Data Pipeline with Quality

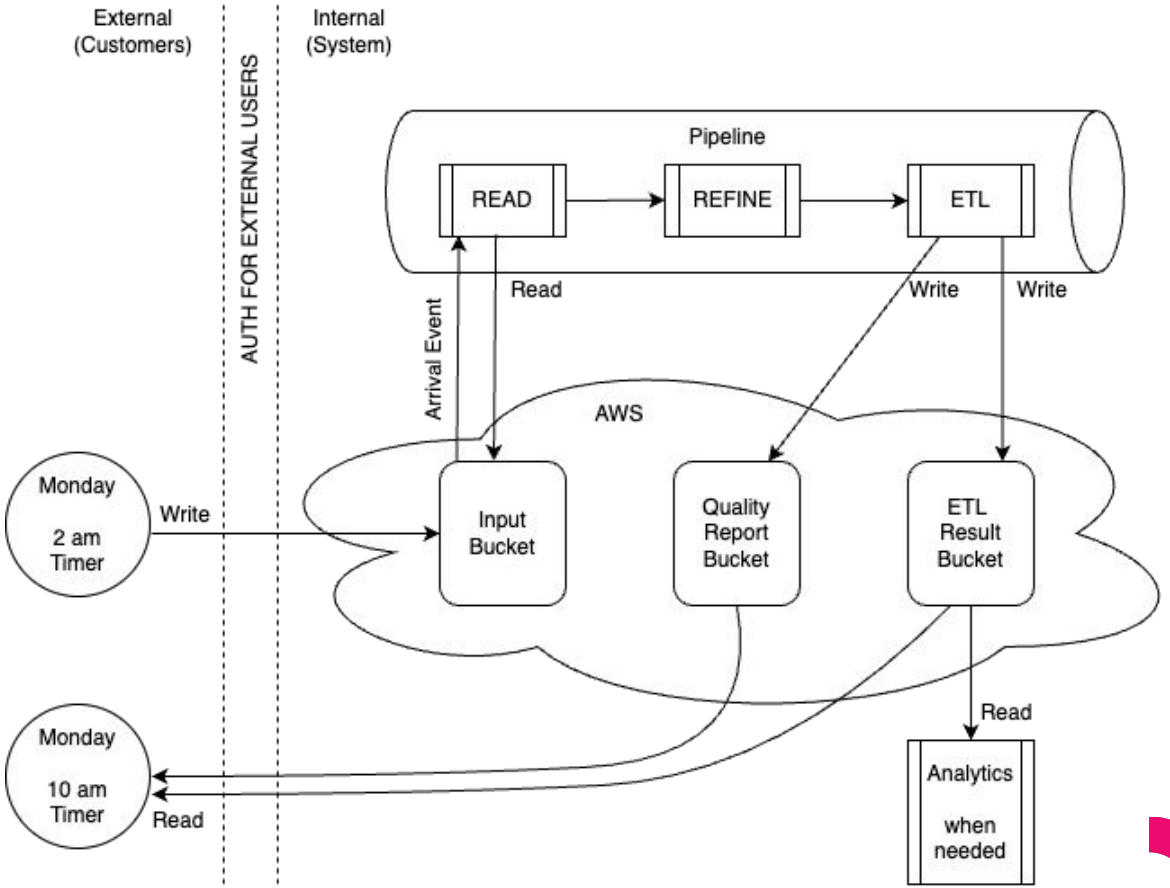
Presented to: Tendo Staff

Presented by: Susan Korgen

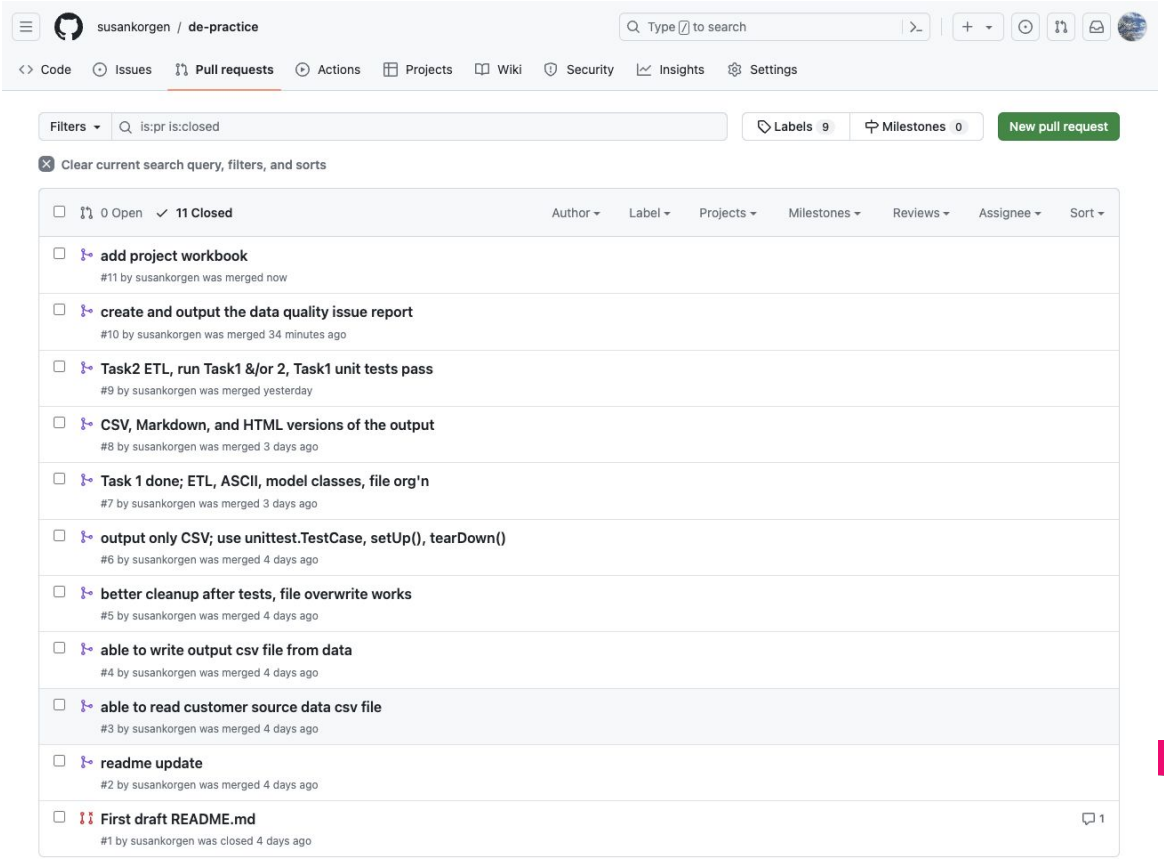
May 2, 2024

tendo

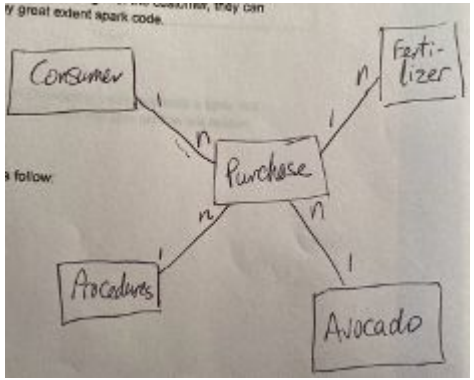
Pipeline.


External
(Customers)
AUTH FOR EXTERNAL USERS
Internal
(System)
Pipeline
READ
REFINE
ETL
Read
Write
Write
Arrival Event
AWS
Monday
2 am
Timer
Write
Input
Bucket
Quality
Report
Bucket
ETL
Result
Bucket
Read
Monday
10 am
Timer
Read
Analytics
when
needed

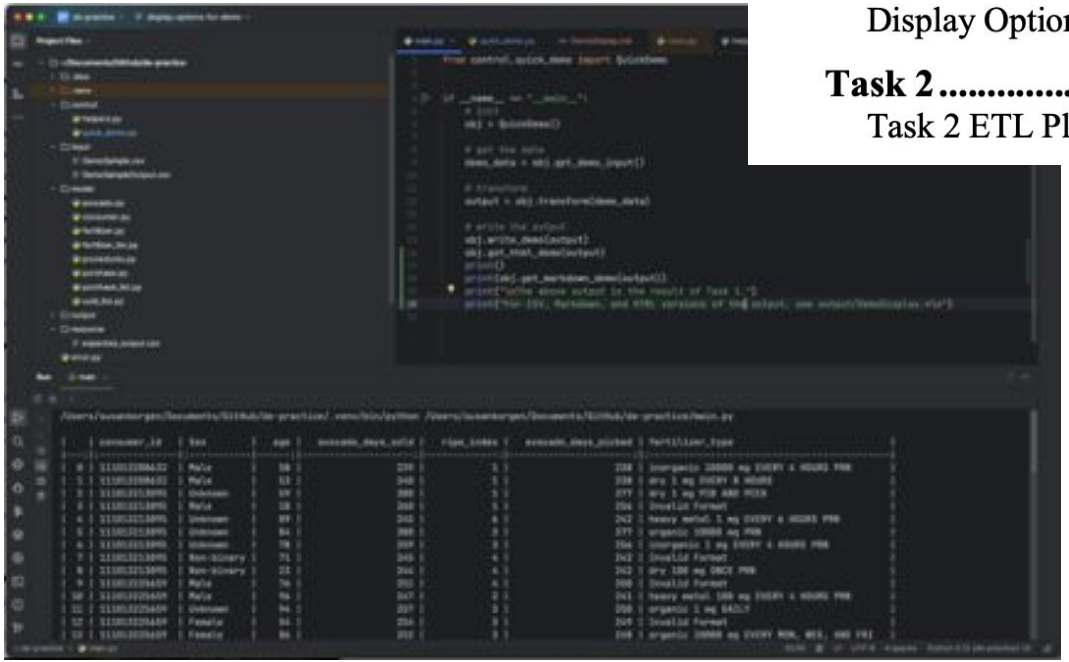
Repo.


susankorgen / de-practice
Type / to search
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
Filters
is:pr is:closed
Labels 9
Milestones 0
New pull request
Clear current search query, filters, and sorts
0 Open
11 Closed
Author
Label
Projects
Milestones
Reviews
Assignee
Sort
add project workbook
#11 by susankorgen was merged now
create and output the data quality issue report
#10 by susankorgen was merged 34 minutes ago
Task2 ETL, run Task1 &/or 2, Task1 unit tests pass
#9 by susankorgen was merged yesterday
CSV, Markdown, and HTML versions of the output
#8 by susankorgen was merged 3 days ago
Task 1 done; ETL, ASCII, model classes, file org'n
#7 by susankorgen was merged 3 days ago
output only CSV; use unittest.TestCase, setUp(), tearDown()
#6 by susankorgen was merged 4 days ago
better cleanup after tests, file overwrite works
#5 by susankorgen was merged 4 days ago
able to write output csv file from data
#4 by susankorgen was merged 4 days ago
able to read customer source data csv file
#3 by susankorgen was merged 4 days ago
readme update
#2 by susankorgen was merged 4 days ago
First draft README.md
1
#1 by susankorgen was closed 4 days ago

Day 1.



Day 2.



Day 3, exploration.

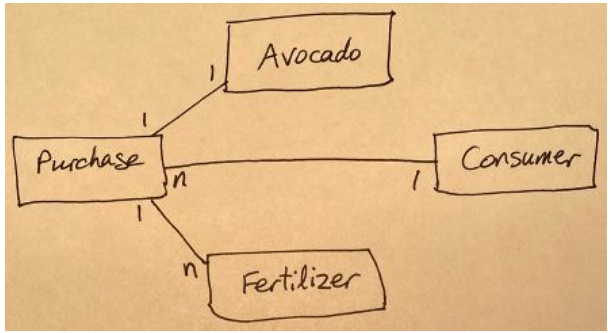


Table Fertilizer:

fertilizerid	NOT NULL PK	
consumerid	NOT NULL FK	→ consumer
purchaseid	NOT NULL FK	→ purchase

Table Purchase:

purchaseid	NOT NULL PK	
consumerid	NOT NULL FK	→ consumer
avocado_bunch_id	NOT NULL FK	→ avocado

Table Avocado:

avocado_bunch_id	NOT NULL PK	
purchaseid	NOT NULL FK	→ purchase
consumerid	NOT NULL FK	→ consumer

Table Consumer:

consumerid (allvalues bad)	NOT NULL PK	

Day 3 .....

Scrum .....

Task 2 .....

Real-World Data .....

Data Quality Issues .....

Data Exploration can reveal DATA DESIGN Quality Issues .....

Pivot Plan for ETL .....

ETL Daily Runs can reveal DATA VALUE Quality Issues .....

Day 3, quality.

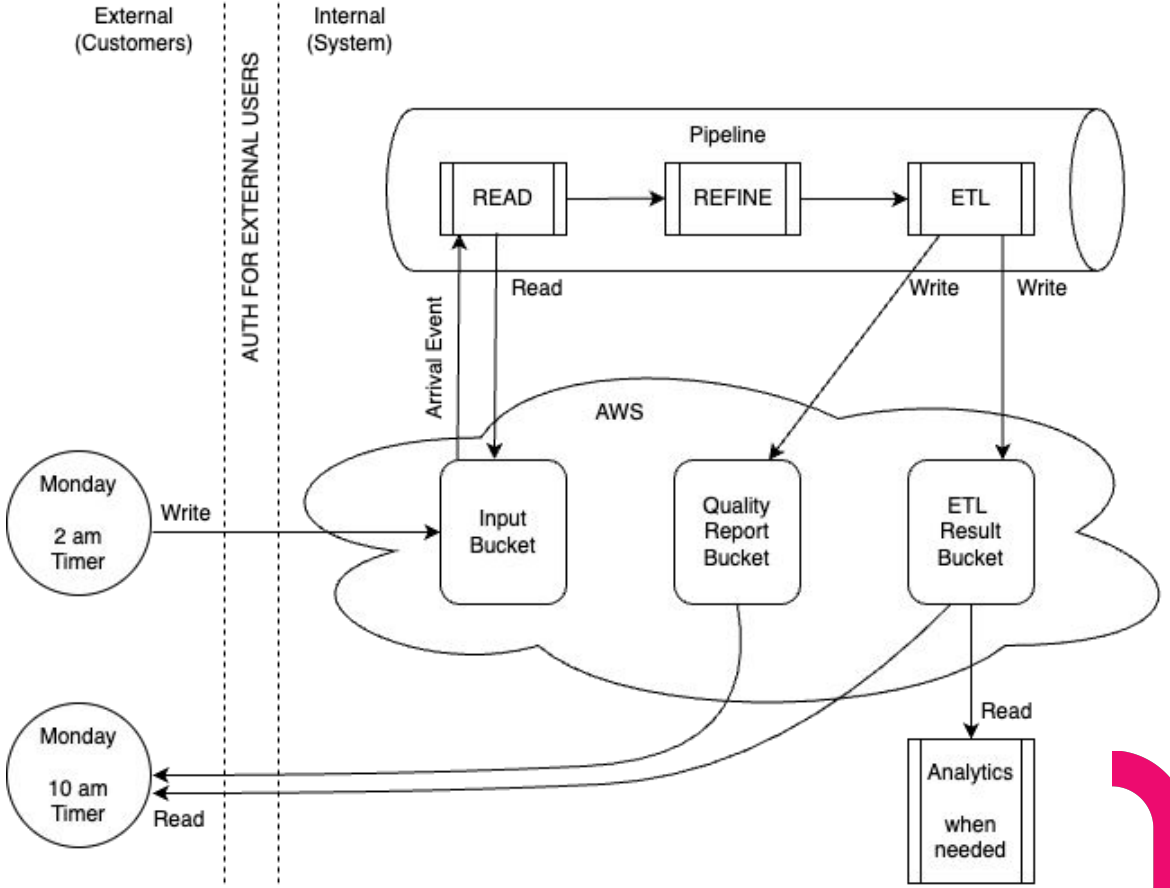
Data value quality issues are discoverable during daily processing. From most to least severe:

1. BLOCKER – for the exercise, will report and work around; in real life these must be fixed:
   a. PK is not unique in table (consumerid) – (for demo: mock values)
   b. Missing value for a field that is NOT NULLABLE (PKs and FKs) – (for demo: skip row)
2. MISSING – (BLOCKER is also reported for these) Missing PK or FK for our scenario:
   a. **consumer.consumerid**
   b. **fertilizer.purchaseid**
   c. **fertilizer.consumerid**
3. MISSING – Missing value for a data field needed for our scenario, demo substitutes:
   a. **consumer.Sex** (for demo: random value)
   b. **consumer.age** (for demo: random value)
   c. **avocado.sold_date** (for demo: make today)
   d. **avocado.born_date** (for demo: make today)
   e. **avocado."ripe index when picked"** (for demo: make 0)
   f. **avocado.picked_date** (for demo: make today)
4. DATATYPE – will flag and discuss with customer; for demo, will try to adjust data type
   a. Inconsistent date format values (for demo: strip time from date strings)
   b. Inappropriate data type for identifier (float for any "id", should be int or str)
   c. Date value that is non-empty but not parseable as a date. (for demo: empty)
5. RANGE – will flag; can help customer decide on path forward; for demo, will adjust
   a. Super old dates (for demo: set to 365)
   b. Negative ages (for demo: make 0)
   c. Negative numbers of days (for demo: make 0)
   d. Negative ripe index (for demo: make 0)
   e. Ripe index > 10 or < 0 (for demo: adjust to 10 or 0 respectively)
   f. Dates in the future as of today (for demo: make today)
6. UNEXPECTED – flag it; can help customer decide on path forward; for demo, adjust
   a. same FK value in 95-100% of rows (consumerid) (for demo: random value)
   b. non-ASCII char in input but ASCII output is required (for demo: "Invalid")
   c. NaN in a float (for demo: 0.0)
   d. NaN in an int (for demo: 0)
7. NONSTANDARD – will flag; could discuss with customer, but often they can't change
   a. Spaces in a field name: avocado."ripe index when picked"
   b. Capitalized column names: Sex, Race Age in consumer.csv

Day 4, quality.

```
data_quality_issue_store: dict = dict(
    {
        "UNKNOWN": ["UNKNOWN", 0.0, "Invalid issue key in issue report", "ignore"],
        "BLOCKER-1a": ["BLOCKER-1a", 50.0, "PK is not unique in table", "mock values"],
        "BLOCKER-1b": ["BLOCKER-1b", 40.0, "Missing value for a field that is NOT NULLABLE (PKs and
        "MISSING-2a": ["MISSING-2a", 28.0, "Missing value for a field that is NOT NULLABLE (PKs and
        "MISSING-3a": ["MISSING-3a", 25.0, "Missing value for a field required for demo", "choose r
        "MISSING-3b": ["MISSING-3b", 23.0, "Missing value for a date field needed for demo", "use t
        "MISSING-3c": ["MISSING-3c", 20.0, "Missing value for a number needed for demo", "use 0"],
        "DATATYPE-4a": ["DATATYPE-4a", 15.0, "Inconsistent date format values", "strip time off the
        "DATATYPE-4b": ["DATATYPE-4b", 13.0, "Inappropriate data type for identifier (float)", "con
        "DATATYPE-4c": ["DATATYPE-4c", 10.0, "Date value that is non-empty but not parseable as a d
        "RANGE-5a": ["RANGE-5a", 8.0, "Value too low, out of range", "set to minimum in range"],
        "RANGE-5b": ["RANGE-5b", 7.5, "Value too high, out of range", "set to maximum in range"],
        "RANGE-5c": ["RANGE-5c", 7.0, "Dates in the future as of today", "set to today"],
        "UNEXPECTED-6a": ["UNEXPECTED-6a", 5.0, "Same PK or FK value in 95-100% of rows", "mock val
        "UNEXPECTED-6b": ["UNEXPECTED-6b", 4.75, "Non-ASCII char in str but ASCII output is require
        "UNEXPECTED-6c": ["UNEXPECTED-6c", 4.50, "NaN in a float", "set to 0.0"],
        "UNEXPECTED-6z": ["UNEXPECTED-6z", 4.50, "NaN in an int", "set to 0"],
        "UNEXPECTED-6d": ["UNEXPECTED-6d", 4.25, "Empty number value, could not cast to str", "use
        "UNEXPECTED-6e": ["UNEXPECTED-6e", 4.12, "Invalid number value, could not cast to str", "us
        "UNEXPECTED-6f": ["UNEXPECTED-6f", 4.0, "Empty string where a date should be", "may use emp
        "UNEXPECTED-6j": ["UNEXPECTED-6j", 3.14159, "Large integer where an int should be", "cast a
        "UNEXPECTED-6k": ["UNEXPECTED-6k", 2.71828, "Unexpected type (not str, int, or float) where
        "UNEXPECTED-6g": ["UNEXPECTED-6g", 0.83, "Invalid str where a date should be", "may use emp
        "UNEXPECTED-6h": ["UNEXPECTED-6h", 0.47, "str where an int should be", "cast as int"],
        "UNEXPECTED-6i": ["UNEXPECTED-6i", 0.39, "float where an int should be", "cast as int"],
        "NONSTANDARD-7a": ["NONSTANDARD-7a", 0.2, "Spaces in a field name", ""],
        "NONSTANDARD-7b": ["NONSTANDARD-7b", 0.1, "Capitalized column name", ""],
    }
)
```

Day 4, design.


External
(Customers)
Internal
(System)
AUTH FOR EXTERNAL USERS
Pipeline
READ
REFINE
ETL
Read
Write
Write
Arrival Event
AWS
Monday
2 am
Timer
Write
Input
Bucket
Quality
Report
Bucket
ETL
Result
Bucket
Read
Monday
10 am
Timer
Read
Analytics
when
needed

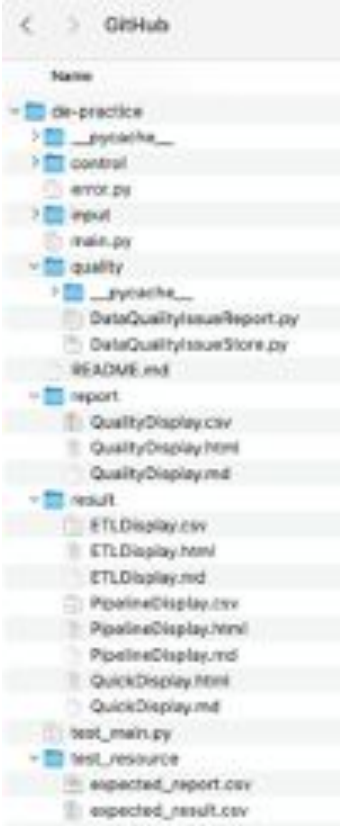
Day 4, report.

	name	severity	description	demo_workaround	pipeline_segment	table_name	field_name
558	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
562	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
566	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
569	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
573	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
577	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
581	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
585	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
588	BLOCKER-1a	50.00	PK is not unique in table	mock values	transform	consumer	consumerid
559	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
563	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
567	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
570	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
574	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
578	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
582	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
586	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
589	MISSING-3a	25.00	Missing value for a field required for demo	choose random row from real data	transform	consumer	consumerid
572	RANGE-5a	8.00	Value too low, out of range	set to minimum in range	transform	consumer	Age
580	RANGE-5a	8.00	Value too low, out of range	set to minimum in range	transform	consumer	Age
561	RANGE-5b	7.50	Value too high, out of range	set to maximum in range	transform	consumer	Age
565	RANGE-5b	7.50	Value too high, out of range	set to maximum in range	transform	consumer	Age
576	RANGE-5b	7.50	Value too high, out of range	set to maximum in range	transform	consumer	Age
584	RANGE-5b	7.50	Value too high, out of range	set to maximum in range	transform	consumer	Age
557	UNEXPECTED-6b	4.75	Non-ASCII char in str but ASCII output is required	use 'other'	refine_input	fertilizer	type
1	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
2	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
3	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
4	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
5	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
6	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
7	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	consumerid
8	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
9	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
10	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
11	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
12	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
13	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
14	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	consumer	Age
15	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	purchase	purchaseid
16	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	purchase	purchaseid
17	UNEXPECTED-6i	0.39	float where an int should be	cast as int	refine_input	purchase	purchaseid

Day 4, buckets.

Using the local file system as if folders represent buckets. Not shown: we must write each different customer's output to different locations to protect against write collisions and privacy leaks.

1. Pipeline Output is in **result**
2. Data Quality Output in **report**



Days 1-4, repo.

https://github.com/susankorgen/de-practice

Questions?

Some topics:

- Stakeholder communication
- Agile for project management
- Coding best practices
- Real-world data
- In-real-life moments
- Demo (Task 1) vs. product (Task 2)
- Extending with new features

Ongoing innovation with Tendo.