

Cassandra

Comenzamos descargando el ya conocido CSV de incidencias. Para trabajar con Cassandra hemos optado por instalar la distribución de DataStax.

Junto a ello, ha sido necesaria una transformación previa de los datos en el archivo CSV. Para ello, hemos desarrollado un script en Python que consiste en realizar transformaciones sobre dos columnas, fecha y hora. En la primera de ellas eliminamos una hora falsa que se repetía a lo largo de todas las filas y, una vez hecho esto, dotamos a la celda del formato 'yyyy-mm-dd'. En el caso de hora, hemos añadido los segundos, con lo que el formato resultante que tuvimos en esta celda fue 'hh:mm:ss'.

```
import numpy as np
import pandas as pd

def convert(date):
    date_aux = str(date)
    return str(date_aux[6:10]+"-"+date_aux[0:2]+"-"+date_aux[3:5])

def seconds(time):
    time_aux = str(time)
    time_aux+=':00'
    return str(time_aux)

df = pd.read_csv('DataSet.csv')
df['Date'] = df['Date'].apply(convert)

df['Time'] = df['Time'].apply(seconds)

df.to_csv('DataSetModTime.csv',index=False)
```

Una vez tenemos el archivo fuente de datos listo para trabajar pasamos a la propia base de datos, mencionando antes que hemos tenido algunos problemas (al menos en el caso de Windows), que han requerido entrar en archivos de configuración y cambiar algunos parámetros que, por defecto, incluyen rutas de Linux, por lo que obteníamos errores debido a ello.

Para comenzar nuestro desarrollo con Cassandra, el primer paso es crear un KEYSPACE que no será más que una colección de consultas o tablas.

```
CREATE KEYSPACE IF NOT EXISTS incidents
WITH replication = {
    'class': 'SimpleStrategy',
    'replication_factor': 1
};
```

Una vez tenemos nuestro KEYSPACE listo, pasamos a crear las consultas. En primer lugar, hemos querido crear una consulta que se limite a almacenar todos los datos tal y como están en el CSV. A

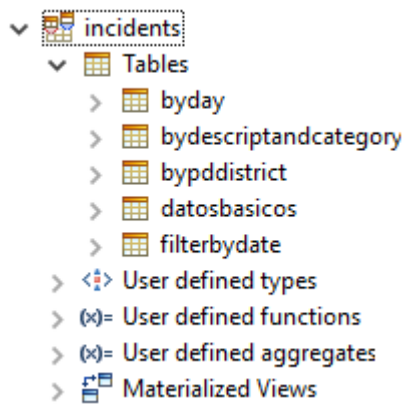
la hora de realizar estas consultas hay que indicar en la sintaxis al menos una variable que actuará como clave de partición y, adicionalmente a esto, otra variable que nos ayudará a ordenar las particiones en base a la clave de partición de acuerdo a esta segunda variable. Esta operación sí podemos llevarla a cabo dentro de DataStax, pero para el volcado de los datos en ella hemos de irnos al Shell propio de Cassandra que obtenemos también al instalar esta distribución.

```
CREATE TABLE IF NOT EXISTS incidents.datosbasicos(  
  incidentNumber text,  
  category text,  
  descript text,  
  dayofweek text,  
  date date,  
  time time,  
  pddistrict text,  
  resolution text,  
  address text,  
  x text,  
  y text,  
  location text,  
  pdid text,  
  PRIMARY KEY (incidentNumber)  
);
```

Este volcado se lleva a cabo mediante las operaciones de COPY, indicando la tabla que queremos rellenar de datos, las columnas, y el fichero fuente a partir del cual obtener esos datos. En nuestro caso, al ser tantas filas y el ordenador tener únicamente ocho gigas de ram, optamos por incluir opciones adicionales en el comando de COPY (CHUNKSIZE y INGESTRATE), para ir tomando pequeños fragmentos del archivo CSV y con ello poder realizar la operación correctamente.

```
COPY incidents.datosbasicos (incidentNumber, category, descript, dayofweek, date, time, pddistrict, resolution, address, x, y, location, pdid) FROM  
'C:\...\CSV.csv' WITH HEADER=FALSE and CHUNKSIZE=500 and INGESTRATE=2000;
```

Hecho esto tendremos nuestra consulta o tabla lista para usar. Algo también a tener en cuenta es que, si queremos realizar operaciones sobre esta tabla del tipo '... WHERE X=...', esta variable X idealmente será parte de la clave de partición en este caso.
























A partir de aquí hemos realizado algunas tablas adicionales, con claves de partición tanto compuestas como simples. Para estas siguientes tablas Cassandra no nos permite coger los datos de nuestra primera tabla, con lo que dos posibles soluciones son o bien utilizar nuestro CSV inicial (o alguna modificación del mismo ya sea con Python u otro lenguaje), o bien volcar nuestra tabla original a un CSV de manera temporal y volcar este archivo CSV a la nueva tabla.

```
CREATE TABLE IF NOT EXISTS incidents.bydescriptandcategory(  
    incidentNumber text,  
    category text,  
    descript text,  
    dayofweek text,  
    date date,  
    time time,  
    pddistrict text,  
    resolution text,  
    address text,  
    x text,  
    y text,  
    location text,  
    pdid text,  
    PRIMARY KEY((descript, category), time)  
) WITH CLUSTERING ORDER BY (time DESC);
```

```
COPY incidents.datosbasicos (incidentNumber, category, descript, dayofweek, date, time, pddistrict, resolution, address, x, y, location, pdid) TO 'temp.csv';
```

```
COPY incidents.bydescriptandcategory (incidentNumber, category, descript, dayofweek, date, time, pddistrict, resolution, address, x, y, location, pdid) FROM 'temp.csv';
```

- ▼  bydescriptandcategory
 - ▼  Columns
 -  descript (text)
 -  category (text)
 -  time (time)
 -  address (text)
 -  date (date)
 -  dayofweek (text)
 -  incidentnumber (text)
 -  location (text)
 -  pddistrict (text)
 -  pdid (text)
 -  resolution (text)
 -  x (text)
 -  y (text)
 - ▼  Partitioning Key
 -  descript (text)
 -  category (text)
 - ▼  Clustering Column
 -  time (DESC) (time)
 - >  Secondary Indexes