# Base de datos NoSQL Cassandra

Carlos Lozano Casado
Marina Jiménez Garrido

# Introducción

- Cassandra es una base de datos NoSQL
- Open source, software libre
- Base de datos distribuida
- Con guardado de columna ancha (wide column store). Utiliza tablas, filas y columnas
- Pensada en acomodar gran cantidad de datos acomodados en varios servidores
- Sin un solo punto de fallo, es decir que si falla un servidor no se cae la base de datos.

# Origen

Cassandra nació de la necesidad de Facebook de potenciar su funcionalidad de búsqueda y de manejar un gran volumen de datos.

Querían encontrar la mejor forma de juntar rapidez en la respuesta a las consultas y rapidez en la lectura de grandes volúmenes de datos.

Fue lanzada como un proyecto open source de Google Code en 2008.

Está inspirada e influenciada por los artículos de Amazon Dynamo de 2007 y de Google BigTable de 2006.

# Características

- **Distribuida**: Cada nodo tiene el mismo rol. Cada nodo podrá contener distintos datos, pero no hay maestro/esclavo.
- **Soporte para replicación y datos repetidos**: Estos son configurables ya que está orientado a múltiples nodos a lo largo de múltiples centros de datos.
- **Escalabilidad**: Diseñada para tener un rendimiento de lectura y escritura, ambos aumentan linealmente a medida que se agregan nuevas máquinas.
- **Tolerancia a fallos**: Los datos son replicados en múltiples nodos para tolerancia de fallos.
- **Consistencia ajustable**: Dentro del teorema CAP, Casandra es AP, es decir, la tolerancia y disponibilidad son más importantes que la coherencia.
- **Soporte de MapReduce**: Cassandra tiene hadoop integrado.

# Ejemplos de uso

- **Facebook**: Utiliza Cassandra en las búsquedas de mensajería instantánea.
- **Netflix**: La utiliza en la búsqueda de películas por palabra clave.
- **Twitter**: La utiliza en su aplicación de Real Time Analytics, Rainbird, pues por su alto volumen de escritura de datos Cassandra ha mostrado una baja latencia.
- **Ebay**: La utiliza por su alta escalabilidad horizontal y su gran rendimiento de operaciones de escritura concurrentes.

# Lenguaje CQL. Crear tabla

El Cassandra Query Language(CQL), que es una alternativa al SQL, añade una capa de abstracción que oculta la implementación de los detalles de la estructura.

- **Crear tabla**:

```
CREATE TABLE [IF NOT EXISTS]
[keyspace_name.]table_name
        column_name tipo,
        ...
        PRIMARY KEY (column_name [, column_name ...])
```

**Ejemplo**:

```
CREATE TABLE cycling.race_winners (
    race_name text,
    race_position int,
    cyclist_name FROZEN<fullname>,
    PRIMARY KEY (race_name, race_position));
```

Lo que hay entre [] es opcional.

La clave primaria se escribe así si es compuesta o

```
    race_name text
PRIMARY KEY,
```

si es simple.

Tiene multiples opciones, pero esto es lo básico.

# Lenguaje CQL. Insertar

- **Insertar**:

INSERT INTO [keyspace_name.] table_name (column_list)

VALUES (column_values) [IF NOT EXISTS][USING TTL seconds | TIMESTAMP epoch_in_microseconds]

**Ejemplo**:

```
INSERT INTO cycling.cyclist_categories
(id,lastname,categories)
  VALUES(
    '6ab09bec-e68e-48d9-a5f8-97e6fb4c9b47',
    'KRUIJSWIJK',
    {'GC', 'Time-trial', 'Sprint'});
```

*USING TTL* es para borrarlo tras una cantidad de segundos y *TIMESTAMP* es para que marque el dato con una marca de tiempo.

Todas las claves primarias tienen que aparecer y se insertará null en las columnas excluidas.

# Lenguaje CQL. Drop table

- **Drop table**:

  DROP TABLE [IF EXISTS] keyspace_name.table_name

- **Delete**:

  DELETE [column_name (term)][, ...]

  FROM [keyspace_name.] table_name

  [USING TIMESTAMP timestamp_value]

  WHERE PK_column_conditions

  [IF static_column_conditions]

Borra los datos de la columna especificada de las filas especificadas, poniendo null, o la fila si no se especifica columna.

La estructura es la siguiente: las columnas que se quieren borrar con su tipo, para las colecciones; de qué tabla son; el *USING TIMESTAP* si se quiere borrar datos más antiguos que la fecha que se ponga; el *WHERE* con las condiciones que tiene que cumplir la clave primaria y el *IF* para condiciones de los campos estáticos.

# Lenguaje CQL. Select

- **Select**:

  SELECT * | select_expression

  FROM [keyspace_name.] table_name

  [WHERE partition_value]

  [ORDER BY PK_column_name ASC | DESC]

  [LIMIT N]

*Select* * muestra todas las columnas (o especifica cuales). *FROM* para saber de qué tabla es, *WHERE* si se quiere poner condiciones para que se muestre una fila, *ORDER BY* si se quiere un orden y *LIMIT* para que solo muestre los *N* primeros resultados.

# Bibliografía


https://docs.datastax.com/en/cql/3.3/cql/cql_reference/

https://www.paradigmadigital.com/dev/cassandra-la-dama-de-las-bases-de-datos-nosql/

https://www.panel.es/blog/base-de-datos-cassandra/