# AdSp Lv2 요약정리

* 데이터 분석기획
- 어떠한 목표를 달성하기 위해 어떠한 데이터를 가지고 어떤 방식으로 수행할지에 대한 일련의 계획
1) 수학&통계학적 지식
2) 정보기술(IT기술, 해킹기술, 통신기술 등)
3) 비즈니스에 대한 이해와 전문성(Domain Knowledge)

*성공적인 분석기획을 위한 고려요소
1) 관련/가용 데이터 파악
2) 이행 장애요소 관리
3) 비즈니스 케이스 확보

** 데이터 분석 대상과 방법

분석의 대상

| 분석의 방법 | ○ | × |
|---|---|---|
| ○ | 최적화 (optimization) | 인사이트 (Insight) |
| × | 솔루션 (Solution) | 발견 (Discovery) |

* 기업의 합리적 의사결정의 장애요소
1) 고정관념(Stereo type)
2) 편향된 생각(Bias)
3) 프레이밍 효과: 동일한 사건을 두고도 개인의 판단이 달라질 수 있는 현상

*ROI(투자자본 수익률, Return Of Investment)
- 투자비용요소(3V): Volume, Variety, Velocity
- 비즈니스 효과(4V): Value

Volume
Variety
Velocity

Value

*데이터 분석업무 주체에 따른 3가지 유형

1) 집중구조: 전사 분석 업무를 별도 분석 전담 조직(COE)에서 담당
- 현업 업무부서의 분석 업무와 이중화/이원화 가능성이 높음
2) 기능구조: 일반적인 분석 수행 구조, 별도 분석조직이 없고 해당 업무부서에서 분석 수행
3) 분산구조: 분석조직 인력들을 현업부서로 직접 배치하여 분석 업무 수행
- 분석 결과에 따른 신속한 Action 가능
- 베스트 프랙티스 공유 가능하며, 전사 차원의 우선순위 수행
- 부서 분석 업무와 역할 분담을 명확히 해야 함
* 객관식/단답식 문제 기출

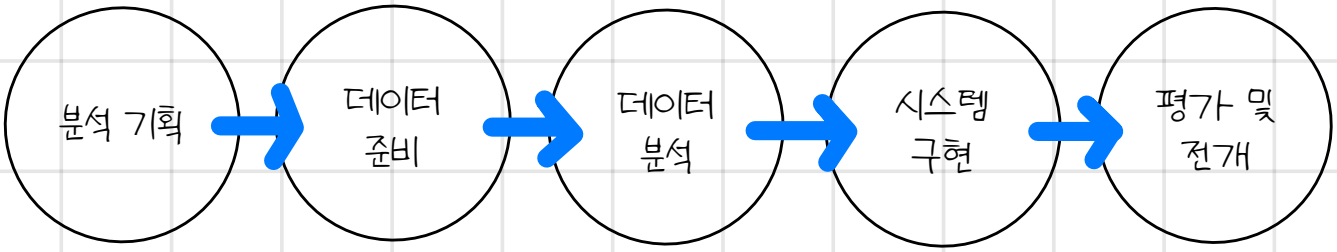
*분석 방법론의 구성 요소 (상방도렘)
1) 상세한 절차
2) 방법
3) 도구와 기법
4) 템플릿과 산출물

* 폭포수 모델
- 단계별로 철저한 검토와 승인 과정을 거쳐 확실히 매듭짓고 다음 단계로 진행하는 모델
- 하향식(Top Down)으로 진행되지만 문제나 개선사항이 발견되면 전 단계로 돌아가는 피드백 수행과정
* 나선형(Spiral) 모델
- 여러 번의 개발 과정을 거쳐 점진적으로 프로젝트를 완성해가는 모델
- 처음 시도하는 프로젝트 적용에 용이
* 프로토타입 모델
- 사용자 요구사항이나 데이터를 정확히 규정하기 어렵고, 데이터 소스도 파악이 힘들 때 일단 분석해보고 그 결과를 보면서 반복적으로 개선해나가는 방법
- 폭포수 모델의 피드백에 대한 어려움을 보완하기 위해 프로토타이핑 제작과 평가 추가

* [KDD] 분석 방법론은 1996년 Fayyad가 체계적으로 정리한 데이터 마이닝 프로세스

* CRISP-DM 분석 방법론
1) 업무이해: 업무 목적 파악, 상황 파악, 데이터 마이닝 목표설정, 프로젝트 계획 수립
2) 데이터 이해: 초기 데이터 수집, 데이터 기술 분석, 데이터 탐색, 데이터 품질 확인
3) 데이터 준비: 분석용 데이터 세트 선택, 데이터 정제, 데이터 통합, 데이터 포맷팅
4) 모델링: 모델링 기법 선택, 모델 테스트 계획 설계, 모델 작성, 모델 평가
5) 평가: 분석 결과평가, 모델링 과정 평가, 모델 적용성 평가
6) 전개: 전개 계획 수립, 모니터링과 유지보수 계획 수립, 프로젝트 종료 보고서 작성, 프로젝트 검토

* 빅데이터 분석 방법론: 계층적 프로세스 모델
-단계(Phrase), 태스크(Task), 스텝(Step)으로 구성




- 비즈니스 이해 및 범위 설정
- 프로젝트 정의 및 계획수립
- 프로젝트 위험 계획 수립

- 필요 데이터 정의
- 데이터 스토어 설계
- 데이터 수집 및 적합성 점검

- 분석용 데이터 준비
- 텍스트 분석
- 탐색적 분석
- 모델링
- 모델 평가 및 검증

- 설계 및 구현
- 시스템 테스트 및 운영

- 모델 발전 계획
- 프로젝트 평가 보고
- 평가 및 전개

* 추가 데이터 필요시
데이터준비~데이터분석 반복

* 프로젝트 위험 계획 수립시 위험에 대한 대응방안(회전수완)
- 회피: 발생원인을 제거
- 전가: 제3자에게 이전, 보험, 보증
- 수용: 실제 발생 시 대응, 리스크가 발생하기 전까지 어떠한 조치도 취하지 않음
- 완화: 용인 가능한 임계치까지 관리

단답형 기출

* 모델링: 분석용 데이터를 이용한 가설 설정을 통해 통계 모델을 만들거나 기계학습을 이용한 데이터 분류, 예측, 군집 등의 가능을 수행하는 모델을 만드는 과정
- 모델을 가동중인 운영시스템에 적용하기 위해서는 모델에 대한 상세 알고리즘 설명서가 필요하며, 필요시 [의사코드]수준의 상세 작성이 필요함.

* 디자인 씽킹: "디자이너처럼 생각하자", 시작 단계에서 대상을 자세히 관찰하고, 그 상황이나 대상에 공감함으로써 많은 가능성과 아이디어를 생각
-> 상향식 접근 방식(발산) + 하향식 접근 방식(수렴) 을 동시에 적용하면서 프로토타이핑과 피드백을 통해 발전

* 하향식 접근법(Top-Down)
- 한계: 문제의 구조가 분명하고 문제를 해결하는 시도에는 적합하나, 새로운 문제의 탐색에 한계가 있음
순서: 문제 탐색 -> 문제 정의 -> 솔루션 탐색 -> 타당성 연구 단답식 기출
(탐정솔타)

* 문제탐색
- 비즈니스 모델 캔버스 기반 탐색 5가지 요소(업제고규지)
: 업무, 제품, 고객, 규제와 감사, 지원 인프라
- 비즈니스 모델기반 탐색/분석 유즈케이스/외부 참조모델 기반 탐색 3가지로 구성

* 혁신적 관점(분석기회 발굴 확장)
1) 거시적 관점(STEEP)
- 사회 영역(Society).
- 기술 영역(Technology).
- 경제 영역(Economy)
- 환경 영역(Environment)
- 정치 영역(Political)

2) 경쟁자 확대 관점
- 대체제 영역
- 경쟁자 영역
- 신규진입자 영역

혁신적 관점
시장 니즈 관점(고채영)
고객 채널 영향자

A/S는 고객영역에 해당
3) 시장의 니즈 관점(비즈니스 모델 캔버스), [고채영]
- 고객 영역, 채널 영역, 영향자들 영역
4) 역량의 재해석 관점
- 내부 역량 영역, 파트너와 네트워크 영역

* Pool: 평상시 지속적 조사와 데이터 분석을 통한 가치 발굴 사례를 정리
-> 향후 과제 발굴 및 탐색시 빠르고 의미있는 분석 기회 도출 가능

** 타당성 검토 단계: 경제적 타당성/ 데이터 타당성/ 기술적 타당성(경데기)

* 목표 시점별 기획방안
-당면한 분석 주제 해결: Speed&Test / Quick&Win / Problem Solving
- 지속적 분석문화 내재화: Accuracy&Deploy / Long Term View / Problem Definition

* 프로토타이핑 프로세스: 사용자가 요구사항이나 데이터를 정확히 규정하기 어렵고, 데이터 소스도 파악이 힘든 상황에서 일단 분석을 시도해보고 결과를 확인하며 반복적으로 개선해나가는 과정
- 신속하게 해결책이나 모형을 제시하는 [상향식 접근 방법]

* 분석프로젝트 관리 방안 주요 5가지(데데스분정)
- 데이터 크기(Size)
- 데이터 복잡성(Data Complexity)
- 스피드
- 분석 복잡성
- 정확도 및 예측

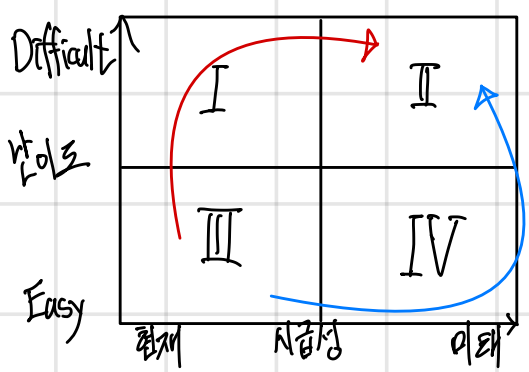
* 분석 프로젝트 영역별 주요 관리항목(QCD, 범통조자리의이)
- 범위 / 시간 / 원가 / 품질 / 통합 / 조달 / 자원 / 리스크 / 의사소통 / 이해관계자
- 시간: 데이터 분석프로세스는 시간이 소요될 수 있기 때문에 품질이 보증된다는 전제 하에 타임방식 기법으로 일정 관리함 (철저한 통제 아님!!)

* 서비스화(Servitization): 제품의 서비스화와 서비스의 상품화를 모두 포함하는 결합 비즈니스 모델, 국내에서는 렌탈이라는 대표적 예시가 있음

* 마스터 플랜 수립 프레임 워크

| 우선순위 고려요소 | 적용 범위/방식 고려 요소 |
|---|---|
| - 전략적 중요도 (시급성) | - 업무 내재화 적용 수준 |
| - 비즈니스 성과/ROI | - 분석 데이터 적용 수준 |
| - 실행 용이성 | - 기술 적용 수준 |

* ISP(정보전략계획, Information Strategy Plan)
- 정보기술을 전략적으로 활용하기 위해 조직 내외부 환경을 분석하여 기회나 문제점을 도출하고 사용자의 요구사항을 분석하여 시스템 구축 우선순위를 결정하는 중장기 마스터 플랜 수립 절차

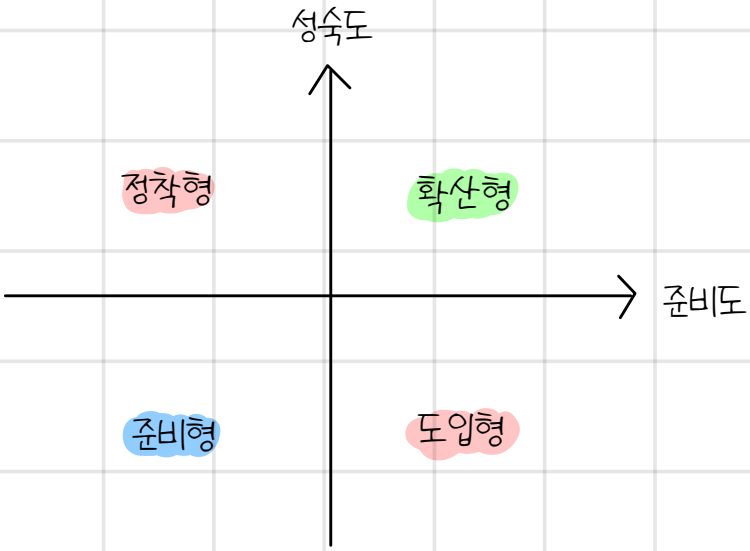



난이도 고려시: Ⅲ→Ⅰ→Ⅱ
시급성 고려시: Ⅲ→Ⅳ→Ⅱ

* 분석 거버넌스 체계 구성요소
- 조직
- 과제기획 및 운영프로세스
- 분석 관련 시스템(IT시스템&프로그램)
- 데이터(데이터 거버넌스)
- 분석 관련 교육 및 마인드 육성체계

* 분석준비도 6개영역
1) 분석 업무 파악
- 발생한 사실 여부 파악
- 예측 분석업무
- 시뮬레이션 분석업무
- 최적화 분석업무
- 분석업무 정기적 개선
2) 인력 및 조직
- 분석전문가 직무 존재
- 분석 전문가 교육 훈련 시스템
- 관리자들의 기본적 분석 능력
- 전사 분석업무 총괄조직 존재
- 경영진 분석업무 이해 능력
3) 분석기법
- 업무별 적합한 분석기법 사용
- 분석업무 도입 방법론
- 분석기법 라이브러리
- 분석기법 효과성 평가
- 분석기법 정기적 개선
4) 분석데이터
- 분석업무를 위한 데이터 충분성/신뢰성/적시성
- 비구조적 데이터관리
- 외부 데이터 활용 체계
- 기준데이터 관리
5) 분석문화
- 사실에 근거한 의사결정
- 관리자의 데이터 중시
- 회의 등에서 데이터 활용
- 경영진의 직관보다 데이터
- 데이터 공유 및 협업문화
6)IT인프라
- 운영시스템 데이터 통합
- EAI, ETL등 데이터 유통체계
- 분석 전용 서버 및 스토리지
- 빅데이터 분석 환경
- 통계 분석 환경/업무
- 비주얼 분석 환경

* CMMI: 능력성숙도 평가모델, 프로젝트를 하는 능력이 얼마나 성숙되었는지 평가하는 모델(분석 성숙도 진단)




* 데이터 거버넌스: 전사 차원의 모든 데이터에 대해 정책 및 지침, 표준화, 운영조직 및 책임 등의 표준화된 관리체계를 수립하고 운영을 위한 프레임워크/저장소를 구축하는 것
- 마스터 데이터/메타 데이터/데이터 사전 등을 관리
- 데이터 가용성/유용성/통합성/보안성/안정성 확보 가능
- 3요소: 원칙/조직/프로세스

* 데이터 거버넌스 체계요소
1) 데이터 표준화: 데이터 표준 용어 설명, 메타 데이터 구축, 데이터 사전 구축 등
2) 데이터 관리체계: 메타 데이터와 데이터 사전의 관리 원칙 수립
3) 데이터 저장소관리: 메타데이터 및 표준데이터 관리를 위한 전사 차원의 저장소 구성
- 저장소는 워크플로우 및 관리용 소프트웨어를 지원하고 관리 대상 시스템과의 인터페이스를 통한 통제가 필요, 데이터 구조 변경에 따른 사전영향평가도 수행해야 효율적
4) 표준화 활동

* 마스터 데이터: 기업의 핵심 데이터인 기준정보를 생성하고, 이를 일관성있게 유지하며 비즈니스 프로세스흐름에 맞춰 정확하게 관리하기 위한 솔루션
* 데이터 사전: DBMS를 효율적으로 사용하기 위해 저장된 정보를 요약하는 것

# CRISP-DM 분석절차에서 "위대한 실패"가 발생하는 구간은?
-> Evaluation - Business Understanding

## ETL: Extraction, Transformation, Load

- ETL의 기능

1. Extraction(추출): 하나 또는 다수의 데이터 원천에서 데이터 획득
2. 변형: 데이터 전처리, 표준화 등
3. 적재: 변형이 완료된 데이터를 특정 목표 시스템에 탑재

- ETL 순서

0. Interface: 데이터 원천으로부터 데이터 얻기위해 인터페이스 메커니즘 구현
1. Staging ETL: 계획에 따라 데이터 원천으로부터 데이터 획득 후 스테이징 테이블에 저장
2. Profiling ETL: 스테이징 테이블에서 데이터 특성 식별 및 품질 측정
3. Cleansing ETL: 데이터 보정
4. Integration ETL: 데이터 충돌 해결, 전처리된 데이터 통합
5. Denormarlize ETL: 데이터 적재를 위해 데이터 비정규화 진행

## ODS: Operational Data Store

ODS는 ETL과정을 통해 정제한 데이터를 저장한 데이터베이스
데이터 웨어하우스: ODS를 통해 통합된 데이터가 데이터 분석 및 보고서 생성을 위해 적재되는 데이터 저장소

*데이터 웨어하우스 특징

- 주제 중심성, 영속성(비휘발성), 통합성, 시계열성

* 데이터 웨어하우스의 테이블 모델링 기법

1) 스타 스키마(조인 스키마)

: 가장 단순하며, 복잡도가 낮아 이해하기 쉽고 쿼리 작성 용이, 조인테이블 개수 적음

하지만, 차원 테이블이 비정규화에 따른 데이터 중복으로 테이블로 데이터 적재시 시간 많이 소요됨
하나의 사실 테이블 중심으로 다수의 차원 테이블로 구성

2) 스노우 플레이크 스키마

: 스타 스키마의 차원 테이블을 제 3정규형으로 정규화한 형태
: 데이터 중복이 제거되어 적재 속도 빠르나 복잡성이 올라가 조인 테이블 개수가 증가하고 쿼리 작성 난이도가 올라감

ODS vs DW(데이터 웨어하우스)

요즘은 ODS를 단순히 DW 구축을 위한 1차 데이터 수집공간 정도로 생각하기도 함

| | ODS | DW |
|---|---|---|
| 데이터 내용 | 현재 또는 최신 데이터 | 이력, 현재, 요약 등의 데이터 |
| 데이터 양 | 소규모 | 대규모 |
| 데이터 갱신 | 지속 갱신, 현재 DB만 반영 | 데이터 추적 보관 |

* CDC(Change Data Capture)
: DB에 변경점이 생기면 이를 캡처하고 데이터 웨어하우스로 전달하는 기술 혹은 이를 자동
화하는 기술
1) Time Stamp on Rows: 타임스탬프 컬럼으로 최근 데이터인지 구분
2) Version Numbers on Rows: 버전 컬럼(혹은 참조 테이블)으로 구분
3) Status on Rows: 타임스탬프나 버전넘버기법의 보완 용도로 활용, 데이터 변경 여부를 True/False Boolean 값으로 구분
4) Triggers on Tables: 데이터베이스 트리거를 활용해 사전 등록된 다수 대상에 변경 데이터를 배포하는 형태, 잘못쓰면 유지보수성 저하되서 주의해야함
5) Event Programming: 데이터 변경 식별을 앱으로 구현, 앱개발 부담 및 복잡도 증가되나 다양한 조건에 CDC 메커니즘 구현 가능
6) Log Scanner on Database: 대부분 DB에서 제공하는 트랜잭션 로그의 변경 내역을 해석해서 CDC 메커니즘 구현 -> 트랜잭션 무결성 영향도 최소화 가능

* CDC 구현방식
1) Push: Source(원천)에서 변경 식별 후 Target(대상)에 변경 데이터 적재
2) Pull: Target에서 Source를 정기적으로 살펴보고, 필요시 데이터 다운

** 빅데이터는 기계학습, 시각화 등으로 분석함. (일반적으로 OLAP(다차원 분석)을 위주로 분석한다. (X!))

** EAI는 실시간, 근접 실시간 처리 중심 / ETL은 배치(batch) 프로세스 중심

** SQL on 하둡기술
- 아파치 드릴, 아파치 스팅거, 샤크, 아파치 타조, 임팔라, 호크, 프레스토

**ODS 구성을 위한 일괄 작업 ETL 작업 단계 순서
Interface layer -> Staging layer -> profiling layer -> cleansing layer -> integration layer -> export layer
인터페이스: 데이터 수집
스테이징: 데이터 저장
프로파일링: 데이터 품질 검사
클렌징: 데이터 전처리(오류 데이터 수정)
인테그레이션: 전처리 완료된 데이터를 단일 통합 테이블에 적재
익스포트: 통합 데이터를 익스포트 테이블에 적재 (OLAP 비정형 질의에 활용)

** 하둡
맵리듀스(대규모 분산병렬 처리 표준) + HDFS(분산 파일시스템) 플랫폼 기술!
비공유 분산 아키텍처를 사용

- 하둡의 특징

1. 선형적 성능 및 용량 확장: 서버 대수 제한 없고, 연산과 저장 성능이 서버 대수에 비례해 증가함
2. 고장 감내성: HDFS에 저장되는 데이터는 3중복제 -> 서버 장애라도 데이터 유실 방지, 맵리듀스 중 특정 태스크에서 에러 발생시 그 태스크만 다른 서버에서 재실행 가능
3. 핵심 비즈니스 로직에 집중: 맵리듀스는 맵, 리듀스 2가지 함수만 구현하며 동작 -> 데이터가 크고 작음, 시스템 수준의 에러는 하둡이 내부적으로 처리함 (오직 비즈니스 로직에만 집중할 수 있음)
4. 풍부한 에코시스템: zookeeper(모니터링), yarn(하둡2.0 자원 관리 프레임워크), Flume-NG(대표적 로그 데이터 수집 시스템으로, 데이터 발생 애플리케이션 단계, 데이터 수집 단계, 데이터 저장 단계, 데이터 저장소 보관 단계로 이루어지며, 특별한 설정 없을시 하둡이 저장소로 활용됨)

- 하둡 데이터 연동

비정형 데이터 분석시 바로 DB에서 맵리듀스로 처리하면 졸라 오래걸려서, 하둡으로 복사해서 처리하고 결과 요약데이터만 다시 DB에 기록하는데, 대표적인 오픈소스 솔루션이 스쿱(Sqoop) 임.
스쿱은 Import 명령어로 RDBMS데이터를 HDFS로 옮기고, 반대로 Export도 가능

로그: 기업에서 발생하는 대표적 비정형 데이터
로그 데이터 수집 시스템: 아파치 Flume-NG, 페이스북 Scribe, 아파치 Chukwa

- EAI 효과

향후 시스템 개발시 유지 보수비용 절감
고객과 상호 협력 프로세스 연계 발전 기반 확보
그룹 및 지주 계열사들간 상호 관련 데이터 동기화 등 데이터 표준화 기반 확보
* EAI는 단일 접점 허브시스템 이용 중앙 집중식 연결구조
* EAI 구현 유형
1. Mediation(Intra-communication)
: EAI 엔진이 중개자(Broker)로 동작, 이벤트 발생 식별 후 미리 약속된 시스템에 해당 내용 전달 (Publish/Subscribe model)
2. Federation(Inter-communication)
: EAI 엔진이 외부 정보 시스템으로부터 데이터 요청들을 일괄적으로 수령해 필요데이터 전달 (Request/reply model)

하둡 에코시스템 구성(하는 역할 별 어떤 것들이 있는지 보기에 자주 출제)
데이터수집: Flume-NG, kafka
데이터 연동: Sqoop
분산 데이터베이스: Hbase
대용량 SQL 질의: Hive, Pig -> 데이터 웨어하우스
(Pig는 Pig Latin언어를 제공해서 맵리듀스 프로그래밍 대체 가능)
실시간 SQL 질의: Impala, Tajo
워크플로 관리: Oozie, Azkaban
*Chukwa: 분산 환경에서 생성되는 데이터를 HDFS에 안정적으로 저장시키는 플랫폼

** Log Scanner on Database(자주 출제됨)
트랜잭션 로그 스캐닝 및 변경 내역에 대한 해석을 통해 CDC메커니즘 구현
각 DB 관리 시스템에 따라 트랜잭션 로그 관리 메커니즘이 상이해서 다수 이기종 데이터베이스 사용시 작업 규모가 증가되므로 주의
장점: 데이터베이스와 애플리케이션 영향도 최소화, 변경 식별 지연시간 최소화, 트랜잭션 무결성 영향도 최소화, 데이터베이스 스키마 변경 불필요

** 전통 데이터 처리 기법과 빅데이터 처리 기법의 차이점 자주 출제
1. 시각화를 통한 인사이트 도출은 빅데이터 처리 기법의 고유한 장점
2. 둘 다 통계와 데이터마이닝은 함
3. 전통은 SQL이나 RDBMS 쓰고, 빅데이터는 NoSQL이나 초대형 분산 데이터 저장소 사용
4. 전통 방법은 DB 데이터를 ODS로 적재 후 이를 다시 데이터 웨어하우스에 적재

** 대용량 질의 기술
하둡: 저비용으로 대용량 데이터 저장하고 신속 처리 가능, 이전에 비해 단순해 졌지만 여전히 코딩이 필요해서 분석가는 어려워 하는 단점
Hive: 친숙한 SQL이라는 질의 기술을 이용하여 하둡에 저장된 데이터를 쉽게 처리하고 분석할 수 있도록 해줌
* 하둡과 하이브는 대용량 데이터를 배치 처리하는데 최적화, 하지만 실무에서는 데이터를 실시간 조회 및 처리해야할 상황이 많음
--> 이런 제약을 극복하기 위해 실시간 SQL 질의 기술인 SQL on 하둡 등장!

SQL on 하둡
1. 아파치 드릴: 하둡 전문 회사 맵알이 주축, 오픈 소스 버전의 드레멜(드레멜: 구글 개발 대규모 데이터셋을 빠르게 분석하고 쿼리하는 분산형 SQL 쿼리 엔진)
2. 아파치 스팅거: 하둡 전문 회사 호튼웍스 주축, 기존 하이브 코드 이용해서 성능 개선하는 개발 진행
3. 샤크: 인메모리 기반 대용량 데이터웨어하우스, 하이브와 호환됨
4. 아파치 타조: 고려대 대학원에서 시작해서 그루터라는 국내 빅데이터 전문회사가 합류해서 개발 진행 중, 아파치 인큐베이션 프로젝트로 등록되어 있음
5. 임팔라: 하둡 전문 회사 클라우데라에서 개발 주도
6. 호크: EMC에서 분사한 피보탈에서 개발
7. 프레스토: 페이스북 자체 개발 하둡 기반 데이터 웨어하우징 엔진

* EAI vs ESB
EAI: 미들웨어(Hub) 이용 비즈니스 로직 중심 어플리케이션 통합 및 연계, 로직연동은 개별 어플리케이션에서 수행, 단일 접점 허브시스템 이용 중앙 집중식 연결구조
ESB: 미들웨어(Bus) 이용 서비스 중심 시스템 유기적 연계, 로직 연동은 ESB에서 수행, 버스(Bus) 형태 느슨하고 유연한 연결구조

* 플럼(Flume): 소스서버에 에이전트 설치, 에이전트로부터 데이터 전달받는 콜렉터로 구성
* Mahout: 하둡 기반 데이터 마이닝 알고리즘 구현 오픈소스 라이브러리