

NoSQL- Column Family

Lic. Gerardo Rossel



DEPARTAMENTO
DE COMPUTACION

2016

Familia de Columnas

BigTable

BigTable - Google

Fay Chang, et al. **“BigTable: A Distributed Storage System for Structured Data.”** OSDI'06: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, Nov., 2006.
<http://research.google.com/archive/bigtable.html>

BigTable - Definición

A Bigtable is a sparse, distributed, persistent multidimensional sorted map. The map is indexed by a row key, column key, and a timestamp; each value in the map is an uninterpreted array of bytes.

BigTable - Derivados

Hbase y Cassandra

Modelos de Datos Column Family

- HBase - BigTable.
- Cassandra.

ATENCION

Hay algunas diferencias de terminología y modelo general entre el modelo de HBase y el de Cassandra.

HBase

Conceptos

- **Namespace** es un agrupamiento lógico de tablas, similar a una base de datos en un SBDR.
- **Tablas**: Una tabla consiste de múltiples filas.
- **Row**: Una fila (row) se organiza como un conjunto de ***familias de columnas***. Las familias de columnas consisten de columnas relacionadas.
- **Cell**: Una combinación *row key*, familia de columnas y calificador de columna único identifica una celda (*cell*). La celda almacena datos que se denominan valores.
- **Timestamp**: Al escribir un nuevo valor el viejo no es sobrescrito sino que se agrega el nuevo junto a un *timestamp*. El *timestamp* permite a las aplicaciones determinar la última versión de un valor en una columna

Conceptos

Densidad

Densidad baja. En una fila puede haber cualquier número de columnas en cada familia incluso ninguna.

Hash-Map

Básicamente una base column-family es un Mapa Multi-Dimensional

Modelo de datos

BigTable - HBase

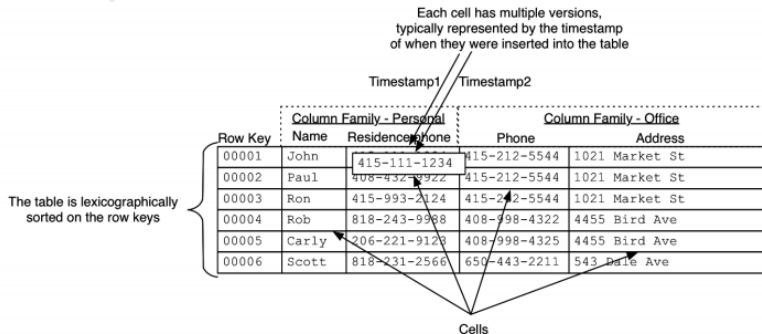


Figura de Amandeep Khurana: Introduction to HBase Schema Design

Mapa Multi-Dimensional

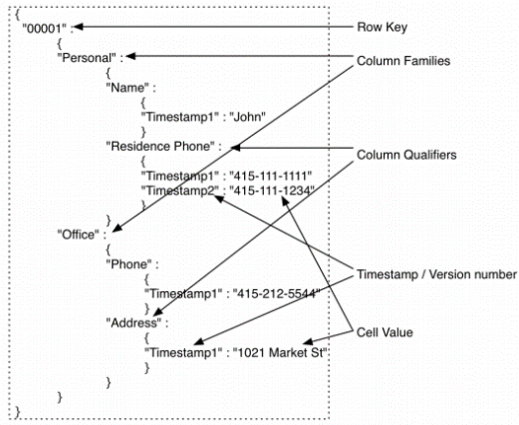


Figura de Amandeep Khurana: Introduction to HBase Schema

Mapa Multi-Dimensional

```
00001 → { Personal : { Name : { Timestamp1 : John }, Residence Phone : { Timestamp1 : 415-111-1234 } },
          { Office : { Phone : { Timestamp1 : 415-212-5544 }, Address : { Timestamp1 : 1021 Market St
```

00001 , Personal → { Name : { Timestamp1 : John }, Residence Phone : { Timestamp1 : 415-111-1234 } }

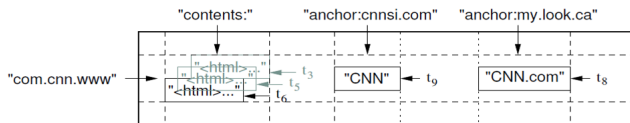
00001, Personal:Residence Phone $\rightarrow \{ \{ \text{Timestamp1} : 415-111-1111 \} , \{ \text{Timestamp2} : 415-111-1234 \} \}$

00001, Personal:Residence Phone , Timestamp1 → { 415-111-1111 }

00001, Personal:Residence Phone , Timestamp2 → { 415-111-1234 }

Figura de Amandeep Khurana:Introduction to HBase Schema

Ejemplo Webtable



Row Key	Time Stamp	ColumnFamily contents	ColumnFamily anchor	ColumnFamily people
"com.cnn.www"	t9		anchor:cnnsi.com = "CNN"	
"com.cnn.www"	t8		anchor.my.look.ca = "CNN.com"	
"com.cnn.www"	t6	contents.html = "<html>..."		
"com.cnn.www"	t5	contents.html = "<html>..."		
"com.cnn.www"	t3	contents.html = "<html>..."		

Relacional vs. Column Family

① Raw Data

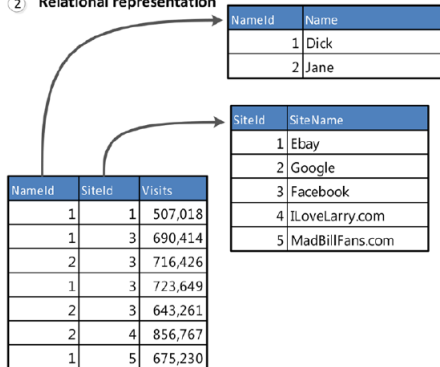
Name	Site	Visits
Dick	Ebay	507,018
Dick	Google	690,414
Jane	Google	716,426
Dick	Facebook	723,649
Jane	Facebook	643,261
Jane	ILoveLarry.com	856,767
Dick	MadBillFans.com	675,230

Relational vs. Column Family

① Raw Data

Name	Site	Visits
Dick	Ebay	507,018
Dick	Google	690,414
Jane	Google	716,426
Dick	Facebook	723,649
Jane	Facebook	643,261
Jane	ILoveLarry.com	856,767
Dick	MadBillFans.com	675,230

② Relational representation

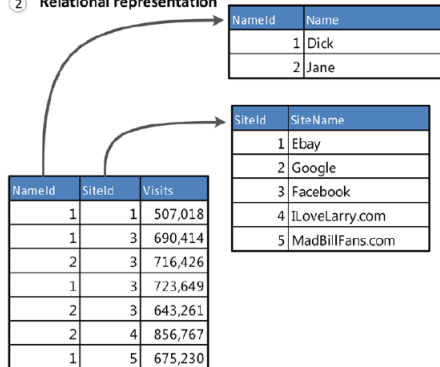


Relational vs. Column Family

① Raw Data

Name	Site	Visits
Dick	Ebay	507,018
Dick	Google	690,414
Jane	Google	716,426
Dick	Facebook	723,649
Jane	Facebook	643,261
Jane	ILoveLarry.com	856,767
Dick	MadBillFans.com	675,230

② Relational representation



③ HBase version

Id	Name	Ebay	Google	Facebook	(other columns)	MadBillFans.com
1	Dick	507,018	690,414	723,649	675,230

Id	Name	Google	Facebook	(other columns)	ILoveLarry.com
2	Jane	716,426	643,261	856,767

Cassandra

Cassandra

- **Keyspace:** Es el contenedor exterior de datos en Cassandra.
 - Factor de Replicación
 - Estrategia de replica
- **Column Families** -Un *Keyspace* es un contenedor para una lista de una o más familia de columnas. Una familia de columnas es una colección de filas. Cada fila contiene columnas ordenadas. Las familias de columnas representan la estructura de los datos. Cada keyspace tiene al menos una (y en general muchas) familia de columna.

Modelo de datos

Cassandra

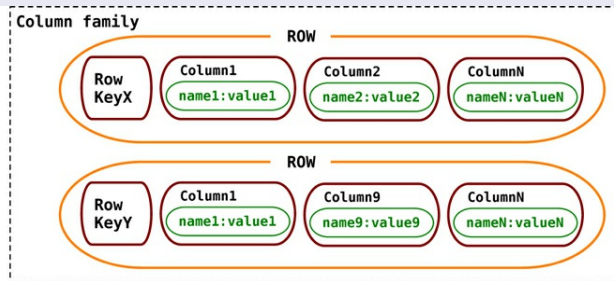


Figura de Pramod J. Sadalage Martin Fowler: NoSQL Distilled

Diseño

Diseño Aspectos Generales

Cómo diseñar

Establecer los patrones de escritura y lectura.

- Desnormalizar!
- Usar columnas sin valor
- Usar nombres y valores de columna para almacenar datos
- Modelar una entidad como una fila simple
- Mantener un número adecuado de versiones en los valores de las columnas.
 - HBase permite especificar el mínimo y máximo número de versiones.
- Evitar estructuras complejas en los valores de las columnas.

Cassandra - Notación/Método Chebotko

- Modelo Conceptual: DER
- Modelo Lógico
 - Identificar Patrones de acceso
 - Buscar el subconjunto del modelo conceptual que satisface la consulta
 - Elegir claves
 - Usar diagramas Chebotko para describir el modelo lógico

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.

- ❶ **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
- ❷ **MR2** (*Equality Search Attributes*): Si se utilizan en una consulta por igualdad de atributos, entonces, éstos forman la clave primaria y tales columnas se incluirán en las particiones por clave

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 forman la clave primaria y tales columnas se incluirán en las particiones por clave
- 3 **MR3** (*Inequality Search Attributes*): Si se utilizan en consultas por desigualdad, estos atributos mapean como columnas dinámicas, conteniendo los valores

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 forman la clave primaria y tales columnas se incluirán en las particiones por clave
- 3 **MR3** (*Inequality Search Attributes*): Si se utilizan en consultas por desigualdad, estos atributos mapean como columnas dinámicas, conteniendo los valores
- 4 **MR4** (*Ordering Attributes*): Mapea a columna dinámica con orden ascendente o descendente según se especifique en la 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.

- ❶ **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
- ❷ **MR2** (*Equality Search Attributes*): Si se utilizan en una consulta por igualdad de atributos, entonces, éstos forman la clave primaria y tales columnas se incluirán en las particiones por clave
- ❸ **MR3** (*Inequality Search Attributes*): Si se utilizan en consultas por desigualdad, estos atributos mapean como columnas dinámicas, conteniendo los valores
- ❹ **MR4** (*Ordering Attributes*): Mapea a columna dinámica con orden ascendente o descendente según se especifique en la consulta
- ❺ **MR5** (*Key Attributes*): Mapea a clave primaria. Una tabla que almacena datos de entidades o relaciones como filas debe incluir atributos claves que identifique estos datos univocamente

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: DER

Supongamos el siguiente DER:

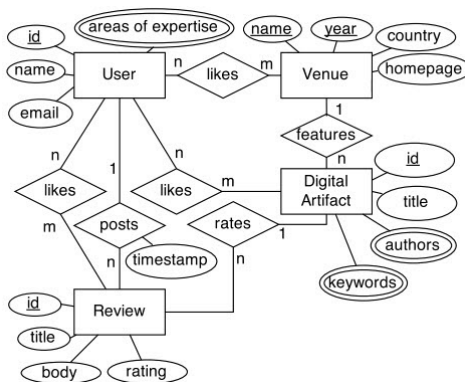


Diagrama de Entidad-Relación Completo

Ejemplo: DER

Tomamos el siguiente trozo:

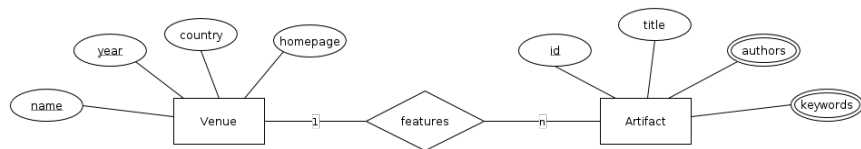


Diagrama de Entidad-Relación

Ejemplo: DER

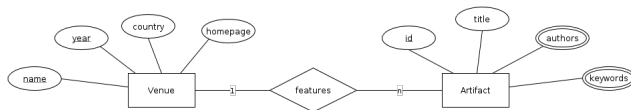


Diagrama de Entidad-Relación

- Q1: Obtener los artefactos publicados en *venue* con un nombre dado después de un año dado. El resultado debe estar ordenado en forma descendente

