Base de Datos 2
Trabajo Práctico Obligatorio

Integrantes:
Franco Morroni (60417) , fmorroni@itba.edu.ar
Bernardo Zapico (62318) , bzapico@itba.edu.ar
Alejo Padula Morillo (63571) , apadulamorillo@itba.edu.ar

20 de diciembre de 2025

Elección de bases:

Inicialmente, pensamos en utilizar MongoDB para la información de los proveedores y sus teléfonos, Cassandra para sus productos y Neo4j para las órdenes. La idea era tener MongoDB para proveedores por su capacidad de rápidos filtrados y accesos mediantes índices, Cassandra para productos pues habíamos anticipado que tal vez habría muchas inserciones y consultas solo por sus identificadores, y Neo4j para las relaciones entre proveedores, órdenes y varios productos. Pero finalmente descartamos el uso de Cassandra para los productos y dejamos MongoDB para manejar también eso en su lugar, después de llegar a la conclusión de no habría realmente muchas inserciones en simultáneo y MongoDB ya tiene suficiente capacidad de manejar grandes volúmenes de datos para el contexto de una base de productos. Por lo que terminó quedando solo dos bases de datos:

- MongoDB para proveedores y productos, para estas tablas se guardó toda la información de dichos elementos, los teléfonos se agregaron dentro de un arreglo para cada proveedor, ya que pueden tener más de uno.
  También creamos un índice para los CUIT del proveedor, ya que consideramos como una clave secundaria y la usaríamos como input cuando el usuario de nuestra aplicación quisiera cambiar o borrar algún proveedor. Con una misma idea, creamos un índice para la descripción y marca de un producto (clave compuesta), nuevamente considerándolo como una clave secundaria. Esto se usaría para la actualización de productos.
- Neo4j para las órdenes y relaciones entre estas con los proveedores y productos. Tenemos un nodo por cada producto solo con la información de su id, un nodo por cada proveedor solo con la información de su id y, más importante, un nodo para cada orden con la información de su fecha y porcentaje de iva.
  Cada orden tiene relaciones “HasItem” hacia los productos que forman parte de ella, esta relación contiene el valor de la cantidad pedida del producto y su precio al momento de realizar la orden. También tiene una relación “OrderedFrom” para indicar a qué proveedor se realizó ese pedido.

Cabe mencionar que el sistema desarrollado no maneja transacciones entre las dos bases de datos, esto es obviamente un problema, pero uno que no pudimos resolver. Tanto mongodb como neo4j se encargan de las transacciones para las queries propias, pero en los casos donde hay que modificar datos en ambas

bases a la vez podrían surgir problemas de consistencia en caso de que haya un fallo en una base pero no en la otra.

Por otro lado, para los queries 10 y 11 se pide crear vistas, pero los datos necesarios para estas vistas en nuestro caso están en parte en mongodb y mayoritariamente en neo4j. Neo4j no cuenta con manejo de vistas por lo cual hubo que optar por realizar queries normales sin vistas para estos casos. La vista pedida en el query 12 sí fue realizada correctamente ya que toda la información necesaria se encuentra en mongodb que sí admite el uso de vistas.

Resultados de los queries

Los resultados se entregan en archivos aparte adjuntos en el mail.

Instrucciones para ejecutar el trabajo:

- Diríjase a la página de Codespaces de Github: Codespaces (github.com)
- Seleccione “new codespaces”
- En “Select a repository” busque y seleccione el repositorio “berni-245/BD2-TP”, finalmente creé el Codespace
- Dejar que cargue el Codespace y desde ahí seguir los pasos del readme. Ahí se especificará sobre instalar pipenv, en caso de tener que elegir la versión de python para este, usar “which python”, copiar la dirección del ejecutable y luego correr “pipenv --python <path_python>”

Aclaraciones:

- Una vez corrido el docker-compose up -d y con los containers creados, esperar un poco antes de correr el el comando para inicializar la base de datos con los datos de los csv, ya que estará usando los puertos unos segundos adicionales
- Desde el Codepsace la interfaz visual normalmente ubicada en http://localhost:7474 de Neo4j, no funciona