Laboratory 3.2

1. Keys and **relation** for which DB and why

Redis: Catalog data and cart data for each user.

- The **products** data is stored in an **external json file**. After first request this data is saved in a JSON data type in Redis like:

  ```
  JSON.SET product:{productId} . '{ "id": "productId", "name": "Product Name", "price": "375.00", "stock": 10 }'
  ```

- The **cart** data is stored in a **hash** like:

  `HSET cart:{cartId} product:{productId} {productQuantity}`, where cartId is random generated value and stored in user session.

Why? **R:** Shopping cart data is stored on the client-side as a **cookie**.

Advantages:

- We don't need to store such temporary data in your database.
- Retrieve the items very fast at any time and persist this data if needed (Abandoned cart example)
- Hash cart allows to relate multiples products to differents carts

MySQL: Product information

Why? **R:** Ensure data consistency and availability/ large data

Advantages:

- Store permanent and large data
- Retrieve only when it's necessary

Situations:

- DELETE BOTH: Si eliminas un producto, se debe eliminar el producto MySQL y en el catálogo de productos de Redis, eliminandolo tambien de todos los carritos en los que estubiera.
- UPDATE BOTH: Si modificas el atributo de un producto (nombre o precio) hay que modificarlo en todos los carritos que contengan ese producto.
- INSERT BOTH: Si añades un producto, se debe añadir el producto en la BD de MySQL y Redis

Modificaciones estructurales: El catálogo de productos debe pasarse a MySQL (revisar si es el uso normal de las bd key-value)

Relation: Relacionates purchased carts to user history

Follow: https://developer.redis.com/howtos/shoppingcart/