AirbusAerothon5.0

IntoTheAir.

Presented On

MAY 13, 2023

Prepared By

PREET SAGAR

VANSH JOHARI

SAUMYA BHARTI

TANISHQ GUPTA

SHUBHAM CHAUDHARY

AGENDA

01 PROBLEM STATEMENT

02 PROPOSED SOLUTION

03 ADVANTAGES

04 METHODOLOGY

05 FUNCTIONAL PROTOTYPE

PROBLEM STATEMENT

- Create a data lake with a normalized DB to reduce redundancy.
- Identify the current redundant data.
- Create an automation process for data stamping(approval) the real-time data.
- Create a dashboard for the Users, and Data Officer

PROPOSED SOLUTION

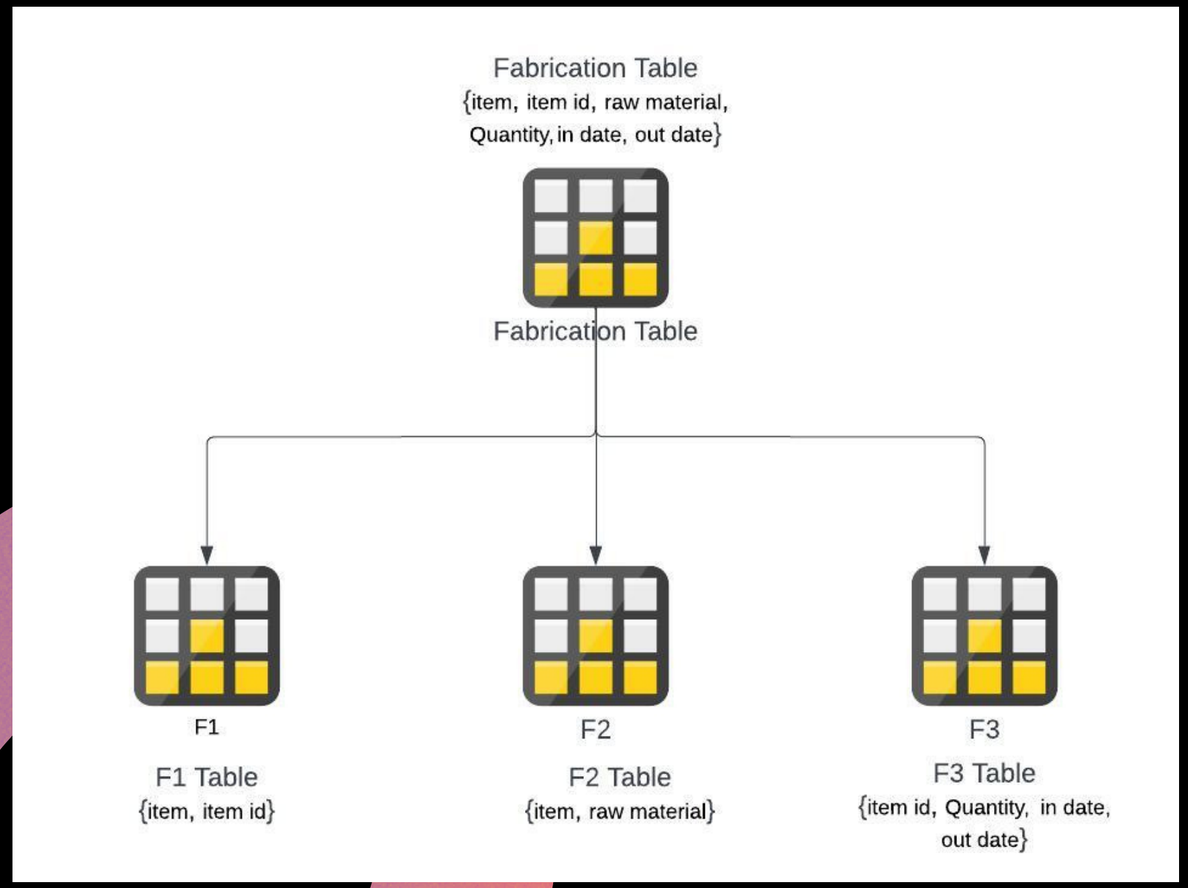
- Create a central data lake to store all data on AWS S3
- Use data profiling techniques to identify redundant data
- Automate data stamping using a workflow
- Workflow approves and timestamps real-time data when a production milestone is reached

ADVANTAGES

- Easy To Access
- Improved data quality
- Improved consistency
- Increased data security
- Improved data analysis capabilities
- Improved decision-making capabilities

METHODOLOGY

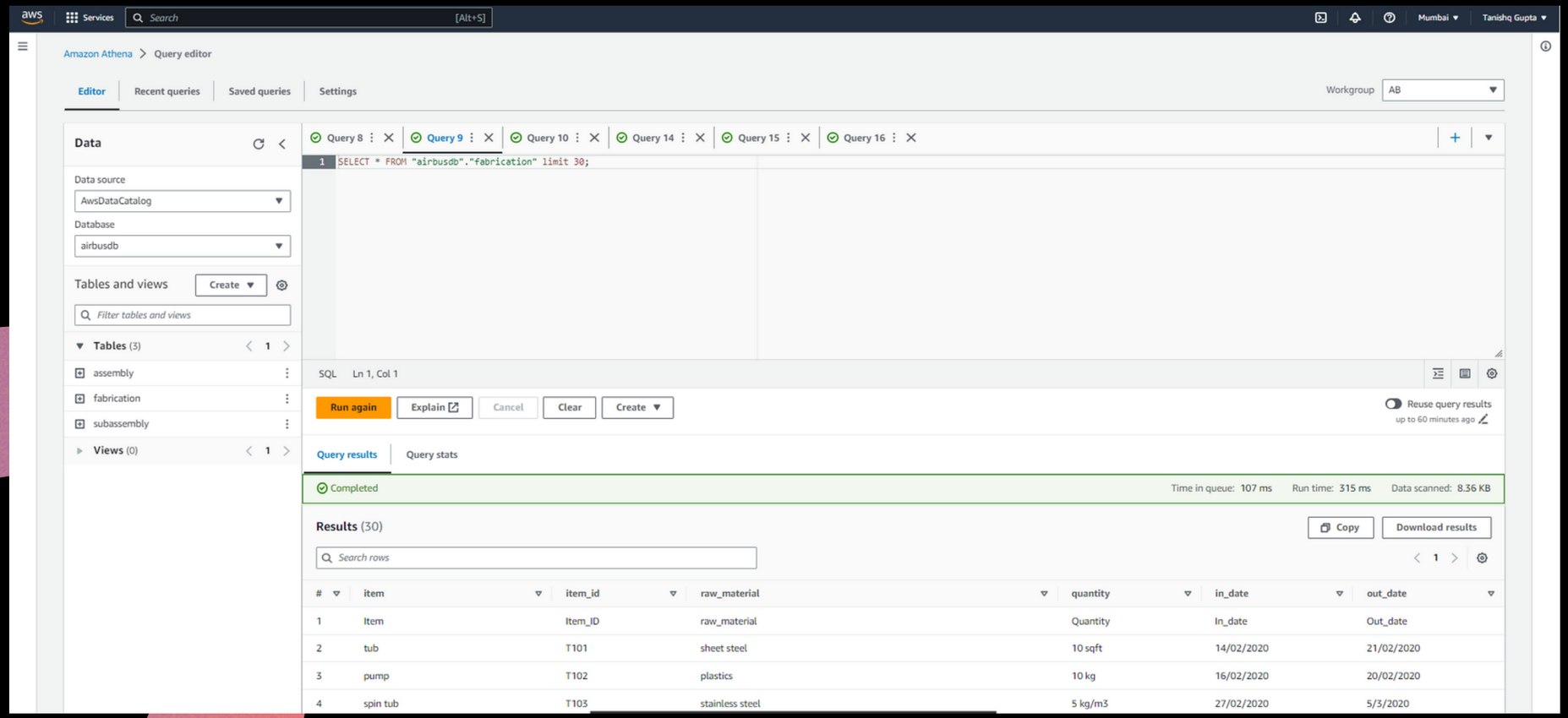
Normalize the table



- Find the redundancy in the dataset
- Normalize the table by dividing it into smaller Table
- Creating SQL Queries
- Uploading the Original Dataset into AWS S3
- Running the created queries on AWS Athena to generate a non-redundant dataset on AWS
- Create a User Interface
- Connecting the AWS S3 API with the FrontEnd

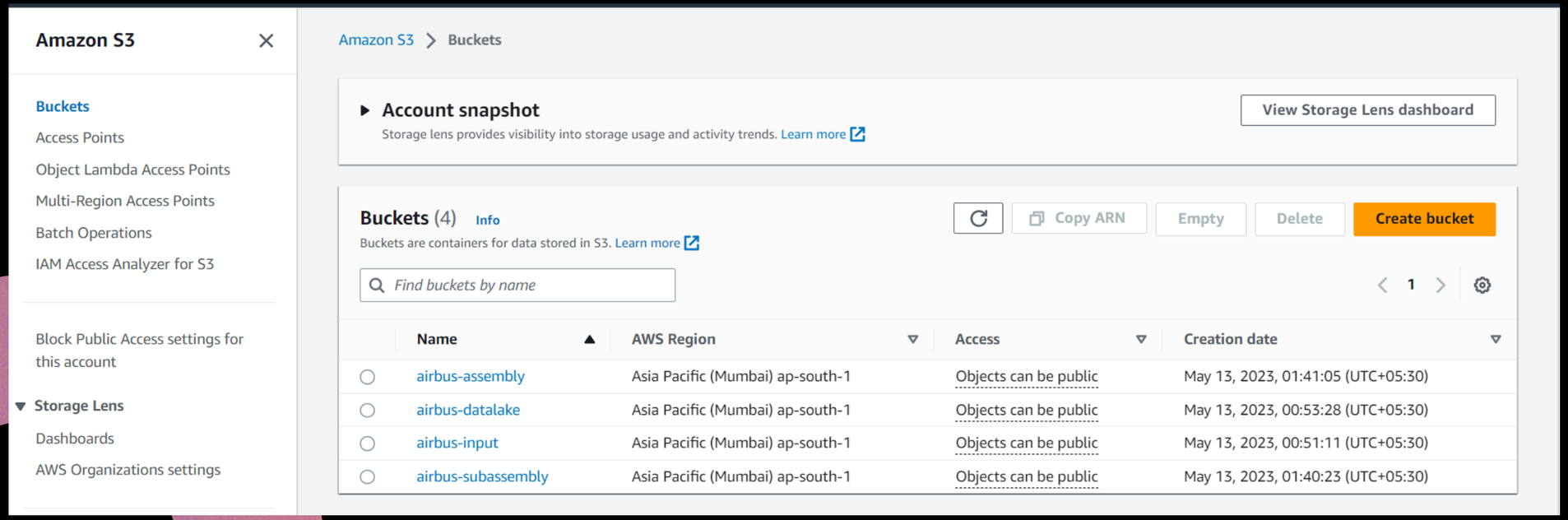
FUNCTIONAL PROTOTYPE

2.1 Running the Query on provided Dataset in AWS Athena



FUNCTIONAL PROTOTYPE

2.2 Creating a Normalized Database in AWS S3



FUNCTIONAL PROTOTYPE

2.3 Non Redundant Dataset in Bucket [airbus-datalake] created in AWS S3

○	airbus-datalake	Asia Pacific (Mumbai) ap-south-1	Objects can be public	May 13,

Objects (6)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions | Create folder | Upload

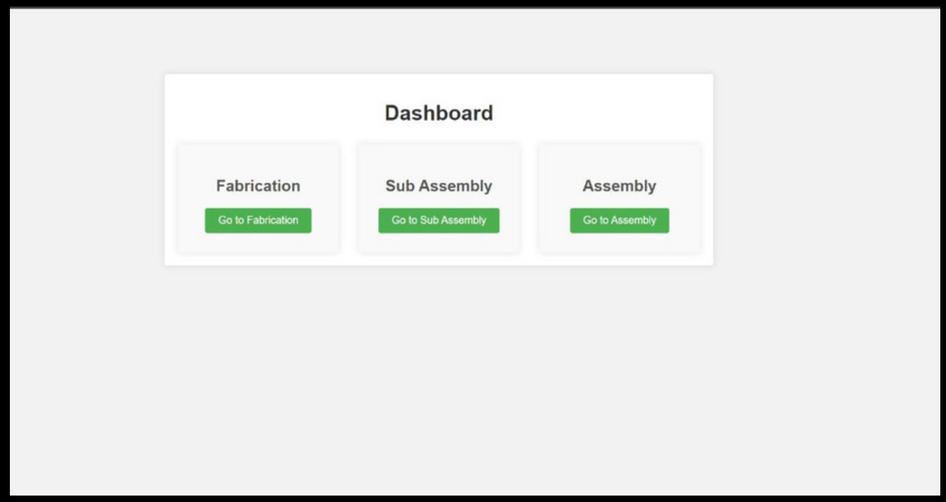
Find objects by prefix

1

Name	Type	Last modified	Size	Storage class
F1.csv	csv	May 13, 2023, 02:21:41 (UTC+05:30)	7.2 KB	Standard
F2.csv	csv	May 13, 2023, 02:21:42 (UTC+05:30)	13.5 KB	Standard
F3.csv	csv	May 13, 2023, 02:21:42 (UTC+05:30)	13.5 KB	Standard
assembly.csv	csv	May 13, 2023, 02:21:41 (UTC+05:30)	2.9 KB	Standard
fabrication_new.csv	csv	May 13, 2023, 02:21:42 (UTC+05:30)	2.2 KB	Standard
newsubassembly.csv	csv	May 13, 2023, 02:21:41 (UTC+05:30)	4.0 KB	Standard

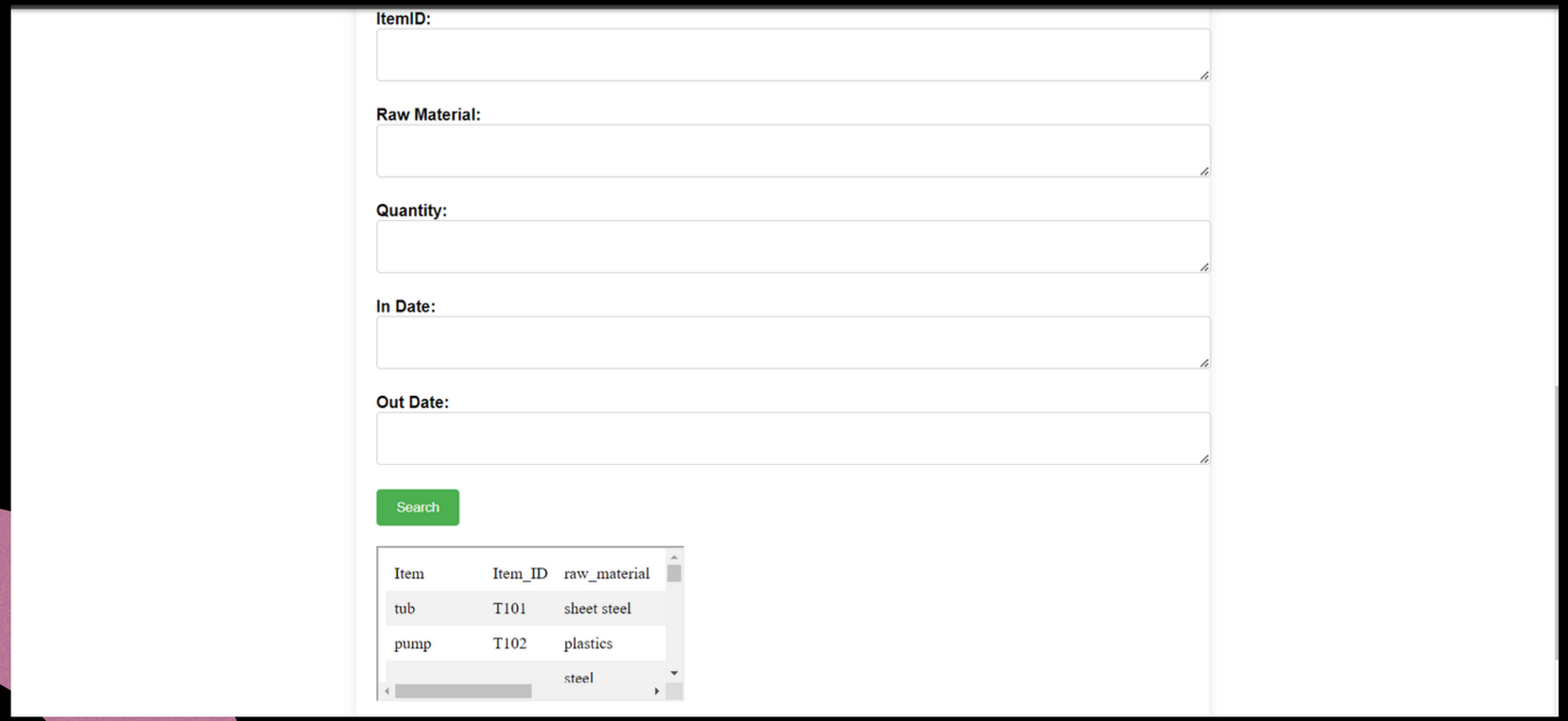
FUNCTIONAL PROTOTYPE

3. Designing the User Interface [Authentication & Authorisation Included]



FUNCTIONAL PROTOTYPE

3.1 Fetched Data on the Dashboard & Option to add Real-Time Data



DATA ANALYSIS

4. Analysis of the Provied Dataset using Python in Jupyter Notebook

jupyter IntoTheAir Last Checkpoint: 20 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [8]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
```

In [91]:
```
exf=pd.ExcelFile("data.xlsx")
df1=exf.parse("fabrication")
df1.describe()
```

Out[91]:

	item	item id	raw material	Quantity	in date	out date
count	28	28	28	28	28	28
unique	7	27	8	11	6	8
top	tub	T124	plastics	10 sqft	27/02/2020	2020-05-03 00:00:00
freq	5	2	7	5	8	4

In [26]:
```
df.head()
```

Out[26]:

	item	item id	raw material	Quantity	in date	out date
0	tub	T101	sheet steel	10 sqft	14/02/2020	21/02/2020
1	pump	T102	plastics	10 kg	16/02/2020	20/02/2020
2	spin tub	T103	stainless steel	5 kg/m3	27/02/2020	2020-05-03 00:00:00
3	wash tub	T104	(enameling iron) Porcelain coating	24 gauge	27/02/2020	2020-03-02 00:00:00
4	balance ring	T105	plastics	3 kg	16/03/2020	21/03/2020

In [36]:
```
df2=exf.parse("assembly")
df2.describe()
df2.head()
```

Out[36]:

	process	Process ID	Machine ID	2020-09-03 00:00:00	2020-11-30 00:00:00	Start Date	END DATE	2020-12-02 00:00:00	2021-02-02 00:00:00
0	Component Integration	SAWM3036_FA_WM128	MAII1017ECME3020079	NaN	NaN	2020-10-19	2020-12-11	NaN	NaN

DATA ANALYSIS

4.1 DATA CLEANING

0	Component Integration	SAWM3036_FA_WM128	MAII1017ECME3020079	NaN	NaN	2020-10-19	2020-12-11	NaN	NaN
1	Electrical Testing	SAWM3042_FA_WM134	MAII1017ECME3020080	NaN	NaN	2020-11-26	2020-12-05	NaN	NaN
2	Complaince	SAWM3042_FA_WM135	MAII1017ECME3020081	NaN	NaN	2020-09-10	2021-02-02	NaN	NaN
3	Certification Standards	SAWM3042_FA_WM136	MAII1017ECME3020082	NaN	NaN	2020-08-03	2021-01-06	NaN	NaN
4	pivot dome	SAWM3042_FA_WM137	MAII1017ECME3020083	NaN	NaN	2020-10-18	2021-01-05	NaN	NaN

In [37]:
```
print(df2.isnull().sum())
```

```
process                0
Process ID             0
Machine ID             0
2020-09-03 00:00:00    36
2020-11-30 00:00:00    36
Start Date             0
END DATE               0
2020-12-02 00:00:00    36
2021-02-02 00:00:00    36
dtype: int64
```

In [38]:
```
df2 = df2.dropna(axis=1)
df2.head()
```

Out[38]:

	process	Process ID	Machine ID	Start Date	END DATE
0	Component Integration	SAWM3036_FA_WM128	MAII1017ECME3020079	2020-10-19	2020-12-11
1	Electrical Testing	SAWM3042_FA_WM134	MAII1017ECME3020080	2020-11-26	2020-12-05
2	Complaince	SAWM3042_FA_WM135	MAII1017ECME3020081	2020-09-10	2021-02-02
3	Certification Standards	SAWM3042_FA_WM136	MAII1017ECME3020082	2020-08-03	2021-01-06
4	pivot dome	SAWM3042_FA_WM137	MAII1017ECME3020083	2020-10-18	2021-01-05

In [71]:
```
# Check for duplicate rows
duplicate_rows = df2.duplicated()
if duplicate_rows.any():
    print("There are duplicate rows in the dataframe")
else:
    print("There are no duplicate rows in the dataframe")
```

```
There are no duplicate rows in the dataframe
```

In [72]:
```
# Check for duplicate rows
duplicate_rows = df2.duplicated()
if duplicate_rows.any():
    print("There are duplicate rows in the dataframe")
else:
    print("There are no duplicate rows in the dataframe")
```

```
There are no duplicate rows in the dataframe
```

In [92]:
```
new_row = {'item': 'tub',
           'item id': 'T101',
           'raw material': 'sheet steel',
           'Quantity': '10 sqft',
           'in date': '14/02/2020',
           'out date':'21/02/2020'}
new_row
```

Out[92]:
```
{'item': 'tub',
 'item id': 'T101',
 'raw material': 'sheet steel',
 'Quantity': '10 sqft',
 'in date': '14/02/2020',
 'out date': '21/02/2020'}
```

DATA ANALYSIS

4.2 CLEANED DATA

In [94]:
```
if is_duplicate.any():
    print("New row is a duplicate row")
else:
    print("New row is not a duplicate row")
```

```
New row is a duplicate row
```

In [96]:
```
df1.drop_duplicates(inplace=True)
```

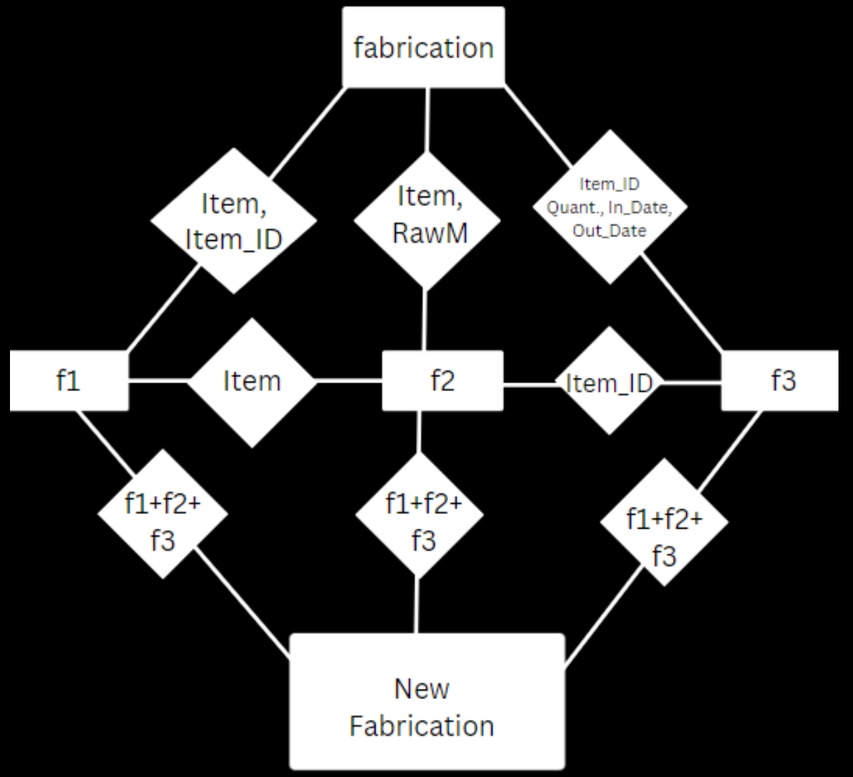
In [97]:
```
df1
```

Out[97]:

	item	item id	raw material	Quantity	in date	out date
0	tub	T101	sheet steel	10 sqft	14/02/2020	21/02/2020
1	pump	T102	plastics	10 kg	16/02/2020	20/02/2020
2	spin tub	T103	stainless steel	5 kg/m3	27/02/2020	2020-05-03 00:00:00
3	wash tub	T104	(enameling iron) Porcelain coating	24 gauge	27/02/2020	2020-03-02 00:00:00
4	balance ring	T105	plastics	3 kg	16/03/2020	21/03/2020
5	transmission gears	T107	cast aluminum	ingots—20	26/03/2020	2020-09-04 00:00:00
6	plastic brackets	T108	plastics	2 kg	2020-08-04 00:00:00	2020-12-04 00:00:00
7	tub	T109	sheet steel	10 sqft	14/02/2020	21/02/2020
8	pump	T110	plastics	10 kg	16/02/2020	20/02/2020
9	spin tub	T111	steel (enameling iron) Porcelain coating	5 kg/m3	27/02/2020	2020-05-03 00:00:00
10	wash tub	T112	(enameling iron) Porcelain coating	24 gauge	27/02/2020	2020-03-02 00:00:00
11	balance ring	T113	stainless steel	2 kg/m3	16/03/2020	21/03/2020
12	transmission gears	T115	cast aluminum	ingots—20	26/03/2020	2020-09-04 00:00:00
13	plastic brackets	T116	plastic	2 kg	2020-08-04 00:00:00	2020-12-04 00:00:00
14	tub	T124	sheet steel	10 sqft	14/02/2020	21/02/2020
15	pump	T117	plastics	10 kg	16/02/2020	20/02/2020
16	spin tub	T118	stainless steel	5 kg/m3	27/02/2020	2020-05-03 00:00:00
17	wash tub	T119	(enameling iron) Porcelain coating	24 gauge	27/02/2020	2020-03-02 00:00:00

ER-DIAGRAM

ER-DIAGRAM



*DataSource: Fabrication.csv

*DataLake:f1,f2,f3,New_Fabrication,Subassembly,Assembly

*NormalizedDB: New_Fabrication, Subassembly, Assembly

Thank you