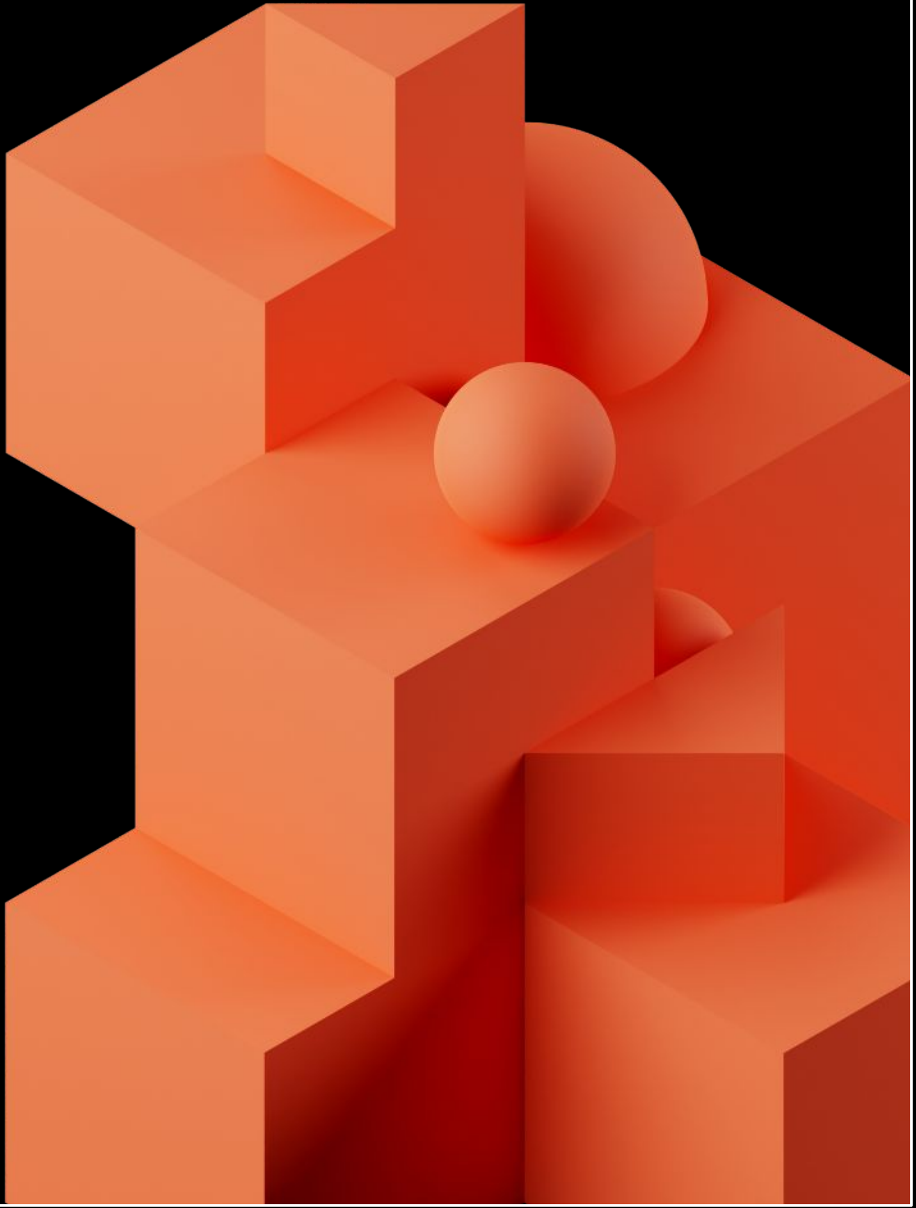
DATA+AI SUMMIT
BY databricks

DiscoverX

Map your Lakehouse Content

Erni Durdevic, David Tempelmann
2023



What’s in my Lakehouse?

What’s in my Lakehouse?

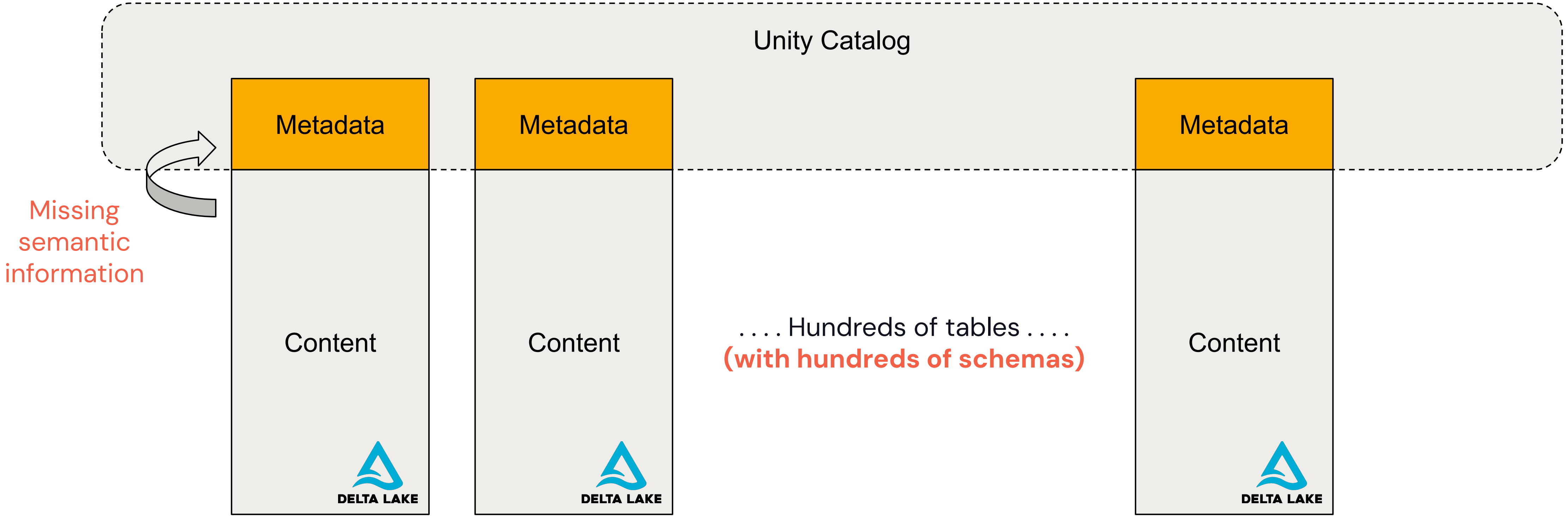
How many columns contain **emails**?

Is there any **SSN** number?

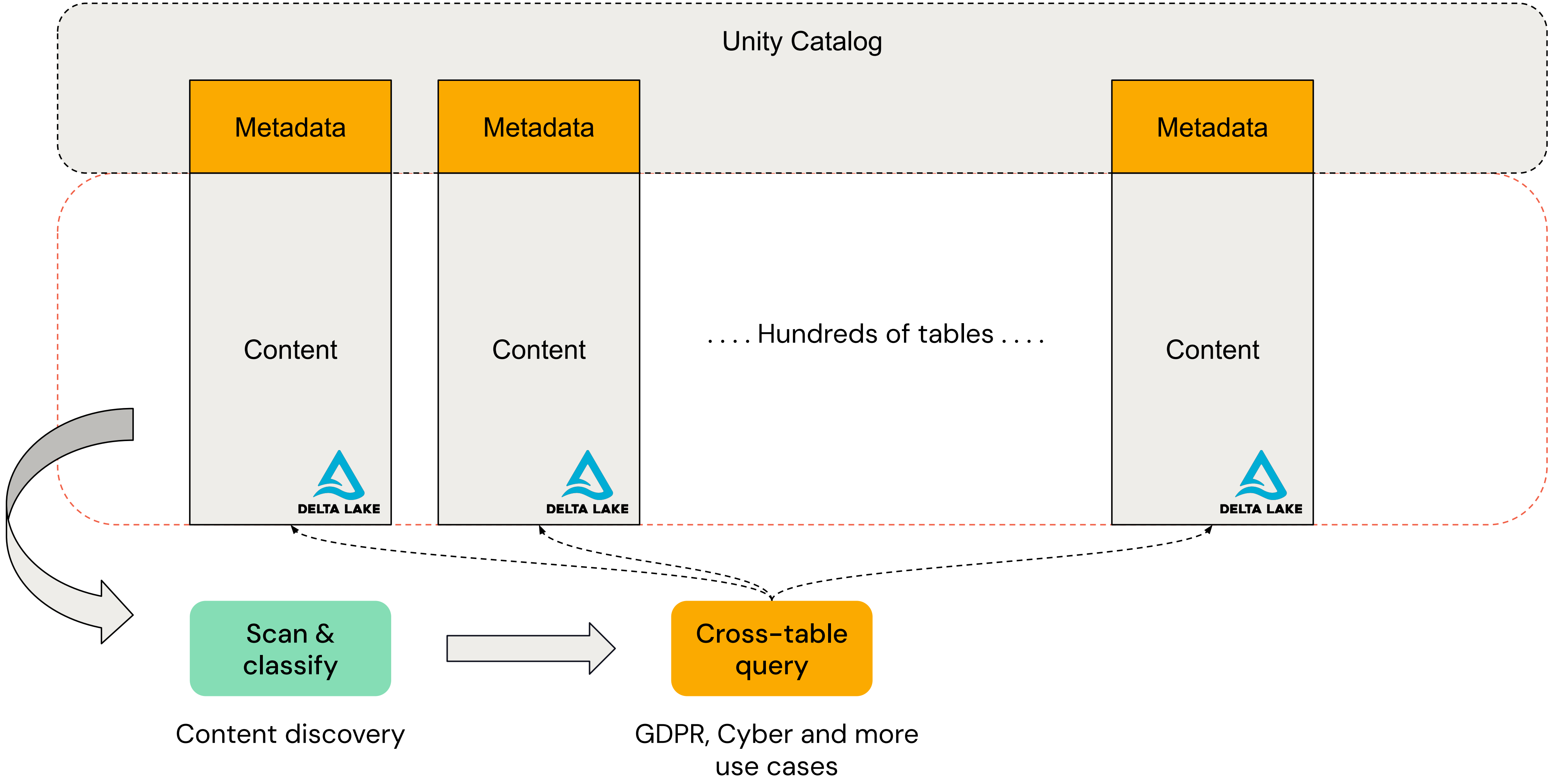
Do I have any record for **email** “erni@databricks.com”?

Find all records with SSN number “123-45-6789” (GDPR compliance)

Why is it difficult?

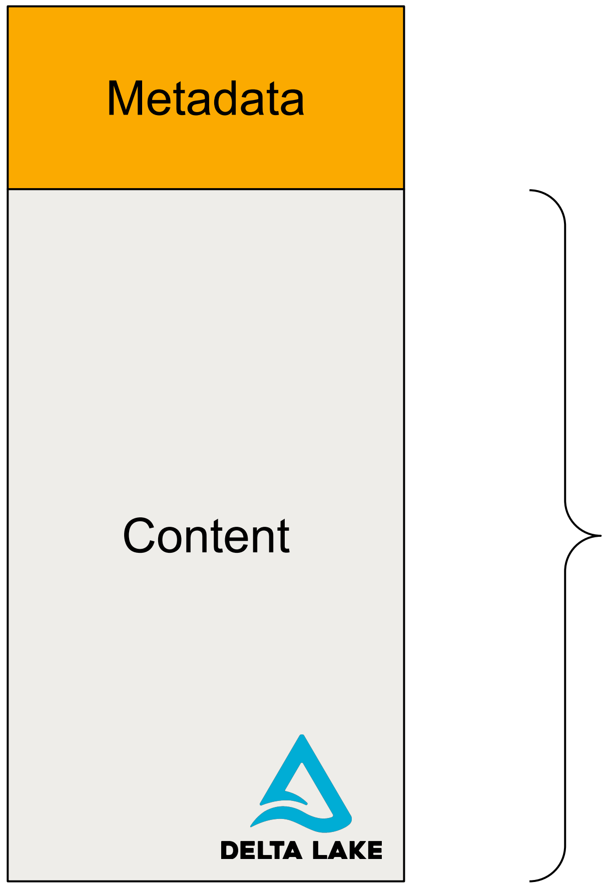

Unity Catalog
Metadata
Metadata
Metadata
Missing semantic information
Content
Content
Content
DELTA LAKE
DELTA LAKE
DELTA LAKE
. . . . Hundreds of tables . . . .
(with hundreds of schemas)

How can DiscoverX help?



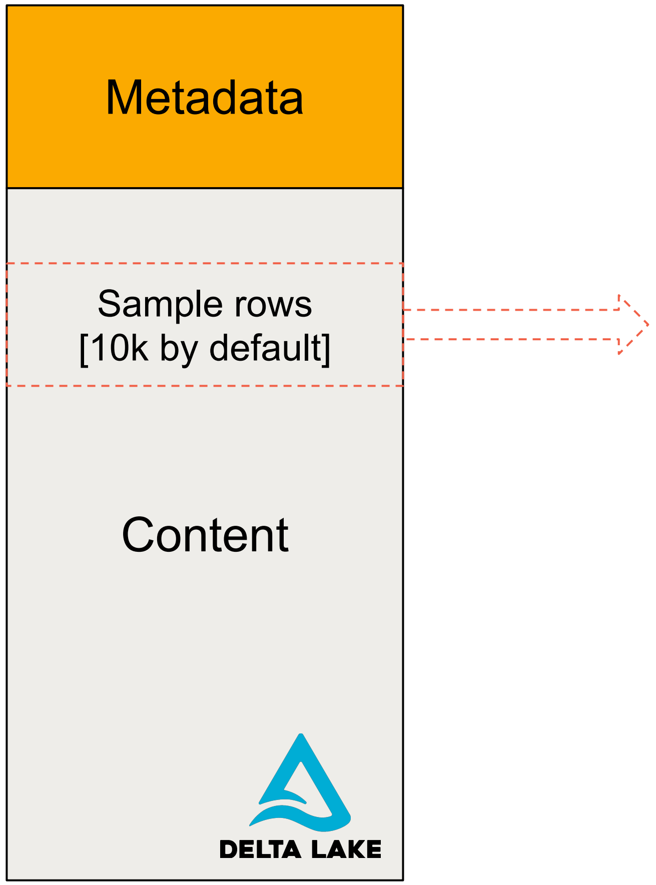
How does it work?

What does it scan for?



- Email address
- IPv4, IPv6, MAC address
- Phone numbers
- SSN
- Credit card number
- Zip code
- URL
- …
- Anything that can be identified by a REGEX

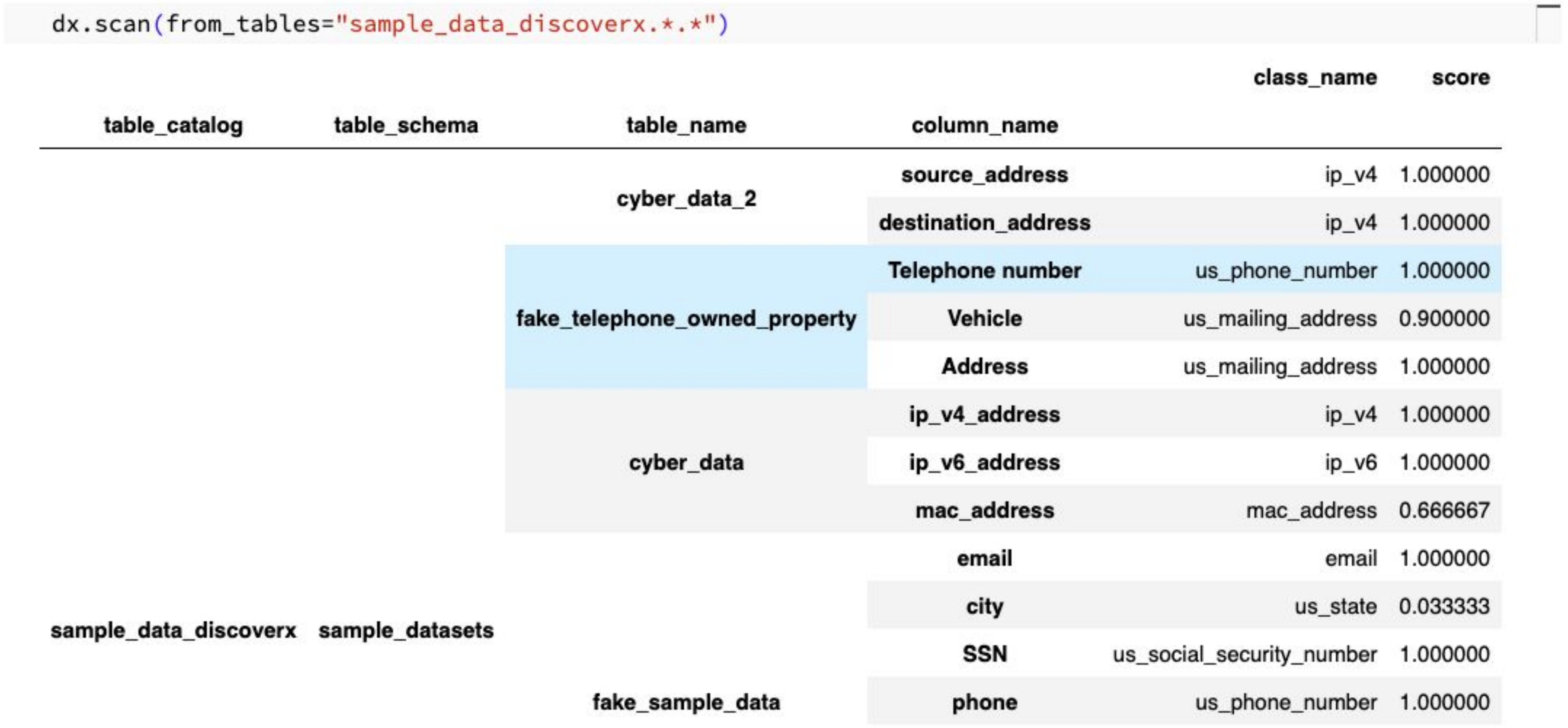
How does it scan?



For each string column, count the **matching frequency** of:

- Email address
- IPv4, IPv6, MAC address
- Phone numbers
- SSN
- Credit card number
- Zip code
- URL
- ...
- Any custom defined REGEX

Scan example

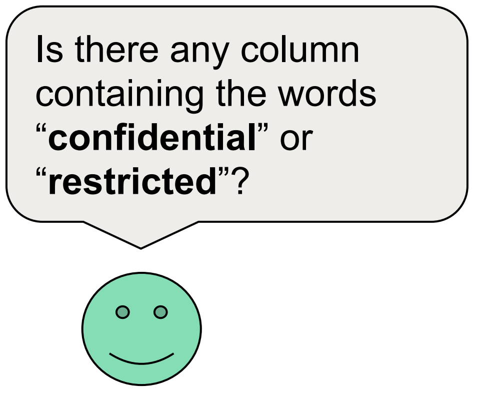


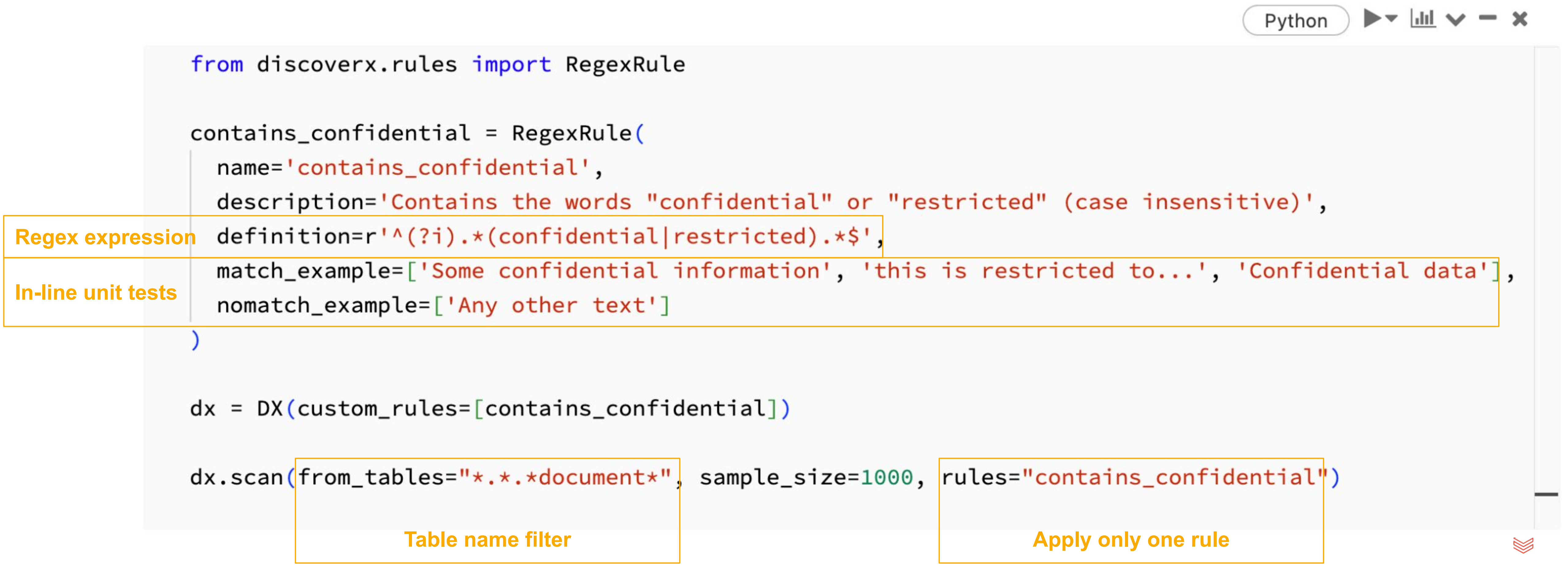
```
dx.scan(from_tables="sample_data_discoverx.*.*")
```

table_catalog	table_schema	table_name	column_name	class_name	score
sample_data_discoverx	sample_datasets	cyber_data_2	source_address	ip_v4	1.000000
			destination_address	ip_v4	1.000000
		fake_telephone_owned_property	Telephone number	us_phone_number	1.000000
			Vehicle	us_mailing_address	0.900000
			Address	us_mailing_address	1.000000
		cyber_data	ip_v4_address	ip_v4	1.000000
			ip_v6_address	ip_v6	1.000000
			mac_address	mac_address	0.666667
		fake_sample_data	email	email	1.000000
			city	us_state	0.033333
			SSN	us_social_security_number	1.000000
			phone	us_phone_number	1.000000

Custom scan rules

Search for specific content

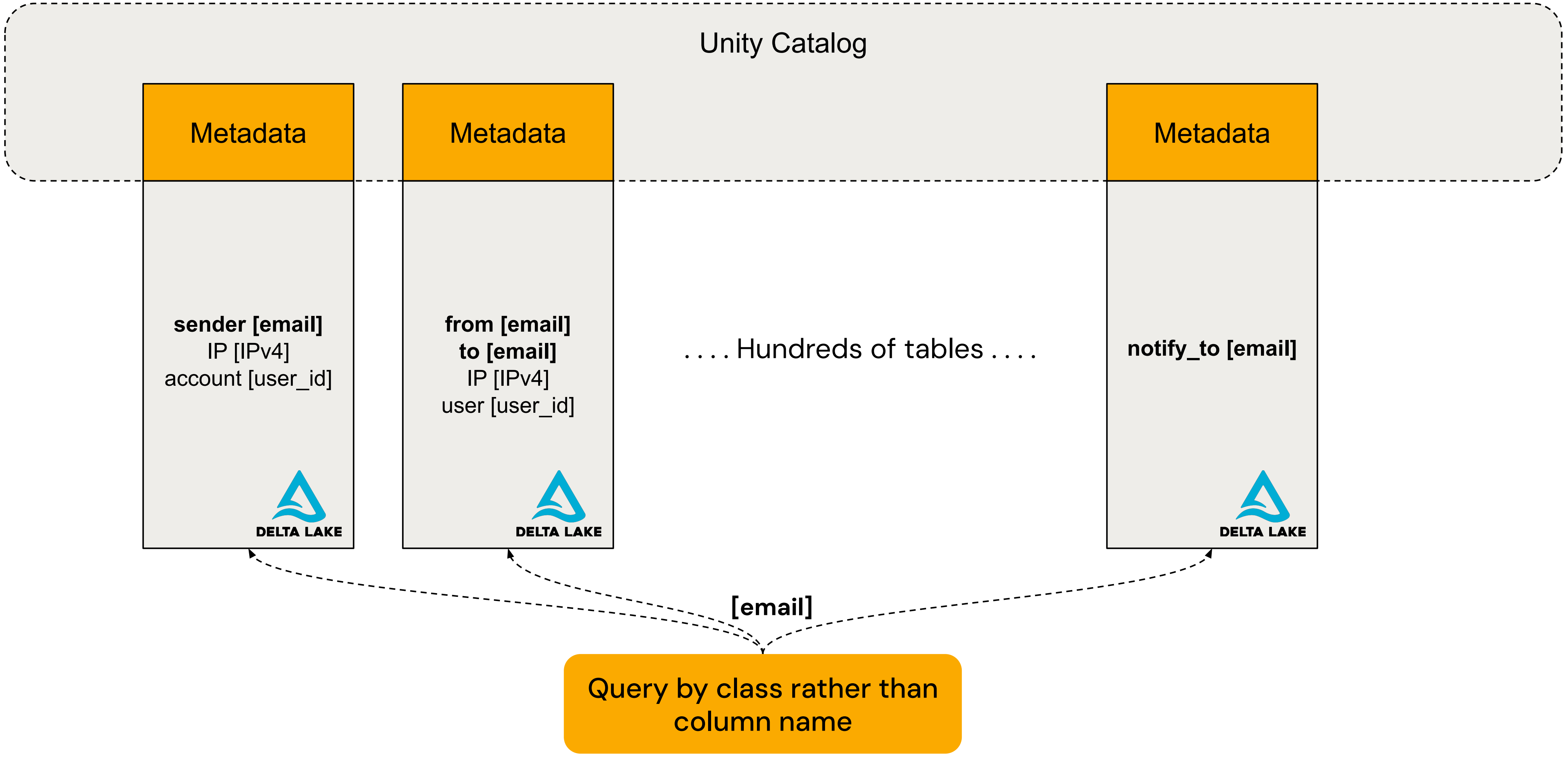

Is there any column containing the words “**confidential**” or “**restricted**”?



```python
from discoverx.rules import RegexRule

contains_confidential = RegexRule(
    name='contains_confidential',
    description='Contains the words "confidential" or "restricted" (case insensitive)',
    definition=r'^(?i).*(confidential|restricted).*$',
    match_example=['Some confidential information', 'this is restricted to...', 'Confidential data'],
    nomatch_example=['Any other text']
)

dx = DX(custom_rules=[contains_confidential])

dx.scan(from_tables="*.*.*document*", sample_size=1000, rules="contains_confidential")
```

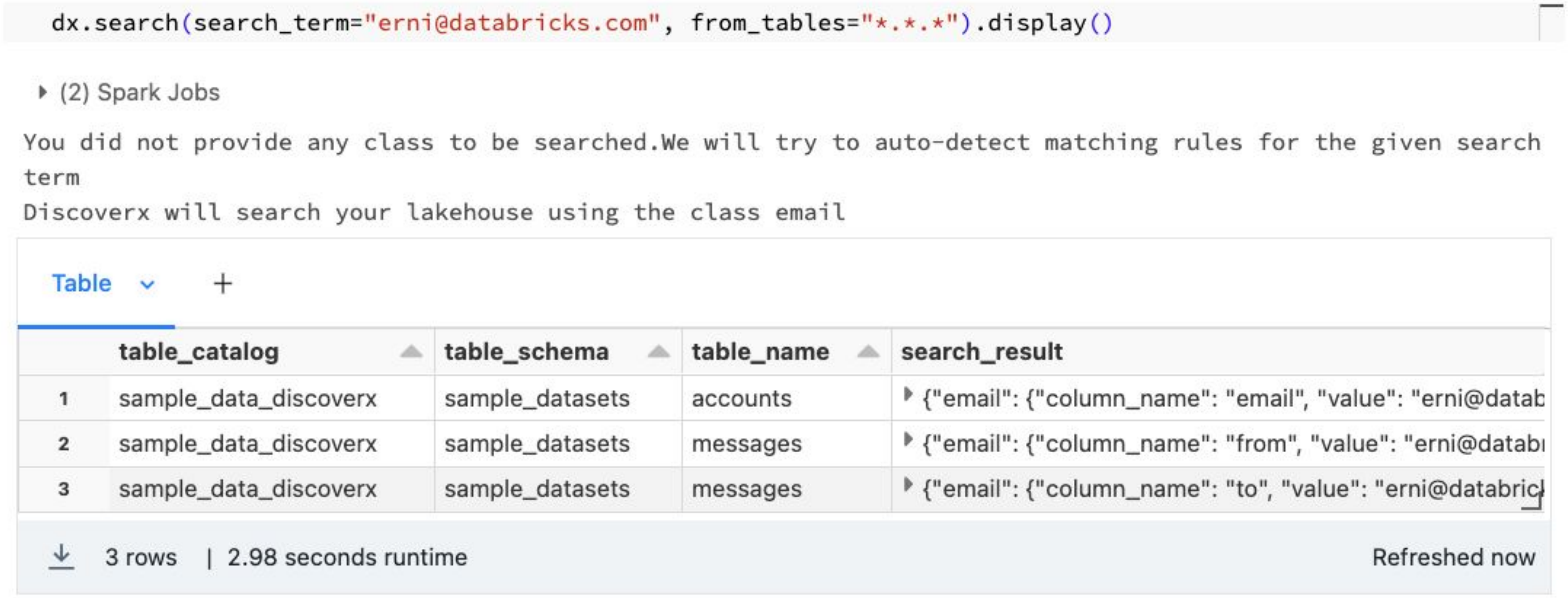
Ok, we scanned...

And now what?



Discover

GDPR right of access



```
dx.search(search_term="erni@databricks.com", from_tables="*.*.*").display()
```

▸ (2) Spark Jobs

```
You did not provide any class to be searched.We will try to auto-detect matching rules for the given search
term
Discoverx will search your lakehouse using the class email
```

	table_catalog	table_schema	table_name	search_result
1	sample_data_discoverx	sample_datasets	accounts	▸ {"email": {"column_name": "email", "value": "erni@datab
2	sample_data_discoverx	sample_datasets	messages	▸ {"email": {"column_name": "from", "value": "erni@datab
3	sample_data_discoverx	sample_datasets	messages	▸ {"email": {"column_name": "to", "value": "erni@databric

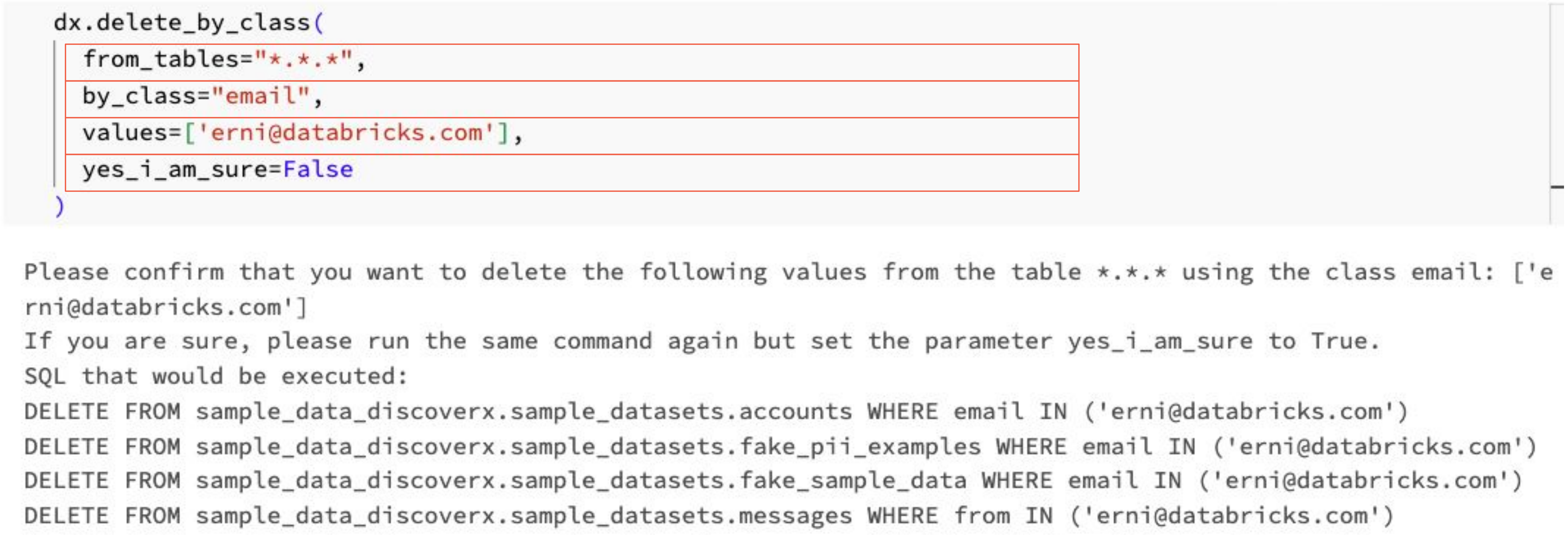
3 rows | 2.98 seconds runtime

Refreshed now

Delete

GDPR right to be forgotten

Requires a **vacuum** to be run after deletion to permanently delete



```
dx.delete_by_class(
    from_tables="*.*.*",
    by_class="email",
    values=['erni@databricks.com'],
    yes_i_am_sure=False
)
```

```
Please confirm that you want to delete the following values from the table *.*.* using the class email: ['e
rni@databricks.com']
If you are sure, please run the same command again but set the parameter yes_i_am_sure to True.
SQL that would be executed:
DELETE FROM sample_data_discoverx.sample_datasets.accounts WHERE email IN ('erni@databricks.com')
DELETE FROM sample_data_discoverx.sample_datasets.fake_pii_examples WHERE email IN ('erni@databricks.com')
DELETE FROM sample_data_discoverx.sample_datasets.fake_sample_data WHERE email IN ('erni@databricks.com')
DELETE FROM sample_data_discoverx.sample_datasets.messages WHERE from IN ('erni@databricks.com')
```

A more advanced common Use Case

Cyber Security

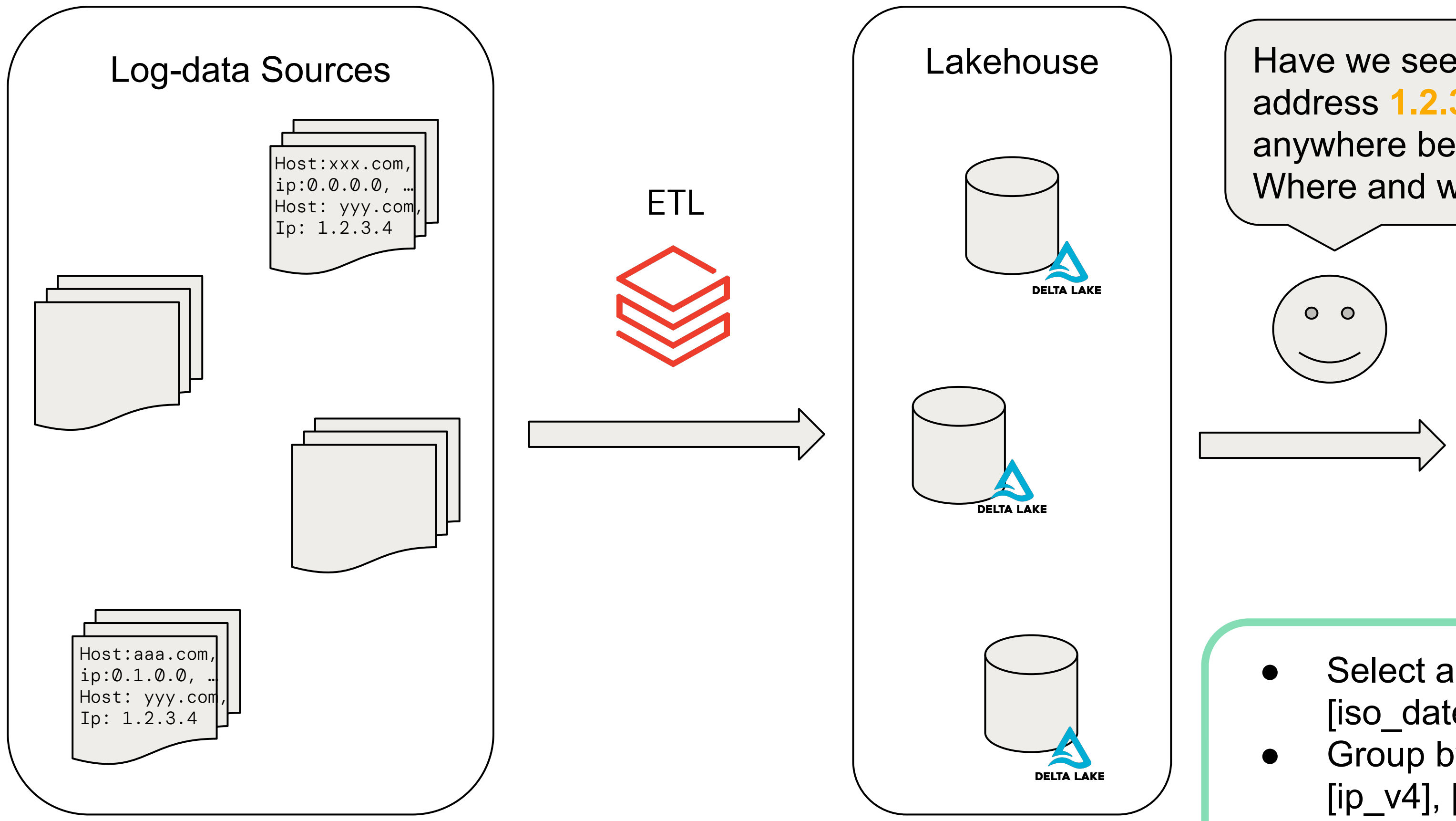
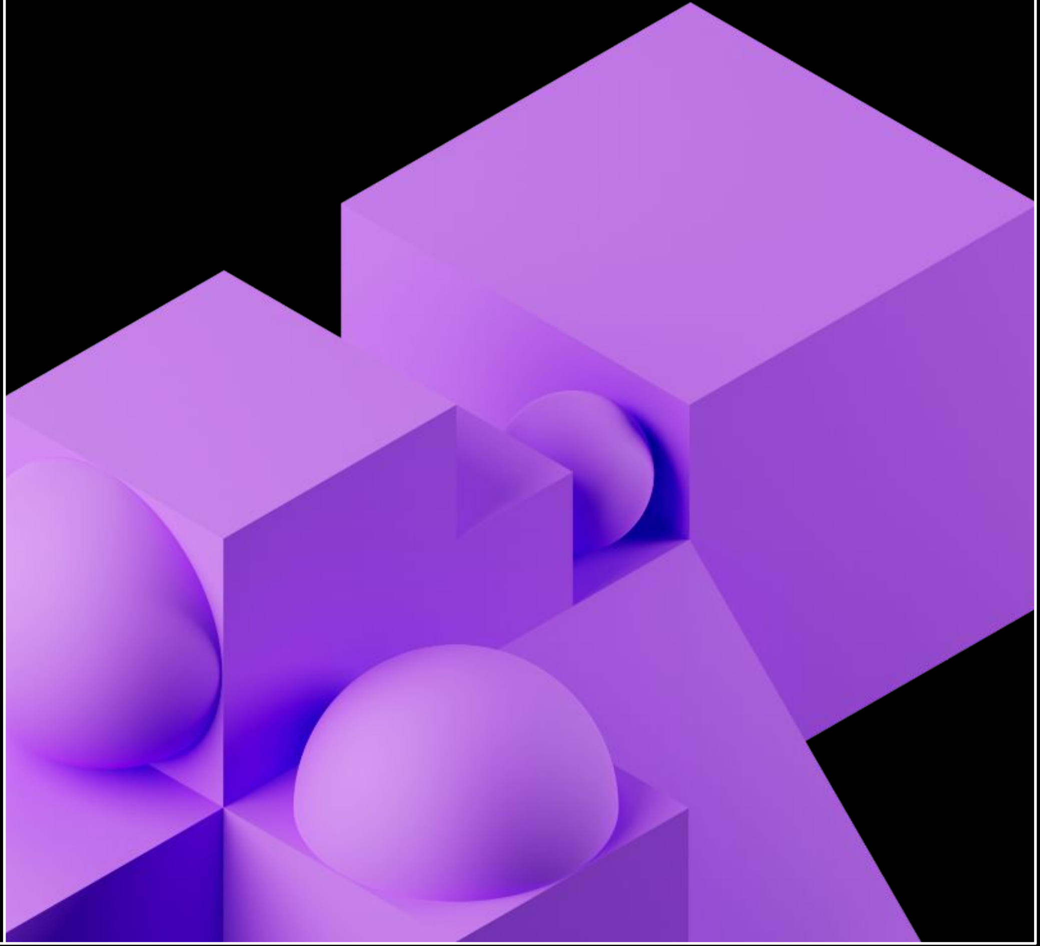


table	column	ip_v4	iso_date	count
a	source	1.0.0.1	2023-07-27	150
a	dest	1.2.3.4	2023-07-27	6
b	ip	4.4.4.4	2023-07-27	32

- Select all [ip_v4] and [iso_date] from all tables
- Group by table, column, [ip_v4], [iso_date]
- count

DEMO

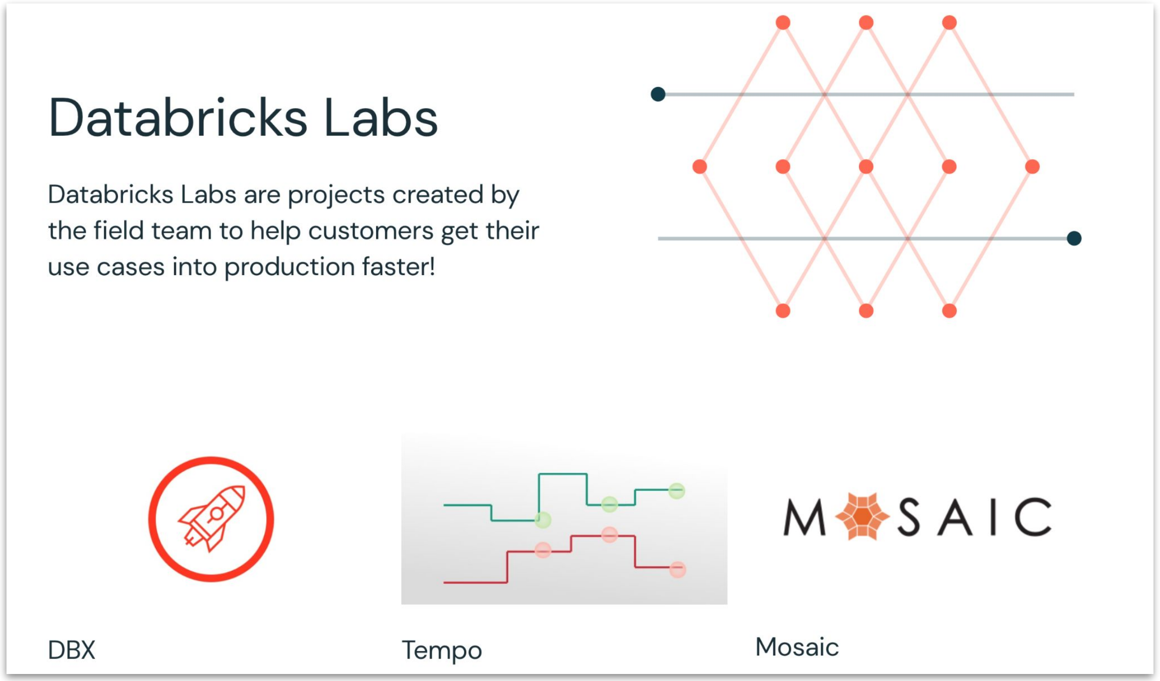


How can I get it

DiscoverX by Databricks Labs

Open Source

- DiscoverX is open source and provided through Databricks Labs
- Available on PyPi
  - `pip install dbl-discoverx`
- Code on Github
  - Provide feedback via Github issues



Databricks Labs

Databricks Labs are projects created by the field team to help customers get their use cases into production faster!

Requirements and limitations

Requirements

- Databricks with **Unity Catalog**

Limitations

- Available through Databricks Labs (best-effort support)
- It only works with string type columns (Complex types coming soon)

Roadmap

Features coming soon

- Integration with new upcoming Databricks features in Unity Catalog and Data Monitoring
  - Talk: Learn What's New in Data Science and Machine Learning
    Wednesday, June 28 @4:30 PM
- Scanning of complex types (struct, map, arrays)
- Column-name rules
- AI-based rules

Please provide your feedback and suggestions via Github issues.

databricks