

## Trabajo Práctico Obligatorio

- **Cantidad de integrantes:** 3
- **Fecha de entrega:** 19 de junio de 2024
- **Forma de entrega:** a través de **Codespace** en Github. Adicionalmente realizar un informe/presentación para su defensa, donde se muestren los comandos utilizados para resolver cada punto
- **Repositorio de datasets:** <https://11nk.dev/aTrRE>

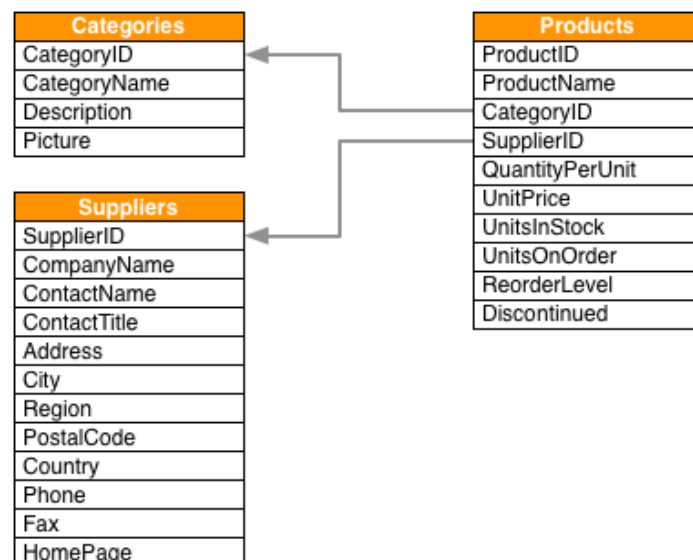
### Resolver los siguientes 3 ejercicios

#### Ejercicio 1 - MongoDB

- Importe el archivo **albumlist.csv** (o su versión RAW) a una colección. Este archivo cuenta con el top 500 de álbumes musicales de todos los tiempos según la revista Rolling Stones.
- Cuente la cantidad de álbumes por año y ordénelos de manera descendente (mostrando los años con mayor cantidad de álbumes al principio).
- A cada documento, agregarle un nuevo atributo llamado 'score' que sea 501-**Number**.
- Realice una consulta que muestre el 'score' de cada artista.

#### Ejercicio 2 - Neo4j

Utilizar un sandbox en blanco provisto por Neo4j (<https://sandbox.neo4j.com/>) para importar un dataset provisto por los desarrolladores de Neo4j que contiene productos, categorías y proveedores simulando ser una base de datos de un comercio minorista o almacén.



Ejecutar el comando **:play northwind-graph** desde la interfaz web de Neo4J (con los : antes del play) para generar la base de grafos en cuestión.

Al ejecutarlo dirigirse a la página 2. Allí encontrarán un total de 7 comandos disponibles para usar. Vamos a correr los primeros tres, que cargan nuestra base a través de un LOAD CSV.

Continuar en la página 3 y ejecutar los únicos 2 comandos disponibles, que cargan las relaciones para los datos que subimos anteriormente.

Con la base ya cargada, responder con una sola consulta cada punto:

- ¿Cuántos productos hay en la base?
- ¿Cuánto cuesta el “Queso Cabrales”?
- ¿Cuántos productos pertenecen a la categoría “Condiments”?
- Del conjunto de productos que ofrecen los proveedores de “UK”, ¿Cuál es el nombre y el precio unitario de los tres productos más caros?

### Ejercicio 3 - Redis

Se pone a disposición el archivo **bataxi.csv** (con los viajes realizados por los taxistas que usan la app BAtaxi) para importarlo en Redis utilizando el comando [GEOADD](#), considerando los siguientes **atributos**.

- key → "bataxi"
- longitude → **origen\_viaje\_x**
- latitude → **origen\_viaje\_y**
- member → **id\_viaje\_r**

Dada la siguiente lista de lugares y sus coordenadas:

```
places = [{"place": "Parque Chas", "lon": -58.479258, "lat": -34.582497},  
          {"place": "UTN", "lon": -58.468606, "lat": -34.658304},  
          {"place": "ITBA Madero", "lon": -58.367862, "lat": -34.602938}];
```

- Importar los datos del archivo a Redis
- ¿Cuántos viajes se generaron a 1 km de distancia de estos 3 lugares?
- ¿Cuántas KEYS hay en la base de datos Redis?
- ¿Cuántos miembros tiene la key 'bataxi'?
- ¿Sobre qué estructura de Redis trabaja el GeoADD?

**Nota:** Acceder al siguiente sitio <https://redis.io/docs/latest/develop/data-types/geospatial/> para investigar sobre comandos de REDIS para bases de datos geo-referenciados.