

WebToon Grepp

국내 웹툰 크롤링/분석 프로젝트

WT Grepp | 이우람 정용선 최은주

Contents

01

프로젝트 배경

프로젝트 주제 선정 배경

02

팀원 및 역할

팀원 구성 및 역할 분담

03

기술 스택 및 아키텍처

활용 기술 및 인프라 아키텍처

04

데이터 파이프라인

데이터 파이프라인 단계별 흐름

05

대시보드 및 홈페이지

대시보드 및 홈페이지 구현 예시

06

개선 사항

기존 구조와 개선된 구조 비교

07

기대 효과

프로젝트를 통해 얻을 수 있는 기대 성과

08

아쉬운 점

프로젝트 회고 및 개선점

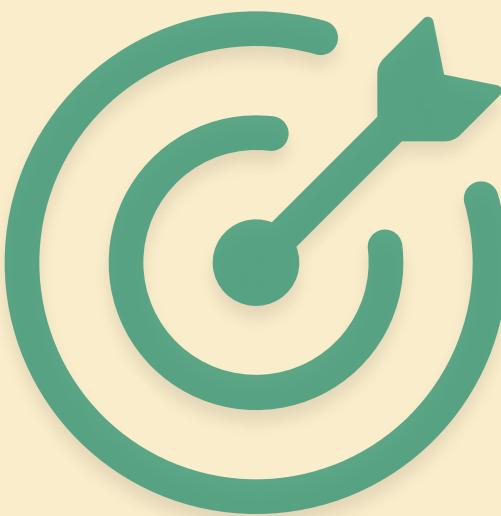
01 프로젝트 배경

웹툰 시장의 성장



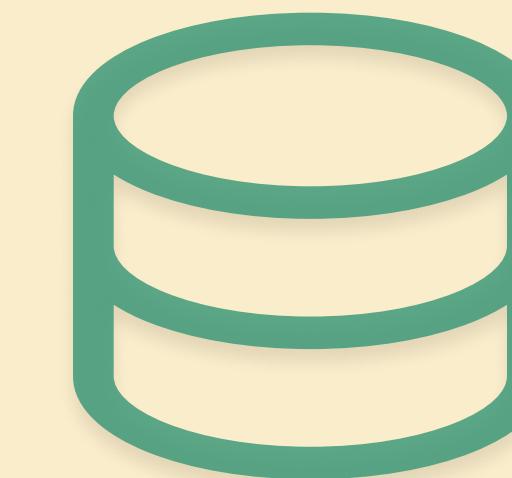
웹툰은 최근 몇 년 간 빠르게 성장한 콘텐츠 분야로, 웹툰 시장의 성장에 따라 소비자의 관심과 소비 패턴을 분석하는 것이 매우 중요해짐

사용자 맞춤형 추천



웹툰 플랫폼들이 점차 개인화된 콘텐츠 추천 시스템을 도입하고 있는데, 이를 위해서 사용자의 소비 패턴을 분석해 인기이는 장르나 특성을 파악해야 함

데이터 기반 의사 결정



웹툰의 조회수, 댓글 수, 좋아요 수 등의 분석을 통해 어떤 콘텐츠가 인기 있는지 파악함으로써 효과적인 마케팅 전략을 수립할 수 있음

02 팀원 및 역할

- 카카오 웹툰 크롤링
- 데일리 데이터 ETL
- 데이터 파이프라인 관리
- AWS 리소스 관리
- 홈페이지 제작

이우람

- 네이버 웹툰 크롤링
- 과거 데이터 ETL
- 크롤링 속도 개선

정용선

- 실사용 데이터 ELT
- 대시보드 제작

최은주

03 기술 스택 및 아키텍쳐

데이터 수집(크롤링)

- Python, Requests

데이터 파이프라인 및 처리

- 워크플로우 자동화
 - Apache Airflow
데이터 처리 및 변환
 - Apache Spark
데이터 변환 및 모델링
 - dbt

인프라 및 저장소

- 서버 인프라
 - AWS EC2
데이터 저장소
 - AWS S3
데이터 웨어하우스
 - AWS Redshift
데이터 베이스
 - PostgreSQL

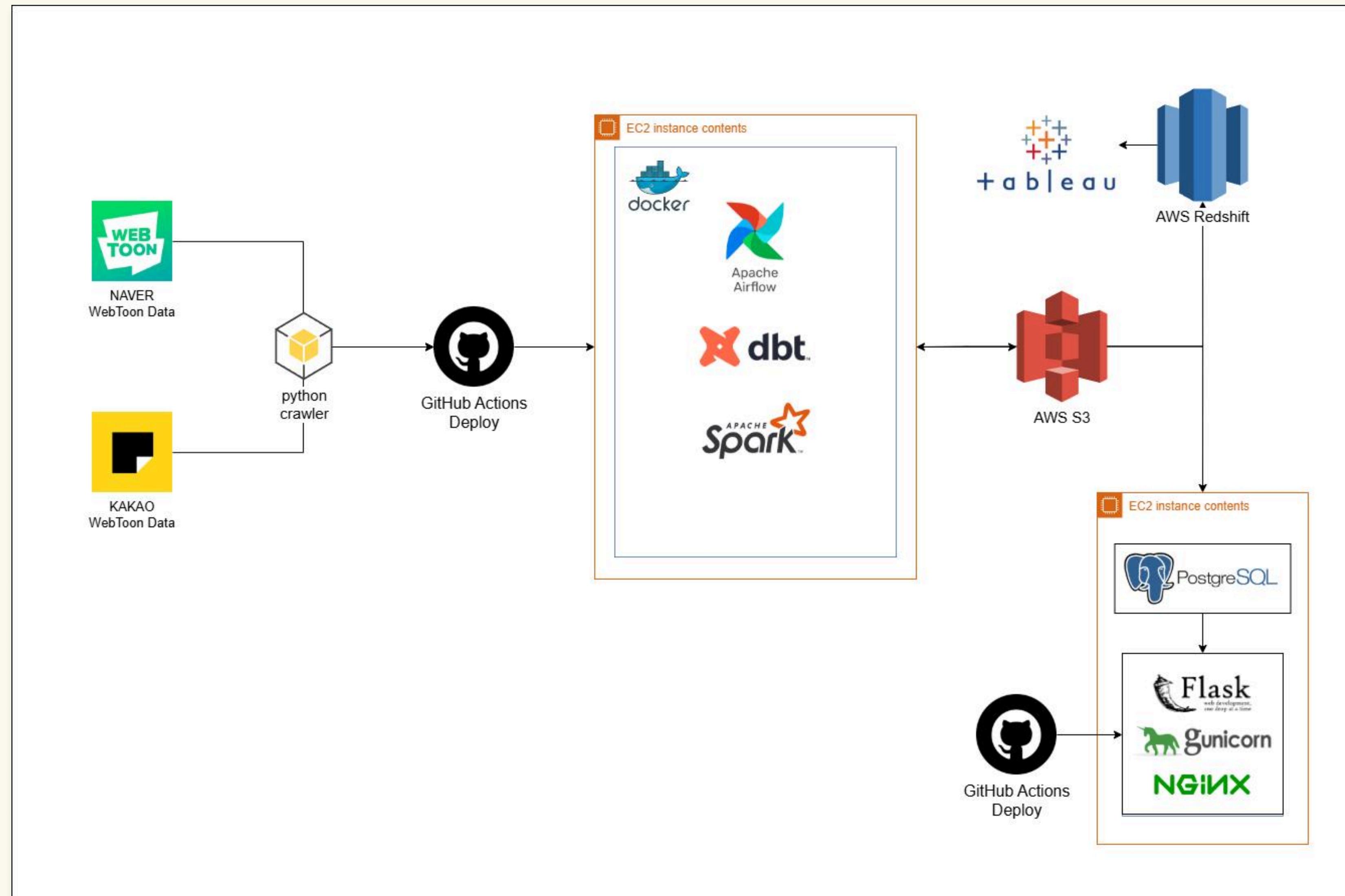
데이터 시각화 및 애플리케이션

- 데이터 시각화
 - Tableau
웹 애플리케이션
 - Flask

배포 및 운영

- 컨테이너화
 - Docker
웹 서버 및 리버스 프록시
 - Nginx
버전 관리 및 CI/CD
 - GitHub, GitHub Actions
WSGI 서버
 - Gunicorn

03 기술 스택 및 아키텍쳐



04 데이터 파이프라인

데이터 소스

네이버 웹툰



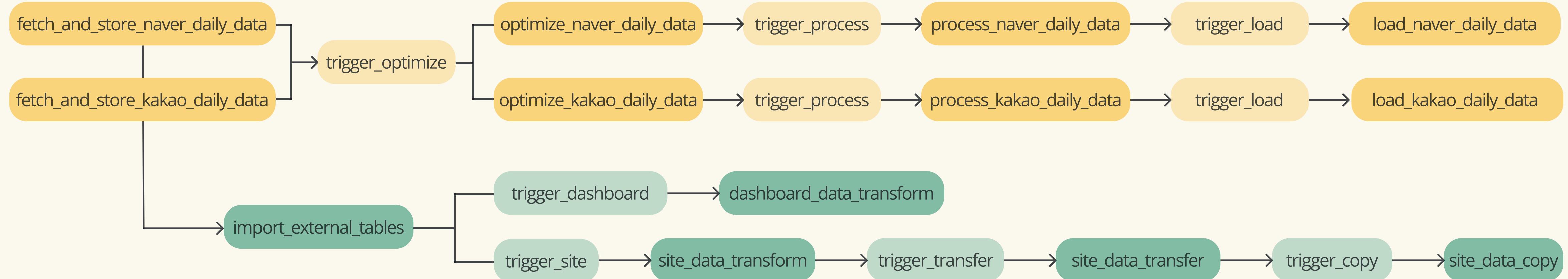
네이버 웹툰의 연재 중 작품 목록,
완결 작품 목록, 작품 정보,
작품 회차 목록, 회차 정보, 댓글 정보,
좋아요 정보 API 활용

카카오 웹툰

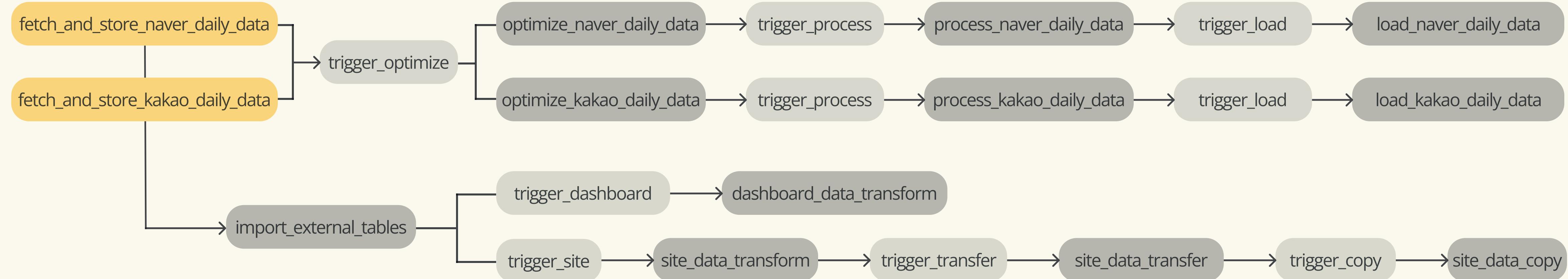


카카오 웹툰의 연재 중 작품 목록(신작 목록,
요일별 웹툰), 완결 작품 목록, 작품 정보,
작품 회차 목록, 댓글 정보,
회차 정보 API 활용

04 데이터 파이프라인



04 데이터 파이프라인



네이버와 카카오 웹툰 데이터 수집

01 데이터 크롤링

사용 기술: Python, Requests

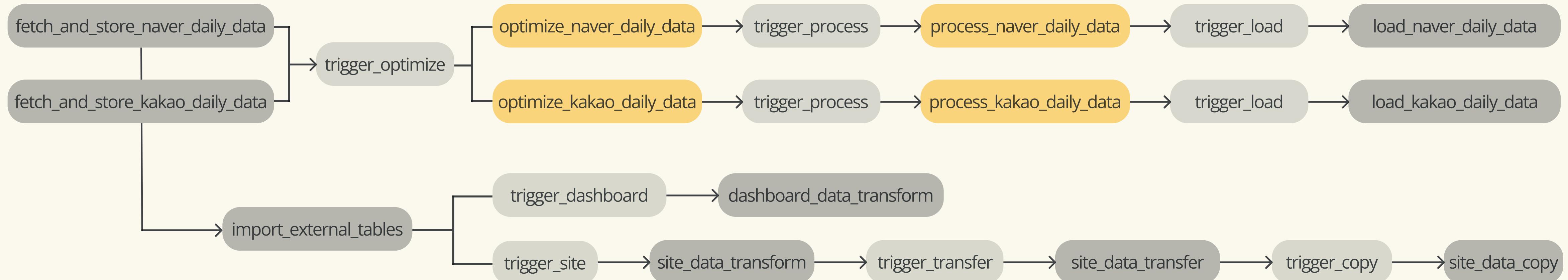
세부 과정

- 네이버와 카카오 웹툰에서 웹툰의 제목, 회차별 조회수, 댓글 수, 좋아요 수 등의 데이터 크롤링
- 수집된 원시 데이터는 S3에 저장

저장형식

- S3 저장 경로: raw/ 폴더 안에 웹툰별, 회차별 정보가 JSON 형식으로 저장

04 데이터 파이프라인



네이버와 카카오 웹툰 데이터 수집

01 데이터 크롤링

사용 기술: Python, Requests

세부 과정

- 네이버와 카카오 웹툰에서 웹툰의 제목, 회차별 조회수, 댓글 수, 좋아요 수 등의 데이터 크롤링
- 수집된 원시 데이터는 S3에 저장

저장형식

- S3 저장 경로: raw/ 폴더 안에 웹툰별, 회차별 정보가 JSON 형식으로 저장

원시 데이터의 최적화 및 정제

02 데이터 정제 및 변환

사용 기술: Apache Spark, PySpark

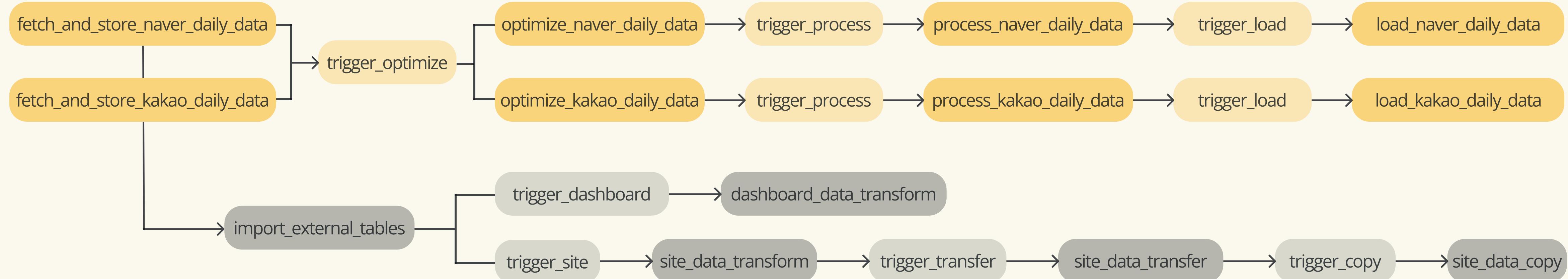
세부 과정

- 여러 개의 파일로 나눈 원시 데이터를 통합
- 중복된 웹툰 정보를 제거하고 결측값 처리
- 효율적인 분석을 위해 Parquet 형식으로 저장

저장형식

- 최적화된 데이터는 optimized/ 폴더에 Parquet 형식으로 저장
- 정제된 데이터는 processed/ 폴더에 Parquet 형식으로 저장

04 데이터 파이프라인



네이버와 카카오 웹툰 데이터 수집

01 데이터 크롤링

사용 기술: Python, Requests

세부 과정

- 네이버와 카카오 웹툰에서 웹툰의 제목, 회차별 조회수, 댓글 수, 좋아요 수 등의 데이터 크롤링
- 수집된 원시 데이터는 S3에 저장

저장형식

- S3 저장 경로: raw/ 폴더 안에 웹툰별, 회차별 정보가 JSON 형식으로 저장

원시 데이터의 최적화 및 정제

02 데이터 정제 및 변환

사용 기술: Apache Spark, PySpark

세부 과정

- 여러 개의 파일로 나눈 원시 데이터를 통합
- 중복된 웹툰 정보를 제거하고 결측값 처리
- 효율적인 분석을 위해 Parquet 형식으로 저장

저장형식

- 최적화된 데이터는 optimized/ 폴더에 Parquet 형식으로 저장
- 정제된 데이터는 processed/ 폴더에 Parquet 형식으로 저장

데이터 수집, 변환, 적재 작업을 자동화

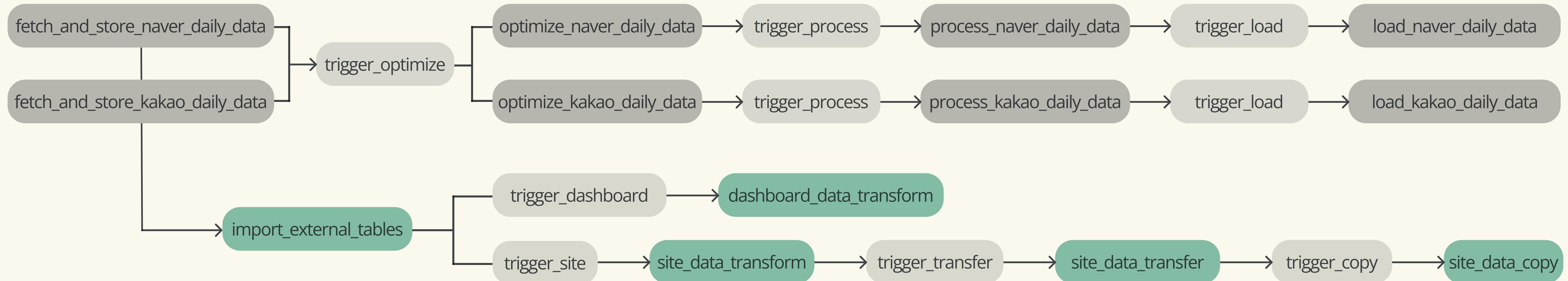
03 데이터 적재 자동화

사용 기술: Apache Airflow

DAG 흐름

- 각 DAG는 정해진 주기에 따라 실행되며, 순차적으로 작업 수행
 - fetch_and_store: 데이터 크롤링
 - optimize: 데이터 최적화
 - process: 데이터 분석 준비
 - load: 데이터 적재

04 데이터 파이프라인



네이버와 카카오 웹툰 데이터 수집

01 데이터 크롤링

사용 기술: Python, Requests

세부 과정

- 네이버와 카카오 웹툰에서 웹툰의 제목, 회차별 조회수, 댓글 수, 좋아요 수 등의 데이터 크롤링
- 수집된 원시 데이터는 S3에 저장

저장형식

- S3 저장 경로: raw/ 폴더 안에 웹툰별, 회차별 정보가 JSON 형식으로 저장

원시 데이터의 최적화 및 정제

02 데이터 정제 및 변환

사용 기술: Apache Spark, PySpark

세부 과정

- 여러 개의 파일로 나눈 원시 데이터를 통합
- 중복된 웹툰 정보를 제거하고 결측값 처리
- 효율적인 분석을 위해 Parquet 형식으로 저장

저장형식

- 최적화된 데이터는 optimized/ 폴더에 Parquet 형식으로 저장
- 정제된 데이터는 processed/ 폴더에 Parquet 형식으로 저장

데이터 수집, 변환, 적재 작업을 자동화

03 데이터 적재 자동화

사용 기술: Apache Airflow

DAG 흐름

- 각 DAG는 정해진 주기에 따라 실행되며, 순차적으로 작업 수행
 - fetch_and_store: 데이터 크롤링
 - optimize: 데이터 최적화
 - process: 데이터 분석 준비
 - load: 데이터 적재

데이터를 모델링하고 다양한 분석 지표 생성

04 데이터 모델링 및 분석

사용 기술: dbt (Data Build Tool)

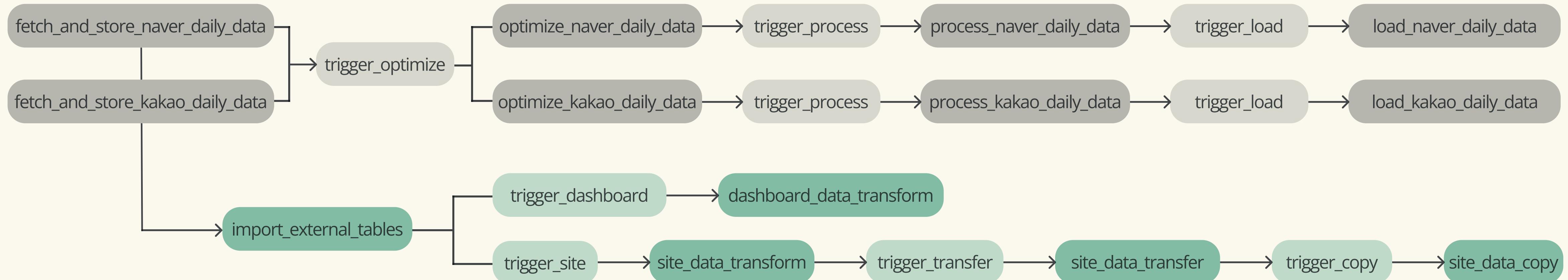
세부 과정

- 웹툰의 장르, 조회수, 좋아요 수 등을 기준으로 데이터 모델링
- 인기 웹툰, 장르별 조회수 등 분석 지표 도출

DAG 흐름

- import_external_table: 외부 테이블을 내부 테이블로 변환
- data_transform: dbt 모델링을 진행하고 테스트 후, Redshift에 적재
- data_transfer: 변환된 데이터를 S3 업로드
- data_copy: S3 데이터를 PostgreSQL로 이동

04 데이터 파이프라인



네이버와 카카오 웹툰 데이터 수집

01 데이터 크롤링

사용 기술: Python, Requests

세부 과정

- 네이버와 카카오 웹툰에서 웹툰의 제목, 회차별 조회수, 댓글 수, 좋아요 수 등의 데이터 크롤링
- 수집된 원시 데이터는 S3에 저장

저장형식

- S3 저장 경로: raw/ 폴더 안에 웹툰별, 회차별 정보가 JSON 형식으로 저장

원시 데이터의 최적화 및 정제

02 데이터 정제 및 변환

사용 기술: Apache Spark, PySpark

세부 과정

- 여러 개의 파일로 나눈 원시 데이터를 통합
- 중복된 웹툰 정보를 제거하고 결측값 처리
- 효율적인 분석을 위해 Parquet 형식으로 저장

저장형식

- 최적화된 데이터는 optimized/ 폴더에 Parquet 형식으로 저장
- 정제된 데이터는 processed/ 폴더에 Parquet 형식으로 저장

데이터 수집, 변환, 적재 작업을 자동화

03 데이터 적재 자동화

사용 기술: Apache Airflow

DAG 흐름

- 각 DAG는 정해진 주기에 따라 실행되며, 순차적으로 작업 수행
 - fetch_and_store: 데이터 크롤링
 - optimize: 데이터 최적화
 - process: 데이터 분석 준비
 - load: 데이터 적재

데이터를 모델링하고 다양한 분석 지표 생성

04 데이터 모델링 및 분석

사용 기술: dbt (Data Build Tool)

세부 과정

- 웹툰의 장르, 조회수, 좋아요 수 등을 기준으로 데이터 모델링
- 정확성과 유효성 검증을 위한 테스트 수행

DAG 흐름

- import_external_table: 외부 테이블을 내부 테이블로 변환
- data_transform: dbt 모델링을 진행하고 테스트 후, Redshift에 적재
- data_transfer: 변환된 데이터를 S3 업로드
- data_copy: S3 데이터를 PostgreSQL로 이동

분석된 데이터를 시각화한 대시보드 제공

05 데이터 시각화 및 웹 구축

사용 기술: Tableau, Flask, Nginx, Gunicorn 등

대시보드 세부 과정

- 웹툰의 조회수, 좋아요 수, 댓글 수 등을 시각화하여 대시보드에 표시
- Redshift에서 실시간 데이터를 연결하여 대시보드에서 시각화

웹 애플리케이션 세부 과정

- 웹툰 목록, 장르별/요일별 웹툰 선택 기능 제공
- 분석된 데이터를 웹 애플리케이션에서 시각화하여 사용자에게 제공

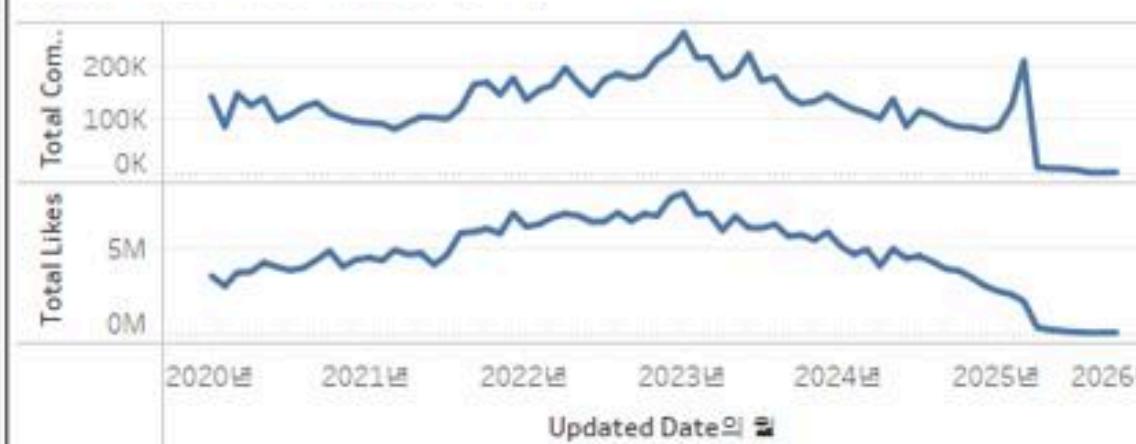
05 대시보드 및 홈페이지 예시



Naver 대시보드



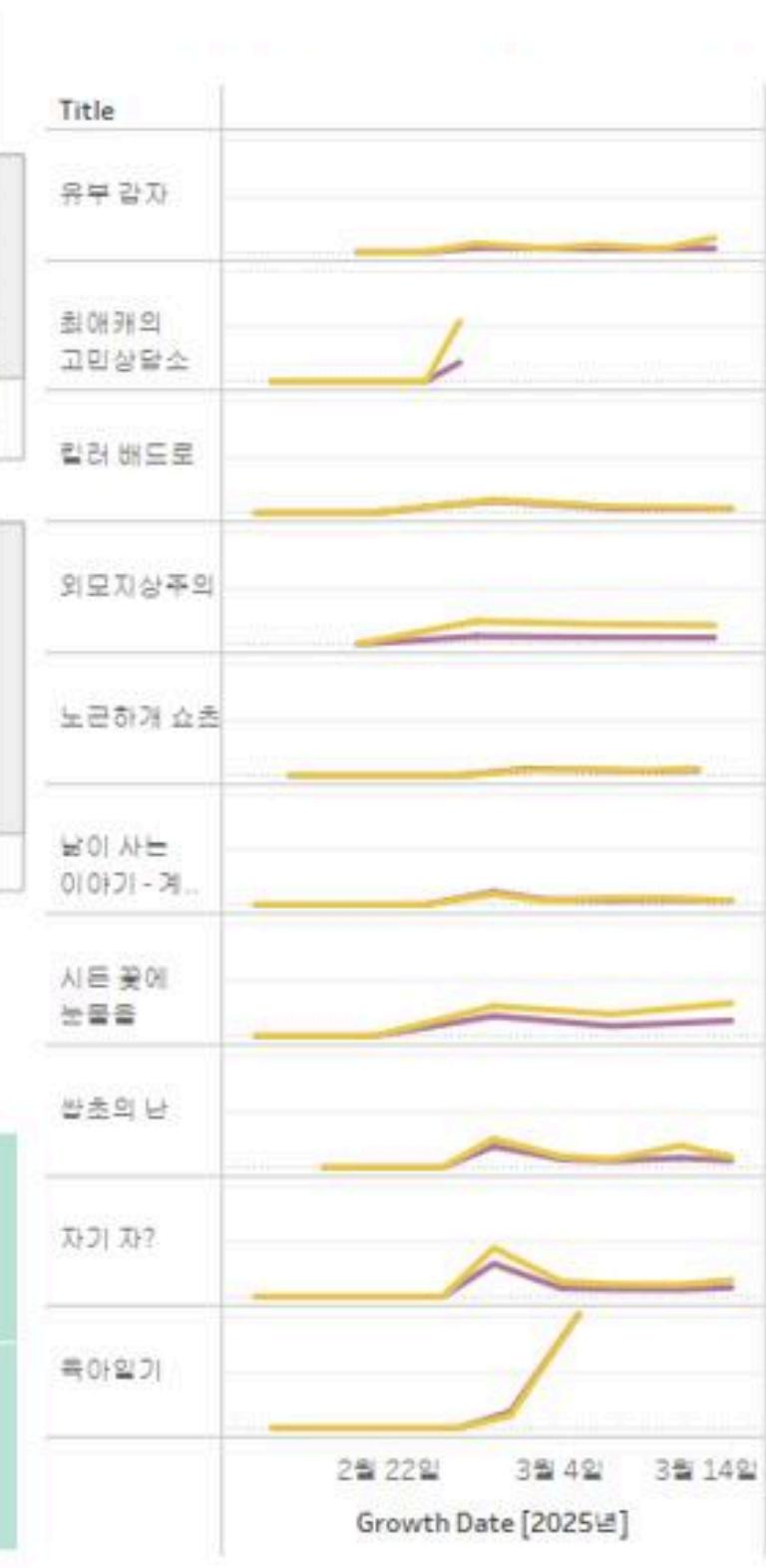
웹툰 시장 사용자 반응 추세



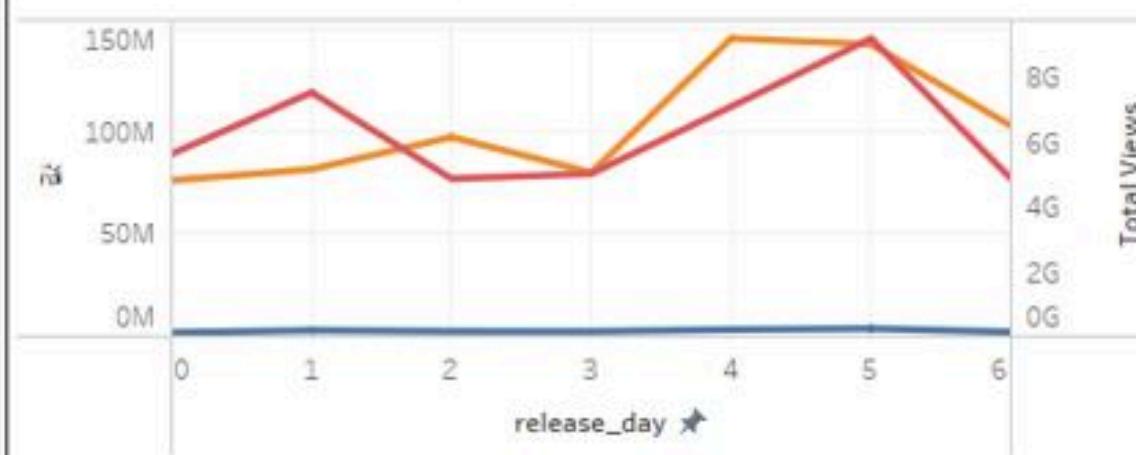
성장률 TOP 10

Author	Title	Start Date	Total Com.	Total Likes
이경민 / 총판회	우리 암 사귀어!!	2025	601,200	2,629,125
자까	죽아일기	2025	260,928	2,339,712
혜통 / 영영	자기 자?	2025	220,549	1,191,827
반초	반초의 날	2025	164,444	937,125
개	시든 꽃에 노를 품	2025	175,056	813,824
서나래	날이 사는 이야기 - 계속되는 미미한 일	2025	68,208	573,398
풀끼	노근하게 쇼츠	2025	46,080	387,036

성장률 TOP 10 추세 모니터링



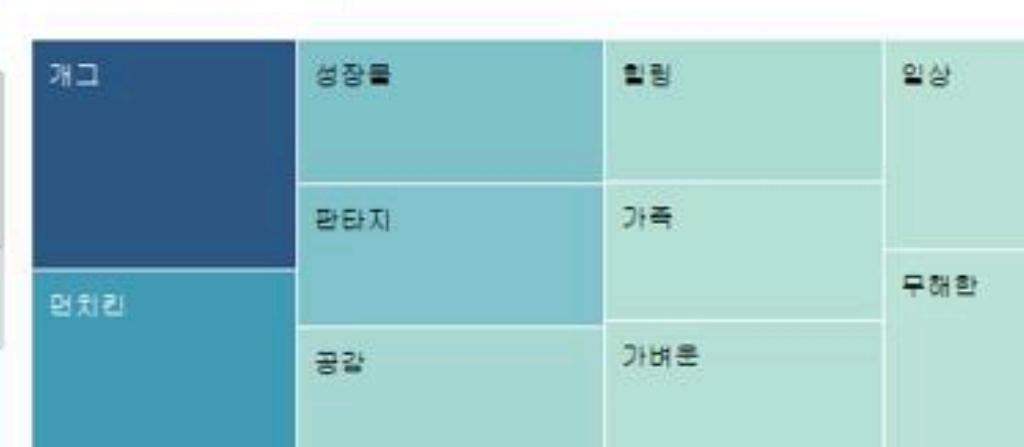
웹툰 시장 요일별 추세



Naver 평균 조회수 TOP 10



Naver 인기 장르



목표

분석된 데이터를 시각화해
Tableau 대시보드에서 제공

사용 기술: Tableau

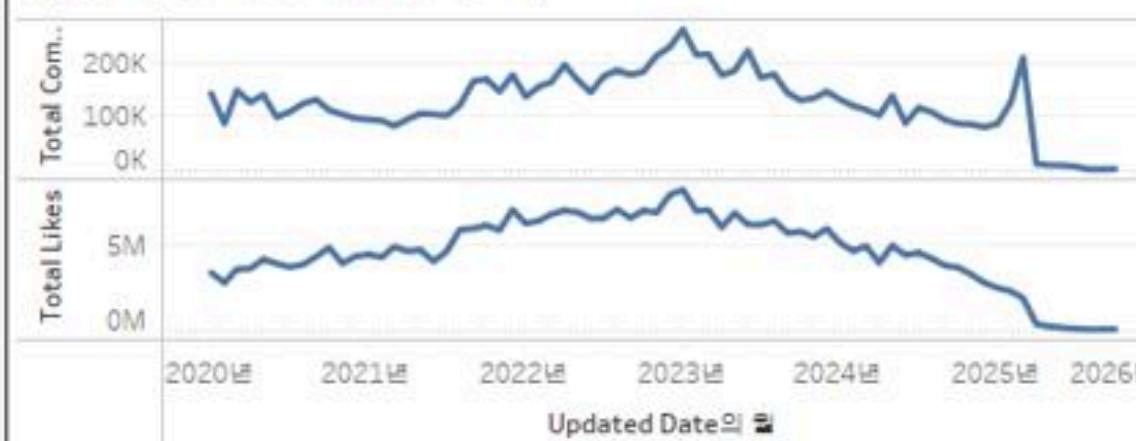
05 대시보드 및 홈페이지 예시



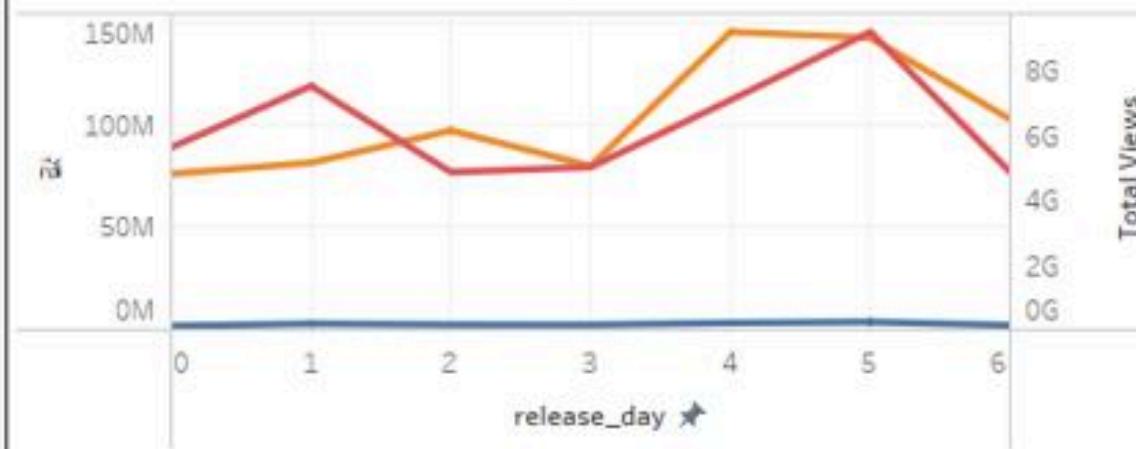
Kakao 대시보드



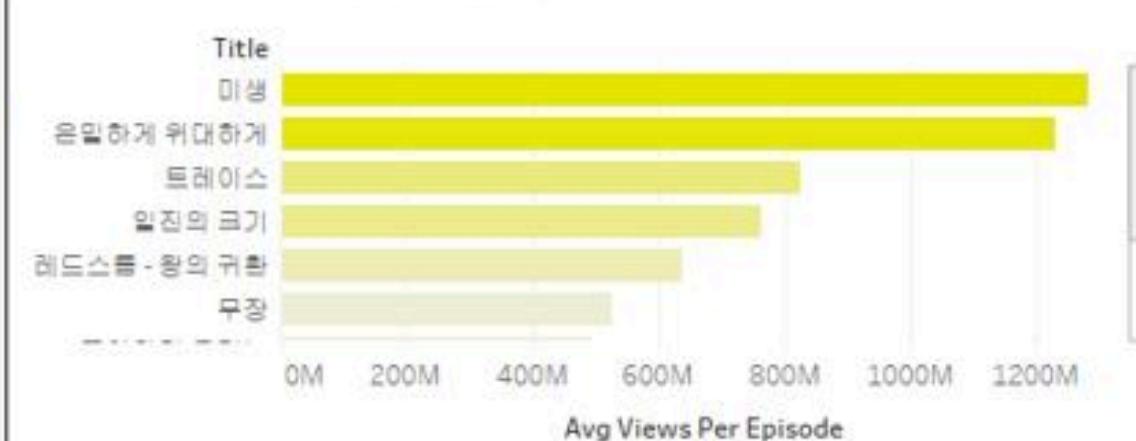
웹툰 시장 사용자 반응 추세



웹툰 시장 요일별 추세



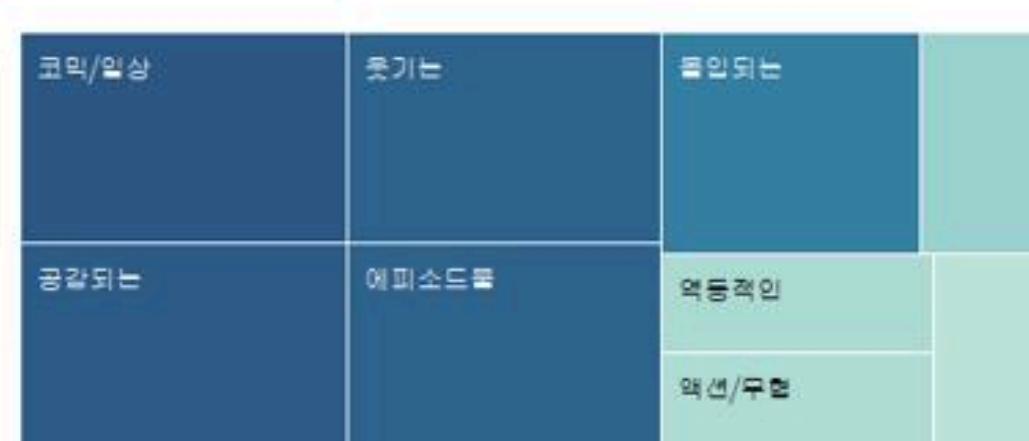
Kakao 평균 조회수 TOP 10



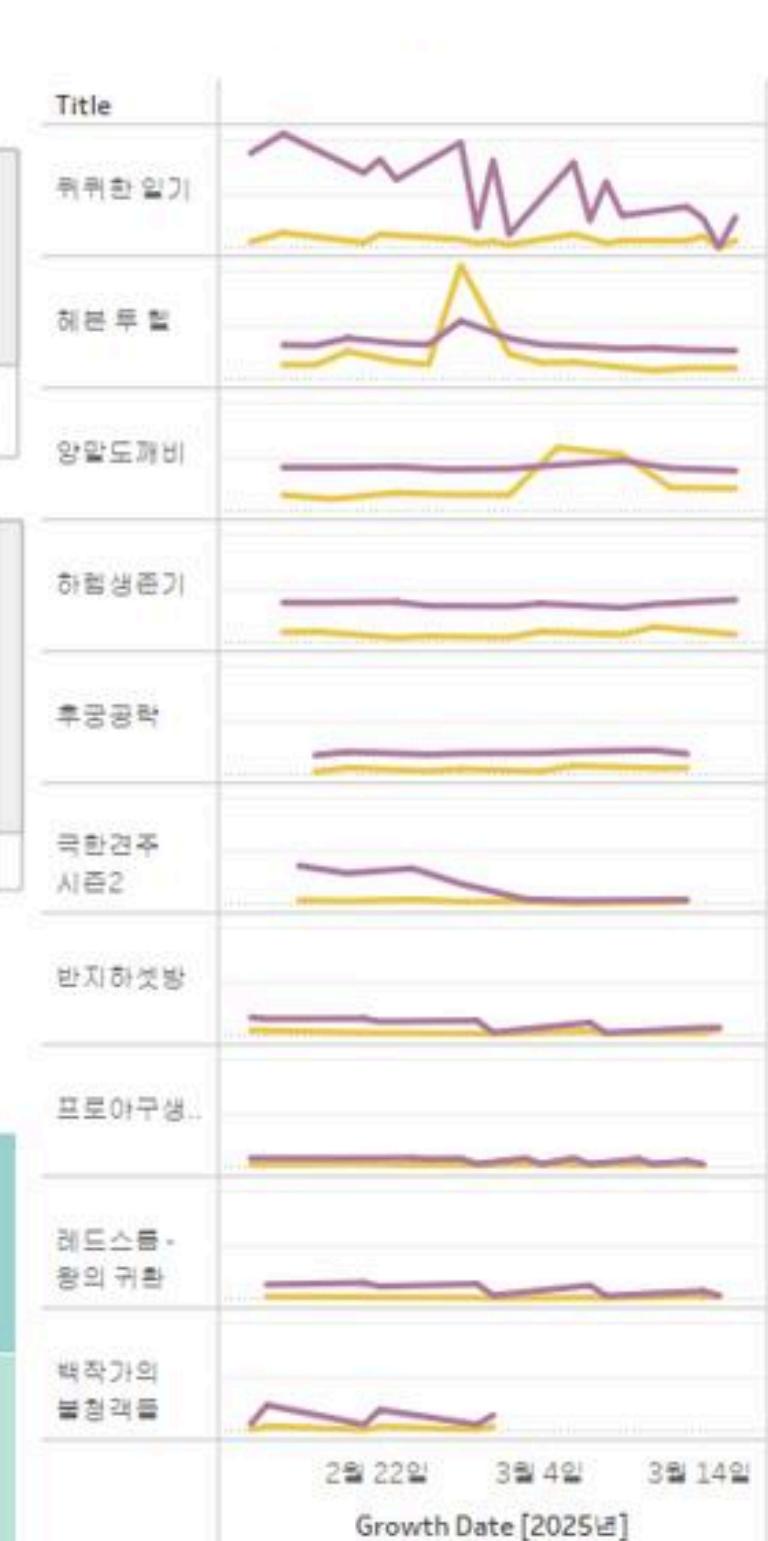
성장률 TOP 10

Author	Title	Start Date	Total Com..	Total Likes
2B/2B/카카오웹툰 스...	퀴퀴한 일기	2025	6,324,626	448,453,145
김종훈/김종훈/카카오...	해븐 투 힐	2025	2,527,524	70,363,890
노경한/알현/카카오웹...	레드스톰 - 王의 귀환	2025	2,301,600	54,213,175
만화상/만화상/카카오...	왕말도깨비	2025	1,778,616	43,619,472
류인/사설/야식먹는중...	트끼와 총표벌의 공생관계	2025	519,425	40,938,550
생일/포야/풀이수/에...	세이린: 악당과 계약 가족이...	2025	683,172	49,237,884
이동률(Redice Studio)...	월월	2025	237,275	38,595,800

Kakao 인기 장르



성장률 TOP 10 추세 모니터링



목표

분석된 데이터를 시각화해
Tableau 대시보드에서 제공

사용 기술: Tableau

05 대시보드 및 홈페이지 예시



Kakao 대시보드

목표

분석된 데이터를 시각화해
Tableau 대시보드에서 제공

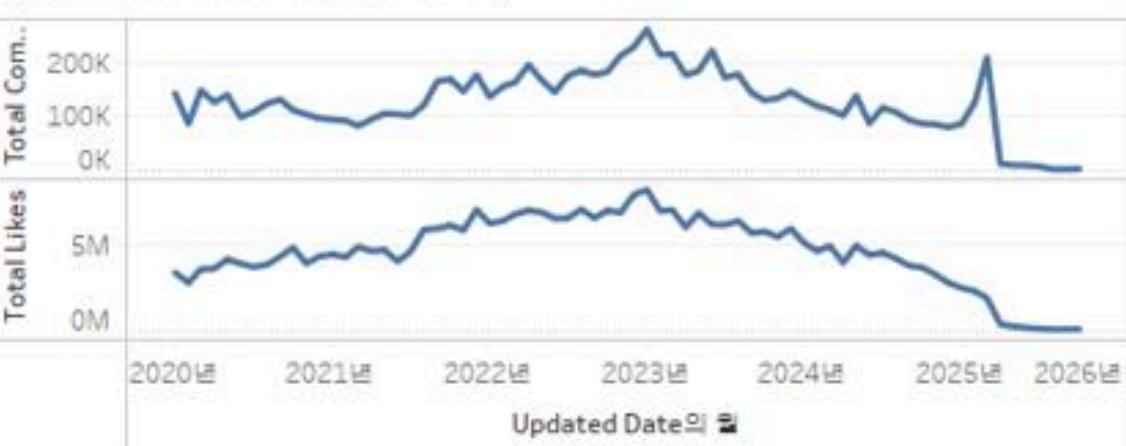
이 대시보드는 웹툰 시장의 성장 트렌드를 분석하고, Kakao 웹툰의 최근 30일 동안 성장률이 높은 웹툰과 장르별 선호도를 시각화한 것입니다.

release_day
0 ~ 6

측정값 이동
Total Comments
Total Likes
Total Views
Comments
Likes
Popularity Score
6T 18T

Avg Views Per Episode
434,721,587 1,280,800,090
Avg Comments Per Day
13,957 372,037

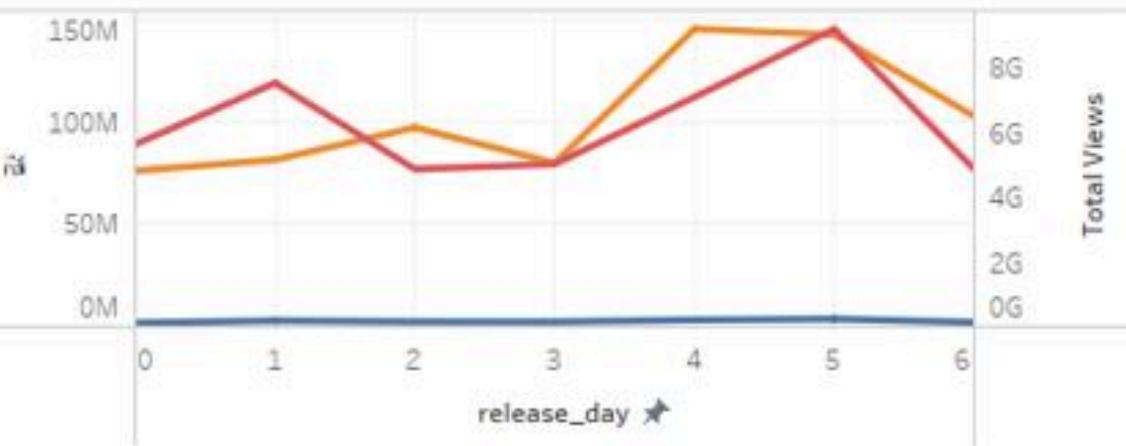
웹툰 시장 사용자 반응 추세



Total Com.: 200K
Total Likes: 5M

Updated Date: 2020년 ~ 2026년

웹툰 시장 요일별 추세

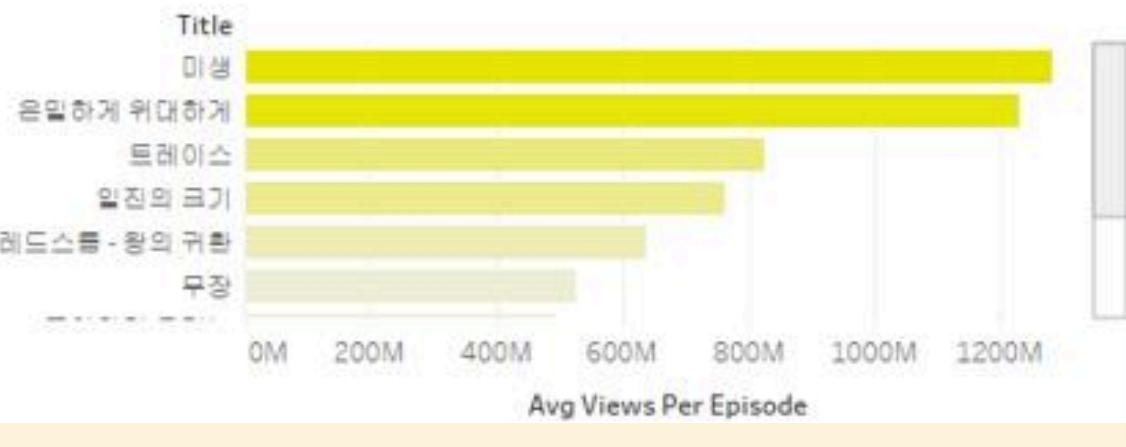


Total Views: 0M ~ 150M

release_day: 0 ~ 6

Legend: Total Comments (Blue), Total Likes (Orange), Total Views (Red)

Kakao 평균 조회수 TOP 10



Title: 미생, 은밀하게 위대하게, 트레이스, 일진의 크기, 레드스톰 - 왕의 귀환, 무장

Avg Views Per Episode: 434,721,587, 1,280,800,090

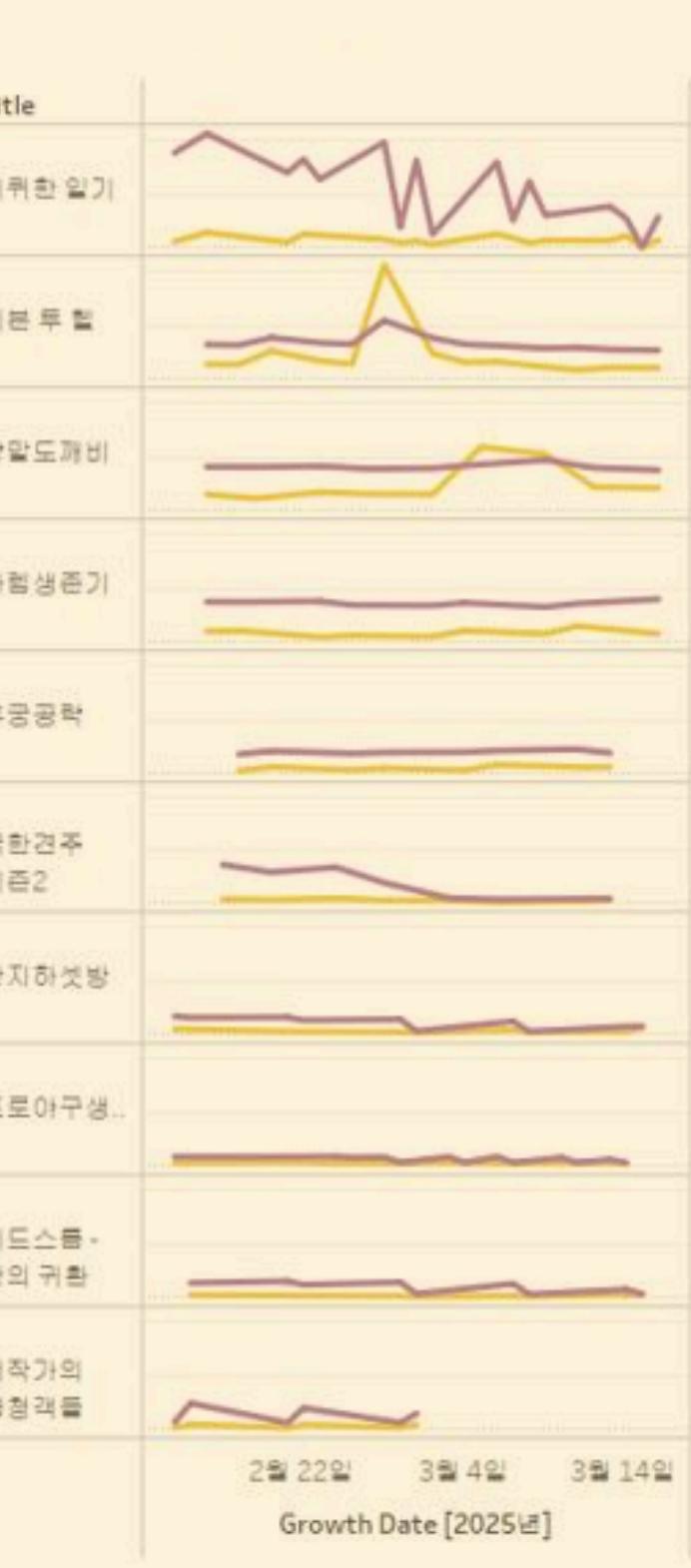
Avg Comments Per Day: 13,957, 372,037

Avg Views Per Episode: 0M ~ 1200M

성장률 TOP 10

Author	Title	Start Date	Total Com..	Total Likes
2B/2B/카카오웹툰 스...	퀴퀴한 일기	2025	6,324,626	448,453,145
김종훈/김종훈/카카오...	해븐 투 힐	2025	2,527,524	70,363,890
노경한/알현/카카오웹...	레드스톰 - 왕의 귀환	2025	2,301,600	54,213,175
만화상/만화상/카카오...	왕말도깨비	2025	1,778,616	43,619,472
류인/사설/야식먹는중...	트끼와 총표벌의 공생관계	2025	519,425	40,938,550
생일/포야/풀이수/에...	세이린: 악당과 계약 가족이...	2025	683,172	49,237,884
이동률(Redice Studio)...	힐힐	2025	237,275	38,595,800

성장률 TOP 10 추세 모니터링



Title: 퀴퀴한 일기, 해븐 투 힐, 양말도깨비, 하늘생존기, 투공공학, 국한전주 시즌2, 반지하햇방, 프로아구생, 레드스톰 - 왕의 귀환, 백작가의 형제들

Avg Likes Per Day: 0M ~ 25M

Growth Date: 2월 22일 ~ 3월 14일

사용 기술: Tableau

05 대시보드 및 홈페이지 예시



Kakao 대시보드

목표

분석된 데이터를 시각화해
Tableau 대시보드에서 제공

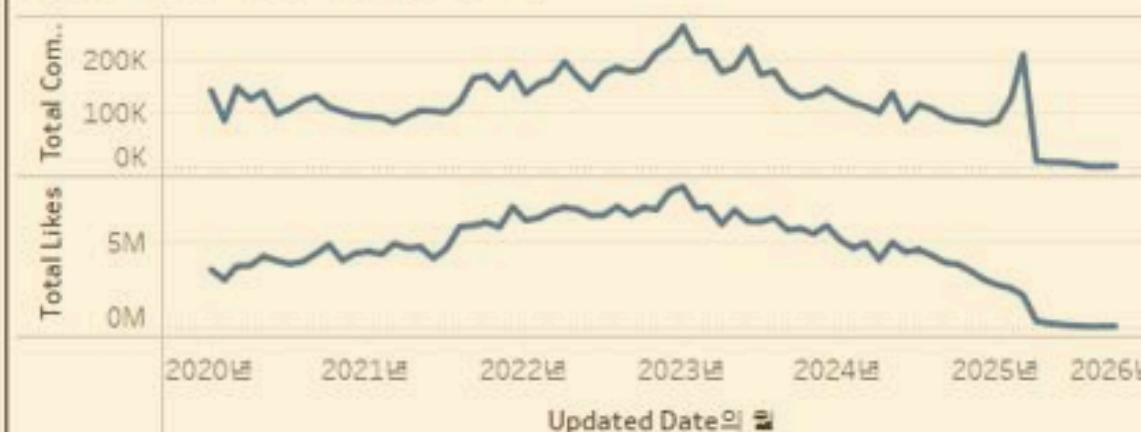
이 대시보드는 웹툰 시장의 성장 트렌드를 분석하고, Kakao 웹툰의 최근 30일 동안 성장률이 높은 웹툰과 장르별 선호도를 시각화한 것입니다.

release_day
0 ~ 6

측정값 이동
Total Comments
Total Likes
Total Views
측정값 이동
Comments
Likes
Popularity Score
6T 18T

Avg Views Per Episode
434,721,587 1,280,800,090
Avg Comments Per Day
13,957 372,037

웹툰 시장 사용자 반응 추세



Total Com.: 200K
100K
0K

Total Likes: 5M
0M

Updated Date [의 월]

웹툰 시장 요일별 추세

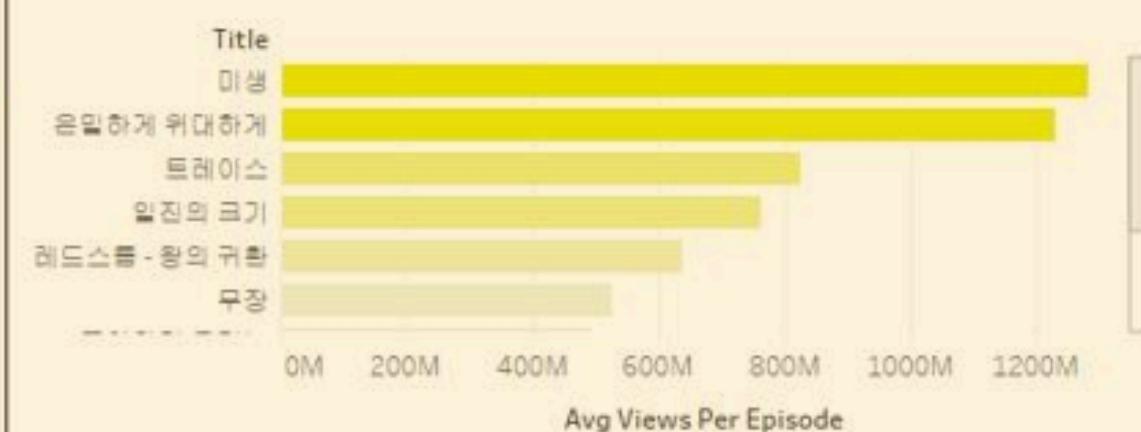


Total Views
150M
100M
50M
0M

release_day ★
0 1 2 3 4 5 6

Legend: Total Comments (Blue), Total Likes (Orange), Total Views (Red)

Kakao 평균 조회수 TOP 10



Avg Views Per Episode
434,721,587 1,280,800,090

Avg Comments Per Day
13,957 372,037

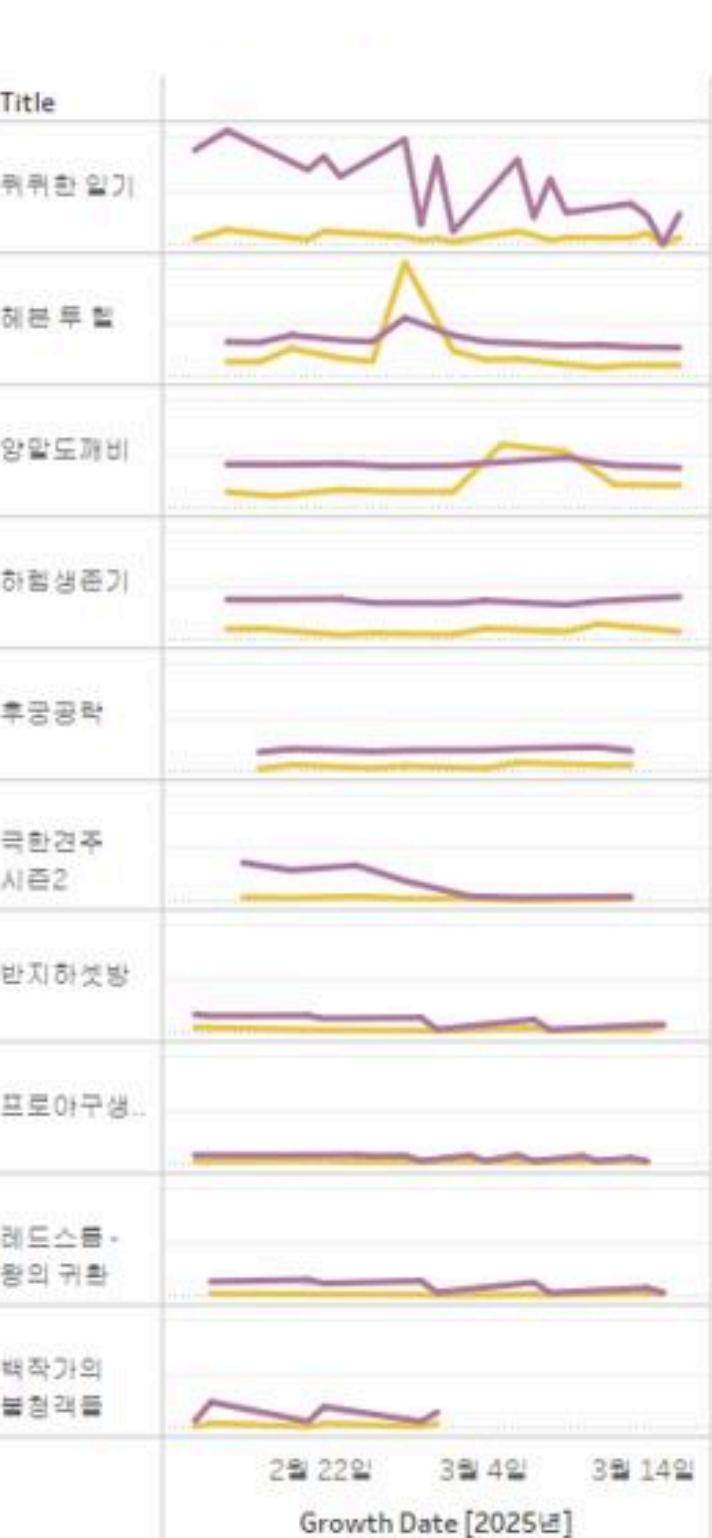
Title
미생
흔밀하게 위대하게
트레이스
일진의 크기
레드스톰 - 왕의 귀환
무장

Avg Views Per Episode
0M 200M 400M 600M 800M 1000M 1200M

성장률 TOP 10

Author	Title	Start Date	Total Com..	Total Likes
2B/2B/카카오웹툰 스...	퀴퀴한 일기	2025	6,324,626	448,453,145
김종훈/김종훈/카카오...	해븐 투 힐	2025	2,527,524	70,363,890
노경한/알현/카카오웹...	레드스톰 - 왕의 귀환	2025	2,301,600	54,213,175
만화상/만화상/카카오...	왕말도깨비	2025	1,778,616	43,619,472
류인/사설/야식먹는중...	트끼와 총표범의 공생관계	2025	519,425	40,938,550
성알/포야/풀이수/에...	세이린: 악당과 계약 가족이...	2025	683,172	49,237,884
이동률(Redice Studio)...	힐랄	2025	237,275	38,595,800

성장률 TOP 10 추세 모니터링



Title
퀴퀴한 일기
해븐 투 힐
반지하셋방
레드스톰 - 왕의 귀환
세이린: 악당과 계약 가족이...
왕말도깨비
트끼와 총표범의 공생관계
힐랄
한국판 미생
한국판 미생 시즌2
반지하셋방
프로야구생...
레드스톰 - 왕의 귀환
책작가의 형제들

Avg Likes Per Day
0M 5M 10M 15M 20M 25M

Growth Date [2025년]
2월 22일 3월 4일 3월 14일

사용 기술: Tableau

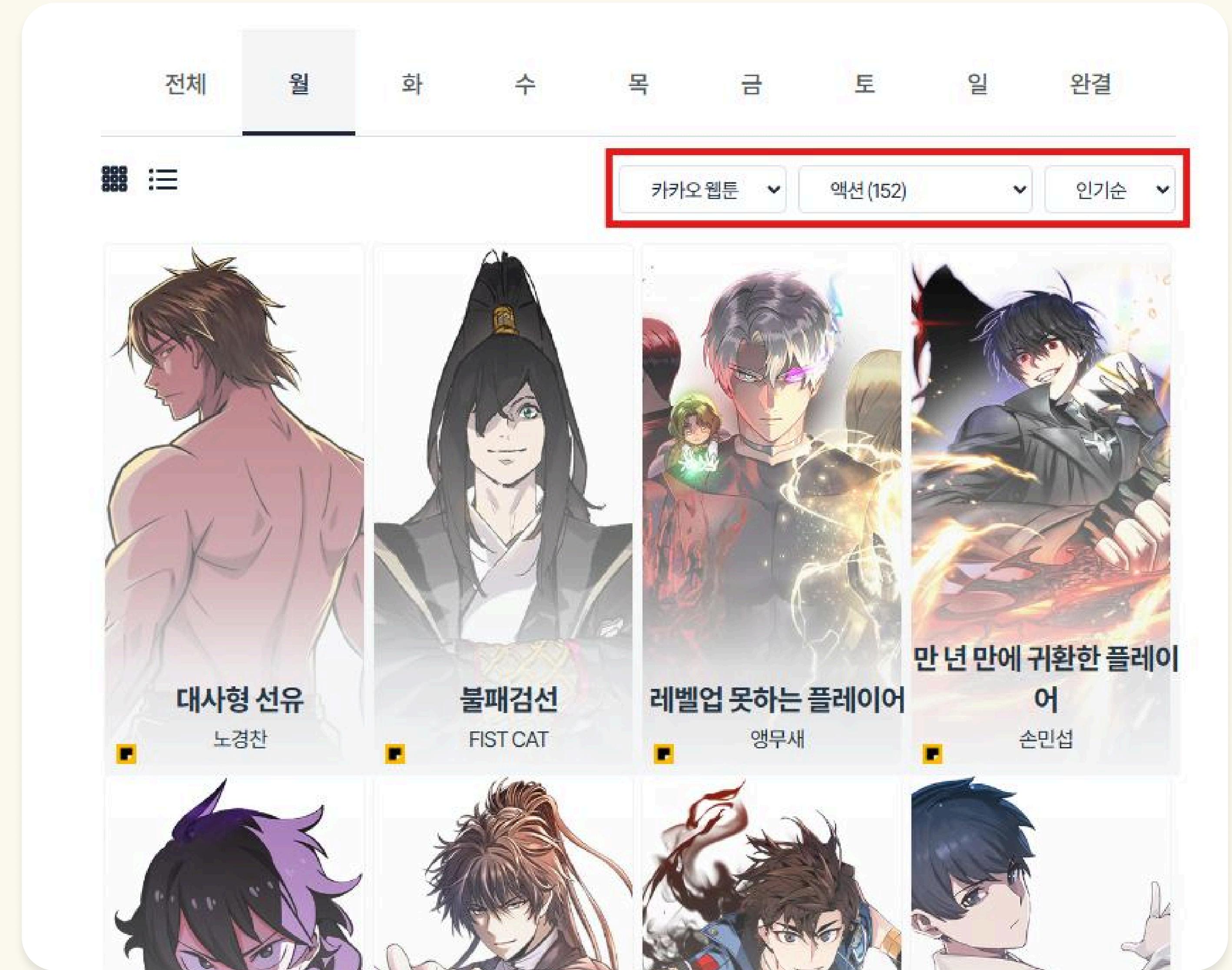
05 대시보드 및 홈페이지 예시

목표

웹툰 데이터를 Flask 웹 애플리케이션에서 사용자에게 제공

웹툰 목록(그리드)

- 요일별 웹툰 목록을 그리드 형태로 나열
- 필터 옵션을 통해 플랫폼, 장르, 정렬 방식별 웹툰 기준 선택 가능



05 대시보드 및 홈페이지 예시

목표

웹툰 데이터를 Flask 웹 애플리케이션에서 사용자에게 제공

웹툰 목록(리스트)

- 요일별 웹툰 목록을 리스트 형태로 나열
- 필터 옵션을 통해 플랫폼, 장르, 정렬 방식별

웹툰 기준 선택 가능

제목	작가	장르	상태
참교육	채용택 / 한가람	한가람	TOP
더블클릭	김장훈, 박수봉 / 박수봉	액션	TODAY'S TOP
잔불의 기사	환당	판타지	TOP
퀘스트지상주의	박만사, 유누니 / 박만사, 태완	판타지	TOP
제왕	김남규 / 애풋, 아쿠아콘	액션	TOP
절대검감	김두루미 / 티아이 / 한중월야	판타지	TOP
어쌔신 크리드 - 잊혀진 사원	VEON, ARD, IlhiaSoft / Tahii	액션	TOP

05 대시보드 및 홈페이지 예시

목표

웹툰 데이터를 Flask 웹 애플리케이션에서 사용자에게 제공

에피소드 목록

- 에피소드 목록을 리스트 형태로 나열
- 해당 웹툰의 기본 정보와 인기도를 확인 가능
- 각 회차별 업로드 날짜 및 좋아요 수, 댓글 수 표시

월요웹툰



불패검선

FIST CAT / 김찬영 / 적하 / 슈퍼코믹스스튜디오

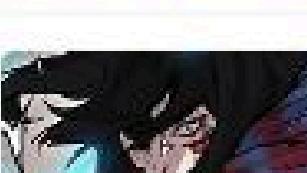
3370.8만

84.1만

5,080

#성공성장물 #액션 #무협 #통쾌한 #몰입되는

업데이트순

 <p>248화(본편완결)</p> <p>2025-01-12</p> <p>좋아요수 535 댓글수 18</p>
 <p>247화</p> <p>2025-01-05</p> <p>좋아요수 520 댓글수 7</p>
 <p>246화</p> <p>2024-12-29</p> <p>좋아요수 436 댓글수 7</p>
 <p>245화</p> <p>2024-12-22</p> <p>좋아요수 421 댓글수 11</p>

06 개선 사항

01 데이터 처리 방식 변경

기존 구조

기존에는 **Pandas**를 사용해 데이터를 처리했지만,
20GB 이상 크기의 데이터는 메모리 부족으로 인해
한 번에 처리할 수 없었음

기존 흐름

S3 → Pandas 처리 → 결과

06 개선 사항

01 데이터 처리 방식 변경

기존 구조

기존에는 **Pandas**를 사용해 데이터를 처리했지만,
20GB 이상 크기의 데이터는 메모리 부족으로 인해
한 번에 처리할 수 없었음

기존 흐름

S3 → Pandas 처리 → 결과

개선된 구조

Spark를 도입해 로컬 환경에서 데이터를
처리할 수 있도록 변경

Spark는 대용량 데이터를 처리하는 데에
효율적이며 Pandas에서 발생한
메모리 부족 문제 해결할 수 있음

핵심 흐름

S3 → Spark 처리 → 결과

06 개선 사항

01 데이터 처리 방식 변경

기존 구조

기존에는 **Pandas**를 사용해 데이터를 처리했지만,
20GB 이상 크기의 데이터는 메모리 부족으로 인해
한 번에 처리할 수 없었음

기존 흐름

S3 → Pandas 처리 → 결과

개선된 구조

Spark를 도입해 로컬 환경에서 데이터를
처리할 수 있도록 변경

Spark는 대용량 데이터를 처리하는 데에
효율적이며 Pandad에서 발생한
메모리 부족 문제 해결할 수 있음

핵심 흐름

S3 → Spark 처리 → 결과

주요 개선 사항

메모리 문제 해결

Spark를 사용해 13GB 크기의 데이터 처리가
용이해짐으로써 기존엔 불가능했던 데이터 처리가 가능

처리 속도 향상

로컬 환경에서 Spark를 사용해
Pandas보다 빠른 데이터 처리 속도를 달성

확장성

향후 클러스터 환경에서 병렬 처리를 통해
성능을 더욱 향상시킬 수 있음

06 개선 사항

02 데이터 처리 구조 개선

기존 구조

기존 구조는 원시 데이터를 S3에 업로드 한 후
Spark로 데이터를 정제하고 다시 S3에 업로드 하는 방식

데이터 처리 중 오류 발생 시
실시간 대응이 어려웠고 기본적인 스크립트 실행 시간이
10분 이상 걸려 비효율적

기존 흐름

raw → processed

06 개선 사항

02 데이터 처리 구조 개선

기존 구조

기존 구조는 원시 데이터를 S3에 업로드 한 후
Spark로 데이터를 정제하고 다시 S3에 업로드 하는 방식

데이터 처리 중 오류 발생 시
실시간 대응이 어려웠고 기본적인 스크립트 실행 시간이
10분 이상 걸려 비효율적

기존 흐름

raw → processed

개선된 구조

개선된 구조는 원시 데이터를 S3에 업로드 한 후
최적화 단계를 추가해 데이터를 최적화를 거치고
다시 S3에 업로드 하고 정제 작업을 진행하는 방식

데이터 처리 효율성을 높이고
중간 오류 발생 시 빠른 대응 가능

핵심 흐름

raw → optimized → processed

06 개선 사항

02 데이터 처리 구조 개선

기존 구조

기존 구조는 원시 데이터를 S3에 업로드 한 후
Spark로 데이터를 정제하고 다시 S3에 업로드 하는 방식

데이터 처리 중 오류 발생 시
실시간 대응이 어려웠고 기본적인 스크립트 실행 시간이
10분 이상 걸려 비효율적

기존 흐름

raw → processed

개선된 구조

개선된 구조는 원시 데이터를 S3에 업로드 한 후
최적화 단계를 추가해 데이터를 최적화를 거치고
다시 S3에 업로드 하고 정제 작업을 진행하는 방식

데이터 처리 효율성을 높이고
중간 오류 발생 시 빠른 대응 가능

핵심 흐름

raw → optimized → processed

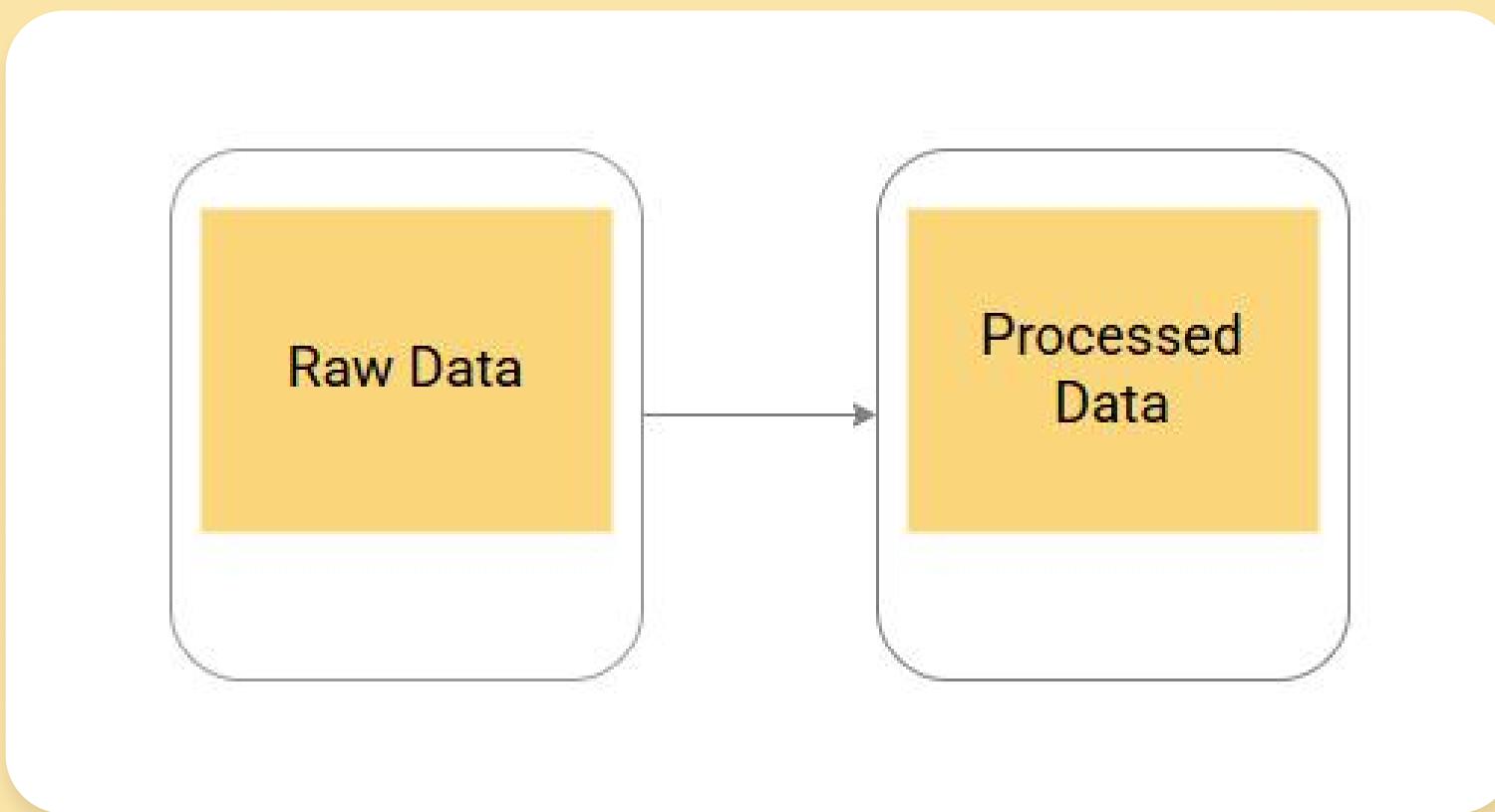
주요 개선 사항

중간 오류 처리 용이
최적화 단계 추가로 중간 오류를
쉽게 파악하고 대응 가능

효율성 향상
최적화된 데이터로 처리 시
실행 시간 단축 예상

02 데이터 처리 구조 개선

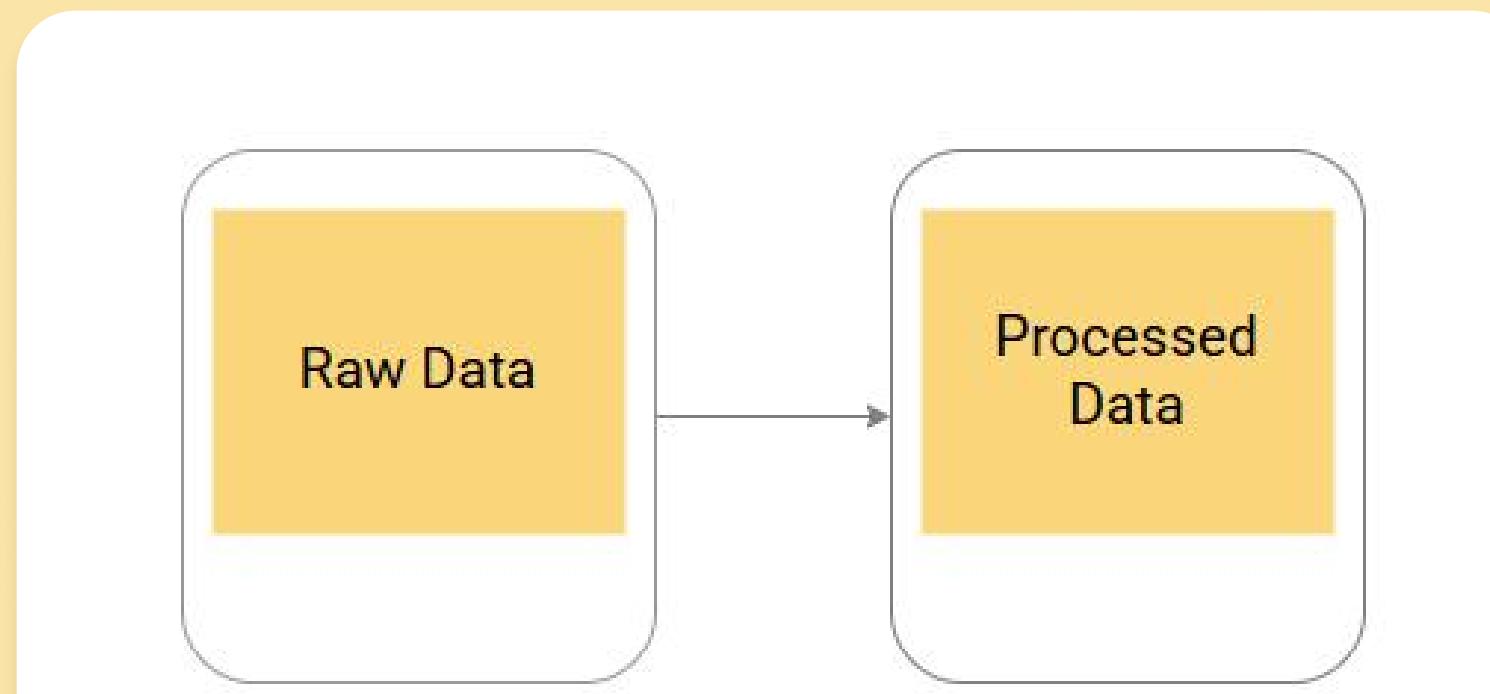
기존 구조



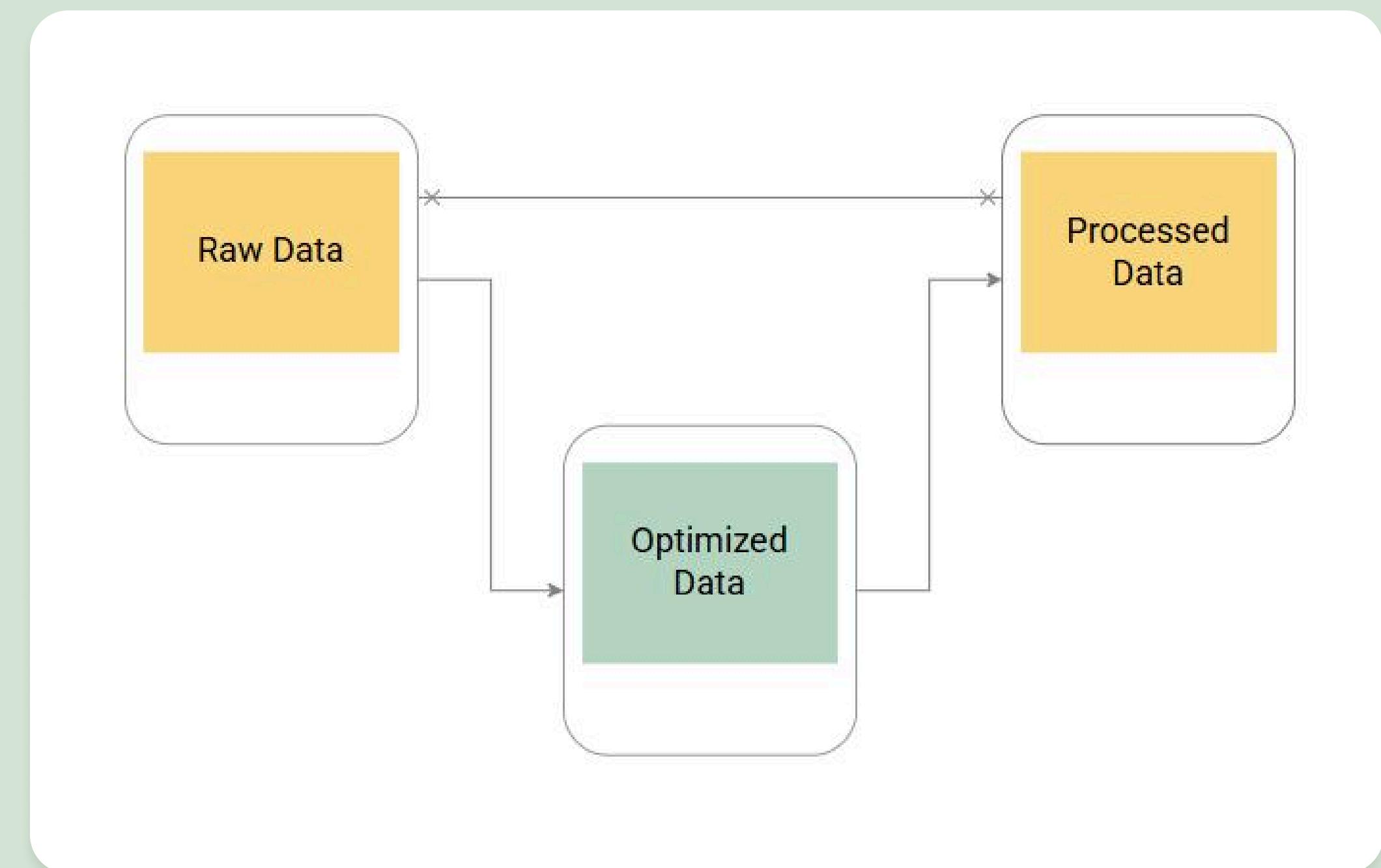
06 개선 사항

02 데이터 처리 구조 개선

기존 구조



개선된 구조



03 데이터 처리 병렬화 및 최적화

기존 구조

기존 구조에서는 크롤링 및 파싱 작업이 순차적으로
진행되어, 전체 데이터를 받아올 경우에
처리 시간이 매우 오래 걸림

기존 처리 시간

네이버 전체: 12시간 이상/데일리: 25초 이상

카카오 전체: 5시간 이상/데일리: 20초 이상

06 개선 사항

03 데이터 처리 병렬화 및 최적화

기존 구조

기존 구조에서는 크롤링 및 파싱 작업이 순차적으로
진행되어, 전체 데이터를 받아올 경우에
처리 시간이 매우 오래 걸림

기존 처리 시간

네이버 전체: 12시간 이상/데일리: 25초 이상
카카오 전체: 5시간 이상/데일리: 20초 이상

개선된 구조

개선된 구조는 파싱 작업을 병렬 처리함으로써
전체 크롤링 시간을 크게 단축하고,
`concurrent.futures` 라이브러리를 활용해
IO-bound 작업을 병렬로 실행함으로써 성능 최적화

개선 처리 시간

네이버(약 60% 감소) 전체: 5시간/데일리: 10초
카카오(약 60% 감소) 전체: 2시간/데일리: 8초

핵심 흐름

sequential → parallel

06 개선 사항

03 데이터 처리 병렬화 및 최적화

기존 구조

기존 구조에서는 크롤링 및 파싱 작업이 순차적으로
진행되어, 전체 데이터를 받아올 경우에
처리 시간이 매우 오래 걸림

기존 처리 시간

네이버 전체: 12시간 이상/데일리: 25초 이상
카카오 전체: 5시간 이상/데일리: 20초 이상

개선된 구조

개선된 구조는 파싱 작업을 병렬 처리함으로써
전체 크롤링 시간을 크게 단축하고,
`concurrent.futures` 라이브러리를 활용해
IO-bound 작업을 병렬로 실행함으로써 성능 최적화

개선 처리 시간

네이버(약 60% 감소) 전체: 5시간/데일리: 10초
카카오(약 60% 감소) 전체: 2시간/데일리: 8초

핵심 흐름

sequential → parallel

주요 개선 사항

처리 시간 단축

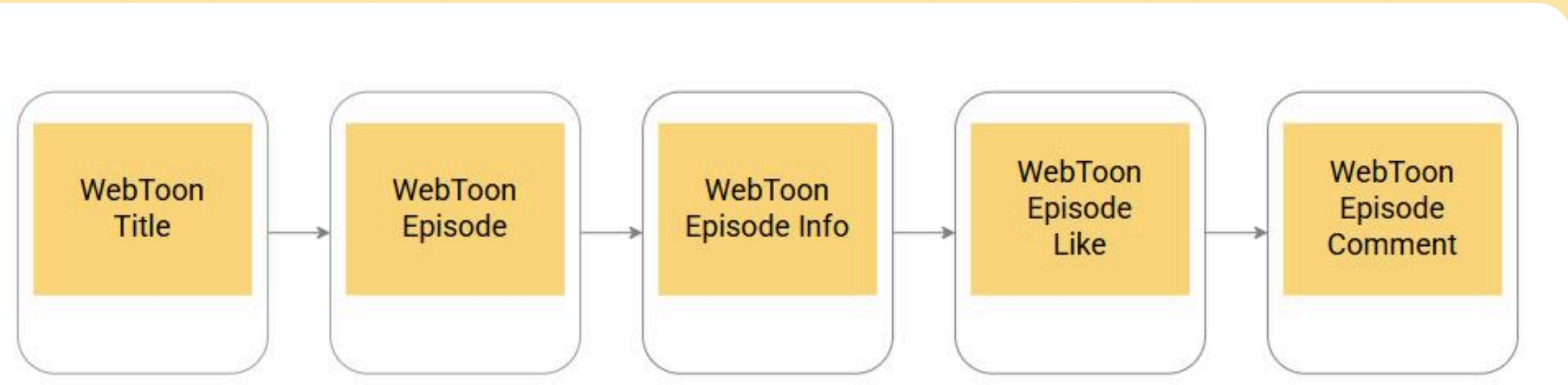
병렬 처리로 크롤링 시간이
최대 60% 단축

효율적인 병렬 처리 관리

`concurrent.futures` 라이브러리를 활용해
여러 네트워크 요청 및 IO 작업의 효율적인 처리

03 데이터 처리 병렬화 및 최적화

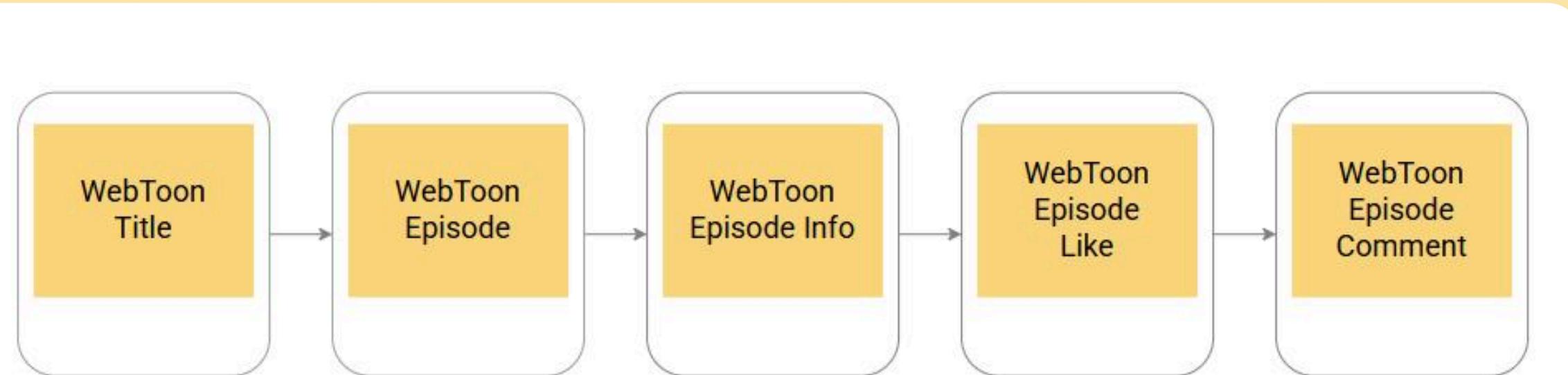
기준 구조



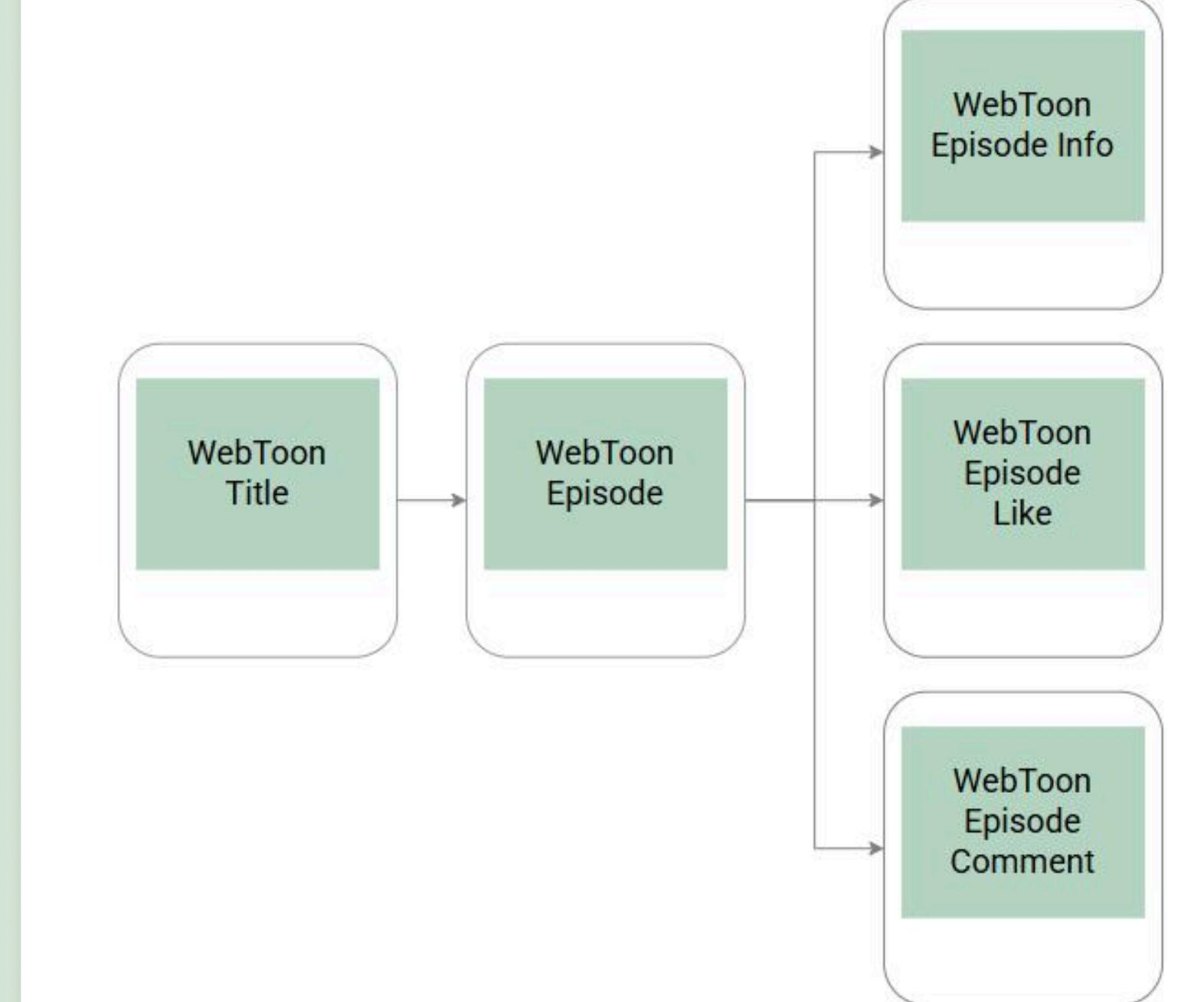
06 개선 사항

03 데이터 처리 병렬화 및 최적화

기존 구조



개선된 구조



사용 기술: concurrent.futures

06 개선 사항

+ concurrent.futures 사용 이유

IO-bound 작업 최적화

‘concurrent.futures’ 라이브러리는
네트워크 요청과 같은 IO-bound 작업에 최적화

대기 시간이 긴 작업을 병렬로 처리함으로써
성능을 향상시킬 수 있음

06 개선 사항

+ concurrent.futures 사용 이유

IO-bound 작업 최적화

`concurrent.futures` 라이브러리는
네트워크 요청과 같은 IO-bound 작업에 최적화

대기 시간이 긴 작업을 병렬로 처리함으로써
성능을 향상시킬 수 있음

간단한 병렬 처리 관리

`ThreadPoolExecutor`를 사용해
병렬 처리를 간단하게 관리하고
유지 보수가 용이함

06 개선 사항

+ concurrent.futures 사용 이유

IO-bound 작업 최적화

‘concurrent.futures’ 라이브러리는 네트워크 요청과 같은 IO-bound 작업에 최적화

대기 시간이 긴 작업을 병렬로 처리함으로써 성능을 향상시킬 수 있음

간단한 병렬 처리 관리

‘ThreadPoolExecutor’를 사용해 병렬 처리를 간단하게 관리하고 유지 보수가 용이함

성능 향상 및 과부하 방지

기본적인 멀티쓰레딩보다 효율적으로 병렬 작업을 처리할 수 있어 전체적인 크롤링 속도 및 성능 개선 가능

비동기 방식인 ‘asyncio’는 많은 네트워크 요청에서 과부하가 발생할 수 있지만, ‘concurrent.futures’는 적절한 쓰레드 수를 설정해 과부하를 방지할 수 있음

06 개선 사항

04 코드 배포 자동화

AWS Systems Manager

Github에 푸시된 코드를 EC2 인스턴스에 자동으로 반영하기 위해 **AWS Systems Manager(SSM)**을 활용한 자동화 작업 설정

SSM을 통해 EC2의 퍼블릭 IP 없이 인스턴스 ID를 통해 직접 연결하고, SSH 접속 없이 코드 변경 사항을 자동으로 반영 가능

개선점

EC2 관리 효율성 향상

06 개선 사항

04 코드 배포 자동화

AWS Systems Manager

Github에 푸시된 코드를 EC2 인스턴스에 자동으로 반영하기 위해 **AWS Systems Manager(SSM)**을 활용한 자동화 작업 설정

SSM을 통해 EC2의 퍼블릭 IP 없이 인스턴스 ID를 통해 직접 연결하고, SSH 접속 없이 코드 변경 사항을 자동으로 반영 가능

개선점

EC2 관리 효율성 향상

자동화 과정

1. SSM 에이전트 설치

EC2 인스턴스에 SSM 에이전트를 설치해 인스턴스를 관리할 수 있도록 설정

2. IAM 역할 설정

EC2 인스턴스에 적절한 IAM 역할을 할당해 SSM을 통해 접근할 수 있도록 설정

3. Github Actions 설정

Github Actions 워크플로우를 설정해 코드가 푸시될 때마다 자동으로 EC2 인스턴스에 반영

4. 워크플로우 구성

푸시 이벤트가 발생하면 Github Actions에서 AWS CLI를 사용해 EC2에 접근하고 최신 코드 자동 배포

5. 배포 스크립트 실행

EC2 인스턴스에 연결된 후, 배포 스크립트를 실행해 최신 코드를 반영하고 필요한 서비스 재시작

06 개선 사항

04 코드 배포 자동화

AWS Systems Manager

Github에 푸시된 코드를 EC2 인스턴스에 자동으로 반영하기 위해 **AWS Systems Manager(SSM)**을 활용한 자동화 작업 설정

SSM을 통해 EC2의 퍼블릭 IP 없이 인스턴스 ID를 통해 직접 연결하고, SSH 접속 없이 코드 변경 사항을 자동으로 반영 가능

개선점

EC2 관리 효율성 향상

자동화 과정

1. SSM 에이전트 설치

EC2 인스턴스에 SSM 에이전트를 설치해 인스턴스를 관리할 수 있도록 설정

2. IAM 역할 설정

EC2 인스턴스에 적절한 IAM 역할을 할당해 SSM을 통해 접근할 수 있도록 설정

3. Github Actions 설정

Github Actions 워크플로우를 설정해 코드가 푸시될 때마다 자동으로 EC2 인스턴스에 반영

4. 워크플로우 구성

푸시 이벤트가 발생하면 Github Actions에서 AWS CLI를 사용해 EC2에 접근하고 최신 코드 자동 배포

5. 배포 스크립트 실행

EC2 인스턴스에 연결된 후, 배포 스크립트를 실행해 최신 코드를 반영하고 필요한 서비스 재시작

주요 개선 사항

SSH 접속 불필요

EC2에 직접 접속할 필요 없이 SSM을 통해 코드 배포 가능

자동화된 배포

코드 푸시 후 EC2 인스턴스에서 자동으로 배포가 이루어져 관리 용이

보안 강화

퍼블릭 IP 없이도 EC2 인스턴스를 관리할 수 있어 보안상 더 안전한 접근 방식 제공

06 개선 사항

05 DAG 알림 설정

Slack 알림 시스템

Airflow에서 작업 상태를 실시간으로
알림 받을 수 있도록 Slack 알림 시스템 도입

성공 및 실패 알림을 통해
DAG 실행 사항을 빠르게 모니터링하고
필요한 조치를 즉시 취할 수 있음

06 개선 사항

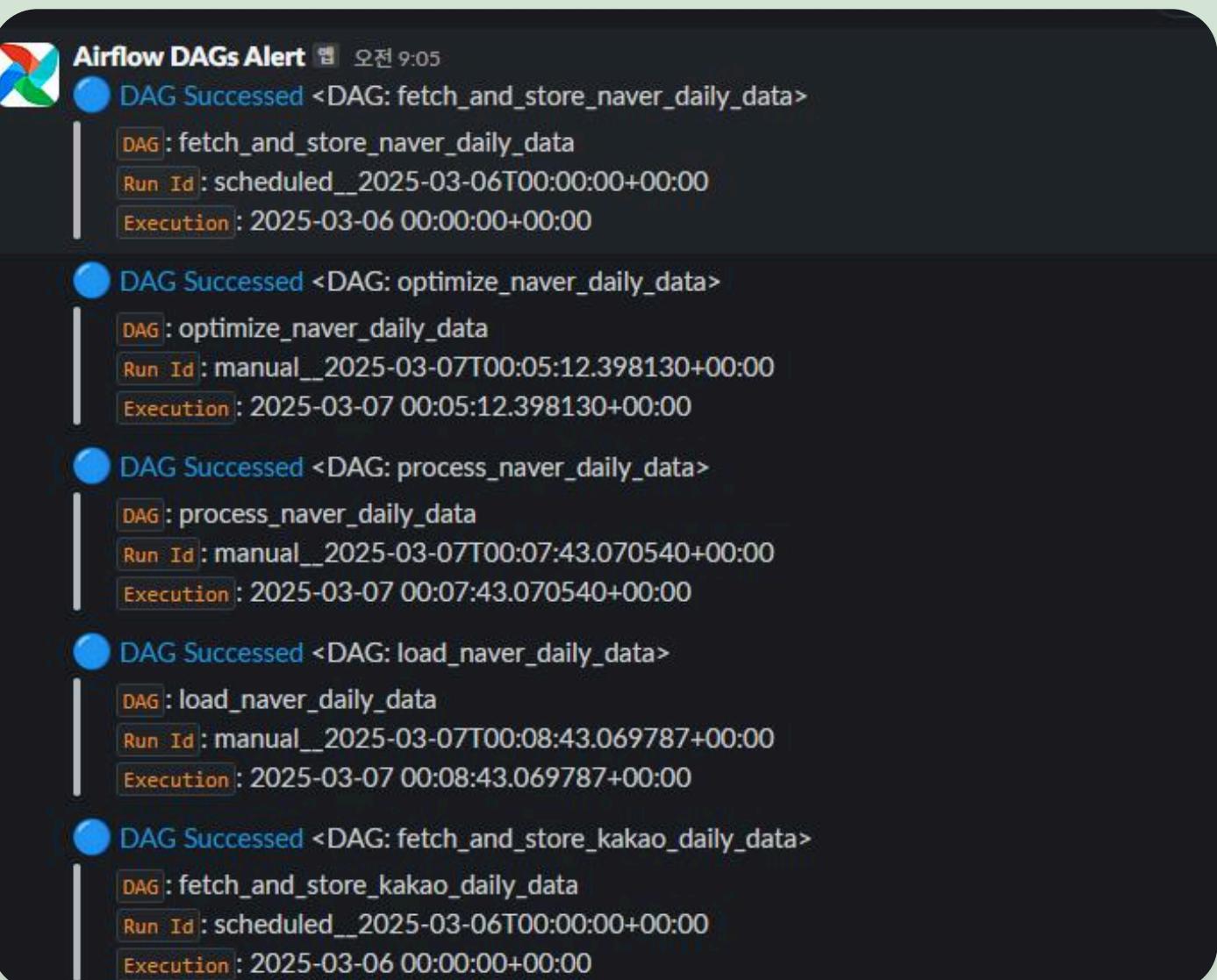
05 DAG 알림 설정

Slack 알림 시스템

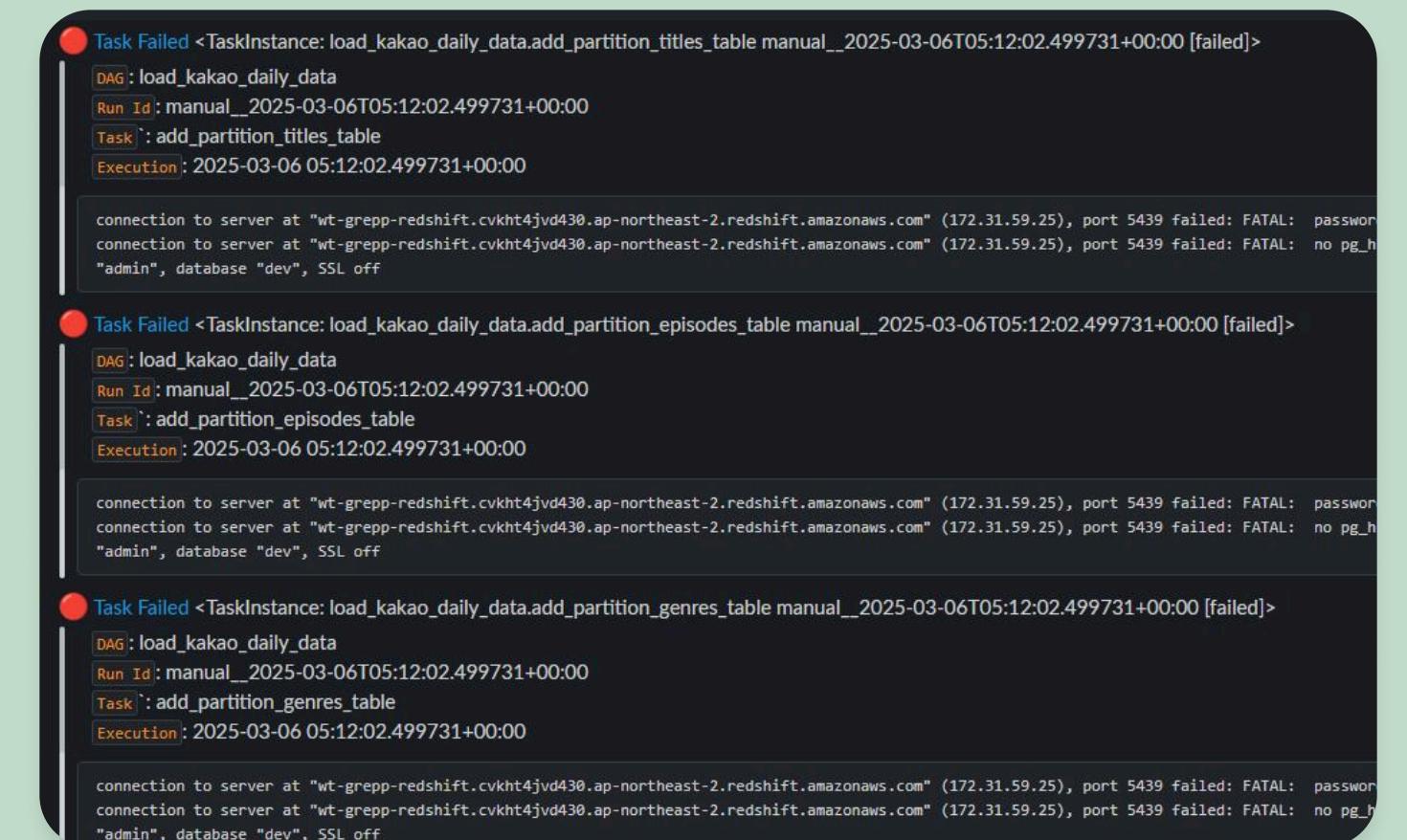
Airflow에서 작업 상태를 실시간으로
알림 받을 수 있도록 Slack 알림 시스템 도입

성공 및 실패 알림을 통해
DAG 실행 사항을 빠르게 모니터링하고
필요한 조치를 즉시 취할 수 있음

성공 알림 예시



실패 알림 예시



06 개선 사항

06 DAG 관리 및 Airflow 튜닝

리소스 제한을 고려한 DAG 관리

기존 구조

Spark 작업이 동시에 실행될 경우

리소스가 과부하되어 서버가 다운되는 문제 발생

개선된 구조

작업의 실행 순서를 제어하는 관리용 DAG를 추가해

태스크 간 실행 타이밍을 조절하고 리소스 활용 최적화

주요 개선 사항

동시에 과도한 Spark 작업이 실행되지 않도록

조율하면서 전체적인 워크플로우의 원활한 진행 보장

06 개선 사항

06 DAG 관리 및 Airflow 튜닝

리소스 제한을 고려한 DAG 관리

기존 구조

Spark 작업이 동시에 실행될 경우

리소스가 과부하되어 서버가 다운되는 문제 발생

개선된 구조

작업의 실행 순서를 제어하는 관리용 DAG를 추가해
태스크 간 실행 타이밍을 조절하고 리소스 활용 최적화

주요 개선 사항

동시에 과도한 Spark 작업이 실행되지 않도록
조율하면서 전체적인 워크플로우의 원활한 진행 보장

Airflow Operator 튜닝

기존 구조

기존 S3 업로드 오퍼레이터는 단일 파일 단위로만
동작해 여러 개의 파일을 한 번에 업로드하는 경우
반복적인 호출이 이루어짐

개선된 구조

폴더 단위로 여러 파일을 한 번에 업로드할 수 있는
새로운 오퍼레이터를 구현해
성능을 최적화하고 관리의 편의성 제고

06 개선 사항

06 DAG 관리 및 Airflow 튜닝

리소스 제한을 고려한 DAG 관리

기존 구조

Spark 작업이 동시에 실행될 경우
리소스가 과부하되어 서버가 다운되는 문제 발생

개선된 구조

작업의 실행 순서를 제어하는 관리용 DAG를 추가해
태스크 간 실행 타이밍을 조절하고 리소스 활용 최적화

주요 개선 사항

동시에 과도한 Spark 작업이 실행되지 않도록
조율하면서 전체적인 워크플로우의 원활한 진행 보장

Airflow Operator 튜닝

기존 구조

기존 S3 업로드 오퍼레이터는 단일 파일 단위로만
동작해 여러 개의 파일을 한 번에 업로드하는 경우
반복적인 호출이 이루어짐

개선된 구조

폴더 단위로 여러 파일을 한 번에 업로드할 수 있는
새로운 오퍼레이터를 구현해
성능을 최적화하고 관리의 편의성 제고

Airflow Hook 튜닝

기존 구조

기본 제공되는 Redshift 및 PostgreSQL 오퍼레이터는
특정 기능에 제한이 있어 다양한 데이터 처리 시
유연하게 대응하기 어려운 문제 발생

개선된 구조

별도의 커스텀 툴을 구현해
기존 툴의 한계를 보완하고 필요에 맞게 기능 확장

07 기대 효과

데이터 기반 의사 결정



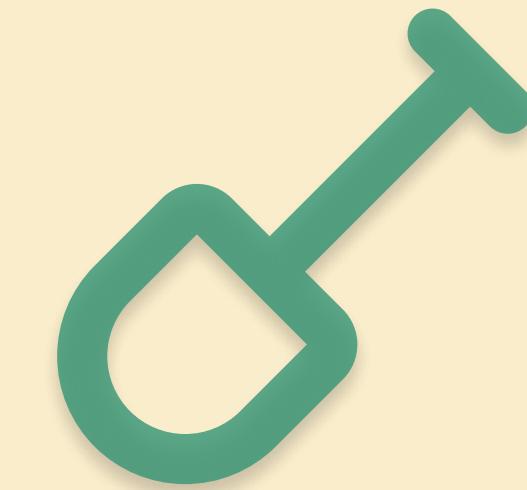
조회수, 댓글 수, 좋아요 수 등의 데이터를 분석해 인기 웹툰 트렌드를 파악하고, 플랫폼 운영자가 보다 정교한 추천 시스템을 구축할 수 있도록 지원

사용자 소비 패턴 분석



요일별, 장르별 선호도 분석을 통해 사용자들이 어떤 웹툰을 주로 소비하는지 파악하고, 타겟 마케팅이나 콘텐츠 기획에 활용할 수 있음

비즈니스 기회 발굴



웹툰 시장의 성장과 함께 광고, IP 비즈니스, 2차 창작물 등의 기회를 찾는데 필요한 데이터 제공

08 아쉬운 점

댓글 데이터 활용 부족

댓글 데이터는 사용자의 상호작용을 분석하는데 중요한 정보를 제공할 수 있지만, 시간적 제약과 데이터 처리 방식에 대한 한계로 활용에 충분히 집중하지 못함

한정된 플랫폼 데이터

현재는 네이버와 카카오 플랫폼에서만 데이터를 수집하고 있는데, 레진코믹스나 리디 등 다른 플랫폼의 데이터를 포함하지 못해 **플랫폼의 다양성을 충분히 반영하지 못함**

Spark 병렬 처리 한계

리소스의 제약으로 인해 **Spark의 분산 처리 장점이 제한적으로만 발휘되었지만, Spark를 활용해 데이터를 효율적으로 읽을 수 있었음** (Pandas의 경우 데이터 크기 문제로 처리 불가)

감사합니다