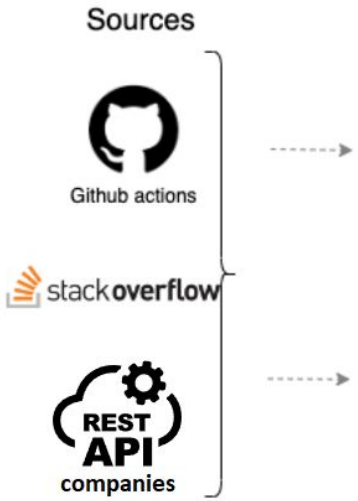
Dusicyon Data Engineering Azure team project

Petr Šturz
Miloš Jánošík
Ondřej Škeřík
Szabolcs Varga
Anita Bogar

Architecture of the project



Azure Data Factory

Microsoft Azure | Data Factory ▸ ADF-GFA

Search

adf_psturz branch | Validate all | Save all | Publish

Factory Resources

Filter resources by name

- Pipelines 5
  - Copy_company_details
  - Copy_GitHub
  - Copy_stackoverflow
  - Get_all_data_from_external_sources
  - Master
- Datasets 8
  - Starting data 8
    - Landing 4
    - Source 4
      - GitHub 1
      - REST company det... 1
      - Stackoverflow 2
- Data flows 0
- Power Query 0
- Templates 0

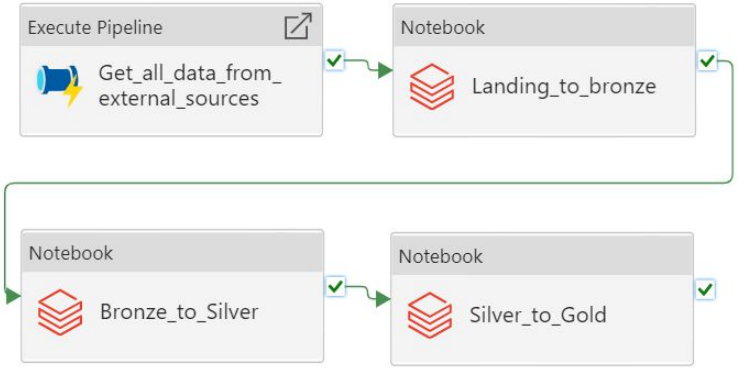
Activities

Search activities

- Move & transform
- Azure Data Explorer
- Azure Function
- Batch Service
- Databricks
- Data Lake Analytics
- General
- HDInsight
- Iteration & conditionals
- Machine Learning
- Power Query

Get_all_data_from_... | Source_Stackoverflo... | Copy_GitHub | Copy_stackoverflow | Copy_company_det... | Master

Save | Save as template | Validate | Debug | Add trigger



Parameters | Variables | Settings | Output

New | Delete

Name	Type	Default value
p_load_date	String	20220731

Petr

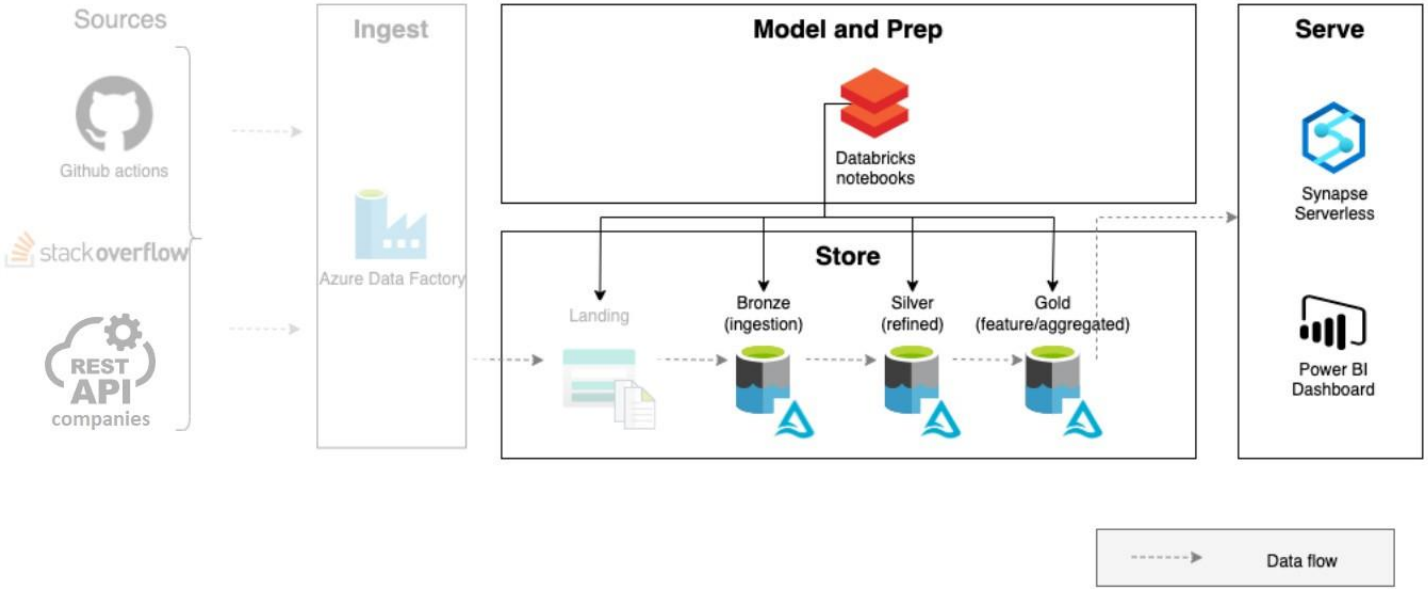
Azure

Introduction

-3 parts(layers)

-purpose of Bronze layer

-Huge datasets



Bronze layer

Load date value

p_file_date

20220831

Cmd 3

```
file_date = dbutils.widgets.get("p_file_date")
```

Cmd 16

```
df_github_schema = df_github_schema \
.withColumnRenamed("created_at","created_at_datetime_utc") \
.withColumnRenamed("public","is_public") \
    .withColumn("load_date_datetime_utc", to_utc_timestamp(current_timestamp(),'UTC')) \
    .withColumn("valid_date", to_date(lit(file_date), "yyyyMMdd")) \
    .withColumn("_pk", col("id")) \
```

```
|-- actor: struct (nullable = true)
|    |-- avatar_url: string (nullable = true)
|    |-- gravatar_id: string (nullable = true)
|    |-- id: string (nullable = true)
|    |-- login: string (nullable = true)
|    |-- url: string (nullable = true)
|-- created_at: string (nullable = true)
|-- id: string (nullable = true)
|-- org: struct (nullable = true)
|    |-- avatar_url: string (nullable = true)
|    |-- gravatar_id: string (nullable = true)
|    |-- id: string (nullable = true)
|    |-- login: string (nullable = true)
|    |-- url: string (nullable = true)
|-- other: string (nullable = true)
|-- public: boolean (nullable = true)
|-- repo: struct (nullable = true)
|    |-- id: string (nullable = true)
|    |-- name: string (nullable = true)
|    |-- url: string (nullable = true)
|-- type: string (nullable = true)
```

Bronze layer

-Delta tables

	database	tableName
1	bronze_db	b_companydetails
2	bronze_db	b_github_20220731
3	bronze_db	b_github_20220831
4	bronze_db	b_github_20220930
5	bronze_db	b_stackoverflowanswers
6	bronze_db	b_stackoverflowquestions

Cmd 18

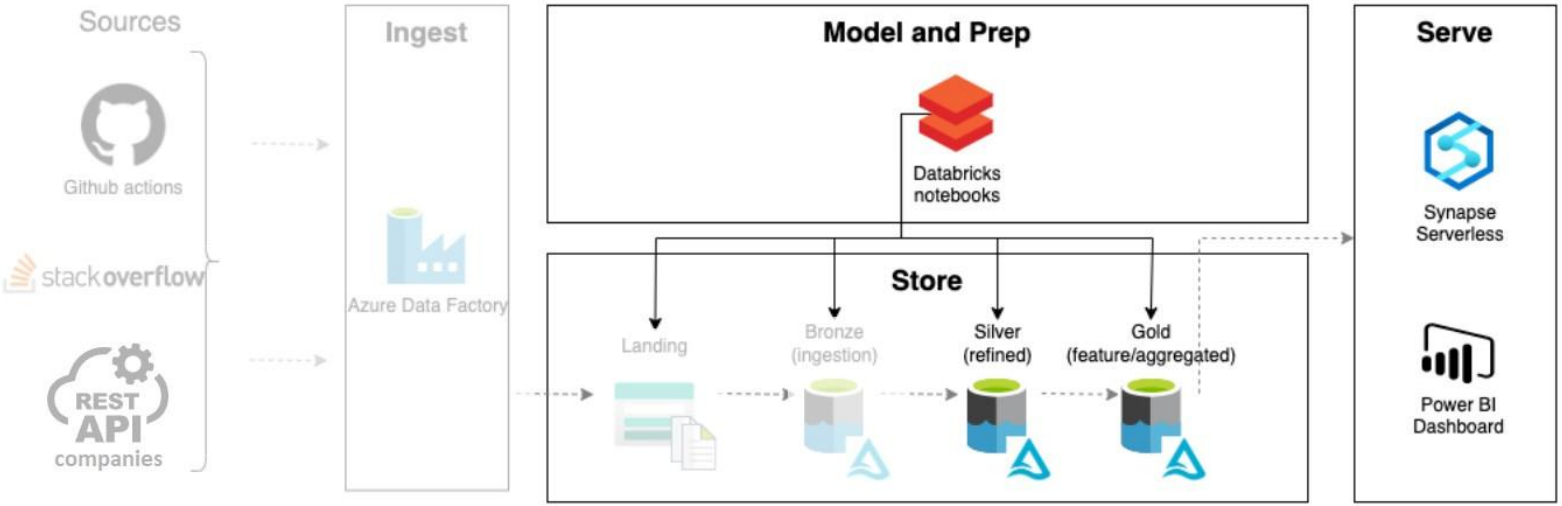
```
df_github_schema.write.format("delta").mode("overwrite").saveAsTable("bronze_db.b_github")
```

Bronze layer

Cmd 3

```
dbutils.notebook.run("./etl/GitHub_etl", timeoutS, {"p_file_date" : "20220731"})
dbutils.notebook.run("./etl/REST_etl", timeoutS, {"p_file_date" : "20220731"})
dbutils.notebook.run("./etl/StackOverflow_answers_etl", timeoutS)
dbutils.notebook.run("./etl/StackOverflow_questions_etl", timeoutS)
```

Silver layer



- b_company
- b_github
- b_stackoverflow_post_answers
- b_stackoverflow_post_questions

→

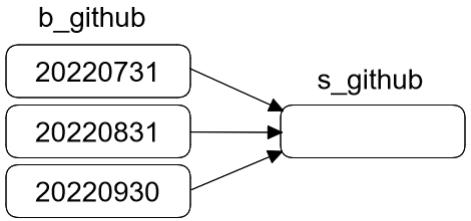
- s_company
- s_github
- s_stackoverflow_post_answers
- s_stackoverflow_post_questions

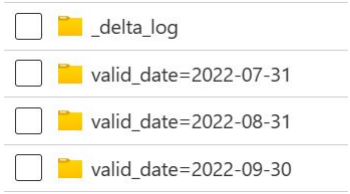
Data flow

Incremental Load

GitHub data

```
final_df.write \
.mode("overwrite") \
.partitionBy("valid_date") \
.format("delta") \
.option("partitionOverwriteMode", "dynamic") \
.saveAsTable("silver_db.s_github")
```





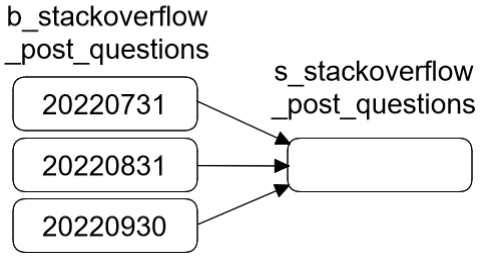
	_pk	repository_account	repository_name	user_id	event_id	type	created_at_datetime_utc	valid_date	dbx_created_at_datetime_utc
1	19541174571	polytomic	pipedrive-api	510875	19541174571	PushEvent	2022-01-01T00:00:04.000+0000	2022-07-31	2023-01-25T14:25:18.805+0000
2	19541183344	grouparoo	sync-engine-example	49699333	19541183344	CreateEvent	2022-01-01T00:01:35.000+0000	2022-07-31	2023-01-25T14:25:18.805+0000
3	23179896442	firebase	friendlyeats-web	6344405	23179896442	WatchEvent	2022-08-01T00:03:30.000+0000	2022-08-31	2023-01-25T14:31:58.320+0000
4	23179912111	metabase	metabase	3309992	23179912111	WatchEvent	2022-08-01T00:05:53.000+0000	2022-08-31	2023-01-25T14:31:58.320+0000
5	23753918452	airbytehq	airbyte	1142800	23753918452	IssueCommentEvent	2022-09-01T00:00:07.000+0000	2022-09-30	2023-01-25T15:09:53.160+0000
6	23753920556	snowflakedb	snowflake-connector-python	63477823	23753920556	PullRequestReviewCommentEvent	2022-09-01T00:00:15.000+0000	2022-09-30	2023-01-25T15:09:53.160+0000

	valid_date	count(1)
1	2022-07-31	642198
2	2022-08-31	123072
3	2022-09-30	123890

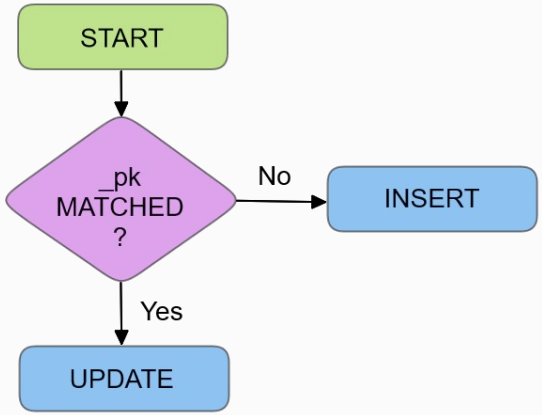
Slowly changing dimension type 1 (SCD1)

Stackoverflow data

- Overwrites old data with new data
- Does not track historical data



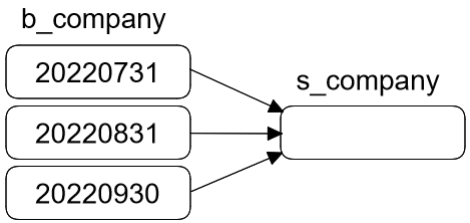

```sql
%sql
MERGE INTO silver_db.s_stackoverflow_post_questions AS target
USING temp_view AS source
ON target._pk = source._pk
WHEN MATCHED THEN UPDATE SET
target.id = source.id,
target.title = source.title,
target.accepted_answer_id = source.accepted_answer_id,
target.answer_count = source.answer_count,
target.comment_count = source.comment_count,
target.community_owned_datetime_utc = source.community_owned_datetime_utc,
target.created_at_datetime_utc = source.created_at_datetime_utc,
target.favorite_count = source.favorite_count,
target.last_activity_datetime_utc = source.last_activity_datetime_utc,
target.last_edit_datetime_utc = source.last_edit_datetime_utc,
target.last_editor_display_name = source.last_editor_display_name,
target.last_editor_user_id = source.last_editor_user_id,
target.owner_display_name = source.owner_display_name,
target.owner_user_id = source.owner_user_id,
target.parent_id = source.parent_id,
target.post_type_id = source.post_type_id,
target.score = source.score,
target.view_count = source.view_count,
target.tags_array = source.tags_array,
target.loaded_at_datetime_utc = source.loaded_at_datetime_utc,
target.valid_date = source.valid_date,
target.dbx_updated_at_datetime_utc = current_timestamp
WHEN NOT MATCHED THEN INSERT (id, title, accepted_answer_id, answer_count, comment_count, community_owned_datetime_utc, created_at_datetime_utc, favorite_count,
last_activity_datetime_utc, last_edit_datetime_utc, last_editor_display_name, last_editor_user_id, owner_display_name, owner_user_id, parent_id, post_type_id, score,
view_count, tags_array, _pk, loaded_at_datetime_utc, valid_date, dbx_created_at_datetime_utc) VALUES (id, title, accepted_answer_id, answer_count, comment_count,
community_owned_datetime_utc, created_at_datetime_utc, favorite_count, last_activity_datetime_utc, last_edit_datetime_utc, last_editor_display_name, last_editor_user_id,
owner_display_name, owner_user_id, parent_id, post_type_id, score, view_count, tags_array, _pk, loaded_at_datetime_utc, valid_date, current_timestamp);
```



Slowly changing dimension type 2 (SCD2)

Company Detail data

- Adds new rows with new data
- Uses current flags to mark the valid rows
- Tracks historical data

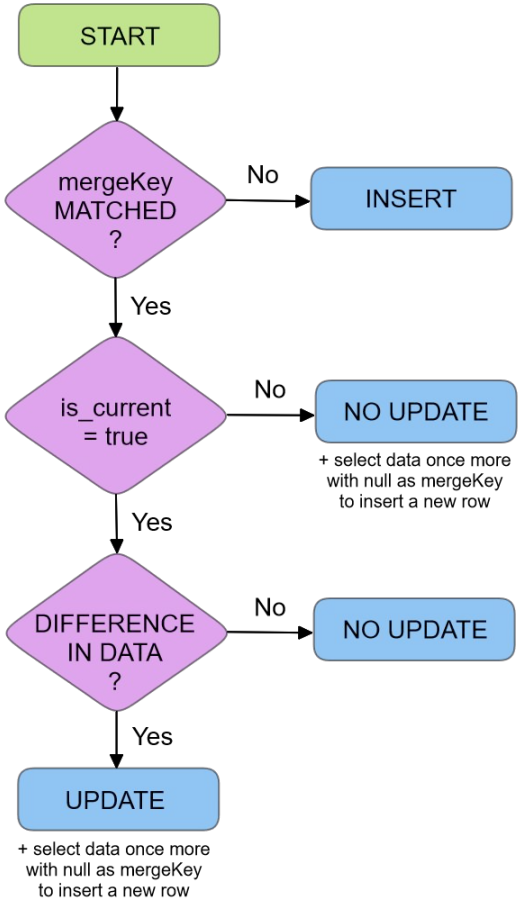


	organization_name	repository_account	repository_name	l1_type	l2_type	l3_type	tags_array	is_current	valid_from_date	valid_to_date	dbx_created_at_datetime_utc	dbx_updated_at_datetime_utc
1	Airflow	apache	airflow	modern_data_stack	orchestration		["airflow"]	true	2022-07-31	null	2023-02-01T10:30:53.060+0000	null
2	Dremio	dremio		modern_data_stack	data_platform		["dremio"]	true	2022-07-31	null	2023-02-01T10:30:53.060+0000	null

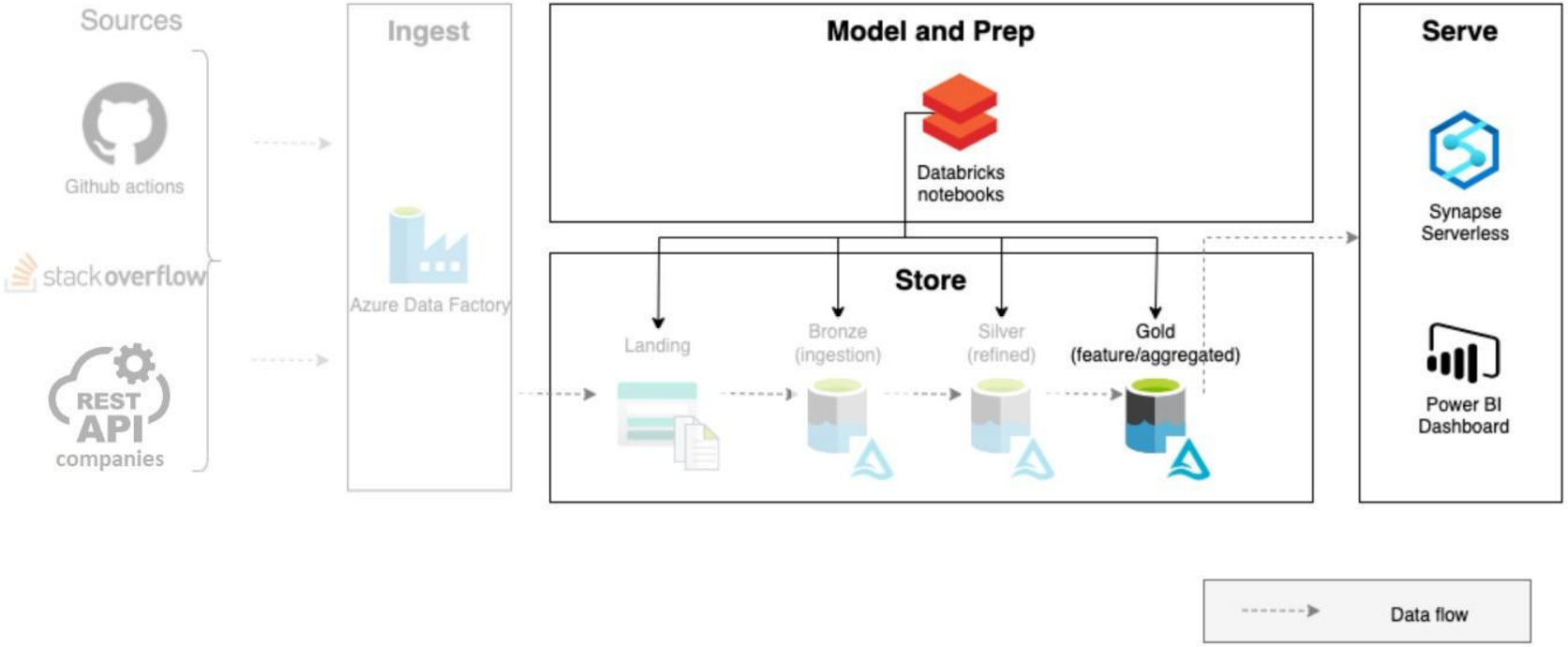
	organization_name	repository_account	repository_name	l1_type	l2_type	l3_type	tags_array	is_current	valid_from_date	valid_to_date	dbx_created_at_datetime_utc	dbx_updated_at_datetime_utc
1	Airflow	apache	airflow	modern_data_stack	orchestration		["airflow"]	false	2022-07-31	2022-08-30	2023-02-01T10:30:53.060+0000	2023-02-01T10:44:33.394+0000
2	Airflow	apache	airflow	modern_data_stack	orchestration		["airflow", "apache-airflow", "apache airflow"]	true	2022-08-31	null	2023-02-01T10:44:33.394+0000	null
3	Dremio	dremio		modern_data_stack	data_platform		["dremio"]	false	2022-07-31	2022-08-30	2023-02-01T10:30:53.060+0000	2023-02-01T10:44:33.394+0000

	organization_name	repository_account	repository_name	l1_type	l2_type	l3_type	tags_array	is_current	valid_from_date	valid_to_date	dbx_created_at_datetime_utc	dbx_updated_at_datetime_utc
1	Airflow	apache	airflow	modern_data_stack	orchestration		["airflow"]	false	2022-07-31	2022-08-30	2023-02-01T10:30:53.060+0000	2023-02-01T10:44:33.394+0000
2	Airflow	apache	airflow	modern_data_stack	orchestration		["airflow", "apache-airflow", "apache airflow"]	false	2022-08-31	2022-09-29	2023-02-01T10:44:33.394+0000	2023-02-01T10:47:47.254+0000
3	Airflow	apache	airflow	Modern data stack	Orchestration		["airflow", "apache-airflow", "apache airflow"]	true	2022-09-30	null	2023-02-01T10:47:47.254+0000	null
4	Dremio	dremio		modern_data_stack	data_platform		["dremio"]	false	2022-07-31	2022-08-30	2023-02-01T10:30:53.060+0000	2023-02-01T10:44:33.394+0000
5	Dremio	dremio		Modern data stack	Data warehouse		["dremio"]	true	2022-09-30	null	2023-02-01T10:47:47.254+0000	null

```
spark.sql(f"""
MERGE INTO silver_db.s_company AS base_table
USING (

SELECT new_table.organization_name AS mergeKey, new_table.organization_name, new_table.repository_account, new_table.repository_name, new_table.l1_type, new_table.l2_type, new_table.l3_ty
hash(new_table.organization_name, new_table.valid_date) AS _pk, new_table.valid_date AS valid_from_date
FROM bronze_db.b_company AS new_table

UNION ALL

-- Extra data 1 for INSERT
-- (some data of this mergeKey already in the base_table, is_current = true, different data in new_table)

SELECT null AS mergeKey, new_table.organization_name, new_table.repository_account, new_table.repository_name, new_table.l1_type, new_table.l2_type, new_table.l3_type, new_table.tags_arra
hash(new_table.organization_name, new_table.valid_date) AS _pk, new_table.valid_date AS valid_from_date
FROM bronze_db.b_company AS new_table
JOIN silver_db.s_company AS base_table
ON new_table.organization_name = base_table.organization_name
WHERE base_table.is_current = true AND (new_table.repository_account <> base_table.repository_account OR new_table.repository_name <> base_table.repository_name OR new_table.l1_type <> ba
OR new_table.l3_type <> base_table.l3_type OR new_table.tags_array <> base_table.tags_array OR new_table.is_open_source_available <> base_table.is_open_source_available)

UNION ALL

-- Extra data 2 for UPDATE
-- (some data of this mergeKey already in the base_table, is_current = true, no data in new_table)

SELECT base_table.organization_name AS mergeKey, new_table.organization_name, new_table.repository_account, new_table.repository_name, new_table.l1_type, new_table.l2_type, new_table.l3_t
new_table.is_open_source_available, hash(new_table.organization_name, new_table.valid_date) AS _pk, new_table.valid_date AS valid_from_date
FROM bronze_db.b_company AS new_table
FULL JOIN silver_db.s_company AS base_table
ON new_table.organization_name = base_table.organization_name
WHERE base_table.is_current = true AND new_table.organization_name IS null

UNION ALL

-- Extra data 3 for INSERT
-- (some data of this mergeKey already in the base_table, is_current = false, some data in new_table)

{one_more_select_query}

) AS staged_updates
ON base_table.organization_name = mergeKey
WHEN MATCHED AND base_table.is_current = true AND (staged_updates.repository_account <> base_table.repository_account OR staged_updates.repository_name <> base_table.repository_name OR st
staged_updates.l2_type <> base_table.l2_type OR staged_updates.l3_type <> base_table.l3_type OR staged_updates.tags_array <> base_table.tags_array OR staged_updates.is_open_source_availab
staged_updates.organization_name IS null) THEN UPDATE SET
base_table.is_current = false,
base_table.valid_to_date = date_sub('{v_valid_date}', 1),
base_table.dbx_updated_at_datetime_utc = current_timestamp
WHEN NOT MATCHED THEN INSERT (organization_name, repository_account, repository_name, l1_type, l2_type, l3_type, tags_array, is_open_source_available, _pk, is_current, valid_from_date, db
(staged_updates.organization_name, staged_updates.repository_account, staged_updates.repository_name, staged_updates.l1_type, staged_updates.l2_type, staged_updates.l3_type, staged_update
staged_updates._pk, true, staged_updates.valid_from_date, current_timestamp);
""")
```



Gold layer


Sources
Github actions
stackoverflow
REST
API
companies
Ingest
Azure Data Factory
Model and Prep
Databricks
notebooks
Store
Landing
Bronze
(ingestion)
Silver
(refined)
Gold
(feature/aggregated)
Serve
Synapse
Serverless
Power BI
Dashboard
Data flow

Features

- Filtered
- Business - leveled
- Aggregated
- Used for dashboarding, reporting

Name
g_github_daily
g_github_monthly
g_github_quarterly
g_stackoverflow_post_questions_daily
g_stackoverflow_post_questions_monthly
g_stackoverflow_post_questions_quarterly

Silver Tables → Gold Views

```sql
-- -- Rows number with this method: 21 912 pcs -- --

CREATE OR REPLACE VIEW gold_db.g_stackoverflow_post_questions_view
AS
SELECT ss.*, sc.organization, sc.tags_array
FROM
  (SELECT *, EXPLODE(SPLIT(tags, '\\|')) stack_tags
   FROM silver_db.s_stackoverflow_post_questions) ss
JOIN
  (SELECT *, EXPLODE(tags_array) company_tags
   FROM silver_db.s_company_detail) sc
   ON ss.stack_tags = sc.company_tags
WHERE sc.is_current = True
```

~ 780 000 rows

```sql
-- -- Rows number with this method: 786 352 pcs -- --

CREATE OR REPLACE VIEW gold_db.g_github_view
AS
SELECT sg.*, sc.organization
FROM silver_db.s_github sg
JOIN silver_db.s_company_detail sc
ON sg.repository_account = sc.repository_account
WHERE (sc.repository_name = sg.repository_name AND sc.is_current = True)
  OR  (sc.repository_name = "" AND sc.is_current = True);
```

~ 20 000 rows

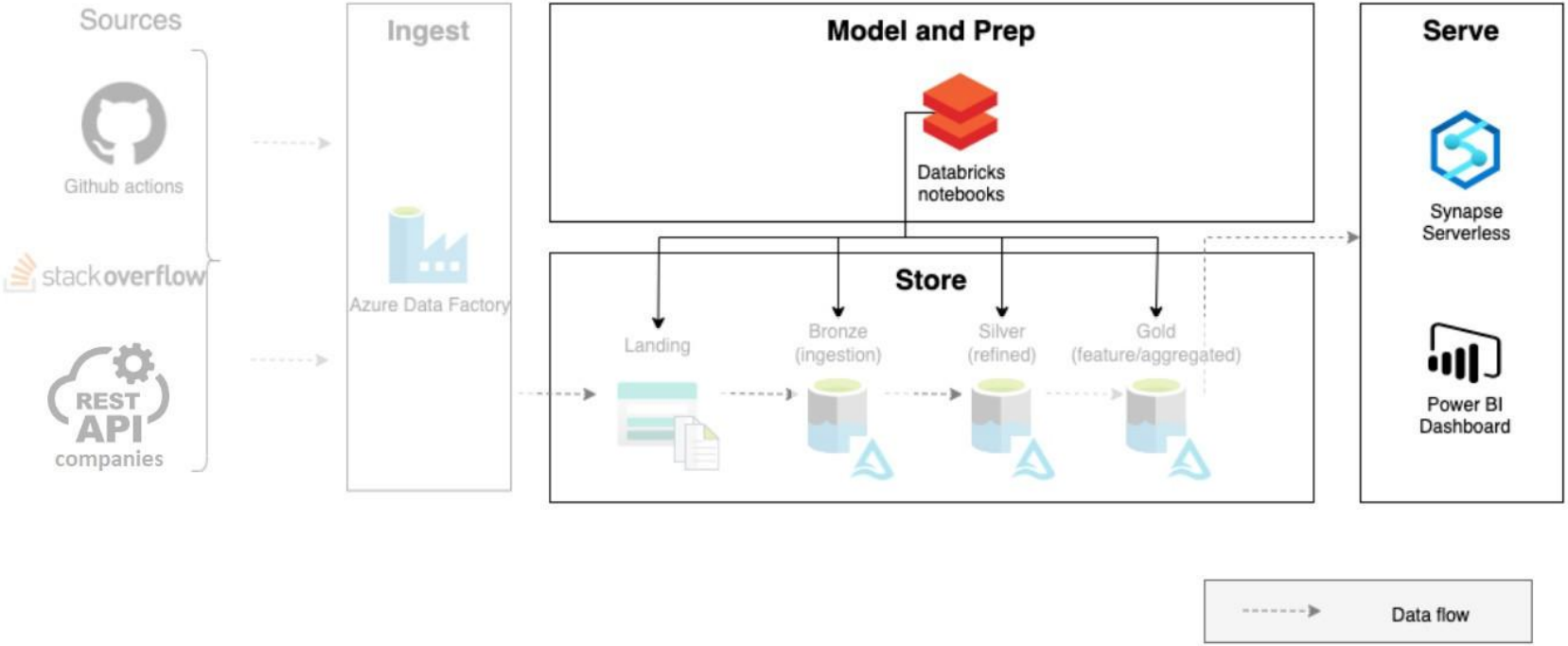
Gold Views → Gold Tables

```sql
CREATE OR REPLACE TABLE gold_db.g_stackoverflow_post_questions_daily
AS
SELECT  Md5(to_date(creation_date_datetime_utc)||organization||MONTH(creation_date_datetime_utc)||QUARTER(creation_date_datetime_utc)) AS _pk,
        to_date(creation_date_datetime_utc) AS first_day_of_period,
        MONTH(creation_date_datetime_utc) AS month,
        QUARTER(creation_date_datetime_utc) AS quarter,
        YEAR(creation_date_datetime_utc) AS year,
        organization AS organization_name,
        COUNT(DISTINCT id) AS post_count,
        SUM(answer_count) AS answer_count,
        ROUND(AVG(answer_count), 3) AS avg_answer_count,
        SUM(comment_count) AS comment_count,
        ROUND(AVG(comment_count), 3) AS avg_comment_count,
        SUM(COALESCE(favorite_count, 0)) AS favorite_count,
        ROUND(AVG(COALESCE(favorite_count, 0)), 3) AS avg_favorite_count,
        SUM(view_count) AS view_count,
        ROUND(AVG(view_count), 3) AS avg_view_countcomment_count,
        COUNT(accepted_answer_id) AS accepeted_answer_count,
        ROUND(COUNT(accepted_answer_id)/COUNT(id), 3) AS avg_accepted_answer_count,
        COUNT(CASE WHEN answer_count = 0 THEN 1 ELSE null END) AS no_answer_count,
        ROUND(COUNT(CASE WHEN answer_count = 0 THEN 1 ELSE null END)/COUNT(id), 3) AS avg_no_answer_count,
        ROUND(SUM(score)/COUNT(id), 3) AS score,
        SUM(SIZE(ARRAY_EXCEPT(SPLIT(tags, '\\|'), tags_array))) AS tags_count,
        MAX(last_activity_date_datetime_utc) AS last_activity_datetime_utc,
        MAX(last_edit_date_datetime_utc) AS last_edit_datetime_utc
FROM gold_db.g_stackoverflow_post_questions_view
GROUP BY Md5(to_date(creation_date_datetime_utc)||organization||MONTH(creation_date_datetime_utc)||QUARTER(creation_date_datetime_utc)),
         to_date(creation_date_datetime_utc),
         MONTH(creation_date_datetime_utc),
         QUARTER(creation_date_datetime_utc),
         YEAR(creation_date_datetime_utc),
         organization_name
ORDER BY first_day_of_period
```

Gold Tables

```
gold_db.g_github_daily

gold_db.g_github_monthly

gold_db.g_github_quarterly

gold_db.g_stackoverflow_post_questions_daily

gold_db.g_stackoverflow_post_questions_monthly

gold_db.g_stackoverflow_post_questions_quarterly
```

Azure Synapse + Power BI



Azure Synapse Analytics

Connect Databricks and Synapse with pyodbc

```
connect = pyodbc.connect(
    f"DRIVER={driver};"
    f"SERVER={server}, 1433;"
    "Trusted_Connection=no;"
    f"uid={userID};"
    f"pwd={password};",
    autocommit=True
)
```

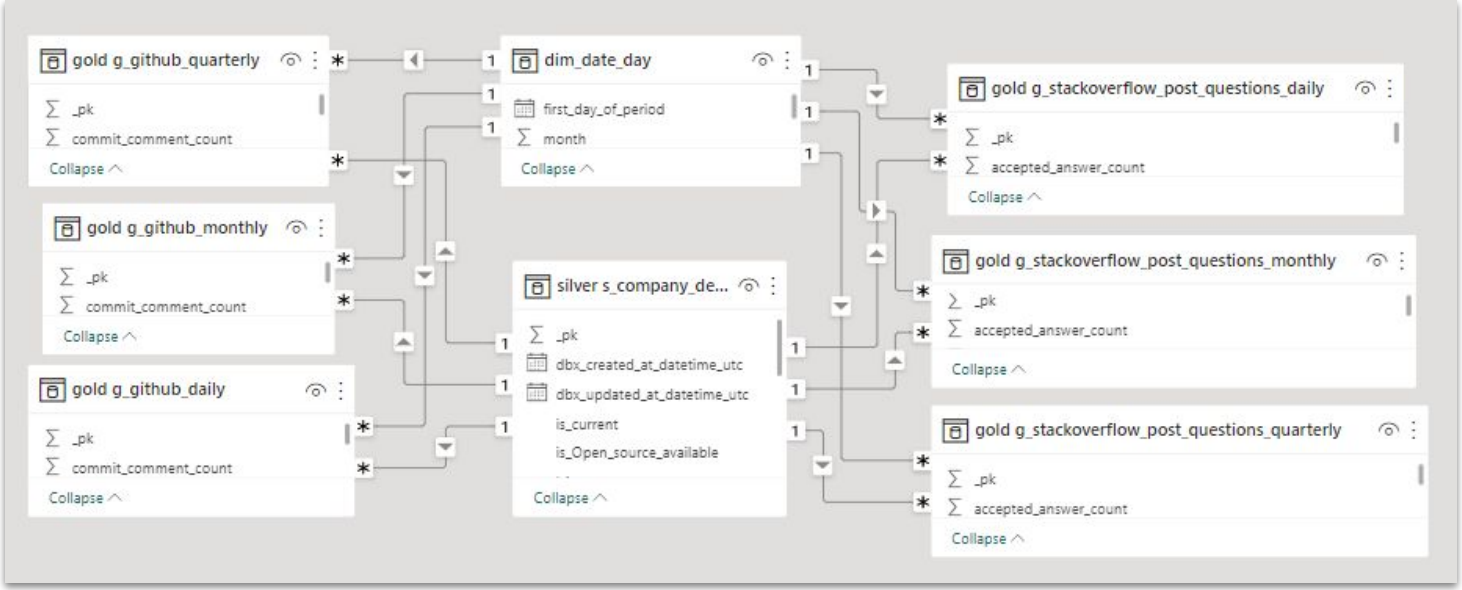
Create credentials and schemas

```
cursor.execute("""if not exists (select * from sys.database_credentials where name = 'ManageIdentityCredential')
                begin
                      create master key encryption by password = '{}'
                      create database scoped credential ManageIdentityCredential with identity = 'Managed Identity'
                end
                """.format(masterPass))

cursor.execute("""if not exists (select * from sys.external_data_sources where name = 'bronze')
                begin
                      create external data source bronze
                      with (
                              location = 'https://{}.blob.core.windows.net/bronze',
                              credential = ManageIdentityCredential)
                end

                if not exists (select * from sys.schemas where name = 'bronze')
                begin
                        exec('create schema bronze')
                end""".format(storage))
```

Create views from tables

```
cursor.execute("""create or alter view gold.g_stackoverflow_post_questions_monthly
                as select *
                from
                      openrowset( bulk 'g_stackoverflow_post_questions_monthly',
                      data_source = 'gold', format = 'delta') as rows
                      """)
```

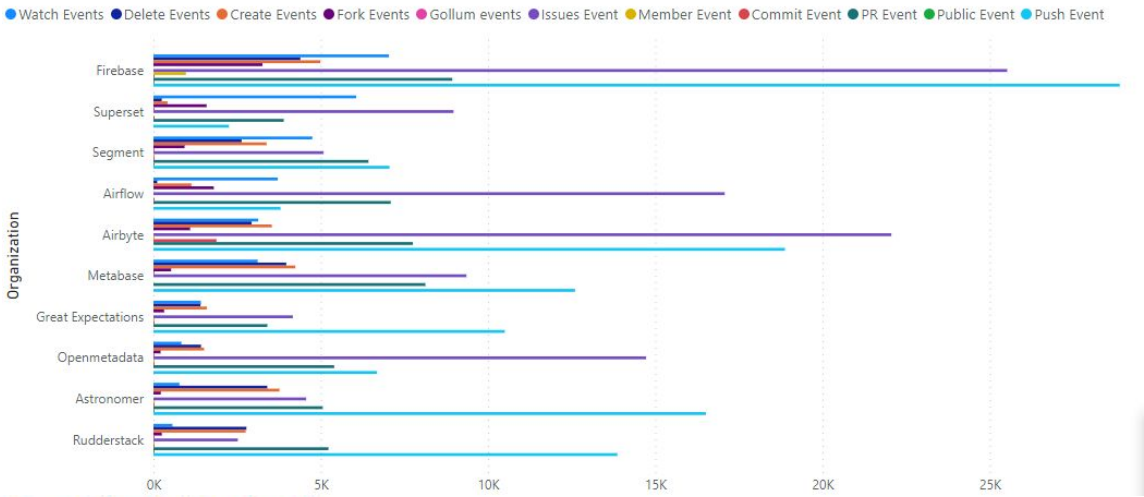
Power BI



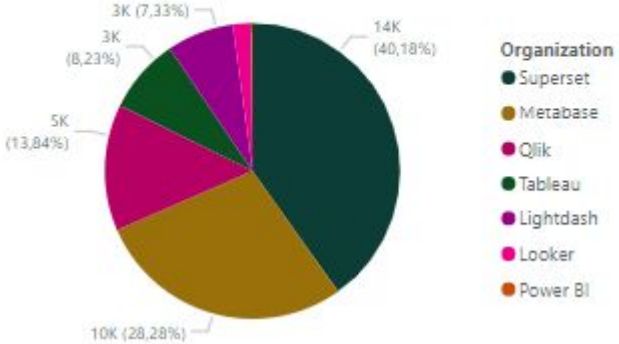
Create one-to-many relationships between tables

Power BI

Back to report | TOP 10 ORGANIZATIONS BASED ON THE NUMBER OF EVENTS



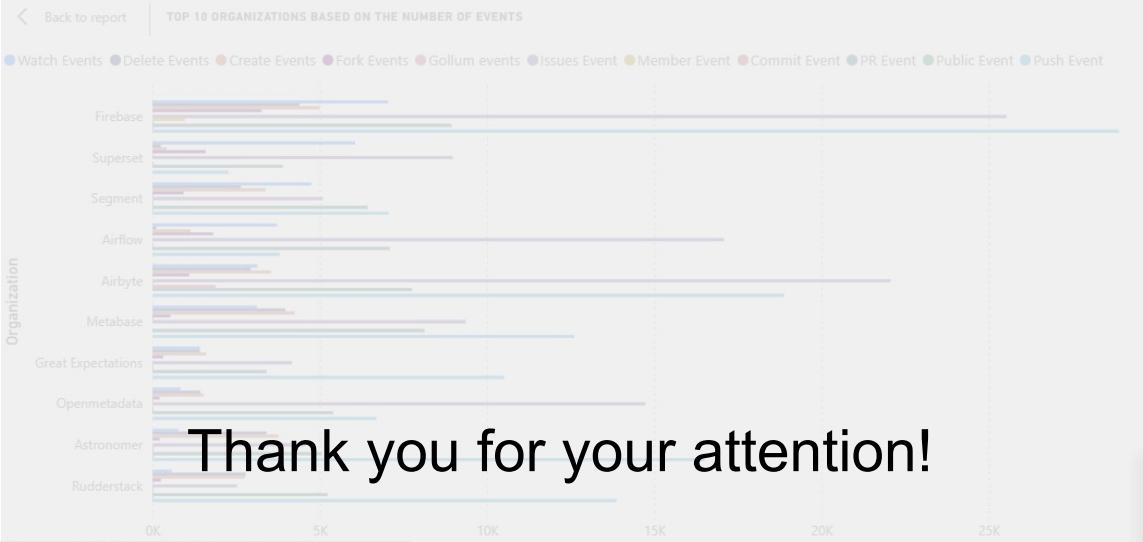
Distribution of users between Organizations from the Analytics Category



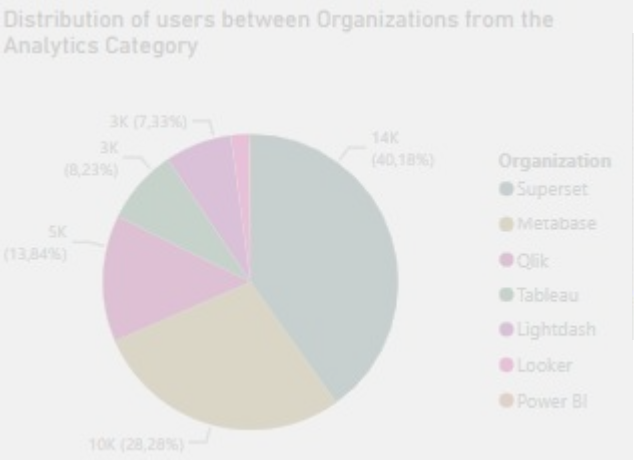
Organization	PR events	Watch events	Commit Comment event	Total event count	User Count
Lightdash	218,22	100,56	0,00	1618,56	123,33
Looker	0,29	0,21	0,01	10,30	1,33
Metabase	903,00	346,00	0,00	4660,56	539,00
Power BI	0,00	1,64	0,00	2,50	2,32
Qlik	11,34	0,25	0,02	60,42	3,56
Superset	432,67	673,22	1,22	2606,89	1037,00
Tableau	2,99	1,90	0,00	16,33	5,46
Total	**18,18**	**10,60**	**0,02**	**105,12**	**18,23**

Anita

Azure



Thank you for your attention!



Organization	PR events	Watch events	Commit Comment event	Total event count	User Count
Lightdash	218,22	100,56	0,00	1618,56	123,33
Looker	0,29	0,21	0,01	10,30	1,33
Metabase	903,00	346,00	0,00	4660,56	539,00
Power BI	0,00	1,64	0,00	2,50	2,32
Qlik	11,34	0,25	0,02	60,42	3,56
Superset	432,67	673,22	1,22	2606,89	1037,00
Tableau	2,99	1,90	0,00	16,33	5,46
Total	**18,18**	**10,60**	**0,02**	**105,12**	**18,23**

Anita

Azure