TRABAJO TEMA 4

## 

## 

## 

## 

## 

## 

Participantes:

David Muñoz Sánchez

Manuel Contreras Orge

Juan Navarro Maldonado

Joaquín Sergio García Ibáñez

Rodrigo Corrales Martínez

ÍNDICE

[Descripción de la descarga e instalación de Apache Cassandra](#_orbn5f58iac0) 3

[Descripción del DDL y DML utilizado](#_abv2hjidc7aq) 4

[DDL](#_eleu6juj0l98) 4

[DML](#_grvn39r2ntx1) 5

[Selección](#_vjgpwtlxyqsq) 5

[Cláusula de selección](#_xrqqqpk9rhx1) 5

[Cláusula WHERE](#_wbzwtv5l3p3l) 5

[Resultados](#_abwftf6bb1a2) 5

[Inserción](#_76ivwtg8ov0t) 5

[Actualización](#_pk2hwwc7gjwb) 5

[Eliminar](#_3x9lngxe0qf1) 5

[Sentencias empleadas para la creación de estructuras](#_uvlbqe8d7hc0) 6

[Mecanismo de conexión al SGDB desde una aplicación](#_xevt8tjlgkyq) 8

[Discusión de la implementación del SI](#_vyggtoqv8mah) 9

[BIBLIOGRAFÍA](#_wc4pp53ybsao) 9

# Descripción de la descarga e instalación de Apache Cassandra

Para la instalación de Apache Cassandra seguimos los pasos que se nos indican en la web oficial: <https://cassandra.apache.org/_/quickstart.html>

1. Instalamos Docker (Docker Desktop en Mac y Windows y docker.io en Linux).
2. Ejecutamos la orden docker pull cassandra:latest, para obtener la última versión del SGDB NOSQL.
3. Para iniciar Cassandra usamos los siguientes comandos:
   1. docker run --name cassandra cassandra
   2. docker run --rm -d --name cassandra --hostname cassandra --network cassandra cassandra

Las principales características de un SGDB NOSQL orientada a columnas con respecto a cualquier otra base de datos relacional es que mientras una base de datos relacional está optimizada para almacenar filas de datos, normalmente para aplicaciones transaccionales, una base de datos en columnas está optimizada para lograr una recuperación rápida de columnas de datos, normalmente en aplicaciones analíticas. El almacenamiento basado en columnas para las tablas de bases de datos es un factor importante en el desempeño de las consultas analíticas, ya que [reduce notablemente los requisitos globales de E/S del disco](https://docs.aws.amazon.com/redshift/latest/dg/c_columnar_storage_disk_mem_mgmnt.html), y disminuye el volumen de datos que hay que cargar desde él.

De la misma forma que otras [bases de datos NoSQL](https://aws.amazon.com/es/nosql/), las bases de datos columnares están diseñadas para reducir la escala utilizando clústeres distribuidos de hardware de bajo coste para aumentar el desempeño, de manera que resultan ideales para el [almacenamiento de datos](https://aws.amazon.com/es/data-warehouse/) y el procesamiento de [Big Data](https://aws.amazon.com/es/big-data/).

Cassandra es una base de datos orientada a columnas de código abierto diseñada para gestionar grandes cantidades de datos en muchos servidores comerciales. A diferencia de una tabla en una base de datos relacional, las diferentes filas en la misma tabla (familia de columna) no tienen que compartir el mismo conjunto de columnas.

# Descripción del DDL y DML utilizado

## DDL

CQL (Cassandra Query Lnaguage) guarda datos en tablas. Las tablas están localizadas en **keyspaces.** Un keyspace define opciones que se aplican a todas las tablas que contiene. Una de las opciones más importantes de un keyspace es el factor de replicación.

A la hora declarar los keyspaces y las tablas se usa una gramática como la siguiente: keyspace\_name::= name

table\_name::= [keyspace\_name '.' ] name+

También existen unas normas a la hora de declarar estas variables, entre las que destacan una que dice que los nombres deben tener sólo carácteres alfanuméricos y que el tamaño de dichos nombres debe comprender entre 1 y 48 caracteres, ambas cifras inclusive.

Para crear un keyspace hay que usar la siguiente sentencia base, si intentamos crear una ya existente, a menos que se use la sentencia *IF NOT EXISTS* aparecerá un error:

create\_keyspace\_statement::= CREATE KEYSPACE [ IF NOT EXISTS ] keyspace\_name

WITH options

Una estrategia simple que define un factor de replicación para que los datos se distribuyan en todo el clúster, normalmente hacer esto no es una buena elección para la producción, debido a que no se respetan los diseños de la base de datos y puede dar lugar a una latencia muy variada. Al modificar los factores de replicación de las keyspaces se añadirán de manera automática nuevos centros de datos por seguridad.

Todas las tablas deben de tener una llave primaria, que se identifica con PRIMARY KEY y están compuestas por una o más columnas de la propia tabla.

Resumen de las cláusulas más importantes: la cláusula *USE* se encarga de cambiar el keyspace actual a la indicada; la cláusula *ALTER KEYSPACE* modifica las distintas opciones existentes en el actual keyspace; *CREATE TABLE* crea una tabla nueva, que pueden tener distintos tipos de columnas, entre las que destacan las estáticas; al crearse las tablas existen distintas opciones, que se especifican con la cláusula *WITH*; para hacer cambios en la tabla se usa *ALTER TABLE*; también existe la opción de eliminar una tabla con *DROP TABLE*; y por último, se puede cortar una tabla con la cláusula *TRUNCATE*.

## DML

DML se refiere a la descripción de las declaraciones permitidas por CQL para insertar, actualizar, eliminar y consultar datos:

### Selección

La consulta de datos se realiza mediante una cláusula *SELECT*. Estas cláusulas leen una o más columnas para una o más filas de una tabla y devolverá un conjunto de resultados de las filas que coinciden con la solicitud, donde cada fila contiene los valores para la selección correspondiente a la consulta. *SELECT* puede contener:

#### Cláusula de selección

Se define como *SELECT\_CLAUSE* y determina qué columnas se pueden consultar y devuelven el conjunto de resultados.

#### Cláusula WHERE

Se define como *WHERE* y especifica qué filas se consultan.

#### Resultados

Los resultados se pueden agrupar mediante *GROUP BY*, la cual condensa todas las filas que comparten los mismos valores. También se pueden ordenar mediante *ORDER BY*, limitar el número de resultados mediante *LIMIT* y un filtrado de datos mediante *ALLOW FILTERING*

### Inserción

La inserción de datos para una fila se realiza mediante un *INSERT*

### Actualización

Podemos actualizar los valores de una fila mediante una instrucción *UPDATE.*

### Eliminar

Podemos hacer un borrado de las filas y columnas mediante la instrucción *DELETE*.

Por último, si necesitamos agrupar todas las cláusulas anteriores en una misma sentencia *BATCH* para mejorar la legibilidad de nuestro código.

## 

# Sentencias empleadas para la creación de estructuras

* Crear tablas: Creamos por ejemplo dos de los subsistemas de nuestro sistema desarrollado en la práctica (trabajadores y maquinaria).

| CREATE TABLE IF NOT EXISTS Trabajadores ( dni text PRIMARY KEY, sueldo int, nombre text, apellidos text, horas\_trabajadas int ); |
| --- |

| CREATE TABLE IF NOT EXISTS Maquinaria ( ref\_maquina int PRIMARY KEY, nombre text, fecha\_inspeccion timestamp, funcion set ); |
| --- |

* Inserción de tuplas: Insertamos dos tuplas una para cada tabla

| INSERT INTO Trabajadores (dni, sueldo, nombre, apellidos, horas\_trabajadas) VALUES ('12345678A' , 1500, 'Pedro', 'Ramirez', 263); |
| --- |

| INSERT INTO Maquinaria (ref\_maquina, nombre, fecha\_inspeccion, funcion) VALUES ('5678', 'fresadora', toTimeStamp(now()), 'corta una pieza con una forma'); |
| --- |

* Consultar las tablas:

| #Consulta de toda la tabla de trabajadores y de maquinaria SELECT \* FROM Trabajadores; SELECT \* FROM Maquinaria; |
| --- |

| #Consulta solo la referencia de la máquina y su nombre asociado. SELECT ref\_maquina, nombre FROM Maquinaria; SELECT dni, sueldo, nombre FROM Trabajadores; |
| --- |

| #Consultamos el DNI y el Nombre de los trabajadores que hayan trabajado más de 100 horas. SELECT dni, nombre FROM Trabajadores WHERE horas\_trabajadas > 100 |
| --- |

* Modificación de tablas:

| #Añade un nuevo atributo a la tabla: ALTER TABLE Maquinaria ADD piezas text;  #Alteramos el tipo del atributo piezas recién añadido: ALTER TABLE Maquinaria ALTER piezas TYPE int; |
| --- |

* Borrado de tablas:

| DROP TABLE IF EXISTS Maquinaria; DROP TABLE IF EXISTS Trabajadores; |
| --- |

## 

## 

## 

## 

## 

# Mecanismo de conexión al SGDB desde una aplicación

Veremos cómo conectarnos con Cassandra desde una aplicación Python. Lo primero que tenemos que hacer es instalar **cassandra-driver** desde el IDE:

| !pip install cassandra-driver |
| --- |

Una vez instalado, lo importamos

| from cassandra.cluster import Cluster |
| --- |

A continuación nos conectamos a la base de datos:

| cluster = Cluster(contact\_points=['192.168.99.100'], port=9042) |
| --- |

Nos conectamos al cluster:

| session = cluster.connect() |
| --- |

Ya estaría todo configurado, para ejecutar código CQL, usamos session.execute, como por ejemplo:

| session.execute("""CREATE KEYSPACE mikeyspace WITH replication = {'class' : 'SimpleStrategy', 'replication\_factor':1};""") |
| --- |

# Discusión de la implementación del SI

No sería útil implementarlo en nuestro Sistema de Información porque Apache Cassandra se utiliza para manejar una gran cantidad de datos como por ejemplo Netflix, Spotify, etc.. Y no sería eficiente para los SI relacionales como el nuestro ya que no podríamos relacionar varias entidades como por ejemplo nuestro subsistema de Pedidos no podríamos relacionarlo con el subsistema de Clientes. También es difícil almacenar datos porque tienes que conocer previamente las consultas que vamos a manejar.

También tenemos ejemplos de que compañías han usado Cassandra pero por ser de código abierto y no estar del todo maduro han abandonado y por lo tanto no lo usaríamos en nuestro SI. Además Cassandra es un producto adecuado para productos con baja interactividad y dado que nuestro SI se basa en la interactividad de nuestros clientes con nuestros productos no lo vemos adecuado.

A todo esto hay que añadir que, el cambiar todo nuestro Sistema de Información que está hecho a medida para usar una base de datos SQL nos resultaría bastante complejo cambiarlo a una base de datos NOSQL

# BIBLIOGRAFÍA

<https://cassandra.apache.org/_/index.html>

<https://cassandra.apache.org/_/quickstart.html>

<https://cassandra.apache.org/doc/latest/cassandra/cql/dml.html>

<https://cassandra.apache.org/doc/latest/cassandra/cql/ddl.html>

<https://docs.aws.amazon.com/es_es/keyspaces/latest/devguide/cql.ddl.html>

<https://docs.aws.amazon.com/es_es/keyspaces/latest/devguide/cql.dml.html>