

신차 출시 전/후 소비자 반응 모니터링

: 데이터 특성을 고려한 파이프라인

The All New Team3
김건아, 김민재, 양주영

목차

1. 프로젝트 소개
 2. SNS 데이터의 특성
 3. 데이터 파이프라인
 4. 데이터의 비대칭성
 5. 장애 대응 전략
 6. 데이터 모델링
 7. 결론
- Appendix

#1. 디 올 뉴 쌘타페, 혁신인가? 실험인가?

2023년 8월, 싼타페 풀체인지



[더 뉴 싼타페]



[디 올 뉴 싼타페]

2023년 8월, 싼타페 풀체인지



[더 뉴 싼타페]

[디 올 뉴 싼타페]

Santafe 소셜 리스닝 대시보드

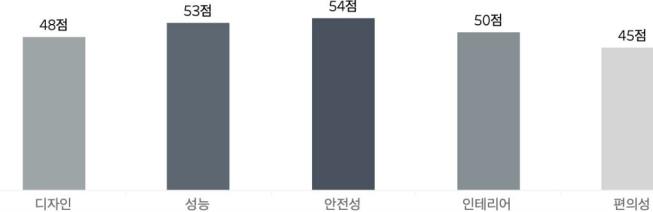
Car

Santafe

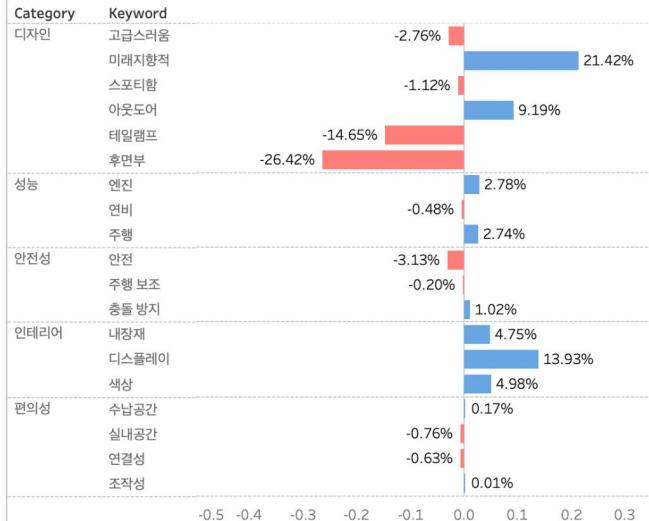
Age

2030

카테고리 긍정 점수



Category



Pos Keywords

설상
조작성 수납공간 후면부연비
충돌 방지 미래지향적
내장재 디스플레이 스포티함
아웃도어 고급스러움 주행
실내공간 안전 주행 보조
예지 헤드라이트

Neg Keywords

작성
!일램프 수납공간 스포티함
!주행 보조 안전 미래지향적
!내공간 디스플레이 내장재
후면부 고급스러움 엔진
충돌 방지 아웃도어
여겨서 헤드라이트

Santafe 소셜 리스닝 대시보드

Car

Santafe

Age

2030

카테고리 긍정 점수

48점

53점

54점

50점

45점

Pos Keywords

연설성

조작성 수납공간 후면부연비

수납부내부 디자인

후면부연비

Category

Keyword

디자인

고급스러움

-2.76%



21.42%

미래지향적

-1.12%



스포티함

아웃도어

테일램프

-14.65%



후면부

-26.42%



종료 강의

인테리어

내장재

디스플레이

색상

0.00% ~ 70

4.75%

13.93%

4.98%

편의성

수납공간

실내공간

연결성

조작성

-0.76%

-0.63%

0.01%

수납부내부 디스플레이 내장재

후면부 고급스러움 엔진

충돌 방지 아웃도어

여겨서

-0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5

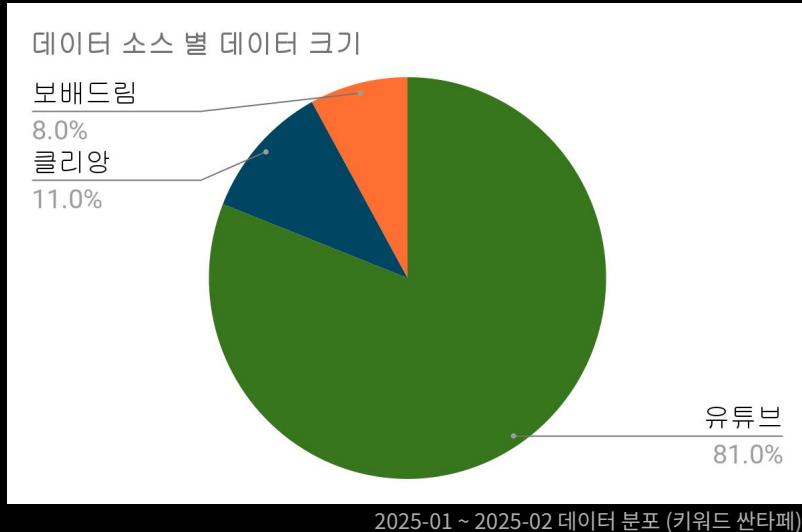
소비자 반응을 어디서 보지?

“SNS 데이터”

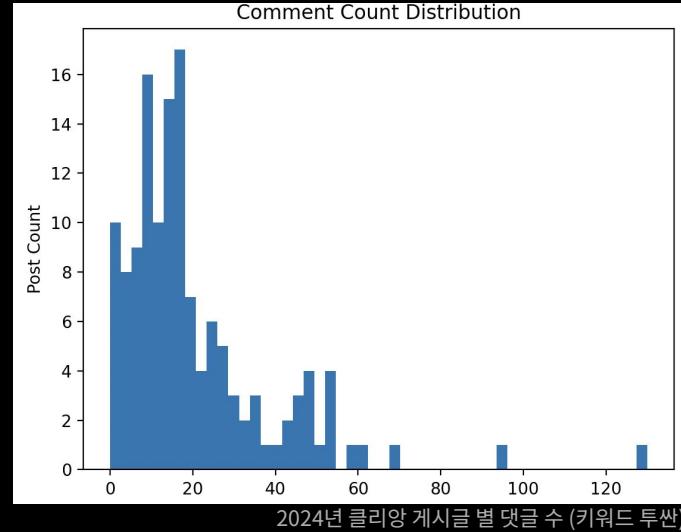
#2. SNS 데이터의 특징

1. 비대칭적인 데이터 분포

플랫폼마다 다른 데이터 분포



인기 게시글에 몰리는 관심



2. 외부 시스템의 불확실성

외부 데이터의 가용성

네트워크 에러

시스템 장애

API Limit

크롤링 차단

데이터 구조의 변동성

API 응답 변경

HTML 구조 변경

3. 시간에 따라 변하는 데이터

제목
작성자
작성일
본문 내용 } 거의 변화 없음

조회수
좋아요 수
싫어요 수
댓글 } 자주 변하는 값 → 최신 데이터에서 발생

비대칭적인 데이터 분포



적절한 부하 분산

외부 시스템의 불확실성



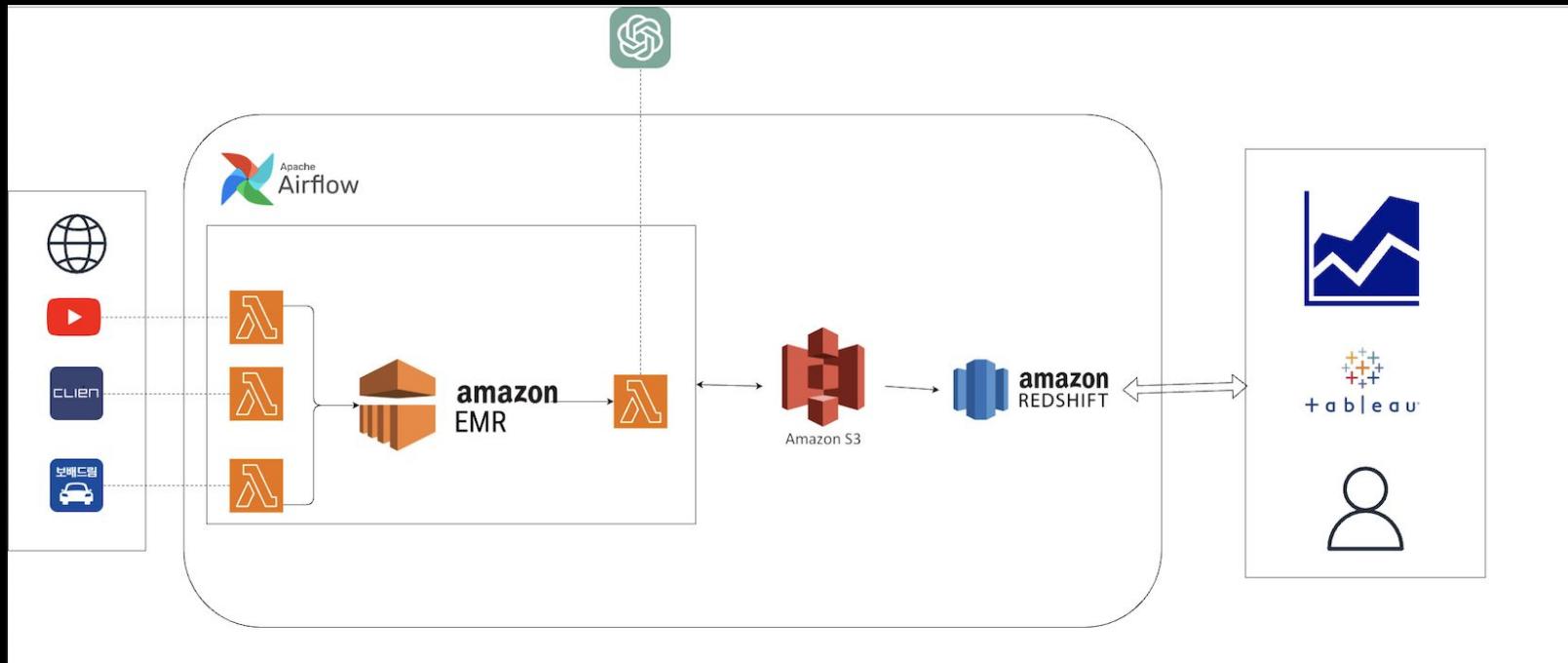
장애에 대비한 시스템

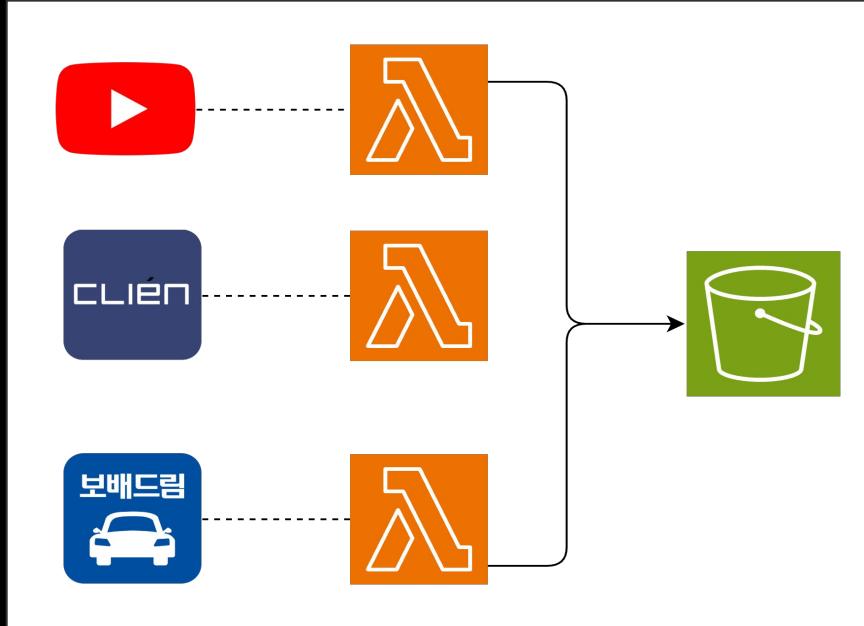
시간에 따라 변하는 데이터



하루 단위 파이프라인

#3. 데이터 파이프라인





1. 데이터 수집

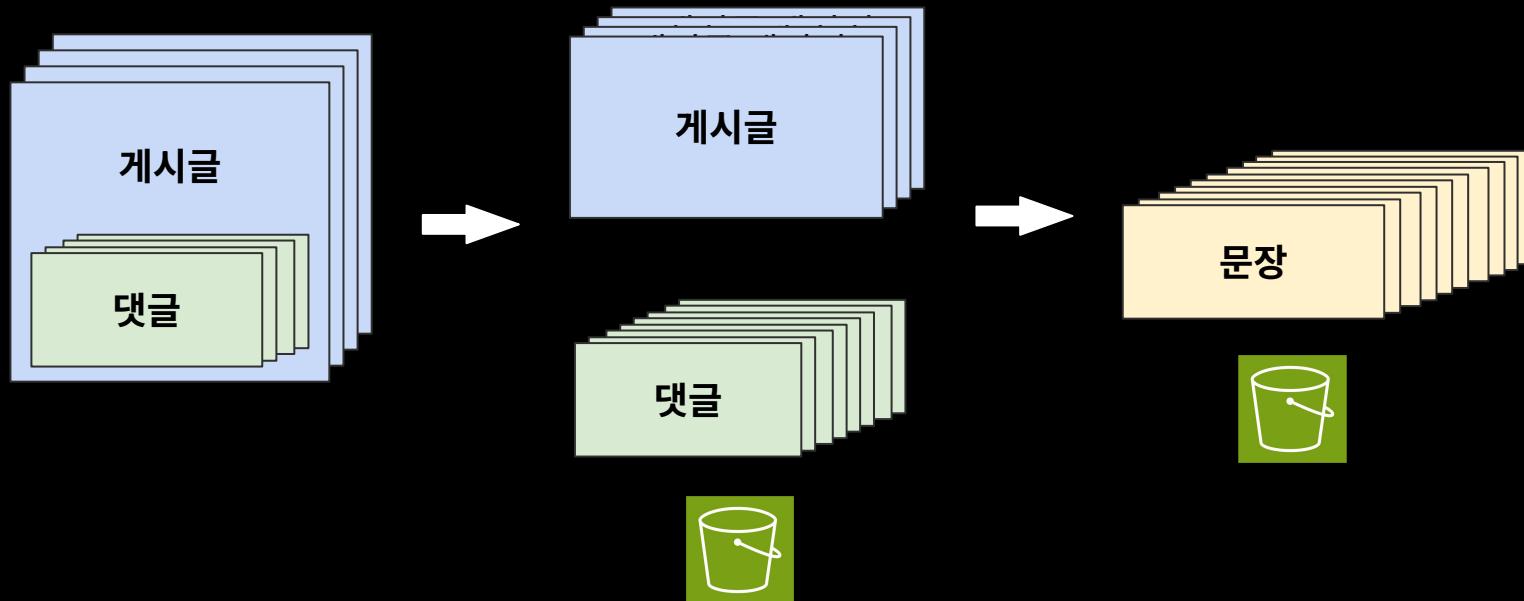
API, 크롤링 통해 데이터를 수집

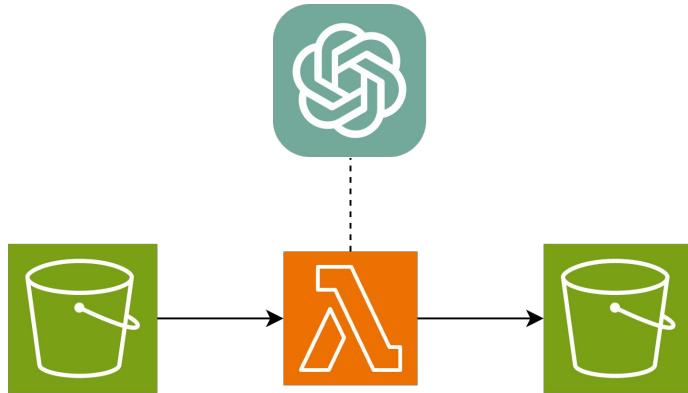


2. 게시글/댓글/문장 분리

게시글 댓글 분리

문장 추출 및 정제





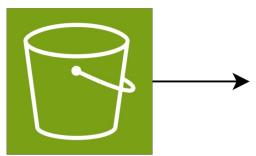
3. 문장 분석

문장의 키워드/감성 분석



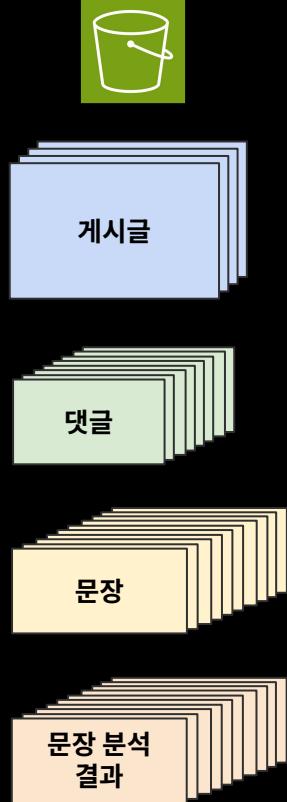
감성: 0.88
카테고리: 디자인
키워드: 측면 디자인

측면 디자인이 옛날 미국
스테이션 왜건 같은 이미지라
레트로하고 멋진거같음



amazon
REDSHIFT

4. 웨어하우스로 적재
게시글 / 댓글 / 문장 / 분석결과



게시물				staging.tb_posts			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL				
게시물 ID	post_id	VARCHAR(255)	NOT NULL				
제목	title	VARCHAR(255)	NOT NULL				
작성자	author	VARCHAR(255)	NOT NULL				
본문	article	VARCHAR(65535)	NOT NULL				
작성일	create_timestamp	BIGINT	NOT NULL				
좋아요 수	like_cnt	BIGINT	NULL				
싫어요 수	dislike_cnt	BIGINT	NULL				
조회수	view_cnt	BIGINT	NULL				
댓글 수	comment_cnt	BIGINT	NULL				
차 모델명	car_name	VARCHAR(255)	NOT NULL				
웹 데이터 소스	source	VARCHAR(255)	NOT NULL				

댓글				staging.tb_comments			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL				
댓글 UUID	comment_uuid	CHAR(36)	NOT NULL				
댓글 ID	comment_id	VARCHAR(255)	NOT NULL				
작성자	author	VARCHAR(255)	NOT NULL				
본문	content	VARCHAR(65535)	NOT NULL				
작성일	create_timestamp	BIGINT	NOT NULL				
좋아요 수	like_cnt	BIGINT	NULL				
싫어요 수	dislike_cnt	BIGINT	NULL				

문장				staging.tb_sentences			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL				
댓글 UUID	comment_uuid	CHAR(36)	NULL				
문장 UUID	sentence_uuid	CHAR(36)	NOT NULL				
문장 타입	type	VARCHAR(255)	NOT NULL				
문장	sentence	VARCHAR(65535)	NOT NULL				

키워드				staging.tb_keywords			
문장 UUID	sentence_uuid	CHAR(36)	NOT NULL				
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL				
카테고리	category	VARCHAR(255)	NOT NULL				
키워드	keyword	VARCHAR(255)	NOT NULL				

스테이징 테이블

게시물 지표				mart.tb_posts_metric			
게시물 ID	post_id	VARCHAR(255)	NOT NULL				
차 모델명	car_name	VARCHAR(255)	NOT NULL				
웹 데이터 소스	source	VARCHAR(255)	NOT NULL				
좋아요 수	like_cnt	BIGINT	NULL				
싫어요 수	dislike_cnt	BIGINT	NULL				
조회수	view_cnt	BIGINT	NULL				
댓글 수	comment_cnt	BIGINT	NULL				
수집일	ingestion_date	TIMESTAMP	NOT NULL				

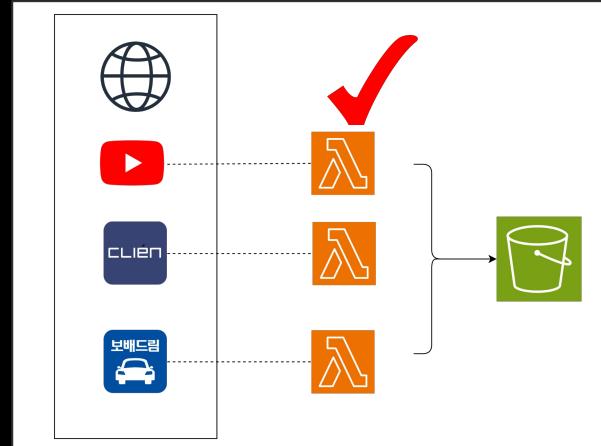
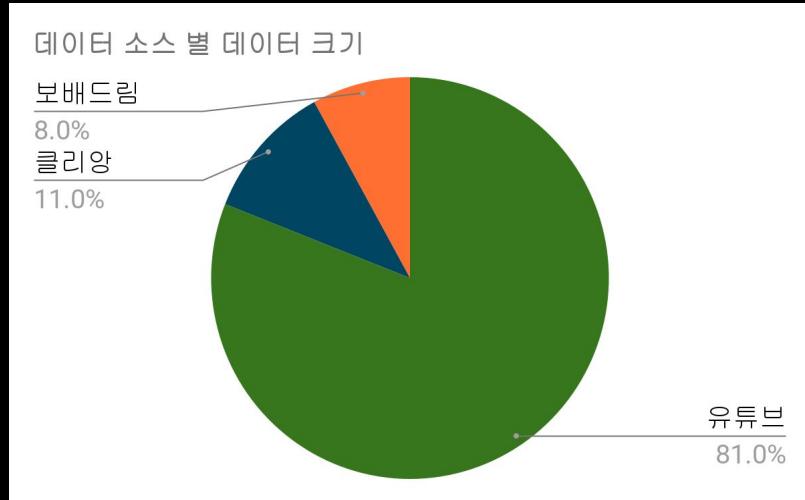
댓글 지표				mart.tb_comments_metric			
게시물 ID	post_id	VARCHAR(255)	NOT NULL				
댓글 ID	comment_id	VARCHAR(255)	NOT NULL				
차 모델명	car_name	VARCHAR(255)	NOT NULL				
웹 데이터 소스	source	VARCHAR(255)	NOT NULL				
Key	Key	Type	NOT NULL				
좋아요 수	like_cnt	BIGINT	NULL				
싫어요 수	dislike_cnt	BIGINT	NULL				
수집일	ingestion_date	TIMESTAMP	NOT NULL				

키워드				mart.tb_keywords			
게시물 ID	post_id	VARCHAR(255)	NOT NULL				
댓글 ID	comment_id	VARCHAR(255)	NULL				
차 모델명	car_name	VARCHAR(255)	NOT NULL				
웹 데이터 소스	source	VARCHAR(255)	NOT NULL				
문장 타입	type	VARCHAR(255)	NOT NULL				
문장	senetence	VARCHAR(65535)	NOT NULL				
카테고리	category	VARCHAR(255)	NOT NULL				
키워드	keyword	VARCHAR(255)	NOT NULL				
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL				
수집일	ingestion_date	TIMESTAMP	NOT NULL				

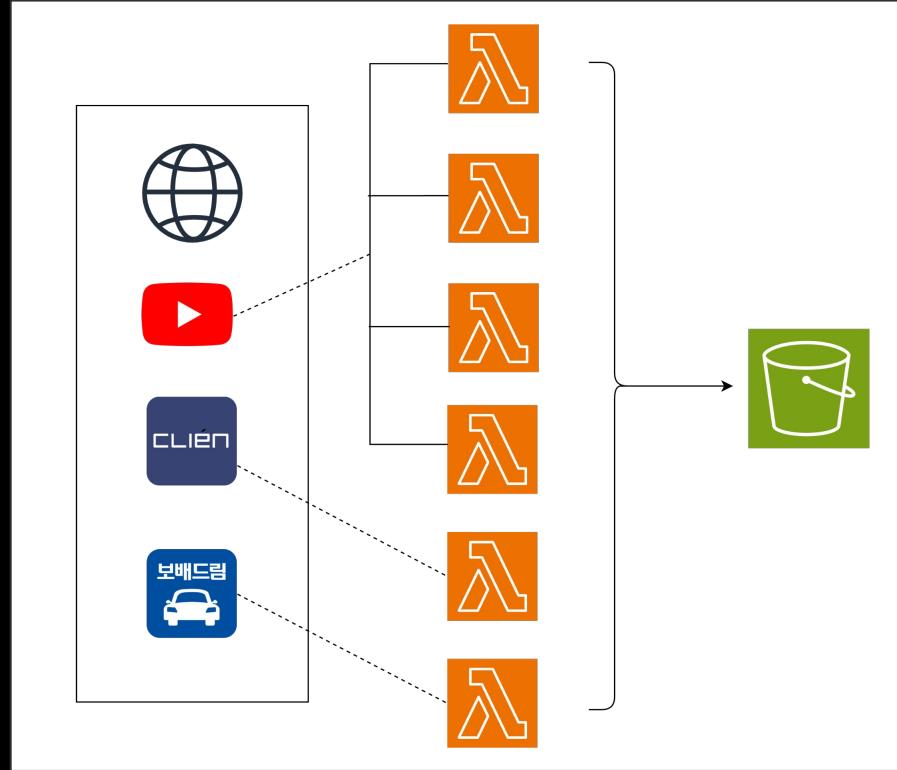
#4. 불균형한 데이터 분산하기

#4. 데이터 비대칭

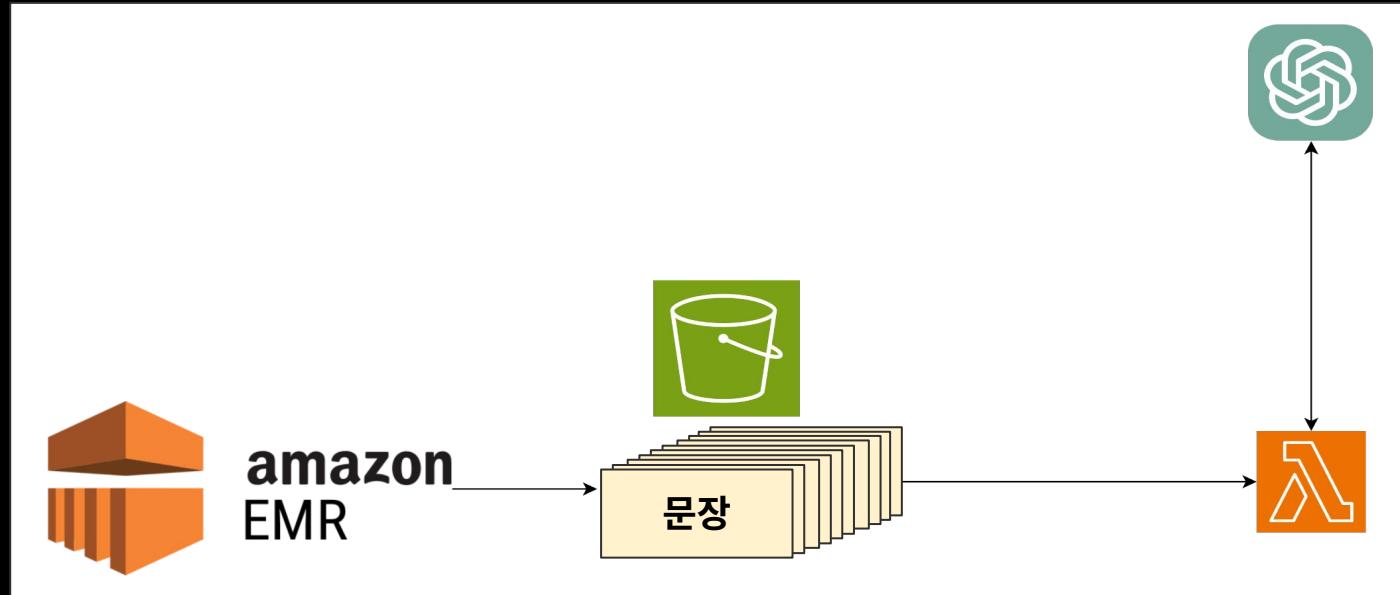
- 실제 데이터 분포는?

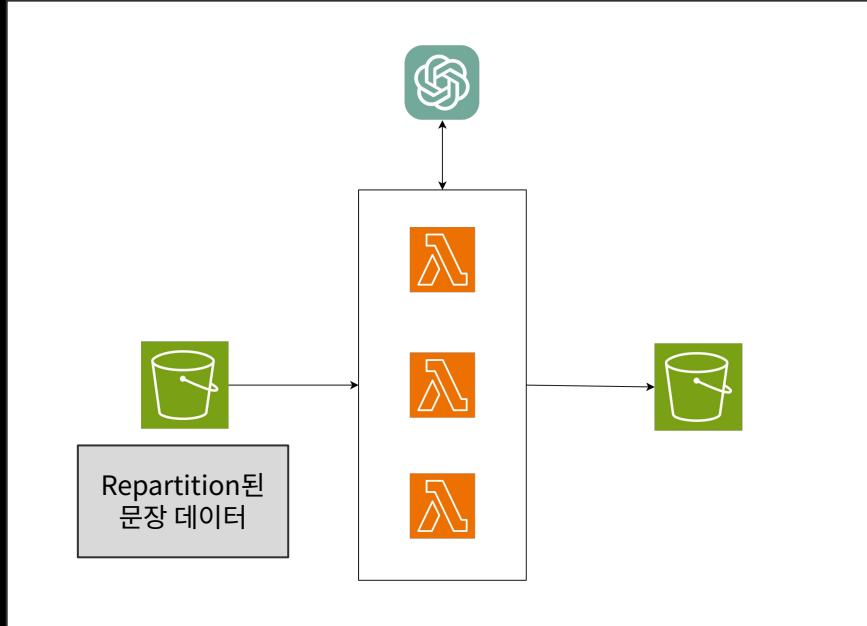


#4. 데이터 비대칭 문제 개선



#4. 문장 데이터 불균형





**Repartition한 데이터는
이후에 병렬적으로 분석처리.
문장의 키워드/감성 분석**

#5. 신뢰할 수 없는 외부 데이터 소스



우리가 컨트롤 할 수 없는 **외부 데이터 소스**

차라리 빠르게 장애를 **감지**하고 **복구**하는 것에 집중하자!

장애 감지

장애 상황, 장애 가능성이 높은 상황을 빨리 알리자!

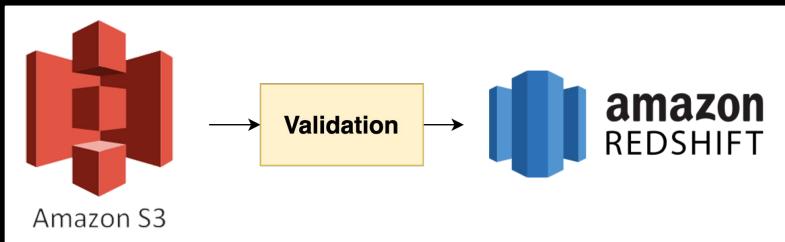
```
{  
    "commentNickname": "slt",  
    "commentContent": "@SweetBeen\n님 캐스퍼 EV, 베뉴 추천이 많네요. 다시 고려해 봐야겠습니다.",  
    "commentLikeCount": 0,  
    "commentDislikeCount": 0,  
    "commentDate": 1735237603  
},  
{  
    "commentNickname": "Zahnarzt",  
    "commentContent": "소형차에 어라운드 뷰 없는거는 큰 단점은 아닌거 같아요",  
    "commentLikeCount": "좋아요 1/",  
    "commentDislikeCount": 0,  
    "commentDate": 1735236686  
},  
{  
    "commentNickname": "slt",  
    "commentContent": "@Zahnarzt\n님 네, 객관적으로는 그런데, 주 운전자가 원하는 기능입니다.",  
    "commentLikeCount": 2,  
    "commentDislikeCount": 0,  
    "commentDate": 1735236809  
},
```

```
{  
    "commentNickname": "slt",  
    "commentContent": "@SweetBeen\n님 캐스퍼 EV, 베뉴 추천이 많네요. 다시 고려해 봐야겠습니다.",  
    "commentLikeCount": 0,  
    "commentDislikeCount": 0,  
    "commentDate": 1735237603  
},  
{  
    "commentNickname": "Zahnarzt",  
    "commentContent": "소현차에 어려운 듯 뷔 없는거는 큰 단점은 아닌거 같아요",  
    "commentLikeCount": "좋아요 1/",  
    "commentDislikeCount": 0,  
    "commentDate": 1735236686  
},  
{  
    "commentNickname": "slt",  
    "commentContent": "@Zahnarzt\n님 네, 객관적으로는 그런데, 주 운전자가 원하는 기능입니다.",  
    "commentLikeCount": 2,  
    "commentDislikeCount": 0,  
    "commentDate": 1735236809  
},
```

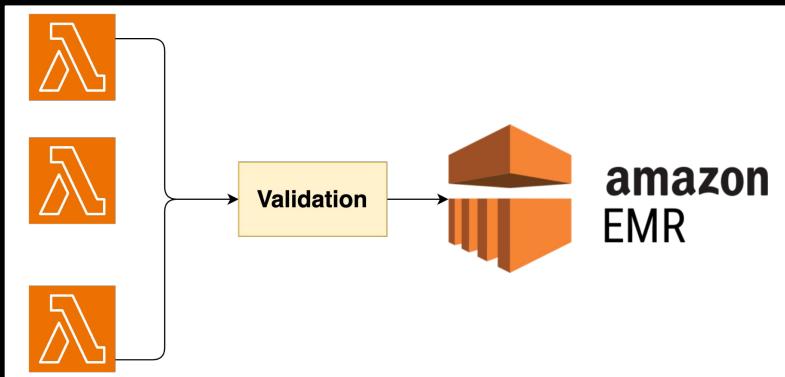
```
{  
    "commentNickname": "slt",  
    "commentContent": "@SweetBee\n님 캐스퍼 EV, 베뉴 추천이 많네요. 다시 고려해 봐야겠습니다.",  
    "commentLikeCount": 0,  
    "commentDislikeCount": 0,  
    "commentDate": 1735237603  
},  
{  
    "commentNickname": "Zahnarzt",  
    "commentContent": "소현차에 어려운 듯 뷔 없는거는 큰 단점은 아닙니다.",  
    "commentLikeCount": "좋아요 1/",  
    "commentDislikeCount": 0,  
    "commentDate": 1735236686  
},  
{  
    "commentNickname": "slt",  
    "commentContent": "@Zahnarzt\n님 네, 객관적으로는 그런데, 주 운전자가 원하는 기능입니다.",  
    "commentLikeCount": 2,  
    "commentDislikeCount": 0,  
    "commentDate": 1735236809  
},
```



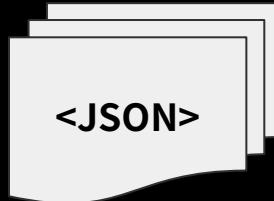
1. 데이터 검증



파이프라인 중간중간
데이터 탑입을 검사하자!



1. 데이터 검증



샘플링 후 타입 검사



메타데이터로 스키마 검사

2. 알림 시스템



Info



Warning



Error



Info



파이프라인 진행 과정에 대한 안내

ex) 파이프라인 시작/성공 안내

Info



Airflow Message 앱 오후 1:47

INFO: ETL 완료했어요!!!



INFO ETL 완료했어요!!!

Dag ID: `etl.single_model-unify`



Open DAG in Airflow UI

대한 안내

공 안내

Warning



개발자의 확인과 대응이 필요한 상황

WARNING: `bobae`에서 실패한 URL이 있습니다.

 **WARNING** `bobae`에서 실패한 URL이 있습니다.

Dag ID: `etl.single_model-unify`



Open DAG in Airflow UI

실패한 데이터 경고

“일부 데이터는 복구가 필요해요.”

 incoming-webhook 작성자: 오후 4:05

classify-senetence-mem-use-alarm state is now ALARM: Threshold Crossed: 1 out of the last 1 datapoints [97.0 (25/02/25 07:04:00)] was greater than the threshold (50.0) (minimum 1 datapoint for OK -> ALARM transition).

⚠️ WARNING classify-senetence-mem-use-alarm state is now ALARM: Threshold Crossed: 1 out of the last 1 datapoints [97.0 (25/02/25 07:04:00)] was greater than the threshold (50.0) (minimum 1 datapoint for OK -> ALARM transition).

Details

```
{'Records': [{"EventSource': 'aws:sns', 'EventVersion': '1.0', 'EventSubscriptionArn': 'arn:aws:sns:ap-northeast-2:910534606964:classify-sentence-memory-monitoring:9035f402-670b-412d-a80b-6170f86a841c', 'Sns': {'Type': 'Notification', 'MessageId': '2c27f91c-69eb-5380-93bb-992b041e8b9f', 'TopicArn': 'arn:aws:sns:ap-northeast-2:910534606964:classify-sentence-memory-monitoring:9035f402-670b-412d-a80b-6170f86a841c'}}]}
```

[더 보기](#)

리소스 사용량 경고

“평소보다 리소스 사용량이 많아요.”

“장애가 생길수도 있으니 조심하세요.”

Error



즉각적인 대응이 필요한 장애 상황

ex) 태스크 실패

데이터 검증 실패

Airflow Message 📡 오후 5:44

Task `validate_parquet` failed.

X ERROR Task `validate_parquet` failed.

Dag ID: `etl.single_model-unify`

Run ID: `scheduled__2025-01-31T12:00:00+00:00`

Reason:

```
('Lambda function execution resulted in error', {'ResponseMetadata': {'RequestId': 'db76c378-8dc3-4d89-a457-e1bb91b4d7fb', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Tue, 25 Feb 2025 08:44:38 GMT', 'content-type': 'application/json', 'content-length': '297', 'connection': 'keep-alive', '...'}}}
```

Task ID: `validate_parquet`

Date: `2025-01-31`

Log URL: [View Log](#)

[더 보기](#)

[Open in Airflow UI](#)

API 사용량 초과

Task `youtube.collect` failed.

X ERROR Task `youtube.collect` failed.

Dag ID: `etl.single_model_santafe`

Run ID: `scheduled__2025-02-20T12:00:00+00:00`

Reason:

```
('Lambda function execution resulted in error', {'ResponseMetadata': {'RequestId': 'f585462e-3533-4ef7-8378-cacb3b787fbb', 'HTTPStatusCode': 200, 'HTTPHeaders': {'date': 'Mon, 24 Feb 2025 10:48:43 GMT', 'content-type': 'application/json', 'content-length': '1100', 'connection': 'keep-alive'...}}}
```

Task ID: `youtube.collect`

Date: `2025-02-20`

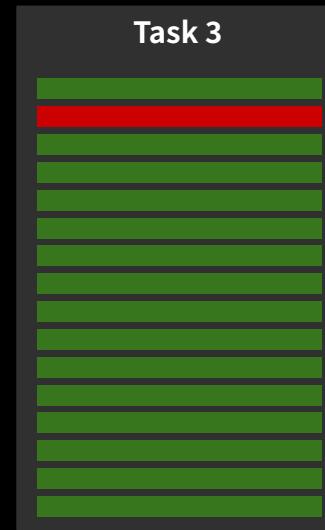
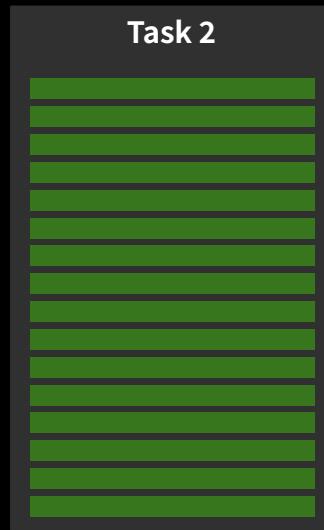
Log URL: [View Log](#)

[더 보기](#)

[Open in Airflow UI](#)

장애 복구

장애로부터 빠르게 정상화 시키자!



Task 1, 3는 처음부터 다시?

1. 실패한 데이터 저장

HTML 파싱을 실패한 URL

삭제된 게시글

Spark에서 발견된 결측/이상치

...

일단 저장



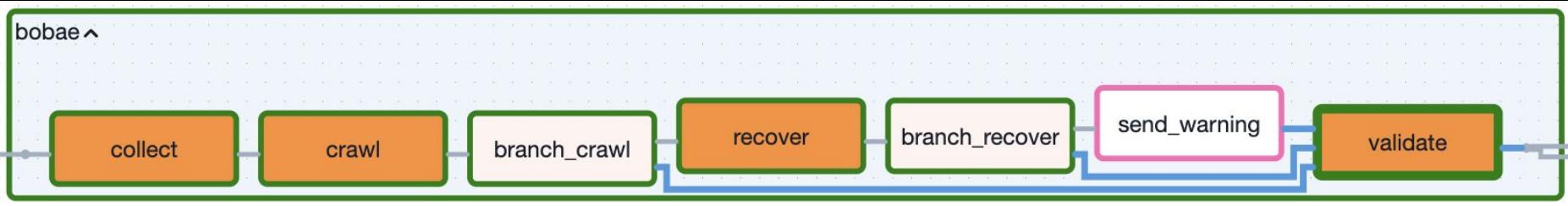
Warning 알림



전체 파이프라인을 실행할 필요 없이
실패한 데이터에 대해서만 처리해주면 끝!

2. 재시도

크롤링 단계의 발생하는 일부 에러는
단순 재시도만으로도 해결된다.



실패한 레코드만 다시 처리

#6 사연 많은 데이터 모델링

SNS 데이터의 계층 정보

- 게시글/댓글

Q/A 디올뉴싼타페 가솔린 승차감이 어떤가요? [7]

4,902 | 2025-02-16 17:44:23 수정일 : 2025-02-16 17:45:41 | 121.♡.37.233

Lynpapa님

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리 나게 안좋았거든요...

댓글 · [7]

[메모등기화] 을 누르면 회원메모를 할 수 있습니다.

허기나_해님

[LINK] IP 25-02-16 | 대댓글 · 공감 | 신고

쏘카에서 가솔린 빌려서 타 보고 태안에서 hmg 시닉드라이브로 하브 타 보고했는데 그래도 준대형세단인 k7프리미어 타고 있는 입장에서는 뭐 불편하거나 그런거 하나도 없었네요

2열은 아주 침깐 타 봤는데 그냥 보통의 suv느낌이었구요

터보니까 k7에서 넘어가고 싶다는 기분으로는 미국 들어버렸습니다. ccc

특히

1.) 승차감이 저속 구간에서는 훌륭하고,

2) 소음도 많이 차단되었다고 평가하고 있는데,

실제 차주분들이 느끼는 1열과 2열의 승차감이 궁금합니다.

답변에 미리 감사드립니다. ^^\n

Lynpapa님

[LINK] IP 25-02-16 | 언급 · 공감 | 신고

@허기나_해님 승차감이 전세대에 비해 상당히 좋아진걸 했나보네요..의견 감사합니다!

houseblended님

[LINK] IP 25-02-16 | 대댓글 · 공감 | 신고

작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.

단점이라면 엉치가 너무 크다고 느꼈습니다.

제이디피님

[LINK] IP 25-02-16 | 대댓글 · 공감 | 신고

제가 최근에 탄 suv중에서 이정도 가격에 이정도 승차감 내주는 차가 있을까 싶을 정도로 좋았어요.

꿀리비님

[LINK] IP 25-02-17 | 대댓글 · 공감 | 신고

개신 곳 기끼이에 현대 시승센터가 있는지 찾아보시고 시승 해보시는것도 추천드립니다. 저도 궁금해서 3월 초순경에 1.6 하이브리드 차량 시승신청 해놨어요 ㅎㅎ

Lynpapa님

[LINK] IP 25-02-17 | 언급 · 공감 | 신고

@꿀리비님 2.5 가솔린은 모델은 시승차가 많이 없더라구요..너무 궁금합니다 ㅎㅎ

디올뉴산타페 가솔린 승차감이 어떤가요?

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리 나게 안좋았거든요...

특히

1.) **승차감이 저속 구간에서는 훌륭**하고,

2) **소음도 많이 차단**되었다고 평가하고 있는데,

실제 차주분들이 느끼는 1열과 2열의 승차감이 궁금합니다.

답변에 미리 감사드립니다. ^^

쏘카에서 가솔린 빌려서 타 보고 태안에서 hmg 시닉드라이브로 하브 타 보고했는데 그래도 준대형세단인

k7프리미어 타고 있는 입장에서는 뭐 불편하거나 그런거 하나도 없었네요

2열은 아주 잠깐 타 봤는데 그냥 보통의 suv느낌이었구요

타보니까 k7에서 넘어가고 싶다는 기변욕구는 마구 들어버렸습니다. ㄷ ㄷ ㄷ

@하기나_해님 승차감이 전세대에 비해 상당히 좋아지긴 했나보네요..의견 감사합니다!

작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.

단점이라면 덩치가 너무 크다고 느꼈습니다.

제가 최근에 탄 suv중에서 이정도 가격에 이정도 승차감 내주는 차가 있을까 싶을 정도로 좋았어요.

계신 곳 가까이에 현대 시승센터가 있는지 찾아보시고 시승 해보시는것도 추천드립니다. 저도 궁금해서 3월 초순경에 1.6 하이브리드 차량 시승신청 해놨어요 ㅎ ㅎ

SNS Data: Overwrite

식별자	BIGINT
문장	VARCHAR
카테고리	VARCHAR
키워드	VARCHAR
감성 점수	DOUBLE
좋아요 수	BIGINT
조회수	BIGINT
데이터 생성일	TIMESTAMP

디올뉴산타페 가솔린 승차감이 어떤가요?

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리 나게 안좋았거든요...

특히

1.) **승차감이 저속 구간에서는 훌륭**하고,

2) **소음도 많이 차단**되었다고 평가하고 있는데,

실제 차주분들이 느끼는 1열과 2열의 승차감이 궁금합니다.

답변에 미리 감사드립니다. ^^

쏘카에서 가솔린 빌려서 타 보고 태안에서 hmg 시닉드라이브로 하브 타 보고했는데 그래도 준대형세단인

k7프리미어 타고 있는 입장에서는 뭐 불편하거나 그런거 하나도 없었네요

2열은 아주 잠깐 타 봤는데 그냥 보통의 suv느낌이었구요

타보니까 k7에서 넘어가고 싶다는 기변욕구는 마구 들어버렸습니다. ㄷ ㄷ ㄷ

@하기나_해님 승차감이 전세대에 비해 상당히 좋아지긴 했나보네요..의견 감사합니다!

작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.

단점이라면 덩치가 너무 크다고 느꼈습니다.

제가 최근에 탄 suv중에서 이정도 가격에 이정도 승차감 내주는 차가 있을까 싶을 정도로 좋았어요.

계신 곳 가까이에 현대 시승센터가 있는지 찾아보시고 시승 해보시는것도 추천드립니다. 저도 궁금해서 3월 초순경에 1.6 하이브리드 차량 시승신청 해놨어요 ㅎ ㅎ

디올뉴산타페 가솔린 승차감이 어떤가요?

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리 나게 안좋았거든요...

특히

1.) **승차감이 저속 구간에서는 훌륭**하고,

2) **소음도 많이 차단**되었다고 평가하고 있는데,

실제 차주분들이 느끼는 1열과 2열의 승차감이 궁금합니다...

답변에 미리 감사드립니다. ^^



작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.

디올뉴산타페 가솔린 승차감이 어떤가요?

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리 나게 안좋았거든요

특히

1.) **승차감이 저속 구간에서는 훌륭**하고,

2) **소음도 많이 차단**되었다고 평가하고 있는데,

실제 차주분들이 느끼는 1열과 2열의 승차감이 궁금합니다...

답변에 미리 감사드립니다. ^^



작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.

디올뉴산타페 가솔린 승차감이 어떤가요?

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리나거나 짜증나기도 했던

특히

1.) **승차감이 저속 구간에서는 훌륭**

2) **소음도 많이 차단**되었다고 평가해

실제 차주분들이 느끼는 1열과 2열의 승

답변에 미리 감사드립니다. ^^

SNS Data: Overwrite		
PK	식별자	BIGINT
	문장	VARCHAR
	카테고리	VARCHAR
	키워드	VARCHAR
	감성 점수	DOUBLE
	좋아요 수	BIGINT
	조회수	BIGINT
	데이터 생성일	TIMESTAMP

작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.

디올뉴산타페 가솔린 승차감이 어떤가요?

안녕하세요.

디올뉴싼타페(2.5)에 대해 궁금한 점이 있어 글을 올립니다.

시승기에서는 대체로 전 세대보다 훨씬 좋아졌다고 이야기하고 있습니다. 차주 인터뷰에서도 다들 전반적으로 만족하시는 것 같구요.

+ 지난 TM시절에는 2열 승차감이 '헉' 소리 나게 안좋았거든요...

특히

1.) **승차감이 저속 구간에서는 훌륭**하고,

2) **소음도 많이 차단**되었다고 평가하고 있는데,

실제 차주분들이 느끼는 1열과 2열의 승차감이 궁금합니다.

답변에 미리 감사드립니다. ^^

“디 올 뉴 싼타페”를

작년에 제주도에서 몇일동안 운전하고 다녔는데 구매하고 싶을 정도로 승차감등등 맘에 들었습니다.



SNS Data: Overwrite

식별자	BIGINT
문장	VARCHAR
카테고리	VARCHAR
키워드	VARCHAR
감성 점수	DOUBLE
좋아요 수	BIGINT
조회수	BIGINT
데이터 생성일	TIMESTAMP

게시물	post_id	CHAR(36)	NOT NULL
제목	title	VARCHAR(255)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL
본문	article	VARCHAR(65535)	NOT NULL
작성일	timestamp	BIGINT	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL
댓글 수	comment_cnt	BIGINT	NULL
댓글 수	source	VARCHAR(255)	NOT NULL
웹 데비아 소스 명	car_name	VARCHAR(255)	NOT NULL
차 도움 명			

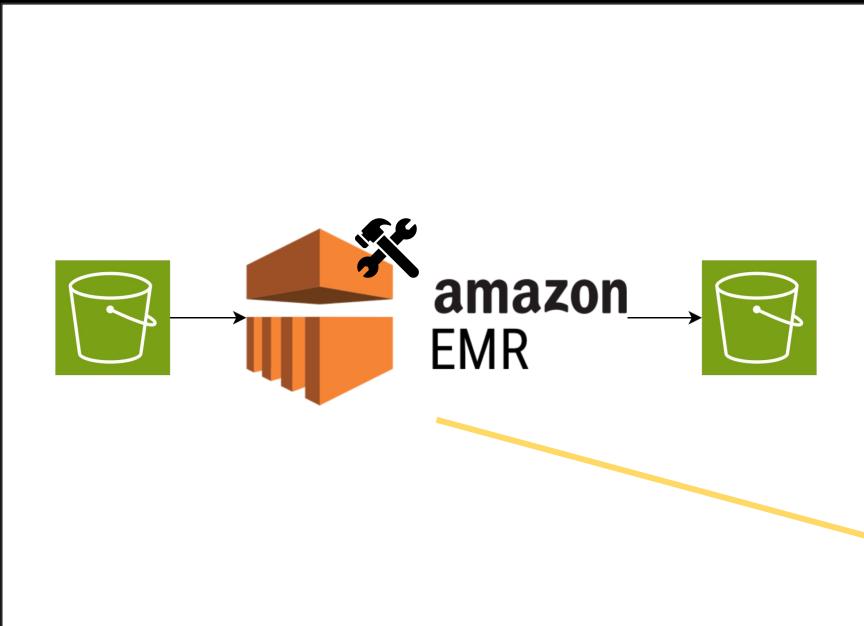
댓글	comment_id	CHAR(36)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL
본문	content	VARCHAR(65535)	NOT NULL
작성일	timestamp	BIGINT	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
게시물 ID	post_id	CHAR(36)	NOT NULL

문장	sentence_id	CHAR(36)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL
텍스트	text	VARCHAR(65535)	NOT NULL
게시물 ID	post_id	CHAR(36)	NOT NULL
댓글 ID	comment_id	CHAR(36)	NULL

문장 감성 분석	sentence_sentiment_tb		
문장 ID	sentence_id	CHAR(36)	NOT NULL
감성 점수	sentiment_score	REAL	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL

SNS Data: Overwrite

	식별자	BIGINT
	문장	VARCHAR
	카테고리	VARCHAR
	키워드	VARCHAR
	감성 점수	DOUBLE
	좋아요 수	BIGINT
	조회수	BIGINT
	데이터 생성일	TIMESTAMP

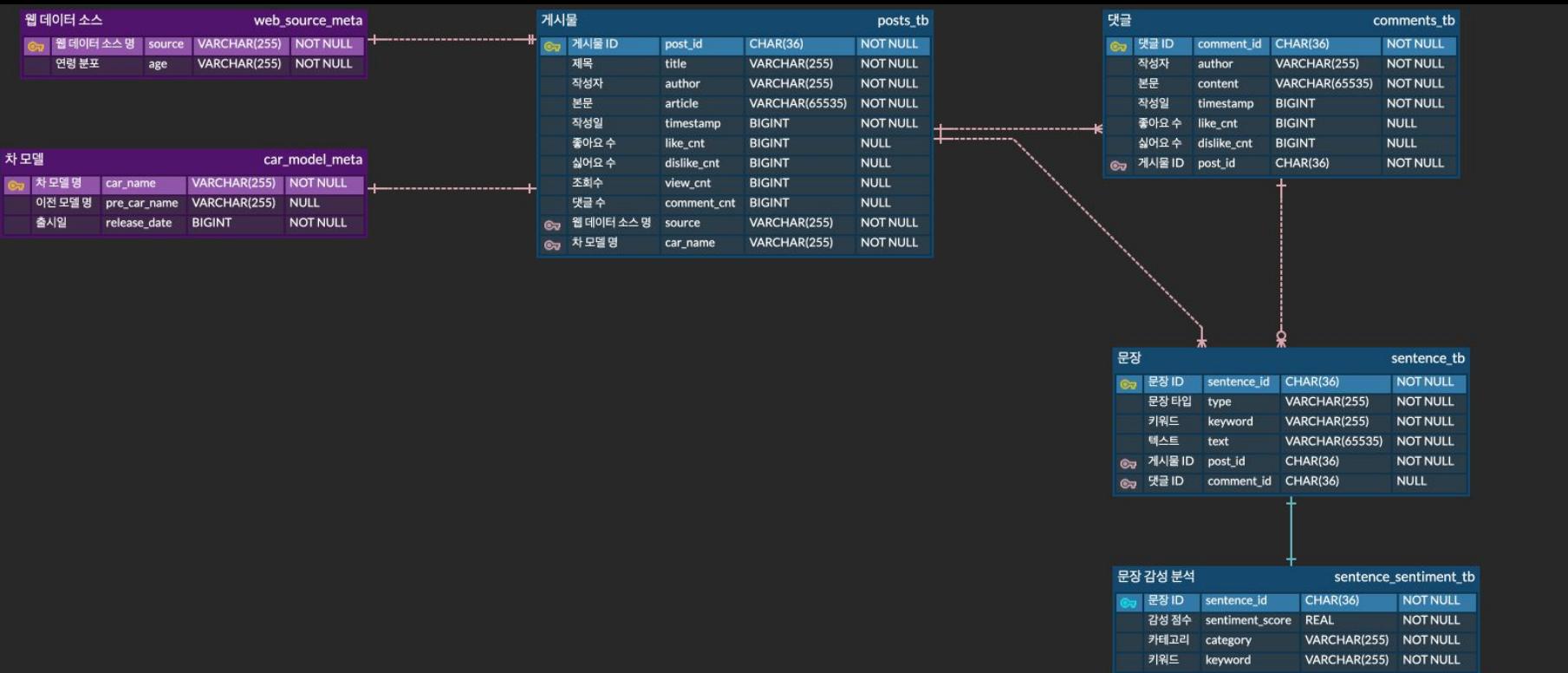


게시물	post_id	CHAR(36)	NOT NULL
제목	title	VARCHAR(255)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL
본문	article	VARCHAR(65535)	NOT NULL
작성일	timestamp	BIGINT	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL
댓글 수	comment_cnt	BIGINT	NULL
댓글 수	source	VARCHAR(255)	NOT NULL
웹 데비아 소스 명	car_name	VARCHAR(255)	NOT NULL
차 도록 명			

댓글	comment_id	CHAR(36)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL
본문	content	VARCHAR(65535)	NOT NULL
작성일	timestamp	BIGINT	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
게시물 ID	post_id	CHAR(36)	NOT NULL

문장	sentence_id	CHAR(36)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL
텍스트	text	VARCHAR(65535)	NOT NULL
게시물 ID	post_id	CHAR(36)	NOT NULL
댓글 ID	comment_id	CHAR(36)	NULL

문장 감성 분석	sentence_sentiment_tb		
문장 ID	sentence_id		
감성 점수	sentiment_score	REAL	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL



SNS 데이터의 시계열 정보

- Place-In-Time

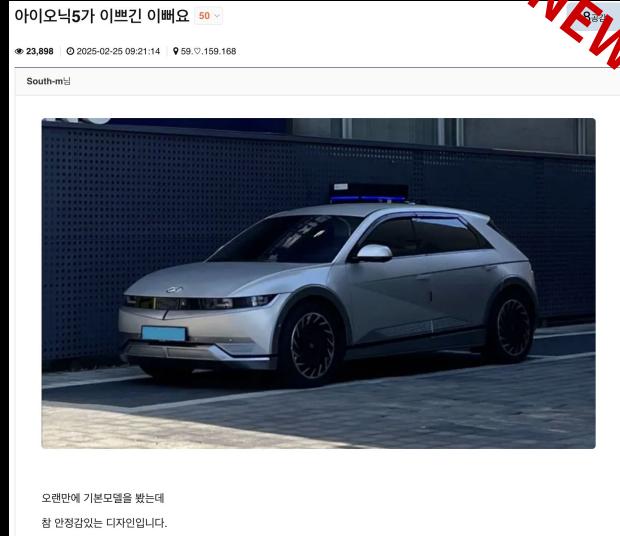
아이오닉5가 이쁘긴 이뻐요

50 ▾

2025-02-27  23,895 | ⏰ 2025-02-25 09:21:14 | 🗺 59.♡.159.168

2025-02-26  19,646

2025-02-25  10,870



SNS Data: Overwrite		
	식별자	BIGINT
문장	VARCHAR	
카테고리	VARCHAR	
키워드	VARCHAR	
감성 점수	DOUBLE	
좋아요 수	BIGINT	
조회수	BIGINT	
데이터 생성일	TIMESTAMP	

2025-02-25



Daily 조회수 변화 추이가 궁금해졌어요

2025-01-28

아이오닉5가 이쁘긴 이뻐요 50 8공감 29

아이오닉5가 이쁘긴 이뻐요 50 8공감 30

아이오닉5가 이쁘긴 이뻐요 50 8공감 31

아이오닉5가 이쁘긴 이뻐요 50 8공감 28-01

• 23,898 2025-02-25 09:21:14 59.0.159.168

South-m

오랜만에 기본모델을 봤는데
참 인정감 있는 디자인입니다.



SNS Data: Time Series

식별자	BIGINT
문장	VARCHAR
카테고리	VARCHAR
키워드	VARCHAR
감성 점수	DOUBLE
좋아요 수	BIGINT
조회수	BIGINT
데이터 생성일	TIMESTAMP
데이터 수집일	2025-02-25

2025-02-26
2025-02-27
2025-02-28



Daily 조회수 변화 추이가 궁금해졌어요



SNS Data: Overwrite

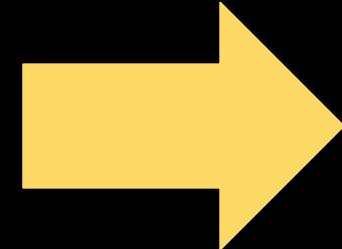
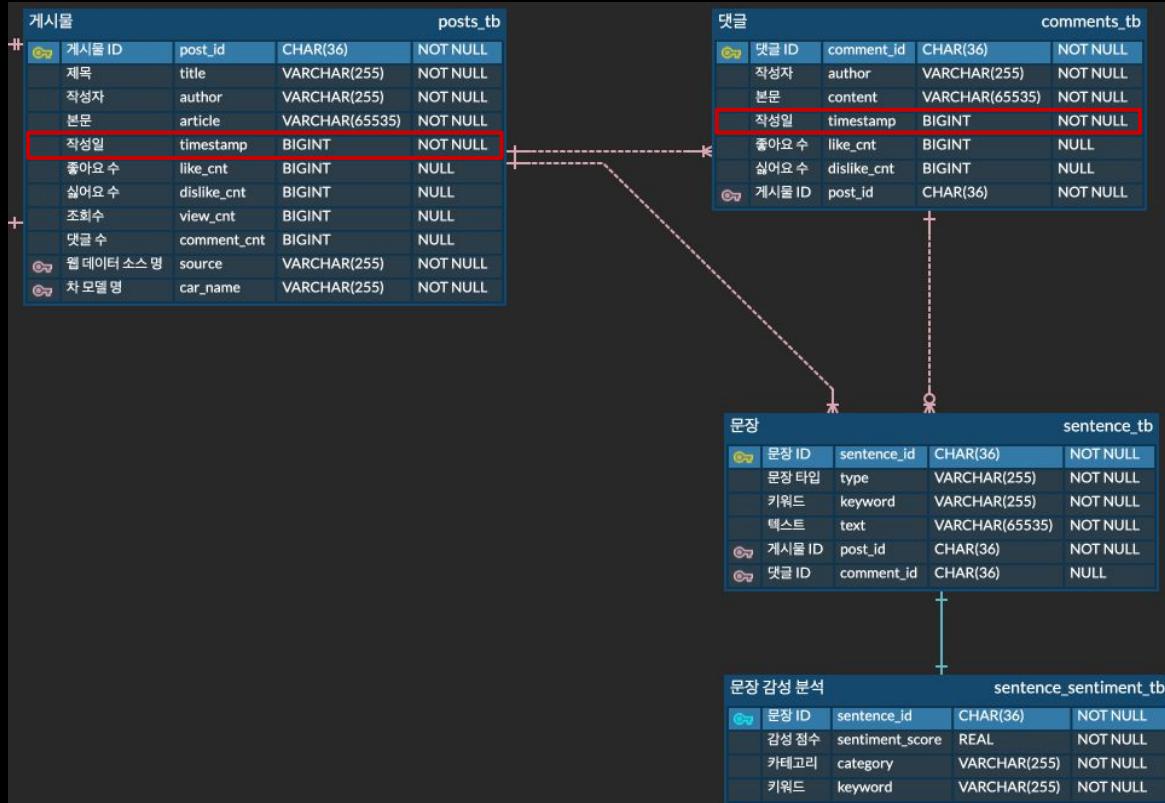
식별자	BIGINT
문장	VARCHAR
카테고리	VARCHAR
키워드	VARCHAR
감성 점수	DOUBLE
좋아요 수	BIGINT
조회수	BIGINT
데이터 생성일	TIMESTAMP



SNS Data: Time Series

식별자	BIGINT
문장	VARCHAR
카테고리	VARCHAR
키워드	VARCHAR
감성 점수	DOUBLE
좋아요 수	BIGINT
조회수	BIGINT
데이터 생성일	TIMESTAMP
데이터 수집일	TIMESTAMP

게시글/댓글 작성일은 변하지 않는 값



수집일 기준으로 시계열 정보 보존

댓글 지표		mart.tb_comments_metric		
게시물 ID	post_id	VARCHAR(255)	NOT NULL	
댓글 ID	comment_id	VARCHAR(255)	NOT NULL	
차 모델명	car_name	VARCHAR(255)	NOT NULL	
웹 데이터 소스명	source	VARCHAR(255)	NOT NULL	
Key	Key	Type	NOT NULL	
좋아요 수	like_cnt	BIGINT	NULL	
싫어요 수	dislike_cnt	BIGINT	NULL	
수집일	ingestion_date	TIMESTAMP	NOT NULL	

게시물 지표		mart.tb_posts_metric		
게시물 ID	post_id	VARCHAR(255)	NOT NULL	
차 모델명	car_name	VARCHAR(255)	NOT NULL	
웹 데이터 소스명	source	VARCHAR(255)	NOT NULL	
좋아요 수	like_cnt	BIGINT	NULL	
싫어요 수	dislike_cnt	BIGINT	NULL	
조회수	view_cnt	BIGINT	NULL	
댓글 수	comment_cnt	BIGINT	NULL	
수집일	ingestion_date	TIMESTAMP	NOT NULL	

키워드		mart.tb_keywords	
게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NULL
차 모델명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스명	source	VARCHAR(255)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL
문장	senetence	VARCHAR(65535)	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL

SNS 데이터의 시계열 정보

- Dynamic VS Static



SNS Data: Time Series		
키	식별자	BIGINT
문장	VARCHAR	
카테고리	VARCHAR	
키워드	VARCHAR	
감성 점수	DOUBLE	
좋아요 수	BIGINT	
조회수	BIGINT	
데이터 생성일	TIMESTAMP	
데이터 수집일	TIMESTAMP	

중복 발생



문장	감성 점수	좋아요 수	조회수	데이터 수집일
싼타페가 ~~	0.7	2	34	2025-02-24
싼타페가 ~~	0.7	4	65	2025-02-25
싼타페가 ~~	0.7	9	100	2025-02-26
싼타페가 ~~	0.7	14	145	2025-02-27



SNS Data: Time Series		
키	식별자	BIGINT
문장	VARCHAR	
카테고리	VARCHAR	
키워드	VARCHAR	
감성 점수	DOUBLE	
좋아요 수	BIGINT	
조회수	BIGINT	
데이터 생성일	TIMESTAMP	
데이터 수집일	TIMESTAMP	

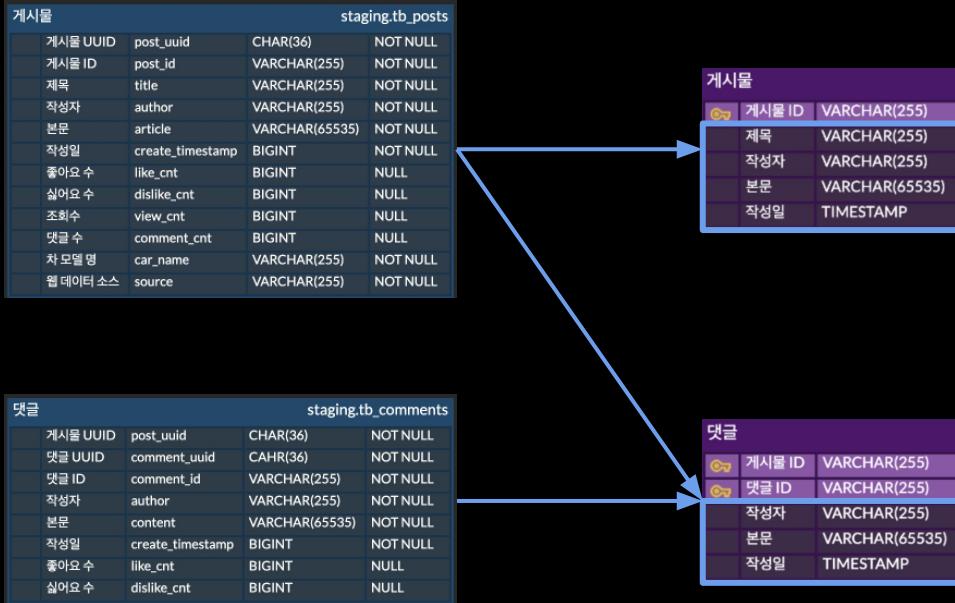


Static Dynamic

문장	감성 점수	좋아요 수	조회수	데이터 수집일
싼타페가 ~~	0.7	2	34	2025-02-24
싼타페가 ~~	0.7	4	65	2025-02-25
싼타페가 ~~	0.7	9	100	2025-02-26
싼타페가 ~~	0.7	14	145	2025-02-27

Merge Load

- 변하지 않는 값



Insert Load

- 시간에 따라 변하는 값

게시물			
staging.tb_posts			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL
게시물 ID	post_id	VARCHAR(255)	NOT NULL
제목	title	VARCHAR(255)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL
본문	article	VARCHAR(65535)	NOT NULL
작성일	create_timestamp	BIGINT	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL
댓글 수	comment_cnt	BIGINT	NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스	source	VARCHAR(255)	NOT NULL

게시물	
게시물 ID	VARCHAR(255)
제목	VARCHAR(255)
작성자	VARCHAR(255)
본문	VARCHAR(65535)
작성일	TIMESTAMP

댓글			
staging.tb_comments			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL
댓글 UUID	comment_uuid	CAH(36)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL
본문	content	VARCHAR(65535)	NOT NULL
작성일	create_timestamp	BIGINT	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL



댓글	
게시물 ID	VARCHAR(255)
댓글 ID	VARCHAR(255)
작성자	VARCHAR(255)
본문	VARCHAR(65535)
작성일	TIMESTAMP



문장			
staging.tb_sentences			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL
댓글 UUID	comment_uuid	CHAR(36)	NULL
문장 UUID	sentence_uuid	CHAR(36)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL
문장	sentence	VARCHAR(65535)	NOT NULL

차 모델	
차 모델 명	VARCHAR(255)
이전 모델 명	VARCHAR(255)
출시일	TIMESTAMP

키워드			
staging.tb_keywords			
문장 UUID	sentence_uuid	CHAR(36)	NOT NULL
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL

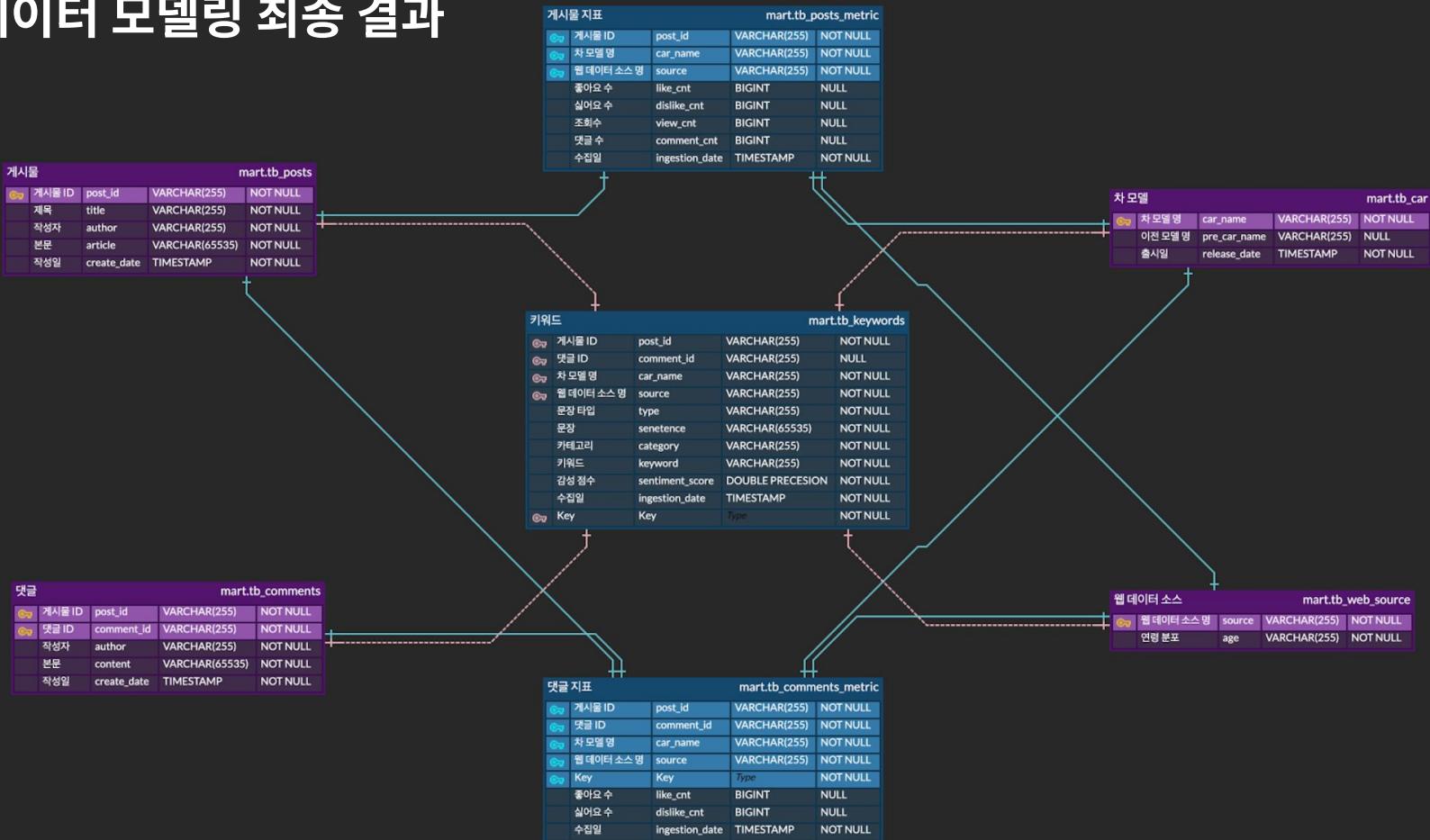
웹 데이터 소스	
웹 데이터 소스 명	VARCHAR(255)
연령 분포	VARCHAR(255)

게시물 지표	
mart.tb_posts_metric	
게시물 ID	post_id
차 모델 명	car_name
웹 데이터 소스 명	source
좋아요 수	like_cnt
싫어요 수	dislike_cnt
조회수	view_cnt
댓글 수	comment_cnt
수집일	ingestion_date

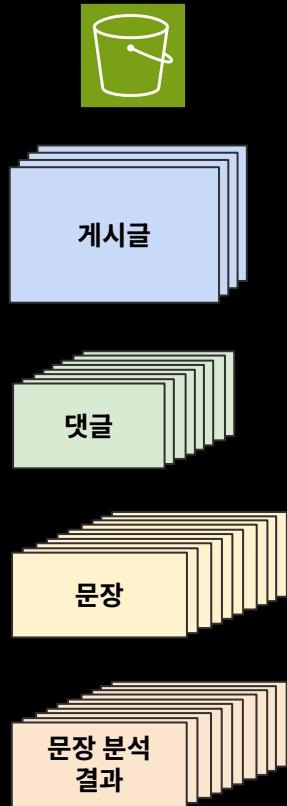
키워드	
mart.tb_keywords	
게시물 ID	post_id
댓글 ID	comment_id
차 모델 명	car_name
웹 데이터 소스 명	source
문장 타입	type
문장	sentence
카테고리	category
키워드	keyword
감성 점수	sentiment_score
수집일	ingestion_date

댓글 지표	
mart.tb_comments_metric	
게시물 ID	post_id
댓글 ID	comment_id
차 모델 명	car_name
웹 데이터 소스 명	source
Key	Key
좋아요 수	like_cnt
싫어요 수	dislike_cnt
수집일	ingestion_date

데이터 모델링 최종 결과



데이터 적재 프로세스

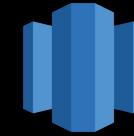


Extract



Transform

Load



게시물 지표		mart.tb_posts_metric	
PK	필드명	타입	Nullable
게시물 ID	post_id	VARCHAR(255)	NOT NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스 명	source	VARCHAR(255)	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL
댓글 수	comment_cnt	BIGINT	NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL

댓글 지표		mart.tb_comments_metric	
PK	필드명	타입	Nullable
게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NOT NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스 명	source	VARCHAR(255)	NOT NULL
Key	Key	Type	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL

키워드		mart.tb_keywords	
PK	필드명	타입	Nullable
게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스 명	source	VARCHAR(255)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL
문장	senetence	VARCHAR(65535)	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL



게시물				staging.tb_posts			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL	게시물 ID	post_id	VARCHAR(255)	NOT NULL
제목	post_id	VARCHAR(255)	NOT NULL	차 모델명	car_name	VARCHAR(255)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL	웹 데이터 소스명	source	VARCHAR(255)	NOT NULL
본문	article	VARCHAR(65535)	NOT NULL	좋아요 수	like_cnt	BIGINT	NULL
작성일	create_timestamp	BIGINT	NOT NULL	싫어요 수	dislike_cnt	BIGINT	NULL
좋아요 수	like_cnt	BIGINT	NULL	조회수	view_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL	댓글 수	comment_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL	수집일	ingestion_date	TIMESTAMP	NOT NULL
댓글 수	comment_cnt	BIGINT	NULL				
차 모델명	car_name	VARCHAR(255)	NOT NULL				
웹 데이터 소스	source	VARCHAR(255)	NOT NULL				

댓글				staging.tb_comments			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL	게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 UUID	comment_uuid	CHAR(36)	NOT NULL	댓글 ID	comment_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NOT NULL	차 모델명	car_name	VARCHAR(255)	NOT NULL
작성자	author	VARCHAR(255)	NOT NULL	웹 데이터 소스명	source	VARCHAR(255)	NOT NULL
본문	content	VARCHAR(65535)	NOT NULL	Key	Key	Type	NOT NULL
작성일	create_timestamp	BIGINT	NOT NULL	좋아요 수	like_cnt	BIGINT	NULL
좋아요 수	like_cnt	BIGINT	NULL	싫어요 수	dislike_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL	수집일	ingestion_date	TIMESTAMP	NOT NULL

문장				staging.tb_sentences			
게시물 UUID	post_uuid	CHAR(36)	NOT NULL	게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 UUID	comment_uuid	CHAR(36)	NULL	댓글 ID	comment_id	VARCHAR(255)	NOT NULL
문장 UUID	sentence_uuid	CHAR(36)	NOT NULL	차 모델명	car_name	VARCHAR(255)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL	웹 데이터 소스명	source	VARCHAR(255)	NOT NULL
문장	sentence	VARCHAR(65535)	NOT NULL	문장 타입	type	VARCHAR(255)	NOT NULL

키워드				staging.tb_keywords			
문장 UUID	sentence_uuid	CHAR(36)	NOT NULL	게시물 ID	post_id	VARCHAR(255)	NOT NULL
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL	댓글 ID	comment_id	VARCHAR(255)	NULL
카테고리	category	VARCHAR(255)	NOT NULL	차 모델명	car_name	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL	웹 데이터 소스명	source	VARCHAR(255)	NOT NULL

게시물 지표

mart.tb_posts_metric

게시물 ID	post_id	VARCHAR(255)	NOT NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스 명	source	VARCHAR(255)	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL
댓글 수	comment_cnt	BIGINT	NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL

댓글 지표

mart.tb_comments_metric

게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NOT NULL
차 모델명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스명	source	VARCHAR(255)	NOT NULL
Key	Key	Type	NOT NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL

키워드

mart.tb_keywords

게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NULL
차 모델명	car_name	VARCHAR(255)	NOT NULL
웹 데이터 소스명	source	VARCHAR(255)	NOT NULL
문장 타입	type	VARCHAR(255)	NOT NULL
문장	senetence	VARCHAR(65535)	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL
감성 점수	sentiment_score	DOUBLE PRECISION	NOT NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL

#7. 결론

**비즈니스와 데이터를 모두 이해하는
데이터 엔지니어**

김건아



<https://github.com/KimGona>

- Spark 데이터 처리
- 문장 분석
- Spark 최적화

김민재



<https://github.com/openkmj>

- 워크플로우 관리
- AWS 서비스 연동
- Spark 최적화

양주영



<https://github.com/yjy323>

- 데이터 모델링
- DW 설계 및 운영
- 시각화

감사합니다.

Appendix

Appendix: Spark UUID 생성기 (1)

UDF

```
def generate_uuid():
    return str(uuid.uuid4())

uuid_udf = udf(generate_uuid, StringType())
```

Appendix: Spark UUID 생성기 (1)

Deterministic: 같은 입력에 대해서는 같은 결과가 나온다.

Catalyst optimizer는 이 성질을 이용해서 최적화 -> **부작용 발생**

```
== Optimized Logical Plan ==
InsertIntoHadoopFsRelationCommand s3://the-all-new-bucket/uuid_test/sentence_data, false, Parquet, [path=s3://the-all-new-bucket/uuid_test/sentence_data], Overwrite, [post_uuid, comment_uuid, sentence_uuid, type, sentence]
+- WriteFiles
  +- Repartition 1, false
    +- Union false, false
      +- Project [pythonUDF0#242 AS post_uuid#25, null AS comment_uuid#229, pythonUDF0#242 AS sentence_uuid#161, post AS type#162, title#3 AS sentence#163]
      :   +- BatchEvalPython [generate_uuid()#241, pythonUDF0#242]
      :     +- Project [title#3]
      :       +- Filter (isNotNull(title#3) AND (length(title#3) > 10))
      :         +- Relation [car_name#0,source#1,id#2,title#3,nickname#4,article#5,view_count#6,like_count#7,dislike_count#8,date#9,comment_count#10,comments#11] json
      +- Project [post_uuid#25, null AS comment_uuid#230, pythonUDF0#244 AS sentence_uuid#194, post AS type#195, sentence#174]
        +- BatchEvalPython [generate_uuid()#1701, pythonUDF0#244]
          +- Filter (length(sentence#174) > 10)
            +- Generate explode(split(article#185,
, -1)), [1], false, [sentence#174]
              +- Project [pythonUDF0#243 AS post_uuid#25, article#185]
                +- BatchEvalPython [generate_uuid()#241, pythonUDF0#243]
                  +- Project [article#185]
                    +- Relation [car_name#180,source#181,id#182,title#183,nickname#184,article#185,view_count#186,like_count#187,dislike_count#188,date#189,comment_count#190,comments#191] json
      +- Project [post_uuid#25, pythonUDF0#246 AS comment_uuid#225, pythonUDF0#246 AS sentence_uuid#226, comment AS type#227, comment#41.comment_content AS sentence#228]
        +- BatchEvalPython [generate_uuid()#561, pythonUDF0#246]
          +- Filter (isNotNull(comment#41.comment_content) AND (length(comment#41.comment_content) > 10))
            +- Generate explode(comments#223), [0], false, [comment#41]
              +- Project [comments#223, pythonUDF0#245 AS post_uuid#25]
                +- BatchEvalPython [generate_uuid()#241, pythonUDF0#245]
                  +- Project [comments#223]
                    +- Filter ((isNotNull(comment_count#222) AND (comment_count#222 > 0)) AND ((size(comments#223, true) > 0) AND isNotNull(comments#223)))
                      +- Relation [car_name#212,source#213,id#214,title#215,nickname#216,article#217,view_count#218,like_count#219,dislike_count#220,date#221,comment_count#222,comments#223] json
```

Appendix: Spark UUID 생성기 (1)

Deterministic: 같은 입력에 대해서는 같은 결과가 나온다.

Catalyst optimizer는 이 성질을 이용해서 최적화 -> **부작용 발생**

```
== Optimized Logical Plan ==
InsertIntoHadoopFsRelationCommand s3://the-all-new-bucket/uuid_test/sentence_data, false, Parquet, [path=s3://the-all-new-bucket/uuid_test/sentence_data], Overwrite, [post_uuid, comment_uuid, sentence_uuid, type, sentence]
+- WriteFiles
  +- Partition 1, false
    +- Union false, false
      :- Project [pythonUDF0#242 AS post_uuid#25, null AS comment_uuid#229, pythonUDF0#242 AS sentence_uuid#161, post AS type#162, title#3 AS sentence#163]
      :  +- BatchEvalPython [generate_uuid()#241, pythonUDF0#242]
      :    +- Project [title#3]
      :      +- Filter (isNotNull(title#3) AND (length(title#3) > 10))
      :        +- Relation [car_name#0,source#1,id#2,title#3,nickname#4,article#5,view_count#6,like_count#7,dislike_count#8,date#9,comment_count#10,comments#11] json
      :- Project [post_uuid#25, null AS comment_uuid#230, pythonUDF0#244 AS sentence_uuid#194, post AS type#195, sentence#174]
      :  +- BatchEvalPython [generate_uuid()#1701, pythonUDF0#244]
      :    +- Filter (length(sentence#174) > 10)
      :      +- Generate explode(split(article#185,
      , -1)), [sentence#174], false, [sentence#174]
      :        +- Project [pythonUDF0#243 AS post_uuid#25, article#185]
      :          +- BatchEvalPython [generate_uuid()#241, pythonUDF0#243]
      :            +- Project [article#185]
      :              +- Relation [car_name#180,source#181,id#182,title#183,nickname#184,article#185,view_count#186,like_count#187,dislike_count#188,date#189,comment_count#190,comments#191] json
      +- Project [post_uuid#25, pythonUDF0#246 AS comment_uuid#225, pythonUDF0#246 AS sentence_uuid#226, comment AS type#227, comment#41.comment_content AS sentence#228]
        +- BatchEvalPython [generate_uuid()#56, pythonUDF0#246]
          +- Filter (isNotNull(comment#41.comment_content) AND (length(comment#41.comment_content) > 10))
            +- Generate explode(comments#223), [false, comment#41]
          +- Project [comments#223, pythonUDF0#245 AS post_uuid#25]
            +- BatchEvalPython [generate_uuid()#241, pythonUDF0#245]
              +- Project [comments#223]
                +- Filter ((isNotNull(comment_count#222) AND (comment_count#222 > 0)) AND ((size(comments#223, true) > 0) AND isNotNull(comments#223)))
                  +- Relation [car_name#212,source#213,id#214,title#215,nickname#216,article#217,view_count#218,like_count#219,dislike_count#220,date#221,comment_count#222,comments#223] json
```

RDD 연산마다 매번 다른 post_uuid를 생성해서 사용함!

Appendix: Spark UUID 생성기 (2)

Non-deterministic UDF

```
def generate_uuid():
    return str(uuid.uuid4())

uuid_udf = udf(generate_uuid, StringType()).asNondeterministic()
```

Appendix: Spark UUID 생성기 (2)

Non-deterministic: 매 실행마다 결과가 달라진다. (ex. random(), uuid(), now())

UDF에 이를 명시해주면 Catalyst 최적화가 일부 비활성화된다.

```
== Optimized Logical Plan ==
InserIntoHadoopRelationCommand s3://the-all-new-bucket/uuid_test/sentence_data, false, Parquet, [path=s3://the-all-new-bucket/uuid_test/sentence_data], Overwrite, [post_uuid, comment_uuid, sentence_uuid, type, sentence]
+- WriteFiles
  +- Repartition 1, false
    +- Union false
      +- Project [post_uuid#25, null AS comment_uuid#217, sentence_uuid#149, post AS type#150, sentence#151]
        :   +- Project [post_uuid#25, pythonUDF#0#233 AS sentence_uuid#149, title#3 AS sentence#151]
        :     +- Filter (isNotNull(title#3) AND (title#3 > 0))
        :       +- BatchEvalPython [generate_uuid()#241, pythonUDF#0#233]
        :         +- Project [title#3, pythonUDF#0#232 AS post_uuid#25]
        :           +- BatchEvalPython [generate_uuid()#241, pythonUDF#0#232]
        :             +- Project [title#3]
        :               +- Relation [car_name#0,source#1,id#2,title#3,nickname#4,article#5,view_count#6,like_count#7,dislike_count#8,date#9,comment_count#10,comments#11] json
      :- Project [post_uuid#25, null AS comment_uuid#218, sentence_uuid#159, post AS type#183, sentence#162]
        :   +- Project [post_uuid#25, pythonUDF#0#235 AS sentence_uuid#159, sentence#162]
        :     +- Filter (length(sentence#162) < 10)
        :       +- BatchEvalPython [generate_uuid()#158], [pythonUDF#0#235]
        :         +- Generate explode(split(article#173,
        , -1)), [1], false, [sentence#162]
          :           +- Project [post_uuid#25, article#173]
          :             +- Project [article#173, pythonUDF#0#234 AS post_uuid#25]
          :               +- BatchEvalPython [generate_uuid()#241, pythonUDF#0#234]
          :                 +- Project [article#173]
          :                   +- Relation [car_name#168,source#169,id#170,title#171,nickname#172,article#173,view_count#174,like_count#175,dislike_count#176,date#177,comment_count#178,comments#179] json
      +- Project [post_uuid#25, comment_uuid#57, sentence_uuid#191, comment AS type#215, sentence#194]
        +- Project [post_uuid#25, comment_uuid#57, pythonUDF#0#238 AS sentence_uuid#192, content#60 AS sentence#194]
          +- Filter (isNotNull(content#60) AND (content#60 > 10))
            +- BatchEvalPython [generate_uuid()#241, pythonUDF#0#238]
              +- Project [post_uuid#25, pythonUDF#0#237 AS comment_uuid#57, comment#41 AS content#60]
                +- BatchEvalPython [generate_uuid()#231, pythonUDF#0#237]
                  +- Generate explode(_extract_comment_content#231), [0], false, [comment#41]
                    +- Project [comments#211,comment_content AS _extract_comment_content#231, post_uuid#25]
                      +- Project [comment_count#210, comments#211, pythonUDF#0#236 AS post_uuid#25]
                        +- Filter ((isNotNull(comment_count#210) AND (comment_count#210 > 0)) AND ((size(comments#211.comment_content, true) > 0) AND isNotNull(comments#211.comment_content)))
                          +- BatchEvalPython [generate_uuid()#241, pythonUDF#0#236]
                            +- Project [comment_count#210, comments#211]
                              +- Relation [car_name#200,source#201,id#202,title#203,nickname#204,article#205,view_count#206,like_count#207,dislike_count#208,date#209,comment_count#210,comments#211] json
```

연산 결과가 재사용하지 않아 post_uuid와 sentence_uuid가 같은 값을 갖는 문제는 없어짐.

“매번 다시 계산”

Appendix: Spark UUID 생성기 (3)

Spark SQL uuid()

```
post_df = df.withColumn("post_uuid", expr("uuid()"))
```

```
== Optimized Logical Plan ==
InsertIntoHadoopFsRelationCommand s3://the-all-new-bucket/uuid_test/sentence_data, false, Parquet, [path=s3://the-all-new-bucket/uuid_test/sentence_data], Overwrite, [post_uuid, comment_uuid, sentence_uuid, type, sentence]
+- WriteFiles
  +- Repartition 1, false
    +- Union false, false
      :- Project [post_uuid#24, null AS comment_uuid#221, sentence_uuid#155, post AS type#156, sentence#157]
      :  +- Filter (isNotNull(sentence#157) AND (length(sentence#157) > 10))
      :  +- Project [post_uuid#24, uuid(Some(4738327968503810595)) AS sentence_uuid#155, title#3 AS sentence#157]
      :  +- Project [title#3, uuid(Some(-1788595205522200821)) AS post_uuid#24]
      :  +- Relation [car_name#0,source#1,id#2,title#3,nickname#4,article#5,view_count#6,like_count#7,dislike_count#8,date#9,comment_count#10,comments#11] json
      :- Project [post_uuid#24, null AS comment_uuid#222, sentence_uuid#164, post AS type#188, sentence#167]
      :  +- Filter (length(sentence#167) > 10)
      :  +- Project [post_uuid#24, uuid(Some(870591813588882640)) AS sentence_uuid#164, sentence#167]
      :  +- Generate explode(split(article#178,
      , -1)), [1], false, [sentence#167]
      :    +- Project [post_uuid#24, article#178]
      :      +- Project [article#178, uuid(Some(-1788595205522200821)) AS post_uuid#24]
      :        +- Relation [car_name#173,source#174,id#175,title#176,nickname#177,article#178,view_count#179,like_count#180,dislike_count#181,date#182,comment_count#183,comments#184] json
      +- Project [post_uuid#24, comment_uuid#55, sentence_uuid#196, comment_AS type#219, sentence#198]
      +- Filter (isNotNull(sentence#198) AND (length(sentence#198) > 10))
      +- Project [post_uuid#24, comment_uuid#55, uuid(Some(-1384931691887032467)) AS sentence_uuid#198, content#58 AS sentence#198]
      +- Project [post_uuid#24, uuid(Some(2353123263973260751)) AS comment_uuid#55, comment#40 AS content#58]
      +- Generate explode(_extract_comment_content#235, [0], false, [comment#40])
      +- Project [comments#215.comment_content AS _extract_comment_content#235, post_uuid#24]
      +- Filter ((isNotNull(comment_count#214) AND (comment_count#214 > 0)) AND ((size(comments#215.comment_content, true) > 0) AND isNotNull(comments#215.comment_content)))
      +- Project [comment_count#214, comments#211, uuid(Some(-1788595205522200821)) AS post_uuid#24]
      +- Relation [car_name#204,source#205,id#206,title#207,nickname#208,article#209,view_count#210,like_count#211,dislike_count#212,date#213,comment_count#214,comments#215] json
```

RDD가 다시 계산되어도 OK.
내부적으로 시드값을 부여하여 재시도
내에서는 deterministic하게!

Appendix: 개발 환경 설정

개발 가이드 문서

Airflow 개발 가이드

Airflow 실행

```
docker compose up
```

컨테이너 띄운 후 <http://localhost:8080>로 접속해주세요.

Airflow 로그인

루트 계정 아이디는 `admin`, 비밀번호는 `/opt/airflow/standalone_admin_password.txt` 파일을 확인해주세요. airflow 계정을 추가하고 싶다면 docker container 내부에서 아래 명령어를 실행해주세요.

```
airflow users create -u $USER -p $PASSWORD -f $FIRST_NAME -l $LAST_NAME -r User -e $EMAIL
```

AWS Connection 추가

Airflow에서 AWS 관련 operator를 사용하기 위해서는 `@aws_default`라는 AWS Connection을 추가해야 합니다.

```
airflow connections add aws_default \
--conn-type aws \
--conn-login $AWS_ACCESS_KEY_ID \
--conn-password $AWS_SECRET_ACCESS_KEY \
--conn-extra '{"region_name": "ap-northeast-2"}'
```

등록된 Connection을 확인하고 싶다면 아래 명령어를 실행해주세요.

```
airflow connections get aws_default
```

AWS 배포 스크립트 관리

AWS Lambda 함수 가이드라인

배포 스크립트

스크립트는 기존 랄다 함수 업데이트만 가능합니다. 새로운 랄다 함수를 생성하시려면 AWS 콘솔을 이용해주세요.

배포 전에 로컬에서 `lambda_function.py`를 충분히 테스트해주세요.

```
cd lambda_functions
./deploy.sh <function_name>
```

배포된 함수 호출

```
cd lambda_functions
./invoke.sh <function_name> <input_file_path> <output_file_path>
```

예시

```
cd lambda_functions
./invoke.sh lambda_example input.json out.json
```

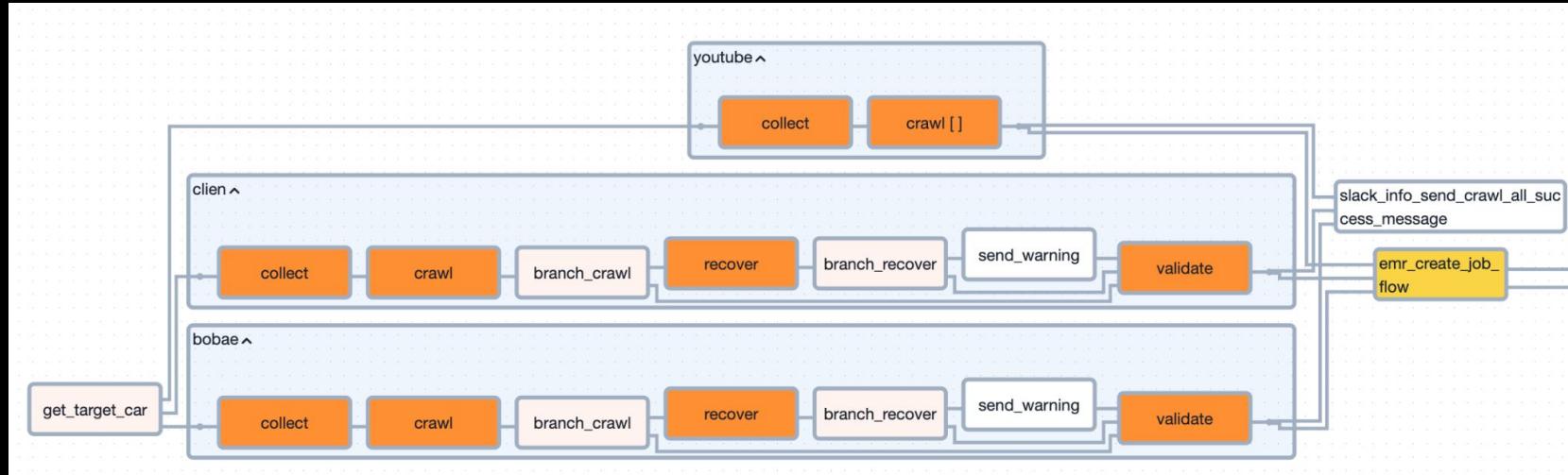
Appendix: 시각화를 위한 View 테이블

소셜 리스닝 뷰		mart.view_social_listening	
게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NULL
타입	type	VARCHAR(255)	NOT NULL
문장	sentence	VARCHAR(65535)	NOT NULL
카테고리	category	VARCHAR(255)	NOT NULL
키워드	keyword	VARCHAR(255)	NOT NULL
감성 점수	sentiment_score	REAL	NOT NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
이전 모델 명	pre_car_name	VARCHAR(255)	NOT NULL
출시일	release_date	TIMESTAMP	NOT NULL
이전 모델 출시일	pre_release_date	TIMESTAMP	NULL
웹 데이터 소스 명	source	VARCHAR(255)	NOT NULL
연령 분포	age	VARCHAR(255)	NOT NULL

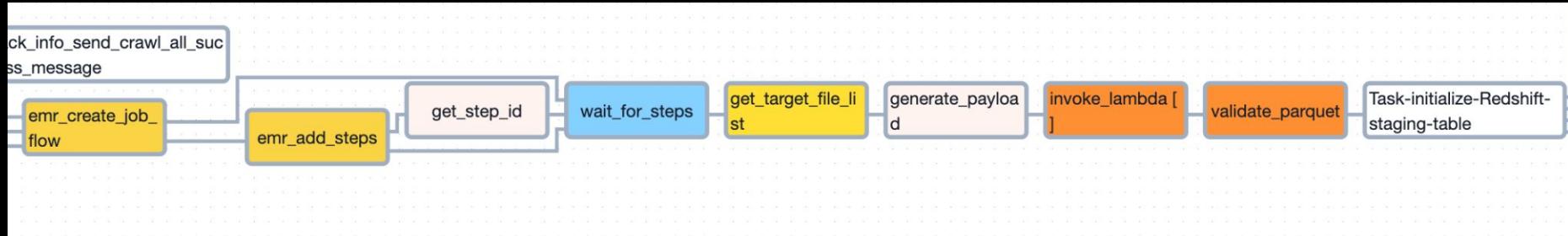
게시물 지표 뷰		mart.view_posts_comments_metric	
게시물 ID	post_id	VARCHAR(255)	NOT NULL
댓글 ID	comment_id	VARCHAR(255)	NULL
좋아요 수	like_cnt	BIGINT	NULL
싫어요 수	dislike_cnt	BIGINT	NULL
조회수	view_cnt	BIGINT	NULL
댓글 수	comment_cnt	BIGINT	NULL
수집일	ingestion_date	TIMESTAMP	NOT NULL
차 모델 명	car_name	VARCHAR(255)	NOT NULL
이전 모델 명	pre_car_name	VARCHAR(255)	NOT NULL
출시일	release_date	TIMESTAMP	NOT NULL
웹 데이터 소스 명	source	VARCHAR(255)	NOT NULL
연령 분포	age	VARCHAR(255)	NOT NULL

The Tableau logo consists of a stylized arrangement of orange and blue plus signs (+) and minus signs (-) forming a grid-like pattern. Below this graphic, the word "tableau" is written in a lowercase, sans-serif font, with each letter separated by a vertical bar.

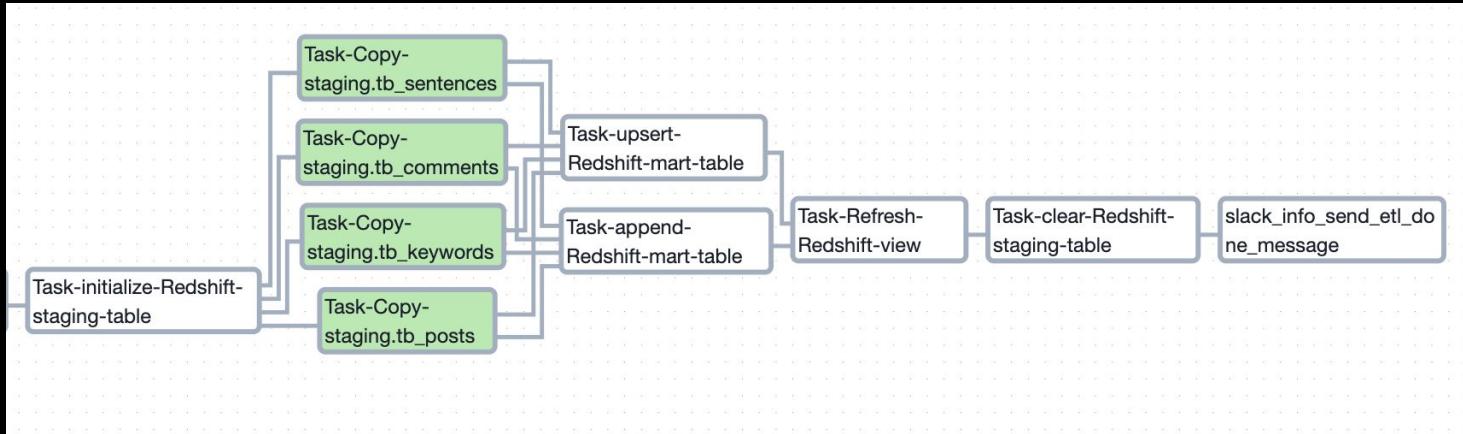
Appendix: Airflow DAG (1)



Appendix: Airflow DAG (2)

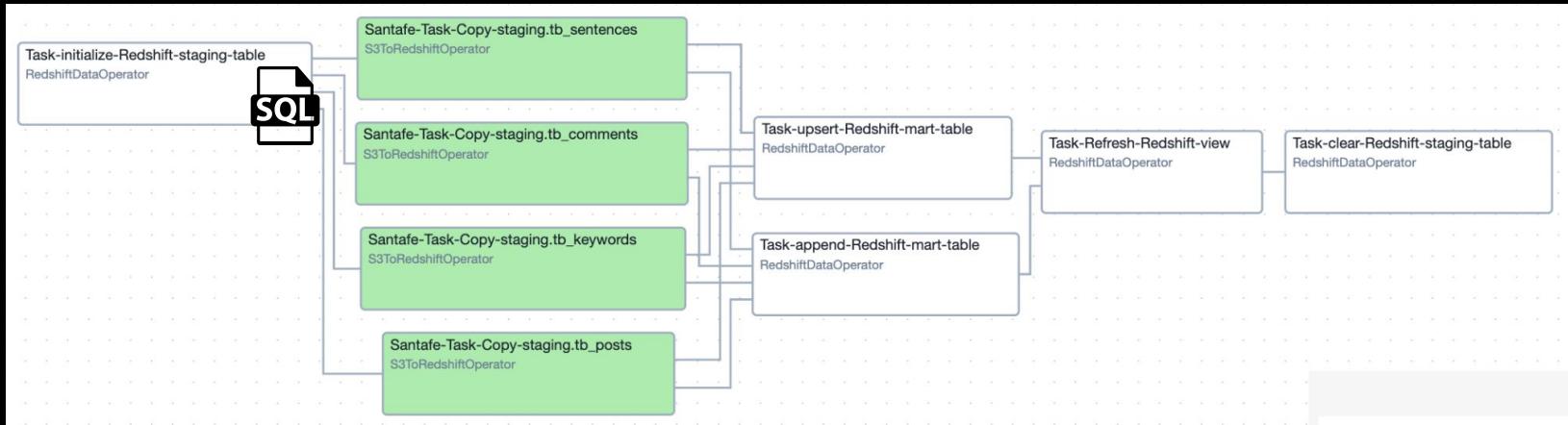


Appendix: Airflow DAG (3)



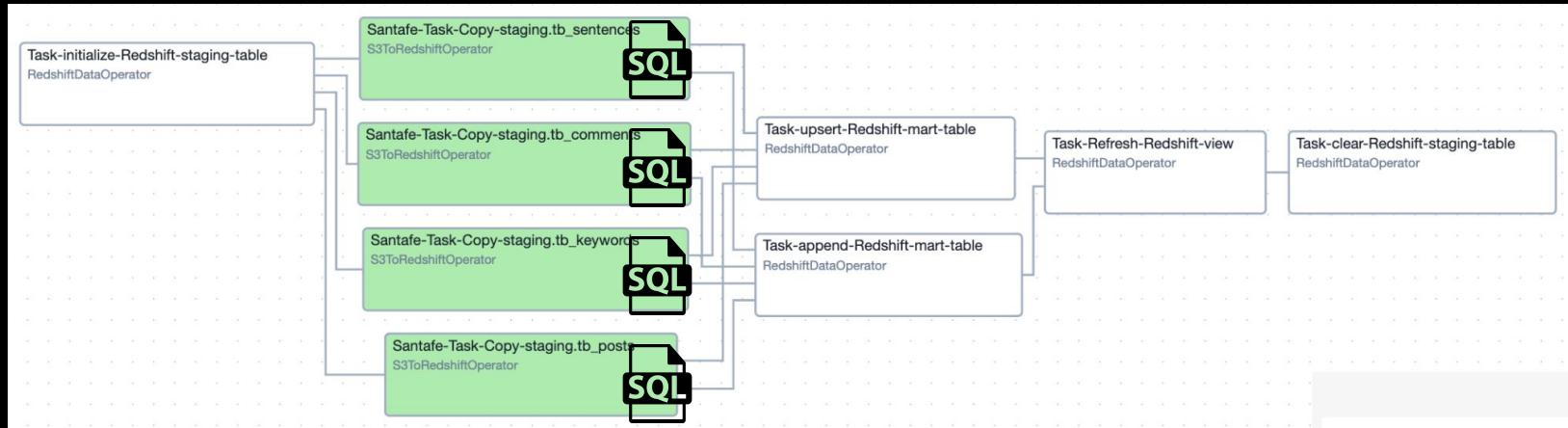
ELT DAG

CREATE



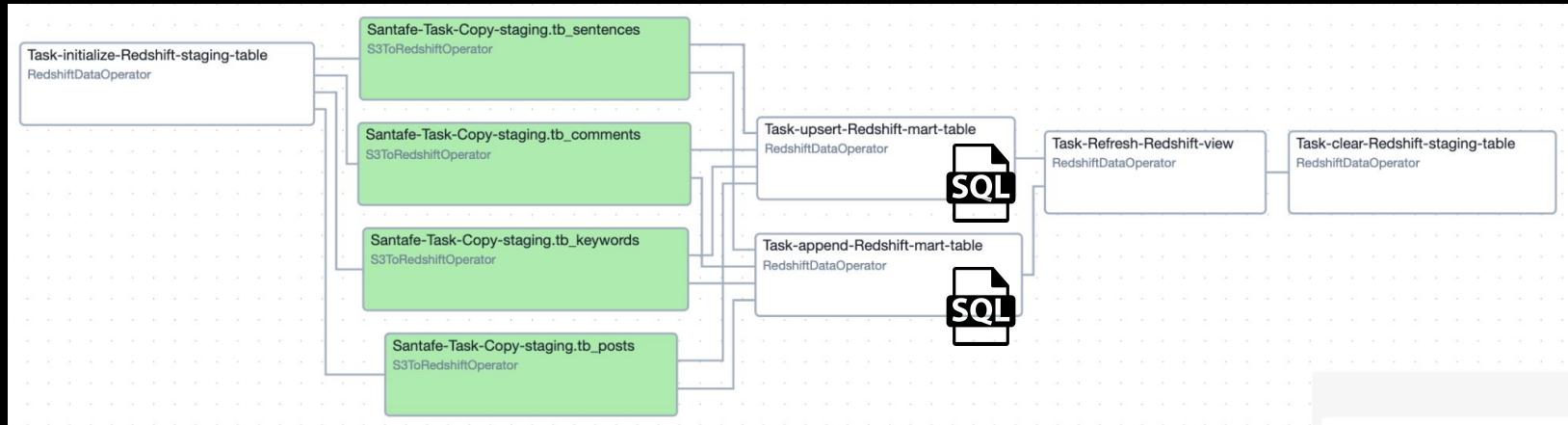
ELT DAG

COPY



ELT DAG

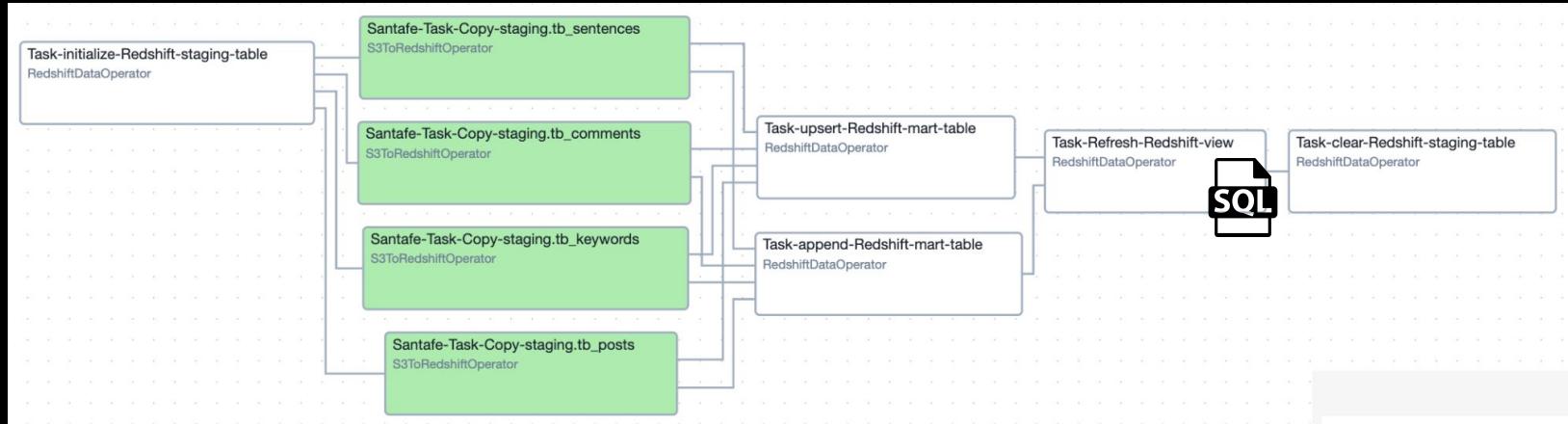
MERGE Load



INSERT Load

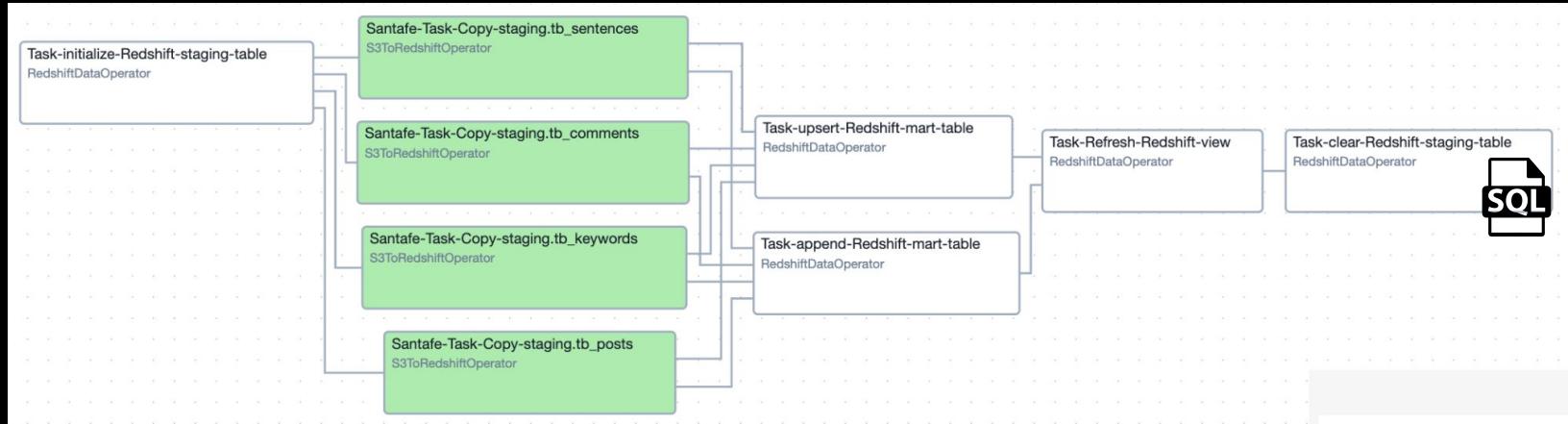
ELT DAG

REFRESH VIEW

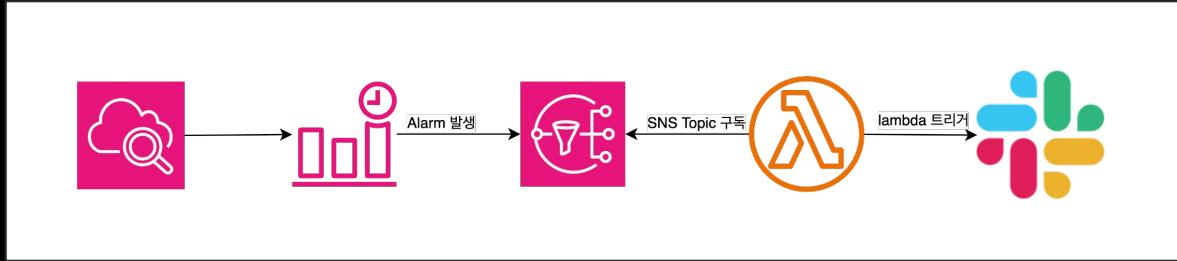


ELT DAG

DROP



Appendix: AWS 리소스 지표 모니터링



Appendix: 협업 과정

<PR 템플릿>

제목 (변경 사항을 한 줄로 간략하게 작성)

개요

- 이번 PR의 목적과 변경 내용을 간단히 설명합니다.
- 예: 로그인 기능 개선, 버그 수정 등

관련 이슈

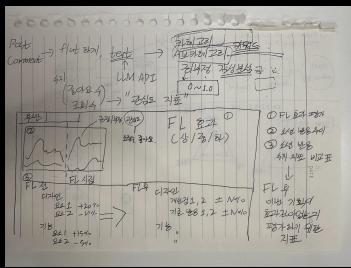
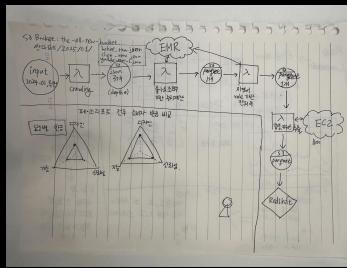
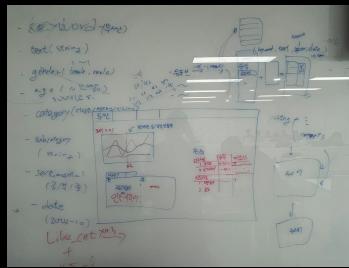
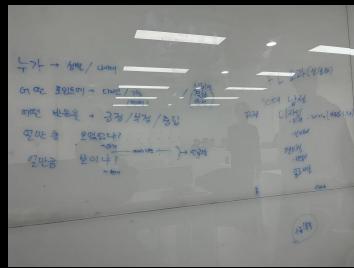
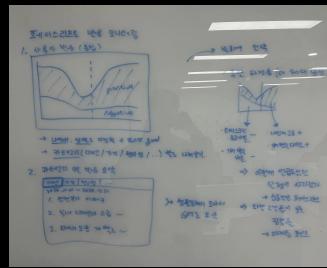
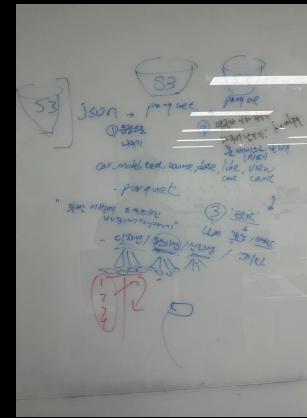
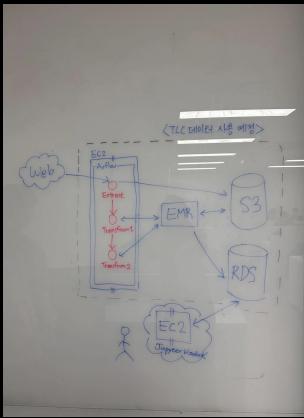
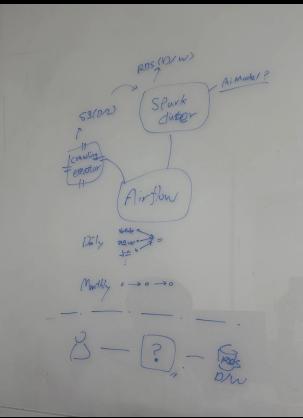
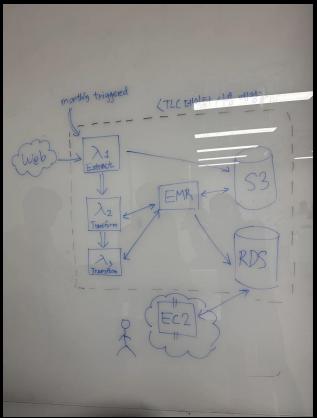
- 연결된 이슈가 있다면 번호를 기재합니다.
- 예: `Resolves #123`

변경 사항

- 주요 변경된 파일이나 기능에 대해 간략하게 목록으로 작성합니다.
- 파일 A: 기능 개선
- 파일 B: 버그 수정

```
* ae7de15 Merge pull request #26 from softteer5th/fix/transform_on_emr
| \
| * e0fb5f7 fix: emr에서 실행할 text_processing.py 코드 생성(post_data, comment_data, sentence_data 블록 생성)
| * c0359c4 fix: 텍스트 전처리(flatten, cleaning)한 데이터 parquet 블록 5개로 나누어서 저장
| * 85f16af Merge pull request #25 from softteer5th/fix/classify-sentence
| \
| * efdb034 (origin/fix/classify-sentence) feat: Use parquet instead of json
| * 5d4e4d2 fix: Use input file path from event
| * 5d8aa33 fix: Format
| * 69aa208 Merge pull request #24 from softteer5th/fix/raw-data-schema
| \
| * a359f72 (origin/fix/raw-data-schema) fix: 데이터 수집 단계 JSON 스키마 변경
| * 4a5f983 Merge pull request #23 from softteer5th/feature/#15-load-to-redshift
| \
| * cc4f090 (origin/feature/#15-load-to-redshift, feature/#15-load-to-redshift) Merge branch 'main' of https://github.com/softteer
| \
| * 026f5ab Merge pull request #22 from softteer5th/fix/transform_on_emr
| \
| * b26ab9d docs: readme 수정
| * 6ca495b feat: 생성된 풀리스터에서 submit-job하는 스크립트
| * 7df65a0 feat: emr에서 수행할 spark job(nlp 처리 및 weight 계산) 로컬 실행 코드
| \
| * 1fc28e docs: Airflow Redshift Connection 개발 규칙 추가
| * 888723d feat: S3ToRedshiftOperator Task 구현
| * 58297f4 docs: Update Dockerfile to Create Redshift Connection
| \
| * f9bf36f (origin/fix/classify-text-with-nlp) feat: Use input file path from event
| * 9a1c2dd fix: Format
| \
| * c7712af Merge pull request #21 from softteer5th/feature/airflow
| \
| * 600525b (origin/feature/airflow) feat: Slack notification test
| * 78cc437 fix: Stringify payload for lambda operator
| * b82758c Merge pull request #20 from softteer5th/feat/lambda_transform
| \
| * 15f8d15 (origin/feat/lambda_transform) feat: llm 수행하는 lambda 함수 구현
| * fab138d (feature/airflow) Merge branch 'feature/airflow' of https://github.com/softteer5th/DE-team3 into feature/airflow
| \
| * dec70d8 feat: EMR trigger test
| * 1f99908 tmp
| \
| * 14b5d18 Merge pull request #18 from softteer5th/feature/lambda-extract-functions
| \
| * b99a6ab (origin/feature/lambda-extract-functions) feat: Add function crawl_youtube
| * e269236 fix: Increase lambda invoke timeout
| * 5a2a20a fix: Add function crawl_allarm_arm
| * c1d1402 feat: Add function crawl_clien
| * 99fae8d feat: Add function crawl_bobae
| * 7f27021 feat: Add function collect_target_clien
| * 9a713b6 feat: Add function collect_target_bobae
```

Appendix: 협업 과정



Appendix: Use Case

