

NoSQL- Wide Column Store

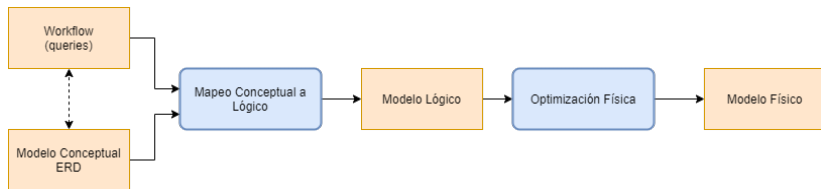
Ignacio Chiapella



1er Cuatrimestre 2022

Diseño

Cassandra - Notación/Método Chebotko



- Modelo Conceptual (DER) y Workflow (consultas)
- Modelo Lógico
 - Buscar el subconjunto del modelo conceptual que satisface cada consulta
 - Elegir claves
 - Usar diagramas Chebotko para describir el modelo lógico

SE CREA UNA TABLA POR
CONSULTA

Reglas de Mapeo

Basados en los DMP (Data Modeling Principles), las reglas de mapeo ayudan a realizar la transición desde el modelo conceptual al modelo lógico.

- 1 **MR1** (*Entities and Relationships*): Los tipos de entidades y relaciones mapean a tablas mientras que los datos se asignan a filas. Los atributos de las entidades y las relaciones se mapean a columnas

Reglas de Mapeo

Basados en los DMP (Data Modeling Principles), las reglas de mapeo ayudan a realizar la transición desde el modelo conceptual al modelo lógico.

- 1 **MR1** (*Entities and Relationships*): Los tipos de entidades y relaciones mapean a tablas mientras que los datos se asignan a filas. Los atributos de las entidades y las relaciones se mapean a columnas
- 2 **MR2** (*Equality Search Attributes*): Si se utilizan en una consulta por igualdad de atributos, entonces, éstos se mapean a columnas del prefijo de la clave primaria. Dichas columnas deben incluir todas las columnas de clave de partición y, opcionalmente, una o más columnas clustering key.

Reglas de Mapeo

Basados en los DMP (Data Modeling Principles), las reglas de mapeo ayudan a realizar la transición desde el modelo conceptual al modelo lógico.

- 1 **MR1** (*Entities and Relationships*): Los tipos de entidades y relaciones mapean a tablas mientras que los datos se asignan a filas. Los atributos de las entidades y las relaciones se mapean a columnas
- 2 **MR2** (*Equality Search Attributes*): Si se utilizan en una consulta por igualdad de atributos, entonces, éstos se mapean a columnas del prefijo de la clave primaria. Dichas columnas deben incluir todas las columnas de clave de partición y, opcionalmente, una o más columnas clustering key.
- 3 **MR3** (*Inequality Search Attributes*): Si se utilizan en consultas por desigualdad, estos atributos mapean como columnas clustering key. En la definición de clave principal, una columna que participa en la búsqueda de desigualdad debe ubicarse después de las columnas que participan en la búsqueda de igualdad.

Reglas de mapeo. Continuación

- ④ **MR4** (*Ordering Attributes*): Mapea a una columna clustering key con orden ascendente o descendente según se especifique en la consulta
- ⑤ **MR5** (*Key Attributes*): Mapea a columnas en la clave primaria. Una tabla que almacena datos de entidades o relaciones como filas debe incluir atributos claves que identifique estos datos unívocamente

Cassandra - Notación/Método Chebotko

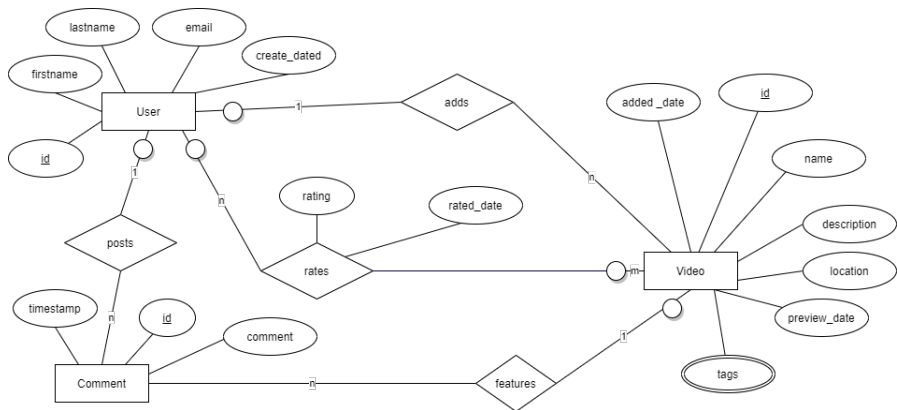
table_name		
column_name_1	CQL Type K	← Partition key column
column_name_2	CQL Type C↑	← Clustering key column (ASC)
column_name_3	CQL Type C↓	← Clustering key column (DESC)
column_name_4	CQL Type S	← Static column
column_name_5	CQL Type IDX	← Secondary index column
column_name_6	CQL Type ++	← Counter column
[column_name_7]	CQL Type	← Collection column (list)
{column_name_8}	CQL Type	← Collection column (set)
<column_name_9>	CQL Type	← Collection column (map)
column_name_10	UDT Name	← UDT column
(column_name_11)	CQL Type	← Tuple column
column_name_12	CQL Type	← Regular column

Ejemplo

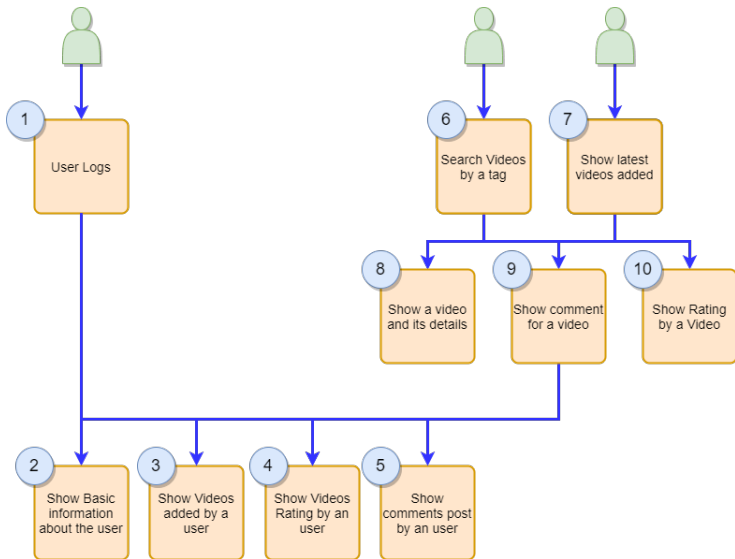
Aplicación de videos, los usuarios suben videos, califican los videos y comentan videos

Ejemplo

Aplicación de videos, los usuarios suben videos, califican los videos y comentan videos



Example Workflow



Identificar queries

Users



1

User logs into the site

Find a user by an email

2

Show basic information about a user

Find a user by an id

Comments



5

Show comments posted by a user

Find comments by a user (latest first)

9

Show comments for a video

Find comments by a video (latest first)

Videos



3

Show videos added by a user

Find videos by a user (latest first)

6

Search for videos by a tag

Find videos by a tag

7

Show latest videos added to the site

Find videos by a date added (latest first)

8

Show a video and its details

Find a video by an id

Ratings



4

Show video ratings by a user

Find ratings by a user (latest first)

10

Show ratings for a video

Find ratings by a user video (latest first)

Queries

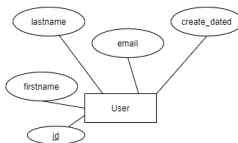
- 1 user_by_email
- 2 user_by_id
- 3 videos_by_user
- 4 ratings_by_user
- 5 comments_by_user
- 6 videos_by_tag
- 7 latest_videos
- 8 videos_by_id
- 9 comments_by_video
- 10 ratings_by_video

Tablas x Query

1-User by email / 2-User by Id

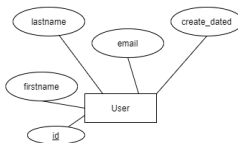
Tablas x Query

1-User by email / 2-User by Id



Tablas x Query

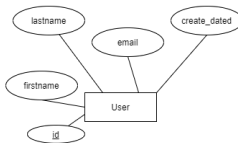
1-User by email / 2-User by Id



user_by_email
email
userid
firstname
lastname
created_date

Tablas x Query

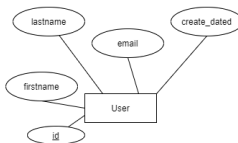
1-User by email / 2-User by Id



user_by_email	
email	K
userid	
firstname	
lastname	
created_date	

Tablas x Query

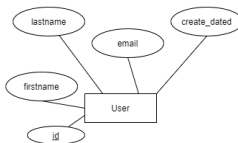
1-User by email / 2-User by Id



user_by_email	
email	K
userid	C↑
firstname	
lastname	
created_date	

Tablas x Query

1-User by email / 2-User by Id

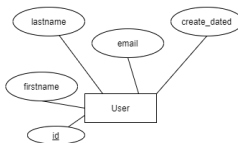


user_by_email	
email	K
userid	C↑
firstname	
lastname	
created_date	

user_by_id	
userid	K

Tablas x Query

1-User by email / 2-User by Id

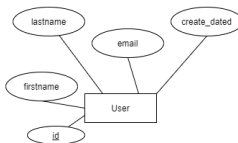


user_by_email	
email	K
userid	C↑
firstname	
lastname	
created_date	

user_by_id	
userid	K
firstname	
lastname	
created_date	
email	

Tablas x Query

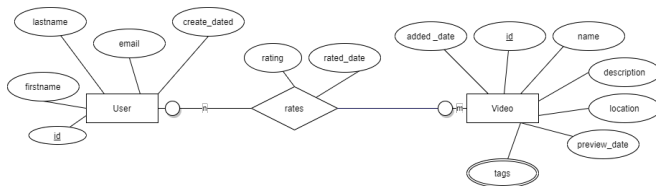
1-User by email / 2-User by Id



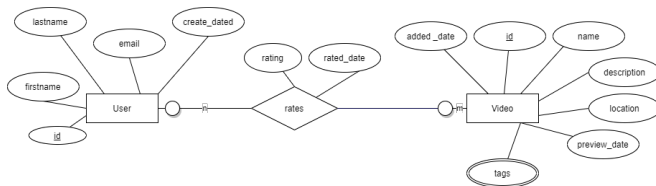
user_by_email	
email	K
userid	C↑
firstname	
lastname	
created_date	

user_by_id	
userid	K
firstname	S
lastname	S
created_date	S
email	S

10-Rating info con rated_date(Desc) by Video + Video info | MR:1y2



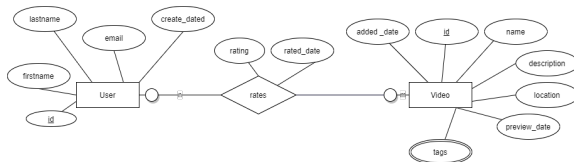
10-Rating info con rated_date(Desc) by Video + Video info | MR:1y2



ratings_by_video

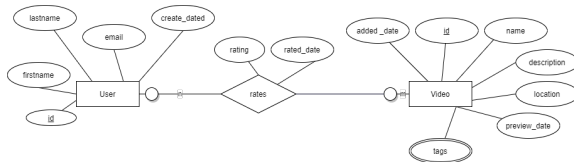
videoid	K
rated_date	
rating	
added_date	
name	
descripcion	
location	
preview_date	
{tags}	

10-Rating info con rated_date(Desc) by Video + Video info | MR:3y4



ratings_by_video	
videoid	K
rated_date	C↓
rating	
added_date	
name	
descripcion	
location	
preview_date	
{tags}	

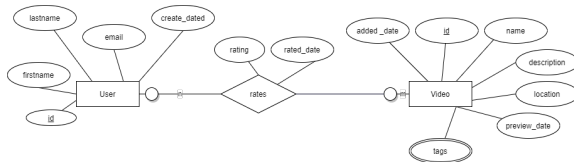
10-Rating info con rated_date(Desc) by Video + Video info | MR:5



ratings_by_video

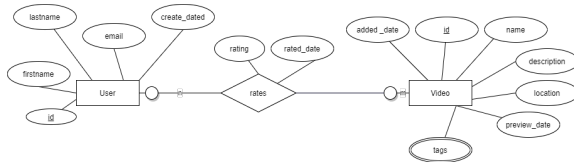
videoid	K
rated_date	C↓
userid	C↑
rating	
added_date	
name	
description	
location	
preview_date	
{tags}	

10-Rating info con rated_date(Desc) by Video + Video info | Statics?



ratings_by_video	
videoid	K
rated_date	C↓
userid	C↑

10-Rating info con rated_date(Desc) by Video + Video info | Statics?



ratings_by_video	
videoid	K
rated_date	C↓
userid	C↑
rating	
added_date	S
name	S
descripcion	S
location	S
preview_date	S
{tags}	S

Modelo Físico

ratings_by_video

videoid	K
rated_date	C↓
userid	C↑
rating	
added_date	S
name	S
descripcion	S
location	S
preview_date	S
{tags}	S



Modelo Físico

ratings_by_video	
videoid	K
rated_date	C↓
userid	C↑
rating	
added_date	S
name	S
descripcion	S
location	S
preview_date	S
{tags}	S



ratings_by_video		
videoid	uuid	K
rated_date	timestamp	C↓
userid	uuid	C↑
rating	int	
added_date	date	S
name	text	S
descripcion	text	S
location	text	S
preview_date	text	S
{tags}	set<text>	S

CQL

```
CREATE TABLE raitnig_by_video(  
    videoid uuid,  
    rated_date,  
    userid uuid,  
    rating int ,  
    added date,  
    name text,  
    descripcion text,  
    location text,  
    preview_date text,  
    tags set<text> ,  
    PRIMARY KEY((videoid), rated_date, userid)  
) WHIT CLUSTERING ORDER BY (rated_date DESC, userid ASC)
```



Bibliografía

- Artem Chebotko, Andrey Kashlev, and Shiyong Lu. 2015. *A Big Data Modeling Methodology for Apache Cassandra*. In Proceedings of the 2015 IEEE International Congress on Big Data.
- Jeff Carpenter and Eben Hewitt. 2020. *Cassandra: The Definitive Guide* (3rd. ed.). O'Reilly Media, Inc.