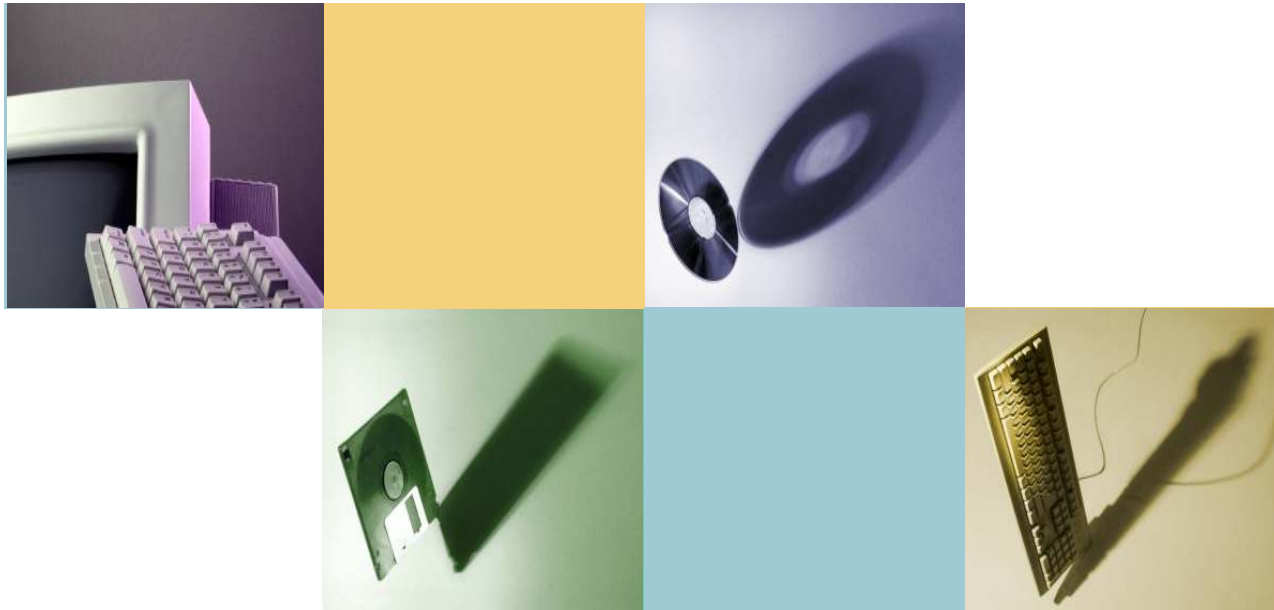


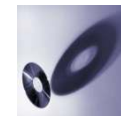
# 인터넷 비즈니스



## 제10장. Data Warehouse

# Data Warehouse의 출현배경

- 경쟁의 심화로 고급정보에 대한 요구 증대 - Demand
  - 기업 실적평가 정보
  - 시계열로 변화상태를 파악할 수 있는 정보
  - 운영상태를 Monitoring 할 수 있는 정보
  - 비효율적인 상황을 파악하여 비용을 절감할 수 있는 정보
  - 사업기회를 포착할 수 있는 정보
- 전산화에 대한 시각의 변화 - Paradigm Shift
  - 단순 작업의 대행
  - 단위부서의 업무 대행 및 정보관리
  - 전사적인 정보의 통합관리
  - 정보의 능동적인 활용



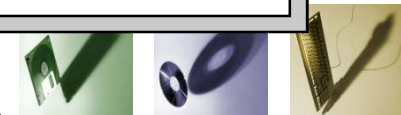
# Data Warehouse의 출현배경

- 정보 기술의 발전 - **Enabler**
  - Hardware 속도 개선
  - DBMS 기능의 향상
  - 통신 속도 개선
  - Replication 기능 도입
  - Access 기능의 다양화
- Operational과 Informational의 분리 필요성 대두 - **Need**
  - Operational System의 구조가 Informational의 요구를 충족시키기에는 부적합
  - Informational에 대한 처리 증가로 Operational

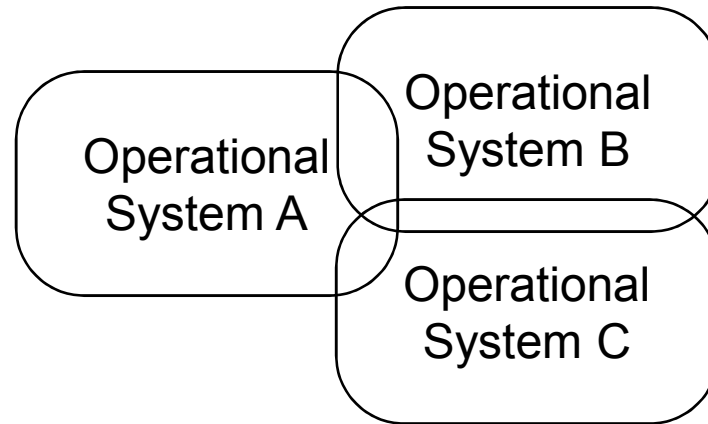


# Operational System과 Informational System이 분리되어야 하는 이유

구 분	Operational	Informational
Data 습득	Real Time	Delay
Data의 보관대상	현 시점의 Data 위주	시계열 정보
Data의 Granularity	상세	기본 및 요약
Data의 사용 범위	해당거래 Data에 국한	여러 부문의 종합 정보
Data의 필요시점	수초 이내로 급박	수분/수시간/수일 여유
Data의 신선도	항상 현상태 유지 필요	수분/수시간/수일 지연가능
Data의 주 사용자	일선 작업자	관리자/업무전문가
Data 사용방법	정형화된 CRUD	정형화/비정형화된 조회
DB의 형태	HDB/RDB	RDB/MDB



# Operational System의 현재 모습



- 문제점
  - 정보의 일관성 결여 (정보의 중복관리)
  - 관리자가 필요한 종합정보 제공시 문제 발생 (정보관리 형태가 서로 상이)
- 해결책
  - Operational System을 통합 정보시스템으로 재개발 (실패 가능성이 높음)
    - 프로젝트의 범위가 너무 커서 위험요인이 산재해 있음
    - Operational 과 Informational을 동시에 만족하는 정보구조 도출이 어려움
  - 현재의 Operational System을 기반으로 Informational System 구축



# Informational을 구현하는 방법(1)

## 1 Layer

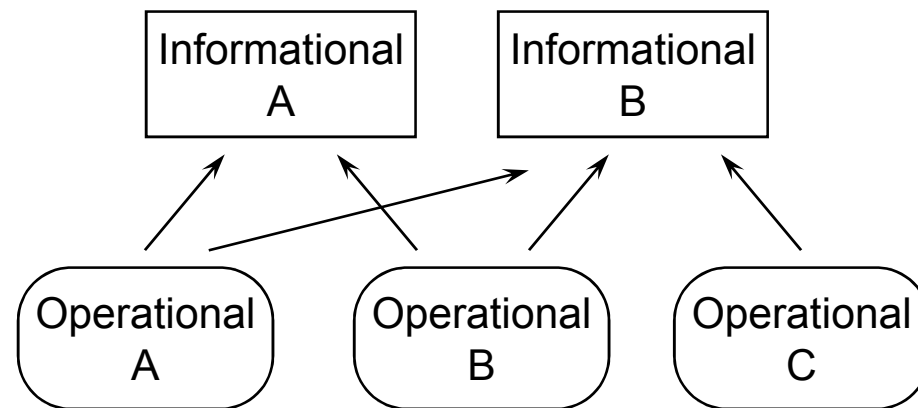
Operational & Informational

- 기업전체의 통합모델로 재개발 (특히 RDBMS 이용시 적합)
- Operational과 Informational의 Contention이 많아 Performance 저하
- System에 부하가 별로 없는 경우 구현 가능
- 시계열 정보의 관리가 용이하지 않음
- 정보 관리비용을 최소화할 수 있음
- 대량의 정보 Access를 피하기 위해서는 집계용 Table 추가 필요



# Informational을 구현하는 방법(2)


## 2 Layer (1 Replication)



- 가장 일반적으로 볼 수 있는 형태 (자연발생적인 형태)
- 현 Operational을 바탕으로 필요시마다 Informational을 추가
- Informational의 Source가 복잡하지 않은 경우 적합
- Operational과 Informational이 동일 환경인 경우 적합 (Replication 용이)



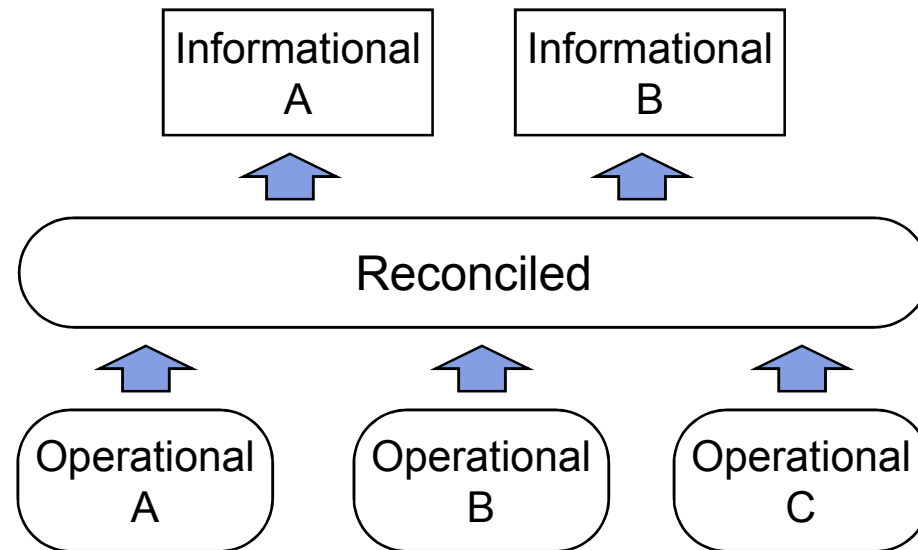
## Informational을 구현하는 방법(2)

- Informational이 부문별로 난립하게 되면 문제점 발생
  - 관리가 복잡해짐
  - Source의 구조 변경시 Informational을 효과적으로 변경할 수 없음
  - Replication이 복잡해지고 시간이 많이 걸림
  - 전산자원의 낭비 (Table수/Program수/중복작업)
  - Informational간의 정보수집 대상/기준/시점이 다를 경우, 동일한 상황에 대하여 서로 상이한 정보를 제공할 수 있음
- Informational을 Source로 사용하는   
Informational이 새기다며 과리가 불가는하 보자하



# Informational을 구현하는 방법(3)

## 3 Layer (2 Replication)

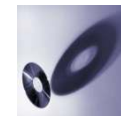


- 기업차원의 통합모델을 **Reconciled**에서 제공 (정보의 일관성 유지)
- **Reconciled**에 보관되는 정보는 시계열로 관리됨
- **Reconciled**를 바탕으로 필요에 따라 **Informational** 구현
- 서로 다른 환경/분산 환경일 경우 우선적으로 고려되어야 하는 구조

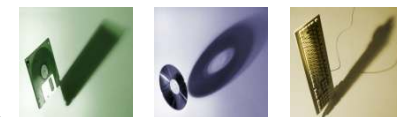
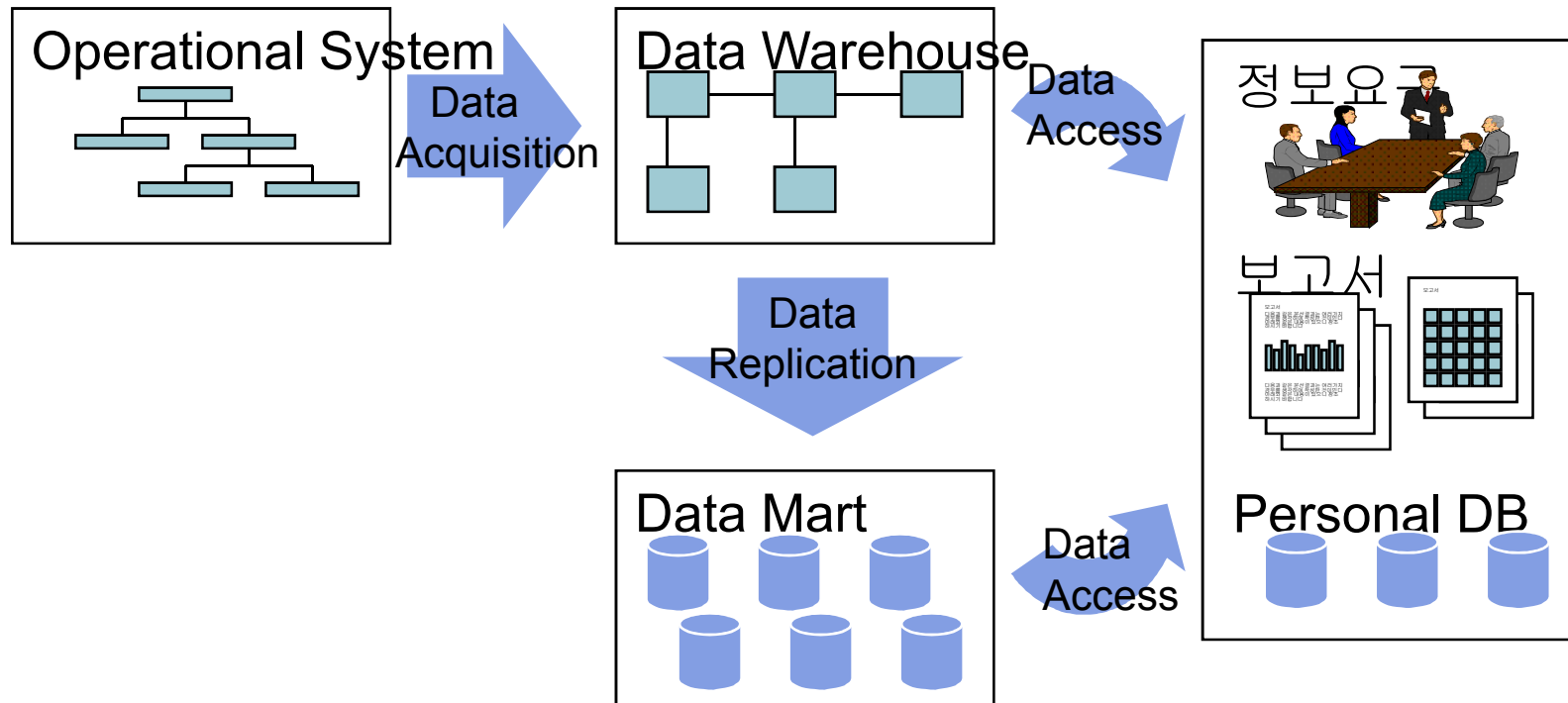


# Informational을 구현하는 방법(3)

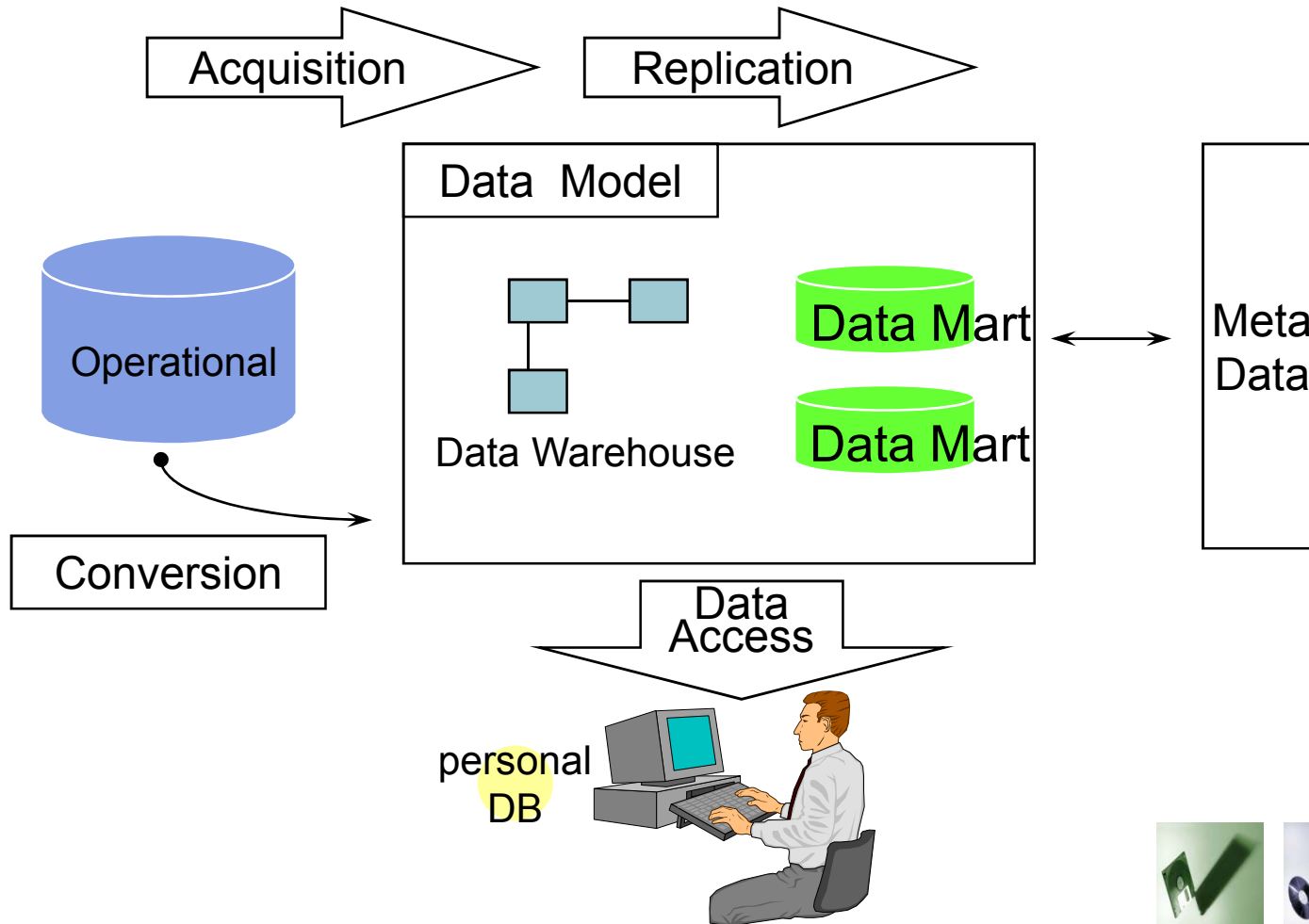
- Reconciled와 Informational을 일원화된 조직에서 체계적으로 관리한다면, 전산자원의 낭비를 최소화할 수 있음
- Reconciled를 구현하는데 상당한 시간이 소요됨
- Reconciled를 Data Warehouse, Informational을 Data Mart라고 함
- 1 Layer와의 차이점
  - Operational을 그대로 유지 (투자비용/Risk)
  - 전산부하를 Operational과 Informational로 분산
  - 시계열 정보 관리
- 2 Layer와의 차이점
  - 정보의 일관성 유지
  - Operational과 Informational간의 Path 최소화



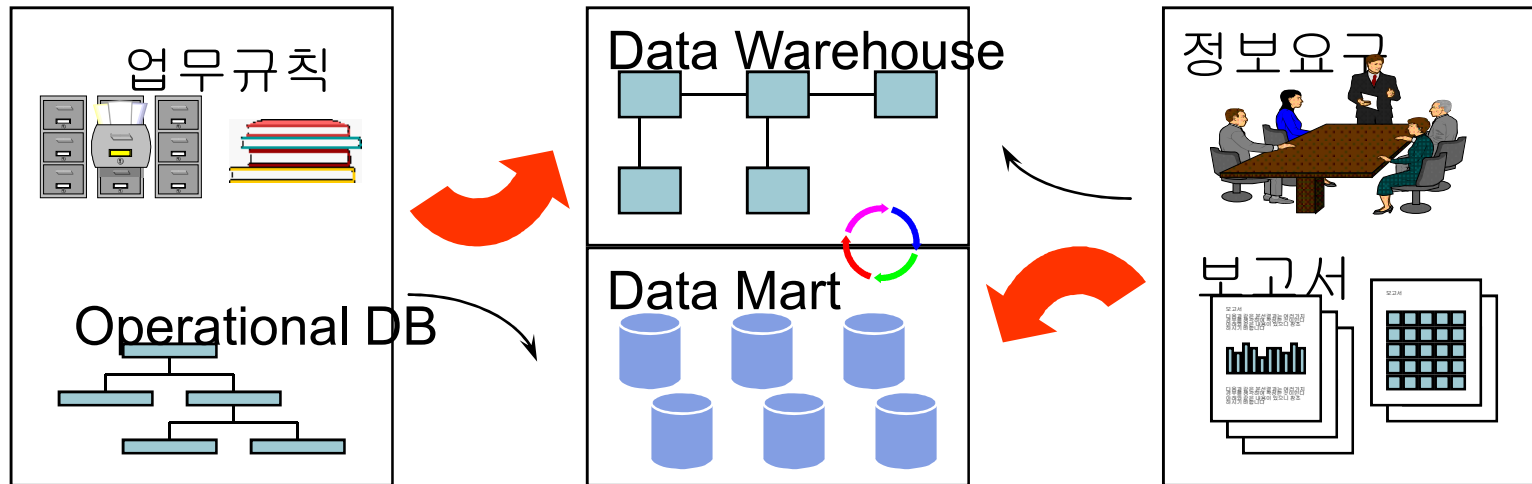
# Data Warehouse에서의 정보 흐름



# Data Warehouse의 구성요소



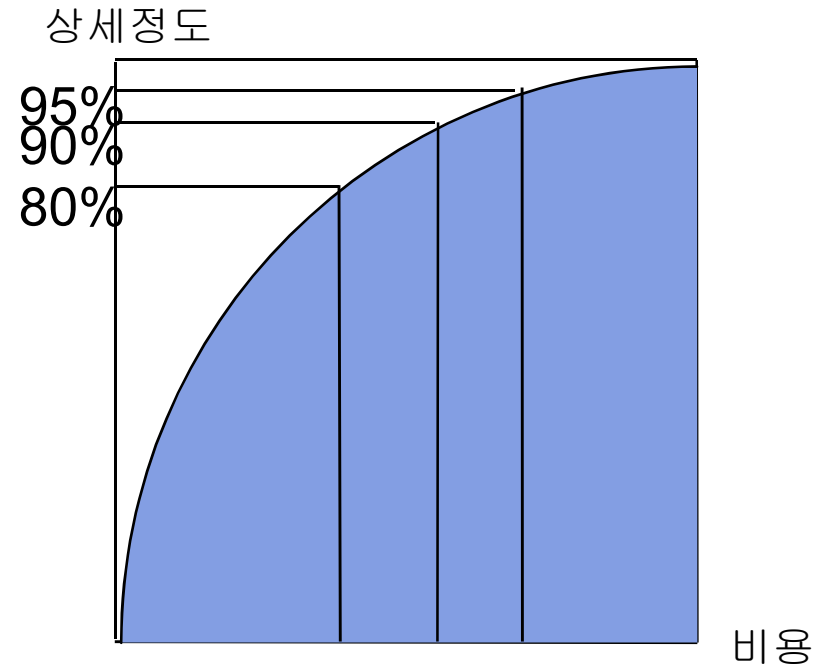
# Data Warehouse의 Data Model



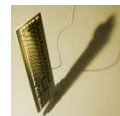
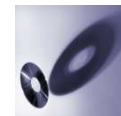
- Data Warehouse Modeling은 다각적인 접근이 필요함
  - 현재의 Operational DB나 업무규칙을 바탕으로 정확한 Business Rule을 추출하여 Data Warehouse Model의 기본 골격을 결정함
  - 정보요구나 보고서를 바탕으로 Data Mart Model을 추출함
  - Data Warehouse Model이 Data Mart Model 및 정보요구를 적절히 지원할 수 있도록 Model을 정비해야 함
- Data Warehouse의 정보요구는 정형화되어 있지 않으므로 어려움이 있음



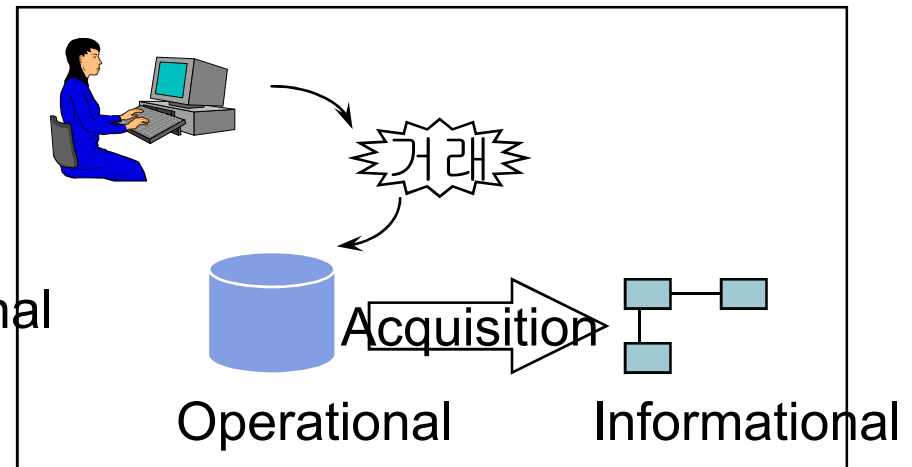
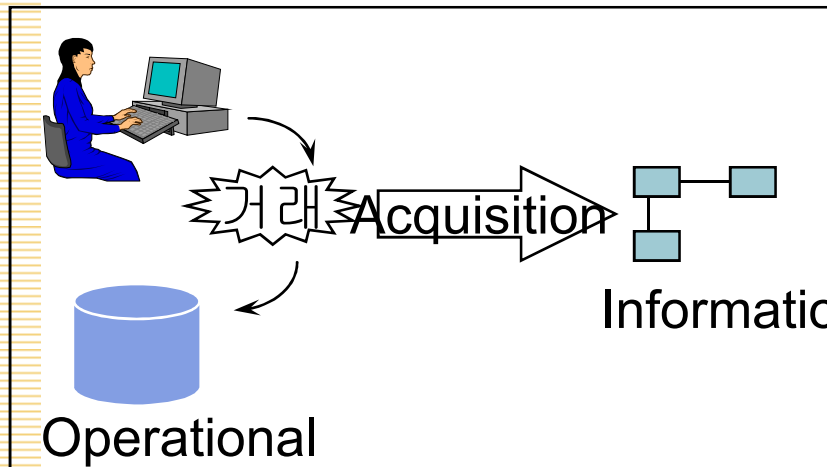
# 정보의 Granularity



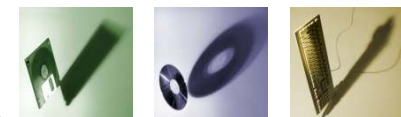
- 상세정도가 낮은 상태에서는 상세정도를 높이는데 별로 비용이 발생치 않음
- 상세정도가 높은 상태에서는 상세정도를 더 높이는데 많은 추가비용 발생
- 상세의 정도는 80-90% 수준이 적당
- Data Warehouse가 100%의 정보를 수용할 필요는 없음



# Data Acquisition 방법

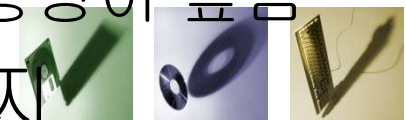


- 거래에서 Data Acquisition하는 방법
  - 거래에서 Real Time으로 Informational DB 갱신
  - 거래사항을 Log로 만들어 Deferred로 Informational DB 갱신
- Operational DB의 갱신사항을 감지하여 Data Acquisition하는 방법
  - Real Time/Deferred
  - DB Log를 이용하는 방법
  - DB Copy



# Data Conversion의 주요 Issue

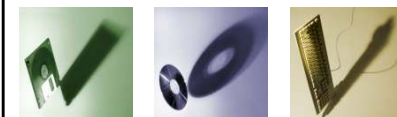
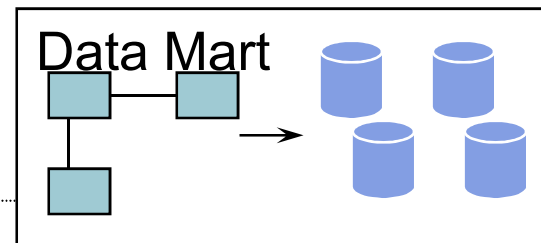
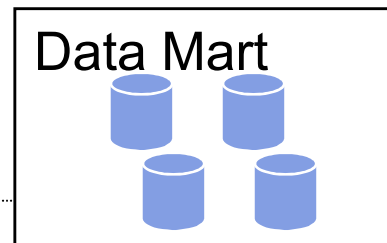
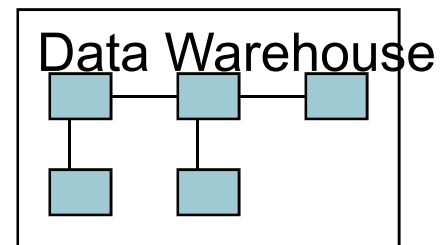
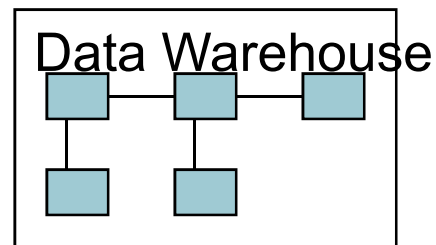
- Operational 정보의 정확성 검증 및 수정 필요
  - Garbage In Garbage Out / Gold In Gold Out
- Conversion 은 많은 시간이 소요됨
  - 대상 DB가 방대하여 Programming에 많은 시간 필요
  - Operational과 Informational의 환경 및 DB 구조가 상이하야 다양한 문제 발생
  - Operational의 비효율적인 Code 체계를 정비하고 변환하여야 함
  - Conversion 대응표 및 Error 유형을 체계적으로 관리하여야 함
  - Conversion 수행시 1일 이상 소요될 가능성이 높음
- Conversion이 정확하게 수행되었는지 검증하는 방법 필요



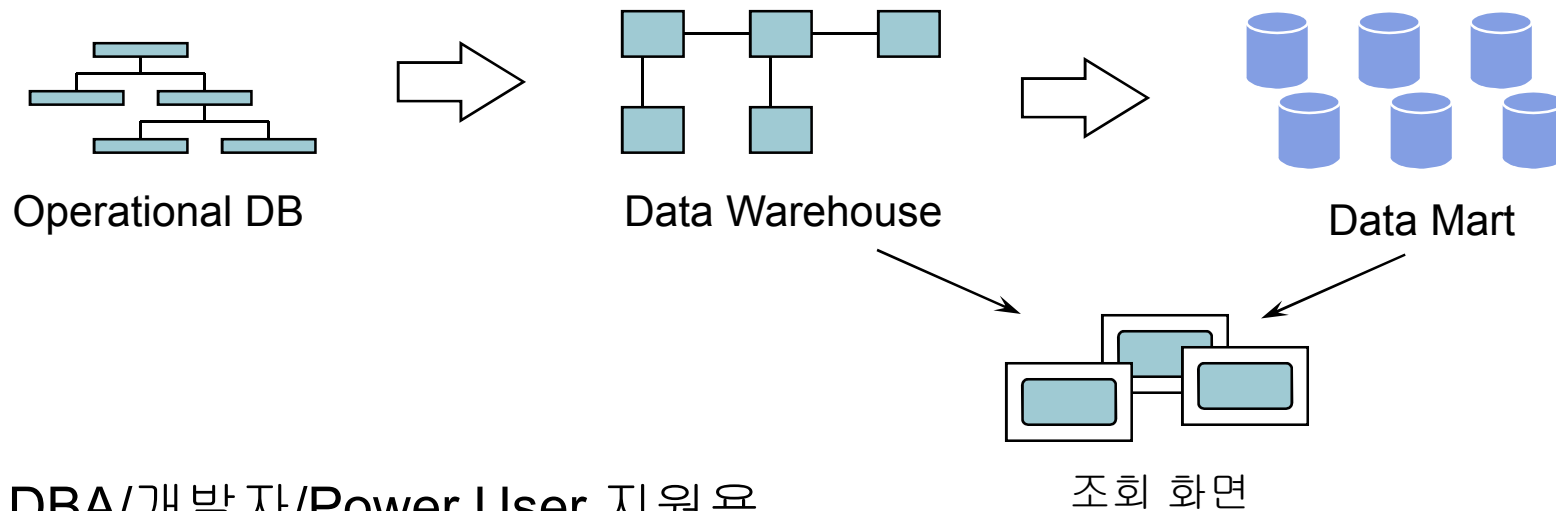


# Data Replication 방법

- Data Warehouse 내용을 변경하지 않고 Replication하는 경우
  - 분산 DB를 위하여 제공되는 DBMS Replication 기능 활용
  - Data Warehouse의 DB를 주기적으로 Snapshot하여 사용
- Data Warehouse 정보를 변형하여 Replication하는 경우
  - Data Warehouse를 갱신하는 TP Monitor가 Data Mart도 갱신
  - Data Warehouse의 정보를 바탕으로 주기적인 Batch 작업 수행



# Data Warehouse Metadata의 기능

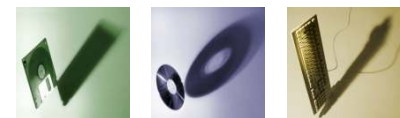


- DBA/개발자/Power User 지원용
- 각 DB 및 화면에 대한 Dictionary 기능
- DB간/DB 화면간 Mapping 정보
  - Acquisition/Conversion/Replication 개발 및 유지보수시 필요
- Data의 량/갱신주기/사용빈도/Authority/보관주기/Backup 주기
- Data구조의 Version 관리



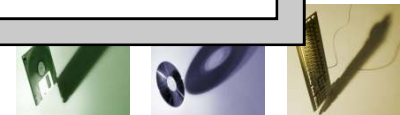
# Data Access

- 웨어하우스 데이터를 액세스 & 분석
  - Query, Reporting, 데이터 분석 도구
  - 관계형 DBMS를 액세스하는 다차원 데이터 분석 도구 (ROLAP)
  - 다차원(Multidimensional) DBMS를 액세스하는 다차원 데이터 분석 도구 (MOLAP)
  - 4GL 또는 Visual programming language를 채택하는 DSS Application 개발 도구



# Data Access 방법 비교

구 분	ROLAP (RDBMS)	MOLAP (MDBMS)
저장공간	적게 소요	많이 소요
업계표준	존재	존재하지 않음
처리용량	대용량 가능	대용량 불가
조회/분석기능	보통	우수
구조 변경	용이	어려움



# Data Warehouse가 적합한 경우

- Informational의 관리가 체계화되어 있지 않은 경우
- Operational의 부하가 심하여 Informational을 수용하기 힘든 경우
- Operational의 형태와 Informational의 형태가 상이한 경우
- Informational을 분산환경으로 구현하고 싶은 경우
- Informational에 대한 요구가 비정형적인 경우
- 다양한 시계열 정보가 필요한 경우
- Operational이 RDB/통합DB가 아닌 경우

