2️⃣

Chapter 2

2-1 크로스 집계의 기본

트랜잭션 테이블, 크로스 테이블, 피벗 테이블

크로스 테이블

- 행과 열이 교차하는 부분에 숫자 데이터가 들어감
- 데이터 베이스에 새로운 행을 추가하는것은 간단하지만, 열을 늘리는것은 간단하지 않음
- 따라서 데이터는 행 방향으로만 증가하게 하고, 열 방향으로는 데이터를 증가시키지 않도록 해야한다
  →
  트랜젝션 테이블

크로스 집계

- 트랜잭션 테이블에서 크로스 테이블로 변환하는 과정
- 스프레드 시트 피벗테이블이 대표적인 예시

룩업 테이블

- 테이블에 새로운 항목을 추가하는 것이 아니라, 다른 테이블과 결합하고 싶은 경우
- 트랜잭션 테이블과 룩업 테이블은 서로 독립적으로 관리 할 수 있음
- 데이터 분석 용도에 따라 변경해도 상관없음

SQL에 의한 테이블의 집계

- 대량의 데이터를 크로스 집계 → SQL을 사용
- 데이터 집계함수를 이용해 데이터 양 감소를 고려할 수 있다

데이터 집계 → 데이터 마트 → 시각화

- 데이터 마트 : 데이터 집계와 시각화 사이에 있는것

- 데이터 마트가 작다
    - 장점: 시각화가 간단하다
    - 단점: 원래 데이터에 포함된 정보를 잃어버려서 시각화 할게 없다
- 데이터 마트가 크다
    - 단점: 데이터 마트의 거대화로 좋은 시각화를 할 수 없게 된다

2-2 열 지향 스토리지에 의한 고속화

→ 큰 데이터를 빠르게 처리하려면, 미리 데이터를 집계에 적합한 형태로 변환하는것이 필요

데이터 베이스의 지연을 줄이기

데이터를 집계에 적합한 형태로 어떻게 변환하냐면....

1. 대량의 데이터를 처리할 수 있는 데이터 레이크 & 데이터 웨어 하우스에 저장
2. 원하는 데이터를 추출, 데이터 마트 구축.
3. 초 단위의 응답을 얻을 수 있도록 함

데이터 처리의 지연

데이터 마트를 만들때는 가급적 지연이 적은 데이터 베이스가 있어야 하는데, 두가지 옵션이 있다.

1. 모든 데이터를 메모리에 올리기.
   모든 데이터를 메모리에 올리면 대신 RDB 같은 경우 급격히 성능이 저하된다.
2. '압축'과 '분산'에 의해 지연 줄이기
   분산된 데이터를 읽어 들이려면 멀티 코어를 활용하면서 디스크 I/O를 병렬 처리하는것이 효과적

열지향 데이터 베이스 접근

행 지향 데이터 베이스

- 레코드 단위의 읽고 쓰기에 최적화 되어 있음
- 새 레코드를 추가할 때 파일의 끝에 데이터를 쓸 뿐이므로 빠르게 추가
- 대량의 트랜잭션을 지연 없이 처리하기 위해 데이터 추가를 효율적으로 할 수 있도록 함
- 검색을 고속화 하기 위해 인덱스를 사용해야 함

열 지향 데이터 베이스

- 일부 칼럼만이 집계 대상
- 데이터 베이스에서 데이터를 미리 칼럼 단위로 정리 → 필요한 칼럼만을 로드 → 디스크 I/O를 줄인다
- 데이터의 압축 효율 우수

MPP 데이터베이스의 접근 방식

쿼리 지연을 줄일 또 다른 방법은 MPP 아키텍쳐에 의한 데이터 처리 병렬화 인데,

행지향 데이터 베이스에서는 잘 안하고,

- 열 지향 데이터 베이스에서는 많은 양의 데이터를 읽기 떄문에 쿼리 시작이 길어진다.
- 압축된 데이터의 전개 등으로 CPU 리소스를 필요 → 멀티 코어를 활용, 고속화 하는게 좋다
- MPP에서는
    1. 하나의 쿼리를 다수의 작은 태스트로 분해
    2. 가능한 한 병렬로 실행
    3. 각 태스크의 결과를 집계

MPP 데이터 베이스와 대화형 쿼리 엔진

구조상 고속화를 위해 CPU와 디스크 모두를 균형있게 늘려야 한다.

→ 일부는 하드웨어와 소프트웨어가 통합된 제품으로 제공

MPP 데이터 베이스

→ 하드웨어 수준ㅇ서 데이터 집계에 최적화된 데이터 베이스

2-3 애드 훅 분석과 시각화 도구

jupyter notebook에 의한 애드 혹 분석

1. 파이썬과 루비, R언어 등의 스크립트 언어를 실행하는데 사용
2. 노트북 안에서는 파이썬 스크립트와 외부 명령어를 실행

3. 실행 내용은 모두 기록되고 과거로 되돌아가서 편집, 재실행 가능
4. 마크다운 평식으로 주석을 넣을 수 있고, 사진이나 수식을 포함 할 수 있음
5. 시각화 툴로는 대표적으로 matplotlib가 있음

대시보드 분석

Redash

- SQL에 의한 쿼리의 실행 결과를 그대로 시각화
- 하나의 쿼리가 하나 또는 여러 그래프에 대응
- 등록한 쿼리는 정기적으로 실행, 결과가 Redash 자신의 데이터 베이스에 저장
- 구조가 알기 쉽다
- SQL로 쿼리 를 작성
- 그래프 수만큼 쿼리를 실행

Superset

대화형 대시보드를 작성하기 위한 파이썬으로 만든 웹 어플리케이션

- 데이터 집계는 외부 데이터 저장소에 의존 → Druid
- Druid는 집계시 테이블 결합 불가능 → 시각화에 필요한 데이터 미리 모두 결합

Kibana

자바 스크립트로 만들어진 대화식 시각화 도구

Elasticsearch이외의 데이터 소스에는 대응하고 있지 않음

BI도구

- 몇 개월 단위의 장기적인 데이터 추이를 시각화
- 집계의 조건을 세부적으로 바꿀 수 있는 대시보드에 적합

2-4 데이터 마트의 기본 구조

시각화에 적합한 데이터 마트 만들기

다차원 모델의 데이터 구조를 MDX(multidimensional expressions)등의 쿼리 언어로 집계

OLAP 큐브: 데이터 분석을 위해 만들어진 다차원 데이터

OLAP: OLAP cube를 크로스 집계하는것

최근에는 OLAP 큐브를 위해 특별한 구조를 준비하진 않고, BI도구와 MPP 데이터베이스를 조합하여 크로스 집계하는 경우가 많아짐

→ 이미 존재하는 테이블을 그대로 시각화 하려고 하는게 아니라, 만들고 싶은 그래프에 맞추어 '다차원 모델'을 설계

테이블을 비정규화 하기

3장 남았는데...이거 좀 자고 일어나서 할래요

다차원 모델 시각화에 대비하여 테이블을 추상화하기