NARA Non-Record Material Database (SQL)

Description:
This database holds the NARA Heritage lab's collection of non-record materials, which includes deaccessioned and naturally aged items, treatment materials, and other relevant samples.

MySQL Password:
NaraDB1804#

Tables:

1. **Product**
    a. **Features:**
        i. **Id (Integer):** auto-incremented generated value from database
            1. Automatically generated by DB
        ii. **Product_Name (String):** name of the product
            1. REQUIRED
        iii. **Company_Name (String):** the name of the manufacturer that made the product
        iv. **Product_Id_C (String):** id given to the product by the manufacturer
        v. **Hazardous (boolean):** whether or not the product is hazardous
        vi. **Approve (String):** whether or not the product is still in use
            1. Values: “yes”, “no”, “with”, null
                a. If the value is “with”, explain it in the description column
        vii. **Date_Created (Date):** date when the product was created
            1. Format: year-month-date
                a. Ex: 2020-09-18
        viii. **Date_Reviewed (Date):** date when the product was reviewed by NARA
            1. Format: year-month-date
                a. Ex: 2020-09-18
        ix. **Purpose (String):** the purpose of the product
            1. Values: “Holding Maintenance”, “Exhibitions”, “Conservation Treatment”, “Record Materials”
            2. Full documentation of this in Google Doc named “Suggested Purpose Categories”
        x. **Photo_URL (String):** the url of the product’s photo

xi. **Descriptions (String):** additional info about product
  1. Length: 1500 characters
  2. Ex: explain size metric (ie. "1.5 means that we have one whole sheet and another half sheet since it was cut")
  3. Ex2: talk about specific product (ie. "one of the items is a defect")

xii. **Quantity_Metric (String):** the metric to quantify the object
  1. Ex: lbs

xiii. **Quantity_Numeric (Double):** the number of Quantity_Metric of the product
  1. Ex: 2
     a. Meaning: 2 → 2 lbs of the product

xiv. **Position (String):** part of the room where the product is at
  1. Ex: Cabinet 2

xv. **Room_Number (Integer):** room number where the product is at
  1. Ex: 1800

2. Material

a. **Features:**

i. **Id (Integer):** auto-incremented generated value from database
  1. Automatically generated by DB

ii. **Material_Name (String):** name of the material

3. Test

a. **Features:**

i. **Id (Integer):** auto-incremented generated value from database
  1. Automatically generated by DB

ii. **Test_Name (String):** name of the test
  1. REQUIRED

iii. **Description (String):** description and additional info about the test
  1. Length: 1500 characters

iv. **Result (String):** result of that test
  1. Values: "pass", "fail", "ongoing", "see description"

v. **Test_Sheet (String):** the csv file of the test
  1. Ex: test1.csv

vi. **Instrument_Id (Integer):** the database id of instrument being used in the test
  1. Connects the Instrument table to the Test table because of the many-to-one relationship

4. Instrument

a. **Features:**
   i. **Id (Integer):** auto-incremented generated value from database
      1. Automatically generated by DB
   ii. **Instrument_Name (String):** name of the instrument
      1. REQUIRED
      2. **Ex: FTIR**
   iii. **Description (String):** description and additional info about the instrument
      1. Length: 1500 characters

5. ProdutMaterial

a. Relationship table that connects the Product Table to the Material Table because of the many-to-many relationship
b. **Features:**
   i. **Product_Id (Integer):** database id of the product
   ii. **Material_Id (Integer):** database id of the material

6. ProductTest

a. Relationship table that connects the Product Table to the Test Table because of the many-to-many relationship
b. **Features:**
   i. **Product_Id (Integer):** database id of the product
   ii. **Test_Id (Integer):** database id of the test

Table Relationships:

1. **Product to Material:** many-to-many relationship
2. **Product to Test:** many-to-many relationship
3. **Test to Instrument:** many-to-one relationship
   a. Instrument can be used in MANY test
   b. Test can use ONE instrument at most

MySQL Workbench:

1. Make sure to declare database:

```
CREATE DATABASE naraDB;
USE naraDB;
```

Common errors:

1. Create database table
    a. CREATE DATABASE naraDB;
2. Unable to access database in MySQL:
    a. USE naraDB;
3. Unable to update or remove from tables in MySQL Workbench:
    a. SET SQL_SAFE_UPDATES = 0;
    b. SET FOREIGN_KEY_CHECKS = 0;
        i. SET FOREIGN_KEY_CHECKS = 1;
            1. Run this line after you remove table

Additional Info:

1. Maybe add "tested by" column to Test table