DATASTAX ACADEMY

DS220

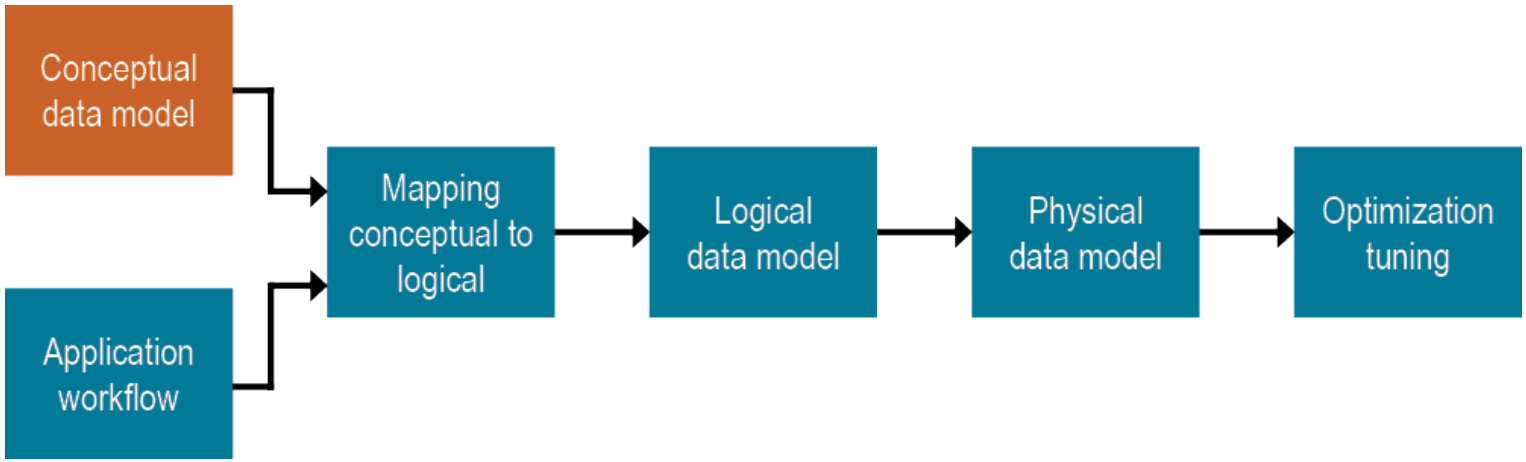
04: Cassandra Data Modeling Methodology

DATASTAX ACADEMY

Conceptual Data Modeling

Data Modeling Methodology Review

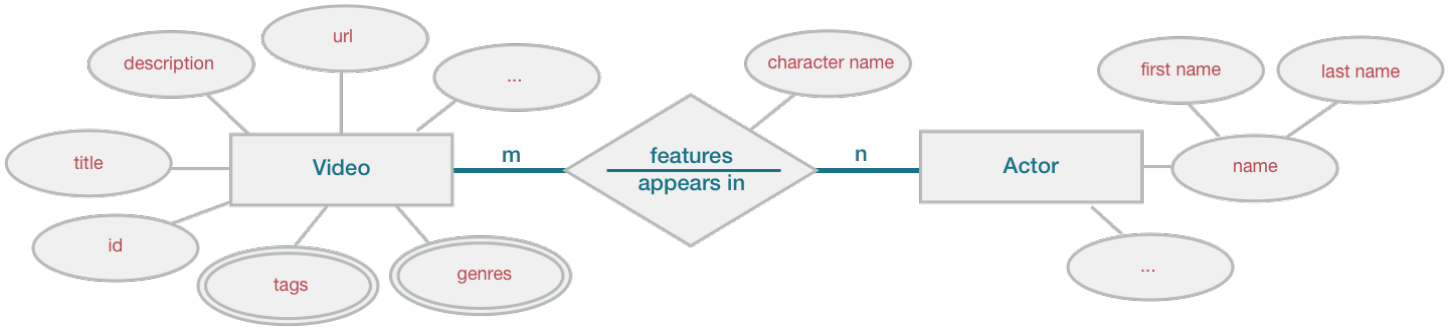
Where are we now?



Conceptual Data Model

Modeling your domain

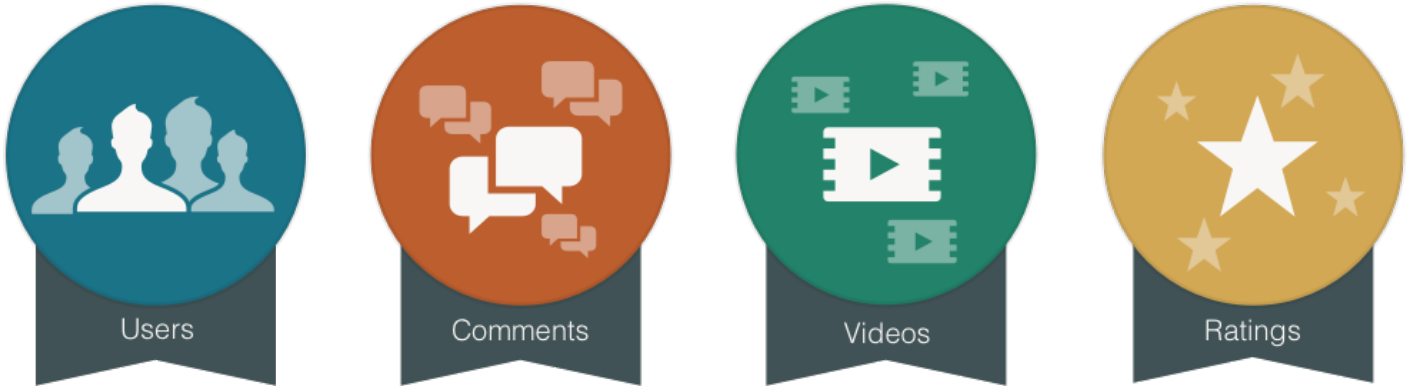
- Abstract view of your domain
- Technology independent
- Not specific to any database system



Purpose of Conceptual Modeling

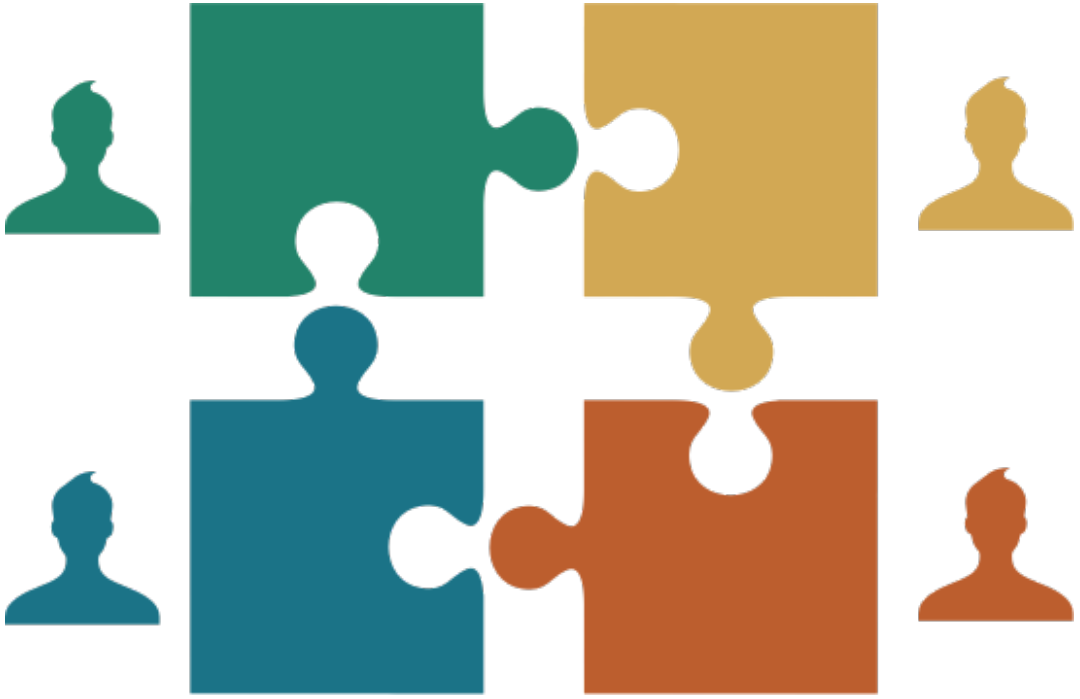
Why do we start here?

- Understand your data
- Essential objects
- Constraints



Advantages of Conceptual Modeling

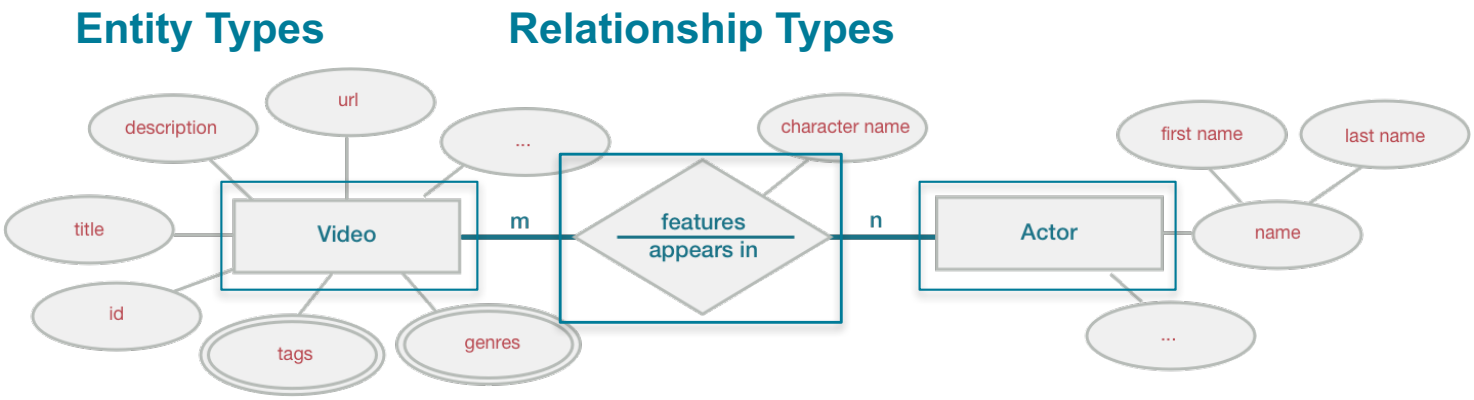
Collaboration



DATASTAX ACADEMY

Entity-Relationship (ER) Model

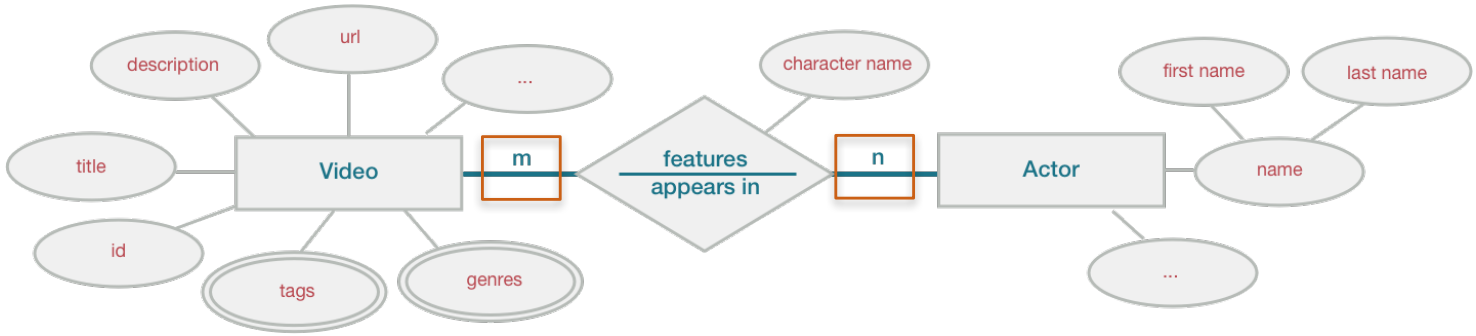
Entity Types - Relationship Types - Attribute Types



Cardinality

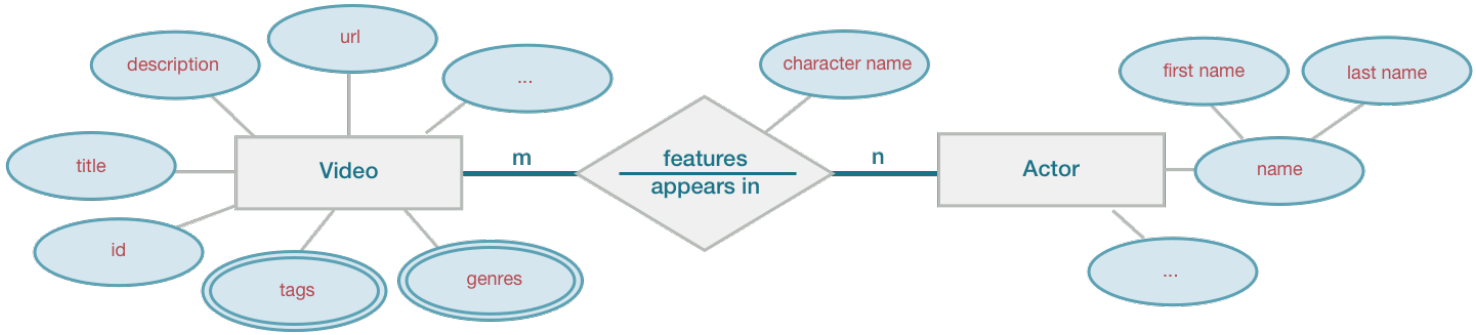
Relationships between entities

- Number of times an entity can/must participate in the relationship
- Other possibilities:
  - 1-n
  - 1-1



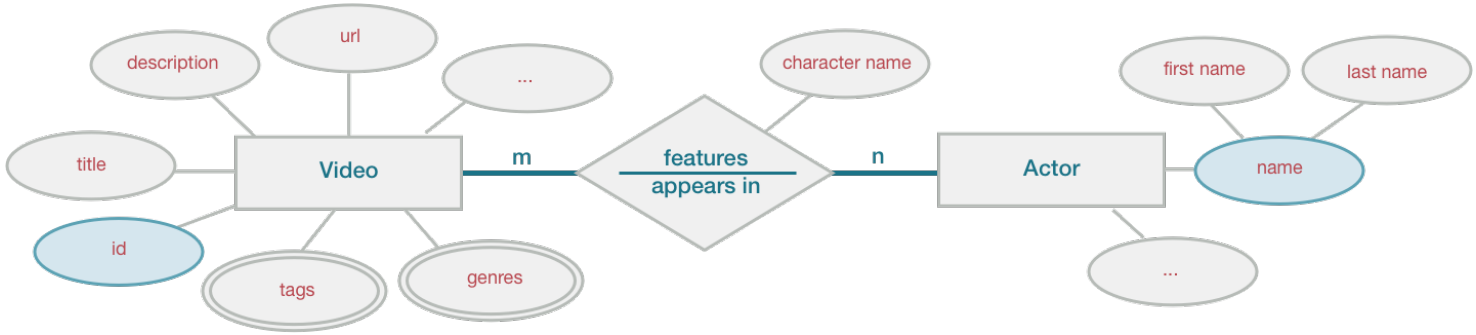
Attribute Types

Fields to store data about an entity or relationship



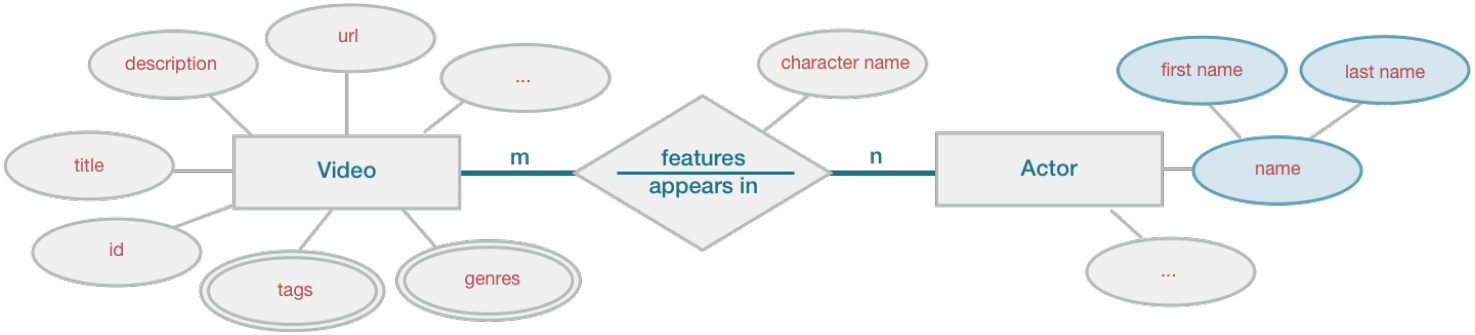
Key Attributes

Identifies an object



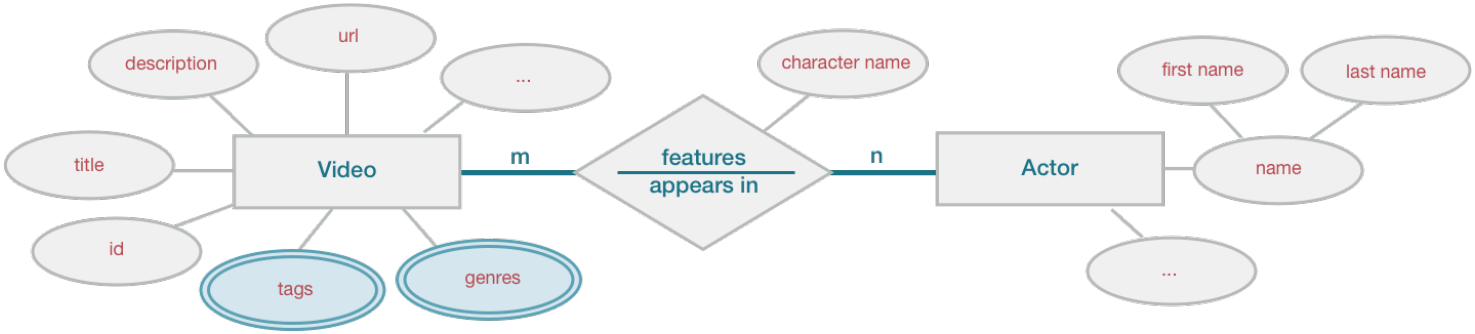
Composite Attributes

Groups related attributes together



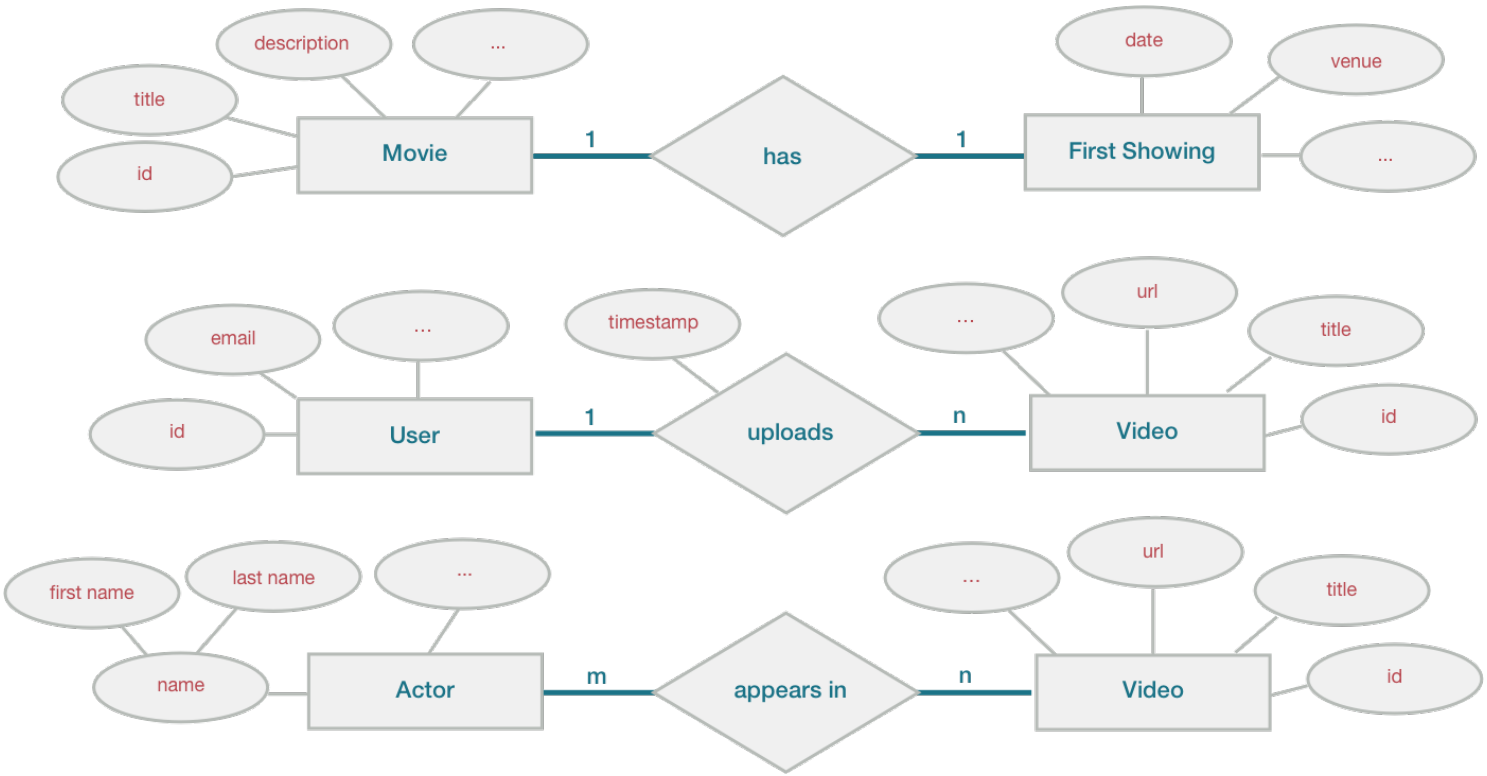
Multi-valued Attributes

Attribute stores multiple values per entity



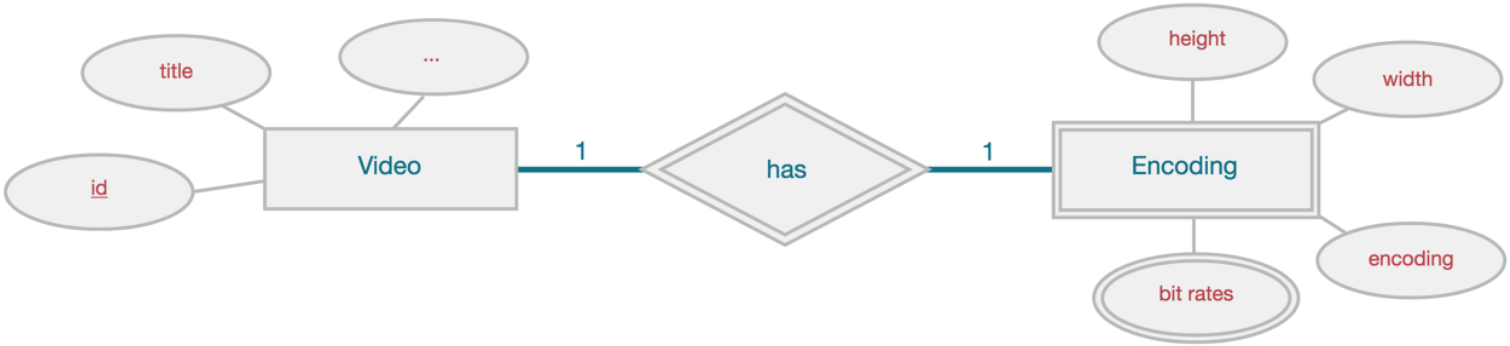
DATASTAX ACADEMY

Relationship Keys



Weak Entity Types

Cannot exist without an identifying relationship to a strong entity type

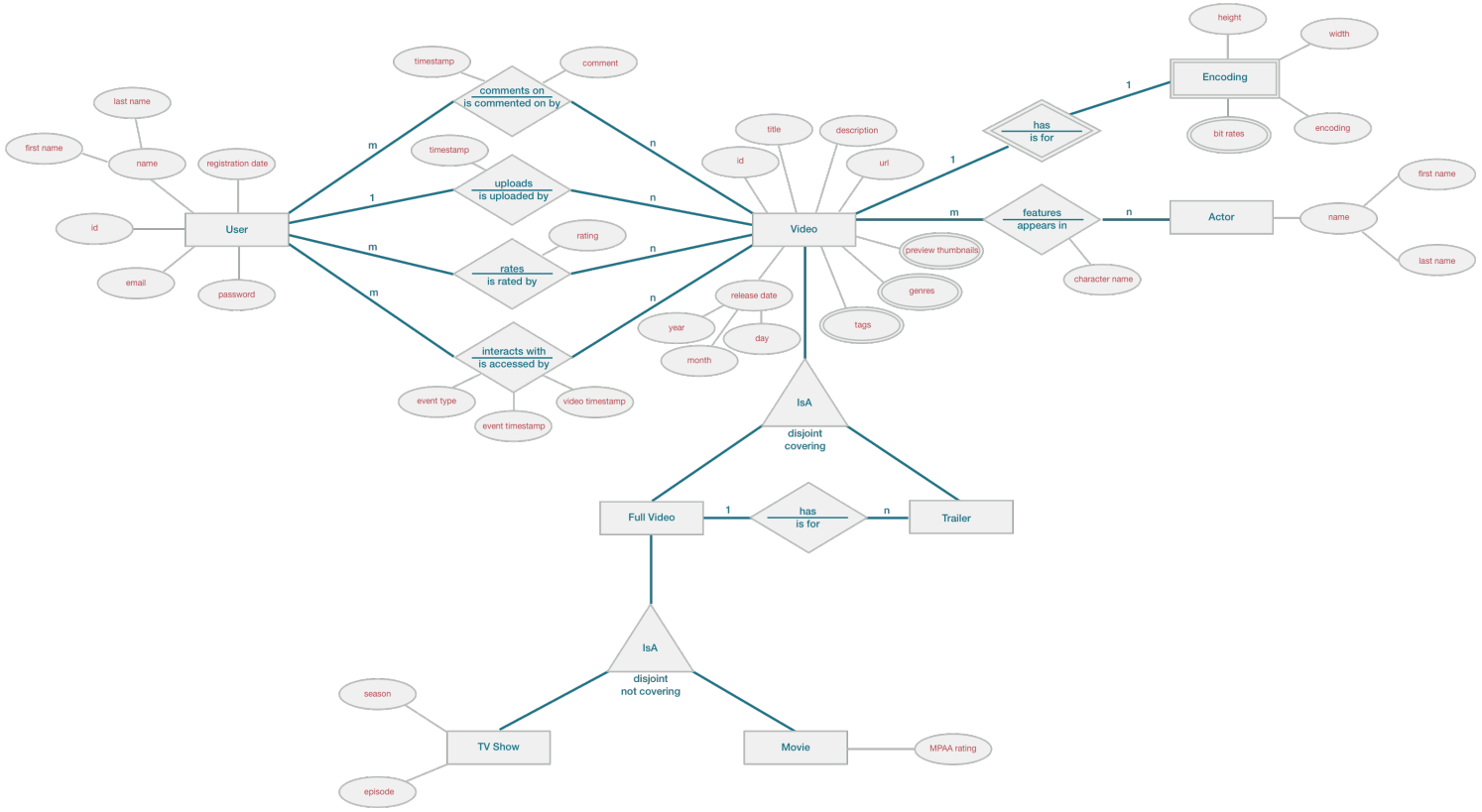


DATASTAX ACADEMY

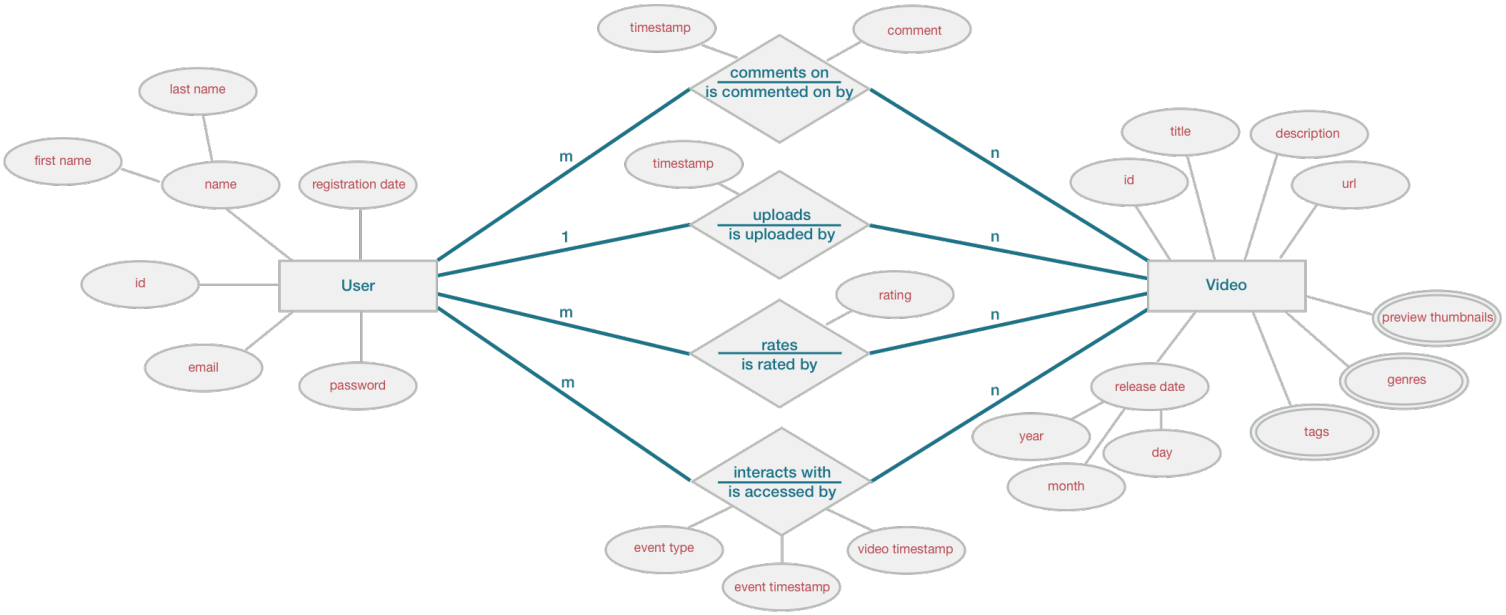
Exercise 4.1: Finish a Conceptual Data Model

DATASTAX ACADEMY

Final Conceptual Model



Final Conceptual Model

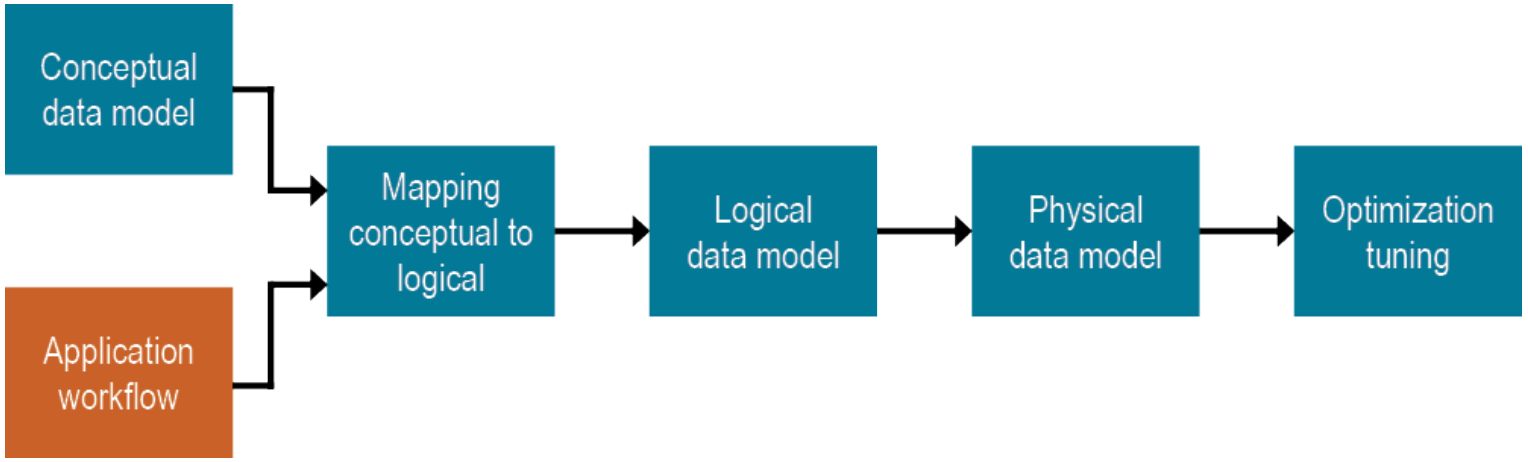


DATASTAX ACADEMY

Application Workflow and Access Patterns

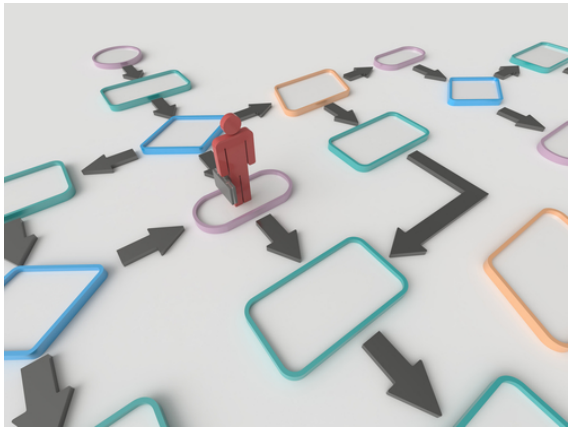
Data Modeling Methodology Review

Where are we now?



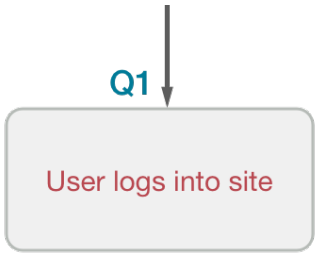
Application Workflow

- Each application has a workflow—Tasks/causal dependencies form a graph
- Access patterns help determine how data is accessed—Know what queries you will run first
- Example Task: Have a user login to a site



Application Workflow, Cont.

Task: User logs into a site



ACCESS PATTERNS

Q1: Find a user with a **specified email**

Application Workflow, Cont.

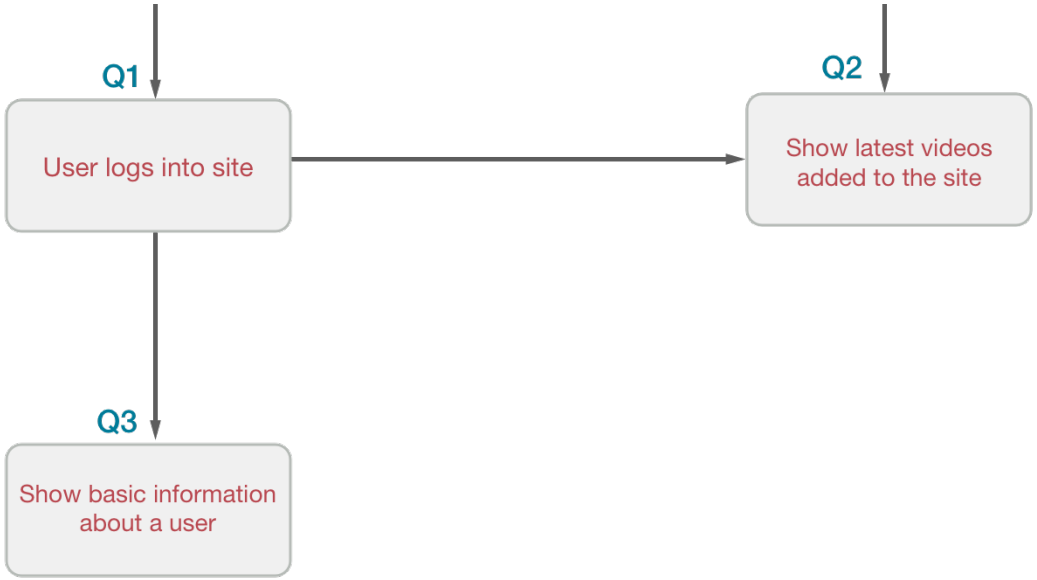
Task: Show videos that were most recently uploaded

ACCESS PATTERNS

Q1: Find a user with a **specified email**
Q2: Find most recently uploaded **videos**

Application Workflow, Cont.

Task: Give me information about a particular user

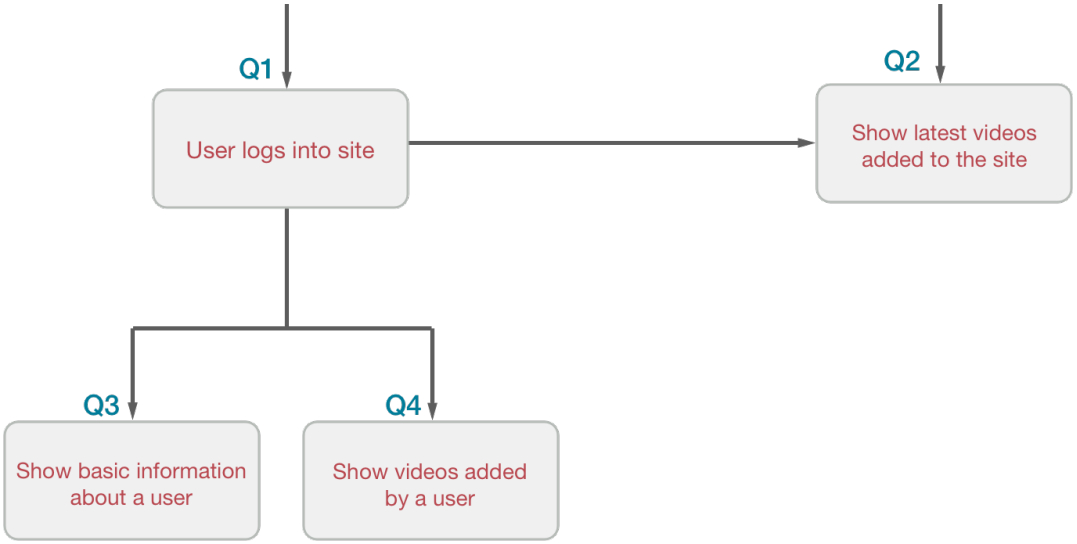


ACCESS PATTERNS

Q1: Find a user with a **specified email**
Q2: Find most recently uploaded **videos**
Q3: Find a **user with a specified id**

Application Workflow, Cont.

Task: Show videos that were uploaded by a particular user

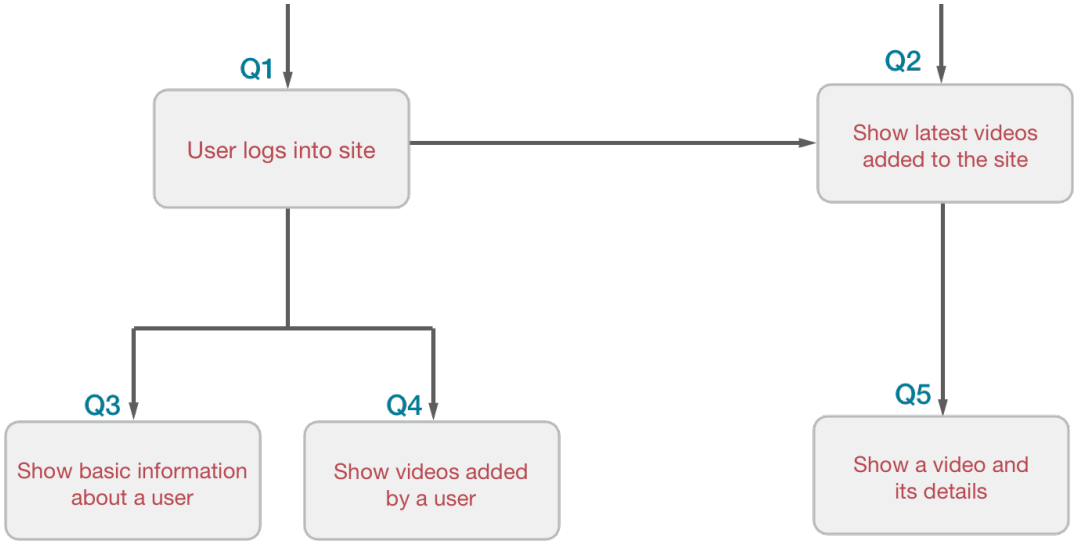


ACCESS PATTERNS

Q1: Find a user with a **specified email**
Q2: Find most recently uploaded **videos**
Q3: Find a **user with a specified id**
Q4: Find videos uploaded by a **user with a known id**(show most recently uploaded videos first)

Application Workflow, Cont.

Task: Find a video that has a particular video id



ACCESS PATTERNS

Q1: Find a user with a **specified email**
Q2: Find most recently uploaded **videos**
Q3: Find a **user with a specified id**
Q4: Find videos uploaded by a **user with a known id**(show most recently uploaded videos first)
Q5: Find a video with a **specified video id**

DATASTAX ACADEMY

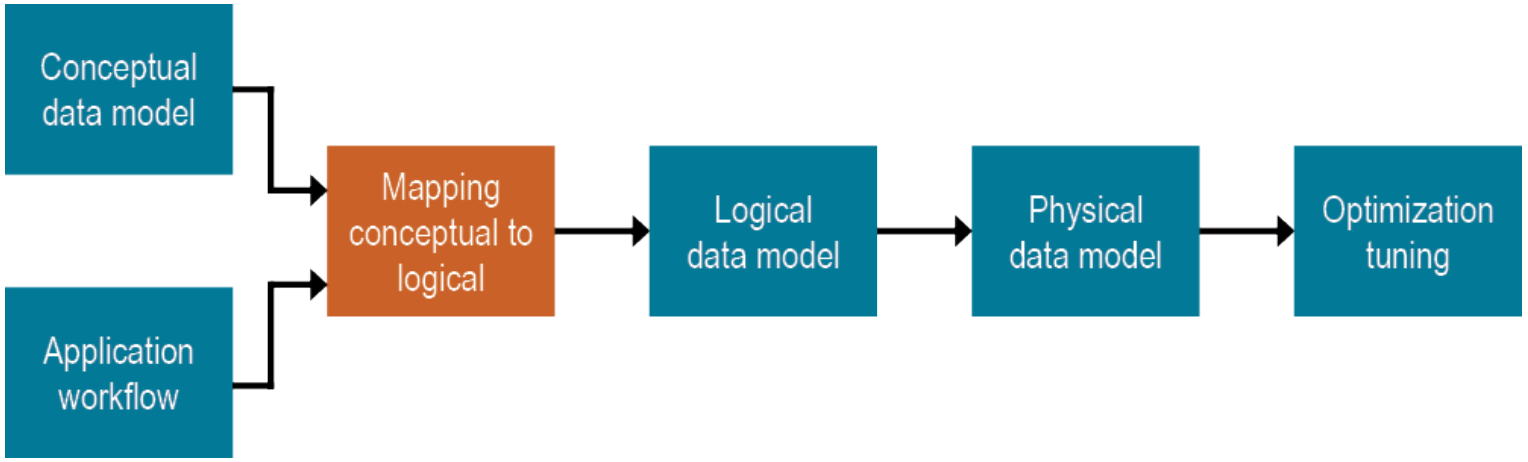
Exercise 4.2: Application Workflow and Access Patterns

DATASTAX ACADEMY

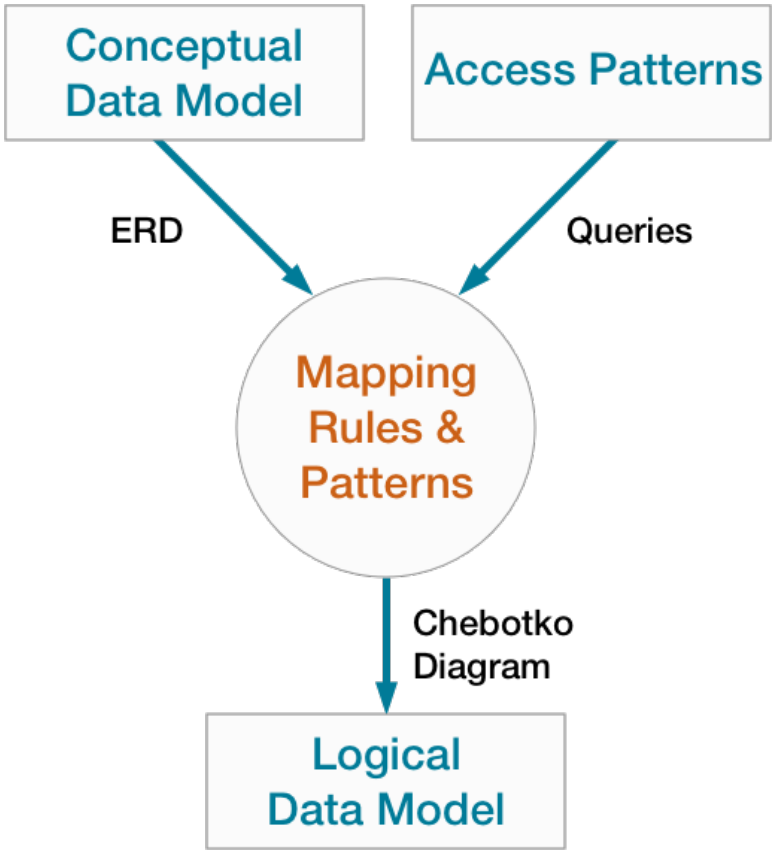
Mapping Conceptual to Logical Model

Data Modeling Methodology Review

Where are we now?

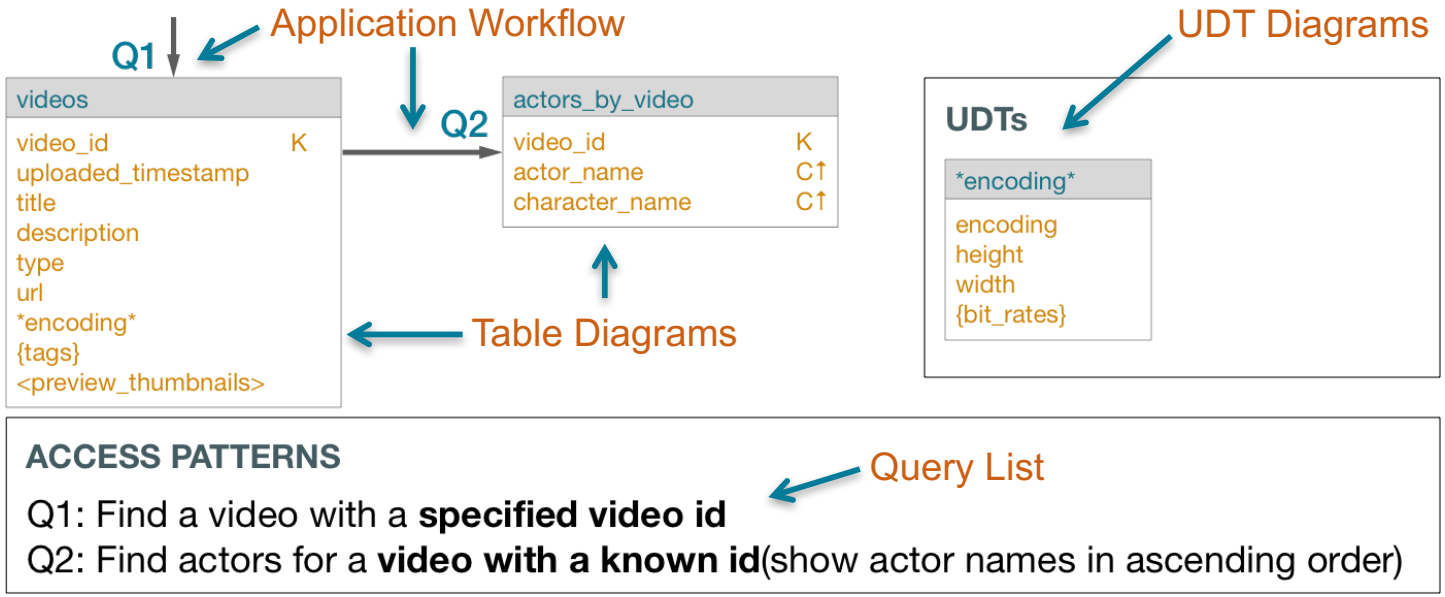


Query-Driven Data Modeling



Chebotko Diagrams

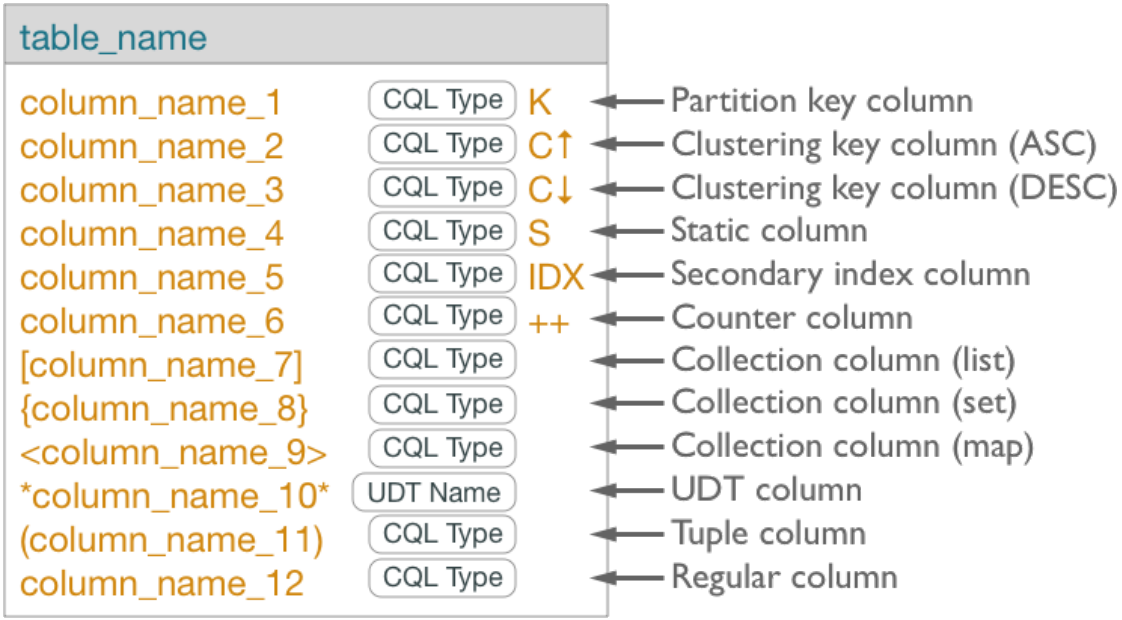
- Graphical representation of Apache Cassandra(TM) database schema design
- Documents the logical and physical data model



Chebotko Diagram Notation

Table representation

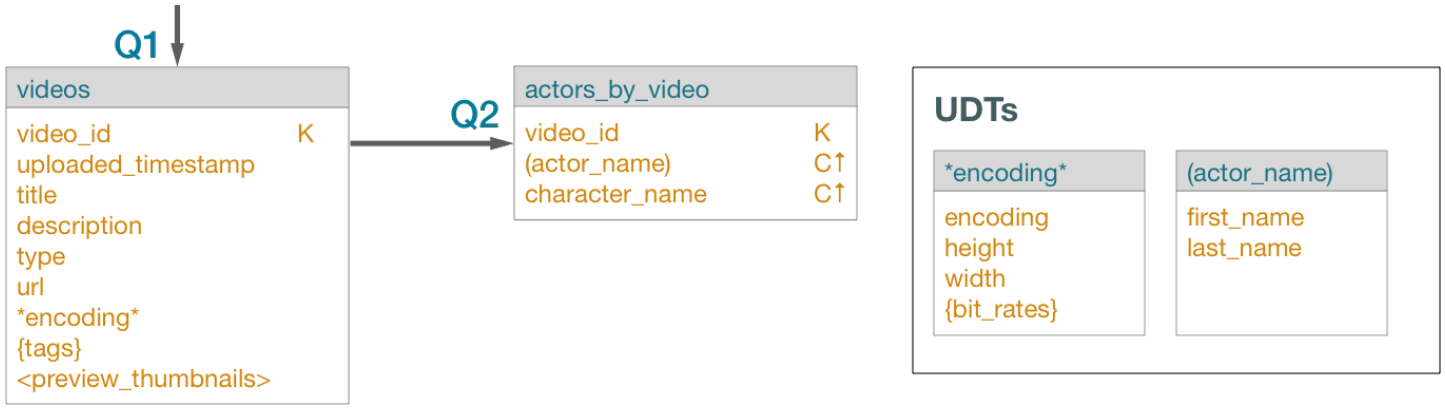
- Logical-level shows column names and properties
- Physical-level also shows the column data type



Chebotko Diagram Notation

Logical UDT Diagram

- Represents user defined types and tuples

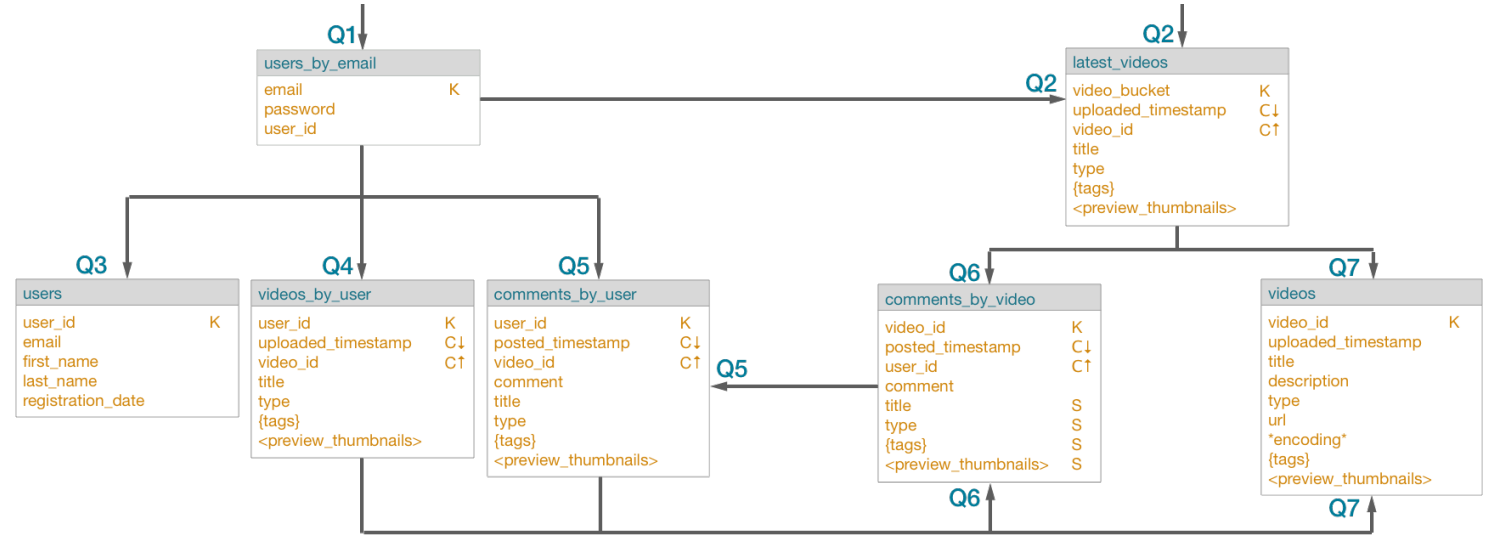


Chebotko Diagram Notation

Physical UDT diagram

- Represents user defined types and tuples

Example Chebotko Diagram



ACCESS PATTERNS

Q1: Find a user with a **specified email**
Q2: Find most recently uploaded videos
Q3: Find a **user with a specified id**
Q4: Find videos uploaded by a **user with a known id**(show most recently uploaded videos first)
Q5: Find comments posted by a **user with a known id**(show most recently posted comments first)
Q6: Find comments posted for a **video with a known id**(show most recent comments first)
Q7: Find a video with a **specified video id**

UDTs

encoding
encoding
height
width
{bit_rates}

Data Modeling Principles

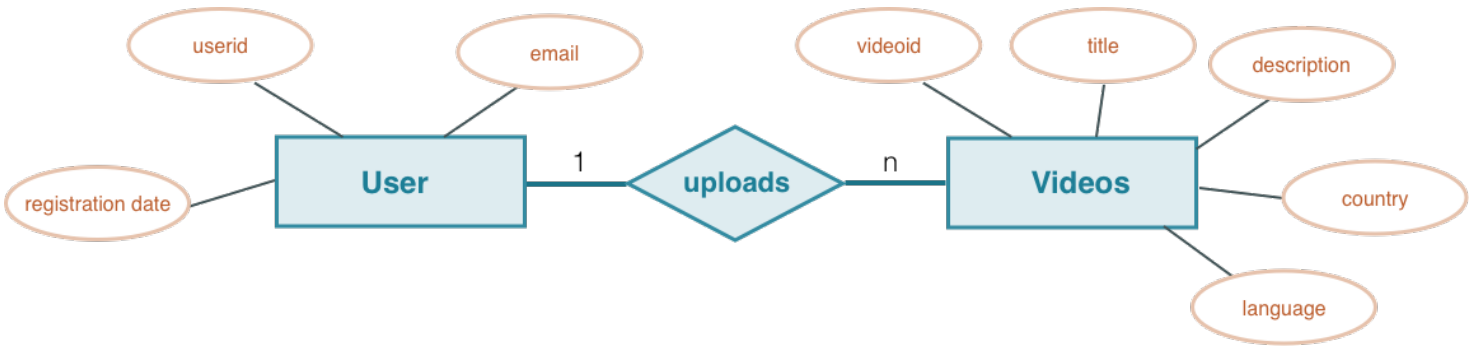
Apache Cassandra™ Principles

- Know your data
- Know your queries
- Nest data
- Duplicate data

Know Your Data

Understanding your data is key to successful design

- Data captured by conceptual data model
- Define what is stored in database
- Preserve properties so that data is organized correctly



Know Your Data

Key constraints affect schema design

- Entity and relationship keys affect the table primary keys
- Primary key uniquely identifies a row / entity / relationship
- Composed of a key and possibly additional columns

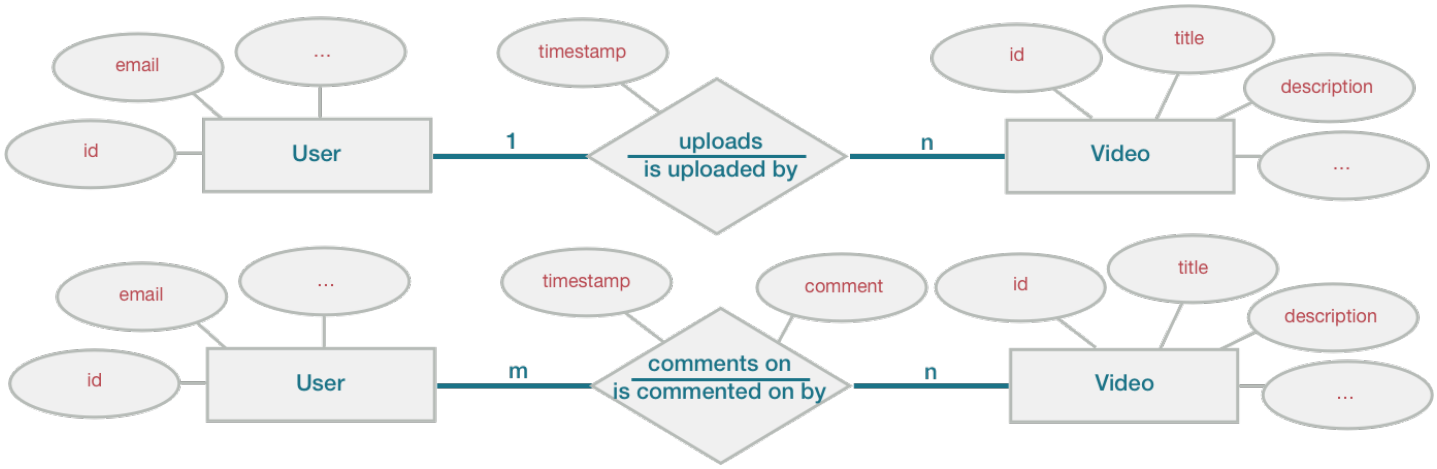
videos	
video_id	K
uploaded_timestamp	
user_id	
title	
description	
type	
encoding	
{tags}	
<preview_thumbnails>	
{genres}	

videos_by_user	
user_id	K
uploaded_timestamp	C↓
video_id	C↑
title	
type	
{tags}	
<preview_thumbnails>	

DATASTAX ACADEMY

Know Your Data

Cardinality constraints affect the key for relationships



Know Your Queries

Queries directly affect schema design

- Queries captured by application workflow model
- Table schema design changes if queries change

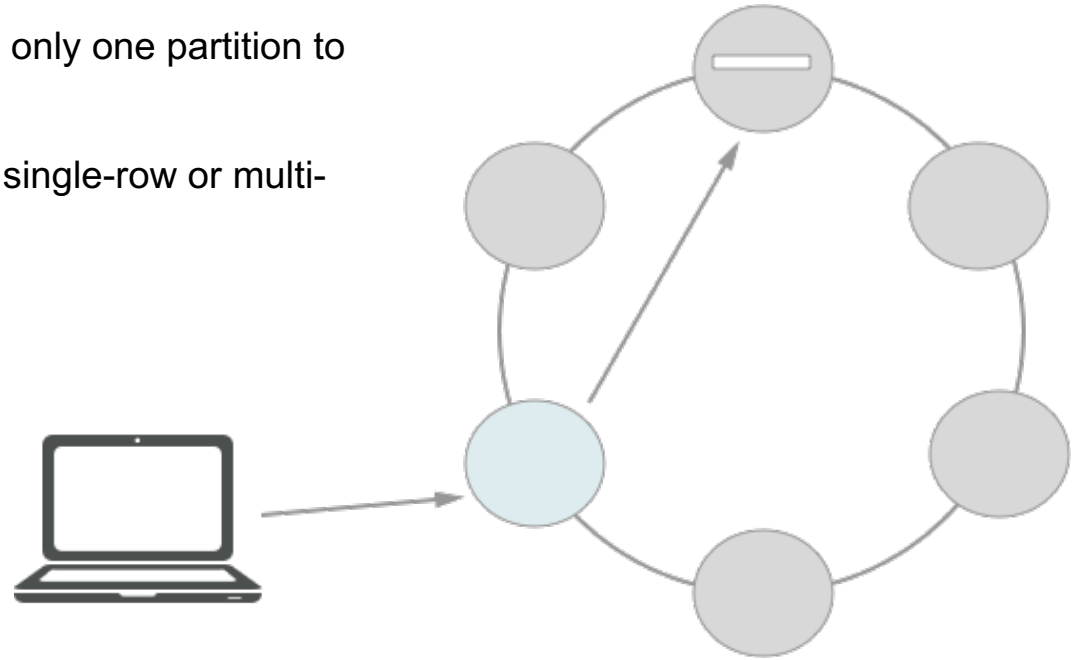
ACCESS PATTERNS

Q1: Find a user with a **specified email**
Q2: Find most recently uploaded **videos**

Single Partition Per Query—Ideal

Schema design organizes data to efficiently run queries

- Most efficient access pattern
- Query accesses only one partition to retrieve results
- Partition can be single-row or multi-row



Partition+ Per Query—Acceptable

Schema design organizes data to efficiently run queries

- Less efficient access pattern but not necessarily bad
- Query needs to access multiple partitions to retrieve results

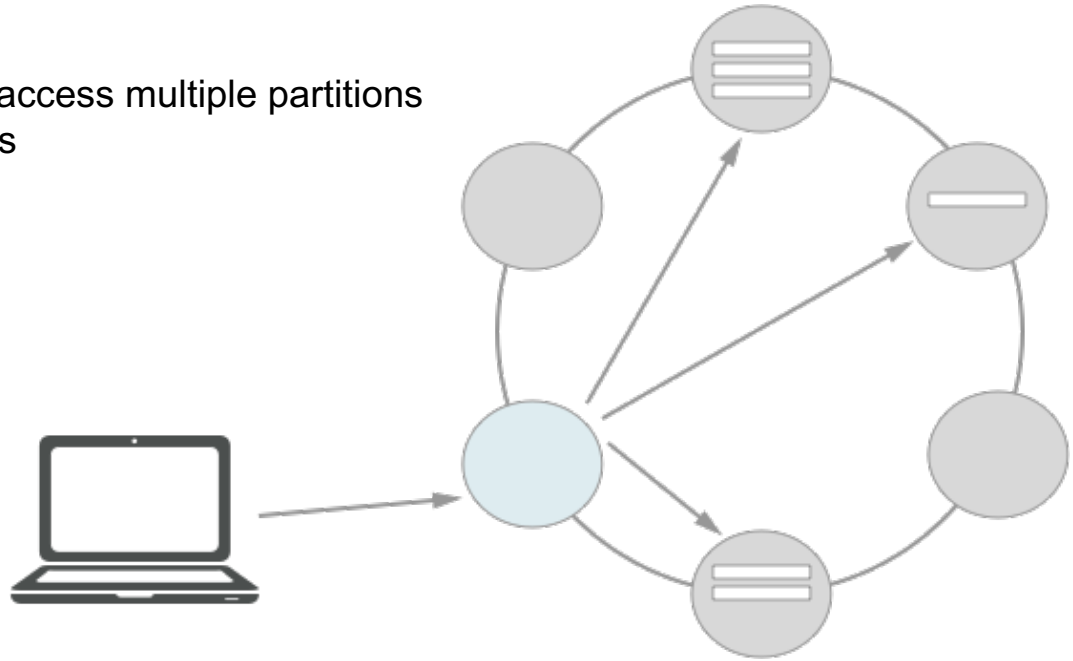
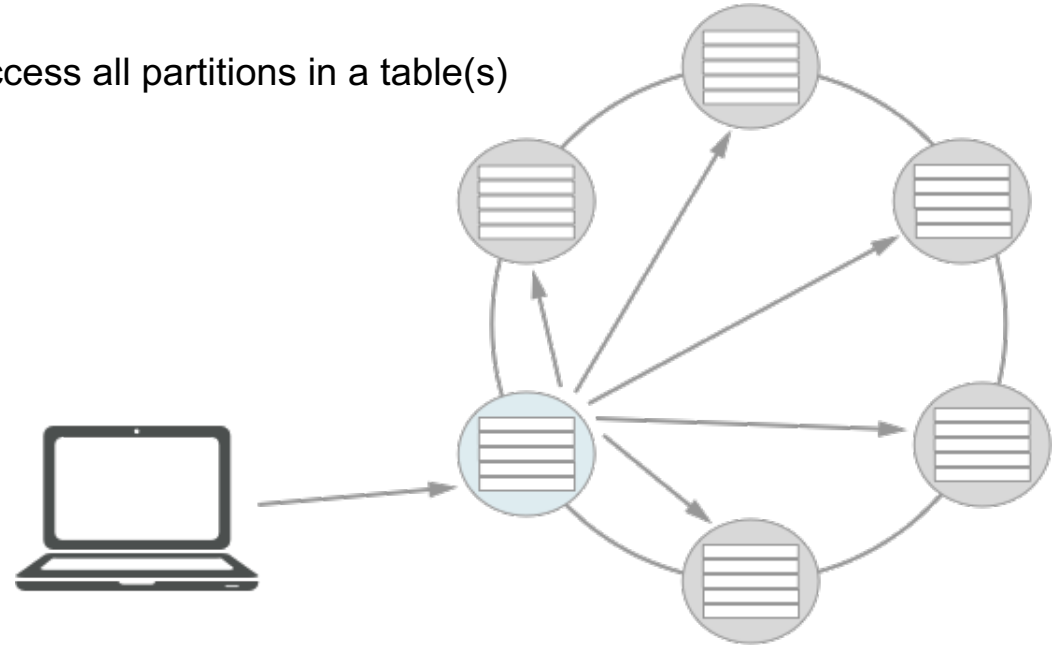


Table Scan/Multi-Table Scan—Anti-Pattern

Schema design organizes data to efficiently run queries

- Least efficient type of query but may be needed in some cases
- Query needs to access all partitions in a table(s) to retrieve results

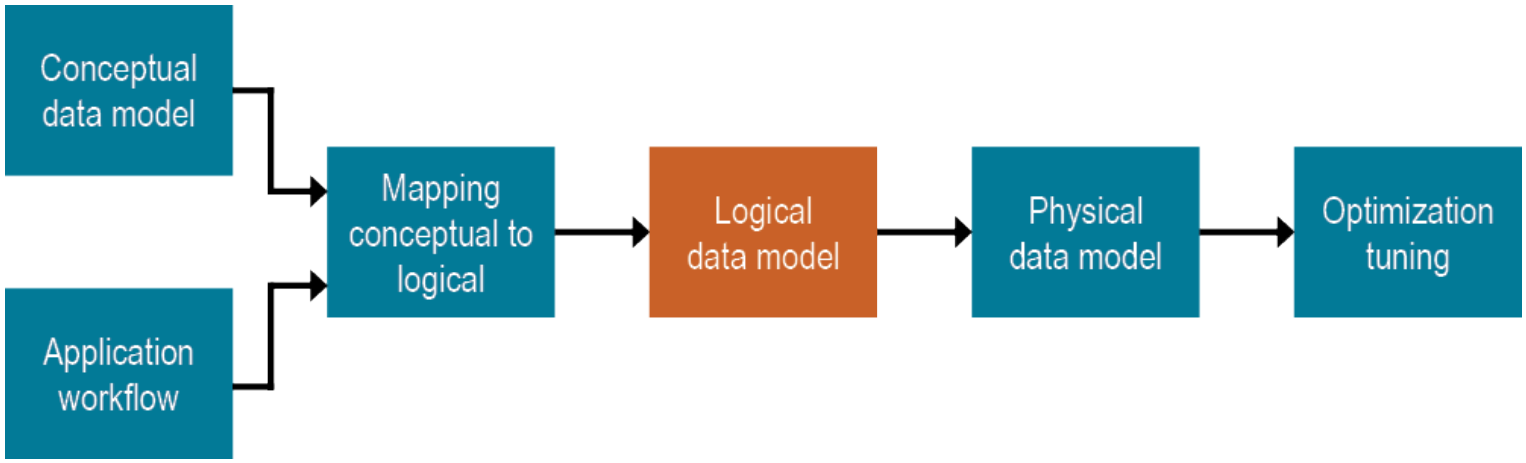


DATASTAX ACADEMY

Logical Data Modeling

Data Modeling Methodology Flow

Where are we now?

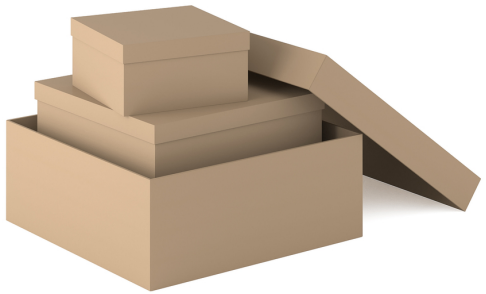


DATASTAX ACADEMY

Nest Data

Data nesting is the main data modeling technique

- Nesting organizes multiple entities into a single partition
- Supports partition per query data access
- Three data nesting mechanisms:
  - Clustering columns – multi-row partitions
  - Collection columns
  - User-defined type columns



Nest Data—Clustering Columns

Clustering column—primary data nesting mechanism

- Partition key identifies an entity that other entities will nest into
- Values in a clustering column identify the nested entities
- Multiple clustering columns implement multi-level nesting

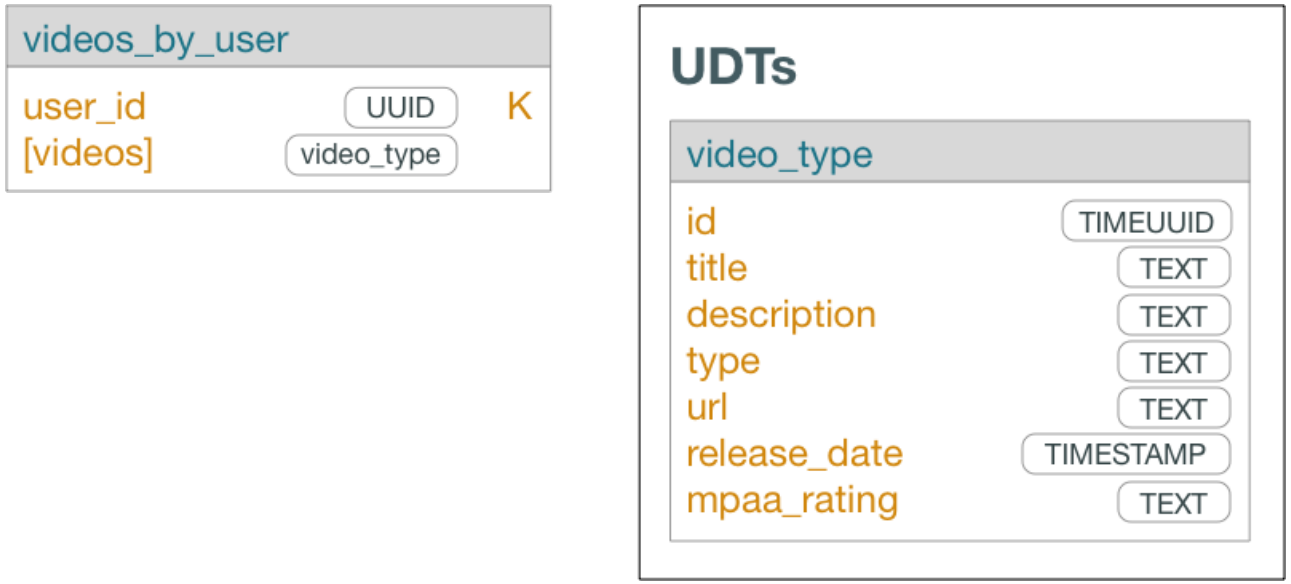
videos	
video_id	K
uploaded_timestamp	
user_id	
title	
description	
type	
{tags}	
<preview_thumbnails>	
{genres}	

actors_by_video	
video_id	K
actor_name	C↑
character_name	C↑

Nest Data—UDT

User-defined type—secondary data nesting mechanism

- Represents one-to-one relationship, but can use in conjunction with collections
- Easier than working with multiple collection columns



DATASTAX ACADEMY

Duplicate Data

Better to duplicate than to join

- Partition per query and data nesting may result in data duplication
- Query results are pre-computed and materialized
- Data can be duplicated across tables, partitions, and / or rows

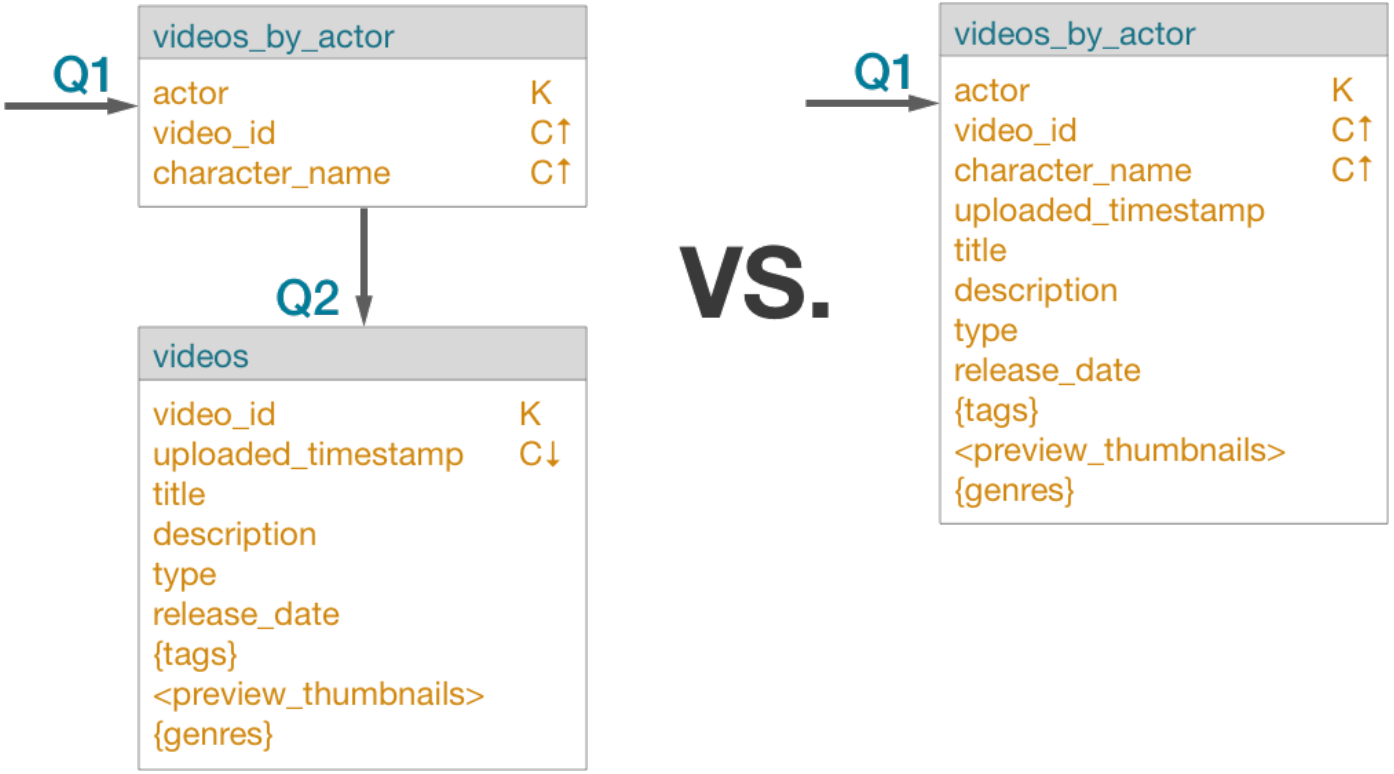
videos_by_actor	
actor	K
release_date	C↓
video_id	C↑
title	
type	
{tags}	
<preview_thumbnails>	

videos_by_genre	
genre	K
release_date	C↓
video_id	C↑
title	
type	
{tags}	
<preview_thumbnails>	

videos_by_tag	
tag	K
release_date	C↓
video_id	C↑
title	
type	
{tags}	
<preview_thumbnails>	

Duplicate Data, Cont.

Data duplication can scale, joins cannot



DATASTAX ACADEMY

Mapping Rules

For query-driven methodology

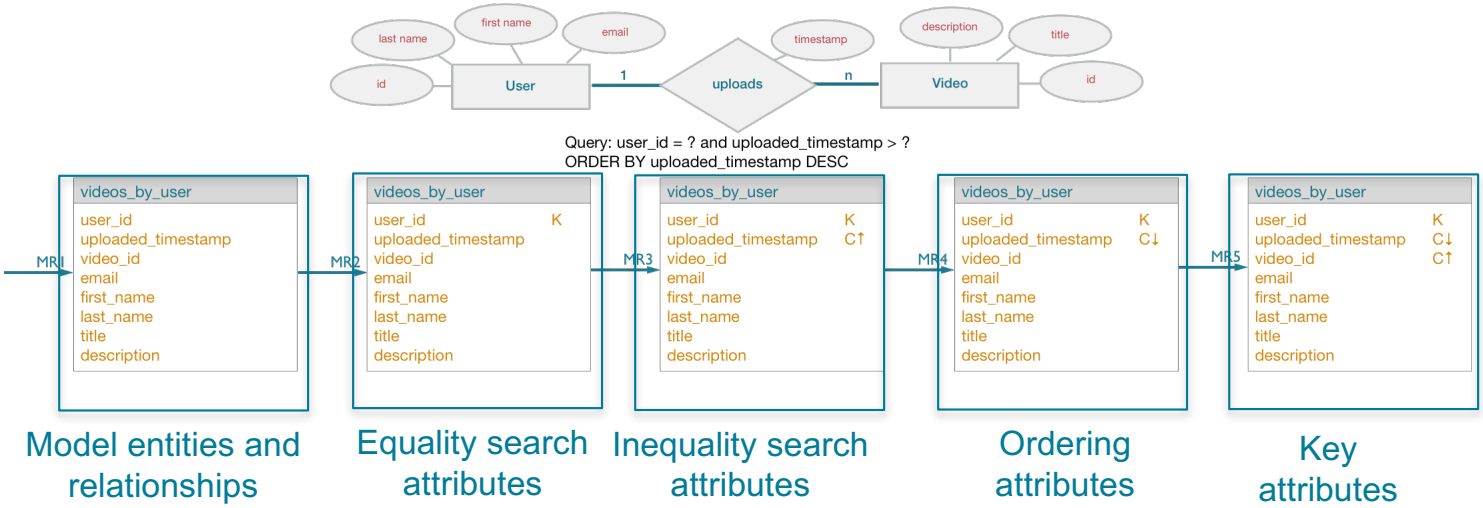
- Mapping rules ensure that a logical data model is correct
- Each query has a corresponding table
- Tables are designed to allow queries to execute properly
- Tables return data in the correct order

What Are The Rules?

- Mapping Rule 1: Entities and relationships
- Mapping Rule 2: Equality search attributes
- Mapping Rule 3: Inequality search attributes
- Mapping Rule 4: Ordering attributes
- Mapping Rule 5: Key attributes

Applying Mapping Rules

- Create a table schema from the conceptual data model and for each query
- Apply the mapping rules in order



DATASTAX ACADEMY

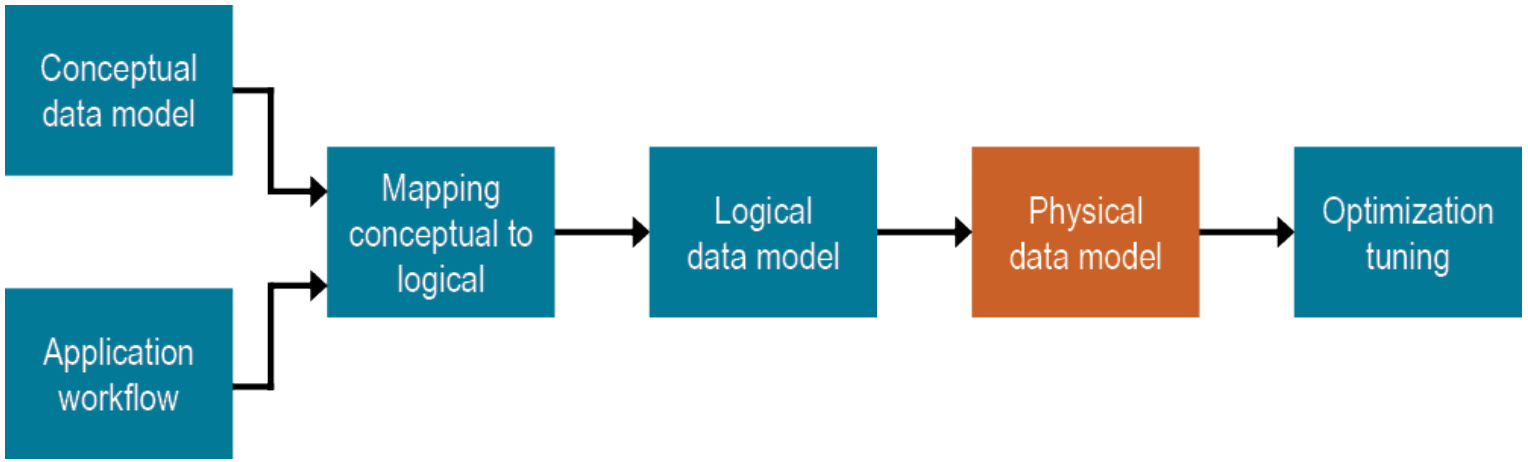
Exercise 4.3: Extend the KillrVideo Logical Model

DATASTAX ACADEMY

Physical Data Modeling

Data Modeling Methodology Flow

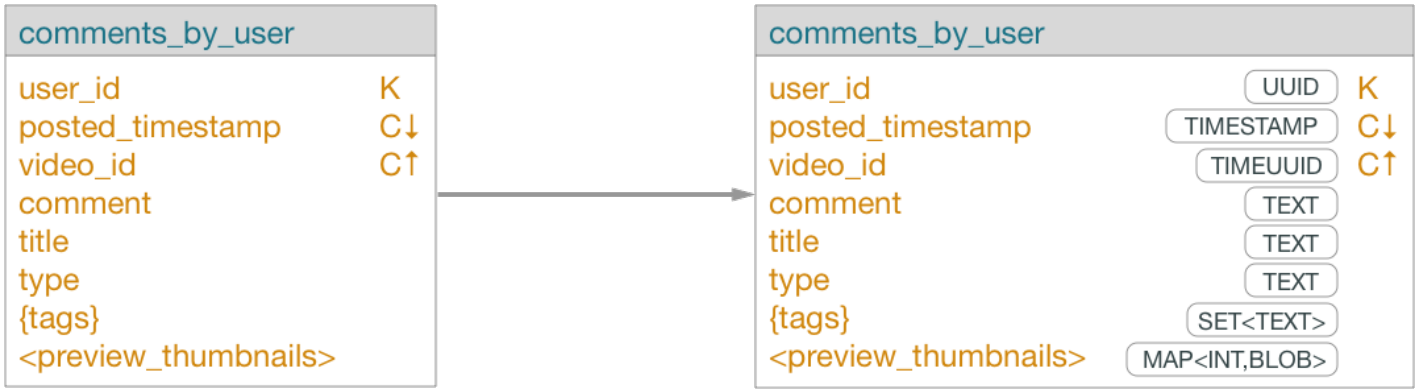
Where are we now?



DATASTAX ACADEMY

Creating the Physical Model

Adding data types and creating tables



Creating Apache Cassandra™ Tables

Writing the CQL statements

```
CREATE TABLE comments_by_user (
    user_id UUID,
     posted_timestamp TIMESTAMP,
     video_id TIMEUUID,
     comment TEXT,
     title TEXT,
     type TEXT,
     tags SET<TEXT>,
     preview_thumbnails MAP<INT, BLOB>,
     PRIMARY KEY ((user_id), posted_timestamp, video_id)
) WITH CLUSTERING ORDER BY (posted_timestamp DESC, video_id
ASC);
```

DATASTAX ACADEMY

Loading Data Methods

What methods are available for loading data?

- COPY command
- SSTable loader
- DSE Bulk Loader

CQL COPY Command

- COPY TO exports data from a table to a CSV file
- COPY FROM imports data to a table from a CSV file
- The process verifies the PRIMARY KEY and updates existing records
- If HEADER = false is specified the fields are imported in deterministic order
- When column names are specified, fields are imported in that order-- missing and empty fields set to null
- Source cannot have more fields than the target table--can have fewer fields

```
COPY table1 (column1, column2, column3) FROM 'table1data.csv'
WITH HEADER=true;
```

DATASTAX ACADEMY

SSTable Loader

Apache Cassandra™ Bulk Loader

- Bulk load external data into a cluster
- Load pre-existing SSTables into
  - an existing cluster or new cluster
  - a cluster with the same number of nodes or a different number of nodes
  - a cluster with a different replication strategy or partitioner
- Example:

```
sstableloader -d 110.82.155.1 /var/lib/cassandra/data/killrvideo/users/
```

DSE Bulk Loader

DataStax Enterprise Bulk Loader for large datasets

- Moves Cassandra data to/from files in the file system
- Uses both CSV or JSON formats
- Command-line interface
- Used for loading large amounts of data fast
- Example:

```
dsbulk load -url file1.csv -k ks1 -t table1
```

DATASTAX ACADEMY

Exercise 4.4: Finalizing the Physical Data Model