# Notes

**Key requirements:**

- Product name
- Table (main) by Purpose
- Company name
- Dates
  - Start
    - Maybe made vs. received
  - End
- Still in use? (No, yes, never)
- Quantity (string)
- Description
- Test table (child of product)
- Machine table (child of product)
- Product ID (NARA)
- Product ID (Company)
- Product photo (url)
- Test sheet url
- Find a way to differentiatie different items under a product
- Test success or fail? Or ongoing?
- Test dates
- Test types
- Test unique ID's
- Connect machine table to test (test → machine)
- Materials column
- Use null to handle missing values

**Maybe:**

- Hazardous?
- Expiration date

**Additional details info examples:**

- Talk about specific products (ie. one of the items is a defect)
- Explain size metric (ie. 1.5 means that we have one whole sheet and a half sheet since it was cut)
- "Meta data is on item, see in photo"

**NOT in scope or NOT needed:**

- Records (only non-records)
- Do not dispose value
- Standard lab chemicals

# Rough Draft 1

**Tables:**

- **Product**
    - Product name
    - Company name
    - Quantity (string)
    - Description
    - Product ID (NARA) (Unique)
    - Product ID (Company)
    - Product photo (url)
    - Materials
    - Hazardous
    - Purpose
    - Dates:
        - Start
            - Maybe made vs. received
        - End
    - Still in use? (No, yes, never)
    - Test ID
- **Test**
    - Test ID (Unique)
    - Test name
    - Test type
    - Test dates
    - Test sheet url
    - Test success or fail? Or ongoing?
    - Product ID (reference to connect test to product)
- **Machine**
    - Machine name
    - Machine ID (Unique)
    - Status (still working)
    - Test ID (reference to connect machine to test)

**Questions:**

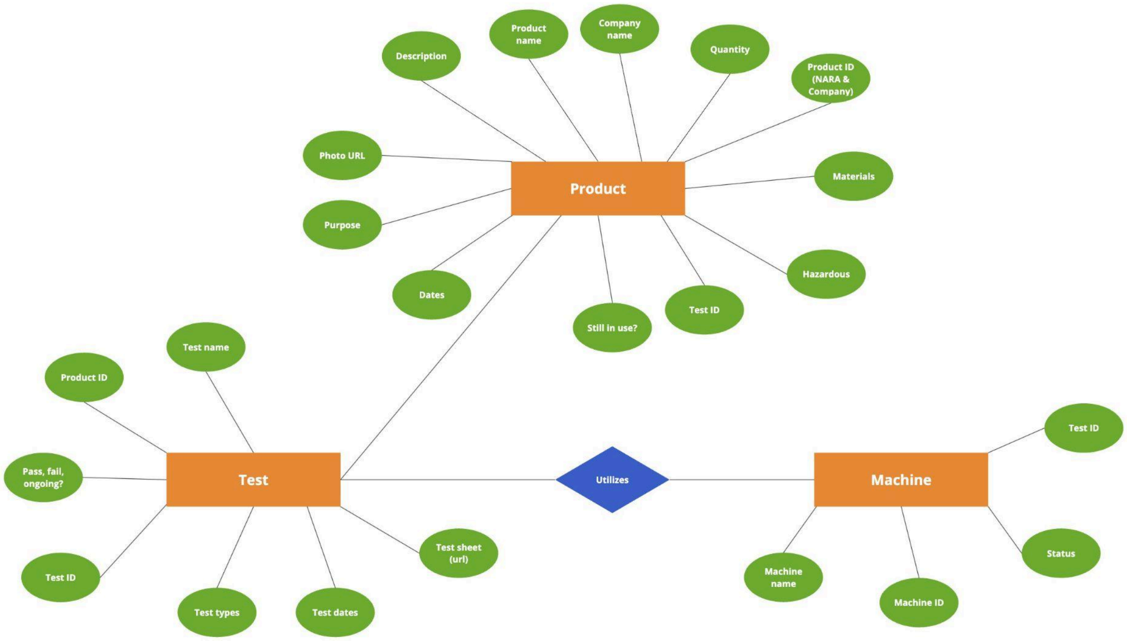
1. Can one test use multiple different products?
2. Can a test utilize multiple machines?

**Feedback:**

1. DB generated unique ID
2. Company product ID
3. Still in use (change to “Approve”)
    a. Yes, no, with

        i. Detials inside descritnion (DOCUMENTATION)

4. Date
    a. Created
    b. Recieved for review
5. Change “pass, fail, ongoing” to “result”
6. Add descrption col to Test table
7. Remove “test type”
8. “Machine” rename to “Instruments”
9. Remove “Status” from Machine
10. Add “descrioption” column to Machine
11. Product → test
    a. Many to many
12. Test → Instrument
    a. Many to one
        i. Instrument can be used in MANY test
        ii. Test can use ONE instrument
        iii. Ex:
            1. Company can make MANY products
            2. Product are from ONE company
            3. Arrow points to Company
13. Maybe add “tested by” column (DOCUMENTATION)



# Rough Draft 2

**Todo:**

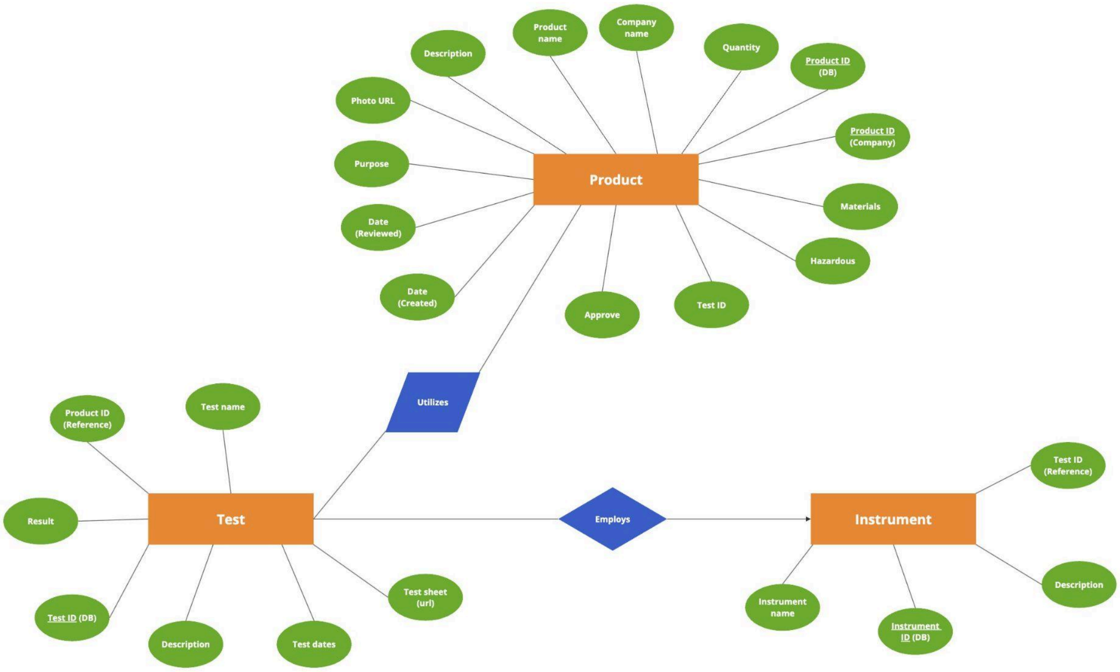
1. Incorporate feedback into ER Diagram
2. Creating tables in SQL
    a. Review feedback

**Table relationships:**

- Product → Test:
    - Verb: utilize
- Test → Machine
    - Verb: employ

**Feedback:**

1. Switch table relationships verb names



## Rough Draft 3

**Todo:**

1. Creating tables in SQL
    a. Review feedback
2. Identify needs to server-side and front-end

**Needs:**

1. **Server Side:**
    a. Java
    b. Spring Boot
    c. XML (Dependencies)
    d. Thunderclient or Postman
2. **Client Side:**
    a. Javascript
    b. React

**Timeline:**

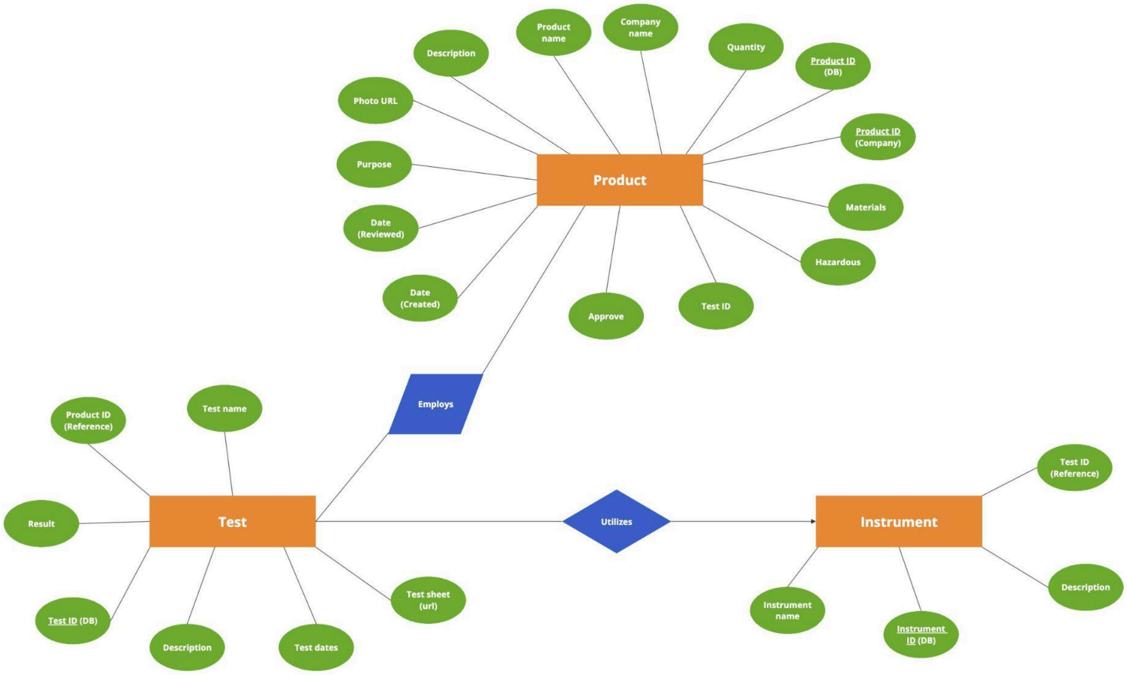
1. Create SQL Table Statements
2. **Server-side:**
    a. Import Spring Boot to Java file(s)
    b. Connect Sprint Boot to MySQL
        i. XML Dependencies
    c. Test Connection to SQL (Don't use SQL yet for url ports)
    d. Create basic models and test url ports
        i. Thunderclient or Postman
    e. Implement SQL queries into models
    f. Implement test to see if server-side ports work as intended
3. **Client-side:**
    a. Create react app
    b. Create and organize components
    c. Create navigation
        i. React-router
    d. Async fetch and call url ports to server-side to grab data
    e. Display data to React
        i. useEffect, useState

**Features:**

1. Filter parameters
    a. Product values
        i. Material type, age, manufacturer, testing data
    b. Additional testing data filter
        i. "Does it have testing data"
2. Sort Parameters

    a. Alphabetical or numerical
        i. Ex:
            1. Date
            2. Quantity
            3. Location

3. Visualize image
    a. Local
    b. Future: other computers
4. Export CSV file & filtered tuples (future)
5. Reading csv file (future)
    a. Check if its a certain test file (XRF, FTIR, etc.)
        i. GMS is very complicated (nested graphs)
    b. Read and plot data



```
CREATE TABLE Product (
 Id INT,
 Product_Id_C VARCHAR(255),
 Materials VARCHAR(255),
 Hazardous BOOLEAN,
 Approve VARCHAR,
 Date_Created DATE,
 Date_Reviewed DATE,
 Purpose VARCHAR(255),
 Photo_URL VARCHAR(255),
 Descriptions VARCHAR(1000),
 Product_Name VARCHAR(255),
 Company_Name VARCHAR(255),
 Quantity VARCHAR(255)
);

CREATE TABLE Test (
 Id INT,
 Test_Name VARCHAR(255),
 Test_Date DATE,
 Descriptions VARCHAR(1000),
 Result VARCHAR(255),
 Test_Sheet VARCHAR(255),
 Instrument_Id INT REFERENCES Instrument(Id)
);

CREATE TABLE Employs (
 Product_Id INT REFERENCES Product(Id),
 Test_Id INT REFERENCES Test(Id)
);

CREATE TABLE Instrument (
 Id INT,
 Instrument_Name VARCHAR(255),
 Descriptions VARCHAR(1000),
);
```