

0. Reference

- <https://datahubproject.io/>
- <https://tech.socarcorp.kr/data/2022/02/25/data-discovery-platform-01.html>
- <https://blog.banksalad.com/tech/the-starting-of-datadiscoveryplatform-era-in-banksalad/>

1. Datahub

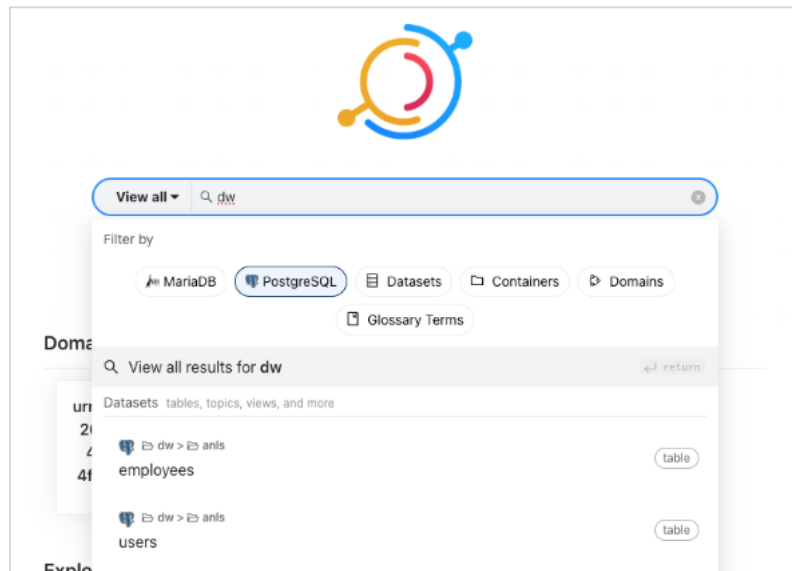
| 메타데이터 관리, 데이터 검색 및 데이터 거버넌스를 간소화하도록 설계된 최신 데이터 카탈로그

메타데이터 관리 플랫폼은 사용자들이 빠르게 변화하는 데이터의 복잡성을 관리하고 **전사적으로 관리되는** 데이터의 가치를 활용할 수 있도록 구축되었다.

1.1 기본 기능

(1) 데이터의 효율적 검색 및 이해

| 테이블 / 컬럼 / 데이터 소스 / 도메인 / 용어 검색



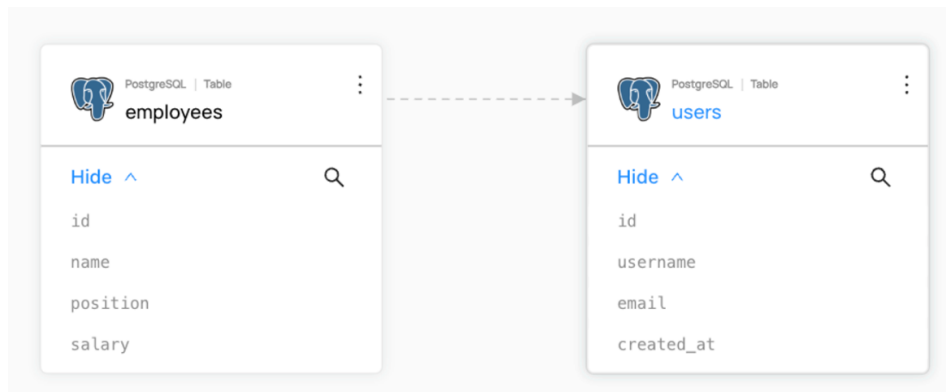
[그림 1] 메인페이지 검색

Field	Description	Tags	Glossary Terms
id Number (Primary Key)	유저ID	(edited)	
username String	유저명	(edited)	
email String	이메일	(edited)	
created_at Time (Nullable)	생성일시	(edited)	

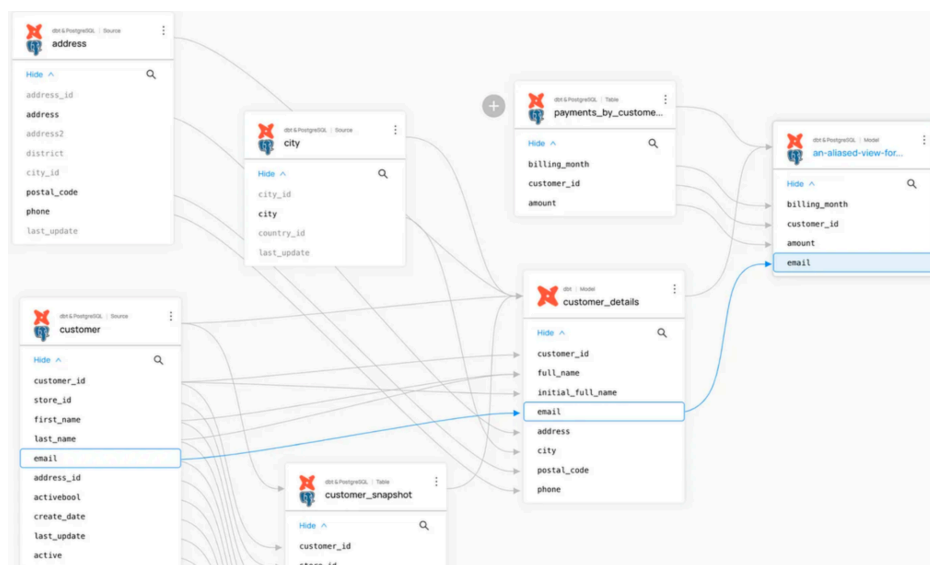
[그림 2] 테이블 메타정보 상세 내역

(2) 데이터 계보(Data Lineage) 추적

| table-level lineage / column-level lineage 2 가지 계보 추적을 지원



[그림 3] table-level 계보 추적



[그림 4] column-level 계보 추적

(3) 데이터 프로파일링 (Data Profiling)

| 컬럼 값 데이터 프로파일링을 지원

** 데이터 품질에서의 데이터 프로파일링은 데이터 현황 분석을 위한 자료수집과 오류 또는 잠재적 이슈를 찾아내는 방법입니다.

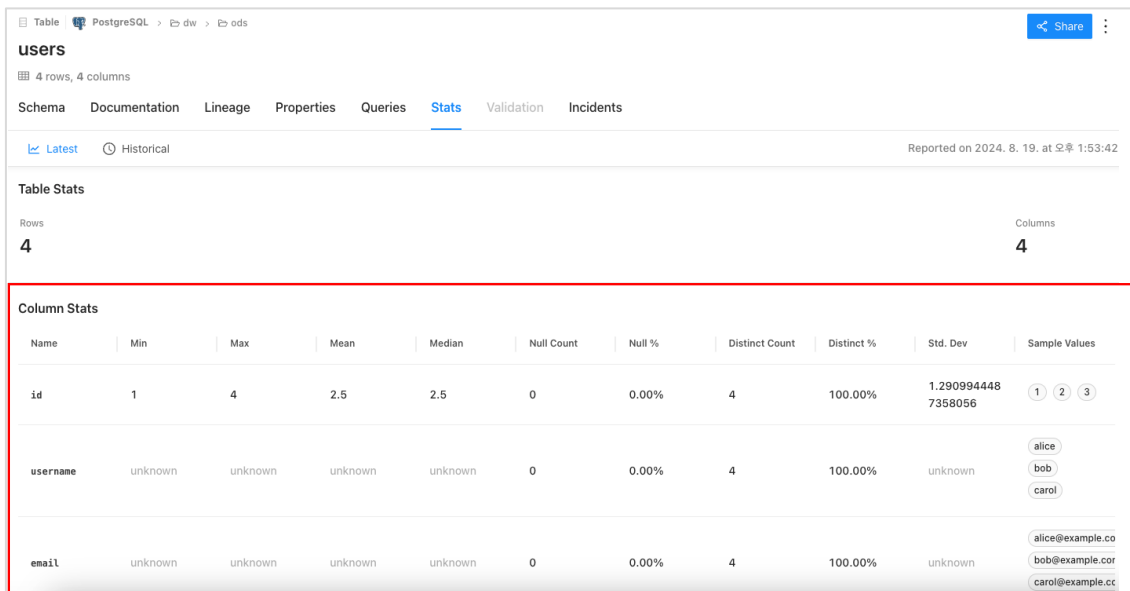


Table: PostgreSQL > dw > ods

users

4 rows, 4 columns

Schema Documentation Lineage Properties Queries **Stats** Validation Incidents

Latest Historical Reported on 2024. 8. 19. at 오후 1:53:42

Table Stats

Rows: 4 Columns: 4

Column Stats

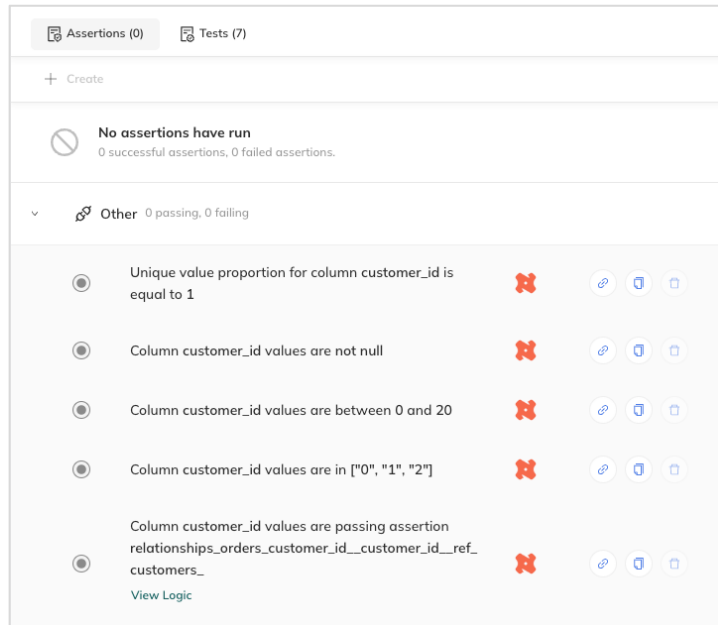
Name	Min	Max	Mean	Median	Null Count	Null %	Distinct Count	Distinct %	Std. Dev	Sample Values
id	1	4	2.5	2.5	0	0.00%	4	100.00%	1.290994448 7358056	1 2 3
username	unknown	unknown	unknown	unknown	0	0.00%	4	100.00%	unknown	alice bob carol
email	unknown	unknown	unknown	unknown	0	0.00%	4	100.00%	unknown	alice@example.co bob@example.cor carol@example.cc

[그림 5] 데이터 컬럼 프로파일링의 결과 모니터링

(4) 데이터 계약 (Data Contracts) 설정

| 데이터 자산의 생산자와 소비자 간의 계약으로, 데이터 품질에 대한 약속 역할을 수행. 데이터 품질관리에 대한 데이터 오너십(ownership) 부여 및 지속적 관리 지원

** 오너십 소유자가 지정한 데이터 품질 검사의 기대 값이 실제 프로파일링 결과와 비교하여 문제가 있다면 빠르게 데이터 품질 문제를 해결할 수 있도록 역할을 관리합니다.



[그림 6] 데이터 계약 예시

2. 메타데이터 수집 방법 (Data Ingestion)

2.1 기본 개념 용어

(1) Recipes

| 메타데이터 수집을 위한 주요 구성 파일(yaml 또는 json). 데이터를 어디에서 가져올지(source)와 어디에 넣을지(sink)를 수집 정보를 기록한 스크립트 파일

```

recipes.yaml > ...
1  # The simplest recipe that pulls metadata from MSSQL and puts it into DataHub
2  # using the Rest API.
3  source:
4    type: mssql
5    config:
6      username: sa
7      password: ${MSSQL_PASSWORD}
8      database: DemoData
9  # sink section omitted as we want to use the default datahub-rest sink
10 sink:
11   type: "datahub-rest"
12   config:
13     server: "http://localhost:8080"

```

[그림 7] yaml 포맷 Recipes 작성 예시

(2) Source

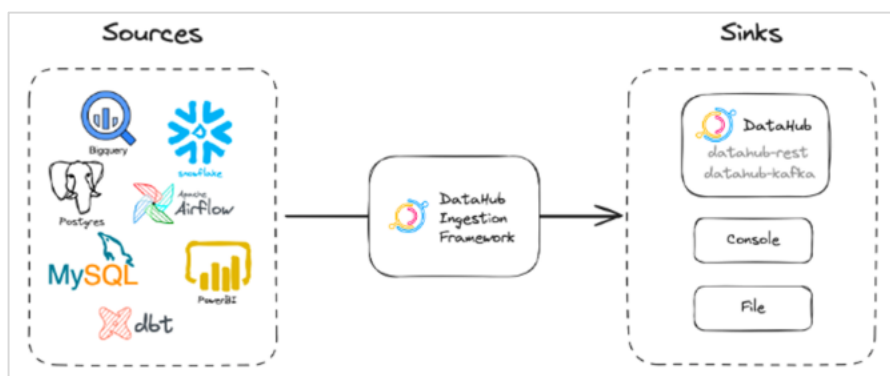
| 메타데이터를 추출하는 데이터 소스

(3) Sink

| Source 에서 추출한 메타데이터의 주입 대상 (datahub 메타 DB 수집 저장소)

2.2 . Data Ingestion

| 여러 데이터베이스 소스로부터 메타 정보를 모아 메타 관리 시스템인 데이터허브로 주입하는 과정



[그림 8] Data Sources – Data Ingestion – Data Sinks 플로우

(1) Web UI Ingestion

| 데이터허브 Frontend Web 에서 데이터 소스에 대한 정보를 직접 입력하는 방법

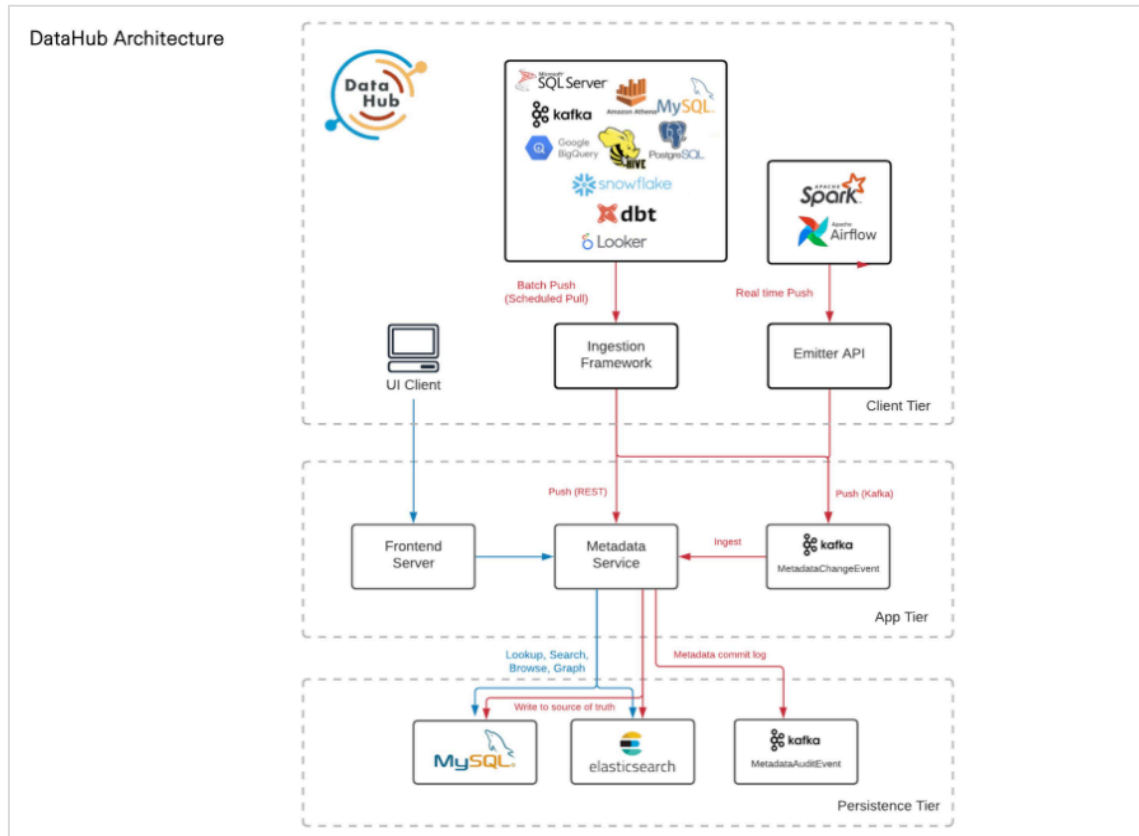
- 장점
 - 웹 UI 에서 소스의 연결 정보를 쉽게 입력 가능
 - 수집할 데이터 소스를 정보를 입력 후 실행 시 필요한 라이브러리를 자동 설치
 - 주입 옵션(특정 스키마, 테이블 등의 Regex 패턴, 컬럼 프로파일링 여부 등)의 체크 옵션 제공
 - datahub 내부에서 메타 데이터를 소싱할 배치 스케줄링을 만들 수 있음 (예: 일별 **시 **분).
- 단점
 - 스케줄러와 연계해 배치가 끝난 직 후 트리거를 걸 수 없음 (Pull 방식으로 메타 수집)

(2) Datahub API Ingestion

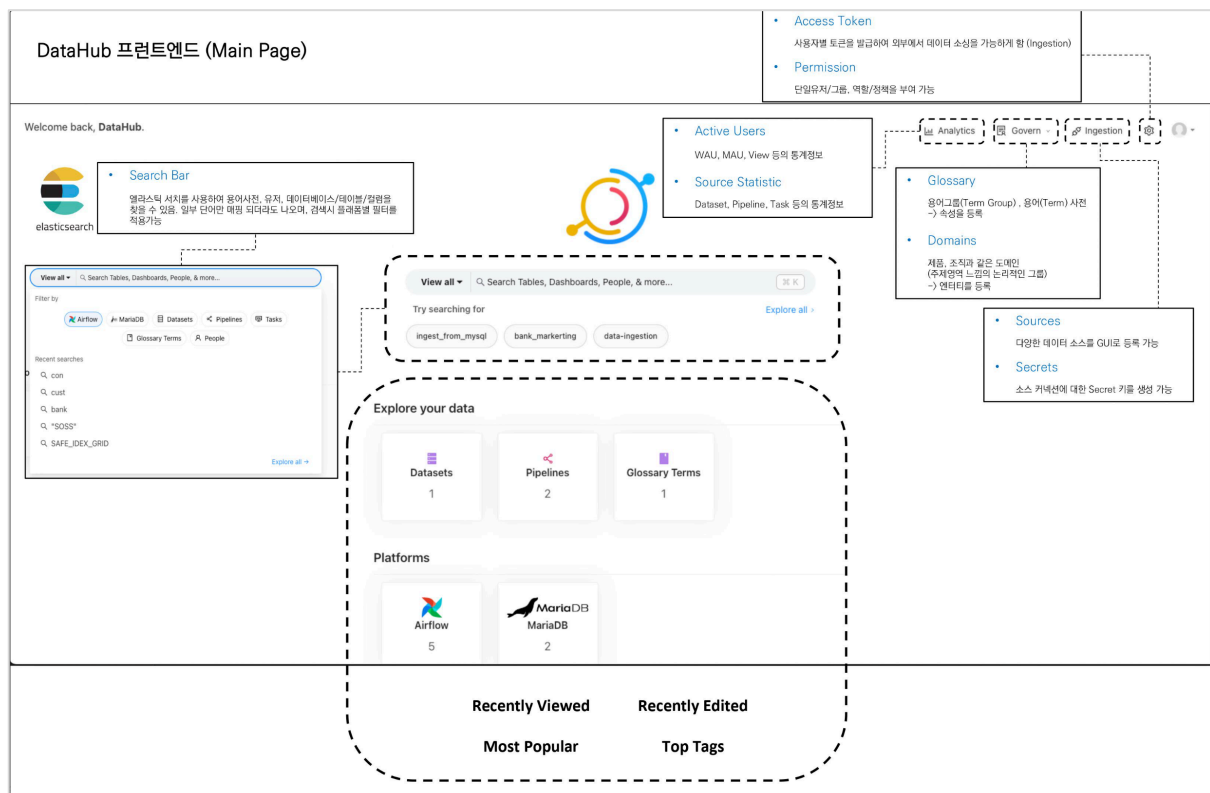
| Recipes 를 API 나 CLI 로 주입하는 방법

- 장점
 - datahub 서버 외부에서도 메타 데이터 수집을 직접 트리거 가능 (Push 방식으로 메타 수집)
 - 유효기간이 있는 토큰을 발급 해야만 사용 가능하도록 기본적인 보안 지원
- 단점
 - Rest API 를 사용하기 위한 중/고급 파이썬 프로그래밍 능력이 요구. 아직은 레퍼런스가 많이 없기 때문에 자체적인 개발 환경에 맞춰 데이터허브 SDK 를 스케줄러(예: 에어플로우)와 연계하기 위해서는 전반적인 파이프라인에 대한 이해 역량이 필요할 것으로 예상
 - Datahub API 의 종속 라이브러리가 많기 때문에 별도의 컨테이너로 종속성 관리가 필요함

3. Appendix



[그림 9] Datahub Architecture



[그림 10] Datahub Web UI 메인페이지 기능 요약