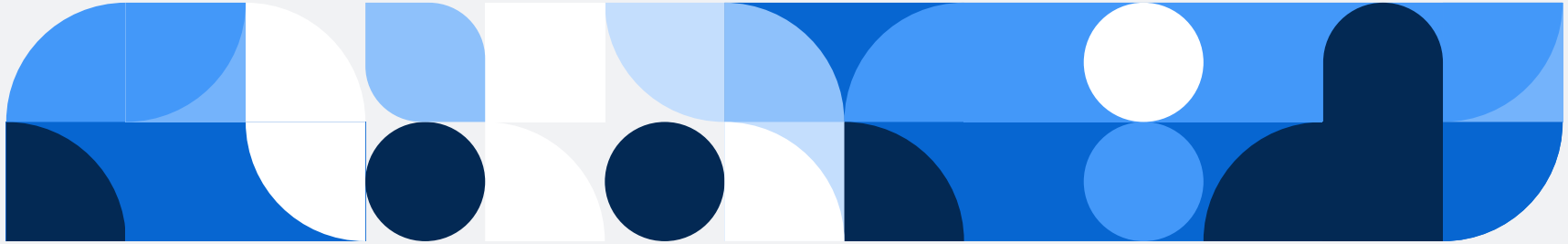
SAS Viya를 통한 스토리지 및 분석 성능 최적화

2026. 02. 05
한국쌔스소프트웨어(유)



SAS Viya

AI 및 모든 분석 라이프사이클 연계를 통해 핵심 질문을 신뢰할 수 있는 의사결정으로 전환합니다.

- 엔드 투 엔드
- 클라우드 비 종속성
- 오픈소스 개방성
- 분석가 Skill-set 비종속성
- 에이전틱 AI (워크플로우)
- 비용-성능 최적화

sas viya

Studies conducted by:

The Futurum Group

Read the full report

성능

30x FASTER

더 **빠른** 인사이트 확보

클라우드 비용 **절감**

생산성

>4x MORE PRODUCTIVE

더 **많은** 결과 산출

경쟁 우위

어떻게

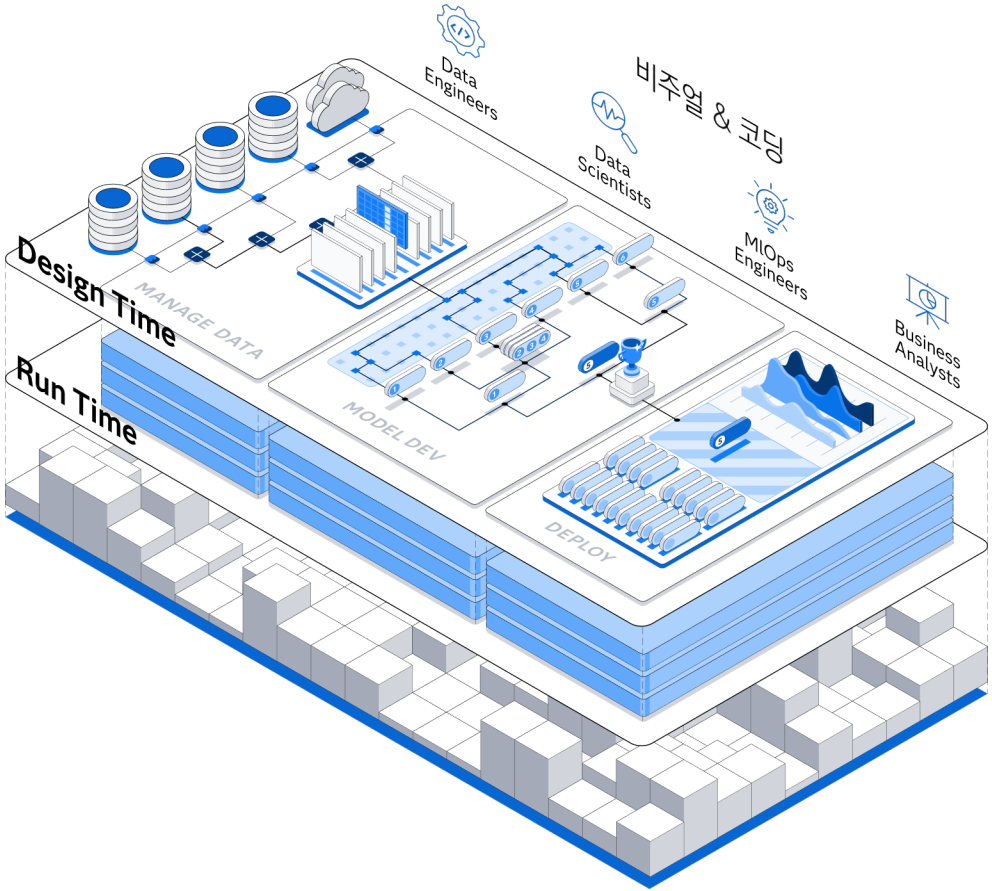
AI 개발 생산성,

성능 및 **확장성**을

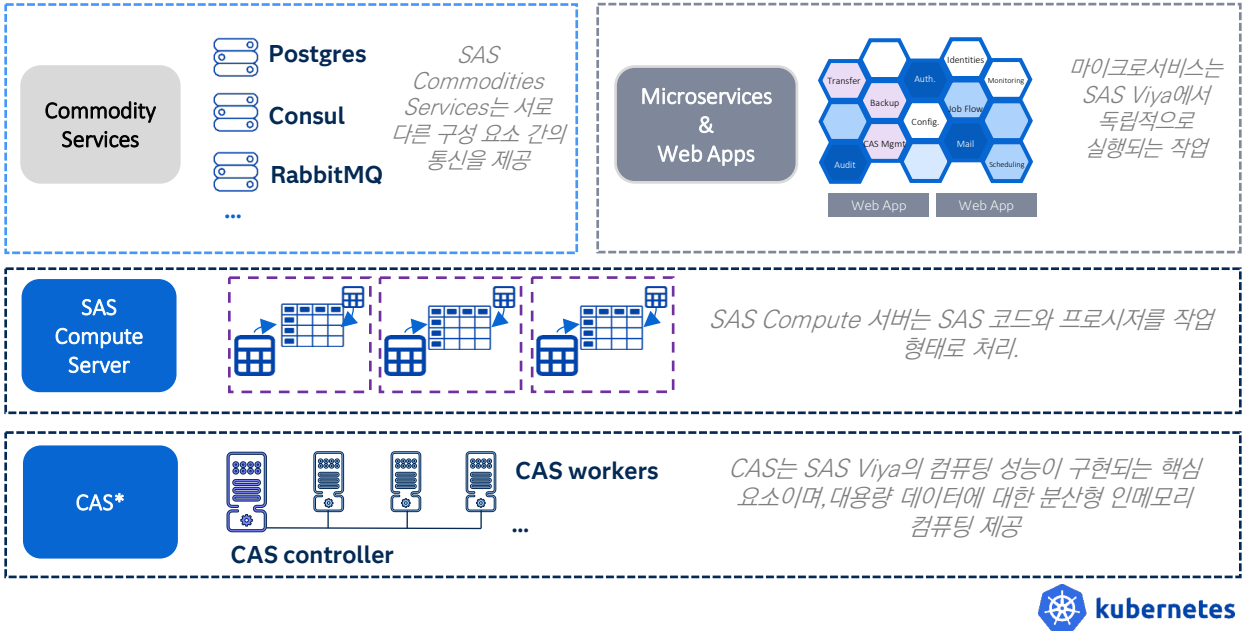
동시에 확보하는가?

- **Design Time** : 생산성
- **Run Time** : 성능 및 확장성

sas viya



The Basics of SAS Viya's Architecture

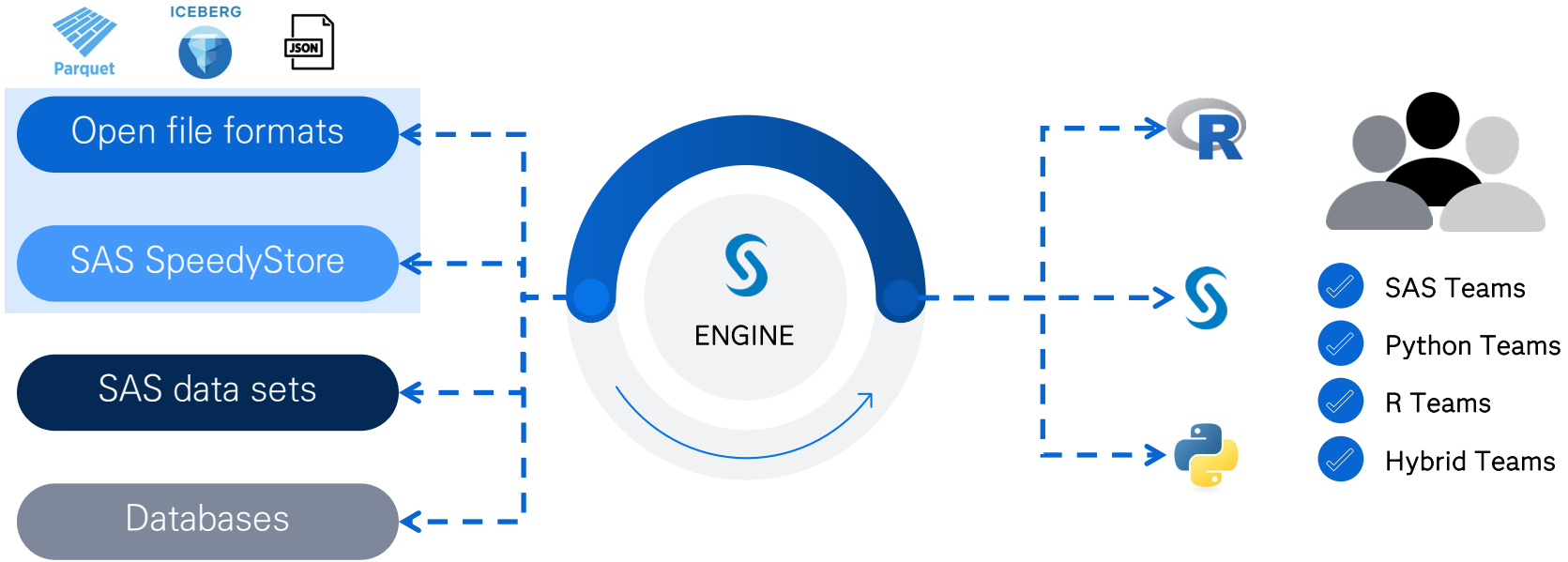


THE ENGINE | **CONTROL SYSTEM** | **SUPPORTING PARTS**

MTV (Move to Viya)

작업 유형	주요 작업	SAS 9 주요 환경	SAS Viya 주요 환경
데이터 준비	-데이터 가공 -테이블 조인 -데이터 품질 개선	• SAS Display Manager • SAS Enterprise Guide • SAS Data Integration Studio	• SAS Studio (코딩) • SAS Data Studio • SAS Enterprise Guide
데이터 시각화	-데이터 탐색 -리포트 생성	• SAS Display Manager • SAS Enterprise Guide	• SAS Visual Analytics
통계 분석	-회귀 모델 -최적값	• SAS Display Manager • SAS Visual Statistics • SAS/STAT	• SAS Studio (코딩) • SAS Visual Statistics
데이터 모델링	-지도 학습 -비지도 학습	• SAS Display Manager • SAS Enterprise Miner	• SAS Studio (코딩) • SAS Visual Statistics • SAS Visual Data Mining and Machine Learning
모델 관리 및 적용	-모델 관리 -모델 적용	• SAS Model Manager	• SAS Model Manager • MAS (Micro Analytics Services) • SCR (SAS Container Runtime)

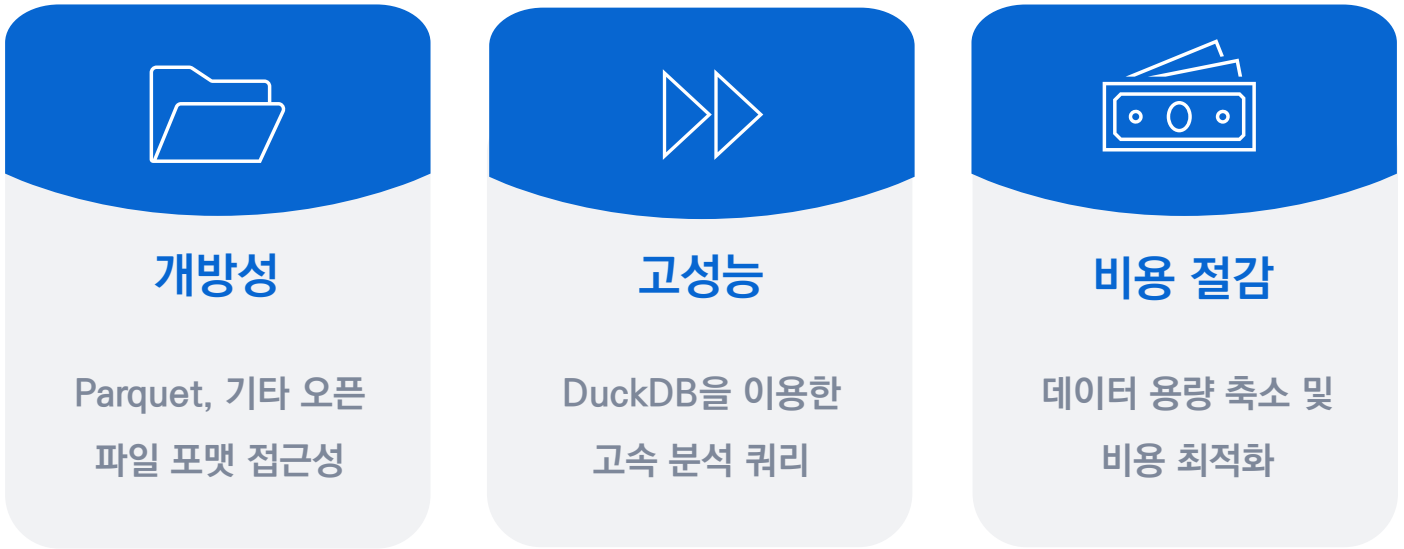
Viya : 다양한 데이터 형식 및 경량화된 분석 엔진



다양한 데이터 옵션

다중 언어 아키텍처

SAS/ACCESS to DuckDB



- SAS/ACCESS to DuckDB에는 프로세스 내에서 실행되는 내장 DuckDB 데이터베이스가 포함되어 있습니다.
- 추가적인 HW이나 설치가 필요하지 않습니다.

고객 데이터 압축 테스트

Columns: 53 Rows: 1,034,930,000

Enter expression

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20	X21	X22	X23	X24	X25	X26	X27	X28	X29
1	1	AR51	8	1	31	경기		0	0	11	31	경기	1	382213	191107	431140	215640	0	0	0	0	0	0	0	0	0	0	0	0
2	10	AR51	61	2	$	해당사항없음		0	0	01	22	인천	2	790869	316347	849800	340000	0	0	0	0	0	0	0	0	0	1	0	0
3	100	AG4090	5	1	31	경기		0	0	01	11	서울	2	483808	435427	589310	530410	0	0	0	1	0	0	0	0	0	0	0	0
4	1000	AF001	89	1	37	경북	20200522	0	0	21	37	경북	1	242496	145497	402280	241380	0	0	0	0	0	0	0	0	0	1	0	0
5	10000	AI639	70	1	11	서울		0	0	01	11	서울	1.5	632795	253118	691730	276730	0	0	0	0	0	0	0	0	1	1	0	0
6	100000	AM952	62	2	22	인천		0	0	01	11	서울	1.5	503471	201388	680860	272360	0	1	0	0	0	0	0	0	1	1	0	0
7	100001	AR42	81	2	25	대전		0	0	11	25	대전	4	1001738	500871	1058980	529580	0	1	0	1	0	0	0	0	0	0	0	0
8	100002	AC3499	76	1	22	인천	20191021	1	0	01	22	인천	1	421239	400177	5888310	5513570	0	0	0	0	0	0	0	0	0	0	0	0
9	100003	AC5090	54	2	31	경기		0	0	01	31	경기	3	672178	638569	746450	709150	0	0	0	0	0	0	0	0	0	0	0	1
10	100004	AG419	35	2	24	광주	20190508	1	0	01	24	광주	2.5	659283	626319	10391870	9846150	0	1	0	1	0	0	0	0	0	0	0	0
11	100005	AR51	51	2	31	경기		0	0	11	31	경기	2.5	551845	303514	599630	329830	1	0	0	0	0	0	0	0	0	0	0	0
12	100006	AI671	60	1	38	경남		1	0	01	21	부산	3	969118	387646	1390870	669990	0	1	0	0	0	0	0	0	0	1	0	0
13	100007	AR42	60	1	21	부산		1	0	11	21	부산	4	1031101	515552	4023990	2778720	1	1	0	0	0	0	0	0	0	0	0	0
14	100008	AH812	31	1	25	대전		0	0	21	25	대전	2.5	565074	339044	760640	456440	0	0	0	1	0	0	0	0	0	0	0	0
15	100009	AG431	79	2	11	서울		0	0	21	31	경기	1	300864													0	0	0
16	10001	AI509	74	2	22	인천		0	0	01	11	서울	1	415220													1	0	0
17	100010	AR42	59	2	37	경북		0	0	11	37	경북	2	571926													0	0	0
18	100011	AH8130	48	2	23	대구		0	0	11	37	경북	1.5	484107													0	0	0
19	100012	AI671	65	2	21	부산		0	0	11	21	부산	1.5	538913													1	0	0
20	100013	AF067	76	1	31	경기		0	0	21	22	인천	1.5	438558													1	0	0
21	100014	AI6352	76	1	32	강원		1	0	11	32	강원	3	855463													0	0	0
22	100015	AG441	45	1	25	대전		0	0	11	25	대전	1	175638															
23	100016	AG510	56	1	34	충남		0	0	11	34	충남	1	175638															

데이터 형식	데이터 용량
SAS 데이터셋	32.4 GB
압축된 SAS 데이터셋	17.5 GB
Parquet 파일 형식	6.9 GB

4.7x

SAS® Studio - 코드 및 플로우 개발

새로 만들기 옵션 뷰 열기 모두 저장

SAS Studio compute context

SAS 서버

이름

- 내 즐겨찾기
- 폴더 바로 가기
  - Global Shortcuts
  - My Shortcuts
    - dlib 바로 가기
    - sasdata 바로 가기
- SAS 서버
  - Home

시작 페이지 | * SAS School_ktkim.sas

실행 취소 지우기 디버그 플로우에 복사 Snippet

코드

SAS 콘텐츠: /Users/keun-Tae.Kim@sas.com/My Folder/SAS School_ktkim.sas

```
1  options fullstimer;
2
3
4  /* SAS libname statement */
5  libname saslib '/data-netapp-ultra/data/demo-quickstart/sasdata';
6
7  /*************************************************************************
8  /* Combine all 12 months of nyc trips into a single data set for 2011.  Th
9  /* Data Step processing takes 2-4 minutes,  if demonstrating for a custome
10 /* you don't necessarily need to run it, but do highlight the need for ext
11 /* processing and note the additional time. Each monthly file is ~2GB
12 /*************************************************************************
13
14 /* data saslib.yellow_tripdata_2011; */
15 /*   set saslib.yellow_tripdata_201101  */
16 /*    saslib.yellow_tripdata_201102 */
17 /*    saslib.yellow_tripdata_201103 */
18 /*    saslib.yellow_tripdata_201104 */
19 /*    saslib.yellow_tripdata_201105 */
20 /*    saslib.yellow_tripdata_201106 */
21 /*    saslib.yellow_tripdata_201107 */
22 /*    saslib.yellow_tripdata_201108 */
23 /*    saslib.yellow_tripdata_201109 */
24 /*    saslib.yellow_tripdata_201110 */
25 /*    saslib.yellow_tripdata_201111 */
26 /*    saslib.yellow_tripdata_201112; */
27 /* run;    */
28
29 /* Query a single Year(2011) SAS dataset (27GB) */
30
31
32 PROC SQL;
33
34   SELECT
35     passenger_count,
36     payment_type,
37     count(*)           AS num_trips,
38     avg(trip_distance) AS avg_distance,
```

로그 결과 출력 데이터

(0) (0) (9)

메시지가 없습니다.

```
1   /* region: 프리앰블 생성 */…
79
80  options fullstimer;
81
82
83  /* SAS libname statement */
84  libname saslib '/data-netapp-ultra/data/demo-quickstart/sasdata';
NOTE: 라이브러리 참조 'SASLIB'이(가) 다음과 같이 할당되었습니다.
      엔진:        V9
      물리적 이름: /data-netapp-ultra/data/demo-quickstart/sasdata
85
86
87  /* region: 포스트앰블 생성 */…
100
```

검색

오전 9:43
2026-02-05

데이터 비용 절감 방안

Datasets

sas7bdat…
100TB

NetApp Ultra

연간 $491,000
월 $0.40 GiB 기준

기준

parquet…
25TB

NetApp Ultra

연간 $122,000
월 $0.40 GiB 기준

75% 절감

AWS S3
Azure Blob

연간 $6,000
월 $0.02 GiB 기준

98% 절감

저장소 비용 20배 저렴

SAS/ACCESS to DuckDB – 예시

New York Taxi Rides 데이터

- 데이터 용량 : 총 2.1 GB (Parquet) | 25 GB (SAS)
- 파일 수 : 12 Parquet 파일
- 레코드 수: 1억 7,100만 건 (19 컬럼)

Average Distance, Tip & Fare

```
PROC SQL;
 SELECT
  passenger_count,
  payment_type,
  count(*)        AS num_trips,
  avg(trip_distance) AS avg_distance,
  avg(fare_amount)   AS avg_fare,
  avg(tip_amount)    AS avg_tip
 FROM
  duklib.'2012/*.parquet'n(file_type=parquet
                file_path="&parquet_data")
 WHERE
  passenger_count is not NULL AND
  passenger_count > 0 AND
  passenger_count < 5 AND
  trip_distance < 100 AND
  trip_distance > 0
 GROUP BY
  passenger_count, payment_type
 ORDER BY
  payment_type, passenger_count;
QUIT;
```

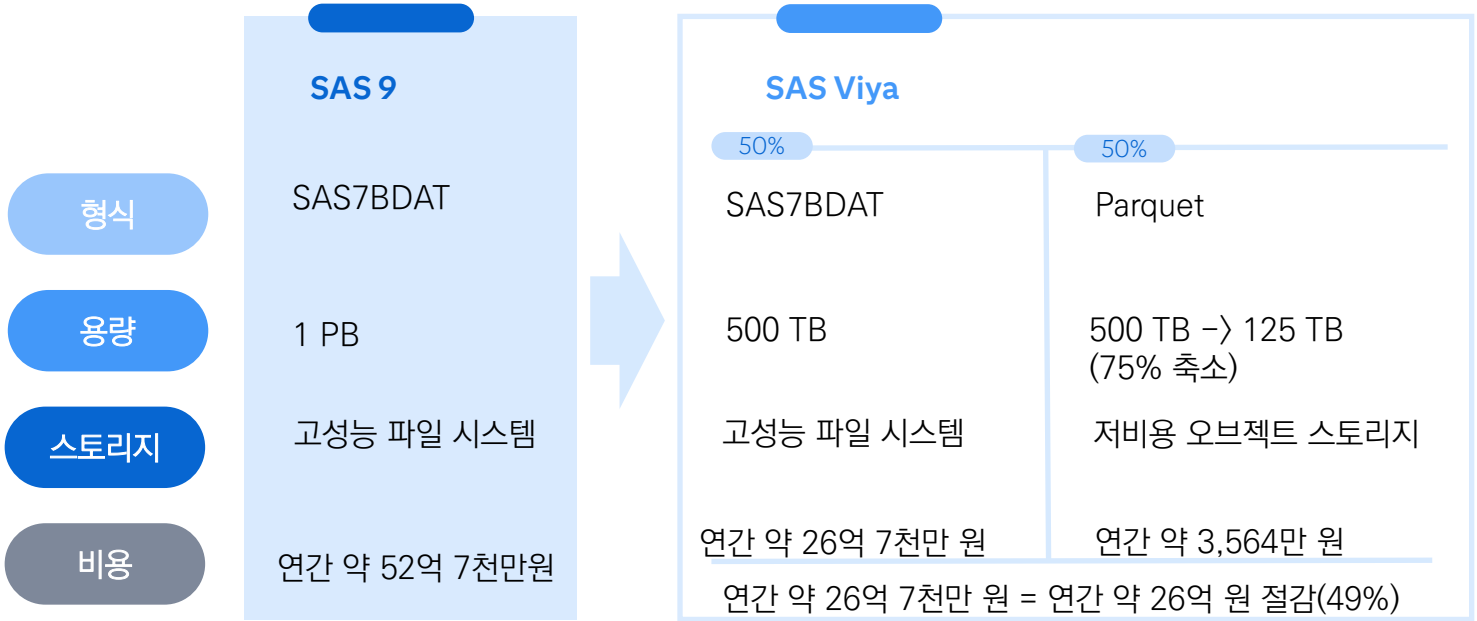
고성능 파일 시스템 데이터 접근

	쿼리 시간	속도 향상
BASE libref (sas7bdat)	1분 9초	기준속도
DuckDB (Parquet)	2.5초	27x

AWS S3에 저장된 데이터 접근

	쿼리 시간	속도 향상
DuckDB (Parquet)	11.62s	5.9x

데이터 저장 비용 최적화 사례



한 대형 은행은 SAS7BDAT 데이터의 약 절반을 Parquet 형식으로 이관했습니다.

Azure NetApp Files Premium 1 yr reserved €27,334 for 100 TiB/month = €328,125 for 100 TB/year
Azure Storage 1 yr Reserved €1,427 for 100 TB/month = €17,124 for 100 TB/year

데이터 스토리지 최적화를 위한 옵션

SAS 데이터셋

단순하고 사용하기 쉬움

SAS Viya 및 SAS 9에서 모든 기능 활용 가능

로컬 또는 연결된 파일 시스템에 저장 가능

압축 사용 고려

+

오픈 파일 형식

비용 효율적

다양한 기술에서 접근 가능

데이터 용량 감소

저비용 오브젝트 스토리지 사용 가능

레이블·포맷 미적용, SAS9 코드 호환성 고려.

+

SAS SpeedyStore

성능 및 제어

SAS VA 성능 향상 (Viya)

실시간 분석 및 업데이트

클라우드에서 오브젝트 스토리지 활용 시 전체 비용 절감 가능

인프라 준비 필요

별도 SpeedyStore 라이선스 필요

감사합니다!

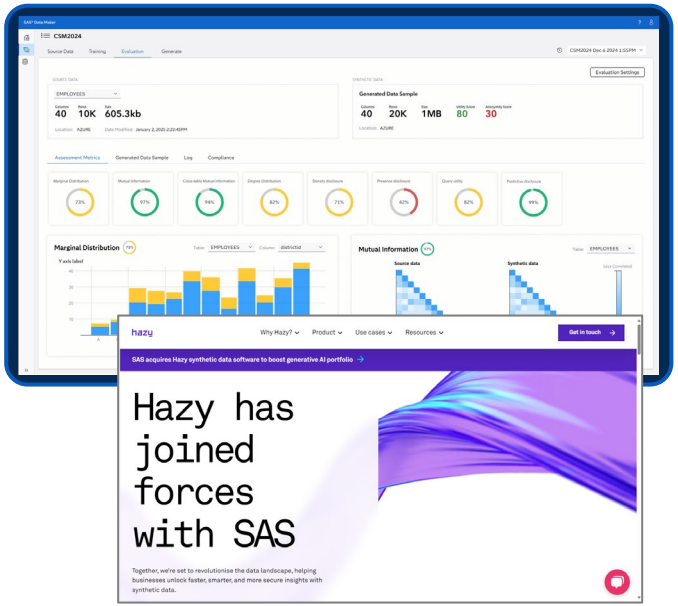
sas.com

Augmentation : 합성 데이터

2026년에는 75%의 기업이 생성형 AI로 합성 고객 데이터를 생성 할 것 (Gartner)

	Nationwide Building Society	British Bank	US Healthcare Provider
Problem	**신용 평가 모델을 학습**하기 위한 데이터의 가용성, 정확성, 확장성 및 데이터 민감도의 제한 사항	데이터 **개인 정보를 보호하면서, 외부 공급업체와 데이터를 공유**함	엄격한 규정으로 인해 **혁신과 연구를 위한 환자 전자 의료 기록(Electronic Health Record, HER) 데이터 공유**의 어려움
Action	SAS를 사용하여 합성 데이터를 생성하고 모델에 통합	**안전한 합성 데이터로 샌드박스에서 신상품을 개발**	SAS는 **EHR을 비식별화하고 합성 데이터를 생성, 평가 및 배포할 수 있는 안전한 통합 플랫폼**을 제공
Result	촉진된 ML 학습으로 **모델 정확도 28% 향상**, 잠재적 손실 감소 지원	**수백 명의 외부 개발자가 안전하게 액세스할 수 있는 샌드박스**를 제공했으며, 데이터 보안 및 접근 향상	환자들은 새롭게 활성화된 연구의 혜택을 받으며 고객 개인 정보 보호에 대한 위험 감소

SAS Data Maker



SAS SpeedyStore

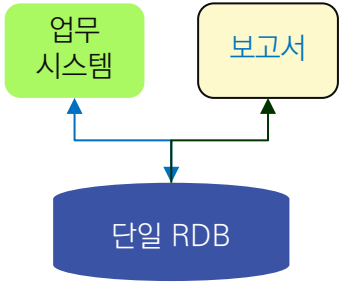
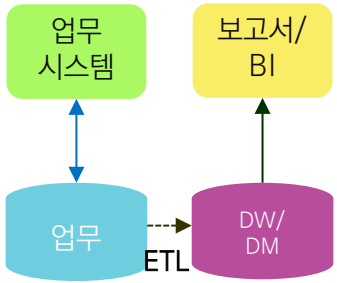
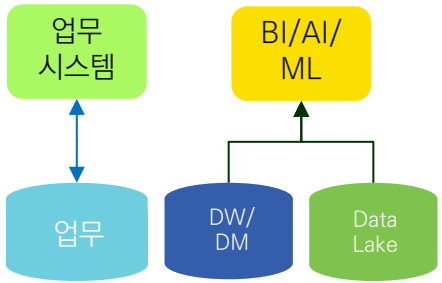
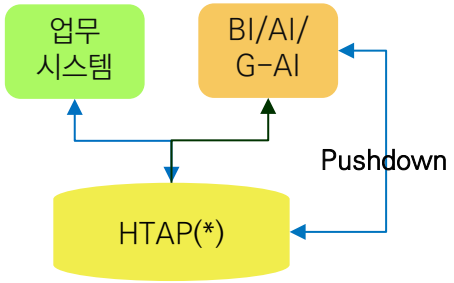
최신 AI/ML Platform과 DB 기술의 융합으로 생산성 향상

2026. 02. 05
한국쌔스소프트웨어(유)

SAS SpeedyStore 란?

SAS SpeedyStore는 데이터 플랫폼(SingleStore)과 와 AI플랫폼(Viya)을 통합한 SAS 제품

관계형 DB의 과거, 현재와 미래

	1세대(운영중심)	2세대(운영/분석 분리)	3세대(빅데이터/어플라이언스)	4세대(융합/Hybrid)
시기	1990년대 초반 ~	1990년대 말 ~	2010년대 초 ~	2020년대 초 ~
용도 / DB				
주요 특징	• **운영업무 데이터 저장소**로 주로 사용 • 동일한 데이터에서 보고서 생성 • 보고서 중심(Offline)의 사업 **결과 분석**	• **데이터 모델 중심의 업무과 분석 시스템 분리** (OLTP(*)/OLAP(**)) • 다차원 데이터 분석의 시작 (DW/BI) • 사업의 **결과와 원인 분석**	• **빅데이터의** 도래 • AI/ML 기반의 **예측 분석** 시작 • **MPP 구조의 어플라이언스 DB** 출현 (대용량/고성능) • 데이터 유형/가치별 저장소 분리 (DW/Data Lake) • No SQL / Column Store….	• **Hybrid DB**의 출현 • **최적화 및 처방적 분석** • 관계형, Vector, Text 유형의 데이터 저장/처리 엔진과 통합 • AI 및 분석 기능 내장 (In-DB 처리) • Cloud 지향 • Universal Storage(*****) 지원
제품	Oracle, DB2, Informix…	Oracle, DB2, **Sybase(***), Teradata(****)**	**Exadata, Netezza, SAP HANA(*****), Vertica**	***Exadata***, **SAP HANA, SingleStore(******)** …

(*) Online Transaction Processing, (**) Online Analytic Processing), (***) 최초 열 지향, (****) 최초 MPP, (*****) 최초 In Memory기반 Hybrid, (******) Disk 기반 Hybrid and AI 통합 (*******) 고성능 및 저비용 고용량의 저장소 통합 지원

4세대 지향 Hybrid Transactional/Analytics DB (From Dual Store To Single Store)

SAS SpeedyStore에 통합된 SingleStore DB는 다양한 형태의 데이터를 저장하고 대용량 및 다양한 데이터에 대한 분석 질의를 신속하게 실행함.

In Memory Rowstore

OLTP 최적
대량의 동시 업데이트
신속한 참조
운영을 위한 DB
신속한 거래 처리

+

Memory & Disc Columnstore

OLAP
수백만 건의 데이터
신속한 분석 질의
Data Warehouse/Data Mart + Data Lake
분석 지향

=

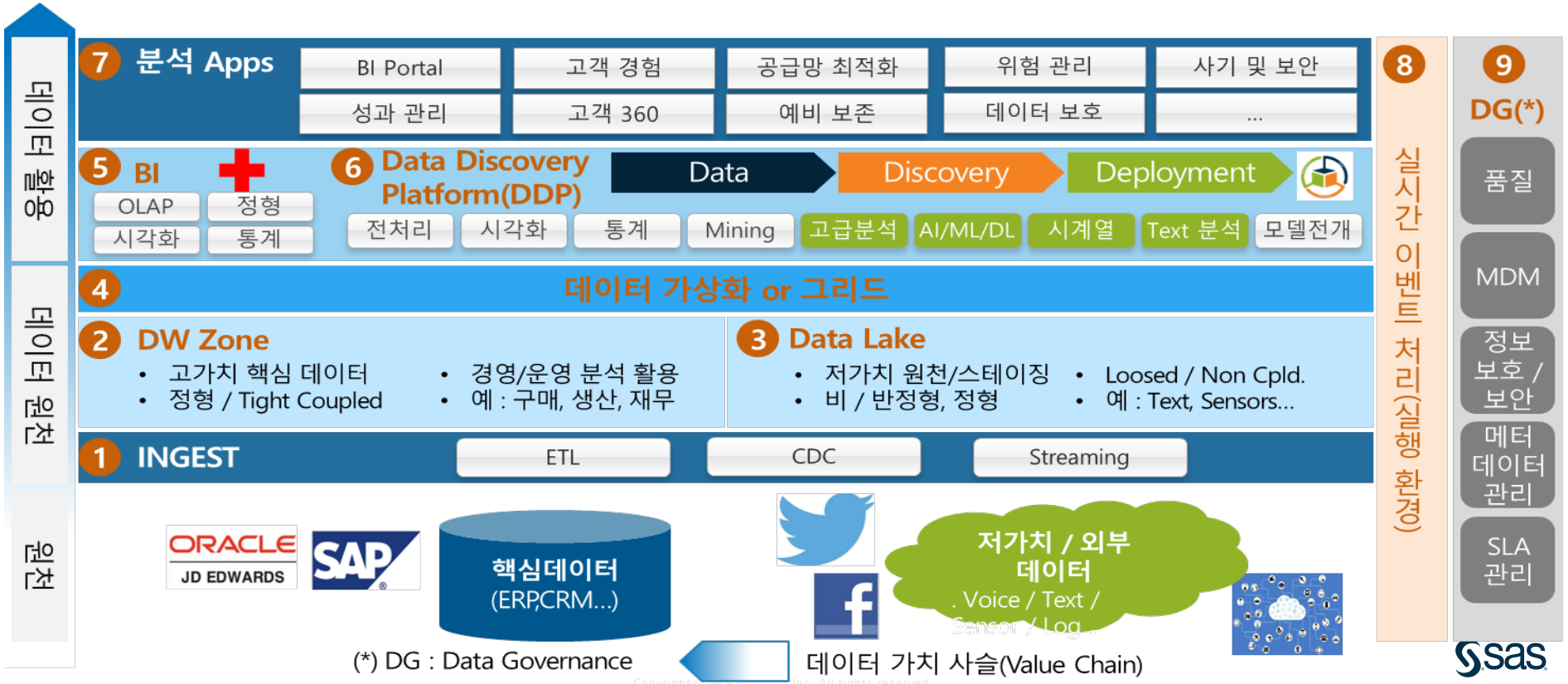
Universal Storage

SingleStore™

- 단순화된 구조
- No More ETL
- 저비용
- 저지연

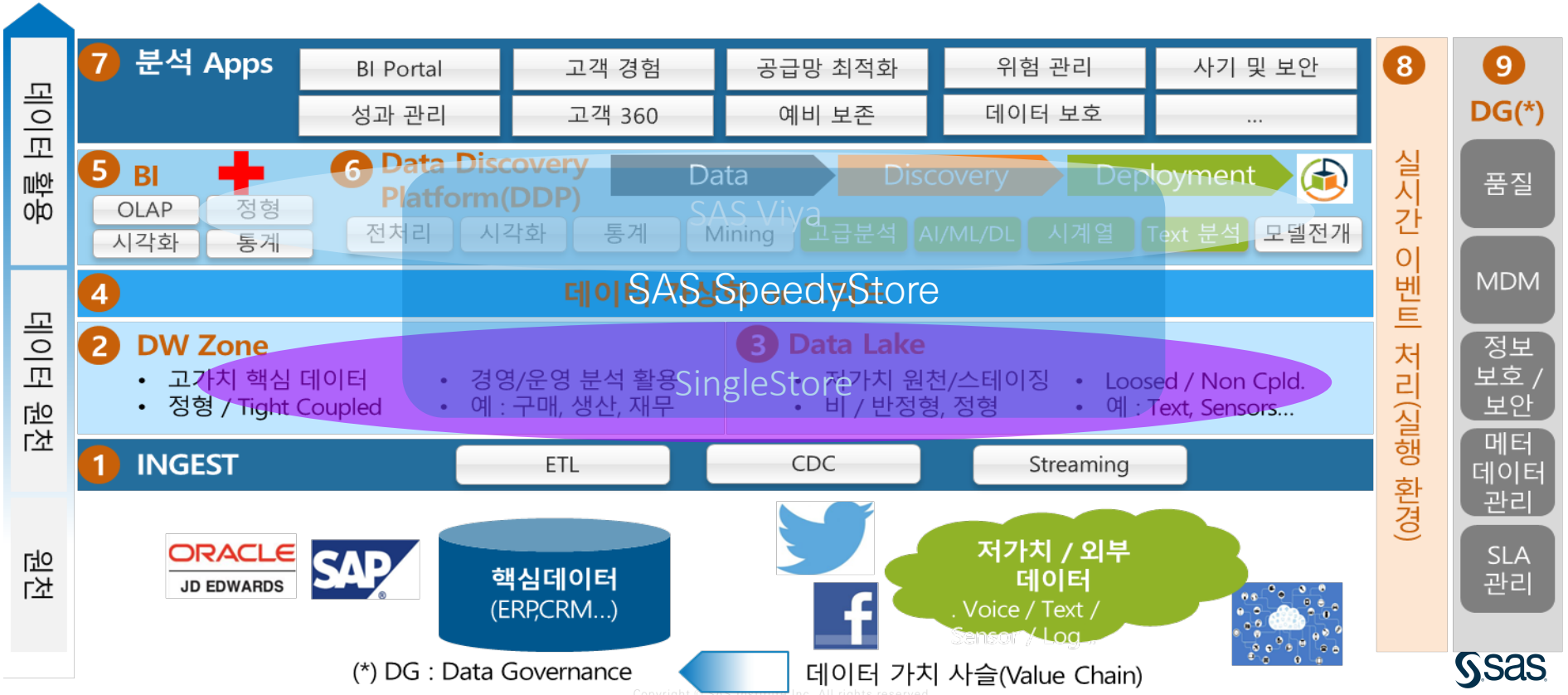
3세대 DB 기반의 Analytics Ecosystem Architecture

Analytics Ecosystem Architecture는 분석에 필요한 주요 영역을 계층화한 Framework으로 시스템 구축 시 복잡성 제거 및 단계적 구축을 지원함.



4세대 지향 – Analytics Ecosystem Architecture과 SAS SpeedyStore

SAS SpeedyStore는 4세대 지향 통합 DB와 최신의 AI/ML의 플랫폼과 상호 통합된 제품임.



Why SAS SpeedyStore

최신의 혁신적인 AI/ML과 DBMS 플랫폼 이 융합으로 데이터의 관리 및 분석의 생산성을 향상 시킴.

AI/ML 플랫폼

최신의 혁신 Platform

- AutoML와 No Code, Low Code로 **Citizen Data Scientist** 현실화
- **분석의 전과정**을 통합 지원
- Model Governance를 통합한 ModelOps를 구현

- AI/ML 과 Data 의 단일 플랫폼
- 복수의 플랫폼 단순화
- 신속한 분석 어플 개발
- 비용 절감

DB 플랫폼

혁신 기술

- **고 병렬/대용량** 데이터 처리
- 정형, 반/비정형 데이터 저장을 위한 **범용 저장소**
- 분석 질의 성능 개선을 위한 **Column Store 및 고압축률**
- 지속적 데이터 적재를 위한 **Pipeline**

기대효과

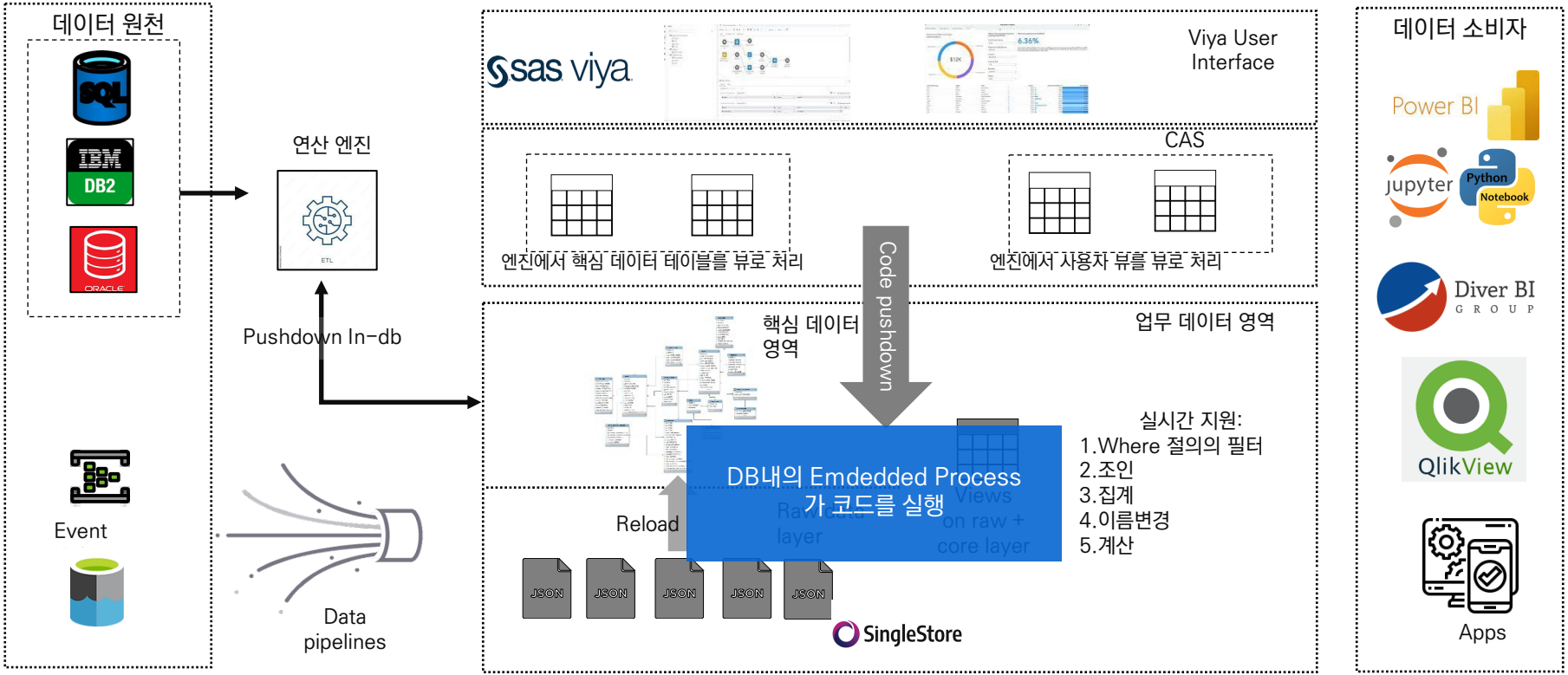
- 신속한 **AI/ML Platform 및 Data Platform** (Data Warehouse, Data Mart 및 Data Lake) 통합 구축
- 데이터 분석, 예측 모델 생성 및 배포 **생산성 향상**
- 통합 구축으로 분석을 이용한 업무 개선 효과에 대한 신속한 모니터링(Plan – Do – **See**)

SAS SpeedyStore 기본 구성

SAS의 Data와 AI를 위해 통합된 데이터 저장소

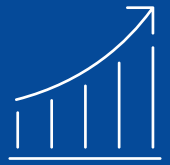
SAS SpeedyStore 논리 구성

SAS의 Data와 AI를 위해 통합된 데이터 저장소



SAS SpeedyStore 특/장점

SAS SpeedyStore로 데이터 가속화, 비용절감, 분석 어플 생성 단순화 및 분석의 통찰력을 확대 할 수 있음



데이터 가속화

데이터는 고성능과 보안성 있는 플레임워에 저장

확장가능한 데이터 패브릭으로 분석의 통제 및 생애주기에 대한 용이한 관리

비용 절감

인프라의 용량 절감은 비용절감을 유도

데이터 관리 절차 및 비용 감소

어플 및 데이터 중복의 기술적 부채 감소

분석 어플 생성

대량의 데이터 이동 및 복잡성 최소화

분석 절차의 효율성을 개선하여 데이터로부터의 가치 증대 유도

분석 통찰력 확대

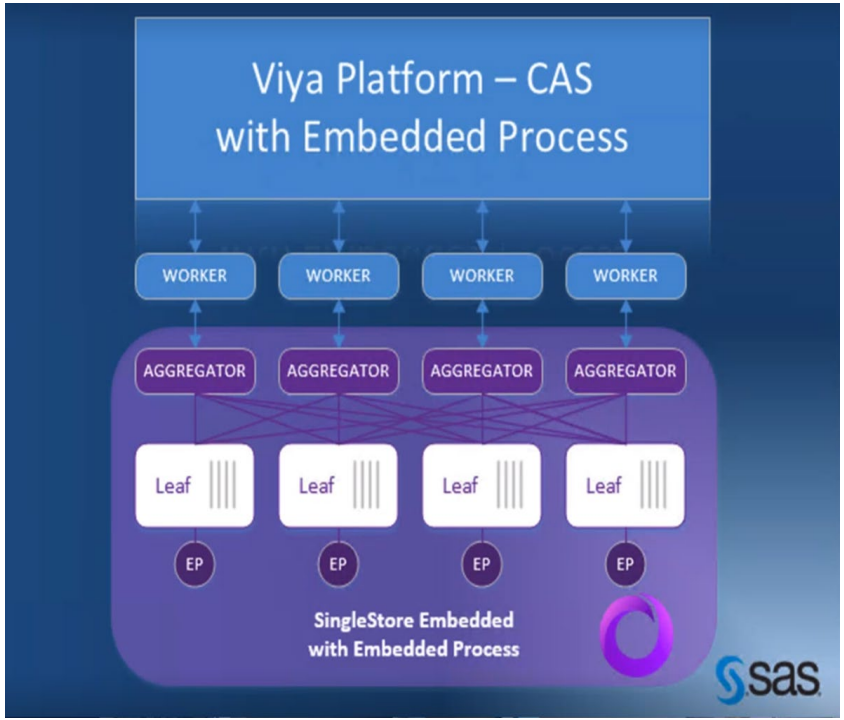
작업부하, 어플리케이션, 사용 자 간의 모든 분석 데이터에 대한 접근 제공

실시간 분석 인사이트를 제공하는 최신의 어플리케이션 제공

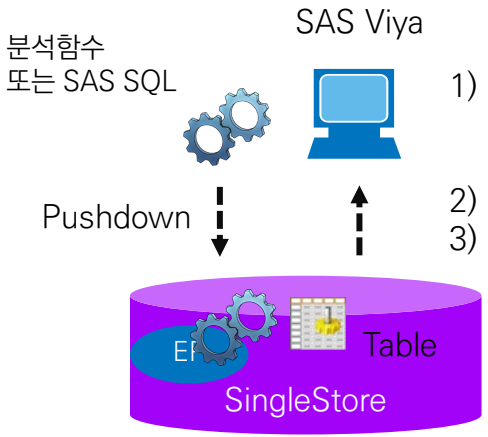
분석을 데이터로 이동

EP(Embed Processor) 엔진이 S2에 내장되어 SAS 코드가 DB에서 실행됨.

CAS와 S2 기본 구조



In-Database 흐름



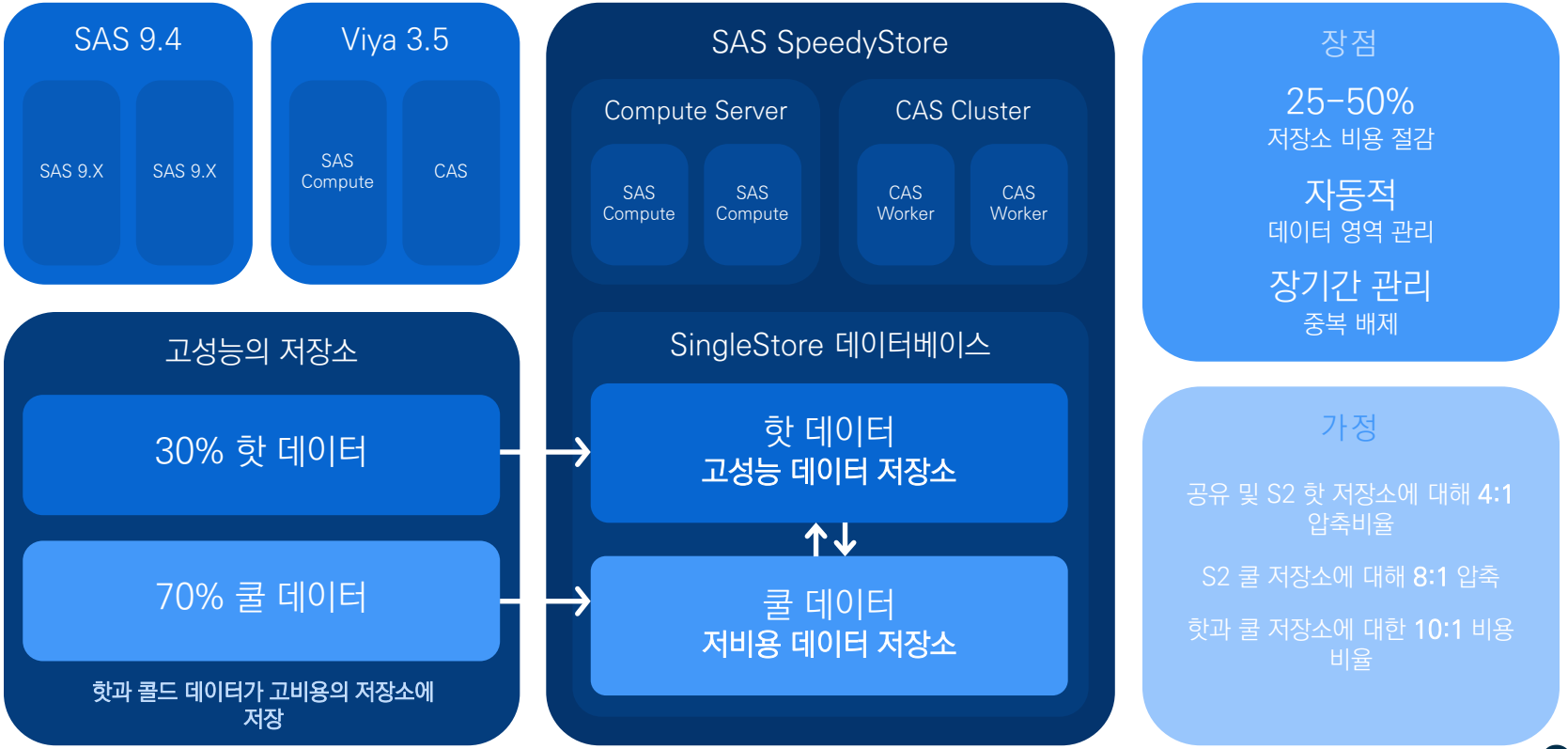
1) Viya의 분석함수 또는 SAS SQL이 S2에 설치된 EP로 Pushdown
2) EP와 S2에서 해당 기능 수행
3) 실행 결과를 Viya에 전송

기대효과

- 고성능의 DB 성능을 활용한 성능 향상
- 결과만을 전송함으로 데이터 전송 최소화
- 전반적인 분석 성능 개선

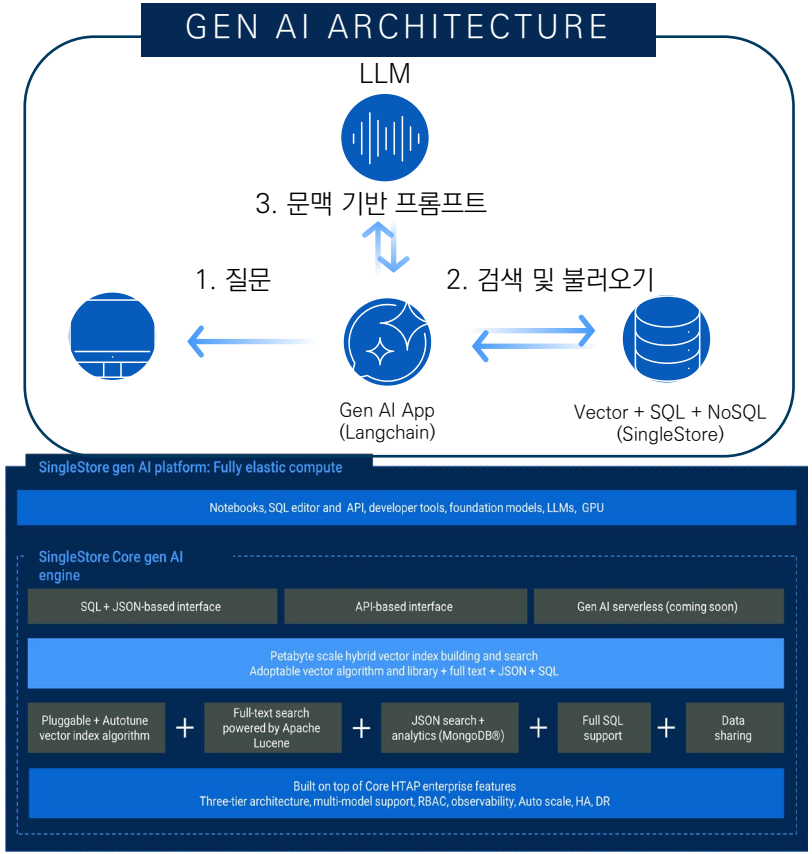
비용 절감

데이터 가치에 따른 저장소 분리, 고 효율 데이터 압축과 데이터의 중복 방지로 저장소 비용 절감



분석과 AI를 위한 단일 데이터 플랫폼

거래처리 및 분석을 위해 필요한 속도, 단순성 및 확장성 제공



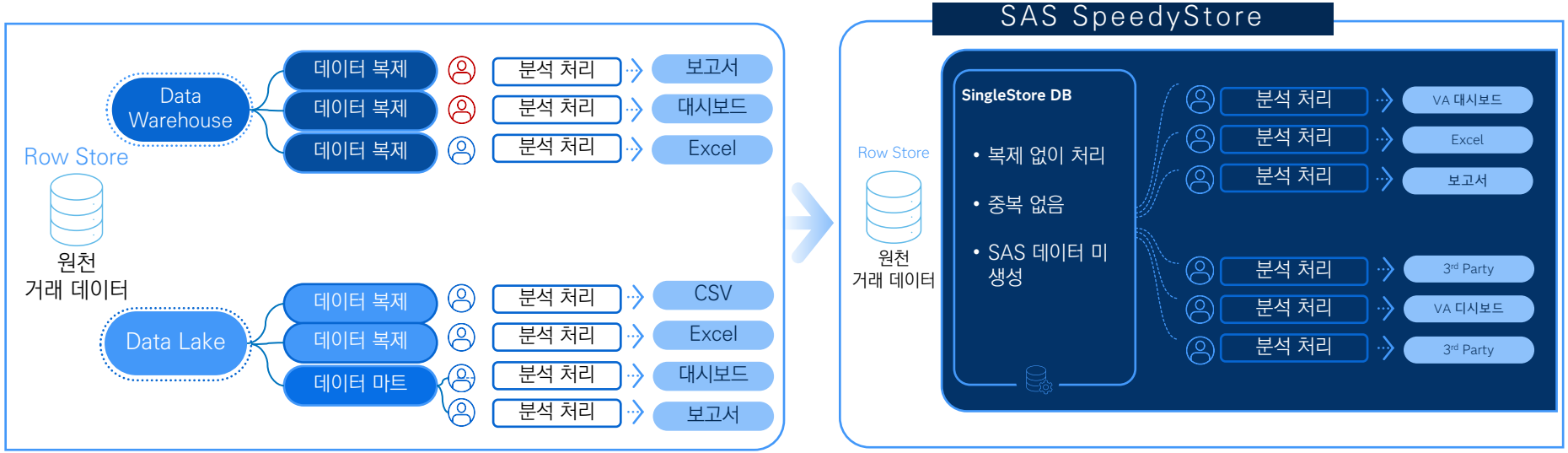
장점

- Gen AI 어플을 위한 데이터 플랫폼
- Vector, SQL, NoSQL 등의 모든 유형에 맞는 단일 데이터 플랫폼
- 실시간 데이터 수집과 효율적인 대량의 벡터 적재를 위한 기능 제공 및 활용
- ACID 거래, 고가용성, 재해복구 및 특정 시점의 데이터 복구를 지원하는 전사 데이터 플랫폼

거래 및 분석	→	Row + Columnar	속도
SQL, NoSQL, Geospatial, Vectors 등 Etc.	→	Multi-model	단순성
인메모리, SSD와 오브젝트 저장소	→	3계층 저장소 + 무한 저장소	확장성
문맥적 검색	→	하이브리드 검색	단순성

데이터 통합 및 중복 제거

고성능의 데이터 처리로 성능 개선 등을 위한 데이터 복제를 회피함.

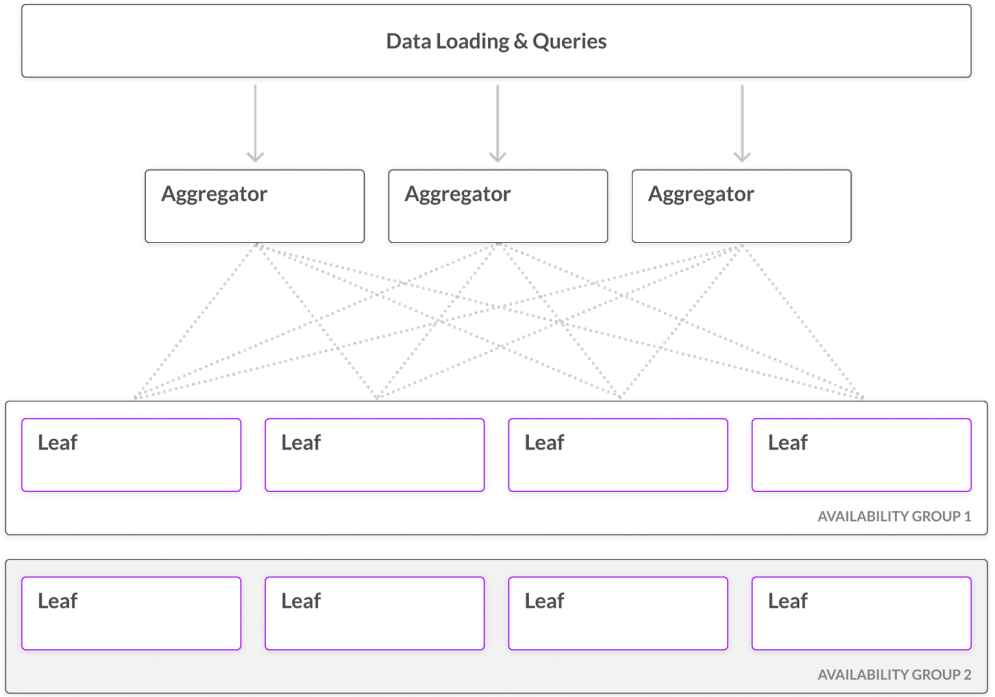


- SAS SpeedyStore는 다수의 데이터 웨어하우스, 데이터 레이크와 운영 데이터 영역 등을 단일 플랫폼으로 통합함.
- 저장이후의 후속 분석 등의 작업이 동일한 데이터를 기반으로 수행되어, 목적에 따라 데이터를 중복 저장할 필요가 없음

분산 및 병렬 처리 Architecture

데이터 저장 및 처리(Leaf Node)와 연산(Aggregator)이 분리된 구조 각 구성 요소는 분산 및 병렬 처리를 수행함.

- **Aggregator 선형적 확장**
  - 고성능 병렬 OLTP 과 OLAP
  - 고성능의 데이터 수집 / 처리
  - 사용자의 DML과 DDL 처리
- **Leaf 선형적 확장**
  - 고용량 데이터 저장
  - 고성능 ELT
  - Agg.에서 요청한 데이터 처리



클라우드 네이티브 기반 MPP 환경

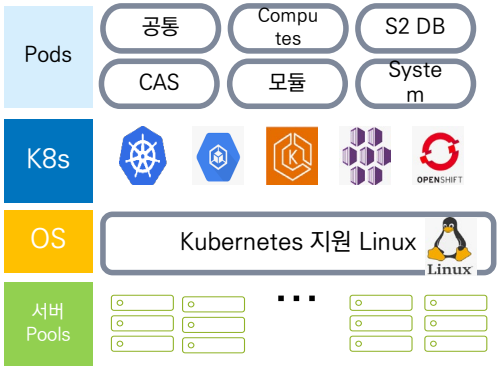
SAS Viya와 SingleStore는 클라우드 네이티브를 지향하는 microservice 기반 설계로 다양한 환경에 이식성과 호환성을 제공합니다.

내용

클라우드 네이티브 지향 설계

- 서비스별 독립 지향인 마이크로서비스 설계
- 서비스는 K8s의 Pod에서 실행
- 특정 서비스 오류의 타 서비스 영향 최소화
- 클라우드 기반의 유연성 및 확장성 담보
- 배포 및 실행이 용이

실행 환경



- 다수의 Nodes로 구성된 Kubernetes(K8s)에서 구동
- 오픈소스, Openshift 및 Google, Azure, Amazon 등의 K8s 환경 지원
- SpeedyStore의 모든 서비스는 Pods에서 실행

주요 특징

- K8s 기반의 고가용성 및 확장성 제공
- 각 서비스의 독립적인 실행으로 오류 영향 최소화
- 서비스 간의 공유 자원 최소 및 제로화로 동시 병렬 처리 가능
- 자원의 활용도에 따른 탄력적 확장 및 축소 가능

장점

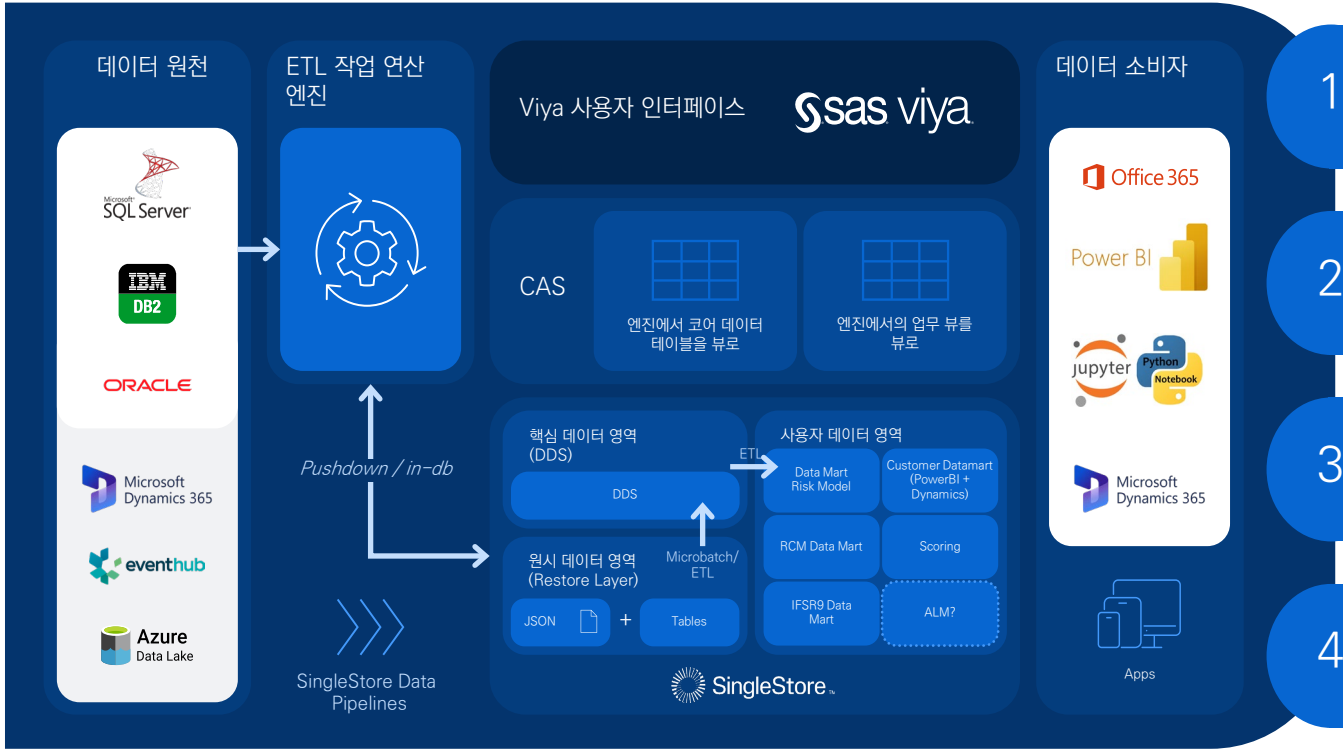
- K8s 기반의 고가용성 및 확장성 제공
- SW의 업그레이드 용이
- 자원의 탄력성으로 비용 효율적인 운영 환경 제공

개선 요인에 대한 업무 영향

개선 요인		업무의 영향	
1	빅데이터 분석	1	보다 빠르고 정확한 의사 결정
2	데이터 이동 축소	2	비용 감소(부분적인 클라우드 활용)
3	분석의 최적화	3	문제에 대한 신속하고 효과적인 분석 제공
4	실시간 의사 결정	4	이동 중인 데이터에 대한 실시간 분석

활용 예시 : VA에서의 모든 데이터 활용

단일 및 클라우드 데이터 플랫폼과의 통합 배포 옵션이 제공



1. SingleStore내에 모든 데이터 저장 –(10TB –100TB –1 PB ~)
2. Viya에서 사용 가능 하도록 CAS로 데이터 등록 (실직적으로 데이터 중복 없이 SingleStore의 View에 연결)
3. 데이터 조회 시, 질의가 SingleStore로 전송되어 실행
4. 레포트에서 사용되는 데이터만 CAS 세션 메모리에 저장

활용 예시 : SAS 코드의 성능 가속

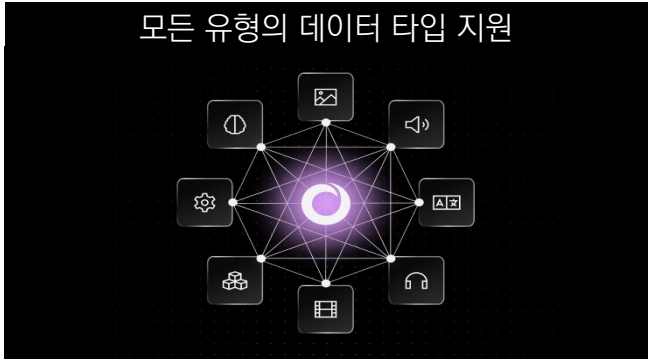
SAS 코드의 수정없이 성능 개선이 가능함.

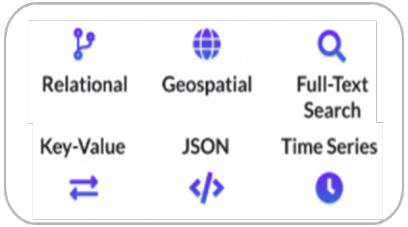
장점

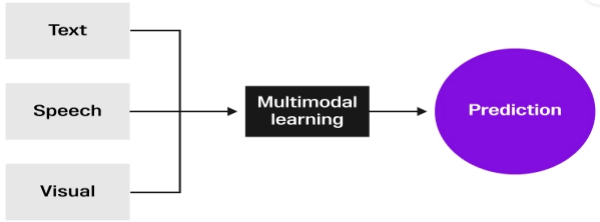
- 기존 SAS 코드를 최적화 하여, SAS Viya에 전달할 내용의 일부로써 SingleStore에서 실행
- Studio Flows은 이관의 일부로 SQL 푸시 다운을 할 수 있도록 최신화
- SQL은 Singlestore로 푸시 다운되어 초고속의 SingleStore의 질의 처리 엔진을 활용하여 수행하여 Viya로의 최소한의 데이터 이동을 제공함
- SingleStore 내장된 SAS EP(Embeds Process)에서 SAS Data 스텝이 고성능으로 처리됨
- 멀티 스레드 방식의 데이터 읽기를 활용하여 더욱 빠르게 데이터를 제공함

활용 예시 : JSON, Key-Value, 시계열 및 Vector 지원

다양한 유형의 데이터 타입을 지원함.





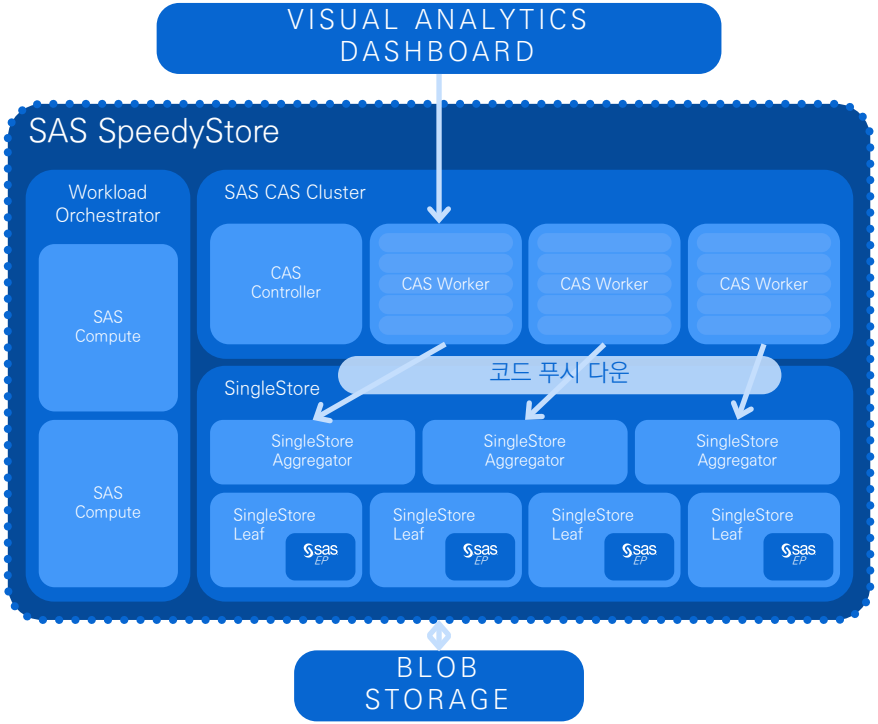


장점

- 모든 다양한 유형별로 존재하는 데이터베이스들을 하나의 데이터 베이스로 통합하여 가능
- 하나의 DB에서 모든 데이터 조회 가능
- 데이터 파이프라인 단순화 및 최소화
- 개발 절차 및 기간 단축
- 다양한 데이터 유형을 손쉽게 결합하여 멀티모달 모델 및 어플리케이션 구축 가능

활용 예시 : Big Data에 대한 동시 사용자 확대

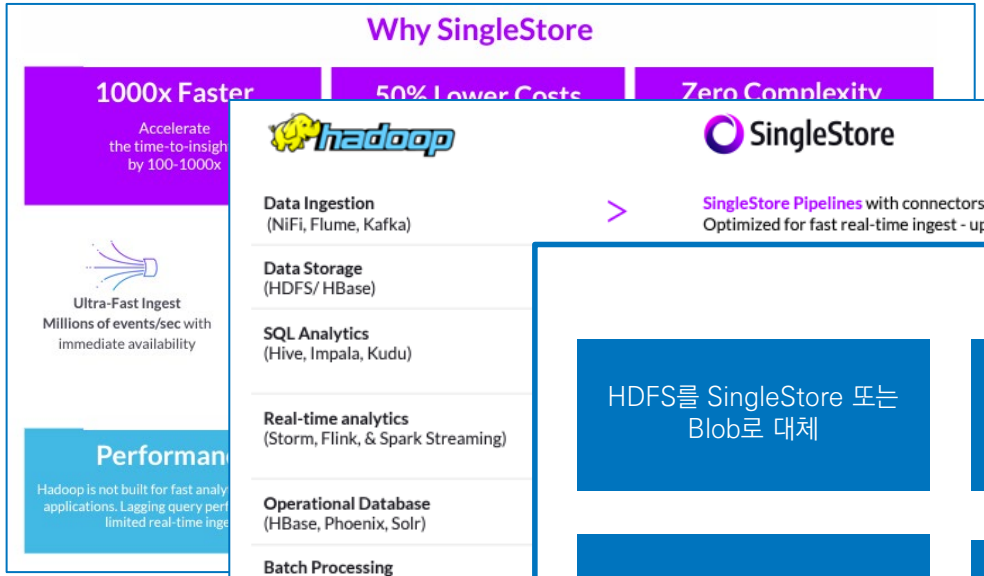
다양한 유형의 데이터 타입을 지원함.



1. SingleStore는 밀리 초 단위의 질의 응답 속도와 뛰어난 확장성을 제공
2. SAS Visual Analytics는 단순 작업을 SQL로 변환하여 SingleStore 푸시
3. 비 표준 SQL의 작업은 SingleStore에 내장된 SAS의 EP에서 처리
4. 복잡한 다중 패스 연산을 위해 데이터는 CAS 인메모리 엔진에 캐시되어 사용

활용 예시 : Hadoop 최신화

Data Lake로 활용되는 Hadoop을 하나의 단일 DB에서 통합하여 활용



해결 방안

- HDFS를 SingleStore 또는 Blob로 대체
- Kafka를 통해 SingleStore 수집
- SingleStore에서 spark 실행
- 기타 다른 작업을 SingleStore로 이동
- 모든 데이터 유형을 위한 효율적인 데이터 저장소
- DB에서 SAS Viya 활용가능

활용 예시 : 실시간 데이터 어플리케이션, 대시보드와 처리

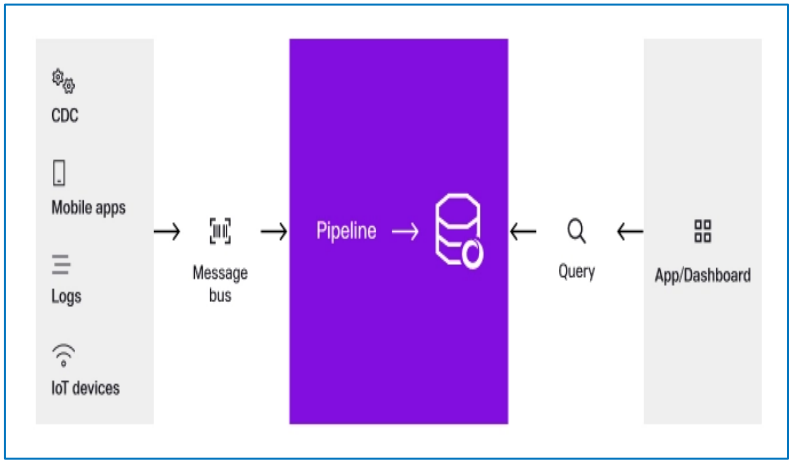
다양한 유형의 데이터 타입을 지원함.

SAS VA에서 실시간 데이터 수집 및 활용

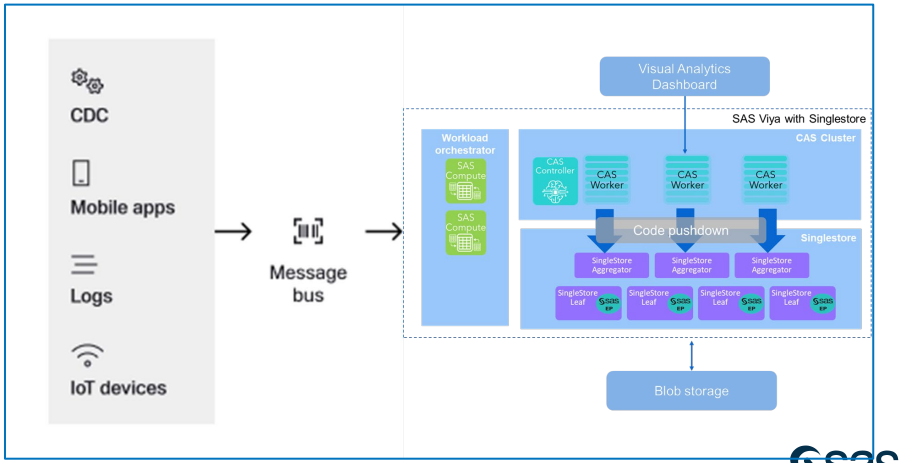
빅 데이터와 실 시간 조합 가능 별도의 실시간 구조 불필요

구조적 데이터 및 JSON과 같은 비구조적 데이터에 동시 작업 가능

SingleStore 참조 구조



SAS SpeedyStore



SAS SpeedyStore의 IT와 Biz의 장점

Business 관점	CDS(*)의 신속한 양성 및 활용 으로 **분석의 생산성 향상** 신속하고 체계적인 모델 적용으로 **즉각적인 Biz 결과 확보** 통합된 모델 주기 관리로 지속인 **모델 개선 및 업무 개선** 즉각적인 **Biz 개선 효과의 모니터링**
IT 관점	Data Layer와 AI/ML 분석 Layer의 **신속한 통합 구축** Data 및 AI/ML Layer의 **유연한 확장성** **고성능 분석** 및 **데이터 처리 제공**

재무적 관점

Platform **구축 비용 절감**

AI/ML **개발/적용 비용 절감**

신속한 적용으로 **이익 극대화**

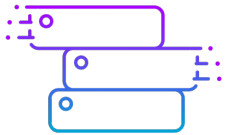
(*) CDS(Citizen Data Scientist) : 시민데이터과학자

별첨 : DB 주요 특징

단일 거래 및 분석용 질의 동시 지원

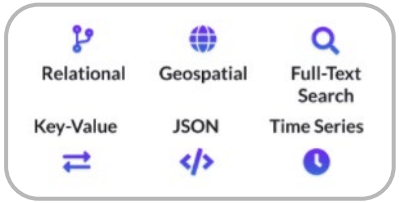
ERP 등을 위한 단일 거래 처리 및 OLAP 등의 분석을 위한 질의 패턴을 하나의 DB에서 동시에 수용하며, 다양한 유형의 데이터를 저장 할 수 있는 범용 데이터 저장소를 제공함.

단일 거래 처리(OLTP)



운영 DB

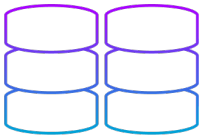
신속한 참조 | 높은 동시 처리 능력



고 병렬 처리 능력을 갖춘 대용량 및 동적인 데이터에 대한 신속한 분석 질의 실행

범용 데이터 저장소

분석 질의 작업(OLAP)

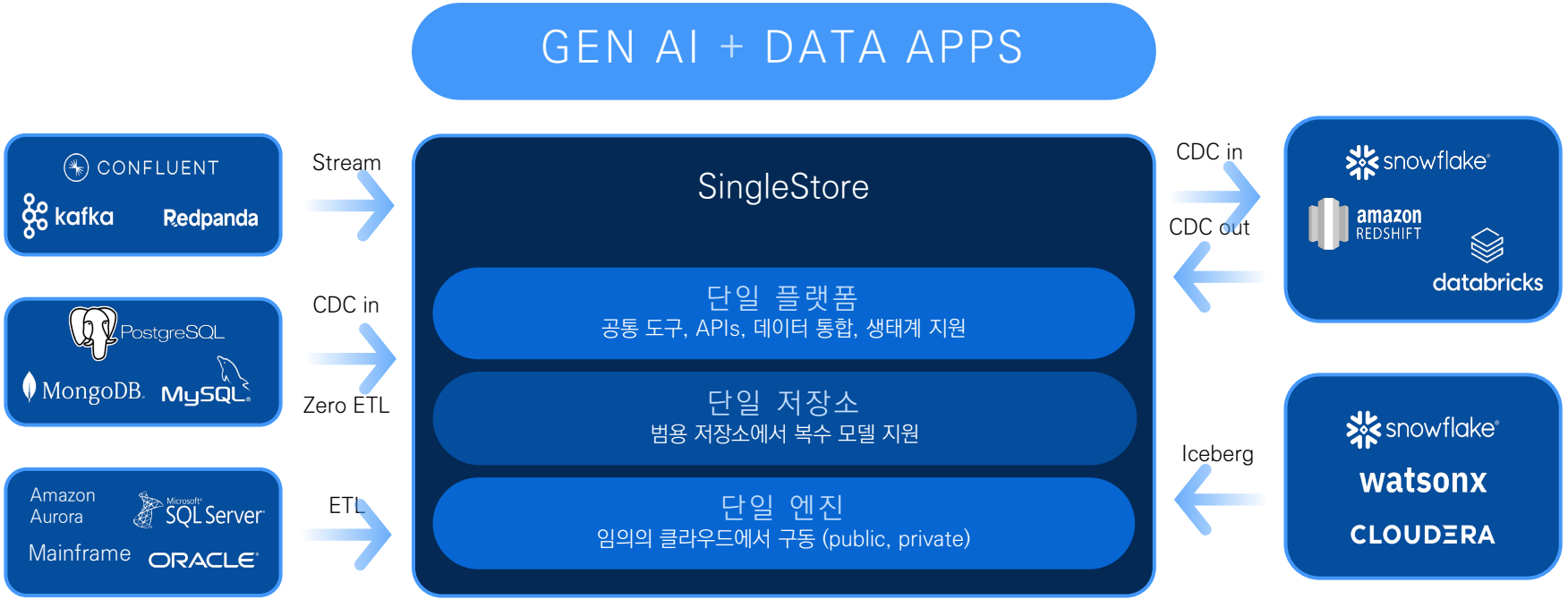


Data Warehouse

- 신속한 질의 처리
- 대량의 데이터 집계

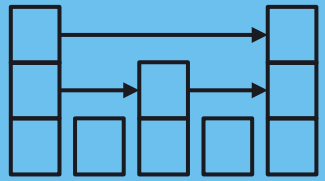
데이터 어플리케이션 개발을 위한 실시간 통합 데이터 플랫폼

생성형 AI와 데이터 어플을 위한 단일 플랫폼, 저장소 및 엔진 제공

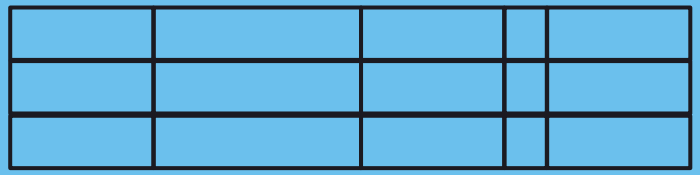


저장 영역별 데이터 관리

Memory, SSD 및 Object Storage별 저장 데이터 유형 분리

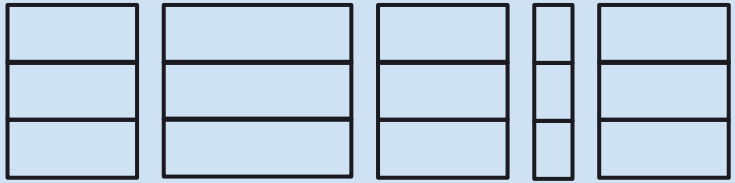


Skip-List Index

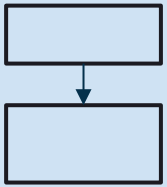


Row-Store

MEMORY

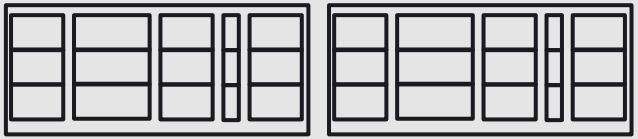


Column Store

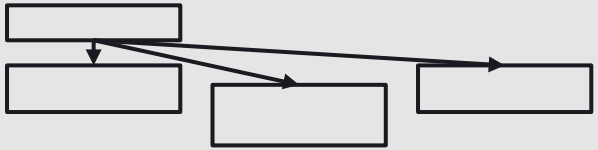


Transaction Log

LOCAL SSD



Blob Store



Transaction Log

OBJECT STORAGE

Columns Stores

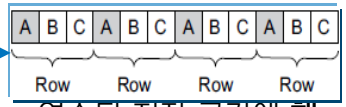
분석 패턴의 SQL에 최적의 성능을 제공하는 Columns Store의 저장 방식의 Table을 제공함.

데이터 저장 방식

테이블

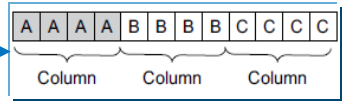
	col1	col2	col3
row1	A	B	C
row2	A	B	C
row3	A	B	C
row4	A	B	C

- 행 기반 저장



– 연속된 저장 공간에 **행 기반** 저장

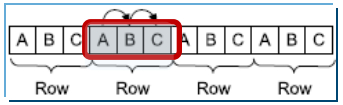
- 열 기반 저장



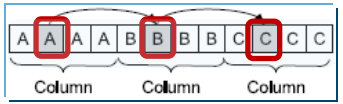
– 연속된 저장 공간에 **열 기반** 저장

운영 업무 SQL 패턴

SELECT * FROM T1 WHERE row2



– 조건에 맞는 영역의 데이터만 읽음
– 결과에 필요한 저장 영역만 접근

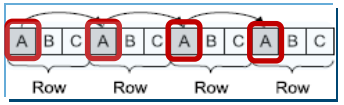


– 모든 영역에 접근하여 필요한 필드 선택
– 결과를 생성하기 위해 불필요한 영역도 접근하여 처리 성능 저하

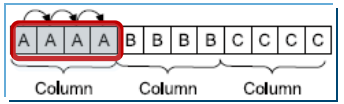
접근 영역 결과선택 영역

분석 업무 SQL 패턴

SELECT SUM(col1) FROM T1;



– 불필요한 영역도 접근하여 필요영역 선택 후 SUM 연산
– 불필요한 영역의 접근으로 조회 성능 저하



– 필요한 영역만 접근하여 연산 실행
– 불필요한 영역의 접근이 없어 처리 성능 개선

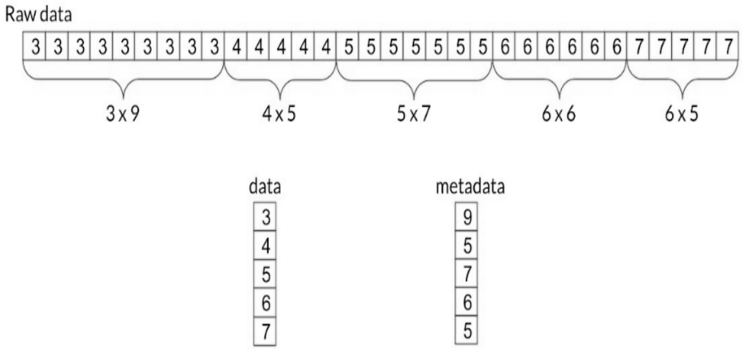
데이터 압축

분석 패턴의 SQL에 최적의 성능을 제공하는 Columns Store의 저장 방식의 Table을 제공함.

Column 기반 저장

압축 기법

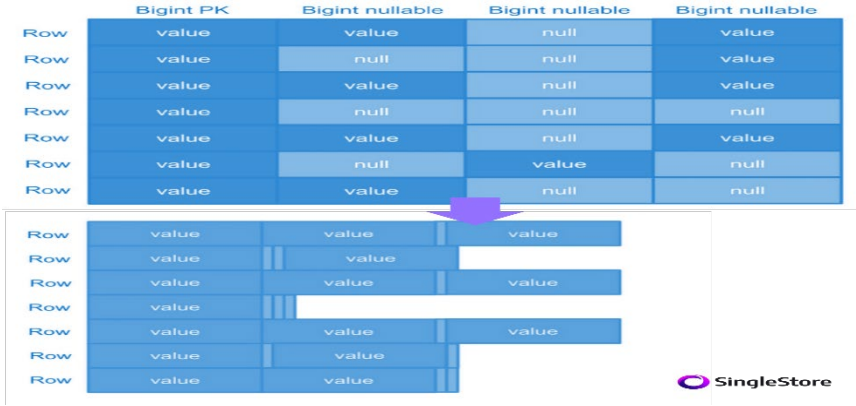
- Run one encoding Compression



Row 기반 저장

압축 기법

- Sparse Compression

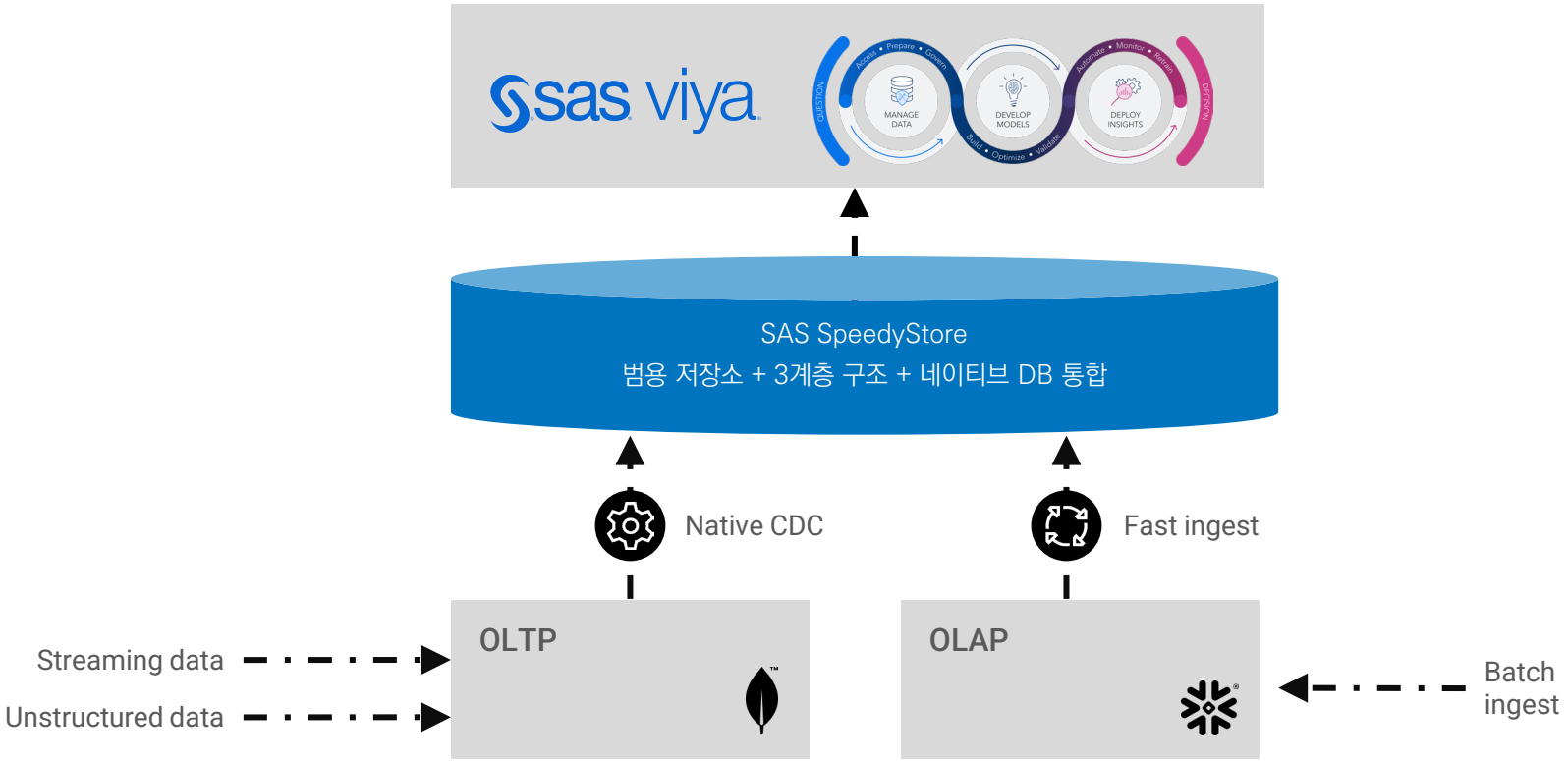


기대효과

- Column 저장 방식의 경우 70 ~ 97% 정도의 압축률
- 데이터 용량 절감 및 관련 비용 절감
- 데이터 처리 성능 향상

신속한 분석 질의 실행과 최소한의 ETL Pipeline

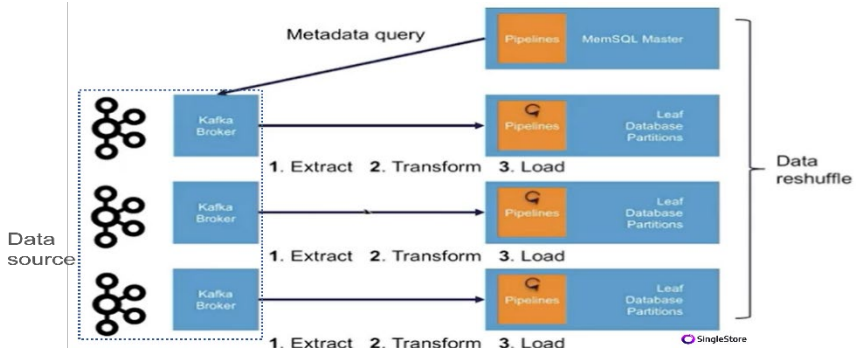
SAS SpeedyStore에 데이터를 적재하는 Pipeline 기능 제공



Pipeline

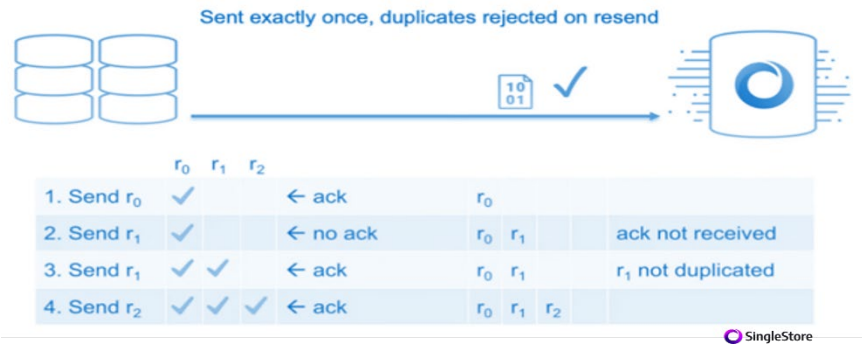
Data의 지속적인 Ingest 기능을 내장하여 근 실시간의 적재, 조회 및 분석을 할 수 있음.

Data Ingest 구성



- 기본 ETL 처리
- 병렬 적재
- 문맥적 1 회 적재
- 다양한 원천 및 형식 지원

Ingest 흐름



지원 데이터 원천

- Apache Kafka
- Amazon S3
- Filesystem Extractor
- Azure Blob
- HDFS
- Google Cloud Storage

지원 File 유형

- JSON
- Avro
- Parquet
- CSV

Data Sharding

데이터를 특정 필드를 기준으로 동일한 값이 하나의 파티션에 왜곡없이 분산 저장을 하는 것으로 실행하는 질의에 따라 단일 파티션만 접근함.

Shard Key 선택 기준

- 높은 cardinality 또는 중복이 없는 필드
- Group By에서 자주 사용하는 필드
- 테이블당 1 개 정의 가능

장점

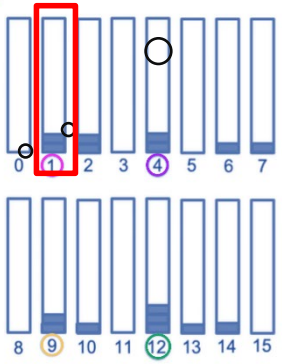
- 물리적 I/O의 최소화로 질의 성능 개선

customer_id 를 Shard Key로 정의

동일한 값을 갖는 레코드를 동일한 파티션에 저장

```
CREATE TABLE orders(
      order_id INT,
      customer_id INT,
      order_date DATETIME,
      status VARCHAR(255),
      SHARD KEY(customer_id)
);

SELECT order_id, customer_id, status
      FROM orders
      WHERE customer_id = xxxxx
```



orderId	customerId	orderDate	status
0	14	3/3/2020	shipped
1	12	3/3/2020	shipped
2	20	3/3/2020	shipped
3	20	3/4/2020	shipped
4	1	3/4/2020	shipped
5	6	3/4/2020	shipped
6	10	3/4/2020	shipped
7	9	3/4/2020	shipped
8	12	3/4/2020	shipped
9	7	3/4/2020	shipped
10	13	3/4/2020	shipped
11	2	3/4/2020	shipped
12	1	3/5/2020	pending
13	2	3/5/2020	shipped
14	9	3/5/2020	shipped
15	12	3/5/2020	shipped

xxxx가 1인 경우 저장된 파티션만 접근

SingleStore

End Of Document