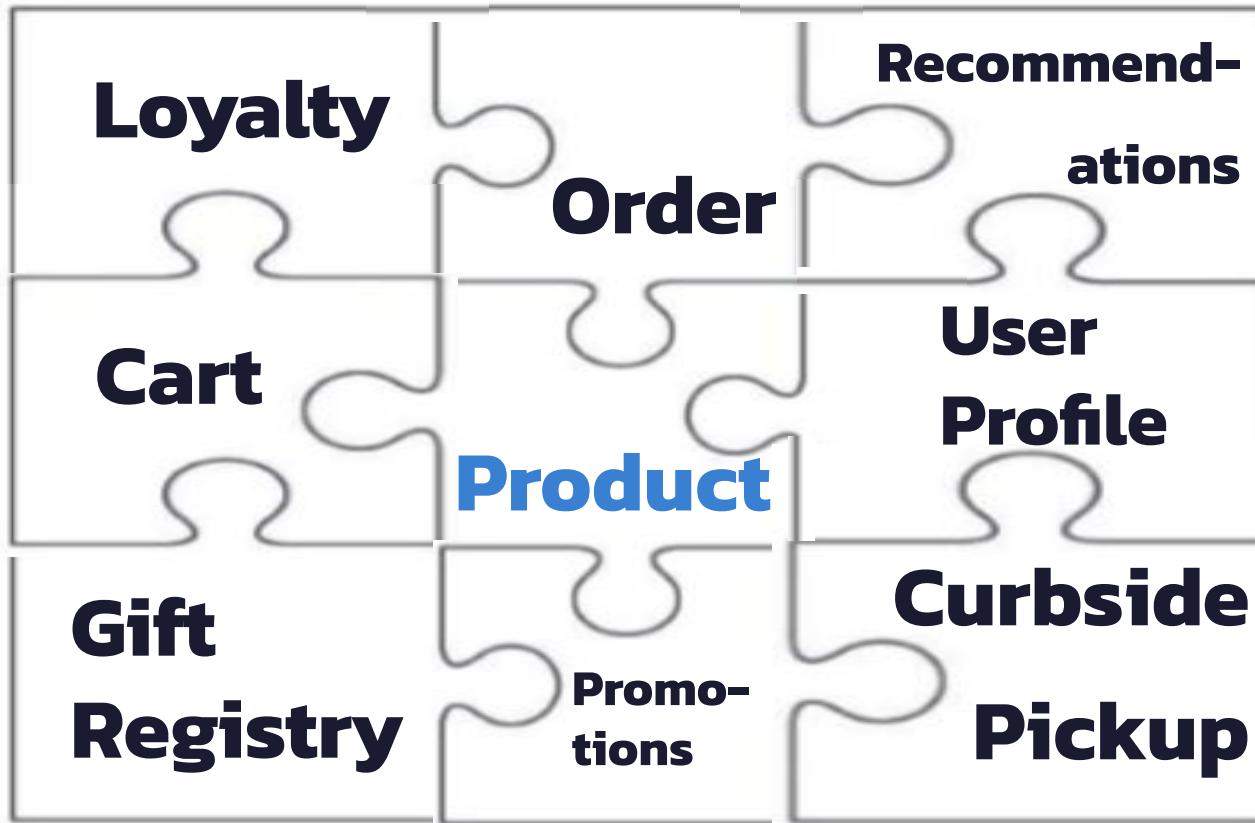
04



Service  
Layer

## Use Cases - Ecommerce Subsystems

Astra DB

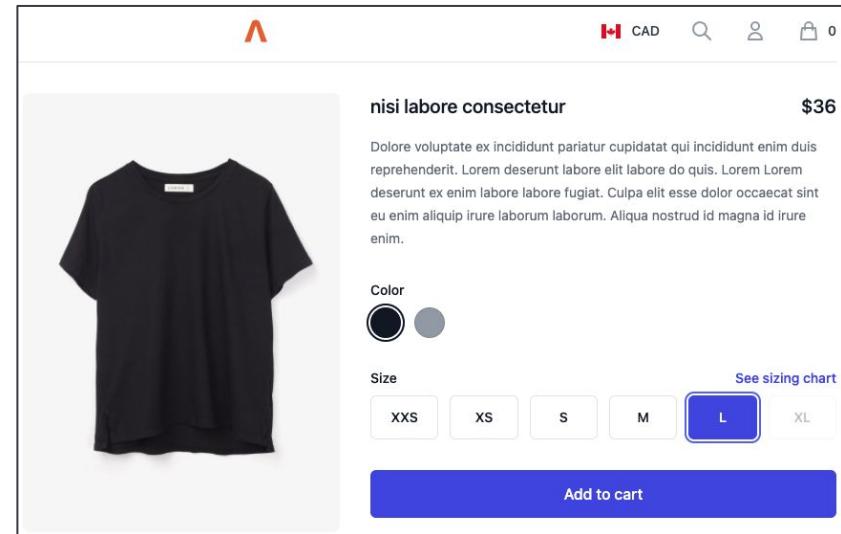


# Focus for today: Product Services

Astra DB

## Product Catalog

- Product (*data*)
- Product Category (*navigation*)
- Pricing



# Agenda

DS

01



Architecture

02



Use Cases

03



Data Models

04



Service  
Layer

# Data Modeling - Normalization

## Data modeling for:

- Efficient storage
- Data integrity

## Pros:

- Simple writes

## Cons:

- Slow reads
- Complex query model (JOINS)

Employees			
userId	deptId	firstName	lastName
1	1	Edgar	Codd
2	1	Raymond	Boyce

Departments	
departmentId	department
1	Engineering
2	Math

## Normalization

# **One table serves many queries**

# Data Modeling - Denormalization

## Data modeling for:

- Performance
- Tables built to suit a query

## Pros:

- Fast reads
- Simple query model

## Cons:

- Data duplication (*but that's ok*)
- Manual integrity enforcement

Employees			
userId	firstName	lastName	department
1	Edgar	Codd	Engineering
2	Raymond	Boyce	Engineering
3	Sage	Lahja	Math
4	Juniper	Jones	Botany

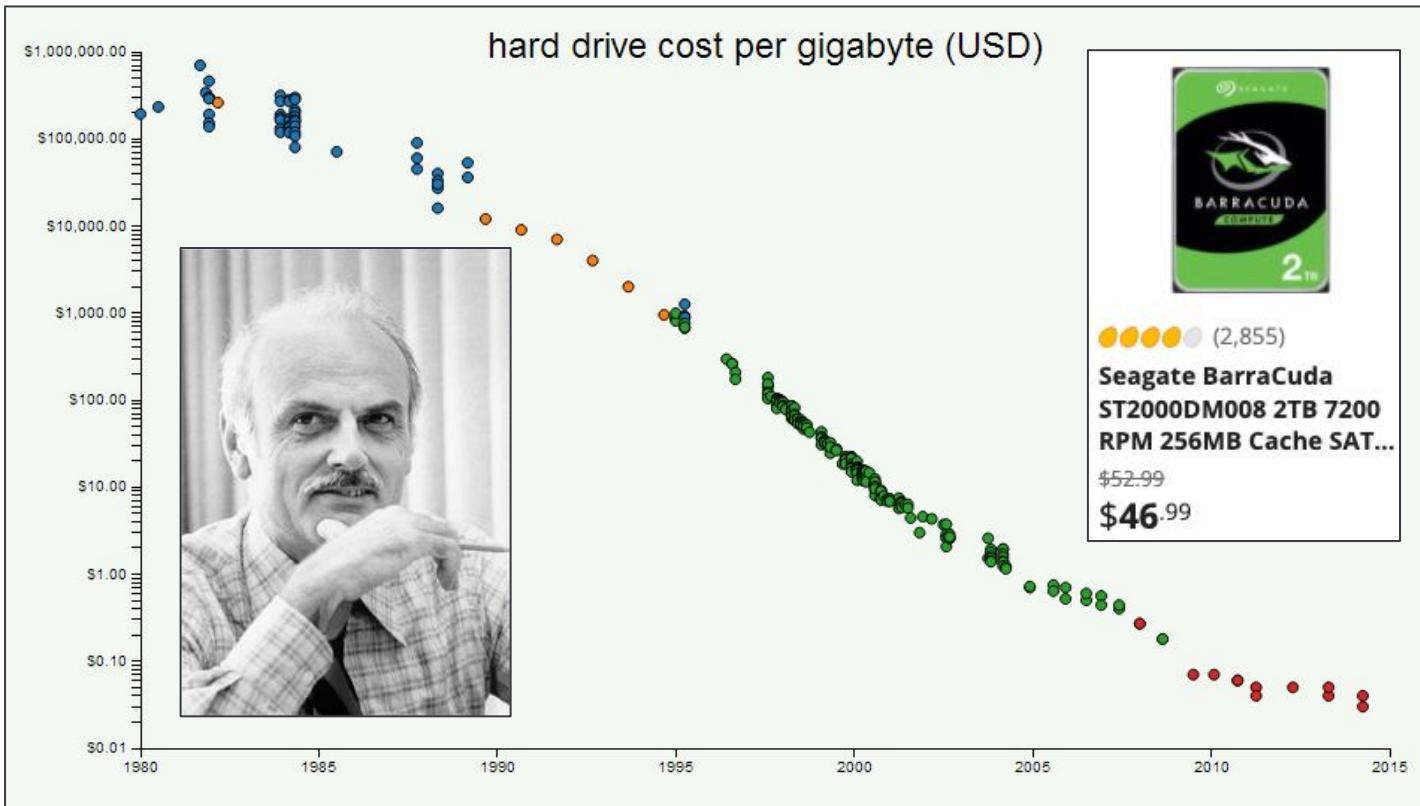
**Denormalization**

**One table serves one query**

Peter Brawley  
MySQL Forums, 2013

*"E.F. Codd must be rolling over in his grave. But by now, he should be used to that."*

# Data Duplication ok? Disk used to be Expensive!



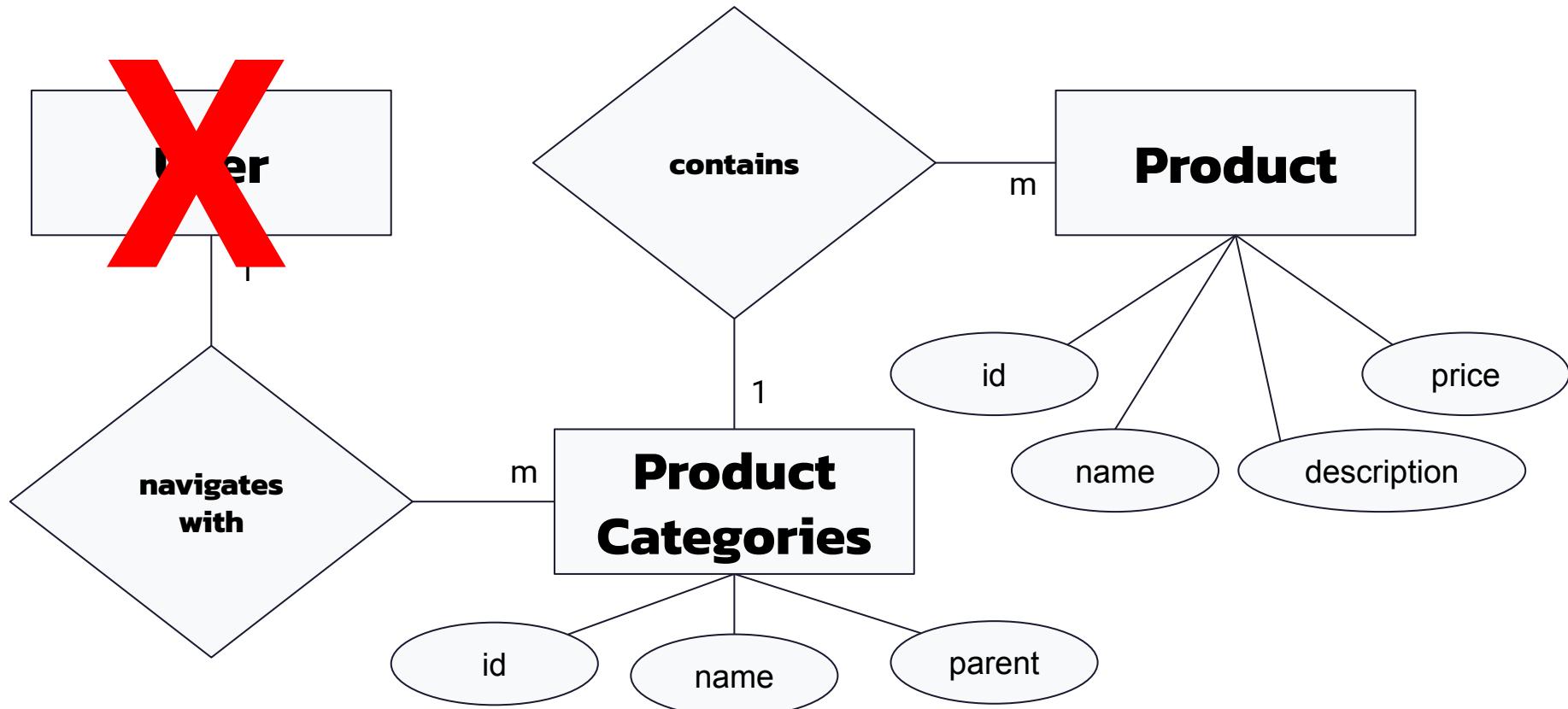
### Tips for Large Scale:

- Data **queried** together should be **stored** together.
- Use high-cardinality keys.
- **No** ALLOW FILTERING!
- Keep things **small!**
  - Partitions
  - Result sets



## Data Model - Conceptual

Astra DB

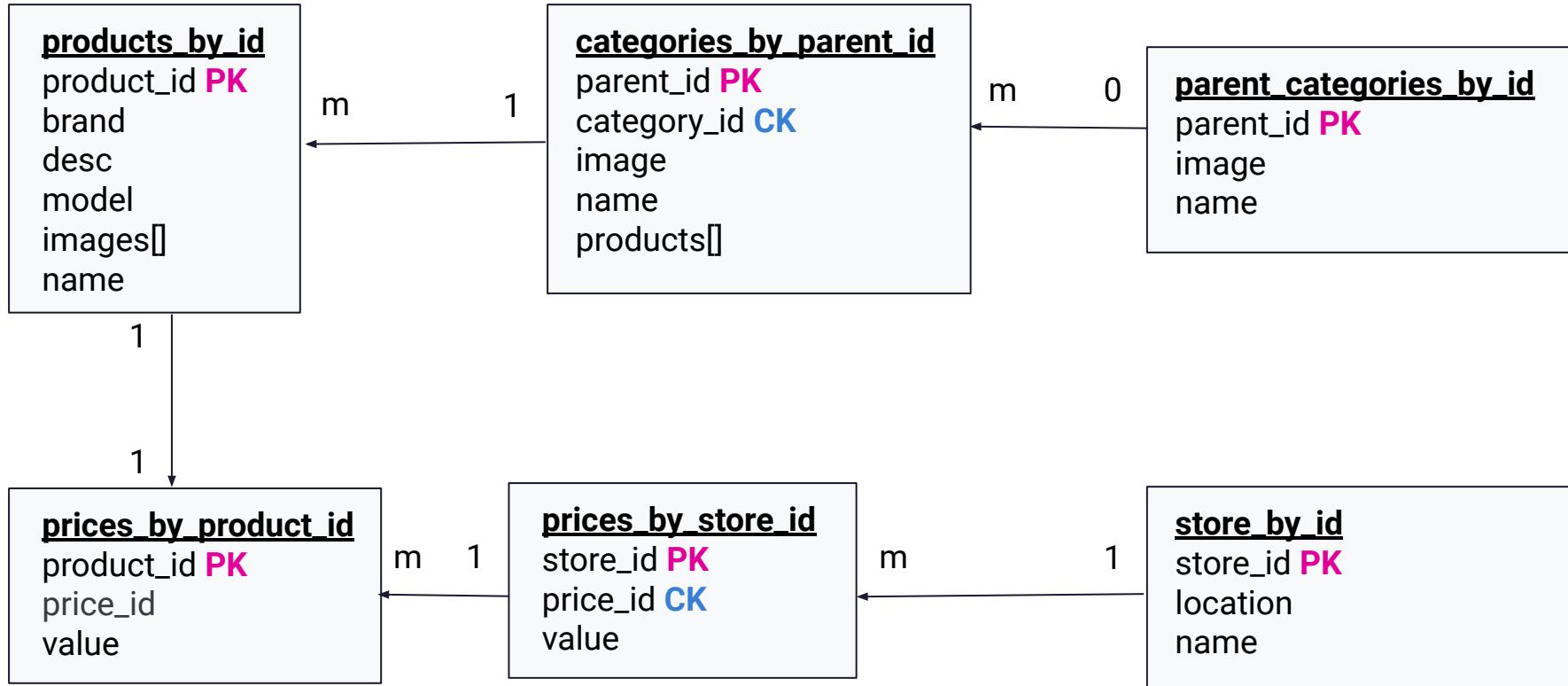


# Queries

- Need to be able to query a product by its identifier.
- Need to be able to query a product's "online" price.
- Need to be able to query product categories in a hierarchy.

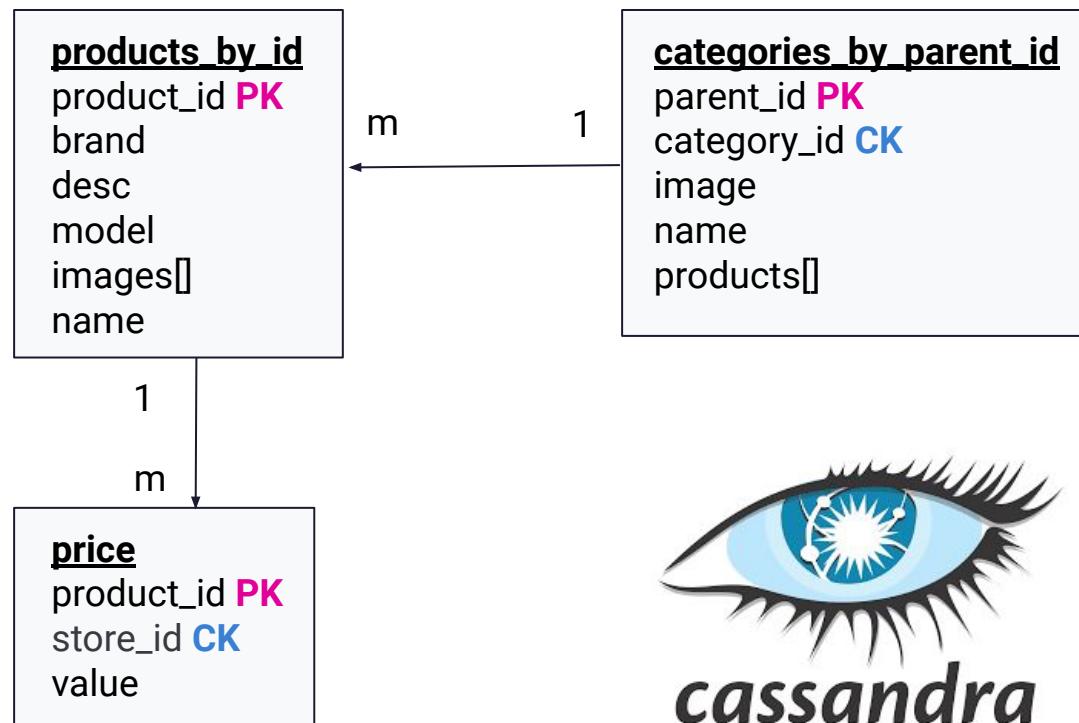
# Data Model - Logical

Astra DB



## Data Model - Physical

Astra DB



# Price

- Separate from product.
- PK (product\_id, store\_id)
- Data distribution a challenge for non-Cassandra DBs.

```
CREATE TABLE price (
    product_id TEXT,
    store_id TEXT,
    value DECIMAL,
PRIMARY KEY(product_id,store_id));
```

# Product

- Separate from price.
- PK (product\_id)
- Balancing of read traffic will be a challenge for non-Cassandra DBs.

```
CREATE TABLE product (
    product_id TEXT,
    product_group TEXT,
    name TEXT,
    brand TEXT,
    model_number TEXT,
    short_desc TEXT,
    long_desc TEXT,
    specifications MAP<TEXT, TEXT>,
    linked_documents MAP<TEXT, TEXT>,
    images SET<TEXT>,
    PRIMARY KEY(product_id));
```

# Category

- Designed to be called *recursively*.
- PK (parent\_id, category\_id)
- Data distribution a challenge for non-Cassandra DBs.

```
CREATE TABLE category (
    parent_id UUID,
    category_id UUID,
    name TEXT,
    image TEXT,
    products LIST<TEXT>,
    PRIMARY KEY
    (parent_id,category_id));
```

# Jump to README step #3

<https://github.com/datastaxdevs/workshop-ecommerce-app#3-create-your-schema>

# Agenda

DS

01



Architecture

02



Architecture

03



Data Models

04



Service  
Layer

# Price

GET

`/api/v1/prices/{productid}` Retrieve prices for a product from its id

GET

`/api/v1/prices/price/{productid}` Retrieve prices for a product and the default store 'web'

GET

`/api/v1/prices/price/{productid}/{storeId}` Retrieve price for a product and a store

## Service Endpoint - Product



### Price example:

```
> SELECT * FROM price WHERE product_id='LS534XL';
```

product_id	store_id	value
------------	----------	-------

LS534XL	web	14.99
---------	-----	-------

# Category

**GET**    `/api/v1/categories/{parentid}`   Retrieve Categories for a product from its parentId

**GET**    `/api/v1/categories/category/{parentid}/{categoryid}`   Retrieve category for a parentId and a categoryId

# Navigation example:

Start with top-level parent\_id:

ffdac25a-0244-4894-bb31-a0884bc82aa9

```
> SELECT name,category_id,products FROM category WHERE parent_id=ffdac25a-0244-4894-bb31-a0884bc82aa9;
```

name	category_id	products
Clothing	18105592-77aa-4469-8556-833b419dacf4	null
Wall Decor	591bf485-de09-4b46-8fd2-5d9dc7ca101e	null
Tech Accessories	5929e846-53e8-473e-8525-80b666c46a83	null
Cups and Mugs	675cf3a2-2752-4de7-ae2e-849471c29f51	null

## Service Endpoint - Category



### Navigation example:

2nd-level parent\_id for “Clothing”:

18105592-77aa-4469-8556-833b419dacf4

```
> SELECT name,category_id,products FROM category WHERE parent_id=18105592-77aa-4469-8556-833b419dacf4;
```

name	category_id	products
Hoodies	6a4d86aa-ceb5-4c6f-b9b9-80e9a8c58ad1	null
T-Shirts	91455473-212e-4c6e-8bec-1da06779ae10	null
Jackets	d887b049-d16c-46e1-8c94-0a1280dedc30	null

## Service Endpoint - Category



### Navigation example:

3rd-level parent\_id for “Hoodies”:

6a4d86aa-ceb5-4c6f-b9b9-80e9a8c58ad1

```
SELECT name, category_id, products FROM category WHERE parent_id=6a4d86aa-ceb5-4c6f-b9b9-80e9a8c58ad1;
```

name	category_id	products
DataStax Vintage 2015 MVP Hoodie	86d234a4-6b97-476c-ada8-efb344d39743	[ 'DSH915S', 'DSH915M', 'DSH915L', 'DSH915XL' ]
DataStax Black Hoodie	b9bed3c0-0a76-44ea-bce6-f5f21611a3f1	[ 'DSH916S', 'DSH916M', 'DSH916L', 'DSH916XL' ]

# Product

GET

/api/v1/products/product/{productid} Retrieve product details from its id

# Service Endpoint – Product



## Product example:

Application

Astra DB

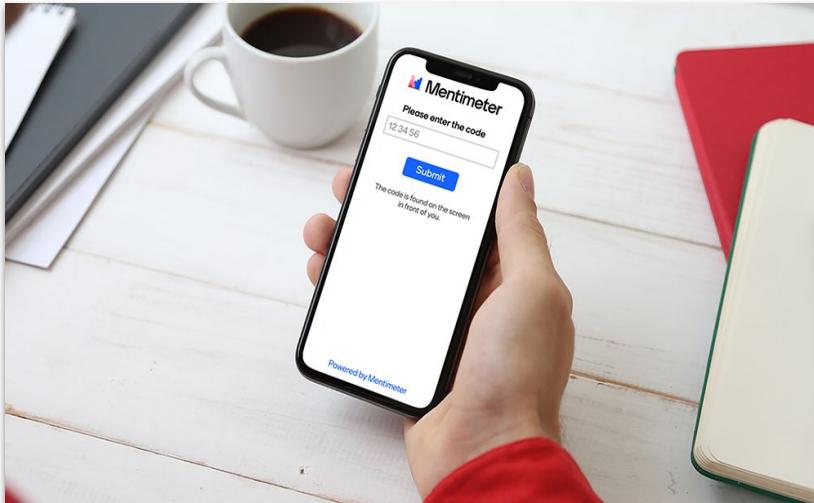


## Jump to README step #6

[https://github.com/datastaxdevs/workshop-ecommerce-app  
#6-setup-your-application](https://github.com/datastaxdevs/workshop-ecommerce-app/blob/main/README.md#6-setup-your-application)

## Quiz Time – Menti.com

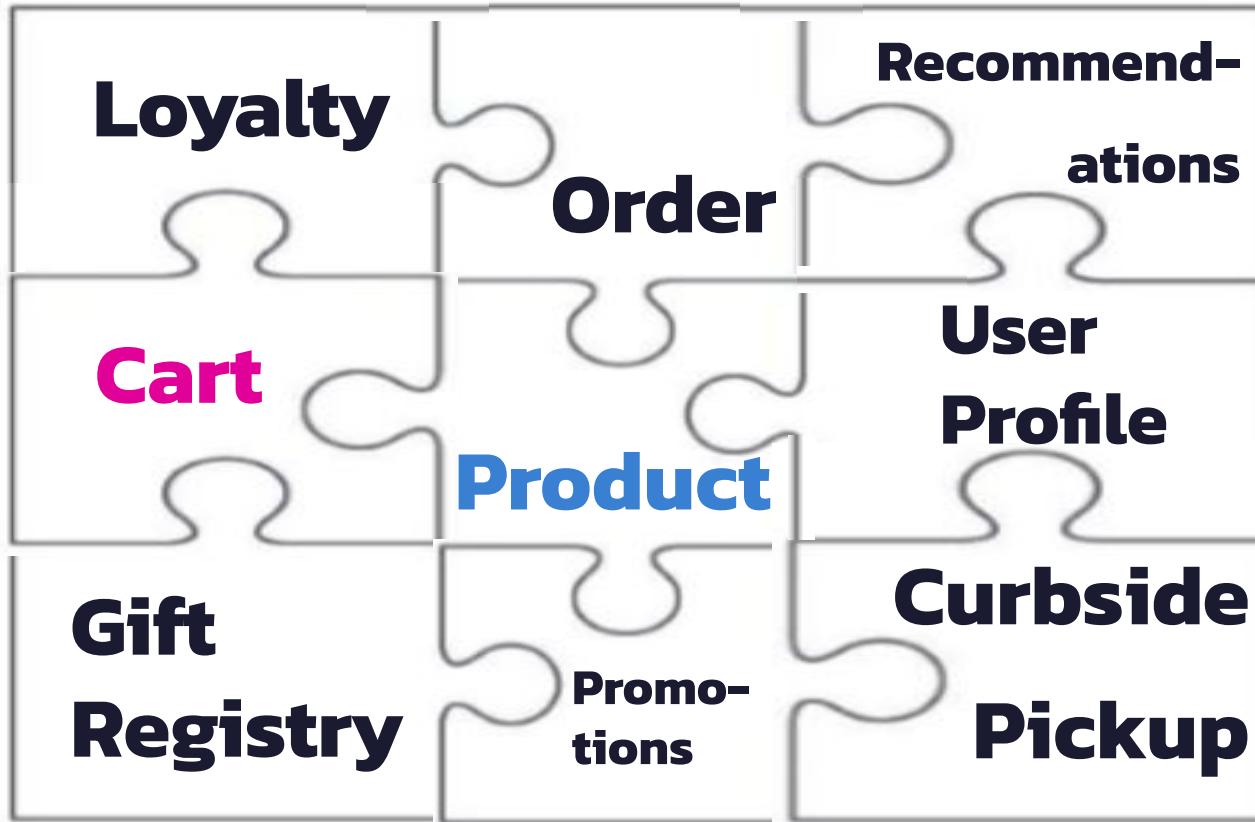
Astra DB



GET IT ON  
Google play

Future Sessions - Stay Tuned!

Astra DB



DataStax



Thank You!