NoSQL- Wide Column Store

Dr. Gerardo Rossel

1er Cuatrimestre 2024

Familia de Columnas o Wide Column

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

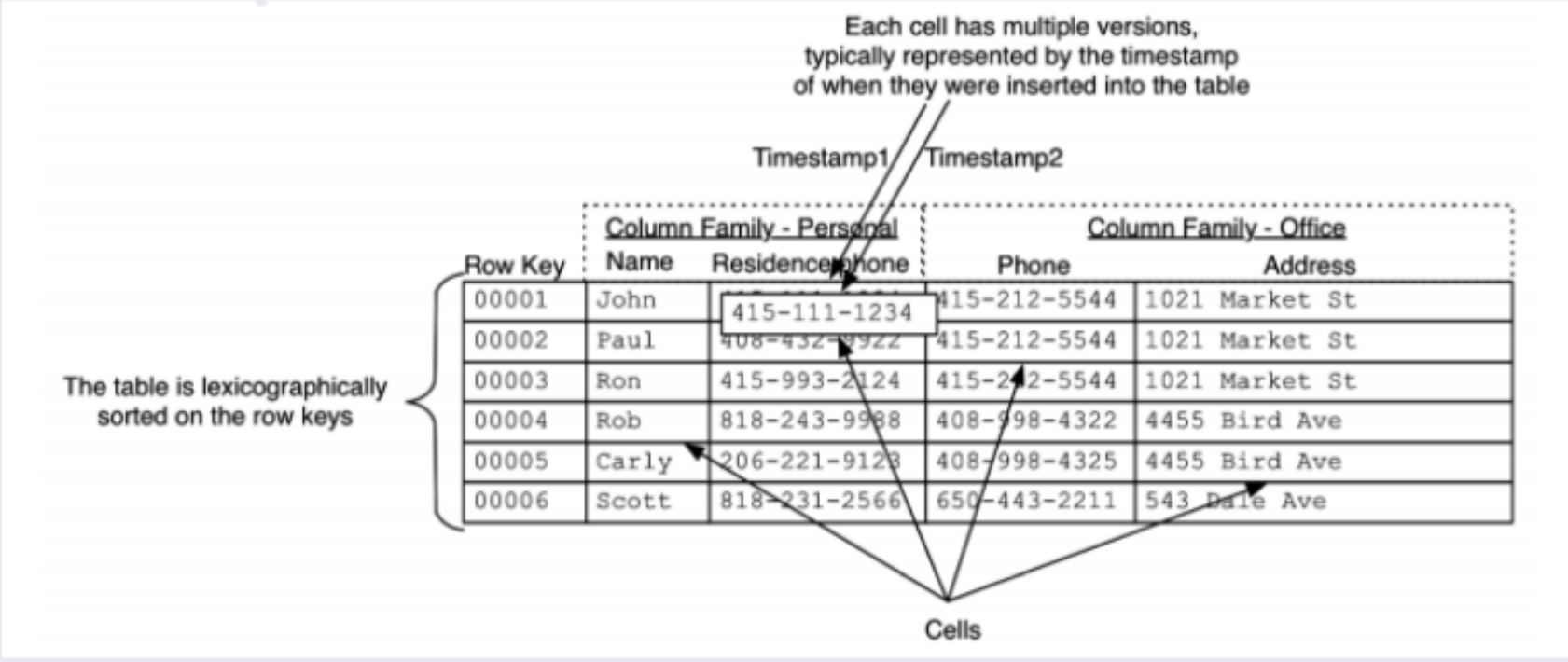


Figura de Amandeep Khurana:Introduction to HBase Schema Design

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

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

2 Relational representation

NameId	Name
1	Dick
2	Jane

SiteId	SiteName
1	Ebay
2	Google
3	Facebook
4	ILoveLarry.com
5	MadBillFans.com

NameId	SiteId	Visits
1	1	507,018
1	3	690,414
2	3	716,426
1	3	723,649
2	3	643,261
2	4	856,767
1	5	675,230

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

② **Relational representation**

NameId	Name
1	Dick
2	Jane

SiteId	SiteName
1	Ebay
2	Google
3	Facebook
4	ILoveLarry.com
5	MadBillFans.com

NameId	SiteId	Visits
1	1	507,018
1	3	690,414
2	3	716,426
1	3	723,649
2	3	643,261
2	4	856,767
1	5	675,230

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 Definición

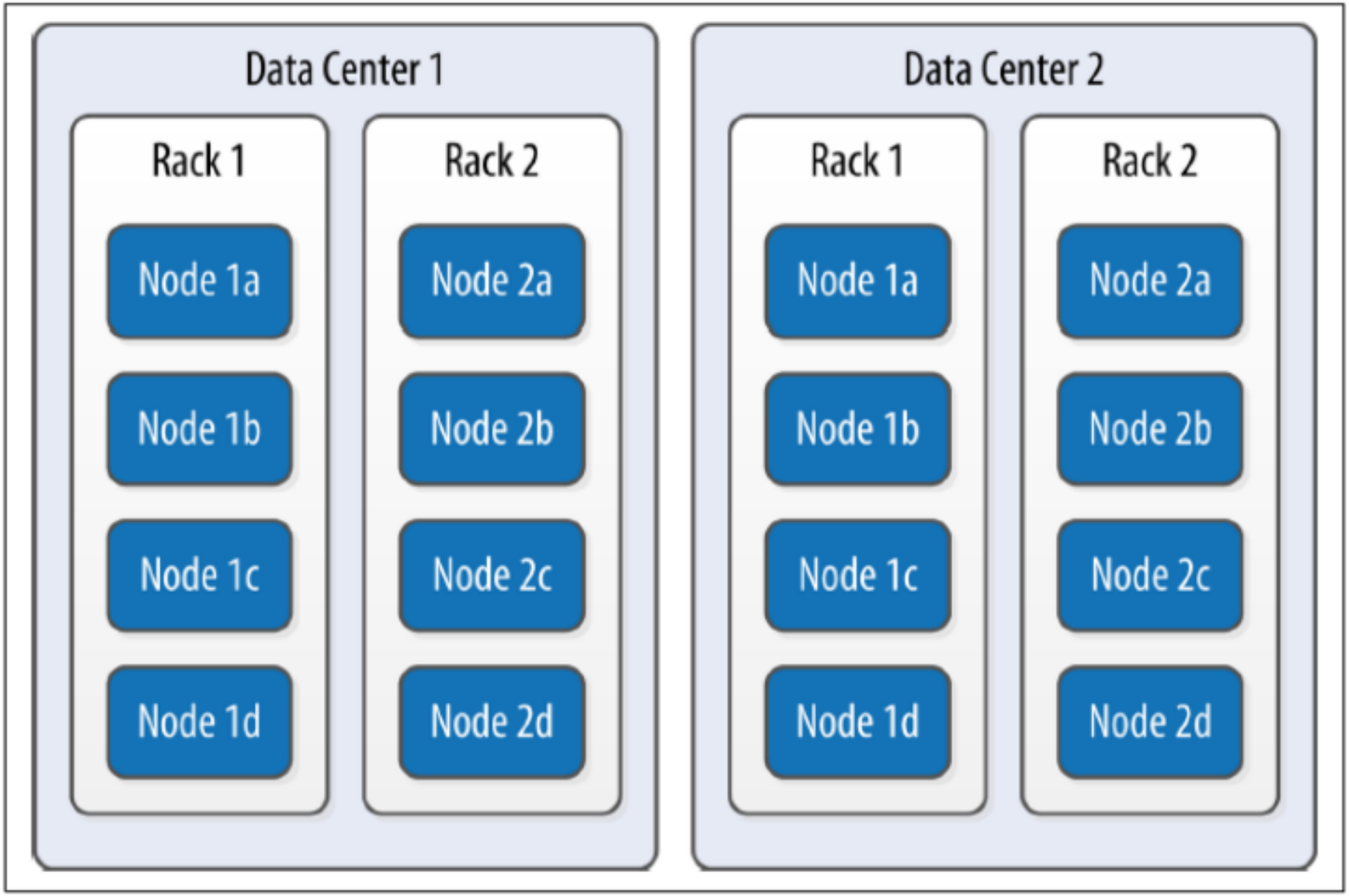
Cassandra en 50 palabras

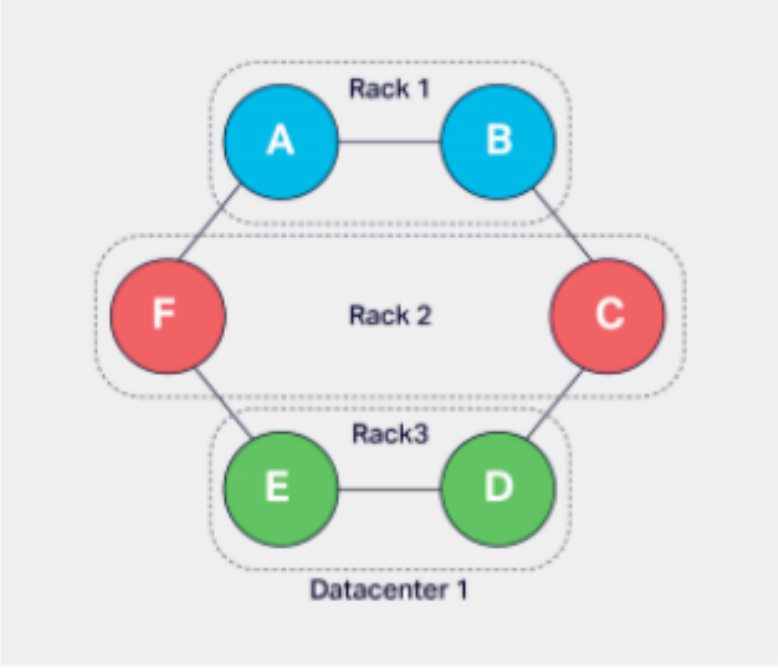
"Apache Cassandra is an open source, distributed, decentralized, elastically scalable, highly available, fault-tolerant, tuneably consistent, row-oriented database. Cassandra bases its distribution design on Amazon's Dynamo and its data model on Google's Bigtable, with a query language similar to SQL. Created at Facebook, it now powers cloud-scale applications across many industries."

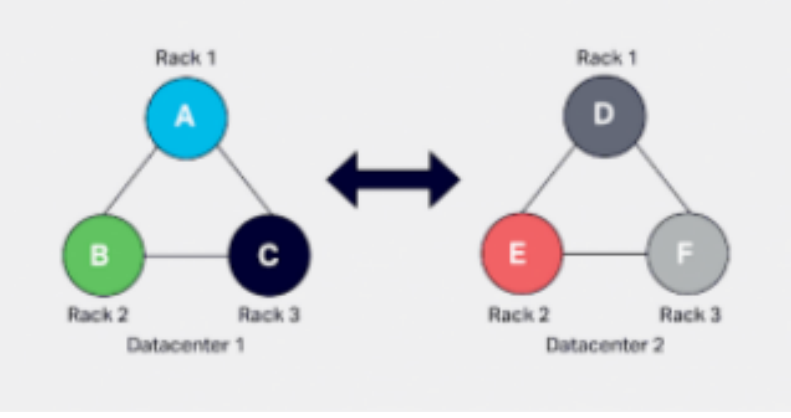
Componentes

- **Node** Un servidor que ejecuta una instancia de Cassandra Un nodo puede ser un host físico, una instancia en la nube o incluso un contenedor Docker
- **Rack** Un conjunto lógico de nodos muy cerca uno de otro, quizás en máquinas físicas en un solo estante de equipo conectadas al mismo switch. En un *deployment* en la nube se refiere a una colección de instancias de maquinas corriendo en la misma zona de disponibilidad.
- **Data center** Un conjunto lógico de Racks quizás ubicado en el mismo edificio y conectado por red confiable. En la nube se mapea generalmente a una región de la misma.
- **Cluster**: el conjunto de data centers y que mapea a un anillo lógico

Organización física



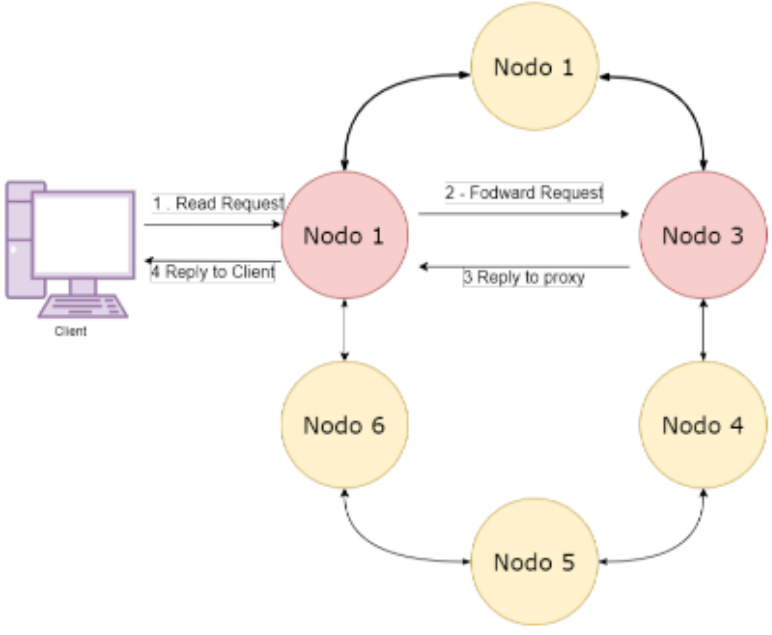

Rack 1
A
B
F
Rack 2
C
Rack3
E
D
Datacenter 1


Rack 1
A
B
C
Rack 2
Rack 3
Datacenter 1
Rack 1
D
E
F
Rack 2
Rack 3
Datacenter 2

Arquitectura Peer-to-Peer

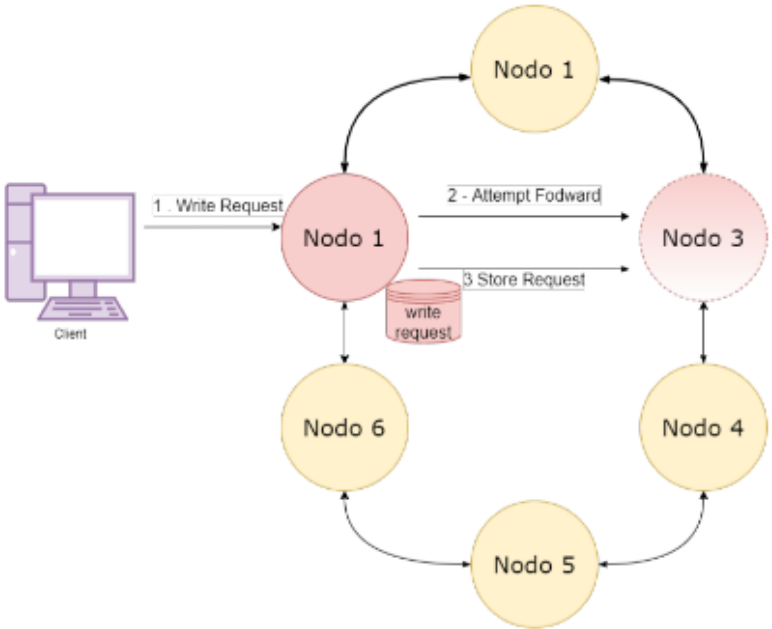
- Gossip: Corre una vez por segundo y se conecta con hasta 3 nodos al azar
  - Los nodos intercambian información entre sí sobre otros nodos sobre los que cada uno ha “chismoseado” previamente para que todos los nodos aprendan rápidamente el estado general del clúster.
- Snitch: Provee información acerca de la topología de la red. Sirve para determinar en que nodos escribir y de que nodos leer.

Lectura en Cassandra



El nodo coordinador actúa como un proxy entre la solicitud de un cliente y los nodos del clúster que realmente tienen los datos solicitados almacenados en ellos.
La cantidad de solicitudes que se envían a los otros nodos depende del factor de replicación. Cuanto mayor sea el factor de replicación, más nodos consultará el nodo coordinador. Pero, para responder a la solicitud del cliente, el nodo coordinador no siempre tiene que esperar a todos. El cliente puede estar satisfecho con la primera y más rápida respuesta de cualquiera de los nodos. Esto depende del nivel de coherencia que el cliente desee para las lecturas o escrituras.

Escritura en Cassandra



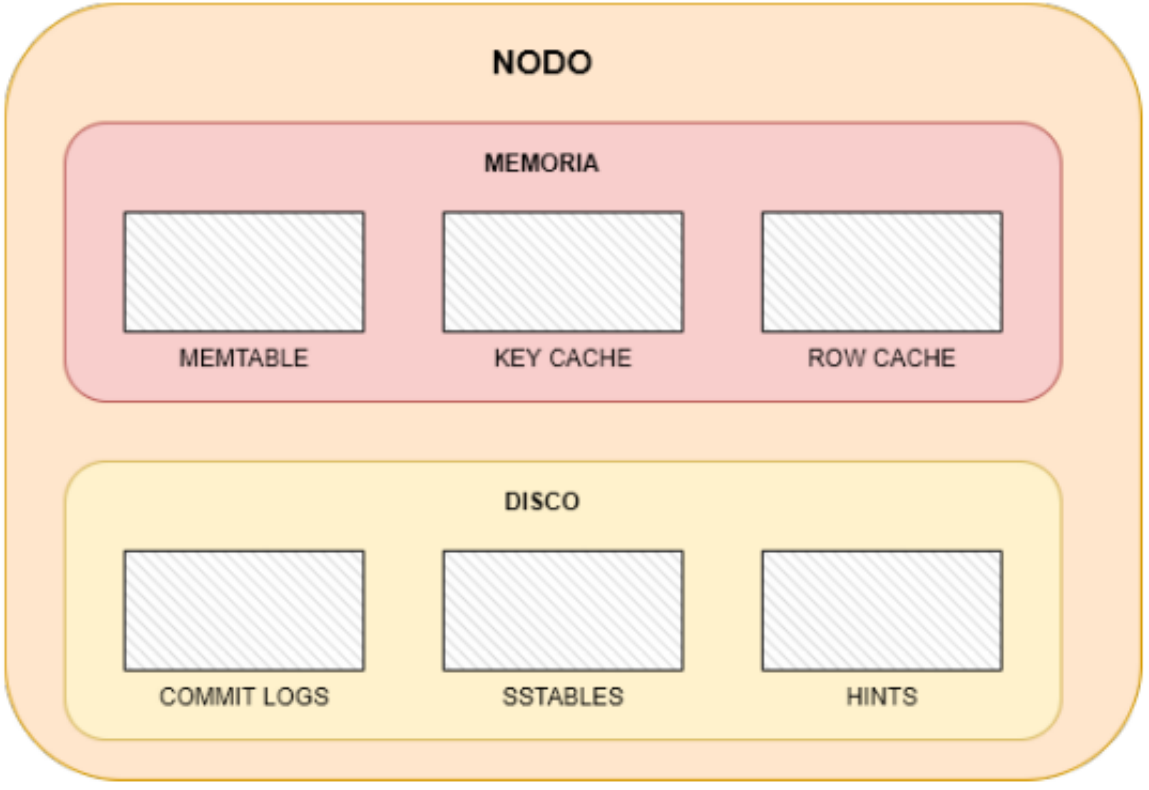
Al escribir, el coordinador a veces no podrá comunicarse con el nodo responsable del registro; por ejemplo, cuando el nodo está inactivo o hay algún tipo de problema de red.

El nodo coordinador puede mantener este registro almacenado y esperar a que los nodos vuelvan a funcionar.

Esta técnica se llama transferencia indirecta. Este registro no será visible para ninguna consulta hasta que no esté escrito en los nodos responsables del mismo.

El nodo que contiene el *hint* nunca responde a las consultas con esos datos.El *hint* tiene toda la información necesaria para transferir los datos a los nodos cuando esten disponibles

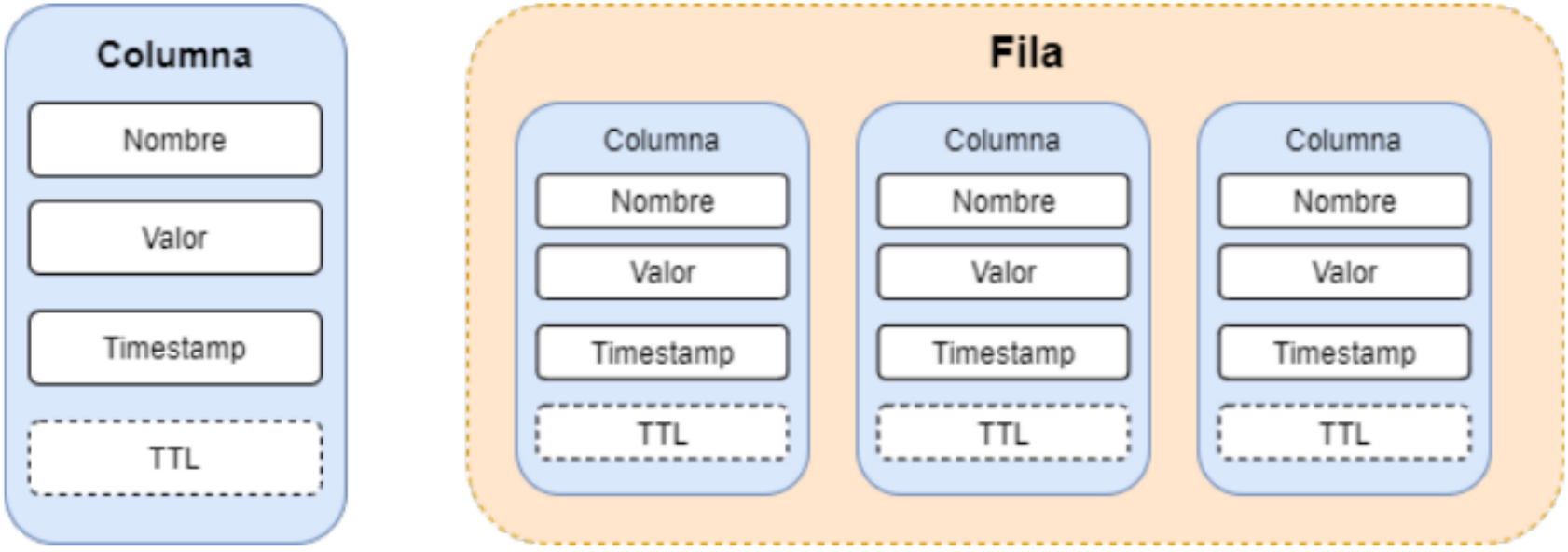
Nodo Cassandra



Modelo de Datos de Cassandra

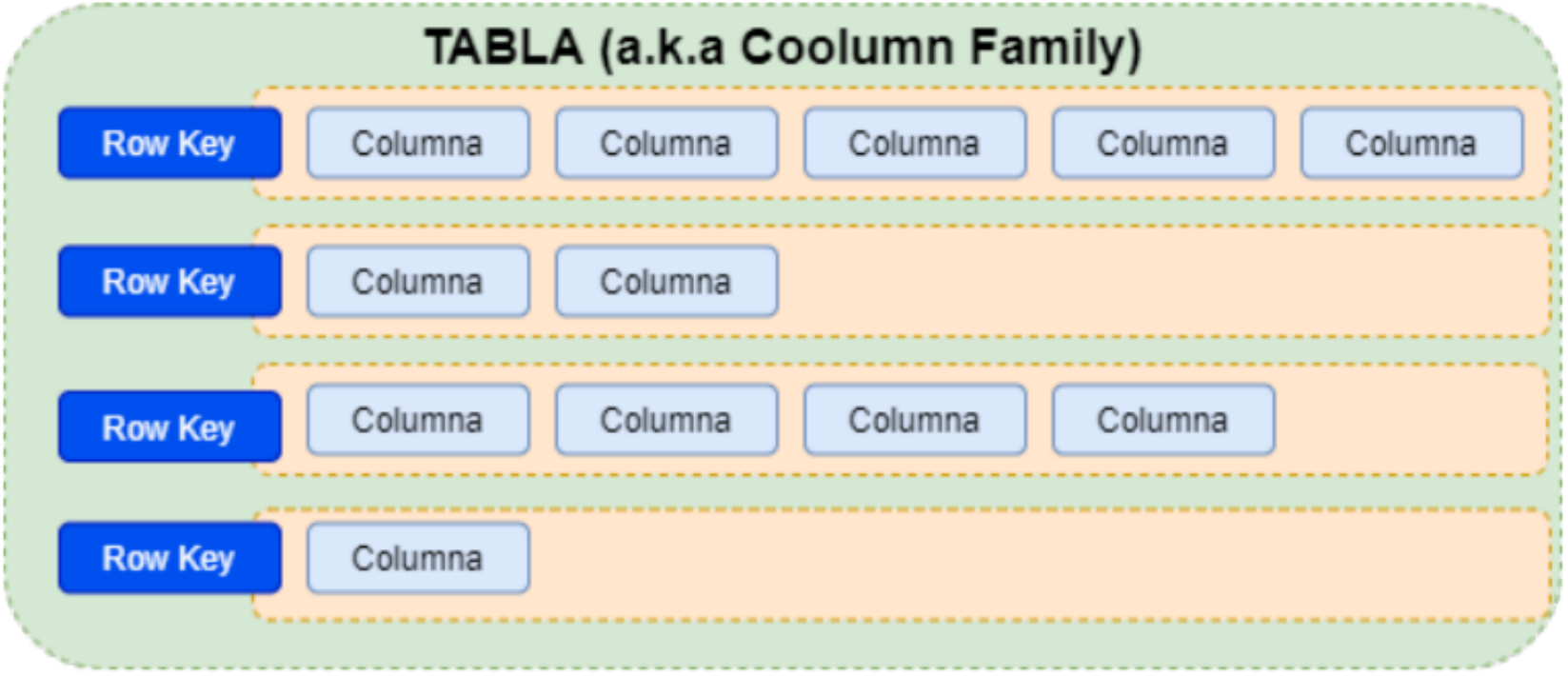
- Columnas: par clave/valor
- Row (fila): contenedor para columnas referenciadas por una clave primaria PK.
- Partición: grupo de filas relacionadas almacendas juntas en el mismo nodo.
- Tabla: que es un contenedor para filas organizadas por particiones.
- The keyspace, es un contenedor para tablas.

Modelo de Datos de Cassandra



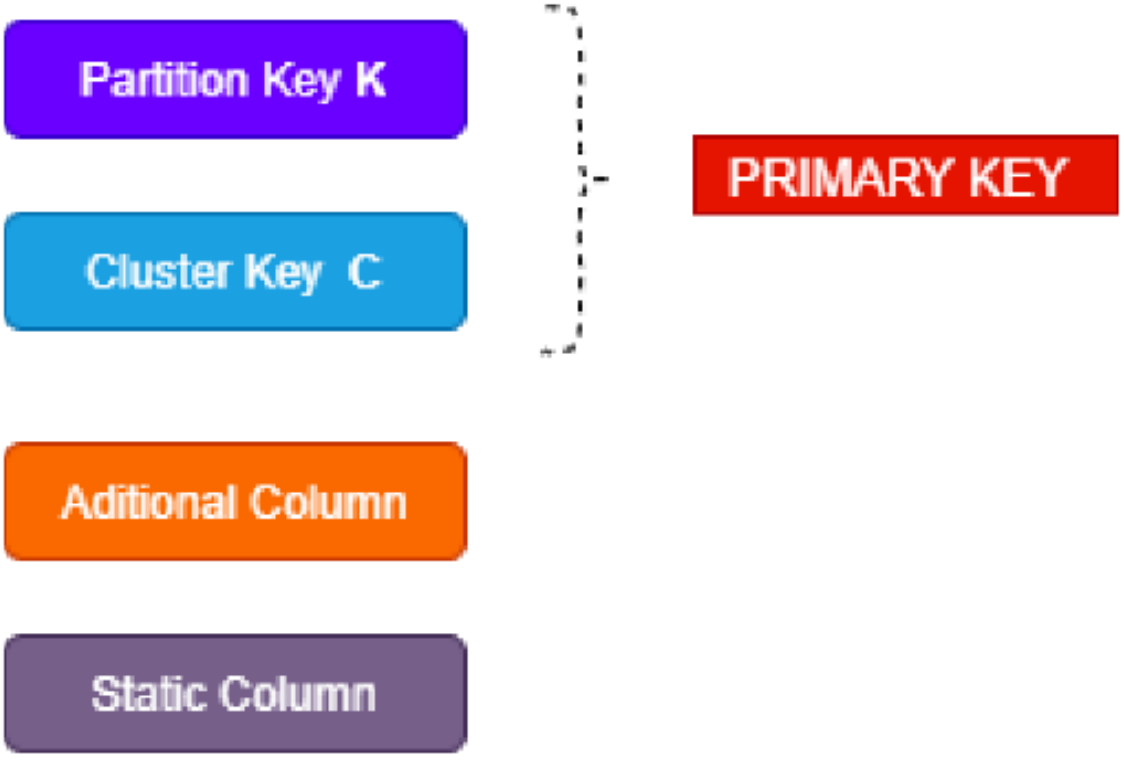
Filas y Columnas Cassandra

Modelo de Datos de Cassandra

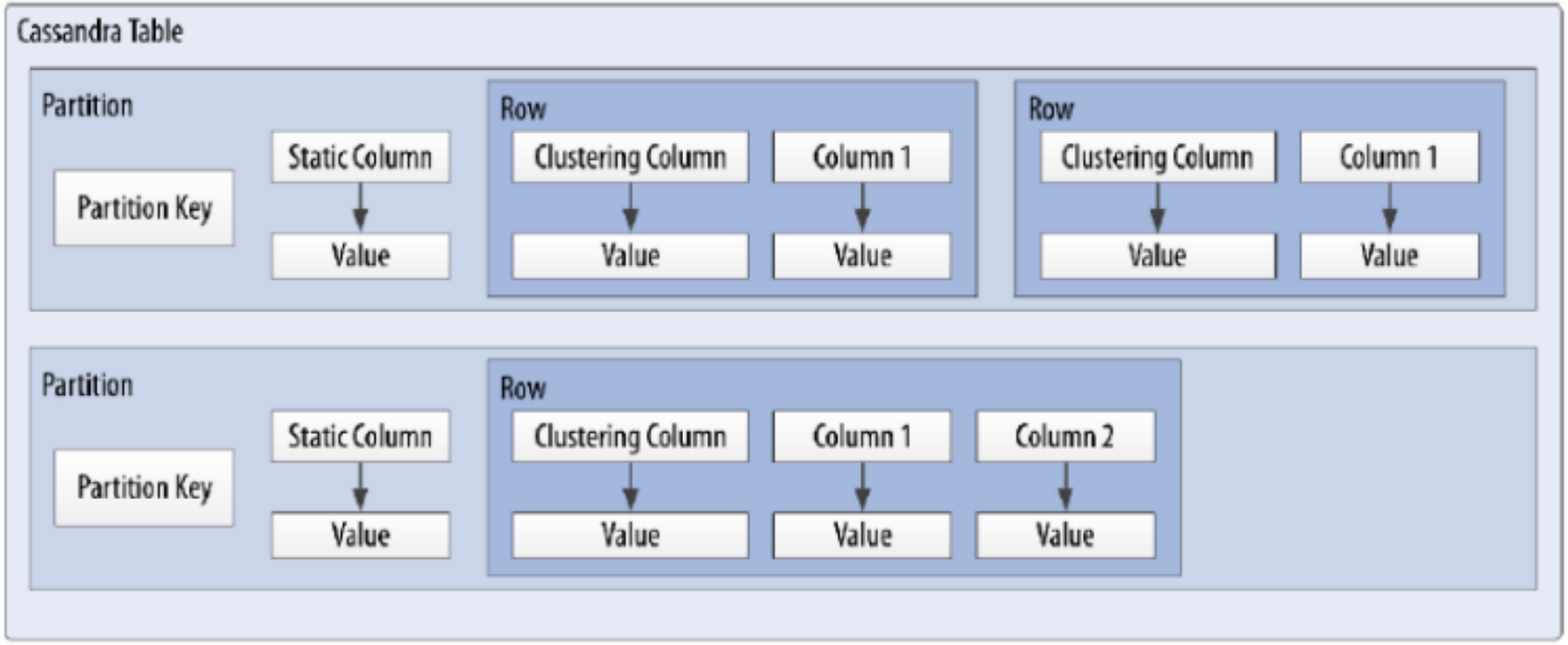


TABLAS Cassandra

Tipos de Columna



Tablas con Partición



Tomado de Cassandra: The Definitive Guide 2020

Ejemplo

Usuarios y Eventos

Una aplicación que rastrea usuarios y eventos. Se debería recordar cada vez que un cliente hace alguna interacción con el sistema: logins, compras, visitas a páginas, etc.

Ejemplo

Usuarios y Eventos

Una aplicación que rastrea usuarios y eventos. Se debería recordar cada vez que un cliente hace alguna interacción con el sistema: logins, compras, visitas a páginas, etc.

Solución relacional: dos tablas Usuarios y Eventos.

¿Que pasaría si hay 10.000.000 de usuarios y cada usuario tiene un promedio de 1000 eventos de 1Kb cada evento?

Ejemplo

Usuarios y Eventos

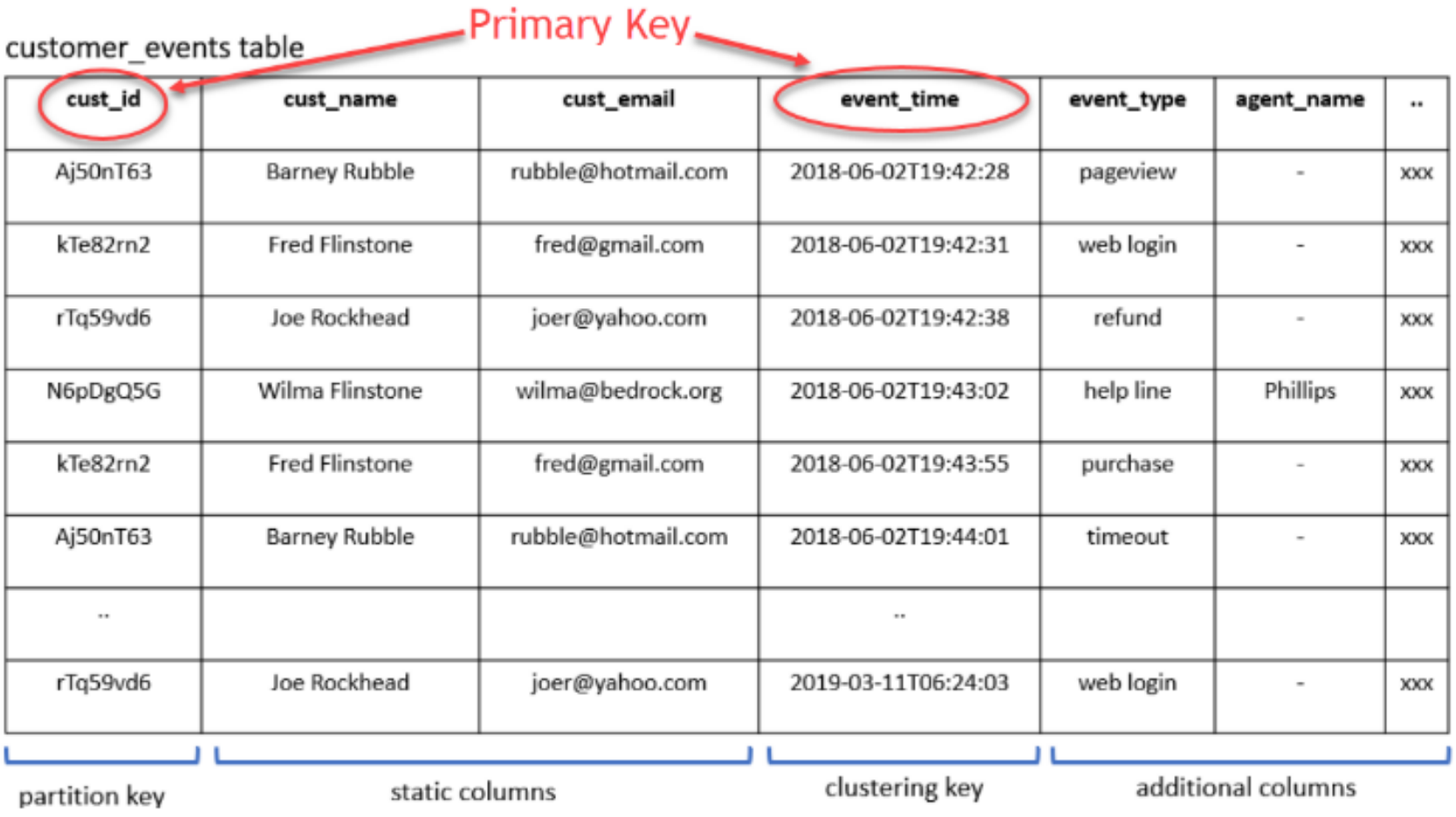
Una aplicación que rastrea usuarios y eventos. Se debería recordar cada vez que un cliente hace alguna interacción con el sistema: logins, compras, visitas a páginas, etc.

Solución relacional: dos tablas Usuarios y Eventos.

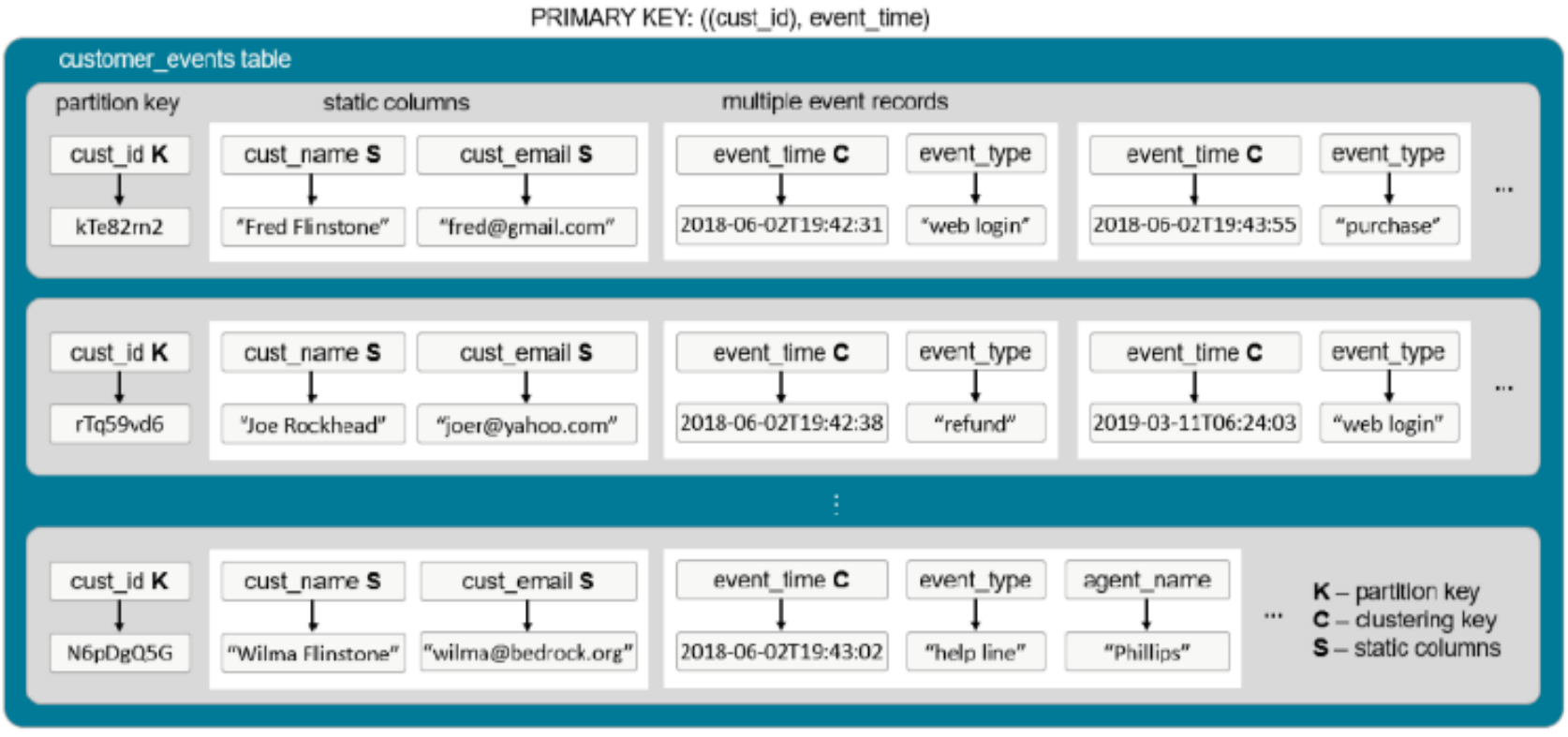
¿Que pasaría si hay 10.000.000 de usuarios y cada usuario tiene un promedio de 1000 eventos de 1Kb cada evento?

10TB de datos en la tabla de eventos.

Tabla en Cassandra



Almacenamiento de la Tabla en Cassandra



Un poco de CQL

Tipos de Datos CQL

- Numéricos: bigint,smallint,tinyint, varint, float, double, decimal
- Texto: text, varchar (UTF-8 character string), ascii
- Tiempo: timestamp,date, time
- Identificadores:uuid, timeuuid
- Otros: boolean, blob, inet, **counter**
- Colecciones: set, list y map.
- Tuple. Ejemplo: address tuple<text, text, text, int>;
- User-Defined Types.

Crear una Tabla

```
CREATE TABLE IF NOT EXISTS rank_by_year_and_name (
race_year int,
race_name text,
cyclist_name text,
rank int,
PRIMARY KEY ((race_year, race_name), rank));

INSERT INTO cycling.rank_by_year_and_name
(race_year, race_name, cyclist_name, rank)
VALUES (2015,
'Tour of Japan - Stage 4 - Minami > Shinshu',
'Benjamin PRADES', 1);
```

Restricción Clave de Partición

Operadores - Clave de Partición

Las columnas de la clave de partición solo admiten dos operadores: $=$ e *IN*

Cassandra requerirá que restrinja todas las columnas de clave de partición o ninguna de ellas a menos que la consulta pueda usar un índice secundario.

Where en Clave de Partición

```
SELECT rank, cyclist_name as name
FROM cycling.rank_by_year_and_name
WHERE race_name = 'Tour of Japan - Stage 4 - Minami > Shinshu'
AND race_year = 2015
AND rank <= 2;
```

Where - Cluster Columns

```
CREATE TABLE numberOfRequests
(cluster text,
date text,
datacenter text,
hour int,
minute int,
numberOfRequests  int,
PRIMARY KEY ((cluster, date), datacenter, hour, minute))
```

Where - Cluster Columns

CORRECTO:

```
SELECT * FROM numberOfRequests
WHERE cluster = 'cluster1' AND date = '2015-06-05'
AND datacenter = 'US_WEST_COAST' AND hour = 14 AND minute = 00;
```

INCORRECTO:

```
SELECT * FROM numberOfRequests
WHERE cluster = 'cluster1' AND date = '2015-06-05'
AND hour = 14 AND minute = 00;
```

Where - Cluster Columns

Correctas:

```
SELECT * FROM numberOfRequests WHERE cluster = 'cluster1' AND
  date = '2015-06-05' AND datacenter = 'US_WEST_COAST' AND
  hour= 12 AND minute >= 0 AND minute <= 30;

SELECT * FROM numberOfRequests WHERE cluster = 'cluster1' AND
  date = '2015-06-05' AND datacenter = 'US_WEST_COAST' AND
  hour >= 12;

SELECT * FROM numberOfRequests WHERE cluster = 'cluster1' AND
  date = '2015-06-05' AND datacenter > 'US';
```

Incorrecta:

```
SELECT * FROM numberOfRequests WHERE cluster = 'cluster1' AND
 date = '2015-06-05' AND datacenter = 'US_WEST_COAST' AND
 hour >= 12 AND minute = 0;
```