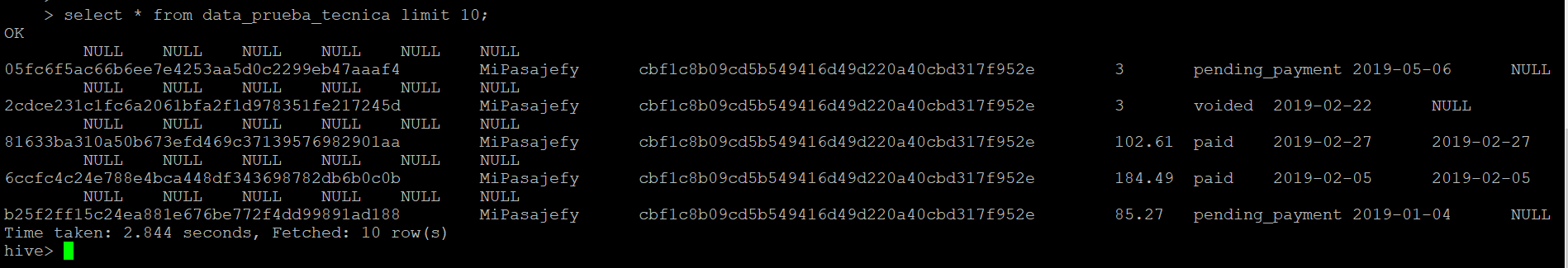
**Prueba técnica: Data Engineer**

**Sección 1: Data pipeline**

**1.1 Carga de información**

La información compartida se cargó a hive, se subió el archivo a un servidor y se tomó desde una ruta especifica. Se adjunta archivo con DDL y script de carga (loadDataConekta\_hive.sql)

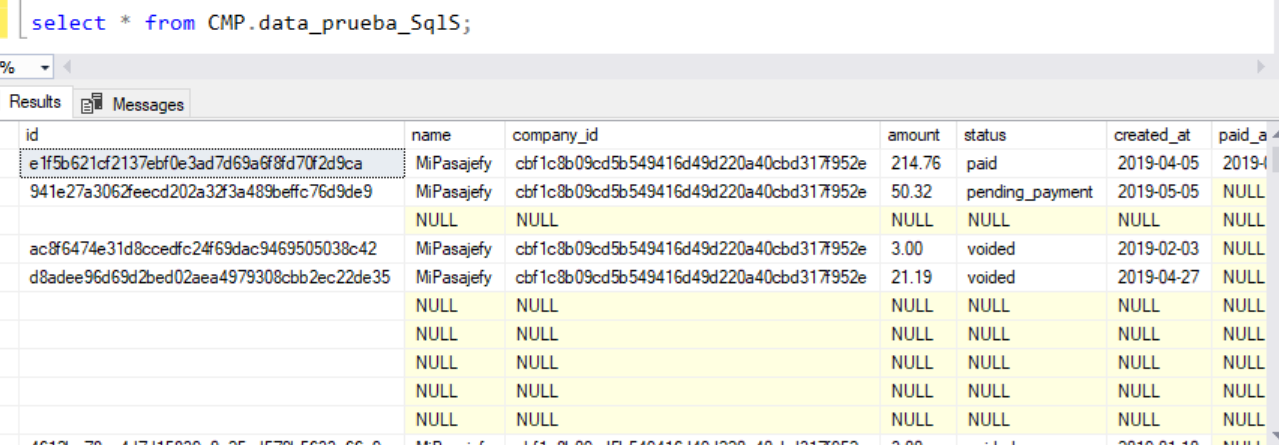
Muestra de la carga:



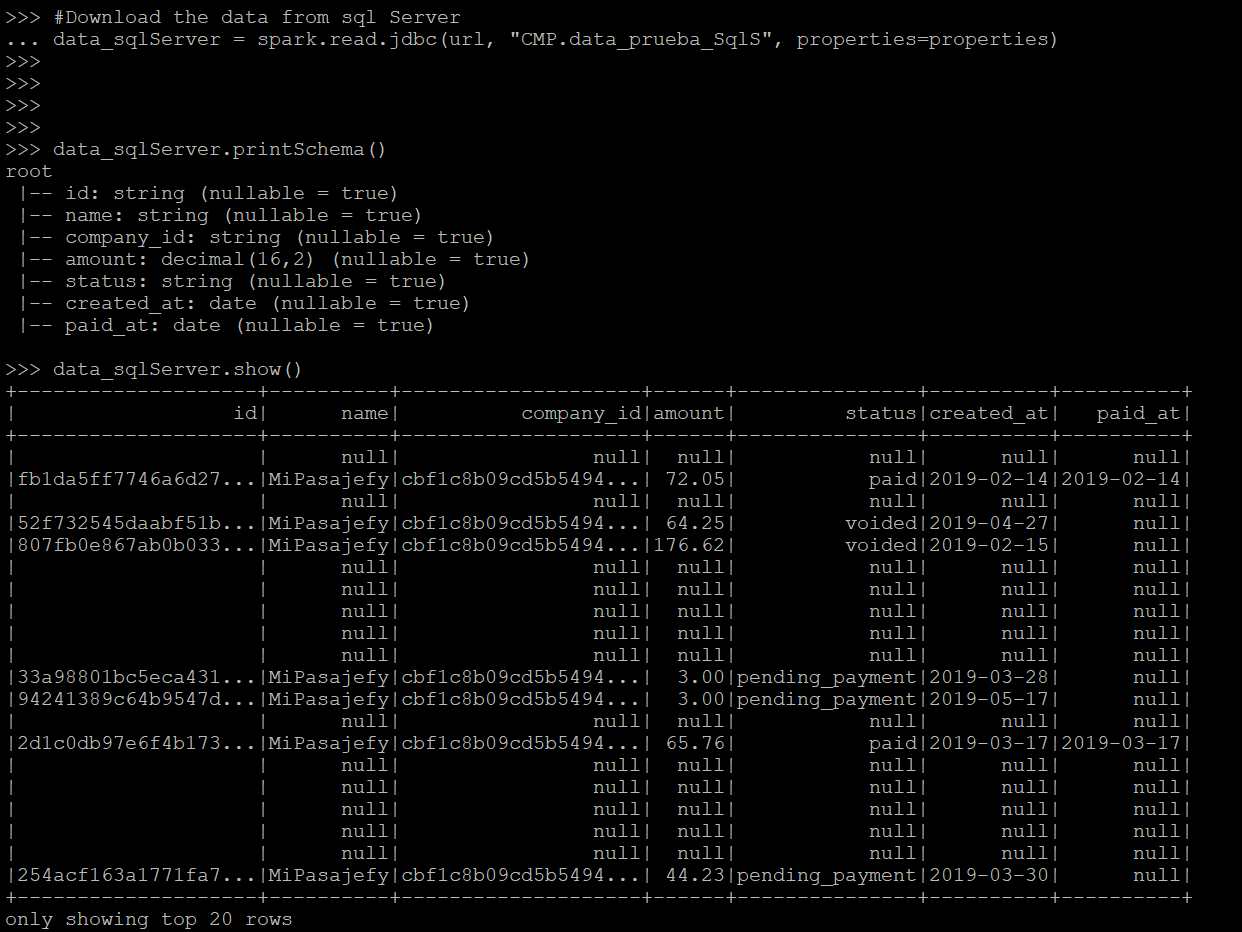
Se eligió este tipo de base para facilitar la trasformación y comunicación con spark para los siguientes puntos.

**1.2 Extracción**

Se realizó la lectura de la información cargada a hive desde el lenguaje spark y posteriormente se cargó mediante una conexión jdbc a un servidor de sql Server.

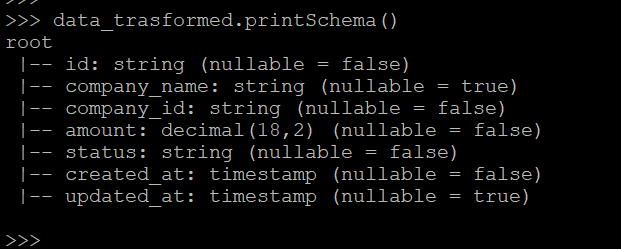


Para posteriormente realizar la extracción de la información desde ese servidor mediante conexión jdbc con el lenguaje pyspark.

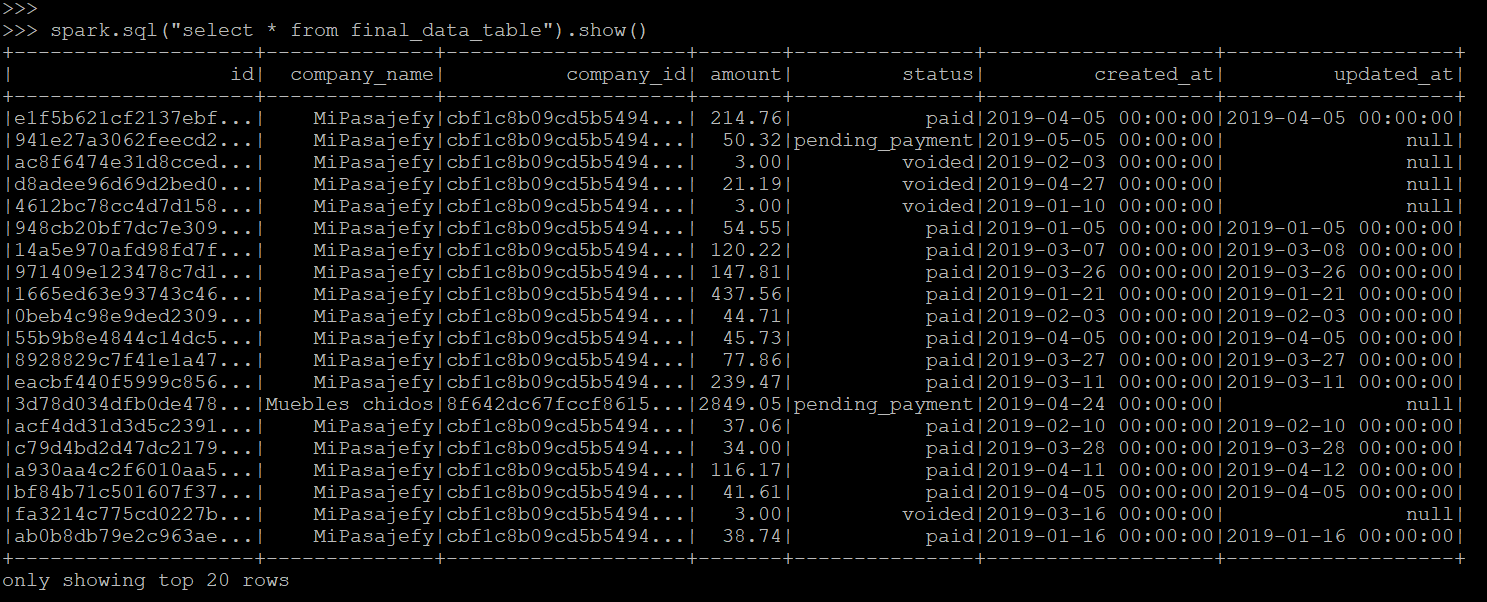


**1.3 Transformación**

Para La transformación de la información se continuó trabajando con spark, se hizo limpieza de la información, ya que el archivo traía líneas en blanco. Para llegar al esquema esperado me topé con algunas dificultades, dado que spark para campos de tipo cadena solo maneja tipo string y la longitud de campo se toma por el tamaño de la cadena, para acotar los tamaños al tipo varchar(), utilicé la función substr() para acotar el campo al tamaño requerido. El esquema de la información final se insertó en una tabla de hive en tipo parquet. Esquema final:



Información final cargada a hive:



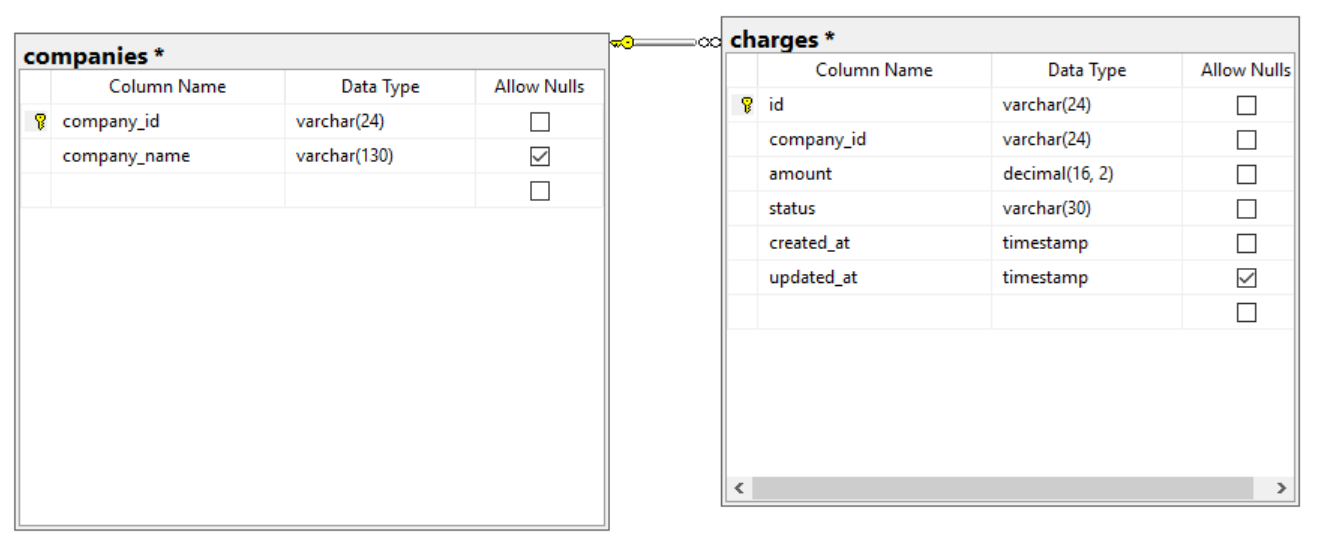
Se eligió este lenguaje porque es el lenguaje en el que tengo más experiencia trabajando, además de que es uno de los lenguajes más recomendados para hacer transformación de la información y uno de los lenguajes en tendencia que se adapta a las nuevas tecnologías del Big Data.

Se adjunta script de transformación de la data (pipelineConekta.py)

Linea de ejecución del script: spark-submit --driver-class-path mssql-jdbc-7.4.1.jre8.jar --jars mssql-jdbc-7.4.1.jre8.jar --driver-memory 8g --num-executors 4 --executor-memory 8g --executor-cores 4 pipelineConekta.py "user" "password" "host" "port"

**1.4 Dispersión de la información**

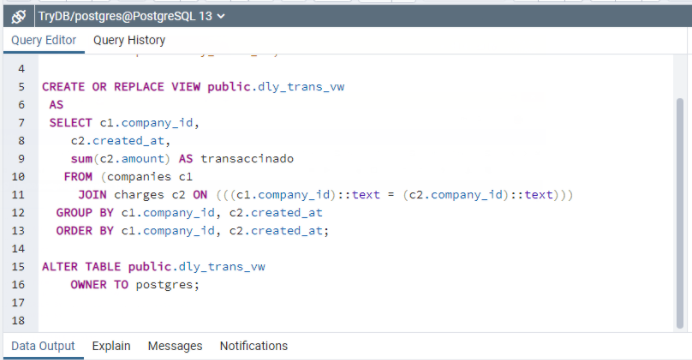
Diagrama relacional de las tablas.



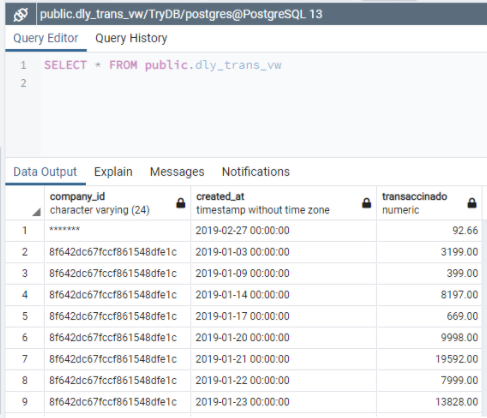
\*Se adjunta companies\_ddl.sql y charges\_ddl.sql

**1.5 SQL**

Vista en PostgreSQL



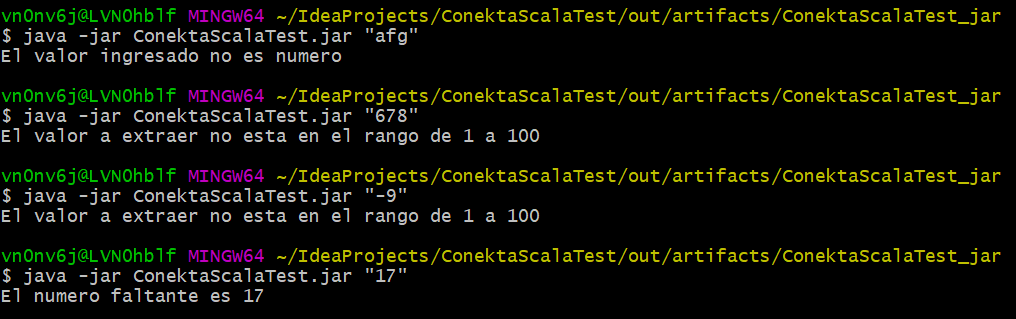
Resultado de la vista



\*Se adjunta dly\_trans\_vw\_ddl.sql

**Sección 2: Scala**

Evidencia de la ejecución



\*Se adjunta proyecto y jar