10g

Oracle Technology Day

BI와 통합

-현실적인 문제점과 이의 대응 방안을 중심으로 -

장 성 우
TSC본부 Technical Architecture 팀
한국 오라클

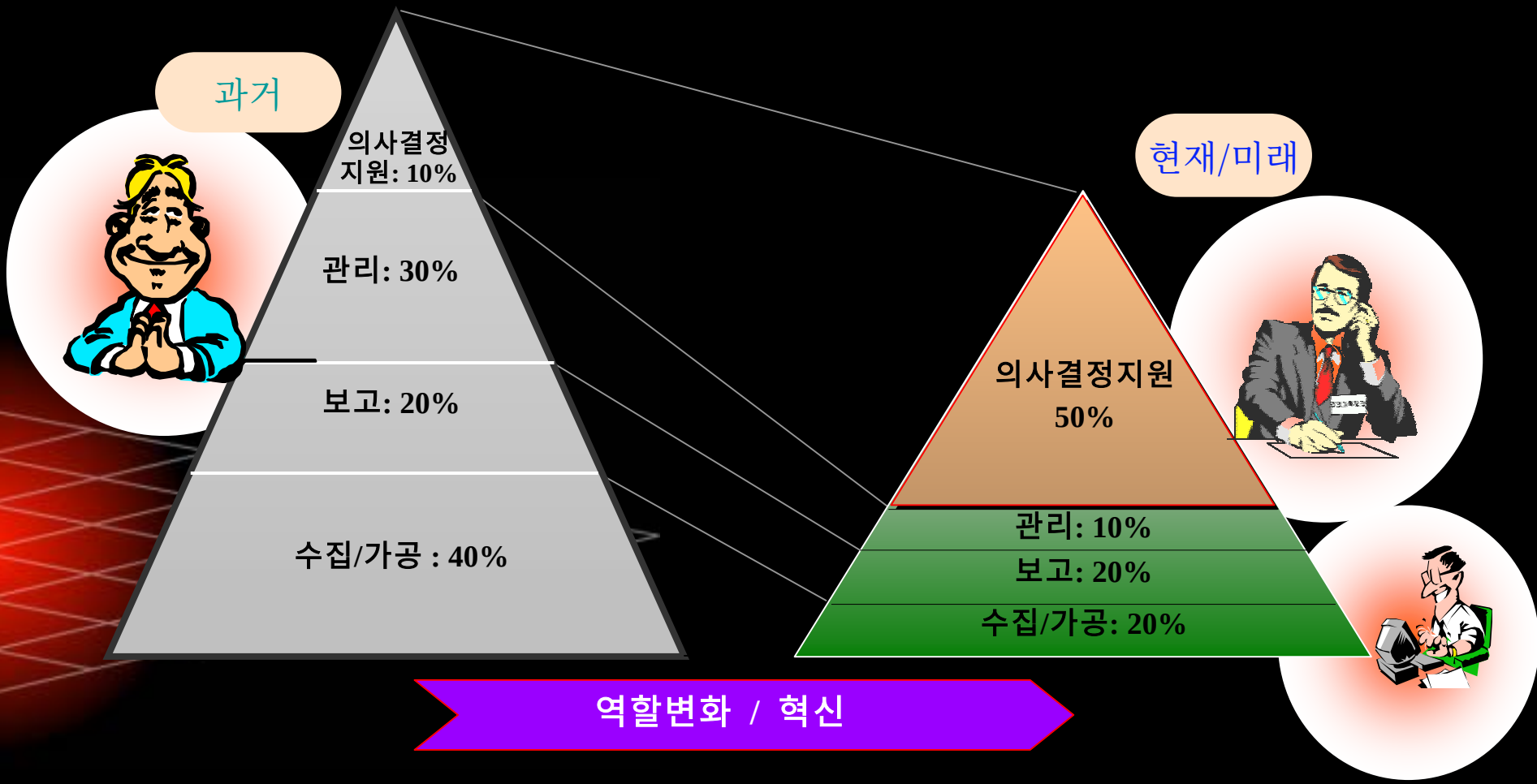
Agenda

- BI(Business Intelligence) 개요
- 현실적인 문제점
- 통합 기반의 BI 추진 방안
  - 전사 통합 데이터 아키텍쳐
  - 통합 방안
- 요약

정보 요구 사항의 변화

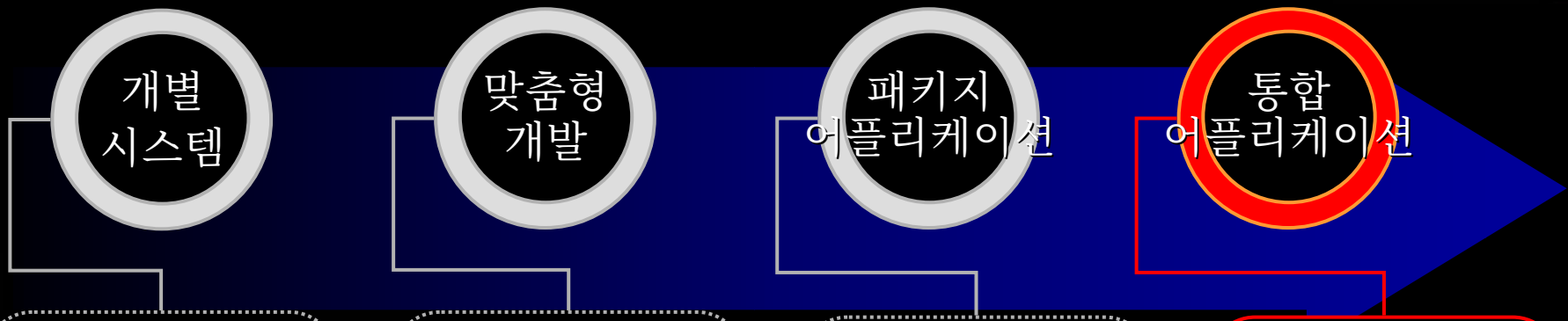
10g

- 새로운 경영환경에 대응하기 위하여 정보요구사항이 다음과 같이 변화되고 있음



IT 시스템의 변화 방향

10g



- 회사의 업무 요구사항에 따라 해당 업무를 처리하기위한 개별 시스템 도입
- 무수히 많은 각각의 시스템들이 운영되며, 각 시스템별 관리자 및 별도의 운영비용, 유지보수의 어려움 발생
- 업무 처리 프로세스가 전체적으로 연계되어 자동화되기 어려움

- IT 담당부서에서 각 사별 업무특성에 맞는 자체 개발 어플리케이션 및 시스템 개발, 운영
- 개발자 담당자 및 운영자 부재시 시스템 유지보수의 어려움
- 장기적인 회사 전략 및 비전을 반영한 설계 및 구현의 어려움

- Global Standard 프로세스 및 Best Practice를 반영한 패키지 형태의 어플리케이션 도입
- 각 회사의 업무에 맞추어 추가적인 개발 및 Customization 발생
- 각 분야별로 Best of Breed의 패키지 어플리케이션 적용으로 인한 통합성 문제 발생
- 통합문제로 인한 추가적인 비용지불

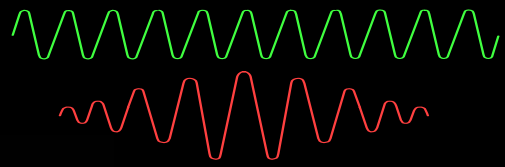
- 분산되어 있는 각각의 시스템과 어플리케이션들을 EAI 등의 도구 및 시스템을 사용하여 통합함으로써 회사의 전체 프로세스가 유기적으로 연계되어 하나의 자동화된 closed-loop 형태로 운영, 관리 될 수 있음
- 패키지 어플리케이션 차원에서의 통합제품과 EAI등을 통한 통합이 존재

ORACLE

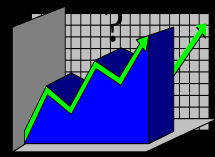
BI(Business Intelligence)란?

BI란 데이터와 정보의 가치를 극대화하는 것

Data



Information : 정제, 정렬, 조합, 결합된 Data



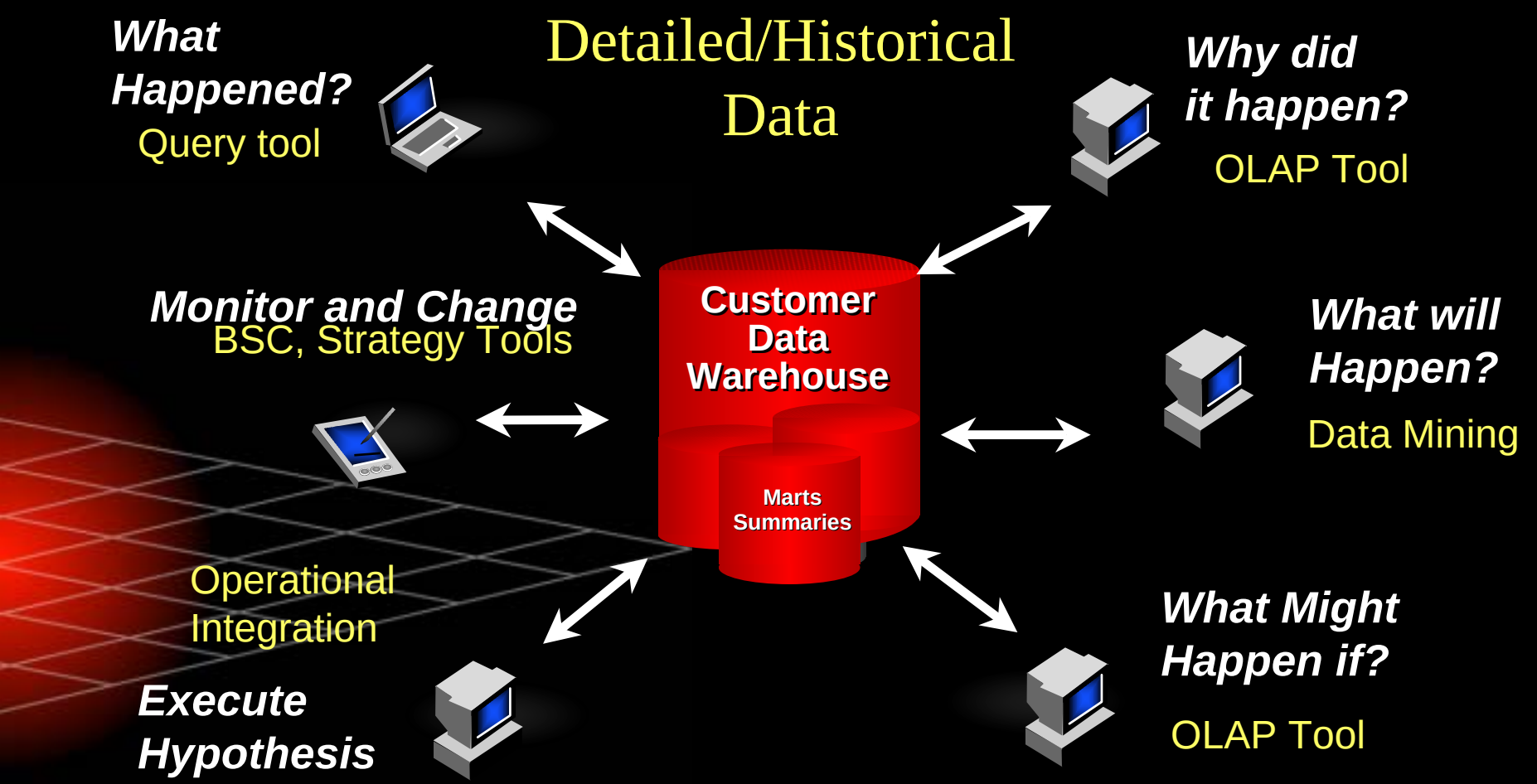
예) 특정 상품 구매자에 대한 성별, 수입별, 지역별 고객리스트

Intelligence : 유기체적인 특징 → 조직 내에서 증식

예) 구매정보를 활용한 마케팅 팀의 프로모션

데이터와 정보를 요약,분석,활용하여
기업 경영에 필요한
의사결정의 질 제고 및 성과향상을
도모하는 일련의 프로세스를 통칭함

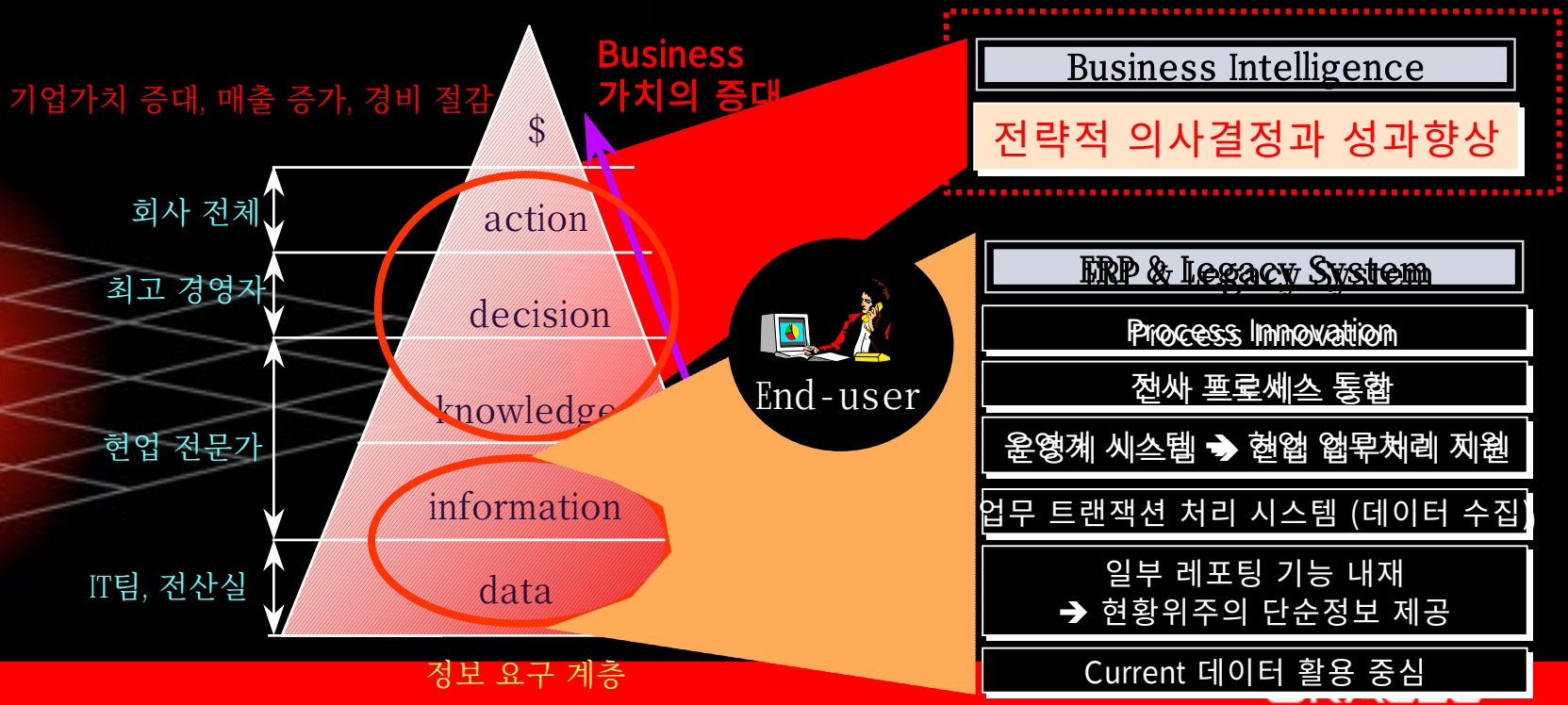
BI는 단일 제품이 아닌 프로세스



BI의 Positioning

■ 기업 정보시스템은 크게 업무 처리의 자동화와 통합을 지원하는 기간시스템(Legacy, ERP)과 이러한 기간시스템으로부터 데이터와 정보를 요약,통합하여 전략적 의사결정과 성과관리를 지원하는 BI 시스템이 있음

기업가치증대는 ERP 또는 기존 시스템으로부터의 데이터 통합에 의한 경영분석 및 성과관리의 효율화로 적시에, 정확한 의사결정과 성과관리를 시스템화 함으로써 가능함



BI 구축 효과

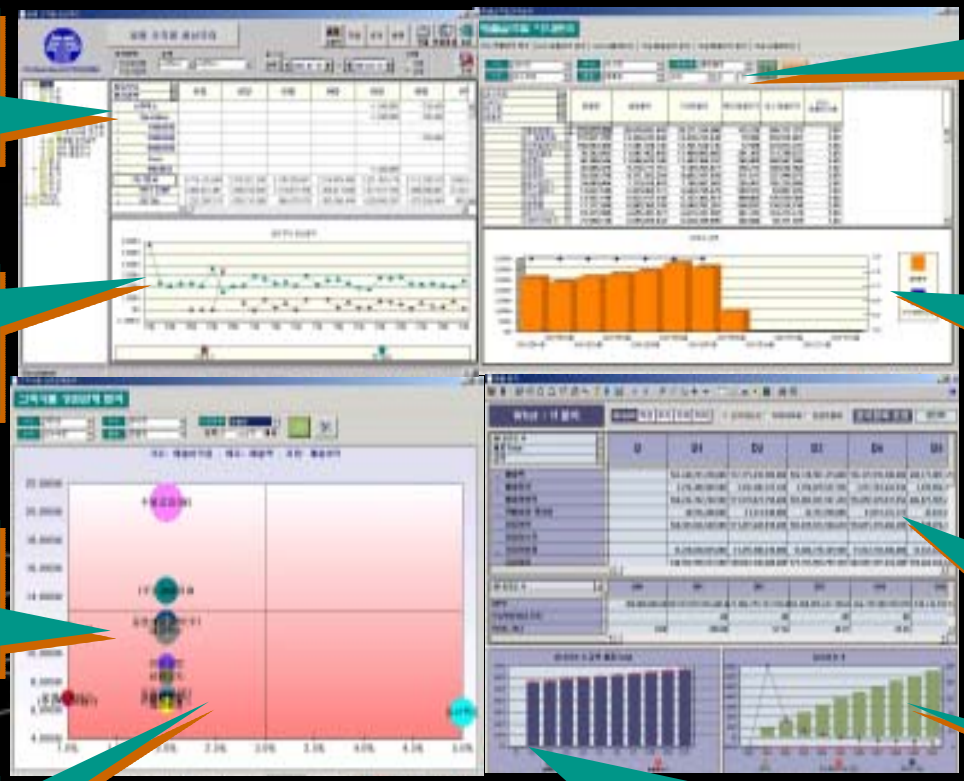
- 다차원 분석, 트렌드, 시뮬레이션, 포트폴리오 분석 등을 통하여 다양한 의사결정 분석정보를 제공.

다차원분석:
주요 정보의 차원별,차원간 다차원분석

트렌드:
데이터 트렌드 분석 및 드릴다운

매트릭스 분석:
의미 있는 변수간 매트릭스 분석을 통한 상관관계 분석

Portfolio분석:
영업 현황 분석 및 의사결정 포트폴리오 분석



분석 정보:
효과적인 고객분석을 통한 이탈 방지 및 이탈고객 회복
효율적인 제품 관리를 위한 고객 성향 분석

비교:
다양한 항목별 비교

투자의사결정분석:
경쟁력 강화 및 수익성 개선 및 신시장 개척을 위한 타당성분석

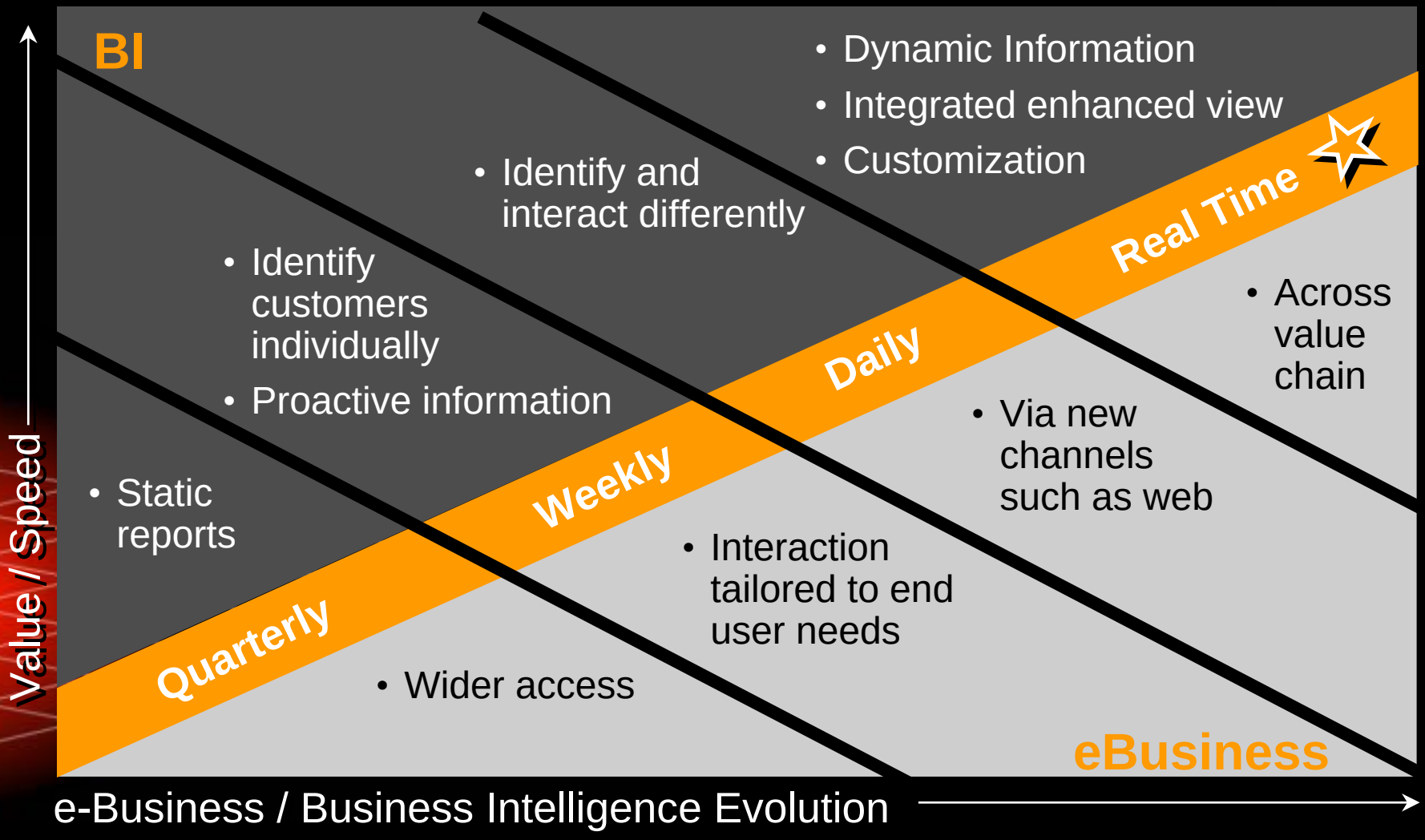
What-if Simulation:
변수별 데이터 조작을 통한 시뮬레이션 및 시나리오 계획

예측:
Forecasting을 통한 SKD Planning

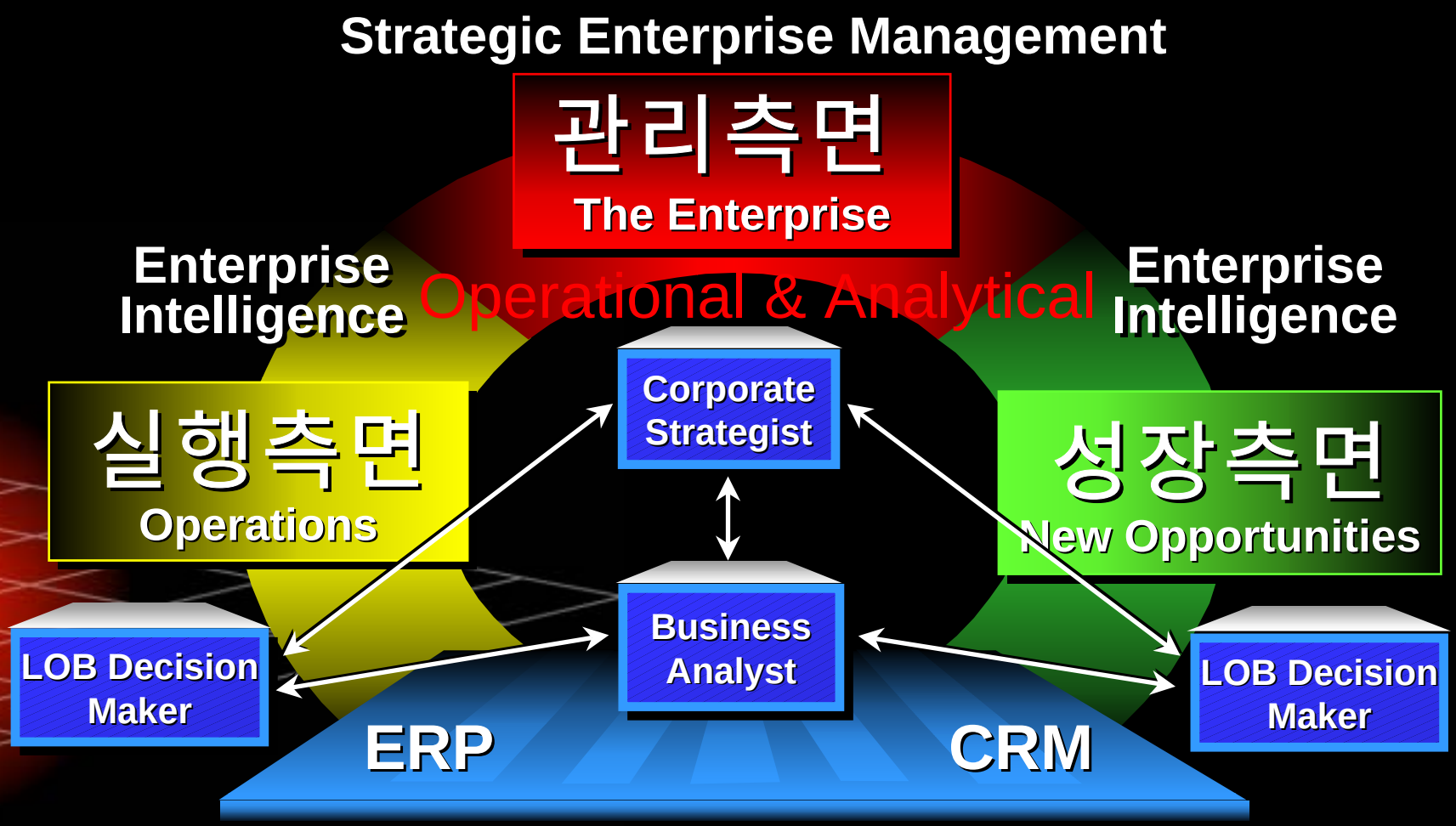
Business Intelligence 구축 추세

10g

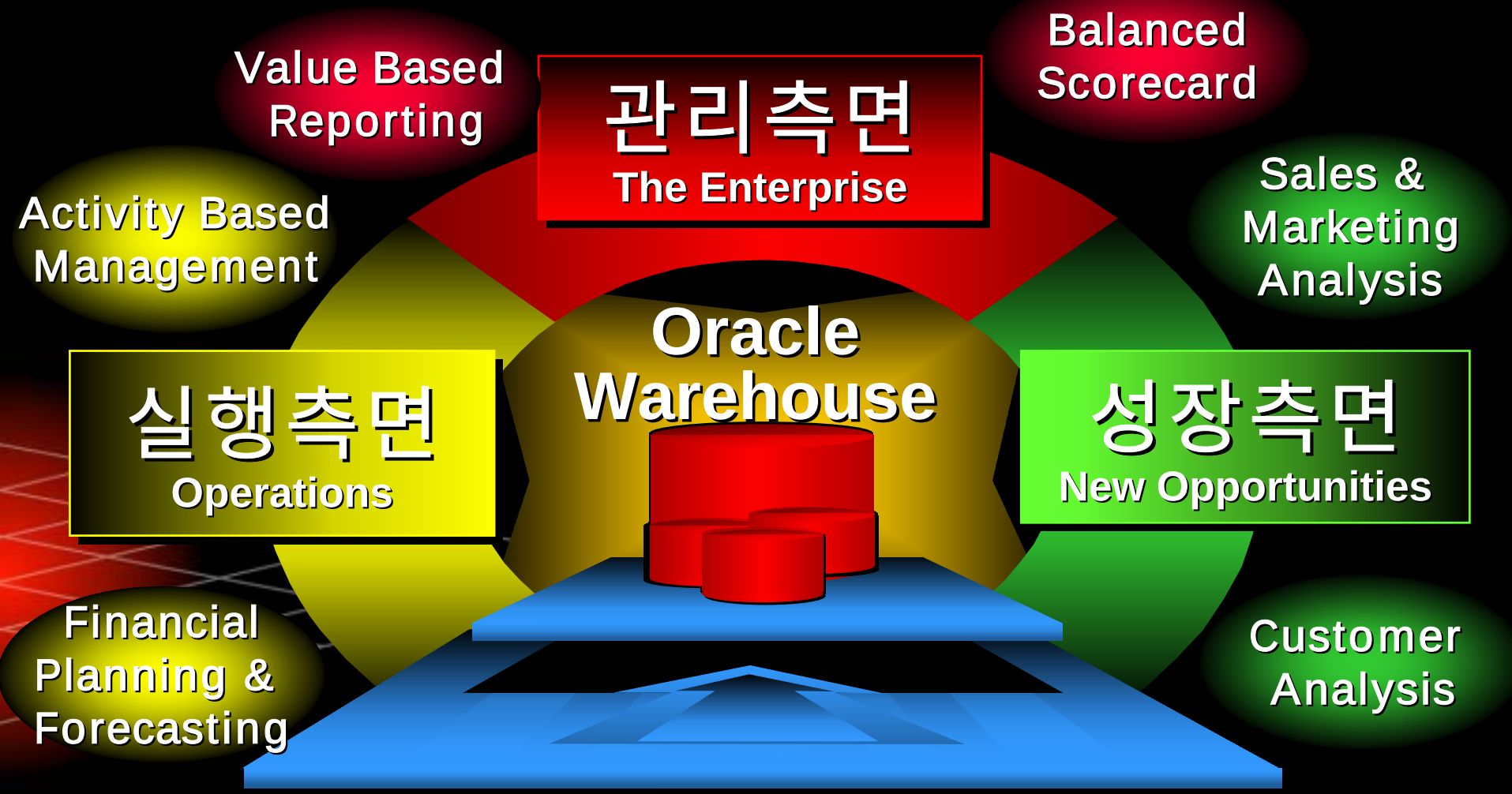
Near Real-time Data



Business Intelligence Infrastructure



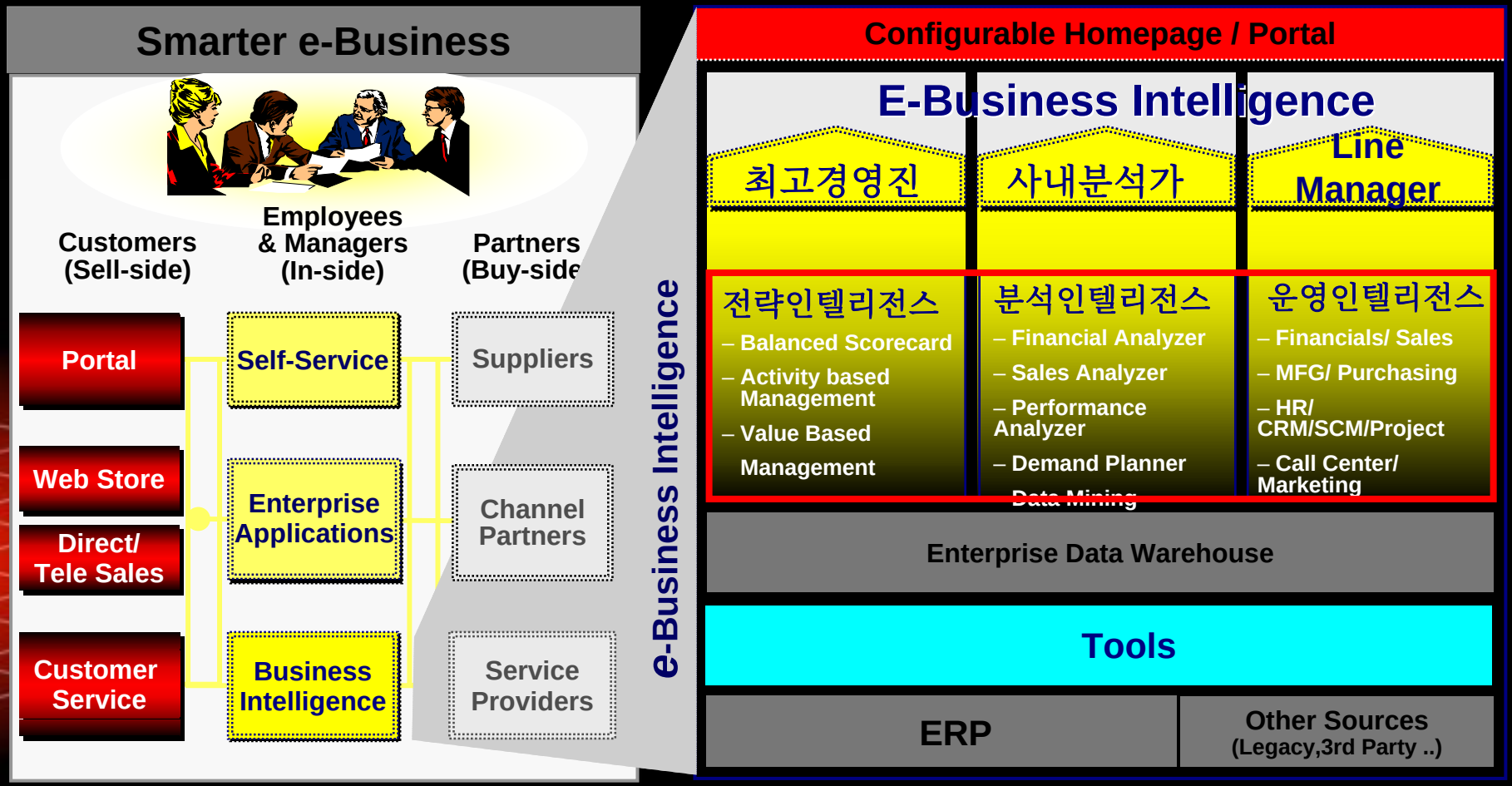
Oracle Business Intelligence



Oracle의 BI 솔루션 모델

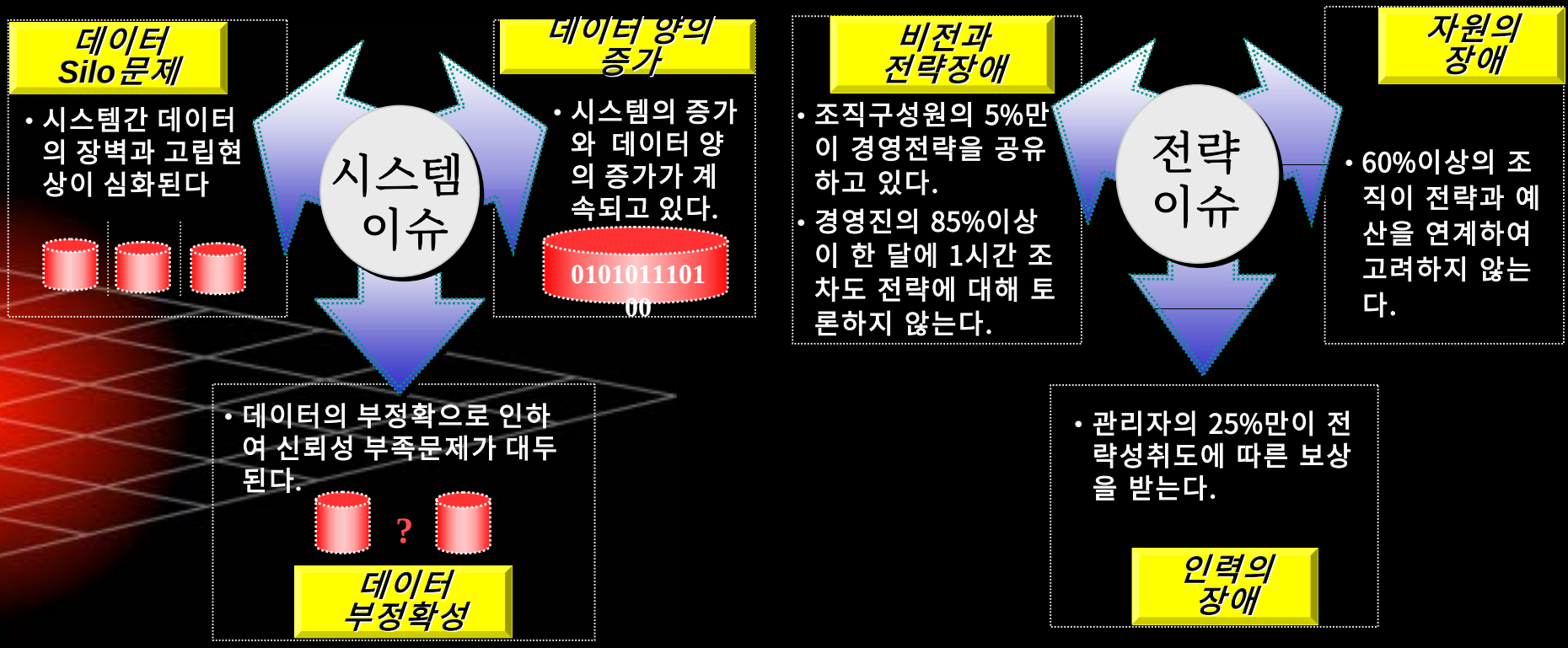
10g

- 오라클이 제시하는 e-Business Intelligence 솔루션 표준 모델

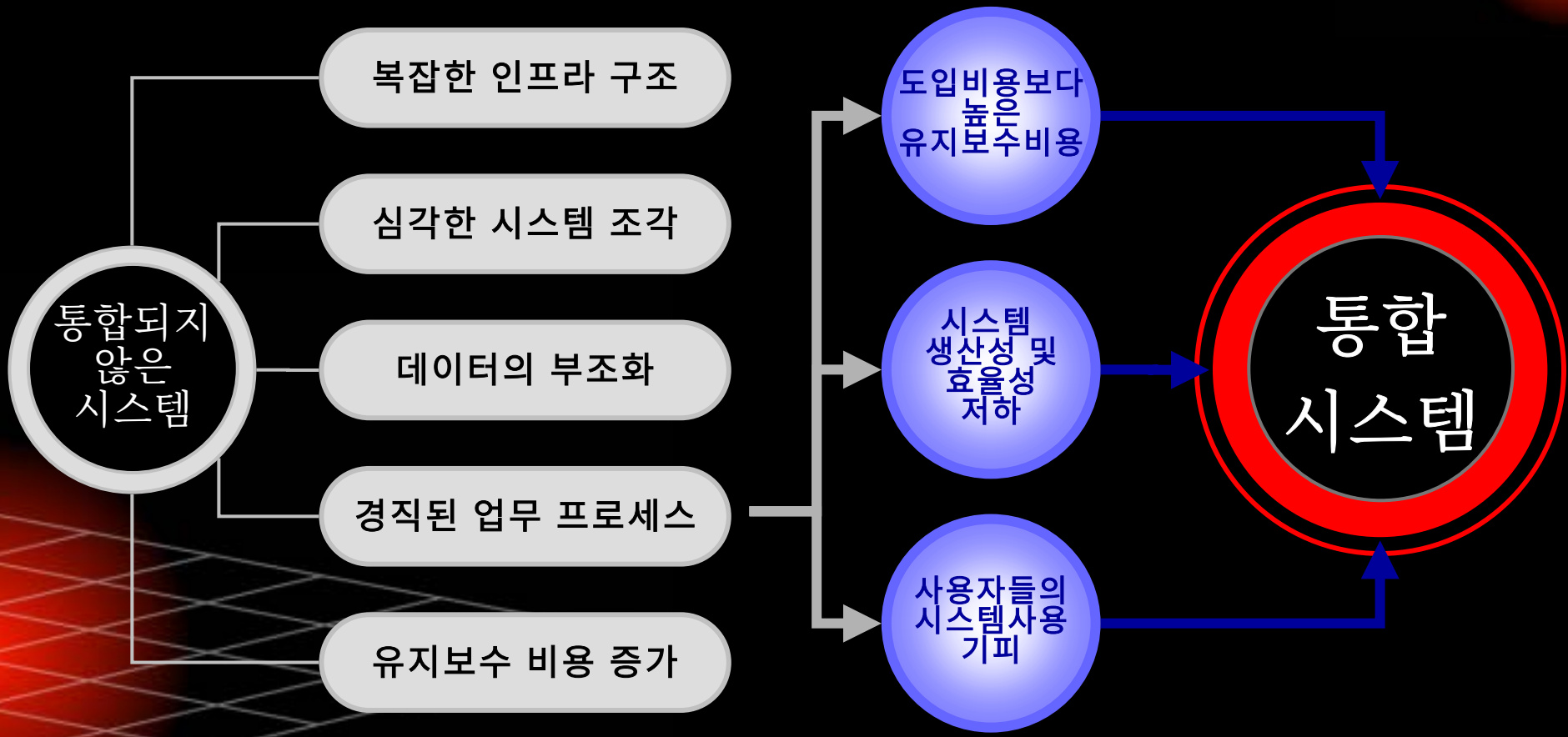


하지만 현실은?

- BI로의 발전을 가로막는 2가지 주요한 문제
  - 문제 1 : 통합되어 있지 않은 데이터와 시스템
  - 문제 2 : 전략과 연계된 성과관리의 미흡



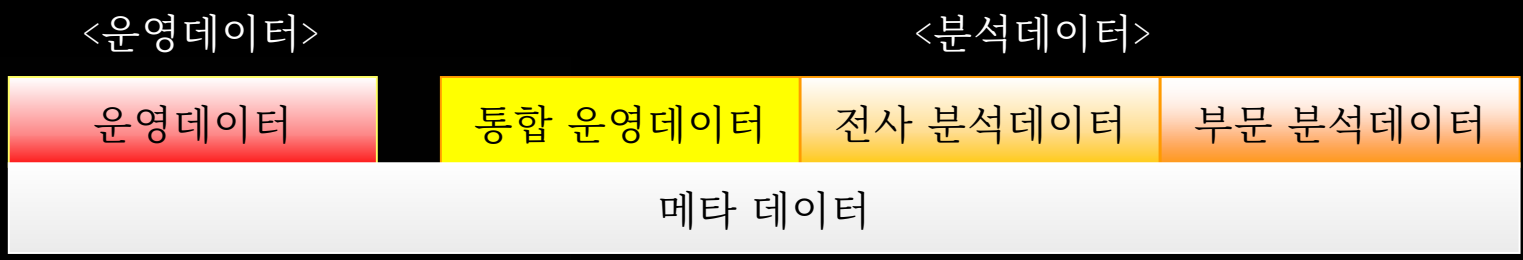
통합 시스템의 필요성



BI 구축을 위해서는 무엇보다도 통합된 데이터 시스템이 우선적으로 필요 되어짐

BI의 전제 조건 : 데이터 통합

효율적인 BI의 구축을 위해서는 먼저 전사에 존재하는 모든 데이터들이 하나의 통합된 관점에서 모델링 되어지고, 이러한 모델이 통합된 시스템 상에서 구현, 운영되어야 함



통합 운영 데이터

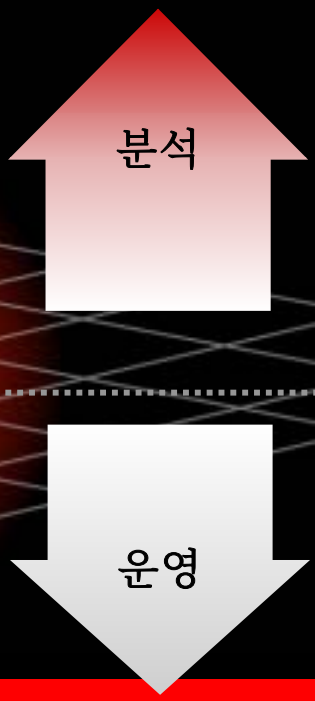
- 운영데이터와 주기적/실시간 일치
- 전사차원 통합구조
- 운영업무데이터 유지, 재정비, 통합
- 주제영역별 모델링
- 현시점 데이터 기준 지속적으로 갱신
- 분석데이터 전환하기 위한 중간 역할

분석데이터

- 의사 결정 지원을 위한 분석 업무에 필요한 데이터
- 이력 데이터
- 주기적 갱신
- 중복 허용
- 주기적인 데이터 갱신

주요 데이터 통합 이슈

- 기업들이 활용하고 있는 기존의 데이터베이스는 단위 업무 시스템들을 개발해나감에 따라 단계적으로 구축
- 통합된 데이터체계 없이, 각각의 기준에 의하여 데이터베이스를 개발해왔기 때문에, 대부분의 기업들에 아래와 같은 문제가 공통적으로 존재



데이터 통합	• 데이터의 중복
데이터 표준	• 데이터의 정의 표준화 되어 있지 않음.
데이터 활용	• 단위 시스템간의 인터페이스 많음 • 인터페이스 방식 복잡

데이터 통합 구축 방향

10g

- 정보의 통합 인프라로 데이터 통합은 전사 모델에 기반을 두었기 때문에 데이터가 특정업무에 편중되지 않도록 중립성이 보장되어야 하며,
- 전사 자원 및 코드를 통합 할 수 있는 현업 사용자의 계속적인 참여가 이루어져야 하고,
- 현재 사용중인 시스템의 대체 효과를 갖을 수 있도록 유연성 있는 구조를 제공하여야 하며,
- 경영전략이나 목표에 따라 변할 수 있는 사용자의 요구를 지속적으로 수용하여 진화하는 데이터 통합 구조.

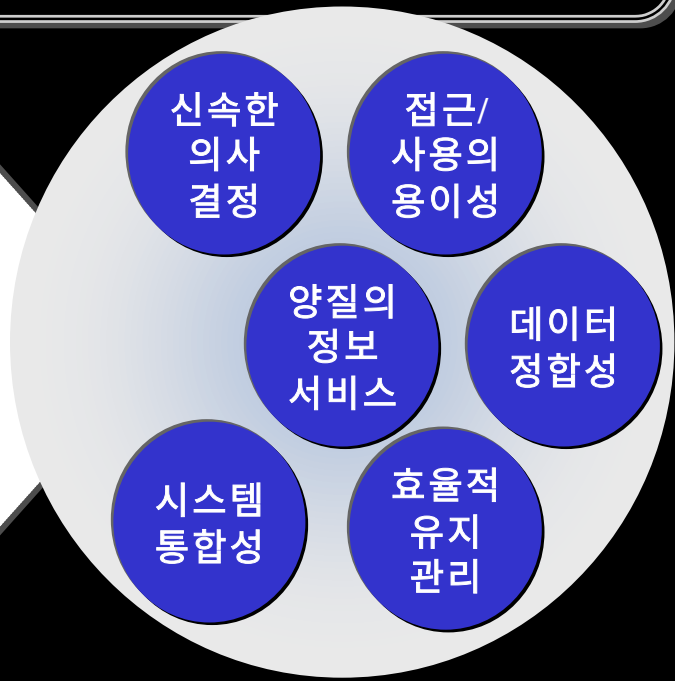
중립적 성격을 갖는 논리 모델

유연하게 대응할 수 있는 진화하는 데이터 구조

확장성을 갖는 데이터 통합

운영시스템을 지원하는 데이터 통합

양질의 분석정보를 제공하는 데이터 통합



ORACLE

기업의 데이터 유형

- 기업에 필요한 데이터를 통합된 데이터베이스 체계를 통하여 일관되게 관리하고, 활용하기 위해서는, 무엇보다도 전사차원에서 데이터베이스를 기획,구축,운영할 수 있는 통합된 데이터 체계와 방법론이 필요
- 기업에서 관리하고, 활용하는 데이터는 크게 <운영데이터> 와 <분석데이터>로 구분

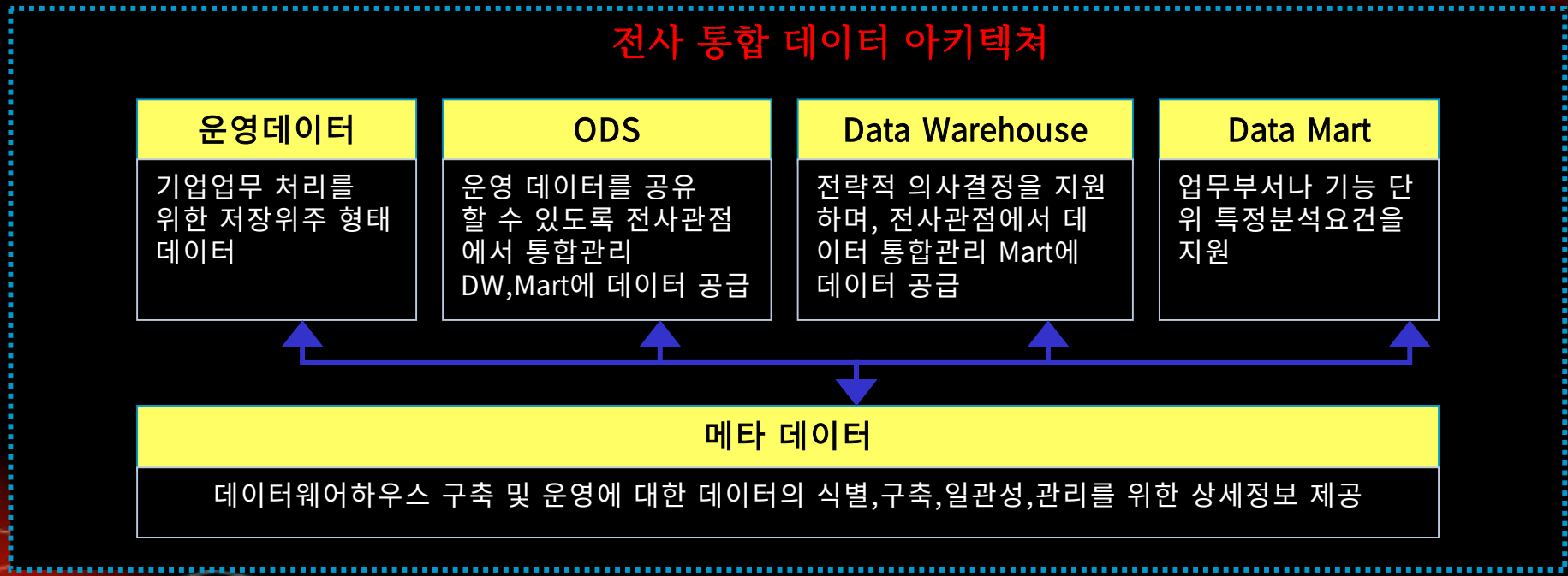
운영업무 vs 분석업무

구분	운영업무	분석업무
기본목표	트랜잭션 지원	분석지원
데이터처리 단위	레코드 단위	데이터 집합 단위
데이터 갱신주기	빈번/수시	주기적/기준의 거
응답시간	수초이내	수분이상
동시사용자	다수	소수
가용성	항상	필요 시
우선순위	높은 성능, 안정성 및 가용성	높은 유연성, 대용량처리

운영데이터 vs 분석데이터

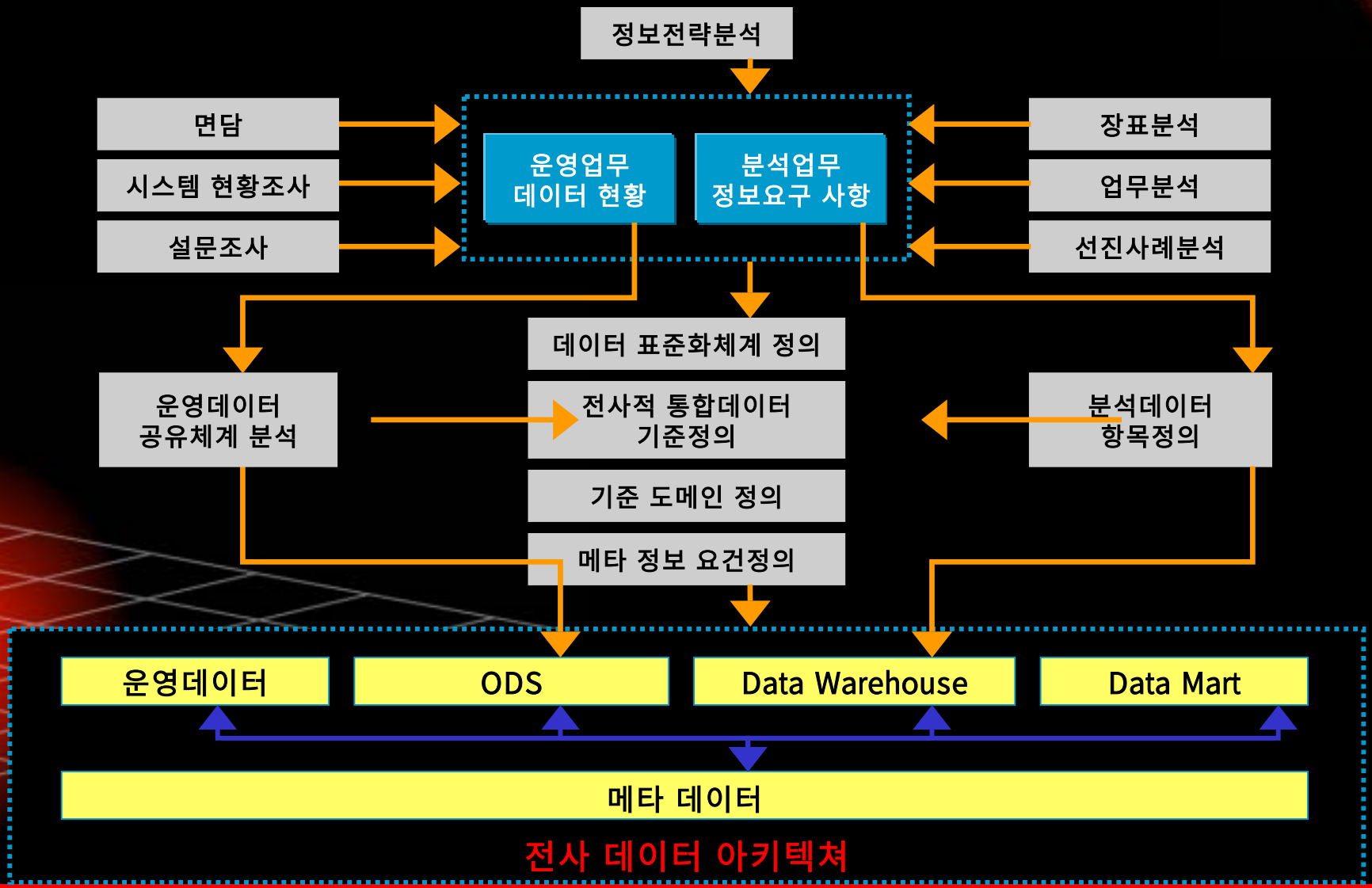
구분	운영데이터	분석데이터
데이터 수준	상세수준	요약수준
데이터 시점	현재시점	이력 데이터
갱신	수시 갱신	필요 시 갱신
데이터 중복	최소화	중복허용
정합성	참조 정합성	이력 데이터 정합성
데이터 예	기업업무 처리를 저장하는 데이터 주문,거래,정산 데이터	의사결정 지원을 위한, 즉 분석 업무에 필요한 데이터

전사 통합 데이터 아키텍쳐의 구성



운영데이터	ODS	Data Warehouse	Data Mart	메타 데이터
-업무처리 기본데이터 -업무 프로세스 중심 -애플리케이션 종속적 -현재시점 데이터 중심 -지속적 데이터 갱신	-운영데이터와 주기적/실시간 일치 -전사차원 통합구조 -주제영역별 모델링	-전사차원 통합구조 -주제중심적 모델링 -이력데이터 중시 -주기적 데이터 갱신	-분석요건 중심 -요약 데이터 -제한된 규모	-데이터에 대한 데이터 -소스데이터 정보(매핑) -변형 및 통합 규칙

전사 통합 데이터 구조 수립 방안



통합 데이터의 활용

10g

전사 통합 데이터

- 독립 운영되는 기간계 시스템을 바탕으로 데이터를 통합하여 전사적인 일관된 View 제공
- 서로 상이한 데이터 체계, 코드 체계 및 중복 관리의 존재를 방지
- 데이터를 체계적으로 요약,축적,변환,공유하여 분석 및 의사결정에 제공
- 목적별, 데이터 운영 주체별 시스템 구축에 따른 데이터 불일치 및 중복 방지
- 데이터 및 주제 영역 추가에 따른 유연한 확장 구조 제공

데이터의 일관성

정보활용 극대화

통합된 One-View

정보 통합 인프라

ORACLE

전사 통합 데이터 아키텍쳐 구성 시 고려 사항

10g

데이터 저장고로서의 역할

소스시스템의 데이터를 축적하여 데이터저장고 역할을 하는 시스템 구축

- 데이터의 활용의 관점이 아닌 데이터를 제공하기 위한 관점에서 원자 데이터 축적
- 중복이 배제된 ERD 형태
- 별도의 Mart를 구성하여 사용자 요구 충족
- 장점
  - 유연성 있는 EDW 구축
  - CRM, SEM 등 확장된 타 시스템과의 연계 고려
  - 분석의 요구가 변함에 따른 모델 변경 최소화
- 단점
  - 사용자가 접근하여 분석이 어려운 구조
  - Mart 구현 시 Business Logic구현 필요

활용 관점

사용자가 접근하기 쉬운 활용 관점의 시스템 구축

- 세부 데이터와 사용자 관점에서 요약된 데이터 축적
- Flat형태의 ERD 또는 MDM (Multi Dimensional Modeling) 형태
- 장점
  - 사용자가 쉽게 접근하여 분석 가능
  - Mart 구현 시 Business Logic의 부담이 적음
- 단점
  - 유연성이 떨어짐
  - 분석의 요구가 증가함에 따라 모델을 변경 및 모델의 추가가 빈번하게 발생
  - 데이터의 중복으로 인하여 일관성이 떨어짐

전사 통합 데이터 아키텍쳐 모델

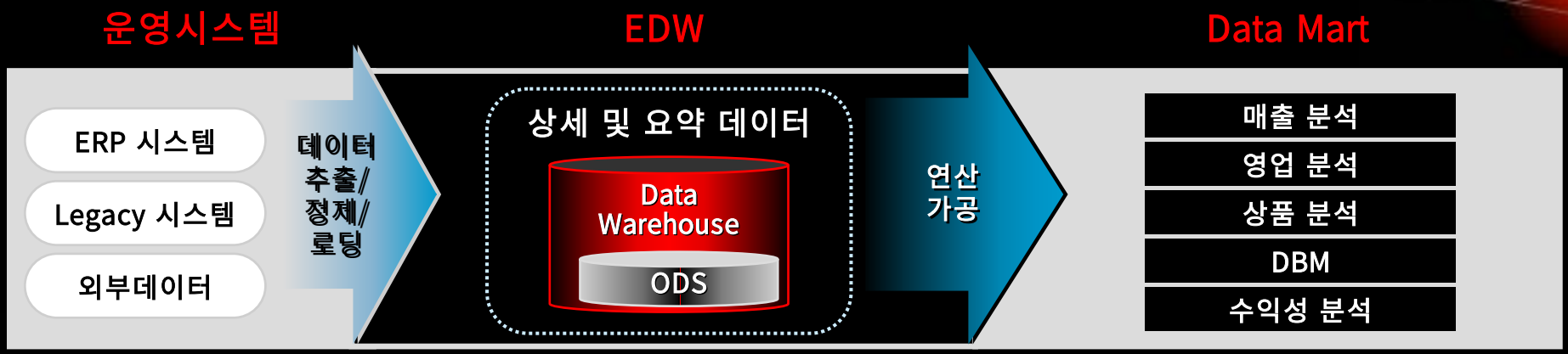
ERD

- OLTP형태의 ERD모델을 이용하여 정보 요구 정의
- 주목적: 소스테이블의 세부 데이터를 축적하는 관점
- 주 사용
  - 운영계 시스템 (OLTP 시스템)
  - 대량의 데이터를 축적하기 위한 전사관점의 EDW 구축
  - Reverse Engineering을 통한 데이터 통합 EDW구축
  - EDW에서 Raw 데이터를 다양한 시스템에 제공하고자 하는 경우 (CRM, SEM)
  - 사용자 요구사항을 파악하기 어려운 경우
  - 조회 & 보고서에 대한 요구가 많은 경우
- 장점
  - 유연성 있는 Model 구축
  - Data의 중복 배제 용이
  - 타 시스템과의 Interface 용이
- 단점
  - 사용자가 직접 정보에 접근하여 조회하기가 어려움 (ER모델보다 MDM 모델이 사용자가 이해하기 편함)

MDM

- 사용자가 이해하기 쉬운 Fact/Dimension Model을 이용하여 정보 요구 정의
- 주목적: 특정업무 또는 특정목적으로 사용자가 사용하기 편리한 데이터 활용관점의 DW를 구축
- 주 사용
  - 특정업무 또는 특정목적으로 Data Mart 구축
  - 사용자 요구사항을 분석하여 DW를 구축
  - DW 구축 경험 부족으로 초기에 DW를 구축하고자 하는 경우
  - 다차원 분석의 요구사항이 많은 경우
- 장점
  - 사용자가 접근하여 사용하기 편한 구조
  - 사용자 요구사항을 즉시 반영하여 EDW 가시적인 효과를 즉시 볼 수 있음
- 단점
  - 사용자 요구사항 중심으로 데이터가 관리됨에 따라 사용자 요구사항이 변함에 따라 유연성 부족
  - 사용자의 요구를 반영하다 보면 테이블의 증가 및 데이터의 중복 및 불일치 발생

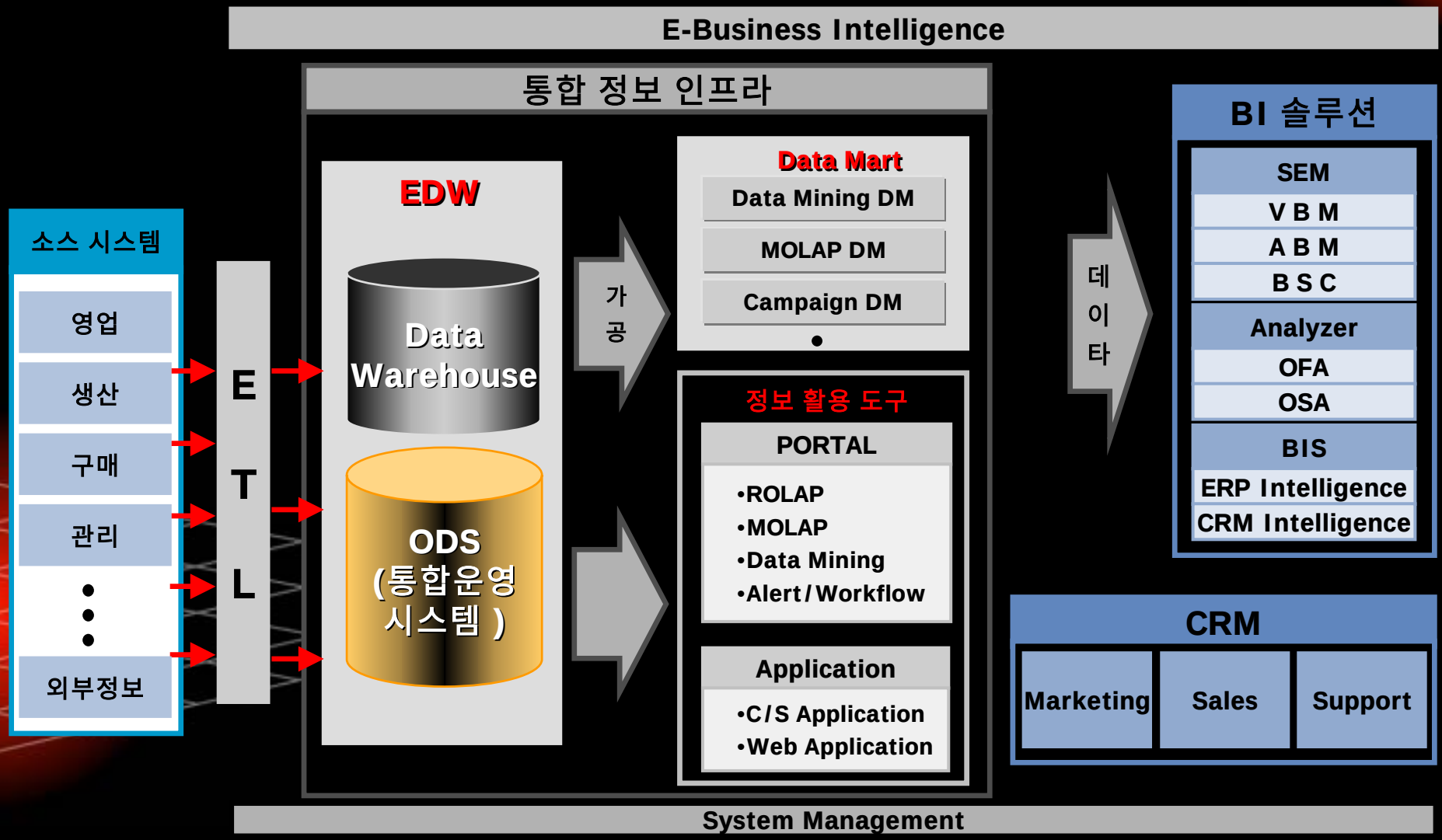
Enterprise Data Warehouse와 Data Mart



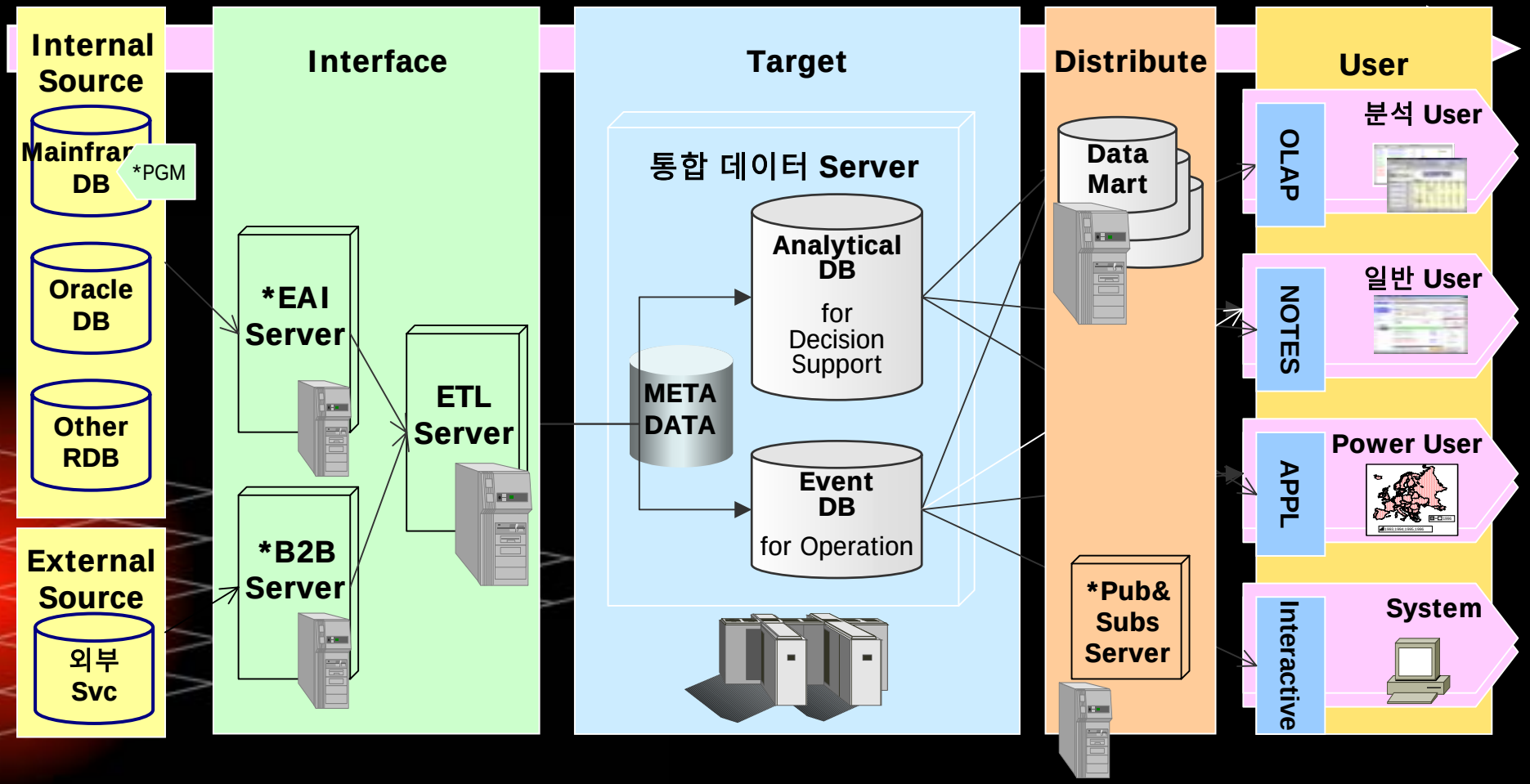
- 데이터 통합 : 데이터 Cleansing 및 축적
  - 데이터 정합성, 완전 무결성 보장
  - 주제중심적으로 데이터를 통합
- 데이터 보관 : 운영데이터의 선별작업을 통한 장기간 보관
  ⇨ 대용량 Raw Data 보관
- TO-BE 관점에서 활용가능성이 있는 데이터 정의 활용 : End User Access 불가, 분석 어려움, 철저한 보안
- 데이터를 제공 및 공유하기 쉬운 형태의 ERD 구조 유지 (기존 Mart 및 타 시스템과의 데이터 통로 역할)
- ER Model : 데이터 분석을 위한 기업 전사적 데이터 Model (Process Model이 아님)

- 데이터 보관 : 요약 데이터
- 활용관점에서 데이터를 가공
- 사용자의 관점에 따라 다양한 형태의 Mart 구성
- EDW 를 통해 기존 Data Mart 를 연계
- 다양한 다차원 모델 분석
- 활용
  - 통합 데이터를 통한 다양한 목적으로 활용
  - Online으로 분석 업무 제공
  - KPI와의 연계
  - 필요 시 Raw Data(EDW)를 볼 수 있음.
    (Online/Offline)

데이터 통합 시스템 논리적 구성도



데이터 통합 시스템 물리적 구성도



통합 데이터 구축 방법

- 주요 데이터 통합 작업
  - Reverse data modeling
  - ETL(Extraction, Transformation, and Loading)
- 데이터 통합 방법론
  - 기업의 전사적인 데이터 아키텍쳐 구성을 위한 데이터베이스를 구축할 때에는 통합성과 일관성을 갖고 구축될 수 있도록 지원하는 데이터 통합 방법론이 필요함

통합 데이터 구성 방안 : Reverse Data Modeling

10g

AS-IS Information

AS-IS Database Schema

DDL utility Tool

DDL Script | SQL Usage

Document

AS-IS Database Design

최종 현재 DDL Script

Repository

TRD

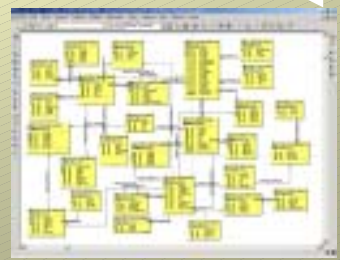
ERD

AS-IS Data Model

AS-IS 논리 모델

To-Be Database Design

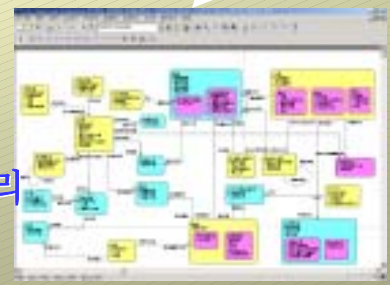
✓ 최적의 성능을 제공하기 위한 데이터베이스 디자인
✓ 분산 데이터베이스 디자인 적용 방안 제공
✓ 대용량 처리에 대한 성능, Storage 전략 방안 제공



디자인

To-Be Data Model

✓ To-Be 통합 데이터 모델
✓ 중복, 불일치 모델 조정
✓ Repository를 통한 Data Architecture 구축 및 관리



Reverse Data Modeling : 단계1

10g

AS-IS Database Schema

기존 Schema 설계서

Get_DDL script

```
Create Table Product_Class
(Product_Class_Code          varchar2(10),
 Product_Class_Name          varchar2(30),
 Product_Class_Level         number,
 Upper_Product_Class_Code    varchar2(1
 Constraint Product_Class_PK
   primary key(Product_Class_Code));

……
……
……

Create Table Sale_Detail
(Sale_Sequence               number,
 Detail_Sequence             number,
 Sale_Date                   varchar2( 8),
 Product_Code                varchar2(10),
 Sale_Quantity               number,
 Constraint Sale_Detail_PK primary key
  (Sale_Sequence,Detail_Sequence));
```

Master Table	Column Name	Child Table	Column Name	SQL Usage
PRODUCT	Product_Code	PRODUCT_HISTORY	Product_Code	Select a.product_name,sum(b.total_sale_amount) from product a, product_history b where b.sale_yyyymm between :v1 and :v2 and a.product_code = b.product_code group by a.product_name;
product	Product_code	Supplier	Main_product_code	Select a.supplier_name,b.product_name as main_product_name from supplier a, product b Where a.main_product_code = b.product_code;

SQL Usage 조사에서 Relationship 반영

추가적인 설계사항 협의/ 도출

용어 사전집을 이용한 한글화 변환 (Optional)

```
Create Table Product_Class
(Product_Class_Code          varchar2(10),
 Product_Class_Name          varchar2(30),
 Product_Class_Level         number,
 Upper_Product_Class_Code    varchar2(10),
 Constraint Product_Class_PK primary key(Product_Class_Code));

Alter Table Product_Class
add constraint Product_Class_Fk1 foreign key(Upper_Product_Class_Code)
  references Product_Class(Product_Class_Code);
……
……
……

Create Table Sale_Detail
(Sale_Sequence               number,
 Detail_Sequence             number,
 Sale_Date                   varchar2( 8),
 Product_Code                varchar2(10),
 Sale_Quantity               number,
 Constraint Sale_Detail_PK primary key(Sale_Sequence,Detail_Sequence));

Alter Table Sale_Detail
add constraint Sale_Detail_FK1 foreign key(Sale_Sequence)
  references Sale_Master(Sale_Sequence);
Alter Table Sale_Detail
add constraint Sale_Detail_FK2 foreign key(Product_Code)
  references Product(Product_Code);
```

Reverse Data Modeling : 단계2

10g

AS-IS
DDL Script

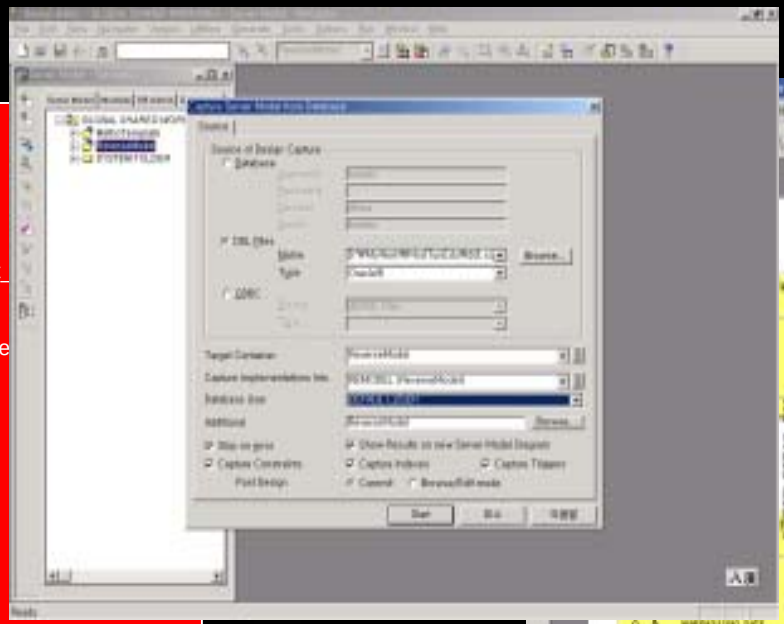
→

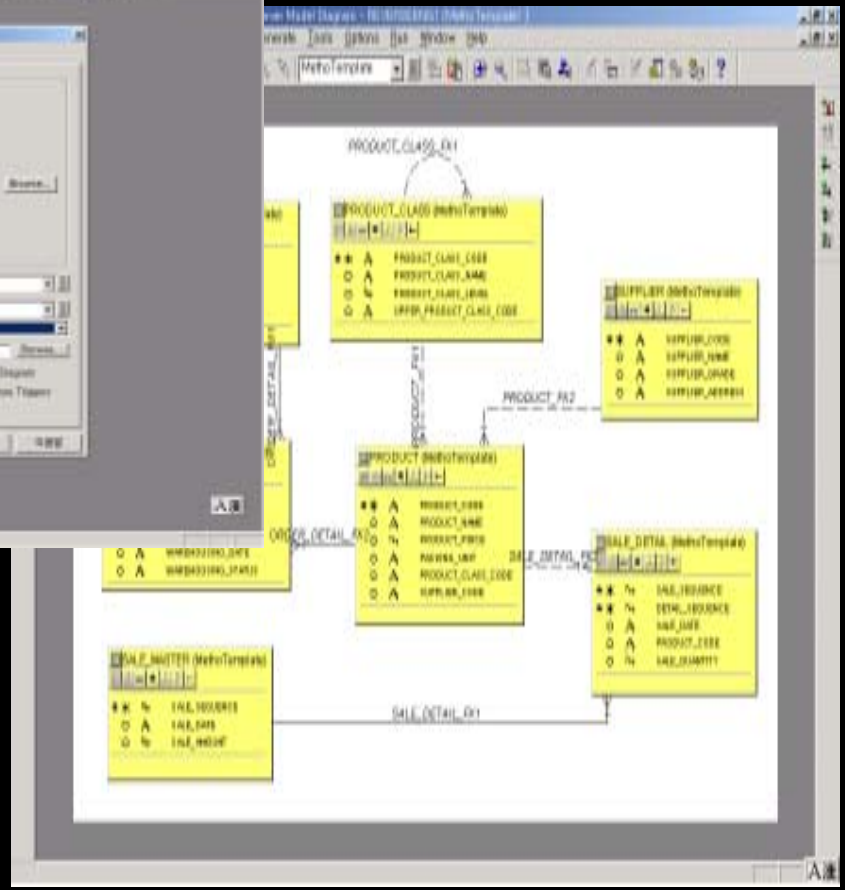
Oracle Designer
Design Editor/Capture Design

→

Schema TR Diagram
(STRD)

```
Create Table Product_Class
(Product_Class_Code          varchar2(10),
 Product_Class_Name          varchar2(30),
 Product_Class_Level         number,
 Upper_Product_Class_Code    varchar2(10),
 Constraint Product_Class_PK primary key(Product_

Alter Table Product_Class
add constraint Product_Class_Fk1 foreign key(Uppe
  references Product_Class(Product_Class_Code);

.....
.....
.....


Create Table Sale_Detail
(Sale_Sequence               number,
 Detail_Sequence             number,
 Sale_Date                   varchar2( 8),
 Product_Code                varchar2(10),
 Sale_Quantity               number,
 Constraint Sale_Detail_PK primary key(Sale_Sequence,Detail_Sequence));

Alter Table Sale_Detail
add constraint Sale_Detail_FK1 foreign key(Sale_Sequence)
  references Sale_Master(Sale_Sequence);

Alter Table Sale_Detail
add constraint Sale_Detail_FK2 foreign key(Product_Code)
  references Product(Product_Code);
```

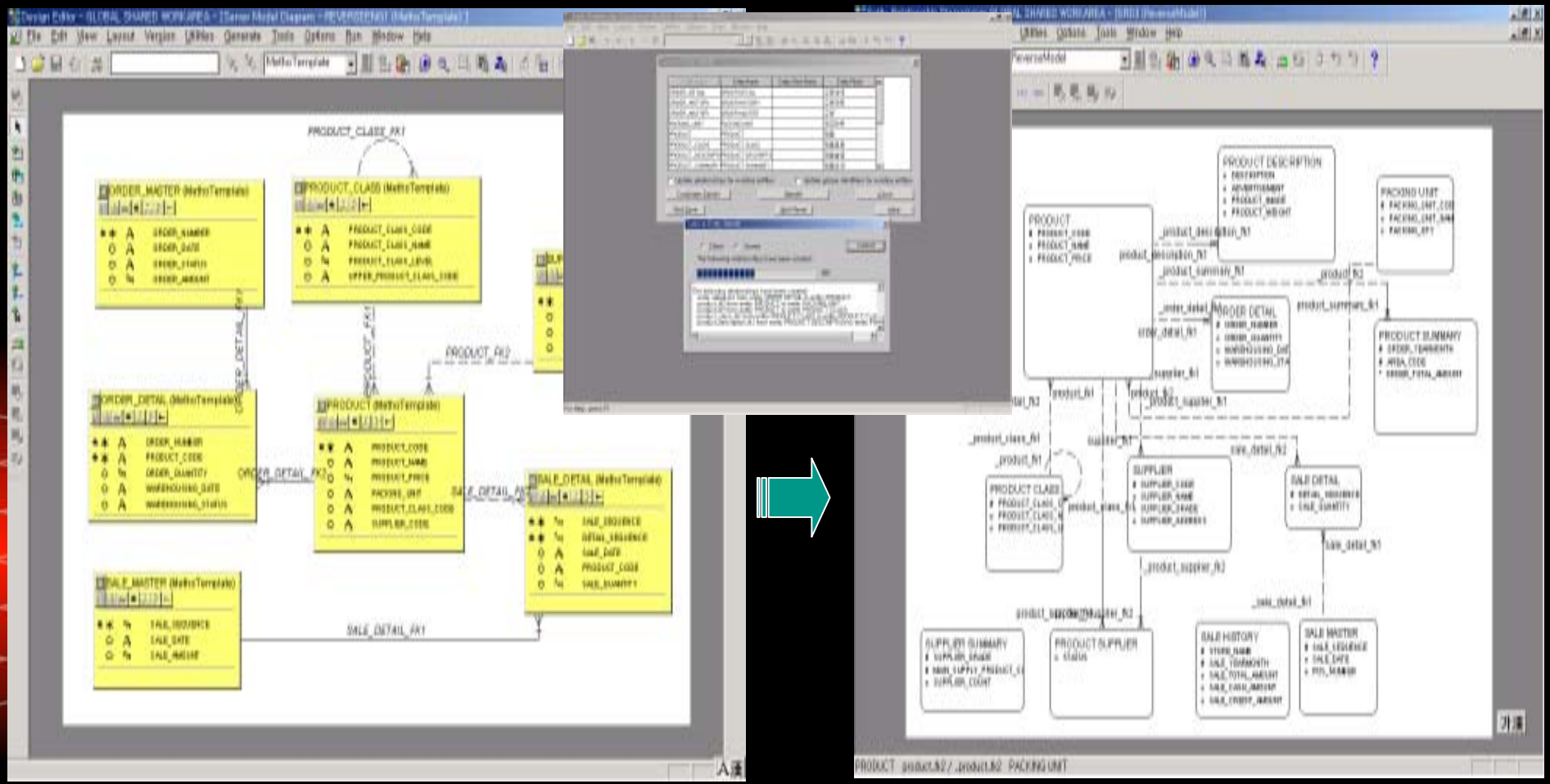




Reverse Data Modeling : 단계3

10g

Schema TR Diagram (STRD) → **Oracle Designer Entity Relationship Diagrammer Utility/Table to Entity Retrofit** → **AS-IS ER Diagram (ERD)**

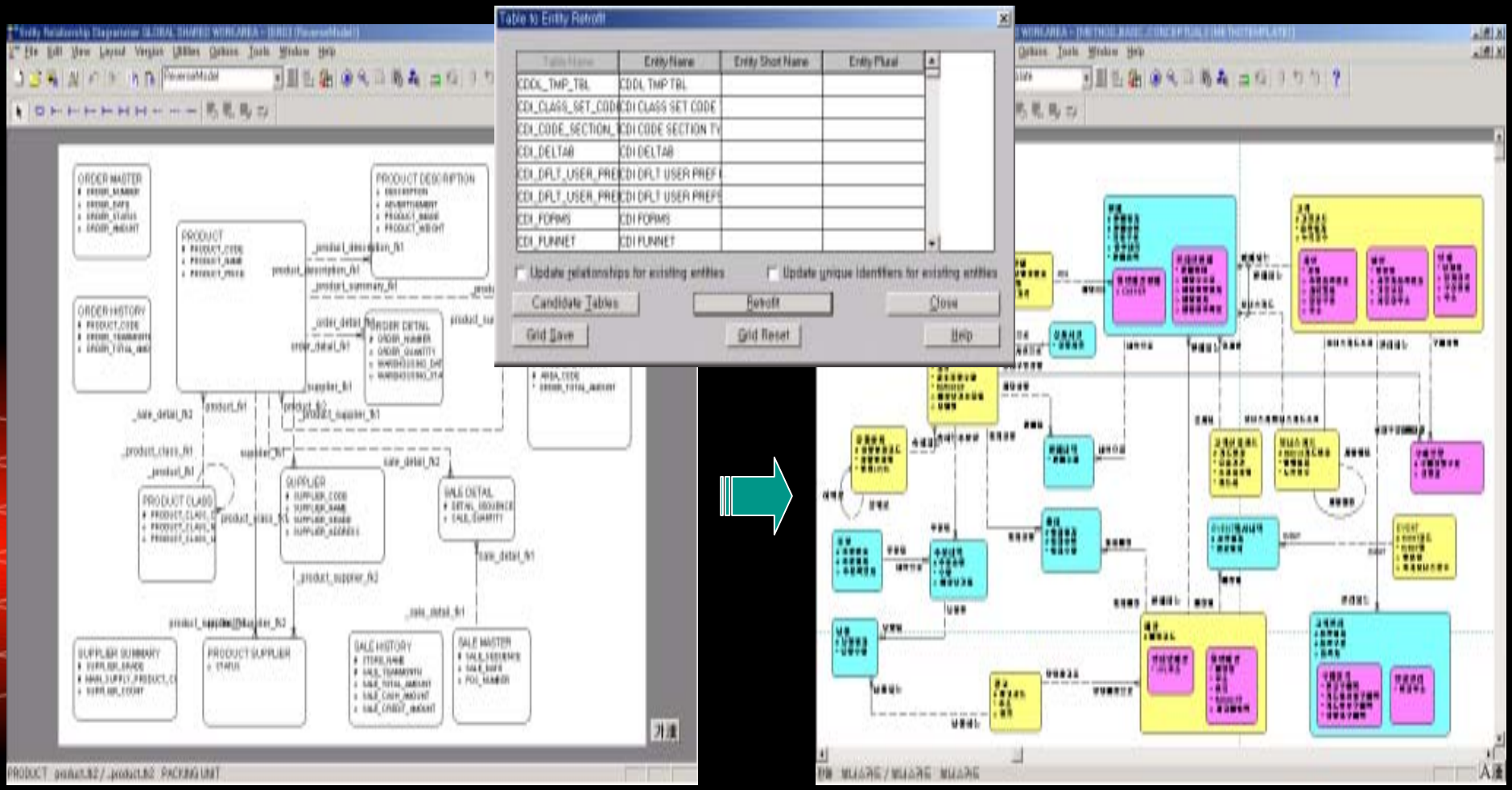


ORACLE

Reverse Data Modeling : 단계4

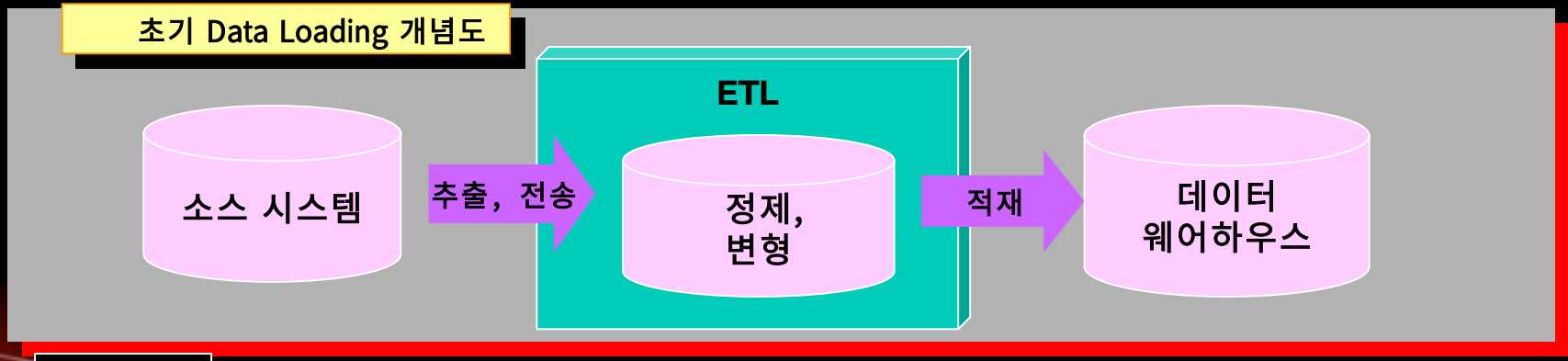
10g

AS-IS ER Diagram (ERD) → **Oracle Designer Entity Relationship Diagrammer** → **TO-BE ER Diagram (ERD)**



통합 데이터 구축 방안 ; ETL

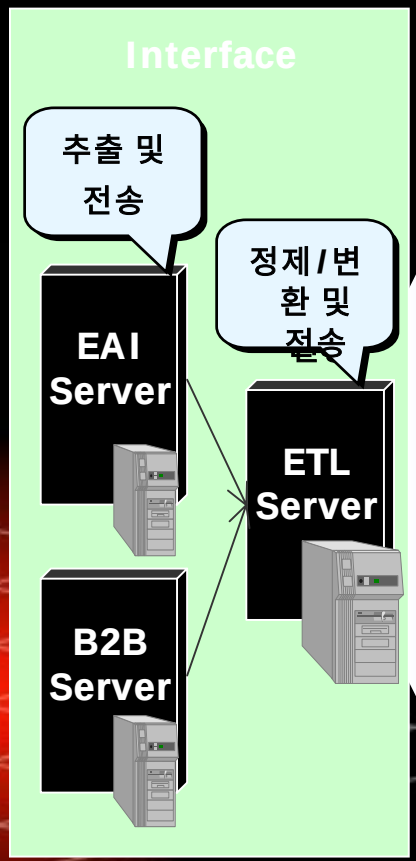
- 통합 데이터 시스템 구축 과정에서 대단히 중요한 단계 중의 하나는, 소스 시스템의 데이터를 추출하고 변형하여 통합 데이터 시스템 서버로 데이터를 적재하는 초기 Data Loading 작업의 효율적인 설계와 개발에 있으며, 통합 데이터 시스템 구축의 성공 여부가 ETL에 의해 좌우될 수 있음



ETL

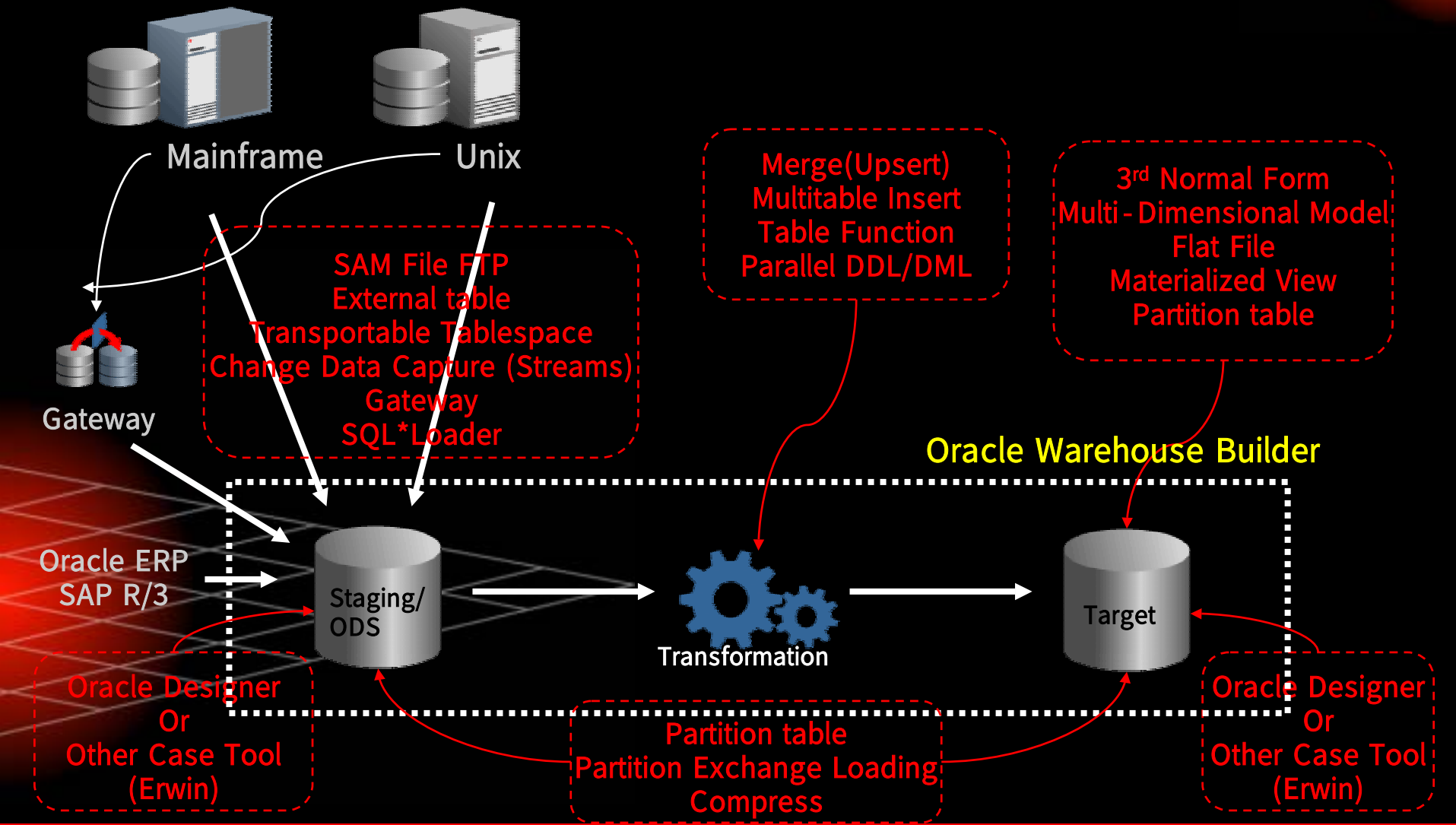
- ETL은 소스시스템에서 데이터를 추출하여 정제, 변형 작업을 거쳐서 이를 데이터 웨어하우스로 적재하는 과정
- Extraction(추출), Transformation(변형), Loading(적재)의 약자
- 소스시스템의 데이터를 데이터 웨어하우스로 Load 하는 것
- 업무 DB 로부터 정보를 추출하여 정제, 변형, 검증 과정을 거쳐 정보 DB에 적재하는 기능을 총칭
- ETL 개념은 단순하지만, 실제 ETL 구현에는 많은 노력 및 투자가 필요
- 대체적으로 전체 데이터 웨어하우징 구축 노력의 50% 이상이 ETL에 소요
- 데이터 로드의 주기, 데이터의 양, 데이터 변환의 정도, 로드 속도, 비용, 소스시스템의 업무마감 형태등에 따라 ETL의 방식이 달라짐

ETL의 주요 고려 요소



추출/전송 방식	• **On-Line 방식 :** ETL 시스템과 소스시스템사이에서 EAI 프로그램이 소스시스템의 데이터를 직접 액세스 한 후, ETL 시스템으로 데이터를 전송하는 방식 • **Batch(Off-Line) 방식 :** 소스시스템의 업무 마감 후 배치 작업으로 데이터를 추출하여 ETL 시스템으로 전송하는 방식 • **Near On-Line 방식 :** 소스시스템에서 업무 중에 추출데이터를 User Log 파일에 생성하면 EAI시스템에서 5분 또는 10분에 한번씩 ETL 시스템으로 데이터를 전송하는 방식
추출 주기	• **일/주/월 단위의 주기적 추출** • **특수한 경우 특정 주기를 설정하여 추출** • **주요 데이터의 경우에는 real-time으로 추출(예:판매 현황)**
정제	• **소스 데이터의 정합성을 검증하여 정확한 데이터로 만드는 과정** • **주요 내용 :** ✓데이터 타입과 값의 불일치 ✓유효 범위를 벗어난 숫자/날짜
변환	• **소스 데이터를 EDW 상에 그대로 적용할 수 없을 때, 데이터를 적용 가능한 형태로 변형시켜주는 과정** • **주요 내용 :** ✓스키마 모델의 변경 ✓업무 코드의 변형

ETL 처리 단계


Mainframe
Unix
Gateway
SAM File FTP
External table
Transportable Tablespace
Change Data Capture (Streams)
Gateway
SQL*Loader
Merge(Upsert)
Multitable Insert
Table Function
Parallel DDL/DML
3rd Normal Form
Multi-Dimensional Model
Flat File
Materialized View
Partition table
Oracle Warehouse Builder
Oracle ERP
SAP R/3
Staging/
ODS
Transformation
Target
Oracle Designer
Or
Other Case Tool
(Erwin)
Partition table
Partition Exchange Loading
Compress
Oracle Designer
Or
Other Case Tool
(Erwin)

통합 데이터 구축 방법론 - Phase

Definition	Analysis	Design	Build	Transition
▪ 프로젝트의 목표와 범위 정의 ▪ 추진 방향 및 전략을 결정 ▪ 프로젝트 팀원들의 목표에 대한 공감대를 형성 ▪ 프로젝트 진행을 위한 제반 환경을 준비 ▪ 전체 기술적 아키텍쳐를 정의 ▪ 역할과 책임의 명확화 ▪ 프로젝트 후원자의 지원과 사용자 참여에 대한 공감대 형성 ▪ EDW의 취지와 목적에 대한 이해를 도모	▪ 업무요구사항을 파악 ▪ 논리적 모델을 정의 ▪ 필요시 역 모델링 수행 ▪ 회사의 비전 및 전략, 현시스템 현황을 반영하여 제반 환경을 재검토	▪ 시스템구축에 필요한 제반 모듈 설계 ▪ 많은 사용자 참여 유도 ▪ 설계내용을 명확히	▪ 설계내용을 실질적으로 구현 ▪ 가동환경으로의 전이가 용이한지 준비 ▪ 각 모듈에 대한 프로그램 코딩을 완료하고, 단위 테스트를 실시 ▪ 가동에 필요한 DB생성을 초기화하고, 초기 데이터로딩을 통해 전체 프로그램의 완전성과 정확성을 확인 ▪ 전체 통합 테스트를 위한 준비	▪ 구축된 시스템을 사용자 환경에 전이 ▪ 가동 Hardware를 준비 ▪ Software를 셋업 ▪ 프로그램 및 설계 확정된 시스템을 가동 장비로 Migration ▪ 시스템 가동에 필요한 데이터 로딩 ▪ 고객의 검수를 위한 Acceptance Test를 실 ▪ 사용자 및 운영자를 위한 매뉴얼 제작 ▪ 사용자 및 운영자를 교육

통합 데이터 구축 방법론 - Process

요건정의 (Data Requirement Definition)

- 사용자 요구사항을 정의하고, 사용자 요구사항에 맞도록 시스템을 설계,구축
- 프로젝트의 추진 방향과 전략을 결정하고 현행 시스템을 분석하여 데이터모델을 정의. DB생성에 대한 준비를 하며, 그 결과를 확인.

데이터 추출(Data Acquisition)

- 기존 시스템으로부터 데이터를 추출하고 로딩하기 위한 요건을 분석하고, 설계 및 프로그래밍
- Tool을 활용할 경우에는 Tool선정 및 기능검토, 추가개발에 대한 요건 정의.
- ETL에 필요한 제반 내용을 정의하고,구현
- 이 과정은 데이터의 품질에 대한 고려가 필요하며, 부정확한 데이터에 대한 처리과정도 고려

기술환경 (Technical Architecture)

- 시스템을 구축하기 위해 필요한 기술적 아키텍쳐 정의
- 시스템 구축 시 상당히 다양한 시스템이나 툴들이 사용될 수 있으며, 현 시스템 기술환경의 확장성 여부 등도 고려

데이터 품질 (Data Quality)

- 데이터의 품질을 확보할 수 있는 방안을 수립하고, 데이터 정제, 통합규칙, 오류, 예외처리 규칙을 정의
- 품질검사 및 관리를 위한 모듈을 구현

통합 데이터 구축 방법론 - Process

메타데이터 관리 (Metadata Management)

- 전사 데이터표준 및 사전을 기반으로 하여 메타 데이터 정의
- 메타데이터의 요건을 분석하고, 시스템에서 효율적으로 사용할 수 있도록 구현하며, 메타 데이터의 관리방법도 정의

데이터베이스 설계 및 구축 (Database Design and Build)

- 통합 데이터 시스템을 구축하기 위해 필요한 관계형 또는 다차원 데이터베이스에 대한 논리적인 모델을 작성하고, 물리적 구조를 설계
- 정의된 논리/물리 모델을 바탕으로 DB생성 스크립트를 작성하고, 이를 이용하여 DB를 구축

데이터 액세스 (Data Access)

- 구축된 시스템을 최종 사용자가 효율적으로 사용할 수 있도록 요건을 정의하고, 설계
- 데이터 액세스를 위한 Tool을 사용할 경우 이에 대한 화면설계
- 툴의 기능으로 구현될 수 없는 경우에는 개발여부를 확정하고, 이에 대한 구현작업을 수행

테스트 (Testing)

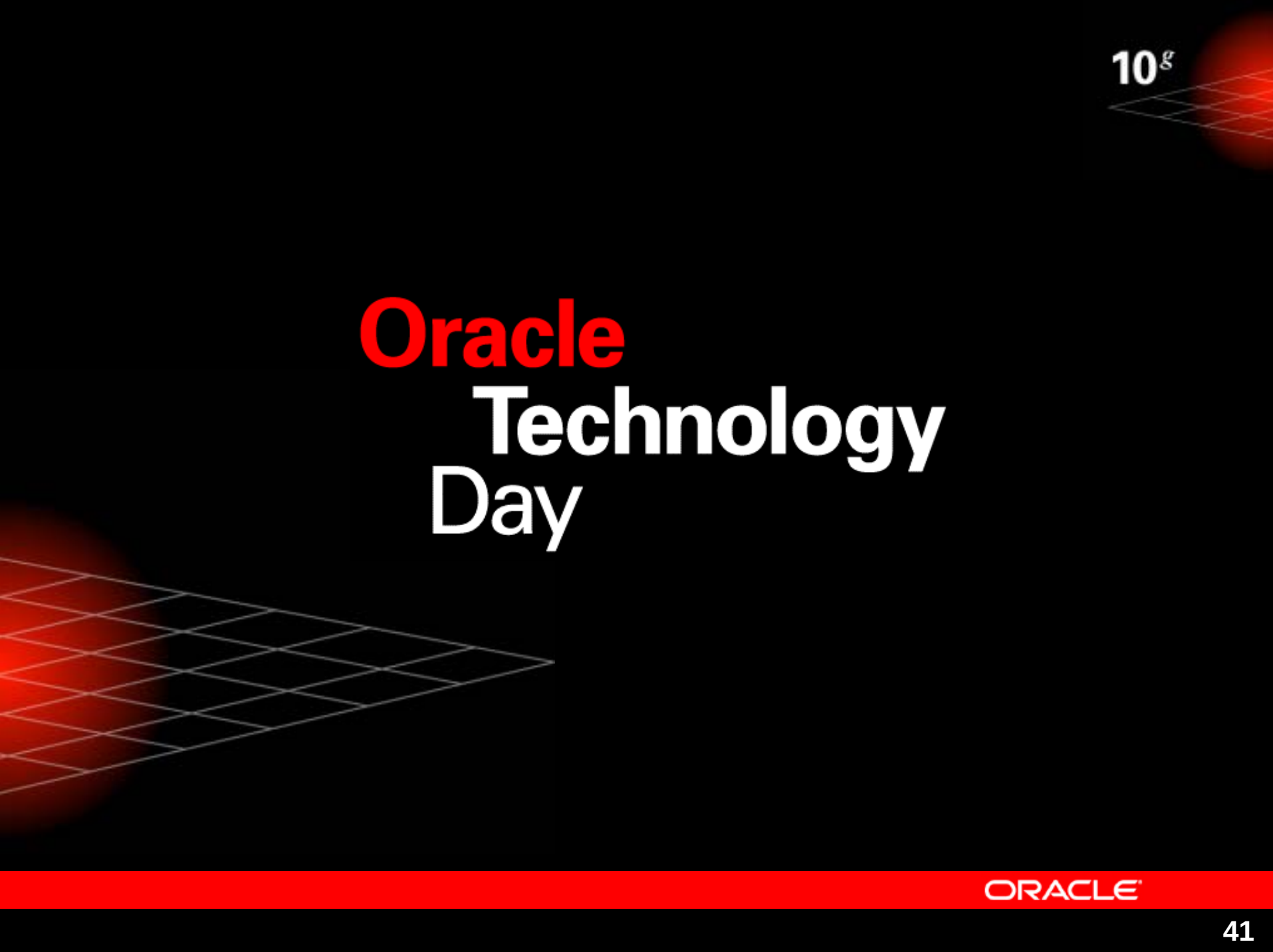
- 테스트 환경 준비 (테스트 요구사항, 테스트 일정, 주체, 시나리오 및 관련 환경) , 통합테스트 및 고객 Acceptance Test 등을 시행

적용 (Adoption and Learning)

- 구축된 시스템을 실제 사용하는 조직에 적용하고 정착시키기 위한 작업을 정의
- 프로젝트 초반에 운영 및 적용에 필요한 계획을 수립, 시스템 관리 및 운영 절차도 포함하여 정의

요약

- BI는 기존 시스템에서 만들어지는 다양한 데이터와 정보의 가치를 극대화 하기 위한 일련의 프로세스임
- BI 구축에 앞서 선행되어야 할 과제는 전사적인 데이터 통합임
  - 전사 데이터 통합 아키텍쳐 구축
  - 데이터 통합 방법론을 통한 실제 시스템 구축


10g
Oracle
Technology
Day
ORACLE