We have a product image component that serves the correct image density for products according to a device’s request.

In the spreadsheet titled: **Backend Test Data - Image URL Component**
There is a sheet showing the sample data for our database.
Often the images will be updated but the products will remain the same.

Considerations:

1. New products will always be added in the future
2. New devices that use this service will be added in future
3. The database structure should be built so that it can easily handle new devices being added without having to modify the table structure. (We only want to insert images for new devices, we do not want to alter the table).

3 Tasks - Time (2 hours)

1. Database Setup

Build a database structure that would efficiently serve this data and populate it with the sample data provided.

Output: • We would like to see your database creation scripts
• Make notes for yourself because we will have a final discussion with you if we have any questions!

2. Function 1

Inputs:

- Operating System (text)
- Image Density (text)

Output:

A. Should output list of Products with:
  - Product Title
  - URL

B. Should only output Products that are available

C. Products should be shown alphabetically

D. Output should be in JSON format

E. Output only needs to be printed to log output - does not need to be returned on a json endpoint

Sample Output:

```
{
  "products": [
    {
      "title": "The Apple",
      "url": "/ios/1x/the_apple_1x.png"
    }, {
      "title": "Orange",
      "url": "/ios/1x/orange_1x.png"
    }, {
      "title": "Pear",
      "url": "/ios/1x/pear_1x.png"
    }
  ]
}
```

3. Function 2

Inputs:

- Operating System (text)
- Image Density (text)
- Start Release Date (integer) eg. 1950
- End Release Date (integer) eg. 1960
  Note: these numbers are both inclusive, meaning that the start date should include items from the year specified so including 1950 and 1960 in output.

Output:

A. Should output list of Products with:
  - Product Title
  - url
  - release Year

B. Should only output Products that are available
C. Products should be shown in order of release date (ascending)
D. Output should be in JSON format
E. Output only needs to be printed to log output - does not need to be returned on a json endpoint

Task 2 - Sample Output:

```
{
  "products": [
    {
      "title": "Pear",
      "url": "/ios/1x/pear_1x.png"
      "released": "1958"
    }, {
      "title": "The Apple",
      "url": "/ios/1x/the_apple_1x.png"
      "released": "1960"

    }
  ]
}
```