

---

# 11

---

대용량의 상품을 빠르게 검색/색인 할 수 있다고?  
(feat. Apache Iceberg)

Part 1. 검색 색인 파이프라인 고도화

---

검색플랫폼개발팀 김두진

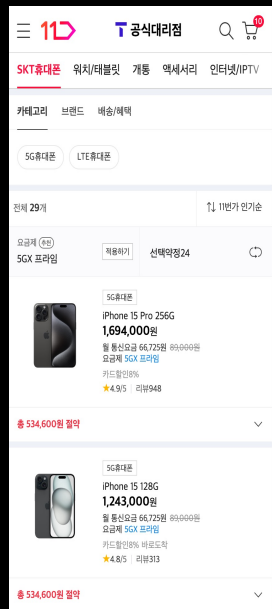
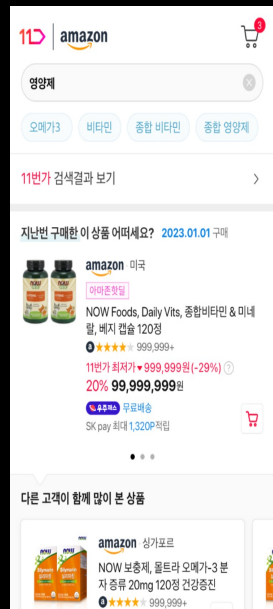
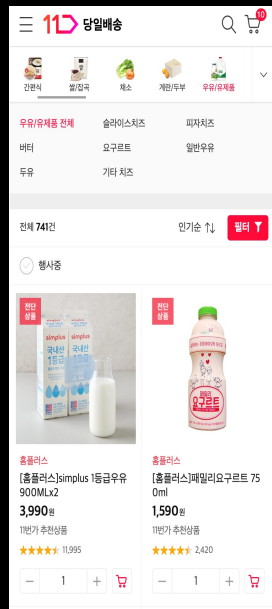
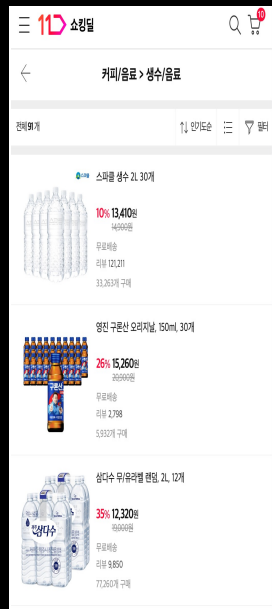
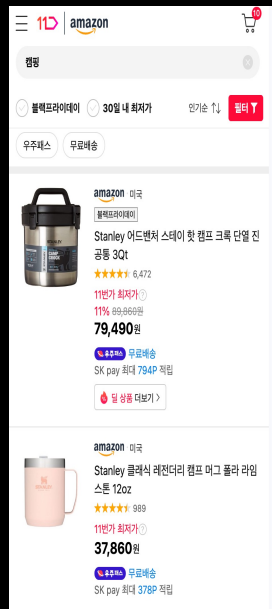
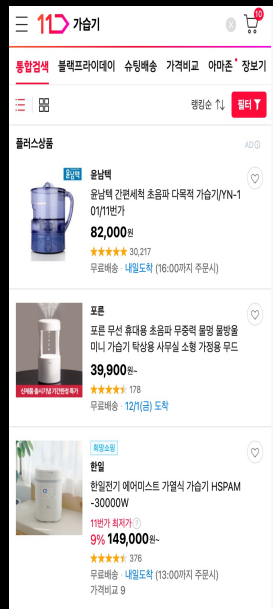
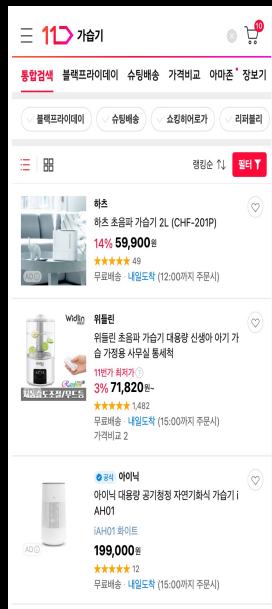
# CONTENTS

1. 팀 소개
2. 검색 색인 파이프라인 고도화 이야기
  1. 무엇을?
  2. 왜?
  3. 어떻게?
  4. 그래서 뭐가 좋아지는거지?

# 팀 소개

## 11번가의 검색서비스에 필요한 검색엔진 및 플랫폼 개발 및 운영

- OM, 광고, 아마존, 쇼킹딜, 마트, Live방송, 반복구매, T공식대리점, 리뷰, 주소
- 검색관리도구, Cache(CCS, rCache), KV, Search API, Search Data Batch



# 색인 파이프라인 고도화 - 뭘하는거지?

11 TechTalk 2023

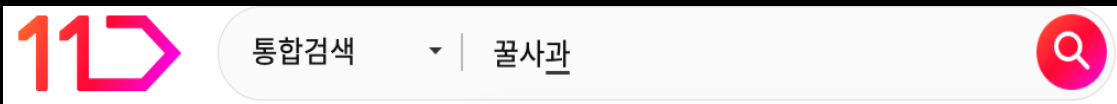


V 8.11.2

Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene™.

# 색인 파이프라인 고도화 - 뭘하는거지?

역색인  
Inverted-Index



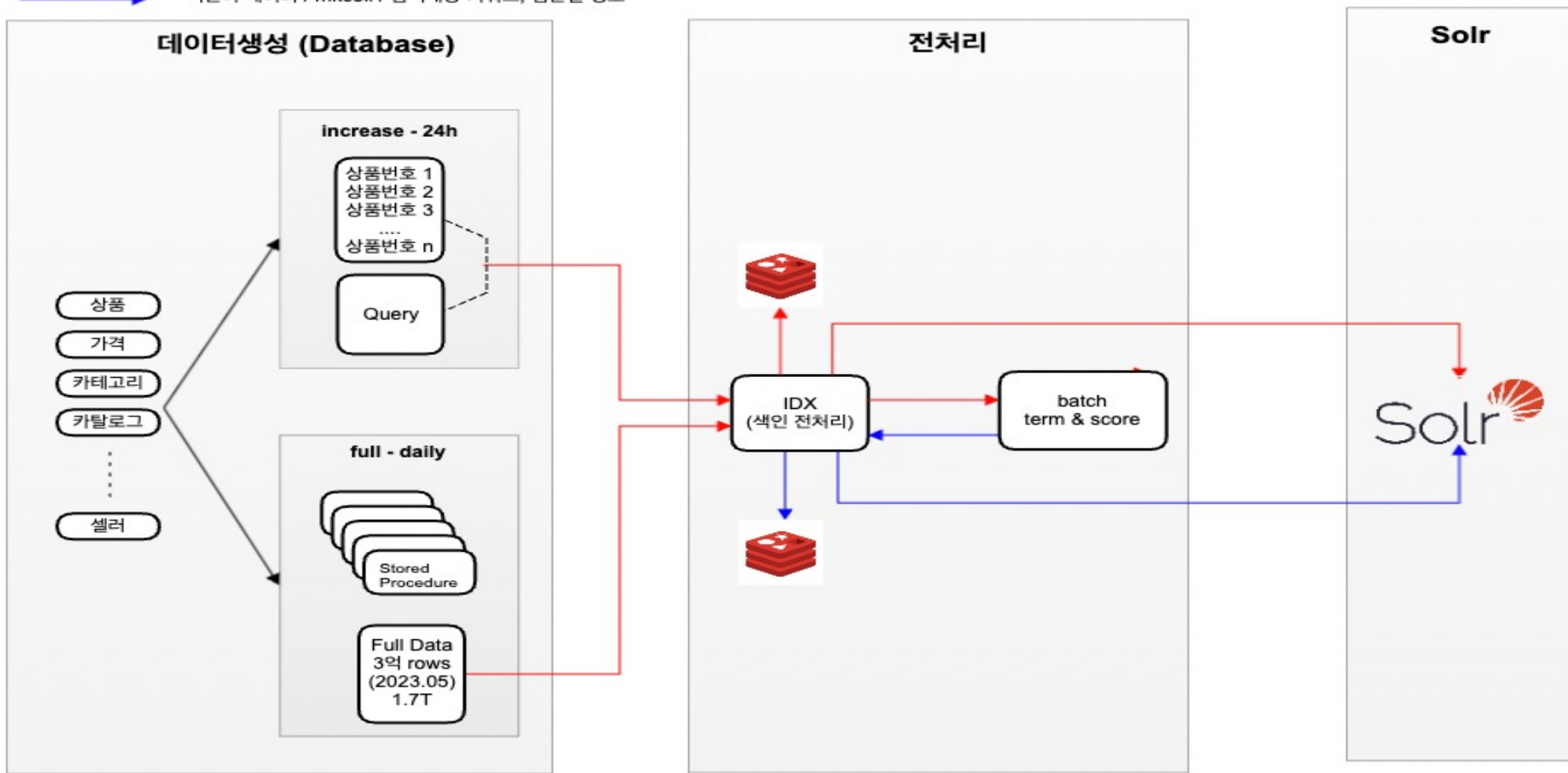
Term	Document
꿀	1, 2, 4, 8
사과	1, 3, 4, 6, 9

# 색인 파이프라인 고도화 - 뭘하는거지?



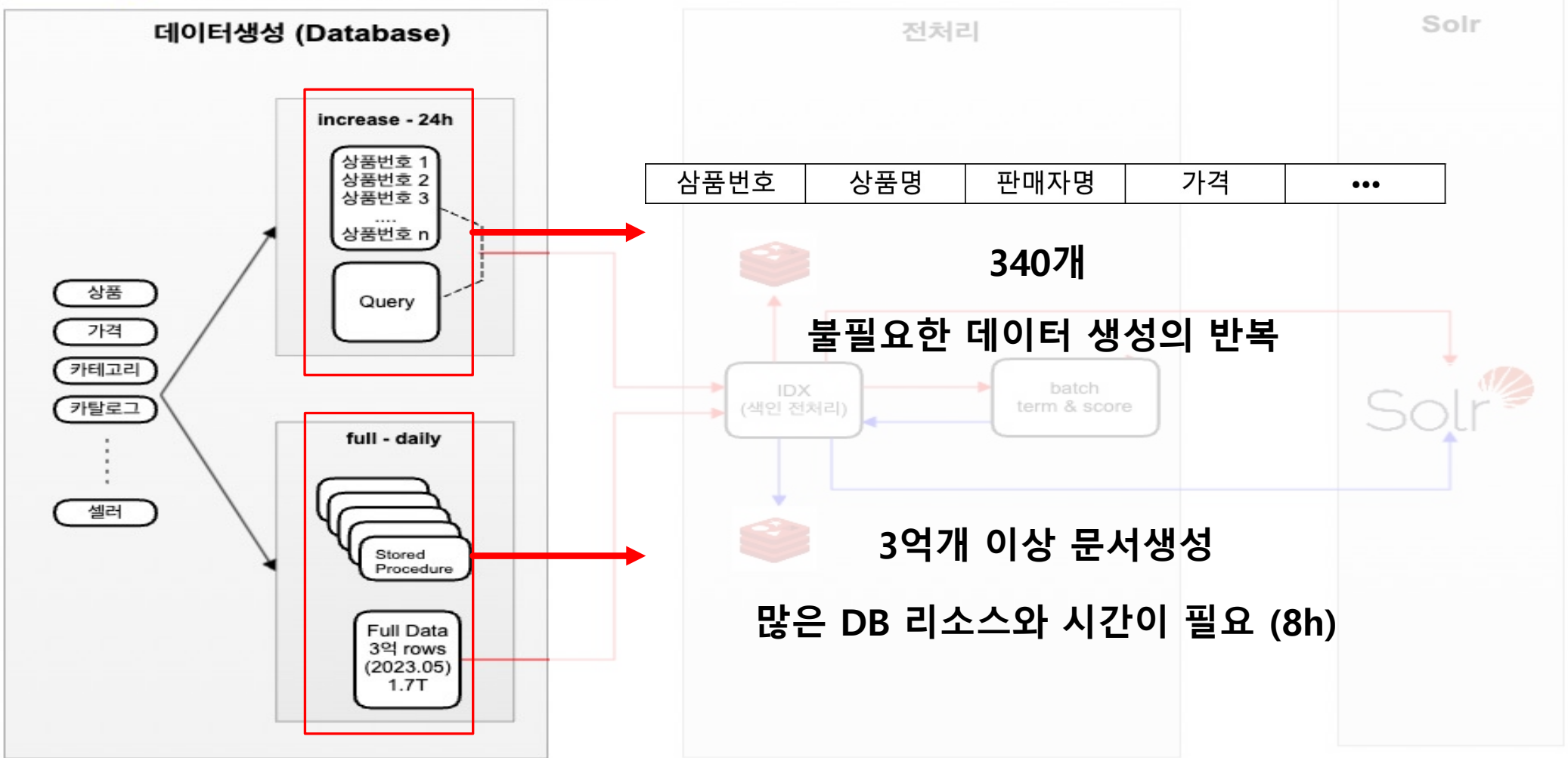
# 색인 파이프라인 고도화 - 뭘하는거지?

- 상품 데이터 / DB / 상품명, 카테고리명, 셀러명 등등
- 색인어 데이터 / mvcoll / 검색대상 키워드, 톰분절 정보



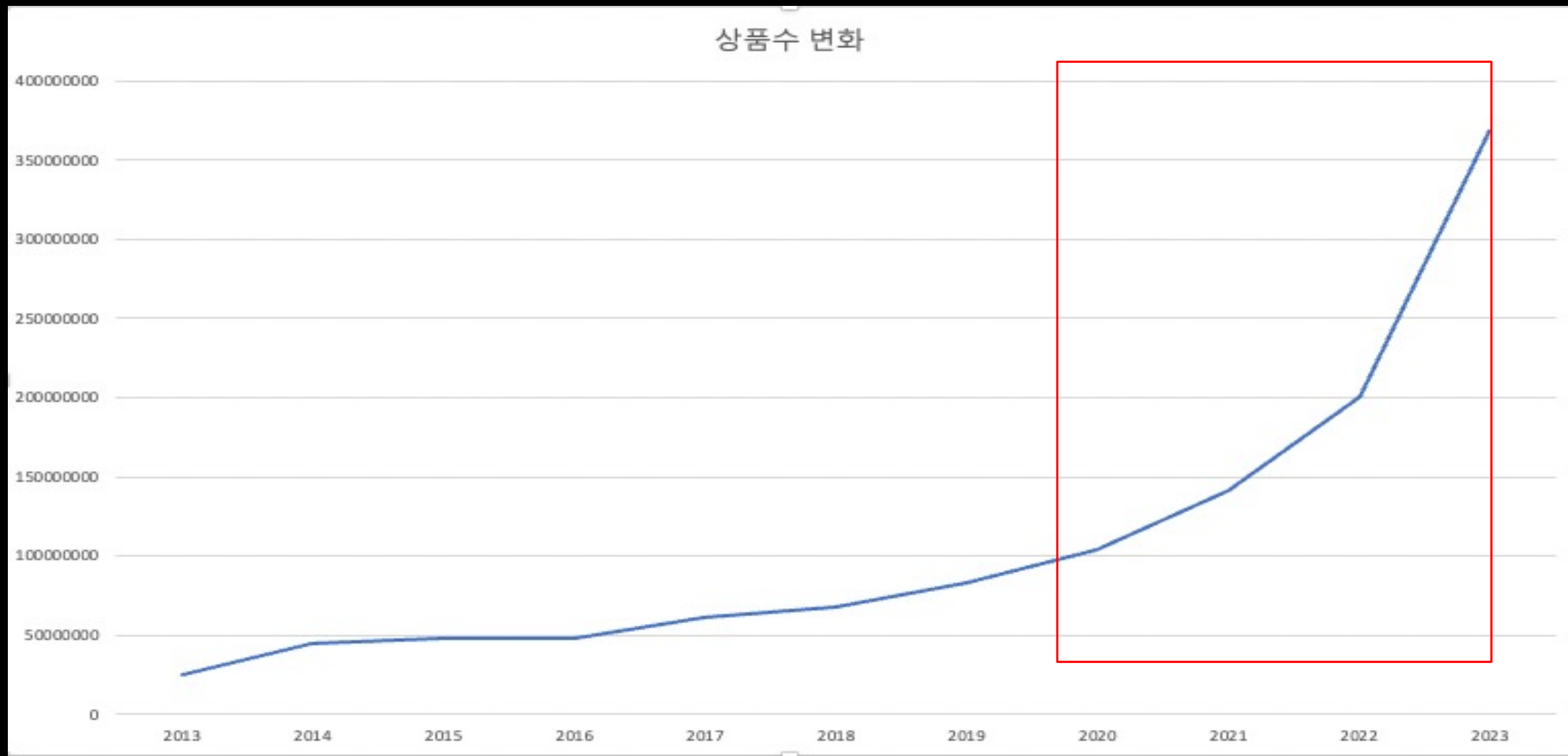
# 색인 파이프라인 고도화 - 뭘하는거지?

→ 상품 데이터 / DB / 상품명, 카테고리명, 셀러명 등등  
→ 색인어 데이터 / mckoll / 검색대상 키워드, 톰분절 정보

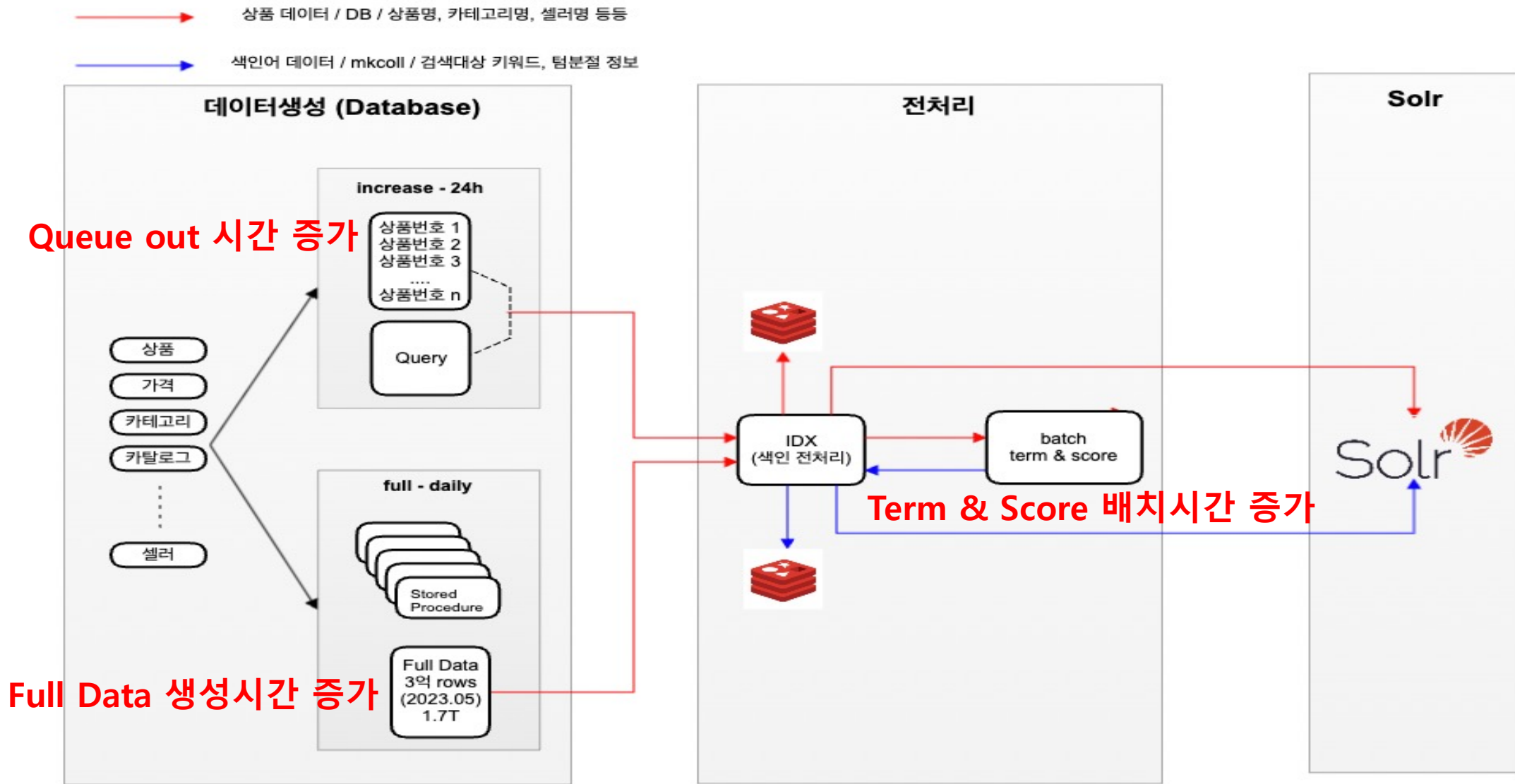




# 색인 파이프라인 고도화 - 왜?



# 색인 파이프라인 고도화 - 왜?



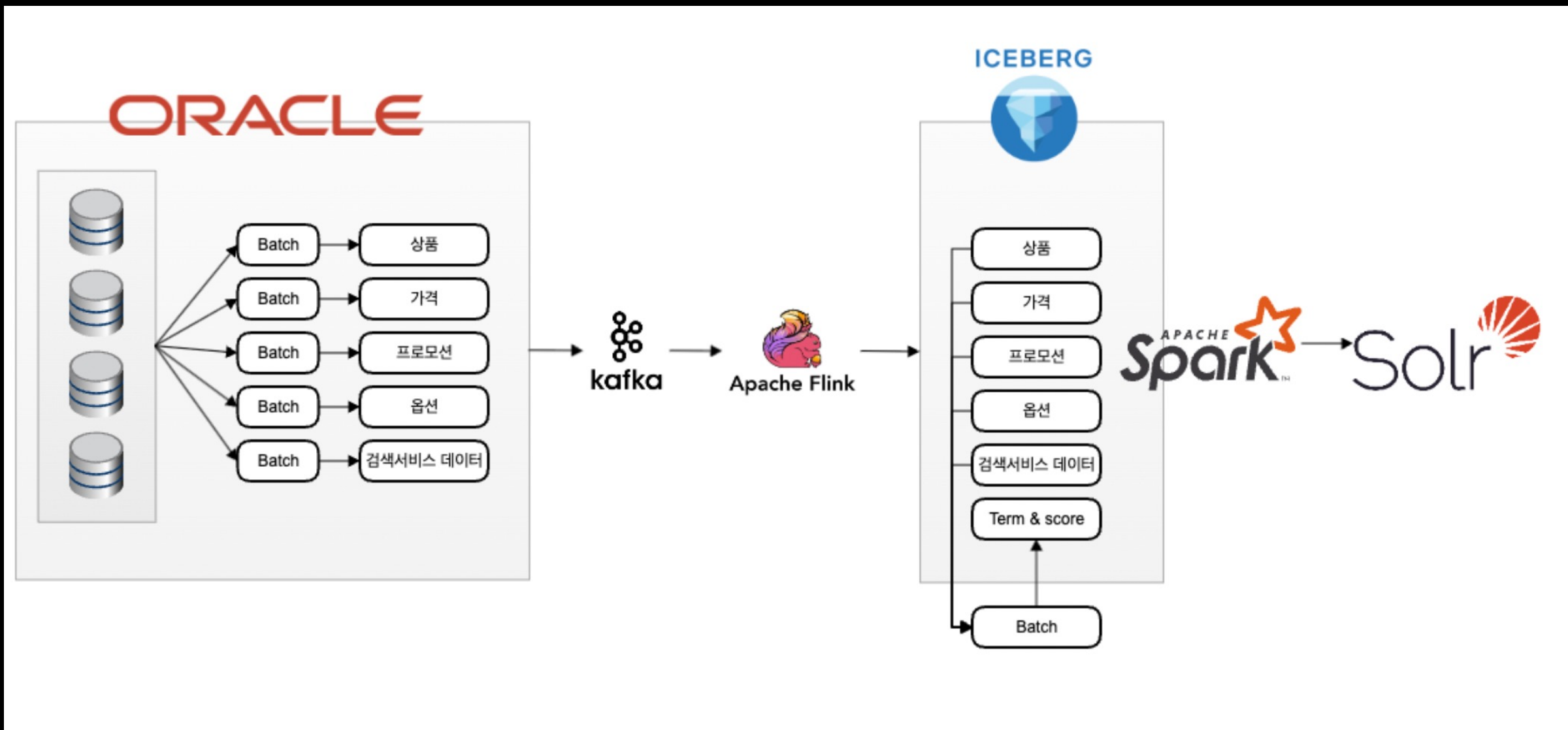
# 색인 파이프라인 고도화 - 왜?



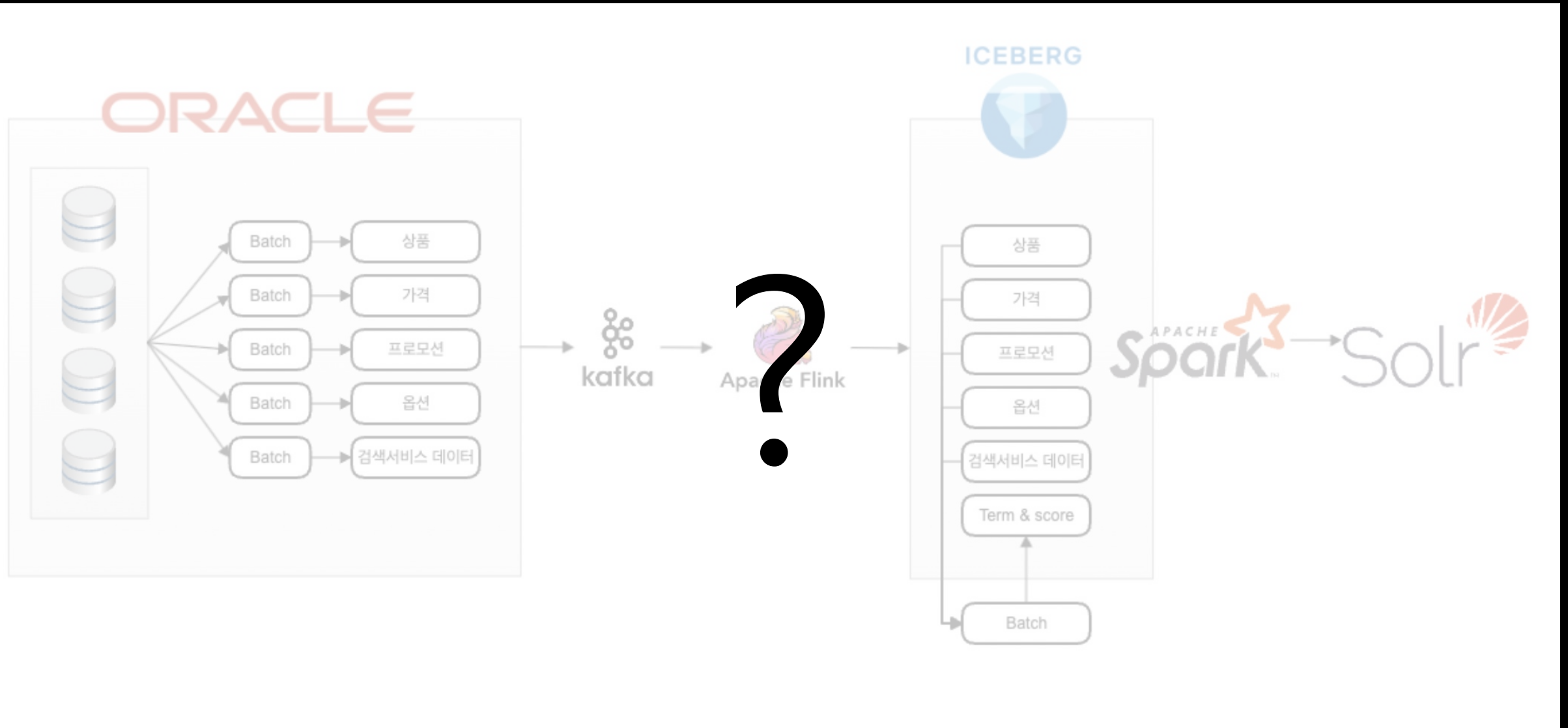
- 검색 데이터 구성 방식변경
- 많은 리소스와 시간이 들어가는 Full Data Batch Fade out
- 분산시스템을 활용하여 대용량 데이터 처리

# 색인 파이프라인 고도화 - 어떻게?

11 TechTalk 2023

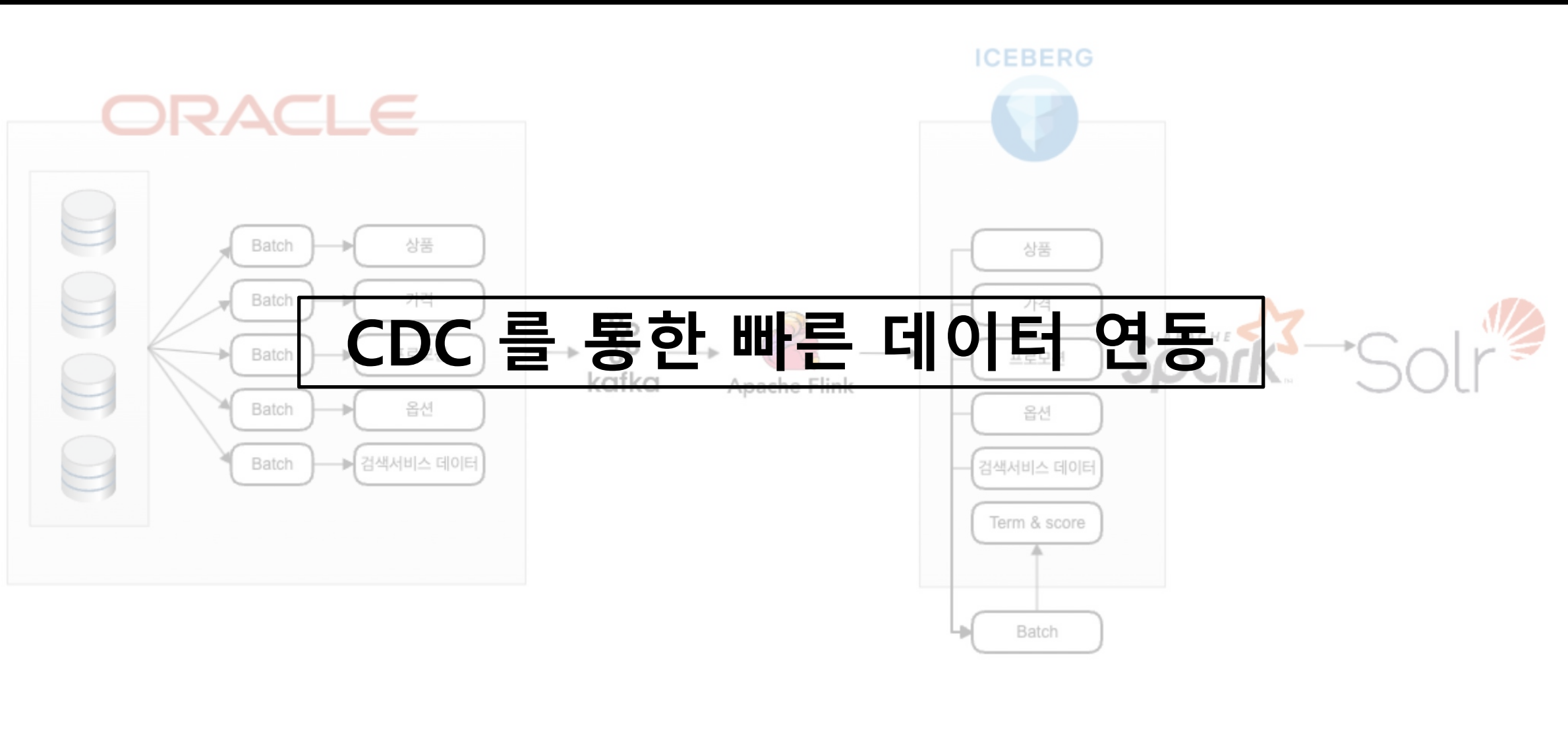


# 색인 파이프라인 고도화 - 좋아지는것?

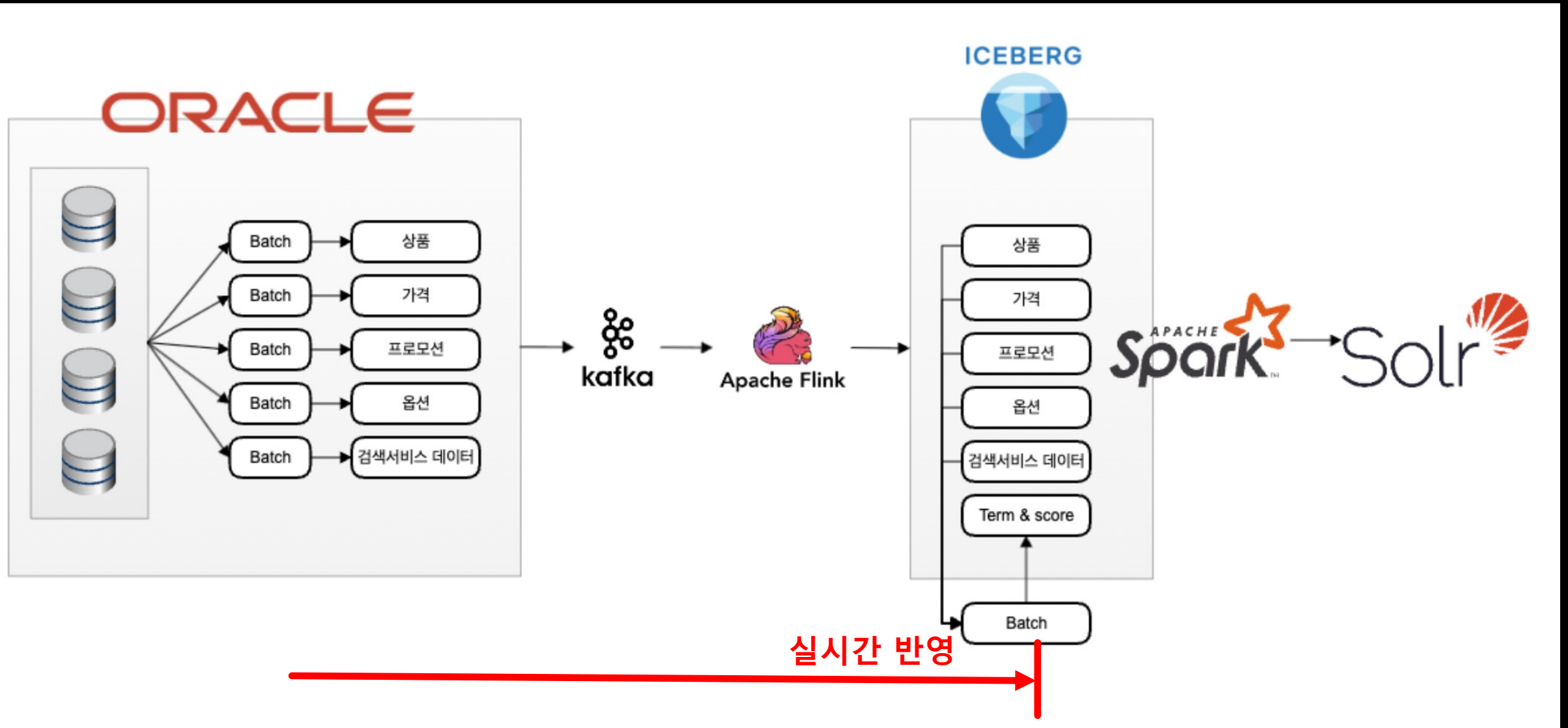


# 색인 파이프라인 고도화 - 좋아지는것?

11 TechTalk 2023

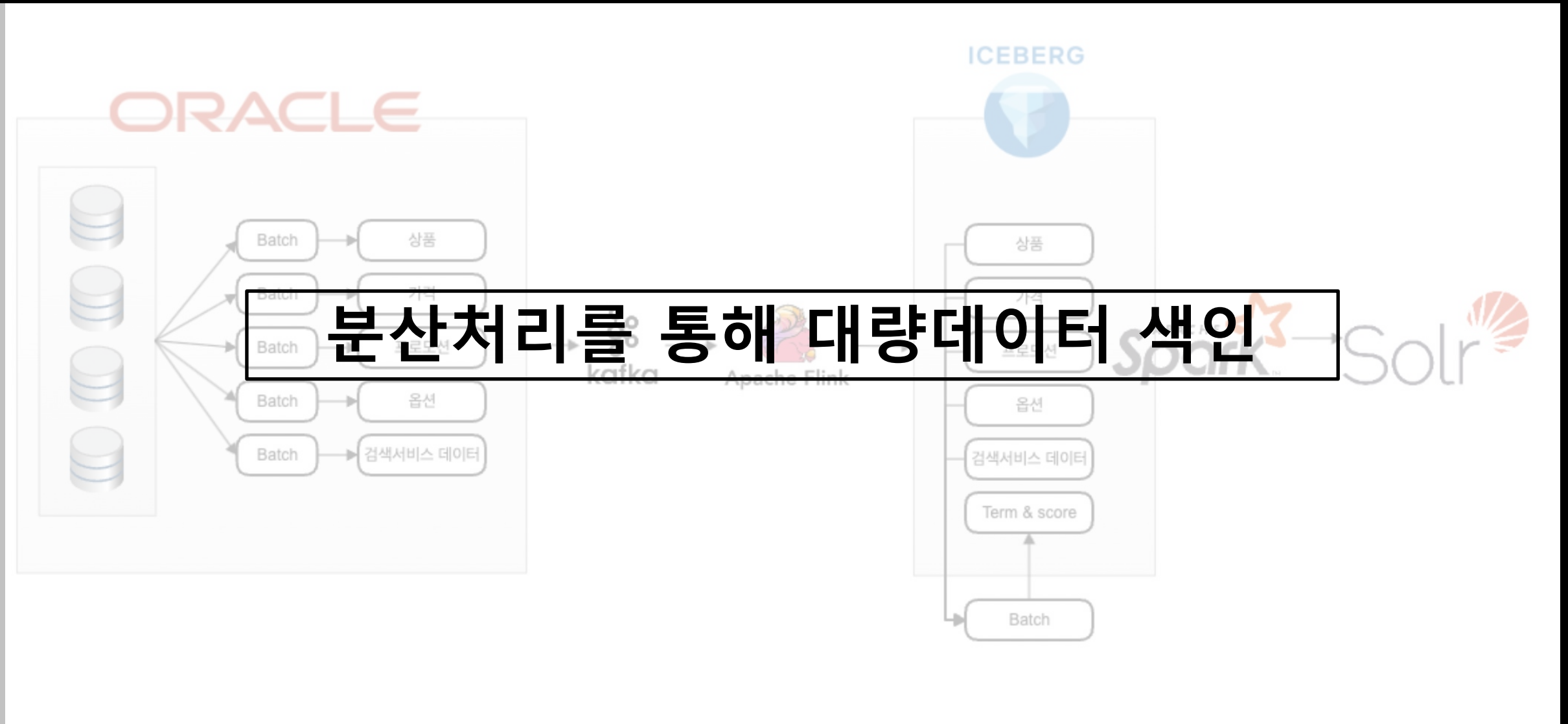


# 색인 파이프라인 고도화 - 어떻게?



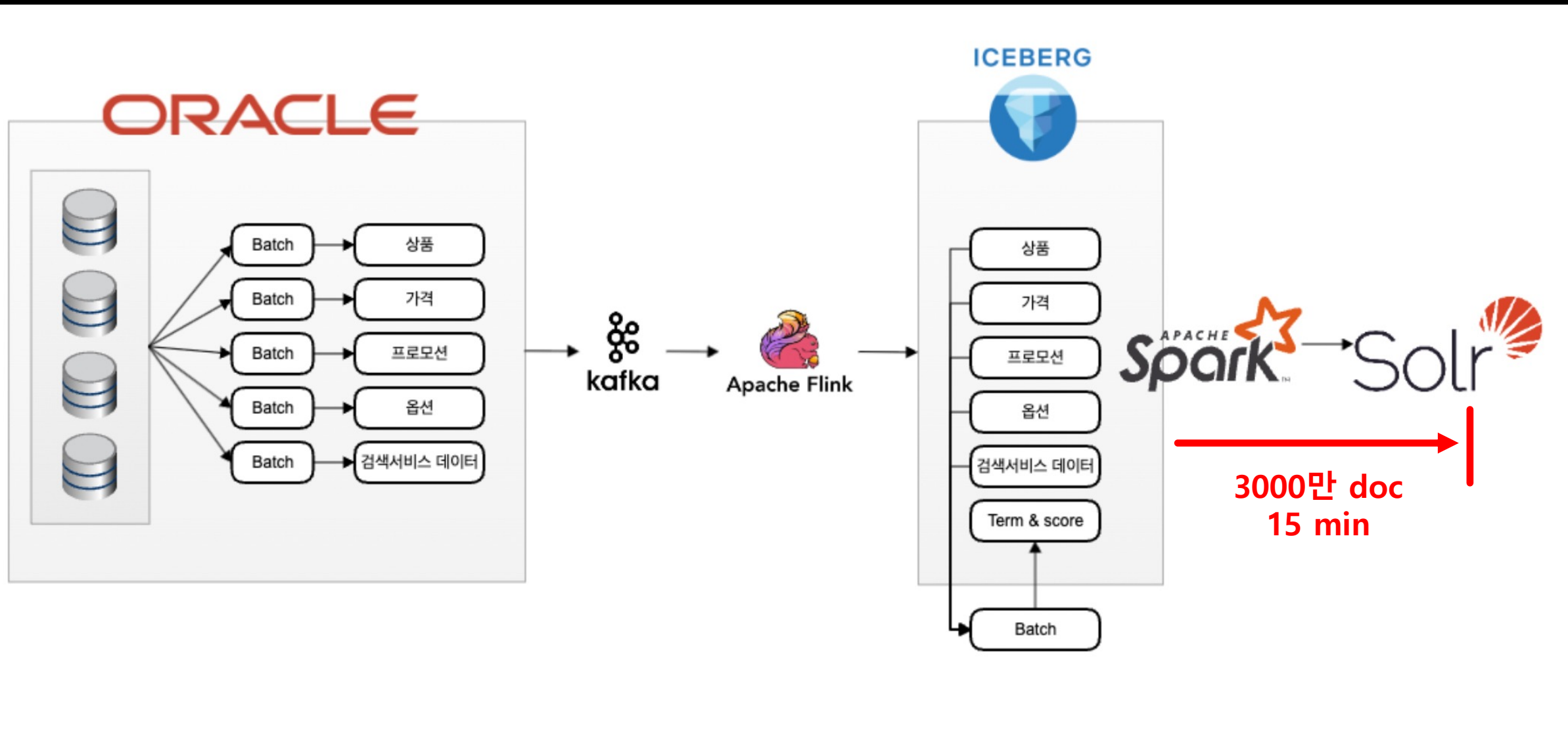


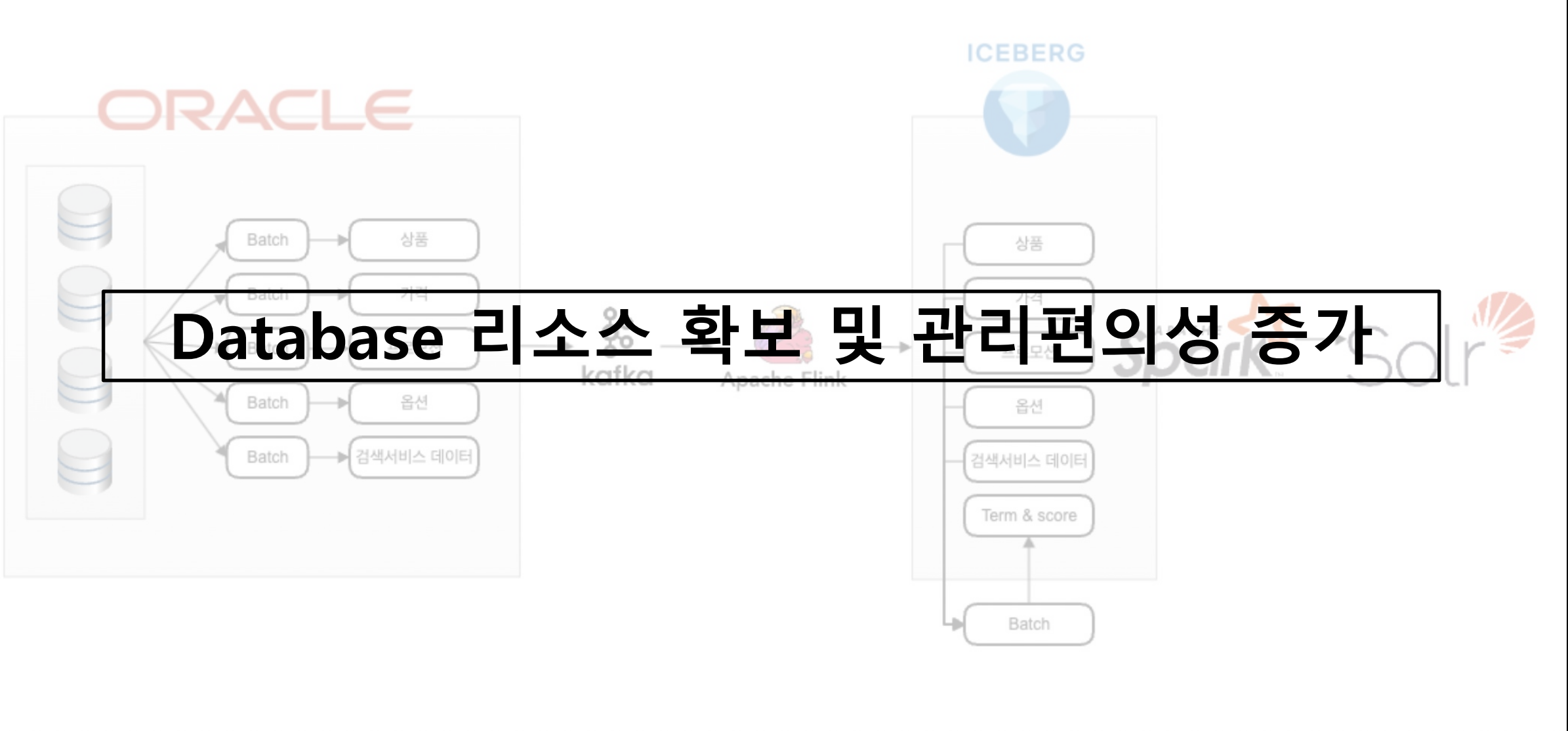
# 색인 파이프라인 고도화 - 좋아지는것?



# 색인 파이프라인 고도화 - 어떻게?

11 TechTalk 2023





# ing

미 구현 기능 개발

퍼포먼스 향상에 대한 고민

---

# 11

---

대용량의 상품을 빠르게 검색/색인 할 수 있다고?  
(feat. Apache Iceberg)

Part 2. Apache Iceberg 도입기

---

데이터플랫폼팀 김용주

# CONTENTS

1. Intro
2. 목표
3. Data Lakehouse
4. Apache Iceberg
5. 결과
6. 결론 및 기대효과

# Intro

2023년 5월 어느날..

데이터플랫폼 00님



1. 검색데이터 색인 속도를  
**획기적으로 올려야 되는**  
요청사항이 들어왔습니다.  
용주님 같이 해보시죠^^



3. Oracle에 데이터가  
입력되자마자 하둡 클러스터에  
거의 **실시간으로 갱신 되는**  
정도?^^



5. 제가 구상해 놓은게 있으니  
일단 시작해 봅시다^^

데이터플랫폼 본인



2. 획기적 이라는게..  
어느정도 인가요?  
(현재는 **FULL 색인에 1일,**  
**동적색인도 어느정도의**  
**시간이 걸리고 있음)**



4. **대규모 분산환경에**  
**실시간 데이터 갱신을**  
한다고요?  
그게 가능한가요..?



6. 냐.. ㅏ ㅏ

# 대규모 분산환경의 실시간 갱신

이게 어려운 건가요??

>> 네, 매우 어렵습니다..

## 1. 빅데이터 환경은 갱신이 안됩니다.

**WORM**(Write Once Read Many)

- 한번 쓴 데이터는 수정할 수 없다
- 갱신을 위해서는 이전 것을 지우고 새로 만든다



대표적인 분산환경 Storage

갱신(recode-level update), 삽입, 삭제 불가

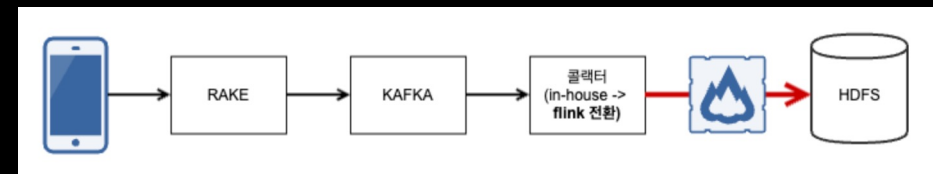


11번가의 표준 DW engine HIVE는  
insert, update, delete 명령어 차단 설정

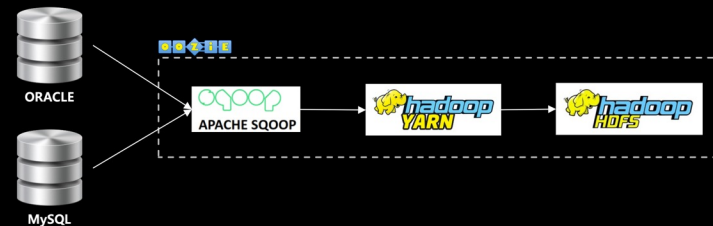
## 2. 실시간 갱신은 더욱 어려워요.

실시간 저장 되는 데이터는 Log data

- 저장(write)과 갱신(upsert)은 개념이 다름



RDBMS의 데이터를 DIC로 이관하는 방식은  
**batch** 처리 방식(hourly, daily)



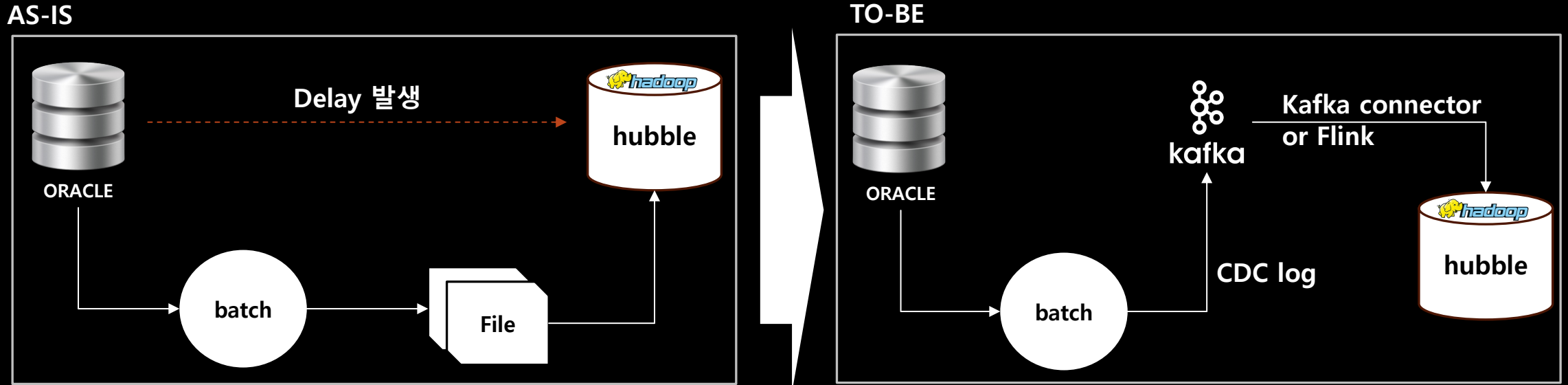


# 개선 목표

Hubble 클러스터(Hadoop)에 저장되는 검색데이터의 갱신 속도(delay) 개선

FULL 색인 : 1일  
동적 색인 : 지속 수행

Oracle에 저장 되고 거의 실시간 Hubble에 갱신

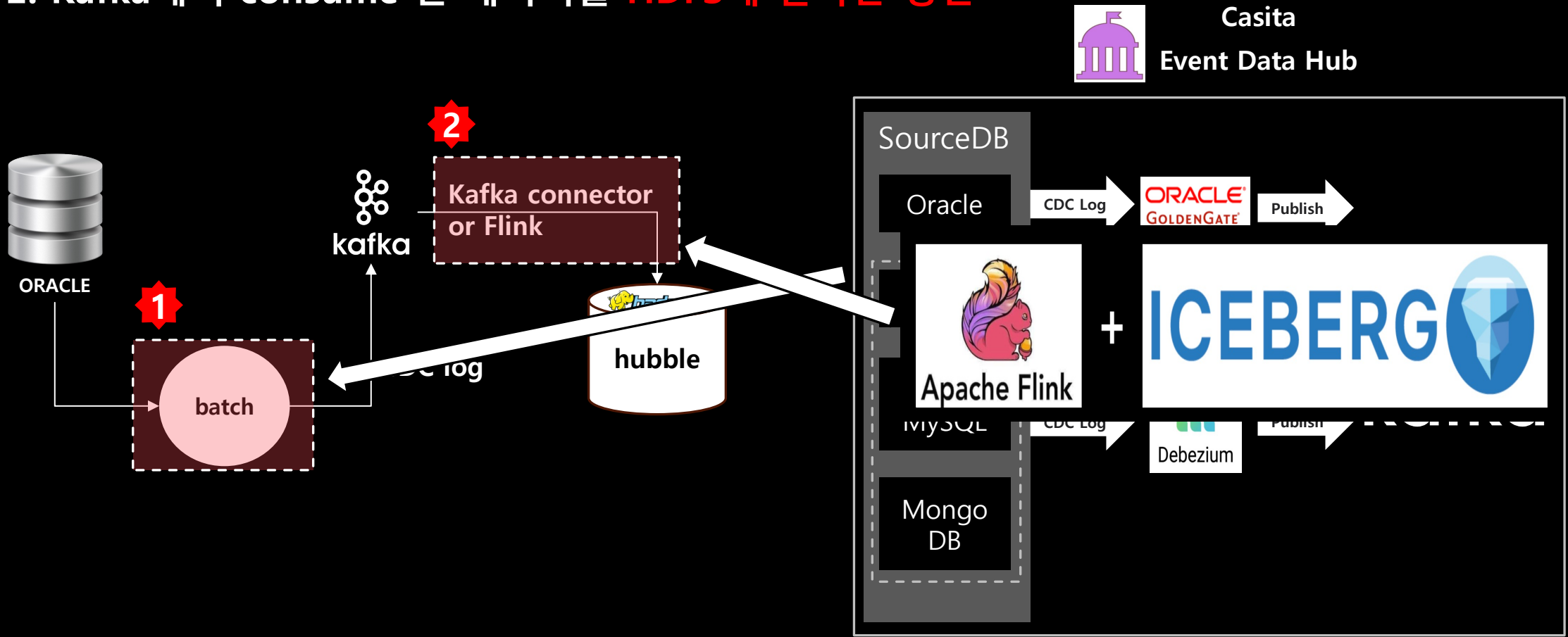


즉 **batch** 방식에서 **stream** 방식으로 바뀌어야 한다

\* CDC(Change Data Capture) : 데이터베이스 내 데이터에 대한 변경을 식별해 필요한 후속처리를 자동화 하는 기술 또는 구조

# 두개의 핵심 포인트

1. CDC Log 를 Kafka에 실시간 publish
2. Kafka에서 consume 한 데이터를 HDFS에 실시간 갱신



[https://techtalk.11stcorp.com/2022/pdf/TECH-TALK-2022\\_SESSION-04.pdf](https://techtalk.11stcorp.com/2022/pdf/TECH-TALK-2022_SESSION-04.pdf)

\* CDC(Change Data Capture) : 데이터베이스 내 데이터에 대한 변경을 식별해 필요한 후속처리를 자동화 하는 기술 또는 구조

# Data Lakehouse

11 TechTalk 2023

먼저 데이터 인프라의 변화 과정을 알아보자

Lakehouse는 Data Lake + Data Warehouse 의 장점을 결합한 패러다임  
(저비용 스토리지에 RDBMS와 유사한 데이터 관리 기능을 제공)

- ACID 보장

- 스키마 진화

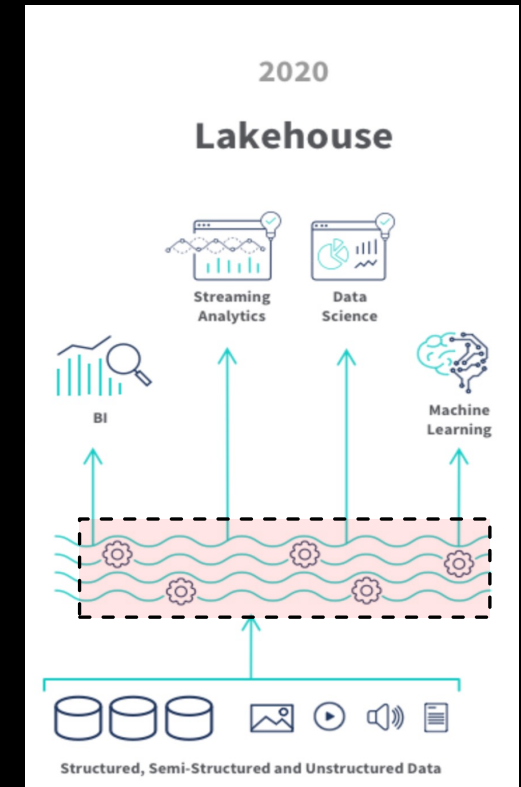
- 오픈 형식의 다양한 데이터 유형 지원

- Time Travel & Rollback 기능 지원




- \*업서트(Upsert) 및 삭제>Delete) 지원

현재 11번가 데이터플랫폼(DIC)은 생성된 데이터(테이블)에 대해서 Insert, Update, Delete를 지원하지 않는다

- 데이터 거버넌스



# Open Table Format

항목			
ACID	Yes	Yes	Yes
Partition Evolution	Yes	No	No
Schema Evolution	Yes	Partial	Limited
Time Travel	Yes	Yes	Yes
Merge	Yes	Yes	Yes
Compaction	API based	Manual	Automated
Data Format	Parquet, Avro, ORC, CSV	Parquet, ORC	Parquet
Current Pointer	Metastore, File system with version file	Timeline commit	Transaction log
Programming Language	Java & Python	Scala, Java & Python	Java & Python

<https://youtu.be/G5vvPX7PtmM?si=klOQE0fLAnDwwwKL>

## 실시간 Upsert 및 Delete 기능 & 성능을 타겟으로 검토

- Delta-lake는 커뮤니티 버전 지원 측면에서 제외
- **Hudi**는 Iceberg 보다는 소규모 테이블 + 스트리밍 처리에 특화 & 제일 먼저 검토한 솔루션 이므로 **선정**
- stream 처리시 delete 기능 미 작동으로 인해 **Iceberg로 변경**

# What is Apache Iceberg?

11 TechTalk 2023



iceberg logo

The **이오픈 테이블 포맷** for analytic datasets

Iceberg is a high-performance format for huge analytic tables.

**RDB 테이블의 안정성과 신뢰성을 빅데이터 환경에 적용할 수 있는 기능을 제공** while making it possible for engines like Spark, Trino, Flink, Presto, Hive and Impala to safely work with the same tables, at the same time.

## 표현형 SQL

데이터 병합, 업데이트, 삭제를  
수행하는 유연한 SQL 제공

## Full Schema Evolution

Schema evolution just works. Adding a column won't bring back "zombie" data. Columns can be renamed and reordered. Best of all, schema changes never require rewriting your table.

## Hidden Partitioning

Iceberg handles the tedious and error-prone task of producing partition values for rows in a table and skips unnecessary partitions and files automatically. No extra filters are needed for fast queries, and table layout can be updated as data or queries change.

## Time Travel and Rollback

Time-travel enables reproducible queries that use exactly the same table snapshot, or lets users easily examine changes. Version rollback allows users to quickly correct problems by resetting tables to a good state.

## 데이터 압축

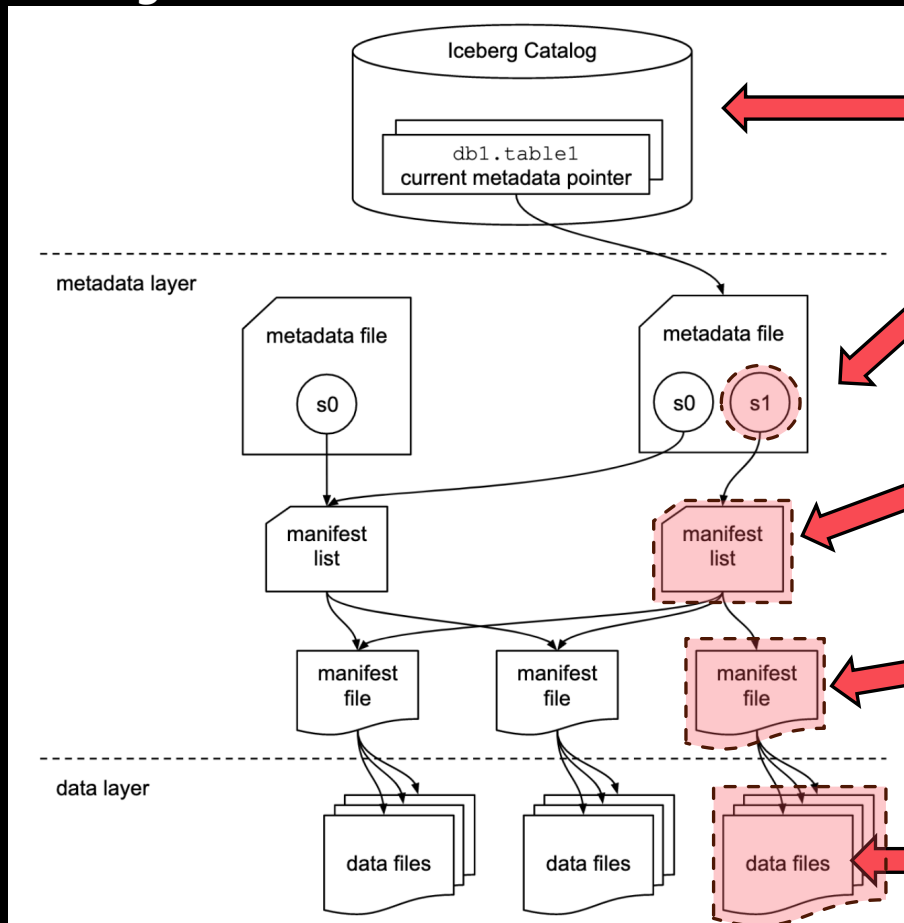
데이터 압축은 기본적으로 지원되며, 파일 크기를 최적화, 조회 성능 향상

# What is Apache Iceberg?

iceberg는 메타정보 및 데이터를 **파일로 관리**

>> Hive 테이블 형식의 **문제점을 해결한 key 포인트!!**

iceberg table architecture



```
select * from db1.table1;
```

- iceberg catalog에 접근
- 현재 metadata pointer 체크

- metadata 파일을 읽음(스키마, 파티션, 스냅샷 및 현재 스냅샷 정보)
- 현재 스냅샷 = **s1**

- manifest list 파일을 읽음(manifest 파일 정보, 스냅샷 정보)
- 두개의 manifest 파일을 참조하고 있지만, **현재의 스냅샷**을 확인하여 최종 1개의 manifest 파일 확인

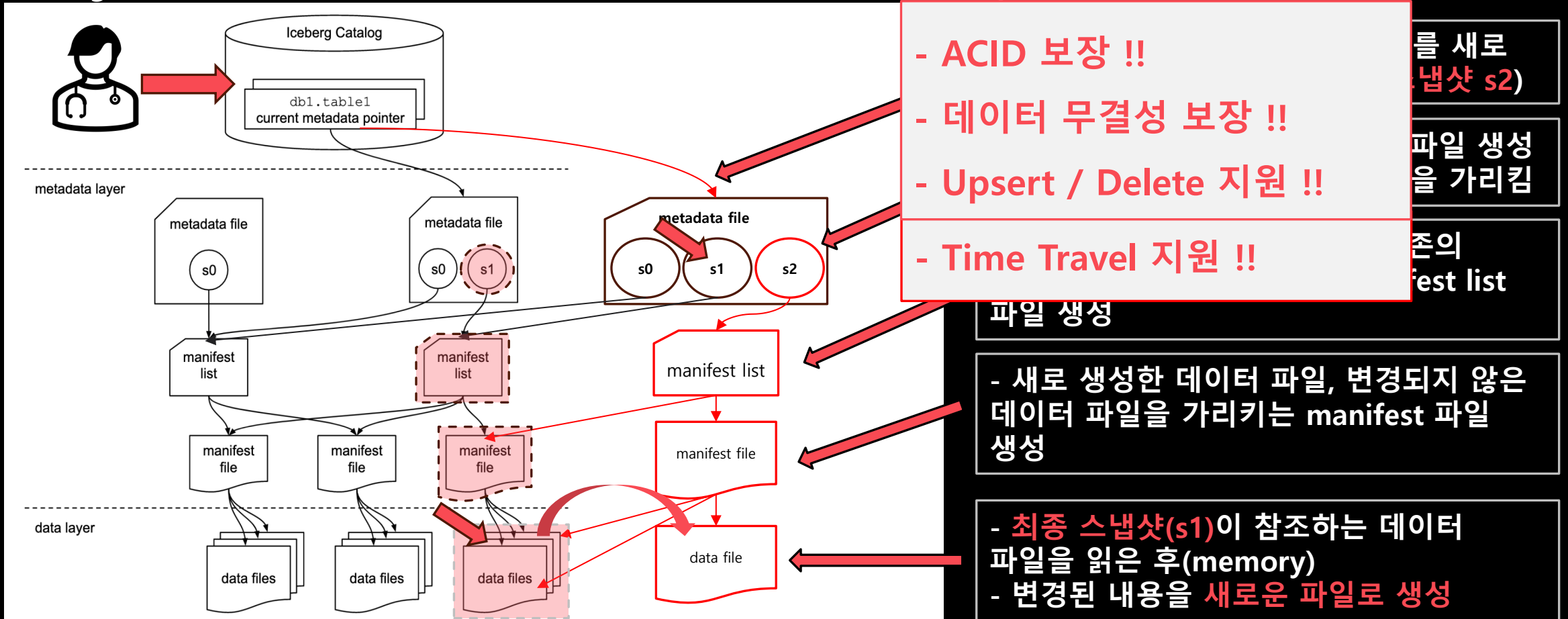
- manifest 파일을 읽음(데이터 파일, 세부 정보 및 통계)
- 최종적으로 읽어야 할 **데이터 파일을 확인**

- **데이터 파일을 읽음**
- SQL에 대한 결과 반환

# What is Apache Iceberg?

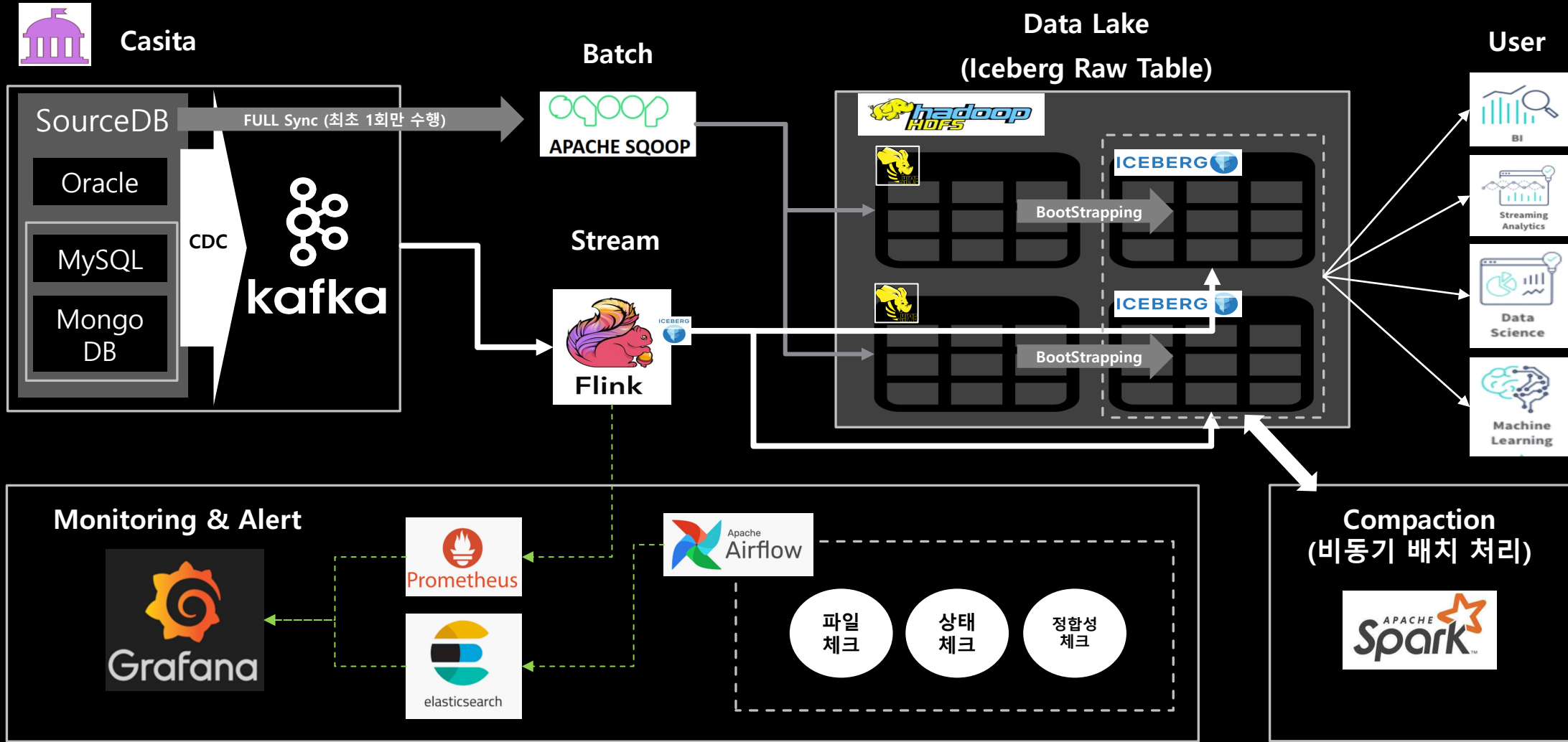
## Iceberg의 주요 기능

### iceberg table architecture



# 운영 환경

전체 운영 환경에 대해 설명 합니다



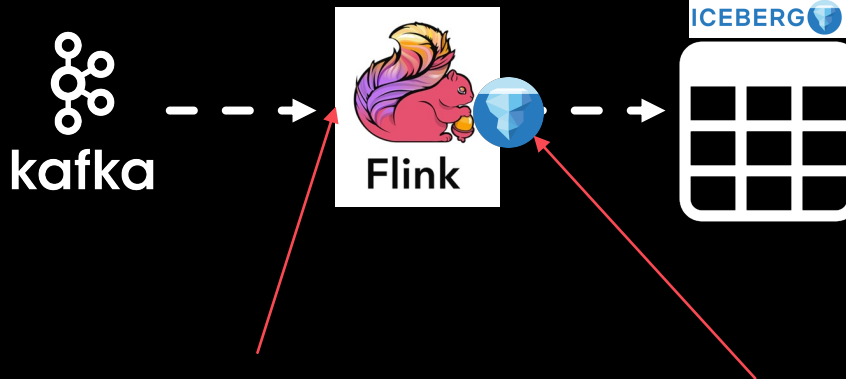


# Flink + Iceberg integration

11 TechTalk 2023

## Apache Flink

- 고성능
- 고가용성
- 3세대 대표 스트림 처리 프레임워크



## 기본 설정 및 catalog 생성

```
SET execution.checkpointing.interval = 10s;
SET sql-client.execution.result-mode =
tableau;

CREATE CATALOG hive_catalog WITH (
  'type'='iceberg',
  'catalog-type'='hive',
  'uri'='thrift://IP:PORT',
  'warehouse'='hdfs://nn/tmp/iceberg'
);

use catalog hive_catalog;
```

## dynamic table 생성(kafka)

```
CREATE TABLE kafka_table (
  `id` STRING,
  `name` STRING,
  `update_dt` TIMESTAMP(3) METADATA
FROM 'value.ingestion-timestamp' VIRTUAL,
  PRIMARY KEY (`id`) NOT ENFORCED
) WITH (
  'connector' = 'kafka',
  'topic' = 'source.dbserver1.test_db.cdc_test',
  'properties.bootstrap.servers' = 'IP:PORT',
  'properties.group.id' = 'consumergroup1',
  'scan.startup.mode' = 'earliest-offset',
  'value.debezium-json.schema-include' = 'true',
  'value.format' = 'debezium-json'
);
```

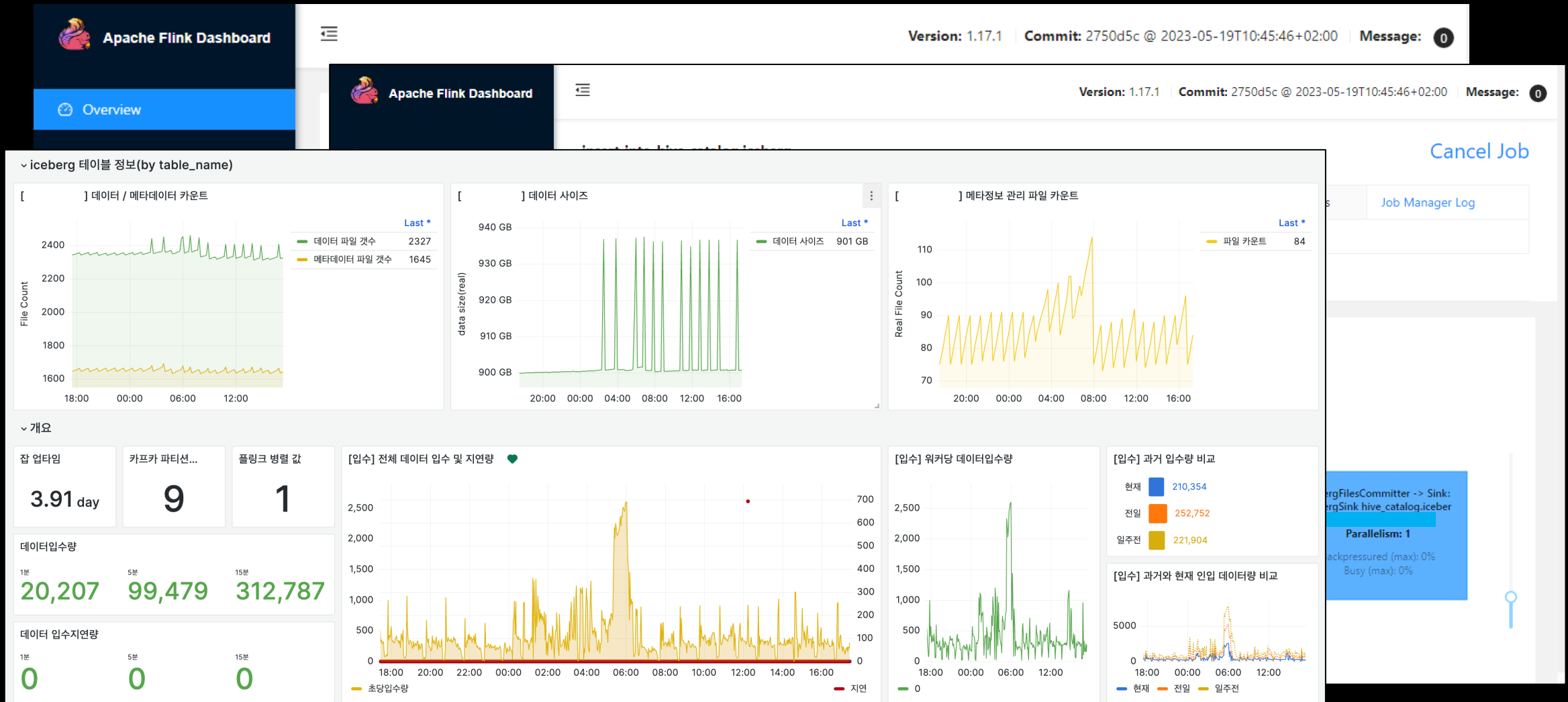
## continuous table 생성(iceberg)

```
CREATE TABLE IF NOT EXISTS
`iceberg_db`.`iceberg_test_01` (
  `id` STRING,
  `name` STRING,
  `update_dt` TIMESTAMP(3),
  PRIMARY KEY (`id`) NOT ENFORCED
) WITH (
  'engine.hive.enabled'='true',
  'format-version'='2',
  'write.upsert.enabled'='true'
);

insert into
`iceberg_db`.`iceberg_test_01` select * from
`default_catalog`.`default_database`.`kafka_table`;
```

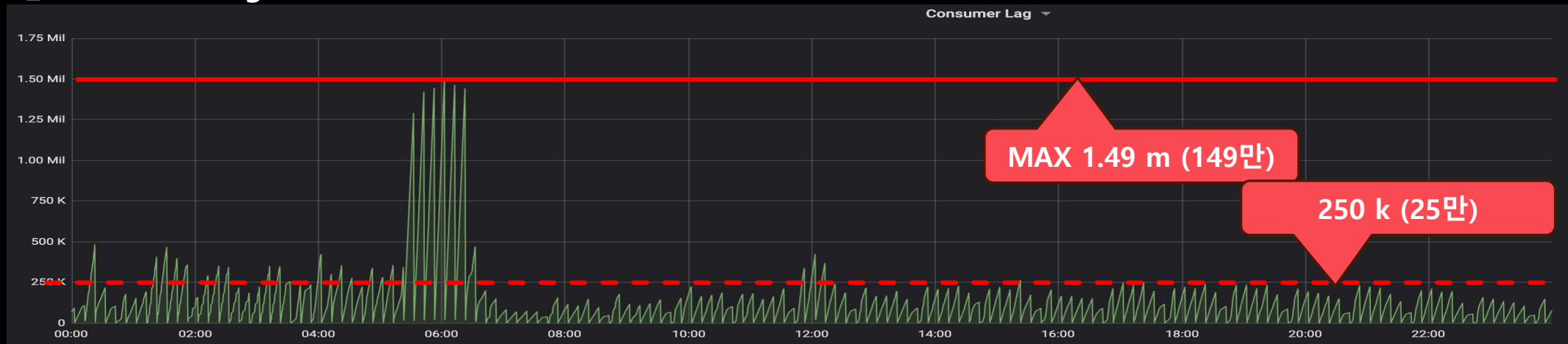
# Flink + Iceberg integration

## Flink Dashboard

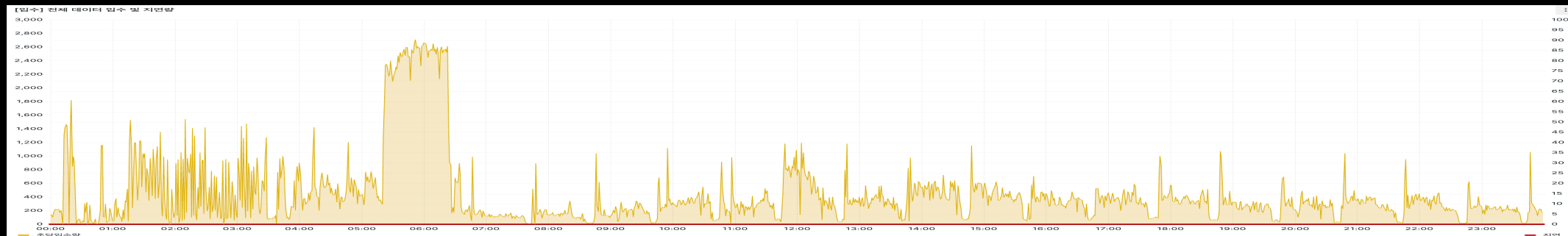


## 스트림 처리 성능

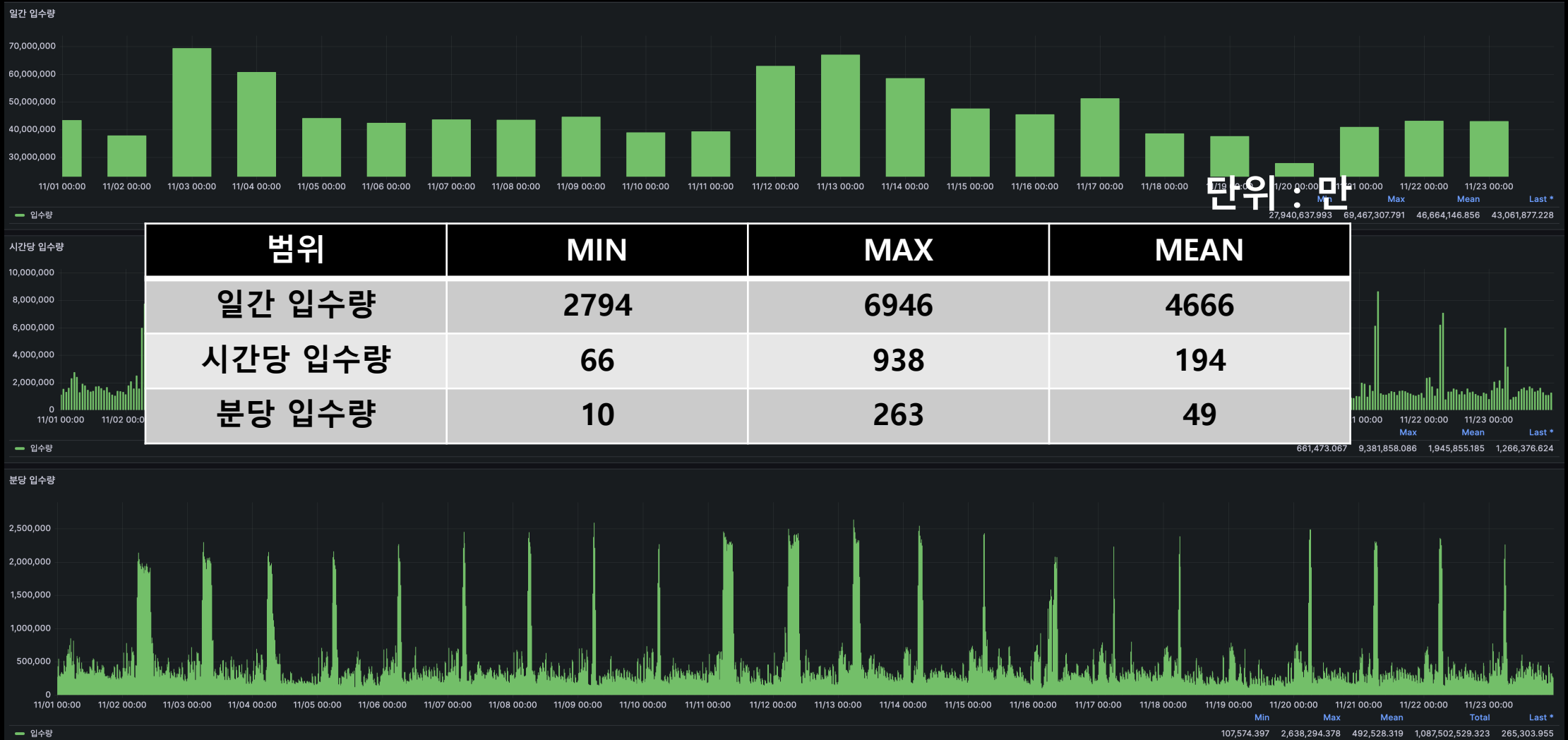
### a\_table kafka lag



### a\_table 인입 데이터 처리량 및 지연

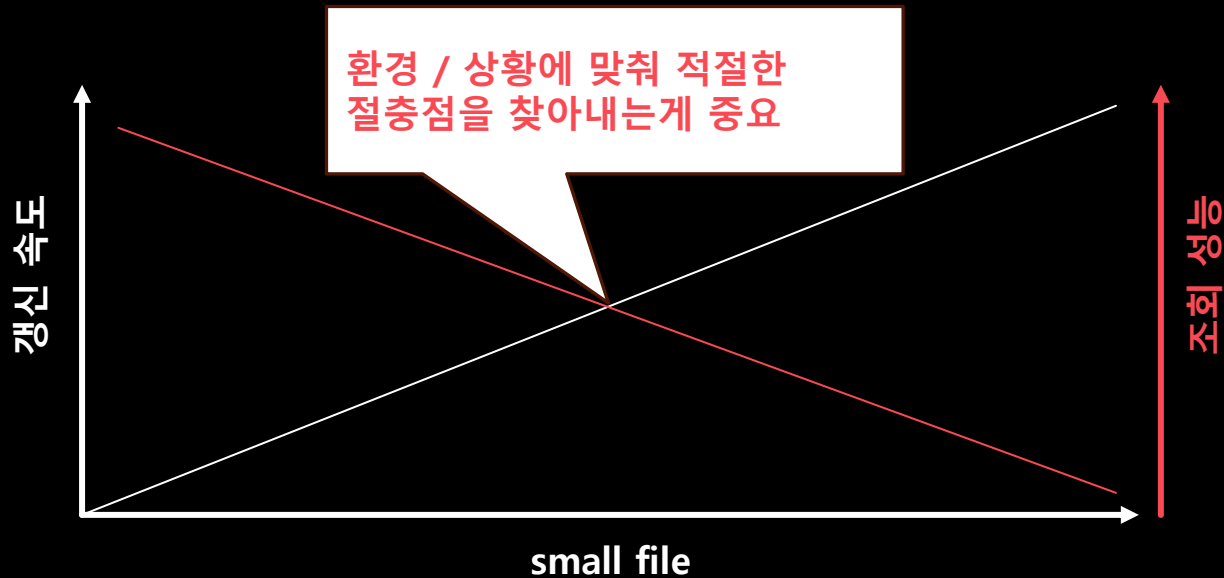


## 스트림 처리 성능



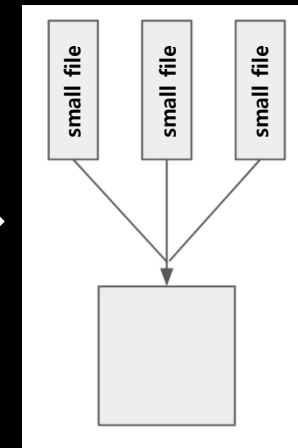
## Trade-off

- 실시간 갱신이 가능하지만 그에 따라 관리 포인트가 증가  
compaction(조회 성능), monitoring(이슈 대응), recovery(초기화)



Compaction은 많은 리소스 필요

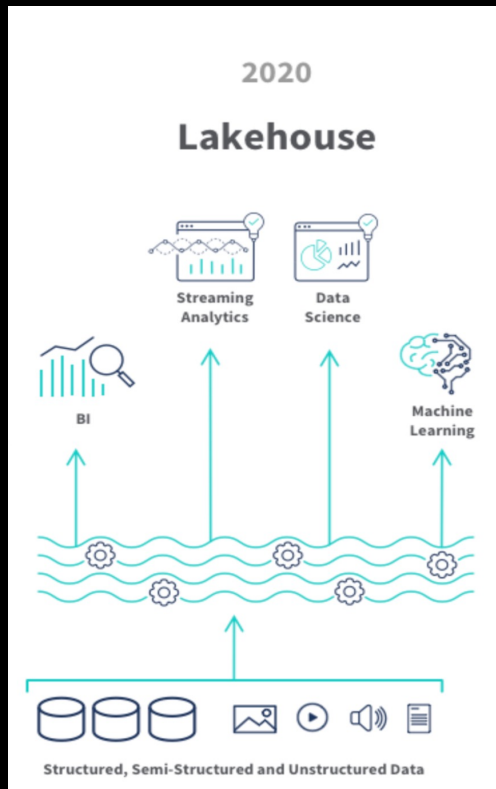
## Compaction(압축)



- small file 끼리 병합 -> 압축
- 조회 성능 향상
- 리소스 매니저 부하 경감

# 기대효과

데이터 기반 의사결정이 빨라 진다



실시간 검색색인 → 판매자와 구매자의 만족도 증가

실시간 추천 → 구매자의 구매결정에 도움

실시간 분석 → 의사결정 속도 증가



서비스 품질 ↑

# Q&A

E-mail)

김두진 : [djkorea@sk.com](mailto:djkorea@sk.com)  
김용주 : [yjkim0083@sk.com](mailto:yjkim0083@sk.com)

**Thank you**