Base de Datos - 72.41
2do cuatrimestre 2023

Felipe Hiba, 61219
fhiba@itba.edu.ar
Patricio Escudeiro,
pescudeiro@itba.edu.ar
Gastón Alasia,
galasia@itba.edu.ar
Matias Della Torre, 61016
mdella@itba.edu.ar

14 de Noviembre de 2023

Introducción

El objetivo de este trabajo práctico fue demostrar nuestros conocimientos tanto de PostgreSQL como de MongoDB y las distintas herramientas que estos poseen.

Se armaron la serie de queries requerida para cada motor de bases de datos y también se desarrolló a la par una API en Javascript usando el framework de Express.js que nos permite acceder a cualquiera de dichas bases y aplicar comandos CRUD. Se trata de dos servidores separados que pueden ser ejecutados de manera individual o simultánea.

Desarrollo

APIs

Se implementaron dos APIs básicas escritas en Express.js: una para PostgresSQL y otra para MongoDB. El usuario primero introduce las credenciales de la base de datos a conectarse. Se trata de un único proyecto Node, donde al ejecutarlo se elige qué API levantar (ver README en el repositoris).

Las APIs no cuentan frontend si no que son solo endpoints para llamar a las funciones pedidas por el Trabajo Práctico. Se recomienda utilizar un programa como Postman para probarlas.

Consultas SQL

Para las consultas en SQL se utilizó lo aprendido en clase para intentar hacerlas lo más eficientes posibles. Se empezó por plantearlas de distintas formas y luego, una vez que se obtuvo el resultado deseado, se procedió a correrlas junto con EXPLAIN ANALYZE para luego decidir cual de las consultas era las de menor costo. No se encontraron grandes inconvenientes a la hora de plantear las queries y se pudo resolver lo pedido con la ayuda del material de la cátedra.

Vistas SQL

Al momento de desarrollar las vistas, se consideró que lo más importante era generar vistas actualizables y luego se utilizó el mismo proceso que con las queries para poder encontrar las de menor costo previo a generar las vistas para hacerlas lo más eficientes posibles.

Consultas Mongo

Para las consultas en Mongo también se utilizó lo aprendido en clase para intentar hacerlas lo más eficientes posibles. Sin embargo, surgieron algunas dificultades adicionales ya que no se tenía tanta práctica de su sintaxis como la que se tenía con Postgres. Se utilizó principalmente el operador lookup para realizar los ensambles así como proyecciones para poder obtener solo los campos deseados. Surgieron algunas dudas adicionales, que tuvieron que ser consultadas en Internet.

Vistas Mongo

Al momento de desarrollar las vistas en Mongo, no se tomaron consideraciones adicionales ya que en Mongo las vistas directamente no son actualizables.

Dificultades durante el desarrollo

La principal dificultad encontrada, sobre todo en la segunda parte, fue la de transformar una API que originalmente estaba pensada solo para Postgres a dos APIs que

compartieran el mismo proyecto Node pero que fueran independientes (es decir, dos servidores distintos de Postgres y de Mongo). Para solucionarlo, se debió reestructurar enteramente el proyecto e investigar sobre Node. Finalmente, se pudo reutilizar el código de la API de Postgres con modificaciones para la de Mongo. Esto, además, llevó a que ambas APIs compartan la estructura de sus endpoints, por lo que la documentación de una sirve para la otra.

Conclusión

Como conclusión, se pudo implementar lo pedido con lo visto en clase con el objetivo de poner en práctica las nuevas herramientas tanto de PostgreSQL como de MongoDB. Cierta información adicional fue obtenida de Internet.