

# Data Modeling

---

## Step 1 - Product Catalog Use Case

In the last session several common Redis use cases were discussed. Now, let's consider using Redis as our primary data base for a product catalog for a new online store. It will need to store product details should including a name, description, vendor, price, main category and some images.

## Step 2 - Requirements:

- Product information stored in the database should include: name, description, vendor, price, category, images associated with that product
- Ability to create/update/delete product details
- Ability to find product by ID
- Ability to find products in category X
- Ability to find product by it's name or part of it's name

## Step 3 - Logical Data Model:

The logical data model is separate from the DBMS being used...it defines the structure of data elements and to set relationships between them.

- Product Image
  - Id : Number
  - Value : Binary
- Product
  - Id : Number
  - Name : String
  - Description: String
  - Vendor : String
  - Price : Number
  - Currency : String
  - MainCategory : Category (1)
  - Images : Image (0..n)
- Category
  - Id : Number
  - Name : String
  - Products : Product (0..n)

## Step 4 - Convert to Physical Data Model for Redis:

How would you take this logical data model and turn it into a physical model in Redis? What data structures would you use to meet the requirements listed above? How to index and query the data?

Be ready to show via redis-cli (or Insight) how you stored the data and how you are able to find the product by ID, list all products in a given category and find a product by full or partial name.

## Step 5 - Bonus: Build a REST API endpoint and simple middleware

If you are looking for an extra challenge, take this data model into an actual implementation! Create a REST API endpoint and middleware to interface with Redis and your model.