

## MCIT AWS PDSA Intake 2 - Use Case A

Name: Ahmed Samir Alnaqa

Email: [ahmed.samir.mohamed@gmail.com](mailto:ahmed.samir.mohamed@gmail.com)

Group: 6

## Steps taken to solve this use case:

### 1- Evaluate the Datasets and check it's consistent:

Tool: Excel:

	A	B	C	D	E
	id	title	authors	venue	year
1	conf/vldb/RusinkiewiczKTWM95	Towards a Cooperative Transaction Model - The Cooperative Activity Mod	FALSE	Vldb	1995
3	journals/sigmod/EisenbergM02	SQL/XML is Making Good Progress	A Eisenberg, J Melton	SIGMOD Record	2002
4	conf/vldb/AmmannJR95	Using Formal Methods to Reason about Semantics-Based Decomposition	P Ammann, S Jajodia, I Ray	Vldb	1995
5	journals/sigmod/Liu02	Editor's Notes	L Liu	SIGMOD Record	2002
6	journals/sigmod/Hammer02	Report on the ACM Fourth International Workshop on Data Warehousing	N/A	N/A	2002
7	conf/vldb/FerrandinaMZFM95	Schema and Database Evolution in the O2 Object Database System	F Ferrandina, T Meyer, R Zicari, G Ferran, J Madec	Vldb	1995
8	conf/vldb/SubietaKL95	Procedures in Object-Oriented Query Languages	K Subieta, Y Kambayashi, J Leszczylowski	Vldb	1995
9	journals/sigmod/BargaL02	Phoenix Project: Fault-Tolerant Applications	R Barga, D Lomet	SIGMOD Record	2002
10	journals/sigmod/Ouksel02	Mining the World Wide Web: An Information Search Approach - Book Rev	N/A	N/A	2002
11	conf/vldb/MoserKK95	L/MRP: A Buffer Management Strategy for Interactive Continuous Data Fl	F Moser, A Kraiss, W Klas	Vldb	1995
12	journals/sigmod/Konig-RiesMMPPRSVW02	Report on the NSF Workshop on Building an Infrastructure for Mobile and	N/A	N/A	2002
13	conf/vldb/ChaudhuriGS95	Retrieval of Composite Multimedia Objects	S Chaudhuri, S Ghandeharizadeh, C Shahabi	Vldb	1995
14	journals/sigmod/Kosch02	MPEG-7 and Multimedia Database Systems	H Kosch	SIGMOD Record	2002
15	conf/vldb/LuTD95	The Fittest Survives: An Adaptive Approach to Query Optimization	H Lu, K Tan, S Dao	Vldb	1995
16	journals/sigmod/RossFS02	Reminiscences on Influential Papers	N/A	N/A	2002
17	conf/vldb/TreschPL95	Type Classification of Semi-Structured Documents	M Tresch, N Palmer, A Luniewski	Vldb	1995
18	journals/sigmod/Geller02	Data Mining: Practical Machine Learning Tools and Techniques - Book Rev	J Geller	SIGMOD Record	2002

I found one problem in columns count in Dataset 1 and changed to have the same column count.

	B	C	D	E	F	G	H
	title	authors	venue	year			
1	g database structure; or, how to build a data quality browser\''''',T	T Johnson	S Muthukrishnan	V Shkaper	SIGMOD Cor	2002	
531							
2620							
2621							
2622	g database structure; or, how to build a data quality browser\''''',T	T Johnson	V Shkapenyuk"	2002			
2618	g database strucrie; or, how to build a data quality browser\''''',T	T Johnson	SIGMOD Conference	2002			
2619							
2620							
2628							

### 2- Check the label file and convert it to match the datasets column count without duplicate:

Tool: Excel

Formula: Vlookup

	A	B	C	D	E	F	G	H	I
	label	id_dataset1	id_dataset2						
1	0	conf/sigmod/AbadiC02	f2Lea-RN8dsj	0	f2Lea-RN8dsj	Visual COKO: a debugger for query optimizer development	DJ Abadi	SIGMOD Conference,	2002
2	1	conf/sigmod/AbadiCCCCCEGHMRSSTXYZ03	eBnT7lhv2LwJ	1	eBnT7lhv2LwJ	Aurora: A Data Stream Management System (demo description)	D Abadi,	Proceedings of the 200	0
4	1	conf/sigmod/AbadiCCCCCEGHMRSSTXYZ03	gBVNSFeS4P8J	1	gBVNSFeS4P8J	Aurora: a new model and architecture for data stream manage	DJ Abadi,	The VLDB Journal The I	2003
5	1	conf/sigmod/AbadiCCCCCEGHMRSSTXYZ03	VuY9Y49GqXgJ	1	VuY9Y49GqXgJ	Aurora: A Data Stream Management System	DJ Abadi,		0
6	2	conf/sigmod/AbiteboulBCMM03	AxpQwgyRyLgJ	2	AxpQwgyRyLgJ	Active XML Documents with Distribution and Replication	S Abitebo	ACM SIGMOD,	0
7	2	conf/sigmod/AbiteboulBCMM03	Rjb06zlxblUJ	2	Rjb06zlxblUJ	Dynamic XML documents with distribution and replication	S Abitebo	SIGMOD Conference,	2003
8	3	conf/sigmod/AbiteboulCM95	4GTOKrd9RP0J	3	4GTOKrd9RP0J	A database interface for le update	S Abitebo	SIGMOD,	0
9	3	conf/sigmod/AbiteboulCM95	cu9DXtjeF24J	3	cu9DXtjeF24J	A database interface for file update - group of 2 &raquo;quo	S Abitebo	Proceedings of the 199	1995
10	3	conf/sigmod/AbiteboulCM95	DakOA4Ew-poJ	3	DakOA4Ew-poJ	A Database Interface for Files Update. To appear:	S Abitebo	Proc. ACM SIGMOD Int	0
11	4	conf/sigmod/AboulnagaC99	WWaxLMlptTMJ	4	WWaxLMlptTMJ	Self-tuning Histograms: Building Histograms Without Looking a	A Abouln	SIGMOD Conference,	1999
12	4	conf/sigmod/AboulnagaC99	xnDzelm2t1QJ	4	xnDzelm2t1QJ	Self-tuning Histograms: Building Histograms Without Looking a	AA AC, S	( Proceedings of the ACM	0
13	5	conf/sigmod/AcharyaAFZ95	_yXA5HLnqoJ	5	_yXA5HLnqoJ	Broadcast disks: Data management for asymmetric communica	SA VI, R	A Proceedings of the ACM	0

3- Concat the two datasets in one Dataset and include the source column to identify the records, in addition change the “/” char to “\_” then convert it to Json file.

Tool: google colab notebook  
Code: Pandas

```
[ ] import pandas as pd

df1 = pd.read_csv('dataset1-2.csv')
df2 = pd.read_csv('dataset2.csv')
#print(df.to_string())

[ ] # Add column for each DS with a static value
df1['Source'] = 'DS1'
df2['Source'] = 'DS2'
#df1.head()

[ ] # add the two DS's to just one DS
df=pd.concat([df1,f2])

# Remove the "/" and replace it with "_" in the column "id"
hh= df['id'].str.replace(r'^0-9a-zA-Z:,\s+', '_', regex=True)
df['id'] = hh

df.head()

[ ] # create the Final file
df.to_json('OneFileFinal.json', orient='records',lines=True )
```

4- replace the “/” char to “\_” in the label file and add the header to it:  
Tool: google colab notebook  
Code: Pandas

```
labels final

[ ] labl=pd.read_csv('labelsFinal.csv')
#Ds=pd.read_json('OneFileFinal.json')

#Ds.head()
labl.columns = ['label','id', 'title', 'authors', 'venue', 'year', 'Source']

hh= labl['id'].str.replace(r'^0-9a-zA-Z:,\s+', '_', regex=True)
labl['id'] = hh

labl.head()

labl.to_json('labelsFinal_linesTrue.json',orient='records',lines=True)
```

5- upload the files to S3:

▼ Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight ⓘ

▶ AWS Marketplace for S3

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	AthenaOutput/	Folder	-	-	-
<input type="checkbox"/>	Generatedlabel/	Folder	-	-	-
<input type="checkbox"/>	JobScript/	Folder	-	-	-
<input type="checkbox"/>	OutPut/	Folder	-	-	-
<input type="checkbox"/>	Raw/	Folder	-	-	-

Block Public Access settings for this account

Storage Lens

Dashboards

AWS Organizations settings

Feature spotlight 3

AWS Marketplace for S3

Find objects by prefix

	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	labels_AWS.csv	csv	April 10, 2022, 10:52:09 (UTC+02:00)	74.0 KB	Standard
<input type="checkbox"/>	labelsFinal_linesTrue.json	json	April 10, 2022, 10:36:03 (UTC+02:00)	100.3 KB	Standard
<input type="checkbox"/>	labelsFinal.json	json	April 10, 2022, 10:01:57 (UTC+02:00)	100.3 KB	Standard
<input type="checkbox"/>	OLD/	Folder	-	-	-
<input type="checkbox"/>	OneFileFinal_linesTrue.json	json	April 10, 2022, 10:25:54 (UTC+02:00)	12.6 MB	Standard
<input type="checkbox"/>	OneFileFinal.json	json	April 10, 2022, 10:01:30 (UTC+02:00)	12.6 MB	Standard

6- start crawler step:

aws

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. California

Ahmed Samir

S3

AWS Lake Formation

AWS Glue

AWS Glue DataBrew

Lambda

Athena

IAM

EC2

AWS Glue

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

ETL

Crawlers

A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Add crawler

Run crawler

Action

Filter by tags and attributes

Showing: 1 - 1

	Name	Schedule	Status	Logs	Last runtime	Median runtime	Tables updated	Tables added
<input type="checkbox"/>	MatchingFuzzyDataFinal		Ready	Logs	45 secs	45 secs	0	1

7- ML transform step:

aws

Services

Search for services, features, blogs, docs, and more

[Alt+S]

N. California

Ahmed Samir

S3

AWS Lake Formation

AWS Glue

AWS Glue DataBrew

Lambda

Athena

IAM

EC2

Data catalog

Databases

Tables

Connections

Crawlers

Classifiers

Schema registries

Schemas

Settings

ETL

AWS Glue Studio

Jobs - New

Jobs (legacy)

ML Transforms

Blueprints

Workflows

Triggers

Dev endpoints

Notebooks

Machine learning transforms

Clean your data using machine learning transforms.

Add transform

Action

Filter by tags and attributes

Showing: 1 - 1

Transform name	Transform ID	Type	Label count	Status	Date created	Last modified	Description
MatchingFuzzyData...	tfm-865f8d7d028da56d0...	Find matching records	469	Ready for use	10 April 2022 10:32 ...	10 April 2022 10:55 ...	

History

Details

Estimate quality

Cancel

Showing: 1 - 5

Run ID	Task type	Status	Error	Start time	Execution time	Last modified	Logs	Error logs	Download label file
tsk-fd426752f...	ETL Job	Succeeded		10 April 2022 1...	7 mins	10 April 2022 1...			
tsk-d00a0c76...	Quality estimation	Succeeded		10 April 2022 1...	7 mins	10 April 2022 1...	Logs		
tsk-f61ccb468...	Uploading labels	Succeeded		10 April 2022 1...	0 secs	10 April 2022 1...			
tsk-d140ef4a8...	Generating labeling file	Failed	Exception i...	10 April 2022 1...	5 mins	10 April 2022 1...	Logs	Error logs	
tsk-d15204ce...	Uploading labels	Failed	java.io.IOE...	10 April 2022 1...	0 secs	10 April 2022 1...			

**AWS** Services Search for services, features, blogs, docs, and more [Alt+S]

S3 AWS Lake Formation Glue AWS Glue DataBrew Lambda Athena IAM EC2

## AWS Glue Teach transform

- Generate labeling file
- Label data
- Upload labels
- Estimate quality

### Estimate quality metrics (optional)

Estimate your transform's ability to find matches. Estimates are calculated by comparing the transform match predictions using a subset of your labeled data against the labels you have provided. These estimates are approximate. To improve your transform quality, provide more labels.

New estimates start a task that you can monitor in the **History** pane of the transform. When the task completes, new estimates can be viewed in the **Quality metrics** pane of the transform.

[Estimate transform quality](#)

Quality metric	Definition	Result	Last modified
Area under the Precision-Recall curve	Single number summarizing the performance of the transform	99.1612%	04/10/22 11:04 AM
Precision	When your transform predicts a match, how often is it correct?	100%	04/10/22 11:04 AM
Recall upper limit	For an actual match, how often does your transform predict a match?	78.2609%	04/10/22 11:04 AM
F1	Indicates transform's accuracy. Harmonic mean of Precision and Recall.	87.8049%	04/10/22 11:04 AM

\* Metrics shown are from the last quality estimation run.  
 \*\* End-to-End recall will tend to be closer to the upper limit as the cost-accuracy slider favors accuracy. See documentation for additional information about End-to-End recall.

Want to improve your results? [Generate a new labeling file](#), label it, and upload the labels to append to our

### 8- Create and run the glue job:

MatchingFuzzyDataFinal

Job has not been saved Save Delete Run

Script Job details **Runs** Schedules

Recent job runs (3) Info

April 10, 2022 11:49:51 AM

Job name	Id	Run status	Glue version
MatchingFuzzyDataFinal	jr_52973d73dc3fb07662cb322c25bedcbdcdfc4af84ea957631b1e6ad43d6a1f9	✔ Succeeded	2.0
Retry attempt number	Start time	End time	Start-up time
Initial run	April 10, 2022 11:49:51 AM	April 10, 2022 11:58:33 AM	7 seconds
Execution time	Last modified on	Trigger name	Security configuration
8 minutes 34 seconds	April 10, 2022 11:58:33 AM	-	-
Timeout	Max capacity	Number of workers	Worker type
2880 minutes	20 DPUs	10	G.2X
Execution class	Log group name	Cloudwatch logs	Performance and debugging recommendations
-	/aws-glue/jobs	<ul style="list-style-type: none"> <li>All logs</li> <li>Output logs</li> <li>Error logs</li> </ul>	<ul style="list-style-type: none"> <li>View in CloudWatch</li> </ul>

9- check the output files:

Objects (160)

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

Name	Type	Last modified	Size	Storage class
run-1649584689264-part-r-00061	-	April 10, 2022, 11:58:17 (UTC+02:00)	60.4 KB	Standard
run-1649584689264-part-r-00062	-	April 10, 2022, 11:58:17 (UTC+02:00)	60.6 KB	Standard
run-1649584689264-part-r-00063	-	April 10, 2022, 11:58:17 (UTC+02:00)	60.8 KB	Standard
run-1649584689264-part-r-00064	-	April 10, 2022, 11:58:17 (UTC+02:00)	53.6 KB	Standard
run-1649584689264-part-r-00065	-	April 10, 2022, 11:58:17 (UTC+02:00)	51.6 KB	Standard
run-1649584689264-part-r-00066	-	April 10, 2022, 11:58:17 (UTC+02:00)	62.0 KB	Standard
run-1649584689264-part-r-00067	-	April 10, 2022, 11:58:17 (UTC+02:00)	60.6 KB	Standard
run-1649584689264-part-r-00068	-	April 10, 2022, 11:58:17 (UTC+02:00)	60.1 KB	Standard
run-1649584689264-part-r-00069	-	April 10, 2022, 11:58:17 (UTC+02:00)	61.8 KB	Standard
run-1649584689264-part-r-00070	-	April 10, 2022, 11:58:17 (UTC+02:00)	61.9 KB	Standard
run-1649584689264-part-r-00071	-	April 10, 2022, 11:58:17 (UTC+02:00)	63.0 KB	Standard
run-1649584689264-part-r-00072	-	April 10, 2022, 11:58:17 (UTC+02:00)	61.3 KB	Standard
run-1649584689264-part-r-00073	-	April 10, 2022, 11:58:17 (UTC+02:00)	62.9 KB	Standard
run-1649584689264-part-r-00074	-	April 10, 2022, 11:58:17 (UTC+02:00)	56.6 KB	Standard

10- Download the 160 part:

Tool: google colab notebook

Code: Python (Boto)

```
[ ] import boto3

# Let's use Amazon S3
s3 = boto3.resource('s3')

[ ] s3 = boto3.resource(
    's3',
    aws_access_key_id='[REDACTED]',
    aws_secret_access_key='[REDACTED]'
)

[ ] for bucket in s3.buckets.all():
    print(bucket.name)

#s3.download_file('tbhandson', 'OutPut/run-1649584689264-part-r-00010', 'S3/GG2.CSV')

s3.download_file('tbhandson', 'OutPut/run-1649584689264-part-r-00000', 'S3/run-1649584689264-part-r-00000.csv')
s3.download_file('tbhandson', 'OutPut/run-1649584689264-part-r-00001', 'S3/run-1649584689264-part-r-00001.csv')
s3.download_file('tbhandson', 'OutPut/run-1649584689264-part-r-00002', 'S3/run-1649584689264-part-r-00002.csv')
s3.download_file('tbhandson', 'OutPut/run-1649584689264-part-r-00003', 'S3/run-1649584689264-part-r-00003.csv')
```

0s completed at 4:44 PM

11- merge the parts:

Tool: google colab notebook

Code: Python (Pandas)

```
import os
import glob
import pandas as pd

os.chdir("S3")

extension = 'csv'
all_filenames = [i for i in glob.glob('*.{}'.format(extension))]

#combine all files in the list
combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])

#export to csv
combined_csv.to_csv( "result.csv", index=False, encoding='utf-8-sig')
```