



How to Manage Data for Successful Business Intelligence

Kyungwon Kim

Assistant Professor
Department of International Trade
College of Global Political Science and Economics
Incheon National University

September 1, 2021

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 목표

- 1) 디지털플랫폼 데이터관리 현실은?
- 2) 데이터베이스관리시스템(DBMS)의 필요성과 기능은?
- 3) 관계형 DBMS와 기능은?
- 4) 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털플랫폼 데이터관리 현실은?

▪ 파일 구성용어 및 개념:

- **비트(Bit)**: 컴퓨터가 다룰 수 있는 데이터의 가장 작은 단위
- **바이트(Byte)**: 비트들의 그룹으로 “하나의 문자”로 표현되며 하나의 글씨, 숫자, 기호가 될 수 있음
- **필드(Field)**: 사람의 이름 또는 나이 같이 하나의 단어로 묶인 문자 그룹, 단어 그룹, 숫자 그룹
- **레코드(Record)**: 학생이름, 수강과목, 날짜, 학점과 같은 연관된 필드들의 그룹
 - . **개체(Entity)**: 하나의 레코드는 하나의 개체를 설명하며 행(Row) 값을 의미
 - . **속성(Attribute)**: 개체의 특성을 기술하는 것으로 학생이름, 수강과목, 날짜, 학점 열(Column)을 의미
 - . 속성 = 변수(Variable) = 피쳐(Feature) = 특징(Specification)
- **파일(File)**: 같은 유형의 레코드 그룹
- **데이터베이스(Database)**: 연관된 파일 그룹

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털플랫폼 데이터관리 현실은?

- 많은 기업 플랫폼에 저장된 데이터들이 잘 관리되지 않아 정확하고, 시기적절하고, 적합한 정보 얻지 못함
 - 정확하다: 데이터에 오류가 없음
 - 시기적절하다: 데이터가 가공된 정보가 의사결정자들이 원하는 시점에 사용 가능
 - 적합하다: 정보가 필요한 작업이나 의사결정에 유용하고 적절하게 활용됨

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털 **그림 6.1** 데이터의 계층구조

컴퓨터 시스템은 0 또는 1을 표현하는 비트로 시작하는 계층구조로 데이터를 구성한다. 비트들은 모여서 하나의 글자, 숫자,

▪ **파일** : 기호를 표현할 수 있는 하나의 바이트를 형성할 수 있다. 바이트들이 모여서 하나의 필드를 형성할 수 있으며, 관련된 필드들

▪ **비트** : 이 모여서 하나의 레코드를 형성할 수 있다. 관련된 레코드들이 모여서 하나의 파일을 형성할 수 있으며, 관련된 파일들이 모

▪ **바이트** : 여서 하나의 데이터베이스를 구성할 수 있다.

▪ **필드(Field)**: 사람

▪ **레코드(Record)**:

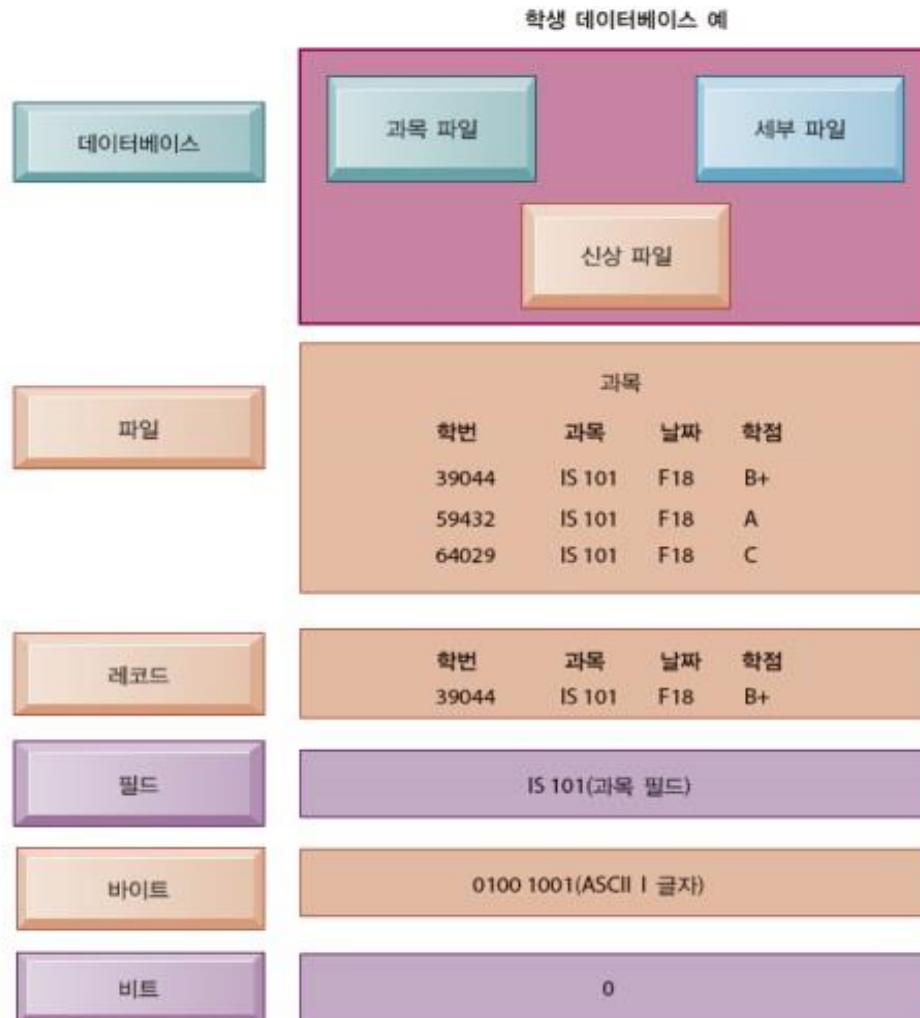
▪ **개체(Entity)**: 객체

▪ **속성(Attribute)**:

▪ **속성 = 변수(Variable)**:

▪ **파일(File)**: 같은

▪ **데이터베이스(Database)**:



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털플랫폼 데이터관리 현실은?

▪ 전통적 파일환경의 문제점:

- 대부분의 조직에서 시스템은 전사적인 계획 없이 독자적으로 성장하는 경향
- 회계, 재무, 제조, 인적자원관리, 판매 및 마케팅과 같은 모든 기능은 그들 자신만의 시스템과 데이터 파일을 개발
- 각각의 조직은 기호에 따라 별도의 프로그램이나 애플리케이션을 구매/개발

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

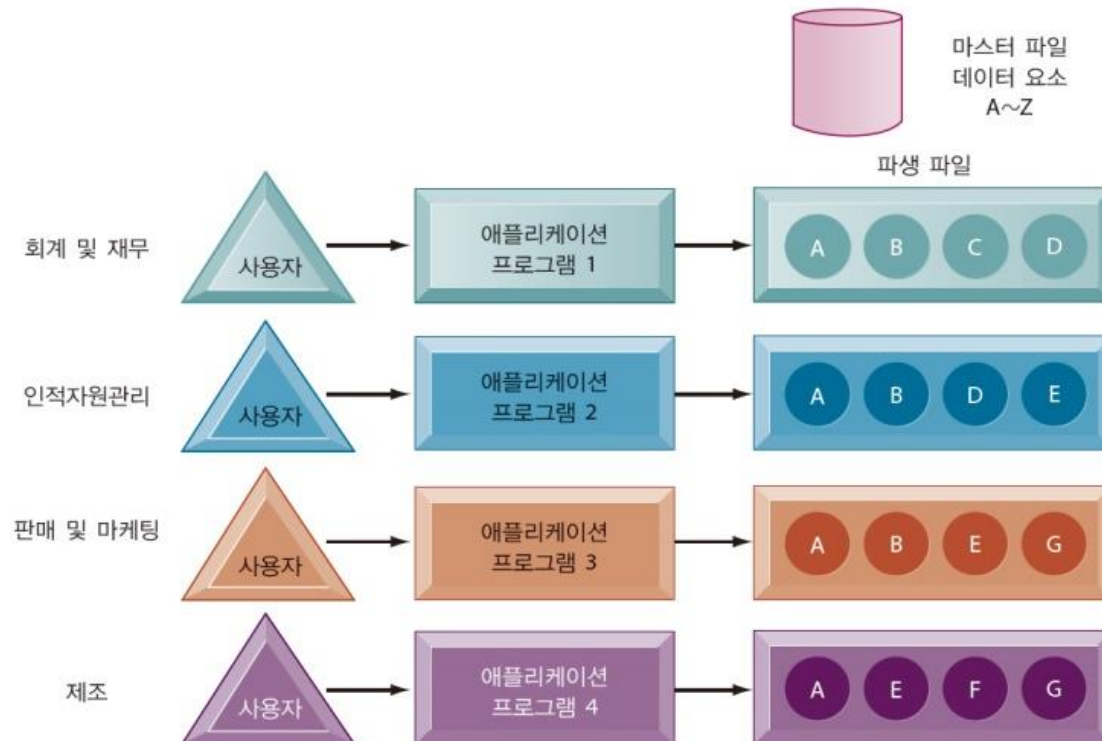
➤ 디지털플랫폼 데이터관리 현실은?

▪ 전통적 파일환경의 문제점:

- 대부분의 조직에서 시스템은 전사적인 계획 없이 독자적으로 성장하는 경향
- 회계, 재무, 제조, 인적자원관리, 판매 및 마케팅과 같은 모든 기능은 그들 자신만의 시스템과 데이터 파일을 개발
- 각각의 조직은 기호에 따라 별도의 프로그램이나 애플리케이션을 구매/개발

그림 6.2 전통적 파일처리

전통적인 파일처리 방식은 기업의 각 기능 영역에 특화된 애플리케이션과 파일을 개발하도록 요구한다. 각각의 애플리케이션은 마스터 파일의 일부분이 될 가능성이 큰 나름대로의 고유한 데이터 파일을 필요로 한다. 이런 파일들은 데이터 중복 및 불일치, 처리 유연성의 저하, 저장 자원의 낭비를 초래한다.



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털플랫폼 데이터관리 현실은?

▪ 전통적 파일환경의 문제점:

- 대부분의 조직에서 시스템은 전사적인 계획 없이 독자적으로 성장하는 경향
- 회계, 재무, 제조, 인적자원관리, 판매 및 마케팅과 같은 모든 기능은 그들 자신만의 시스템과 데이터 파일을 개발
- 각각의 조직은 기호에 따라 별도의 프로그램이나 애플리케이션을 구매/개발
 - . 인적자원부서는 직원 마스터 파일, 급여 파일, 의료보험, 연금, 주소 파일 등을 포함하여 파일과 프로그램이 수백개
 - . 이런 과정은 다수의 마스터 파일의 생성, 유지관리, 그리고 별개의 부문 또는 부서에서 사용을 주도
 - . 5년 또는 10년간 진행되면서 조직은 유지하고 관리하기가 매우 어려운 수많은 프로그램과 애플리케이션에 대한 책임

1) 데이터중복 및 불일치

2) 프로그램-데이터 의존성

3) 유연성 부족

4) 부실한 데이터 보안성

5) 데이터공유 및 가용성 불가능

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털플랫폼 데이터관리 현실은?

1) 데이터중복 및 불일치

- **데이터중복(Data Redundancy):** 데이터가 하나 이상의 장소/위치에 저장되어 여러 데이터 파일에 중복
 - . 조직의 상이한 그룹들이 독자적으로 같은 데이터들을 수집하고 서로 독립적으로 저장할 때 발생
 - . 데이터 중복은 저장 자원의 낭비뿐만 아니라 데이터 불일치를 초래
- **데이터불일치(Data Inconsistency):** 같은 속성이 서로 다른 값을 가지는 것
 - . 과목 개체의 인스턴스에서 날짜는 일부 다른 시스템에서 갱신될 수 있음
 - . 학번 속성은 조직 전반에 걸쳐 상이한 시스템에서 서로 다른 이름들로 표현될 수 있음
 - . 어떤 시스템들은 학번이라는 표현을 사용하고 또 다른 어떤 시스템들에서는 ID
 - . 속성 값으로 표현되는 제품 코드들이 서로 다른 체계를 이용할 때 추가적인 혼란(Extra Large ≠ XL)
- 기업이 상이한 데이터를 통합하는 고객관계관리, 공급사슬관리, 전사적시스템 구축을 어렵게 만들 것

2) 프로그램-데이터 의존성(Program-Data Dependence)

- 프로그램의 변화가 데이터에 대한 변화를 요구하는 경우와 같이
파일에 저장된 데이터와 이런 파일을 갱신/유지하는 특정 프로그램 간의 결합 관계
- 전통적인 파일 환경에서는 소프트웨어 프로그램의 부분이 변경되면 그 프로그램에 접근하는 데이터도 변경해야
 - . 어떤 프로그램은 5자리의 우편번호를 9자리로 수정

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 디지털플랫폼 데이터관리 현실은?

3) 유연성 부족

- 전통적인 파일시스템은 상당한 프로그래밍 노력을 통해 반복적으로 정기적 보고서를 제공
- 특별한 보고서 제공이나 예기치 못한 정보 요구에 대해 시기적절하게 대응할 수 없음
- 특별한 요청에 필요한 정보는 시스템 어딘가 있지만 조회하기에는 너무 많은 비용 초래
- 여러 명의 프로그래머가 새로운 파일에 필요한 데이터 아이템들을 담는 작업은 몇 주가 소요

4) 부실한 데이터 보안성

- 데이터에 대한 통제/관리 메커니즘이 거의 없기 때문에 정보에 대한 접근 및 분배가 통제 밖에 있을 수 있음(정보유출)
- 관리자는 조직의 데이터에 누가 접근하는지 또는 심지어 누가 데이터를 변경하는지 파악할 방법이 없음(정보유출)

5) 데이터공유 및 가용성 불가능

- 상이한 파일과 조직의 상이한 부분에 존재하는 정보들은 서로 연관될 수 없기에 정보가 시기적절히 공유/접근 불가능
- 정보는 조직의 상이한 기능 영역 또는 부분을 관통하여 자유롭게 흐를 수 없음
 - . 사용자들이 2개의 상이한 시스템에 있는 똑같은 정보의 값들이 서로 다르다는 것을 발견하게 된다면, 어떤 고객도 이런 데이터의 정확성을 신뢰할 수 없기 때문에 이런 시스템의 사용을 원치 않을 것

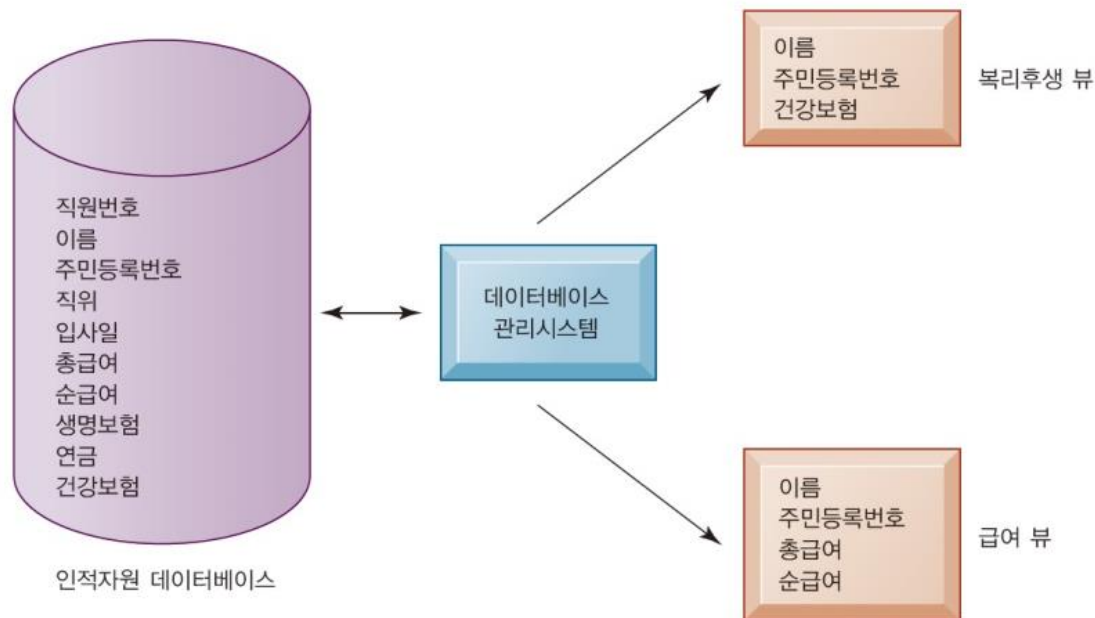
성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 데이터베이스관리시스템(DBMS)의 필요성과 기능은?

- 데이터베이스 기술은 전통적 파일 조직에서 발생하는 다수의 문제점을 줄일 수 있음
- **데이터베이스(Database):**
 - 데이터들의 중앙집중화와 중복 데이터 통제로 다수 애플리케이션을 효율적으로 지원하는 데이터 집합
 - 데이터는 애플리케이션에 별개 파일들에 저장하기보다 사용자들에게 오직 한곳에만 저장된 것처럼 보임
 - 하나의 데이터베이스는 여러 개의 애플리케이션을 지원

그림 6.3 여러 뷰를 제공하는 인적자원 데이터베이스

하나의 인적자원 데이터베이스는 사용자의 정보 요구에 따라 여러 가지 상이한 데이터 뷰를 제공한다. 여기서 설명된 예는 두 가지 뷰의 예를 보여주는데, 하나는 복리후생 전문가가 관심을 갖는 뷰이고 다른 하나는 급여 담당 직원이 관심을 갖는 뷰이다.



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 데이터베이스관리시스템(DBMS)의 필요성과 기능은?

▪ 데이터베이스관리시스템(Database Management System, DBMS):

> 조직이 데이터를 중앙집중화하고, 효율적 관리하며, 애플리케이션 데이터에 접근할 수 있는 소프트웨어

- DBMS는 애플리케이션과 물리적 데이터 파일 간에 인터페이스 역할

- 애플리케이션이 “충급여” 데이터를 호출하면, DBMS는 데이터베이스에서 이 아이템을 찾아 제공

> DBMS는 데이터의 논리적 뷰와 물리적 뷰를 구분함으로써 프로그래머 또는 최종사용자가

데이터들이 실제 어디에 어떻게 저장되는지 이해하는 부담을 덜어 줌

- **논리적 뷰(Logical View):** 데이터들이 최종사용자 또는 비즈니스 전문가들이 인지하는 방식으로 표현

- **물리적 뷰(Physical View):** 데이터가 실제 물리적 저장매체에 어떻게 구조화되는지 보여줌

- 데이터베이스관리 소프트웨어는 사용자들이 요구하는 상이한 논리적 뷰들을 통해서

물리적 데이터베이스가 사용될 수 있도록 만들어 줌

> DBMS는 조직이 데이터 중복을 완전히 제거할 수 있도록 만들 수는 없지만,

조직이 모든 중복 데이터를 같은 값으로 변경/수정 하도록 돕기 때문에 DBMS 사용을 통해 데이터 불일치를 제거 가능

> DBMS로 조직은 중앙에서 데이터, 사용, 보안을 관리할 수 있고, 조직 전반에 데이터 공유가 용이해지는데,

그 이유는 데이터들이 상이한 시스템과 파일에 흩어져 있는 것이 아니라 하나의 장소에 있는 것처럼 사용자들에게 제시

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

- DBMS는 개체, 속성, 관계성을 지속적으로 파악하기 위해 상이한 데이터 모델들을 사용
- **관계형 DBMS:** 오늘날 대형 컴퓨터 및 메인프레임뿐만 아니라 PC에 대해서도 가장 보편적인 DBMS
 - 관계형 데이터베이스는 데이터들을 2차원의 테이블[관계(Relation)라 부름]로 표현하거나 파일로 저장
 - 각각의 테이블은 하나의 개체와 속성의 데이터로 구성
 - . 마이크로소프트 액세스는 데스크톱 시스템용 관계형 DBMS
 - . DB2, 오라클DB, 마이크로소프트 SQL 서버 등은 대용량/중간급 메인프레임 컴퓨터 대상 관계형 DBMS
 - . MySQL은 인기 있는 오픈소스 DBMS 중 하나
 - . 오라클 데이터베이스 라이트(Oracle Database Lite)는 모바일 컴퓨팅 기기에 사용되는 DBMS

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

- DBMS는 **그림 6.4** 관계형 데이터베이스 테이블

하나의 관계형 데이터베이스는 데이터들을 2차원 테이블 형태로 구성한다. 여기서 설명된 공급자와 부품 개체들에 대한 테이블은 각각의 개체와 속성이 어떻게 표현되는지를 보여준다. 공급자_번호는 공급자 테이블에 대해서는 주키이고 부품 테이블에 대해서는 외래키이다.

IMS

- 관계형 데이터베이스

- 각각의 테이블

. 마이크로

. DB2, 오라클

. MySQL

. 오라클

열(속성, 필드)

공급자_번호	공급자_이름	공급자_거리	공급자_도시	공급자_주	공급자_우편번호
8259	CBM Inc.	74 5 th Avenue	Dayton	OH	45220
8261	B. R. Molds	1277 Gandolly Street	Cleveland	OH	49345
8263	Jackson Composites	8233 Micklin Street	Lexington	KY	56723
8444	Bryant Corporation	4315 Mill Drive	Rochester	NY	11344

행
(레코드,
튜플)

부품

키 필드(주키)

부품_번호	부품_이름	단위_가격	공급자_번호
137	Door latch	22.00	8259
145	Side mirror	12.00	8444
150	Door molding	6.00	8263
152	Door lock	31.00	8259
155	Compressor	54.00	8261
178	Door handle	10.00	8259

주키

외래키

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

▪ 관계형 DBMS의 기능:

- 관계형 데이터베이스 테이블은 2개의 테이블이 공통된 데이터 요소 기준 사용자가 원하는 데이터 제공하기 위해 쉽게 결합
- 세 가지 기본적인 연산 1) 고르기(Select), 2) 합치기(Join), 3) 보여주기(Project) 사용가능

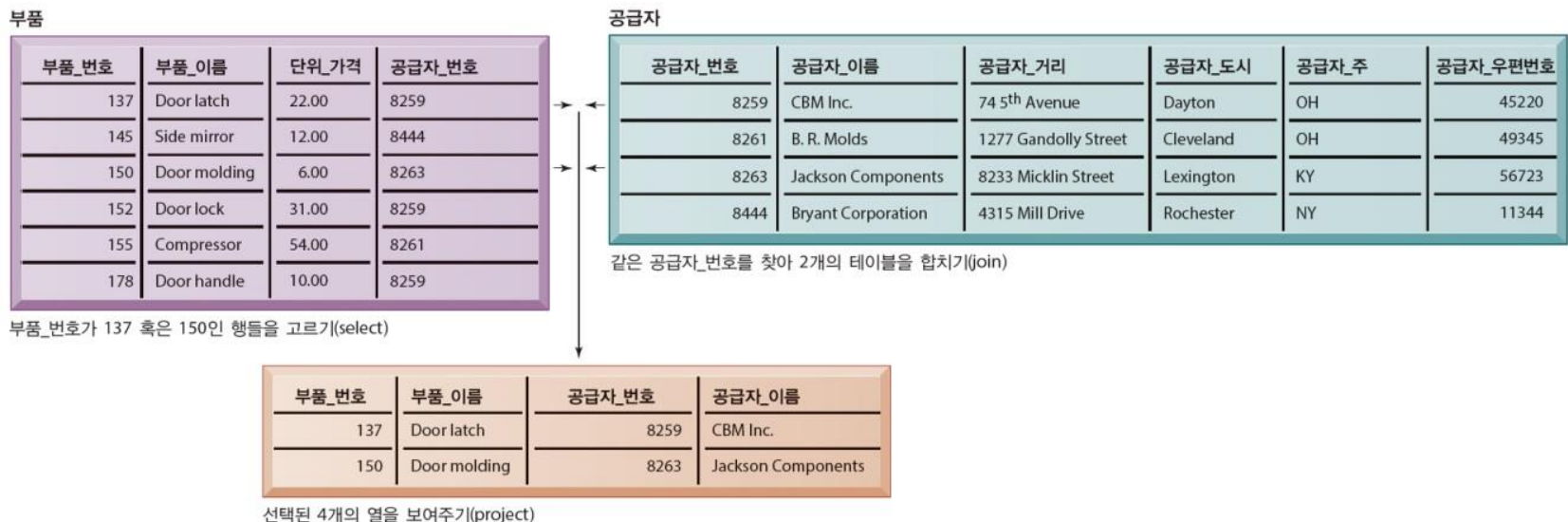
. 고르기(Select): 조건을 만족시키는 파일의 모든 레코드(행들) 또는 속성(열들)로 구성된 부분집합을 생성

. 합치기(Join): 사용자에게 개별 테이블들에서 제공 가능한 정보들을 제공하기 위해 관계형 테이블들을 결합

. 보여주기(Project): 합치기 연산을 통해 만들어진 새로운 테이블로부터 필요한 속성들만 골라서 보여줌

그림 6.5 관계형 DBMS의 세 가지 기본 연산

고르기, 합치기, 보여주기 연산은 2개의 상이한 테이블의 데이터들을 결합하고 오직 선택된 속성들만 볼 수 있도록 해준다.



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

▪ 관계형 DBMS의 기능:

- 관계형 데이터베이스 테이블은 2개의 테이블이 공통된 데이터 요소 기준 사용자가 원하는 데이터 제공하기 위해 쉽게 결합
- 세 가지 기본적인 연산 1) **고르기(Select)**, 2) **합치기(Join)**, 3) **보여주기(Project)** 사용가능
 - . **고르기(Select)**: 조건을 만족시키는 파일의 모든 레코드(행들) 또는 속성(열들)로 구성된 부분집합을 생성
 - . **합치기(Join)**: 사용자에게 개별 테이블들에서 제공 가능한 정보들을 제공하기 위해 관계형 테이블들을 결합
 - . **보여주기(Project)**: 합치기 연산을 통해 만들어진 새로운 테이블로부터 필요한 속성들만 골라서 보여줌
- DBMS는 데이터베이스의 데이터들을 구성하고, 관리하고, 접근하기 위한 기능과 도구들 포함
 - . 대부분의 DBMS는 데이터 **조작어(Data Manipulation Language)**라는 특별한 언어로 데이터를 추가, 변경, 삭제, 조회
 - . 언어는 최종사용자와 프로그래밍 전문가들이 원하는 데이터를 추출해주는 명령어들을 포함
 - . **SQL(Structured Query Language)**: 오늘날 가장 대표적인 데이터 조작어
 - . DB2, 오라클, SQL 서버와 같은 대형/중형 컴퓨터용 DBMS의 사용자들은 필요한 데이터를 조회하기 위해 SQL을 사용

그림 6.7 SQL 쿼리 예

여기에서 기술된 것은 부품 137이나 150에 대해 공급자를 찾아내는 쿼리 SQL문이다. 이 SQL문은 그림 6.5와 동일한 결과를 제공한다.

```
SELECT 부품.부품_번호, 부품.부품_이름, 공급자.공급자_번호, 공급자.공급자_이름
FROM 부품, 공급자
WHERE 부품.공급자_번호 = 공급자.공급자_번호 AND
부품_번호 = 137 OR 부품_번호 = 150;
```


성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

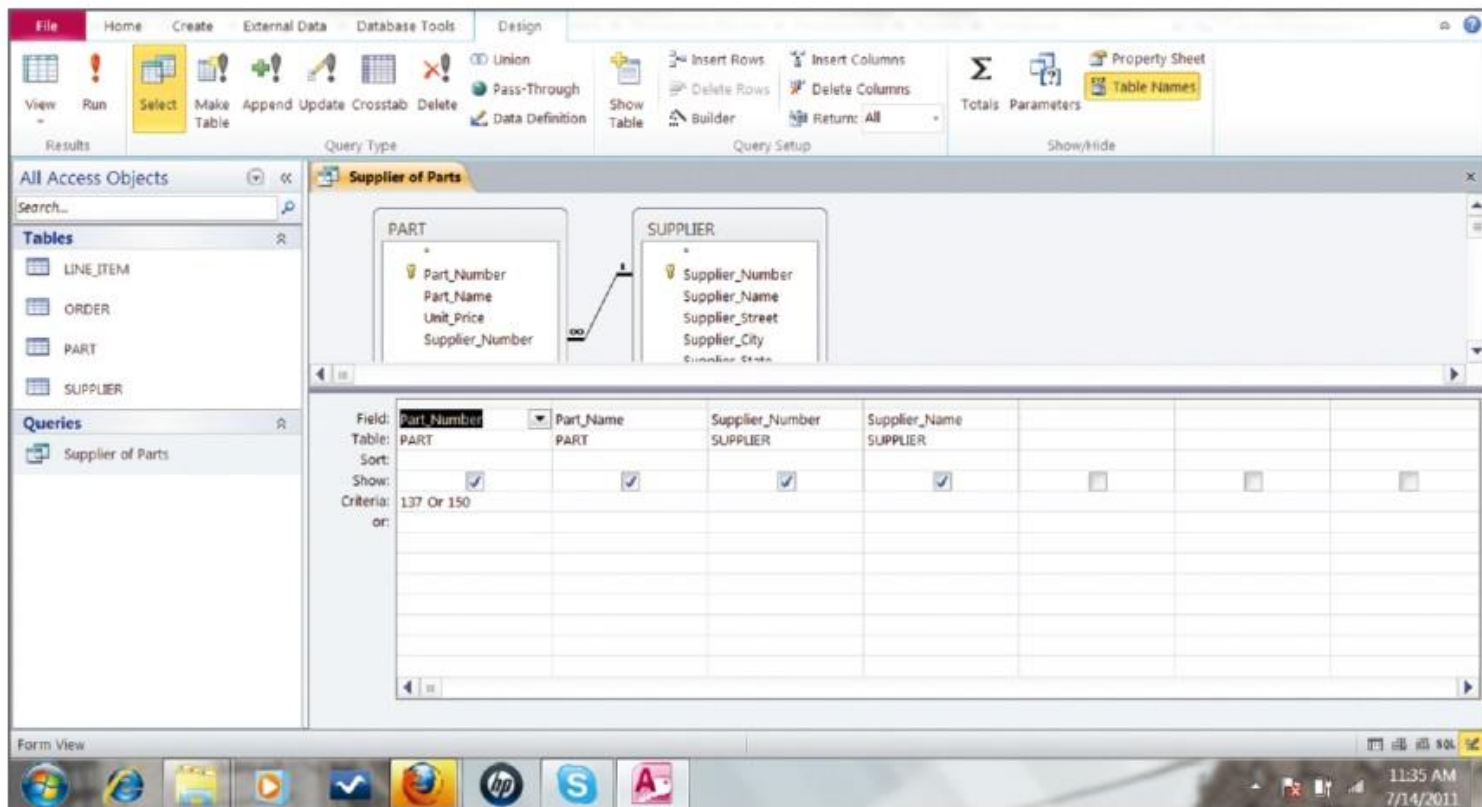
➤ 관계형 DBMS와 기능은?

▪ 관계형 DBMS의 기능:

- 관계
- 세
- . 고
- . 합
- . 보
- DBM
- . 다
- . 연
- . SC
- . DI

그림 6.8 액세스 쿼리

여기서는 그림 6.7의 쿼리가 마이크로소프트 액세스의 쿼리 생성 도구들을 이용할 때 어떻게 만들어질 수 있는지를 보여준다. 이 그림은 쿼리에 필요한 테이블, 필드, 선정 기준을 보여준다.



넓게 결합

회

이용

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

■ 관계

- 관계

- 세

고

합

보

- DBM

다

안

.S

.D

```
SELECT T.[Name], convert(varchar(20),[OrderDate],110) as OrderDate, [SalesOrderID],[CustomerID],  
S.[TerritoryID],[SubTotal],[TaxAmt],[TotalDue]  
FROM Sales.SalesOrderHeader AS S  
INNER JOIN [Sales].[SalesTerritory] AS T  
ON T.TerritoryID =S.TerritoryID
```

관계 결합

100 %

Results

Messages

	Name	OrderDate	SalesOrderID	CustomerID	TerritoryID	SubTotal	TaxAmt	TotalDue
1	Southeast	05-31-2011	43659	29825	5	20565.6206	1971.5149	23153.2339
2	Southeast	05-31-2011	43660	29672	5	1294.2529	124.2483	1457.3288
3	Canada	05-31-2011	43661	29734	6	32726.4786	3153.7696	36865.8012
4	Canada	05-31-2011	43662	29994	6	28832.5289	2775.1646	32474.9324
5	Southwest	05-31-2011	43663	29565	4	419.4589	40.2681	472.3108
6	Northwest	05-31-2011	43664	29898	1	24432.6088	2344.9921	27510.4109
7	Northwest	05-31-2011	43665	29580	1	14352.7713	1375.9427	16158.6961
8	Southwest	05-31-2011	43666	30052	4	5056.4896	486.3747	5694.8564
9	Central	05-31-2011	43667	29974	3	6107.082	586.1203	6876.3649
10	Canada	05-31-2011	43668	29614	6	35944.1562	3461.7654	40487.7233
11	Northwest	05-31-2011	43669	29747	1	714.7043	70.5175	807.2585
12	Central	05-31-2011	43670	29566	3	6122.082	587.5603	6893.2549
13	Northwest	05-31-2011	43671	29890	1	8128.7876	780.8136	9153.6054
14	Canada	05-31-2011	43672	30067	6	6124.182	587.6023	6895.41
15	Northeast	05-31-2011	43673	29844	2	3746.2015	357.9614	4216.0258
16	Canada	05-31-2011	43674	29596	6	2624.382	251.9407	2955.0542
17	Central	05-31-2011	43675	29827	3	5716.3102	546.8759	6434.0848

회

용

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

- 관계형 DBMS의 기능:

```
In [ ]: import pandas as pd
```

```
In [ ]: from fugue_notebook import setup
setup()
```

```
In [ ]: df = pd.DataFrame({'a':[1,2,3,4], 'b':[1,2,3,4]})
df.to_csv('df.csv', index=False)
```

```
In [ ]: %%fsql
-- This SQL cell sees the dataframe defined in the previous cell
SELECT *
  FROM df
 WHERE a > 2
PRINT
```

```
In [ ]: %%fsql
df2 = LOAD "/Users/kevinkho/Work/fugue/df.csv" (header=TRUE, infer_schema=TRUE)
SELECT *
  FROM df2
 WHERE b < 2
PRINT
SAVE OVERWRITE "/Users/kevinkho/Work/fugue/df.csv"
```

```
In [ ]:
```

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

▪ 관계형 DBMS의 기능:

- 정규화(Normalization): 복잡한 데이터 그룹들로부터 작고 안정적이며 유연하고 적응력 있는 데이터 구조들을 생성

그림 6.9 정규화되지 않은 주문 관계

정규화되지 않은 관계는 반복적인 그룹을 포함한다. 예를 들어 각 주문에는 다수의 부품과 공급자들이 있을 수 있다. 주문_번호와 주문_날짜 간에는 일대일 대응관계만이 존재한다.

주문(정규화 이전)

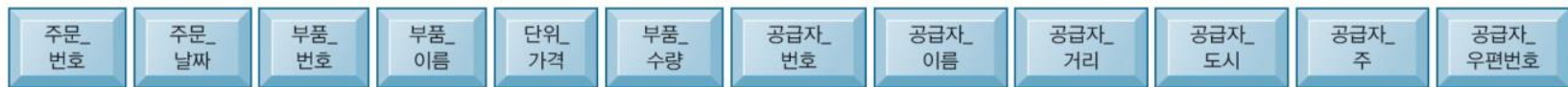
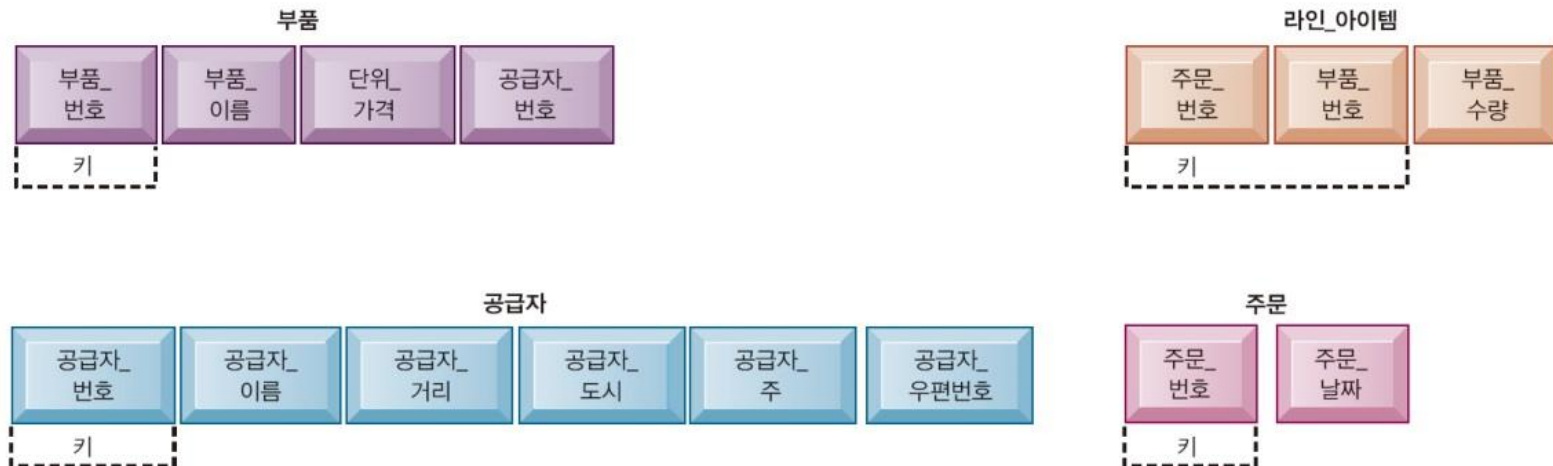


그림 6.10 주문 관계로부터 생성된 정규화된 테이블

정규화를 통해 원래의 주문 관계는 4개의 더 작은 관계로 분할되었다. 주문 관계에는 단 2개의 속성만이 남게 되고, 라인_아이템 관계는 주문_번호와 부품_번호로 구성된 복합 키를 갖는다.



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

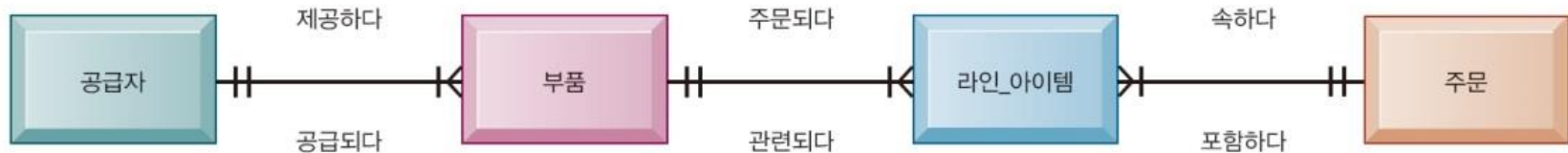
➤ 관계형 DBMS와 기능은?

- 관계형 DBMS의 기능:

- 개체관계성도(Entity-Relationship Diagram): 정규화 테이블 간의 관계를 도식화

그림 6.11 개체관계성도

이 다이어그램은 그림 6.10의 데이터베이스를 모델링하는 데 사용될 수도 있는 공급자, 부품, 라인_아이템, 주문 개체 간의 관계성들을 보여준다.



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

■



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 관계형 DBMS와 기능은?

▪ 관계형 DBMS의 기능:

- 회사가 올바른 데이터 모델을 확보하지 못한다면, 관련 시스템들은 이 회사를 잘 지원할 수 없을 것
- 회사의 시스템은 자신의 성능만큼 효과적이지 못할 것
 - . 시스템들은 부정확하거나 불완전하고 또는 조회하기 어려운 데이터들을 가지고 작업해야 할 것
 - . 49개 주 800여 개의 지점을 보유한 페이머스 풋웨어(Famous Footwear) 신발가게 체인은
“적합한 매장에서 적합한 스타일의 신발을 적합한 가격으로 판매한다.” 목표를 달성 실패
 - . 회사의 데이터베이스가 매장 재고를 신속하게 조정하는 데 적합하도록 설계되어 있지 않았기 때문
 - . 중형 컴퓨터에서 돌아가는 오라클 관계형 데이터베이스를 가지고 있었지만,
이 데이터베이스는 시장의 변화에 대응하기 위해서가 아니라 주로 표준화된 리포트를 생성하도록 설계
 - . 경영진은 각 매장의 재고 목록에 있는 특정 아이템에 관해 정확한 데이터들을 얻을 수 없었음
 - . 분석과 재고관리를 위해 판매/재고 데이터가 잘 구성될 수 있는 새로운 데이터베이스를 구축함으로써 문제 해결

→ 과목의 목적: 조직의 데이터와 이것들이 데이터베이스에서 어떻게 표현되어야 하는지를 이해하는 것

→ 직접적인 비즈니스 프로세스를 현실적으로 이해하는 것!

→ 담당 전문가로서 상황에 대한 대화/의견을 제시할 수 있을 것!

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

▪ 비관계형 데이터베이스:

- 관계형 데이터베이스 기술은 30년 이상 최고의 표준
- 클라우드 컴퓨팅, 전례 없는 데이터량, 웹서비스의 엄청난 작업량, 새로운 유형의 데이터 저장 필요성 등은 테이블, 열, 행과 같은 형태로 데이터를 구성하는 전통적인 관계형 모델에 대한 대안을 필요
- 'NoSQL'이라는 비관계형 데이터베이스로 전환

▪ 비관계형 데이터베이스관리시스템(Non-Relational Database Management System):

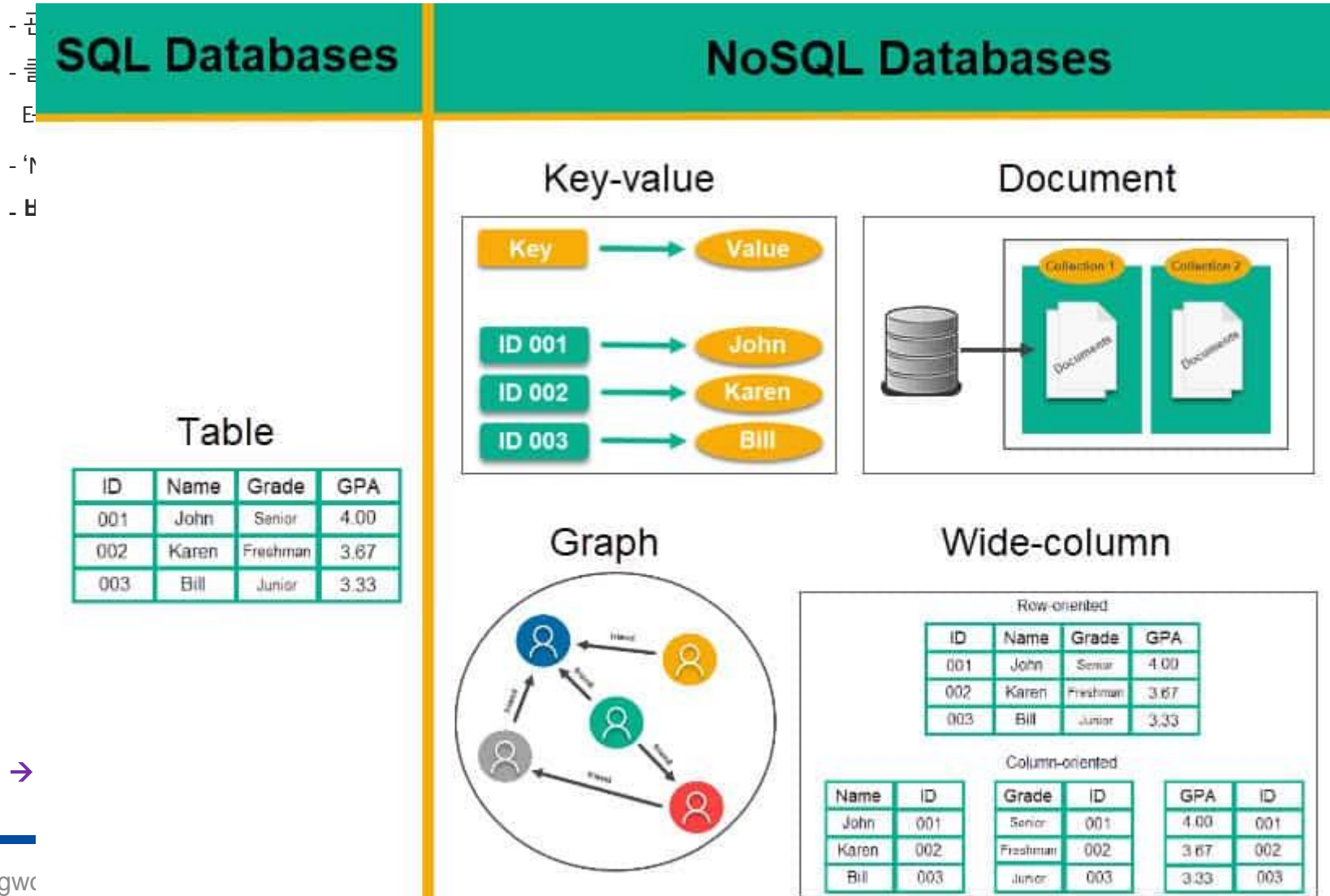
- . 더 유연한 데이터 모델을 활용할 수 있으며, 다수의 하드웨어에 분산되어 있는 대규모 데이터들을 관리할 수 있고 데이터의 규모를 더 수월하게 확장하거나 줄일 수 있도록 설계
- . 기존의 SQL 기반의 도구들로 분석이 어려운 웹, 소셜 미디어, 그래픽, 기타 다양한 형태의 데이터들을 포함한 대용량의 구조적·반구조적 데이터들에 대해 단순하게 쿼리할 수 있도록 해준다는 점에서 유용
- . 미리 정형화된 형태의 데이터베이스 구조를 정의할 필요가 없으며, 나중에 새로운 데이터를 추가할 때 그 데이터의 정의를 변경할 필요가 없음
- . NoSQL 데이터베이스의 종류는 다양한데, 그것들은 각기 나름대로의 기술적 특성과 작동방식
- . 오라클의 NoSQL Database와 클라우드에서 작동되는 아마존 웹서비스 중 하나인 아마존의 SimpleDB
- . 메트라이프(MetLife)의 MongoDB는 오픈소스 NoSQL 데이터베이스는 70개 이상의 분리된 관리업무용 시스템들과 보상금 청구 시스템 그리고 여타의 데이터 원천들로부터 다양한 형태의 데이터(건강/사망에 관한 이미지와 같이 반구조적이고 비구조적인 데이터들을 포함) 활용

→ NoSQL 은 지루하고, 비용/시간이 많이 드는 데이터베이스 매핑 작업 없이 구조적·반구조적·비구조적인 정보 모두 다룸

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

▪ 비관계형 데이터베이스:



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

■ 비관계형 데이터베이스:

- 관계

- 클라

데이

- 'NoS

- 비관

. 더

데

. 기

대

. 미

나

. No

. 오

. 메

보

빈

➔ No

mySQL

```
SELECT
  Dim1, Dim2,
  SUM(Measure1) AS MSum,
  COUNT(*) AS RecordCount,
  AVG(Measure2) AS MAvg,
  MIN(Measure1) AS MMin
  MAX(CASE
    WHEN Measure2 < 100
    THEN Measure2
  END) AS MMax
FROM DenormAggTable
WHERE (Filter1 IN ('A','B'))
  AND (Filter2 = 'C')
  AND (Filter3 > 123)
GROUP BY Dim1, Dim2
HAVING (MMin > 0)
ORDER BY RecordCount DESC
LIMIT 4, 8
```

- ① Grouped dimension columns are pulled out as keys in the map function, reducing the size of the working set.
- ② Measures must be manually aggregated.
- ③ Aggregates depending on record counts must wait until finalization.
- ④ Measures can use procedural logic.
- ⑤ Filters have an ORM/ActiveRecord-looking style.
- ⑥ Aggregate filtering must be applied to the result set, not in the map/reduce.
- ⑦ Ascending: 1; Descending: -1

MongoDB

```
db.runCommand({
  mapreduce: "DenormAggCollection",
  query: {
    filter1: { '$in': [ 'A', 'B' ] },
    filter2: 'C',
    filter3: { '$gt': 123 }
  },
  map: function() { emit(
    { d1: this.Dim1, d2: this.Dim2 },
    { msum: this.measure1, recs: 1, mmin: this.measure1,
      mmax: this.measure2 < 100 ? this.measure2 : 0 }
  ); },
  reduce: function(key, vals) {
    var ret = { msum: 0, recs: 0, mmin: 0, mmax: 0 };
    for(var i = 0; i < vals.length; i++) {
      ret.msum += vals[i].msum;
      ret.recs += vals[i].recs;
      if(vals[i].mmin < ret.mmin) ret.mmin = vals[i].mmin;
      if((vals[i].mmax < 100) && (vals[i].mmax > ret.mmax))
        ret.mmax = vals[i].mmax;
    }
    return ret;
  },
  finalize: function(key, val) {
    val.mavg = val.msum / val.recs;
    return val;
  },
  out: 'result1',
  verbose: true
});
db.result1.find({ mmin: { '$gt': 0 } }).
  sort({ recs: -1 }).
  skip(4).
  limit(8);
```

다름

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

▪ 클라우드 데이터베이스(Cloud Database):

- 아마존과 여타의 클라우드 컴퓨팅 벤더들이 제공하는 서비스 중에는 관계형 데이터베이스 엔진
 - . 아마존 RDS(Relational Database Service)는 MySQL, MS SQL 서버, 오라클 데이터베이스, PostgreSQL, 아마존 오로라 제공
 - . 오라클은 관계형 오라클 데이터베이스를 사용하는 데이터베이스 클라우드 서비스를 제공
 - . 마이크로소프트는 MS SQL 서버 DBMS를 기반 MS Azure SQL 데이터베이스라는 클라우드 기반 관계형 데이터베이스 제공

▪ 분산 데이터베이스(Distributed Database):

- 여러 물리적 위치에 저장된 데이터베이스
- 데이터베이스의 일부분/복사본들이 한 위치에 물리적으로 저장, 다른 부분들이나 다른 복사본들은 다른 위치서 유지관리
 - . 구글은 현재 스패너(Spanner)라고 하는 분산 데이터베이스 기술을 클라우드 서비스로 제공
 - . 스패너는 전 세계 수백 개의 데이터 센터에 있는 수백만 대의 컴퓨터에 정보를 저장할 수 있게 해주며, 특별한 시간관리 도구를 사용하여 모든 장소에 있는 데이터들이 정확하게 동기화되고 항상 일치성을 유지
 - . 구글은 스패너를 이용 구글 포토, 애드워즈(AdWords, 구글의 온라인 광고 시스템), 지메일 등 다양한 클라우드 서비스
 - . 글로벌 비즈니스 운영을 위해 이와 같은 서비스 기능들을 필요로 할 수 있는 기업들에게 스패너 기술을 판매

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

클라우드 데이터베이스(Cloud Database):

AWS

과 여타의 클라우드 컴퓨팅 벤더들이 제공하는 서비스 중에는 관계형 데이터베이스 엔진

수 집

AWS IoT



Kinesis Streams



Kinesis Firehose



GLUE



Database Migration Service



저 장

S3



RedShift



Glacier



DynamoDB



RDS



분 석

Elastic search



Machine Learning



Kinesis Analytics



R Studio



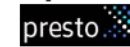
python



EMR



Spark



시 각 화

kibana



QuickSight



tableau



MicroStrategy



looker



Qlik Sense

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

- 클라우드 데이터베이스(Cloud Database):



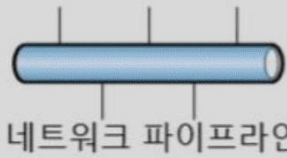
보안



보안 그룹

네트워크 ACL

AWS IAM



네트워킹



Elastic Load Balancing



Amazon VPC



서버



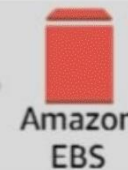
AMI



Amazon EC2 인스턴스



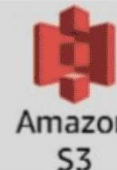
스토리지 및
데이터베이스



Amazon EBS



Amazon EFS



Amazon S3



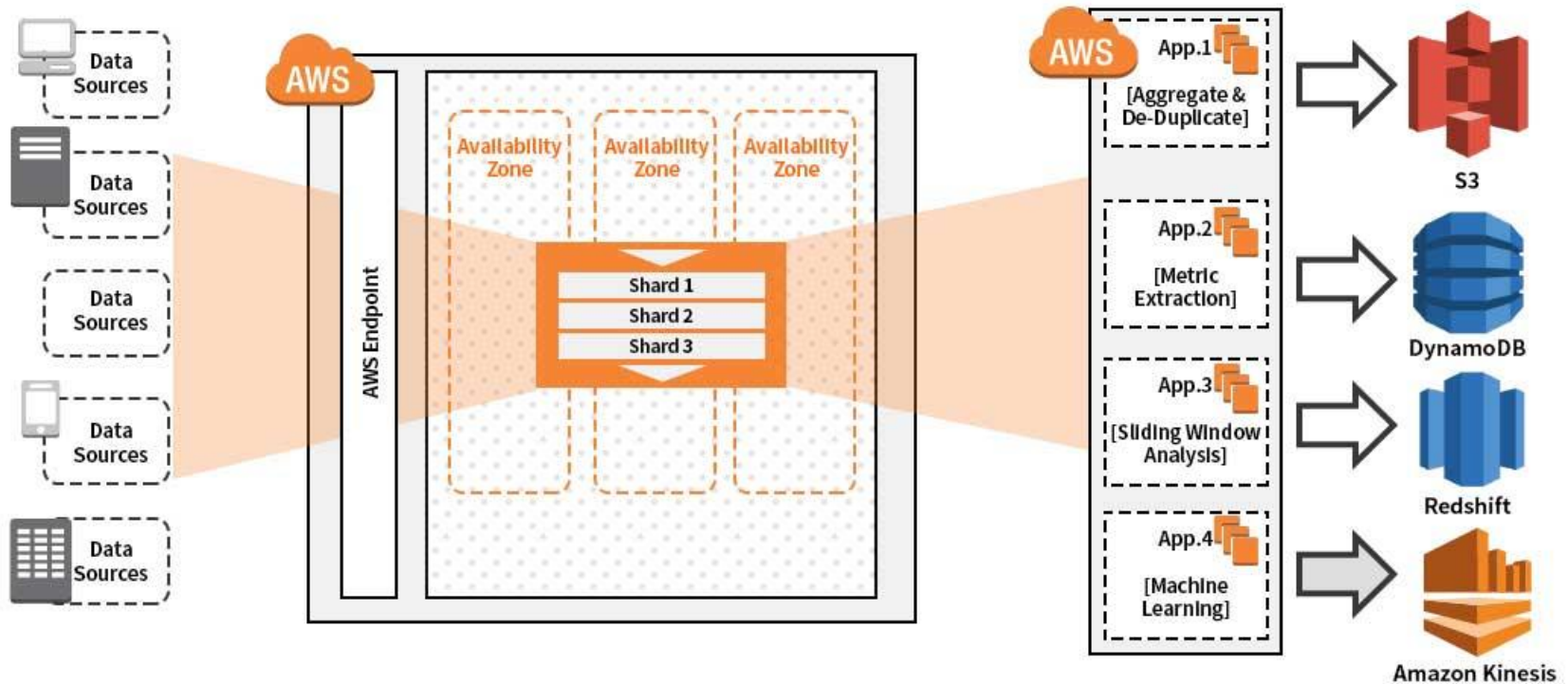
Amazon RDS

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

▪ 클라우드 데이터베이스(Cloud Database):

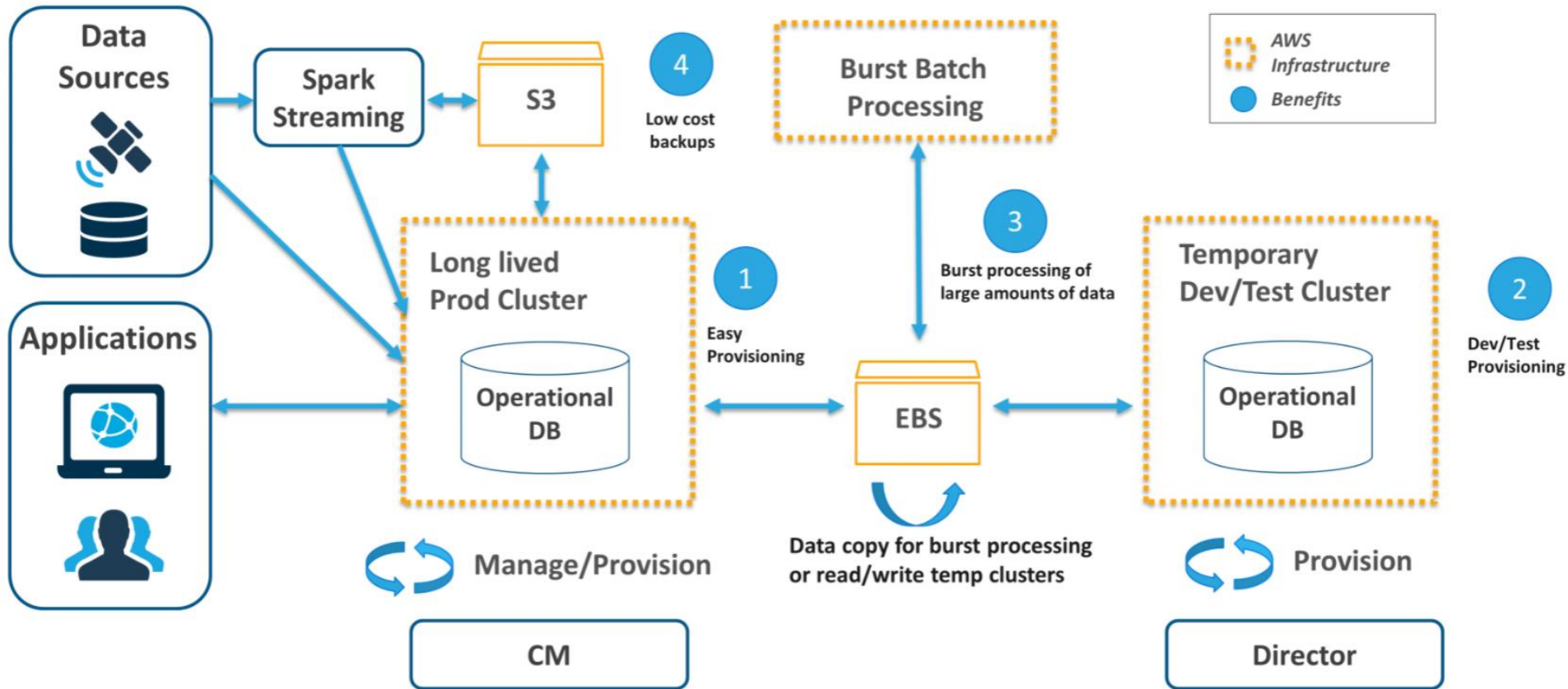
이러한 클라우드 데이터베이스는 클라우드 컴퓨팅 인프라를 기반으로 하는 데이터베이스로, 관계형 데이터베이스와 달리



성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

Operational Database Cloud Architecture



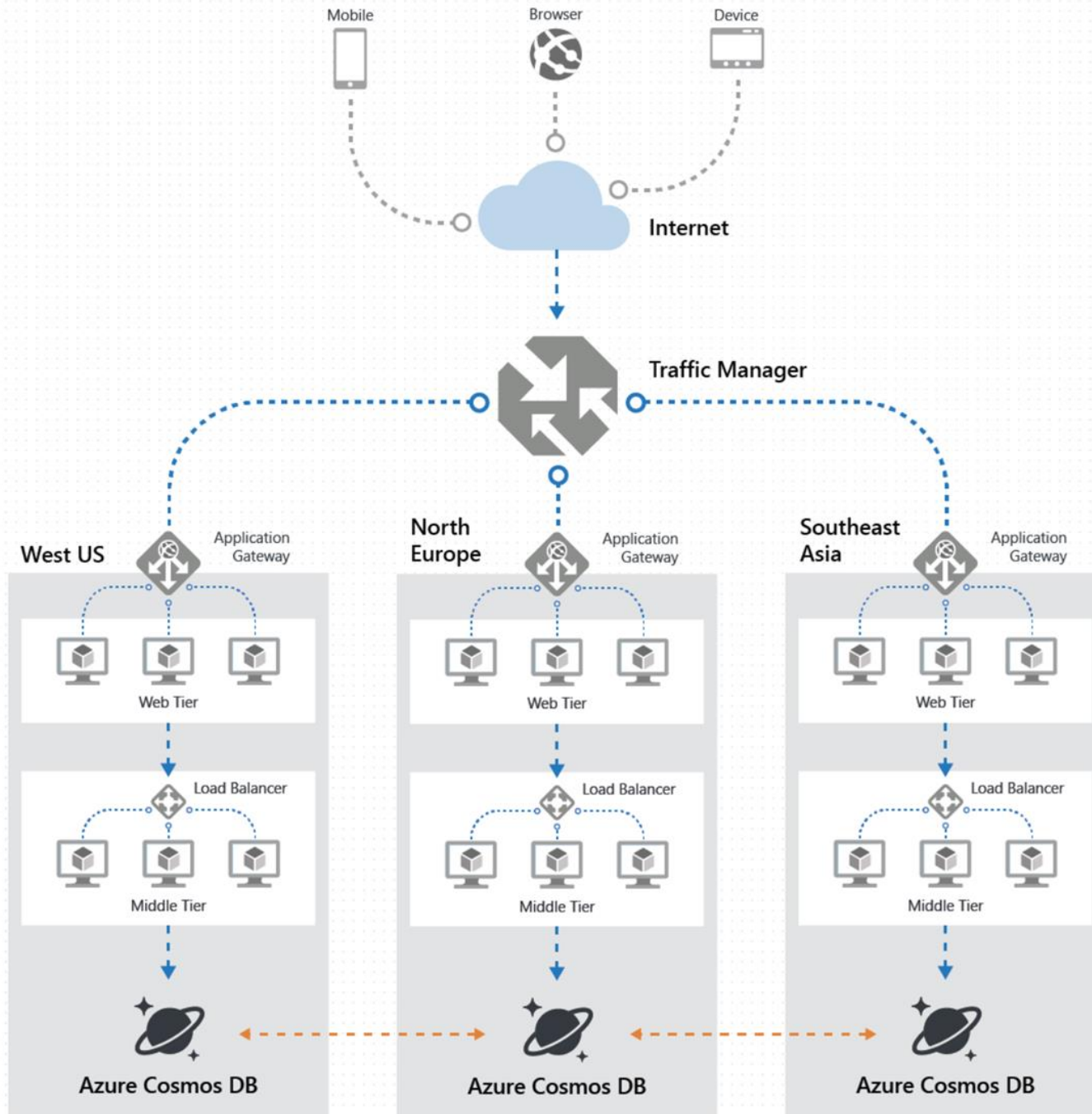
비관기

클라우드

- 아마존
- 아
- 오
- 마

분산

- 여러
- 데이
- 구
- 스
- 특
- 구
- 클



제공

서비스 제공

유지관리

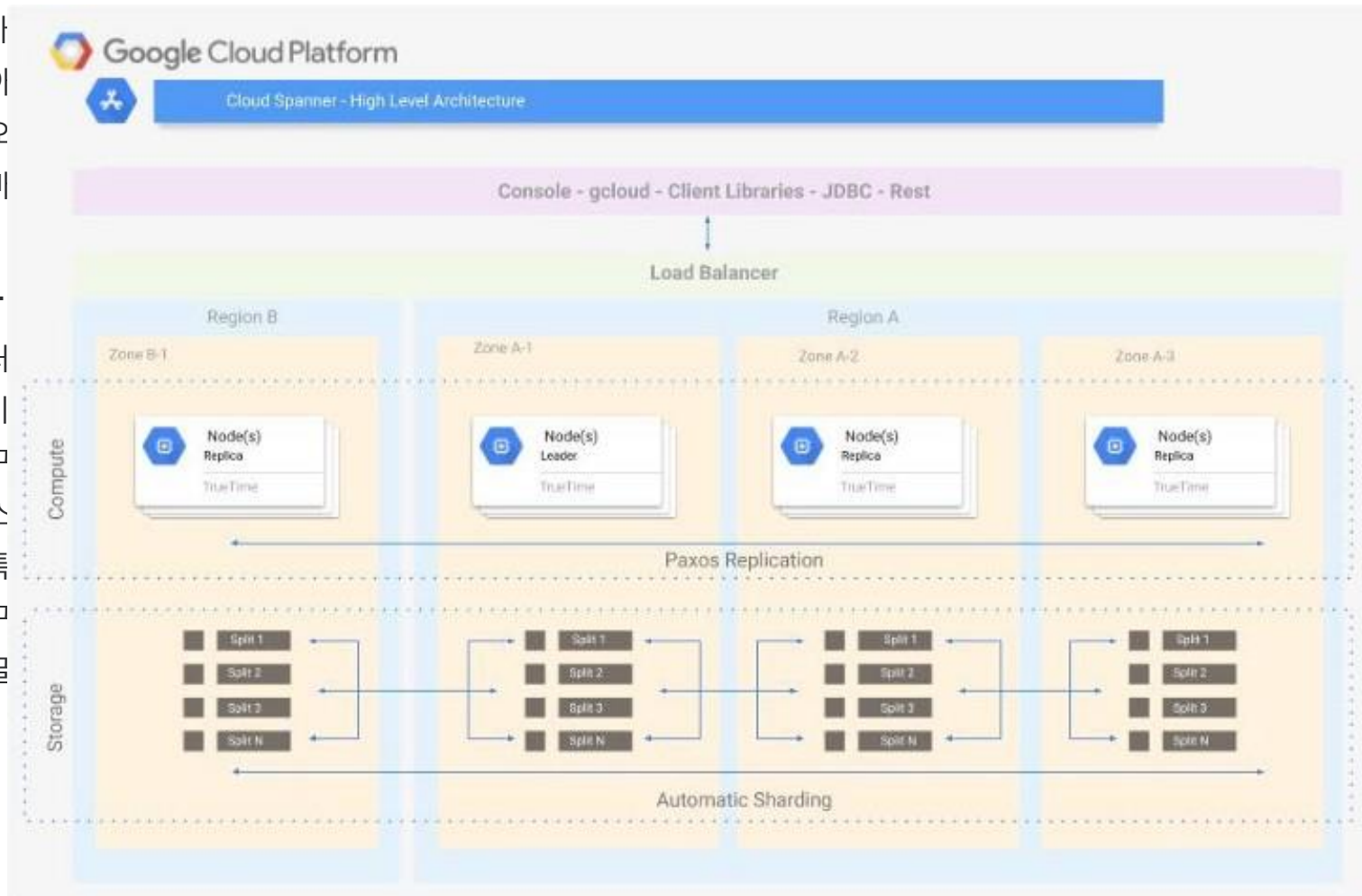
서비스

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

■ 클라우드 데이터베이스(Cloud Database):

- 아마
.이
.오
.미



제공

이스 제공

■ 분산

- 여러
- 데이
.구
.스
특
.구
닐

지관리

서비스

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

▪ 블록체인(Blockchain):

- 기업과 조직들이 중앙집중적인 권한 없이 거의 즉시 네트워크에서 거래를 생성/확인할 수 있는 분산 데이터베이스 기술
 - . 블록체인은 지속적으로 늘어나는 기록 목록을 보유하는데, 각각의 기록은 블록으로 정의
 - . 각 블록은 타임스탬프를 포함하며, 이전 블록과 연결
 - . 데이터 블록은 블록체인 원장에 한 번 기록되고 나면, 소급해서 변경할 수 없음
 - . 거래를 컴퓨터 네트워크의 여러 컴퓨터에 분산 원장으로 저장하고 저장된 정보는 네트워크 컴퓨터들이 지속 조정
 - . 스마트 계약(Smart Contracts) 사용을 통해 기록 거래들을 모두 표준화
 - . 기업 간 거래 관리 규칙을 구현한 프로그램(예: 제품가격, 배송 방법, 거래 완료 시기, 거래 자금 제공자, 자금조달 등)
 - . 누군가 거래를 추가하면, 기존 참가자들(기존 블록체인 사본 보유)은 제시된 거래를 평가/확인 위해 알고리즘 실행
 - . 원장에 대한 정당한 변경사항은 몇 초 또는 몇 분 안에 블록체인에 기록되며 암호화를 통해 보호
 - . 이러한 요소들은 타당한 행위자들만 정보를 입력할 수 있고, 검증된 거래만 허용

→ 블록체인은 사용자 검증하고 거래 유효성 인증하는 비용과 많은 기업들에 걸친 거래 정보 저장/처리 관련 위험 대폭 줄여

→ 블록체인 시스템을 기업들에게 활용 가능하고 매력적으로 만드는 요인은 암호화와 행위자 및 참여 기업들에 대한 인증

→ 수많은 기업들은 자체 거래시스템을 구축하고, 그것을 공급업체, 배송업체, 금융기관의 시스템과 통합해 왔지만,

블록체인은 그렇게 하는 대신 참여 기업들을 위한 하나의 간단한 저비용 거래시스템을 제공

→ 블록체인의 단순성과 보안성은 금융 거래, 공급망 거래, 의료 기록 등 여러 유형 데이터들을 저장/보호하는 데 매력적

→ 블록체인은 비트코인, 이더리움을 비롯한 여타 암호 화폐의 토대가 되는 기술

비관

블록

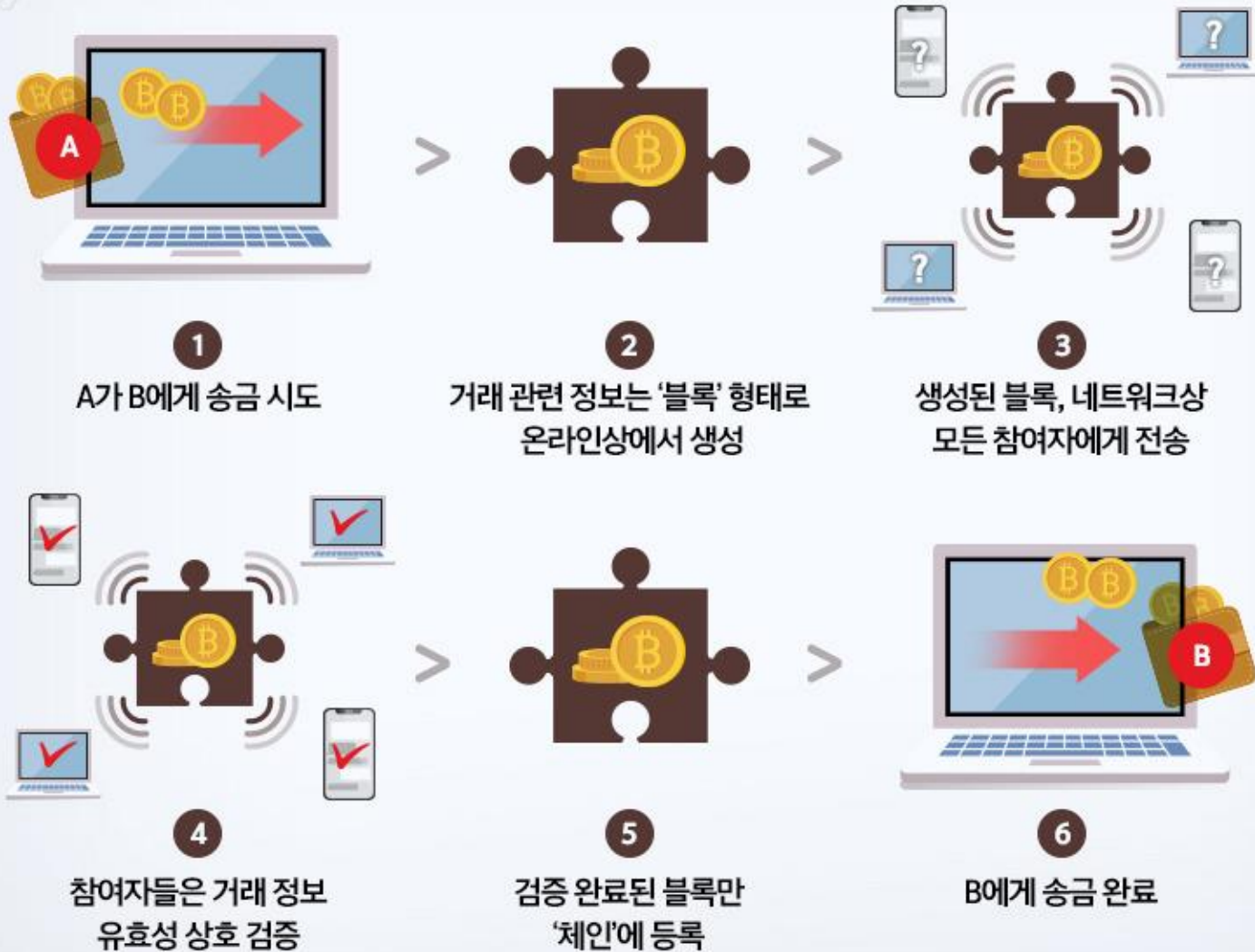
- 기업

블록
체인
기술
의
장
점
과
단
점

블록
체인
기술
의
장
점
과
단
점

→ 블록
→ 블록
→ 수
블
→ 블록
→ 블록

블록 체인 거래 과정



기술

성

달 등)
행

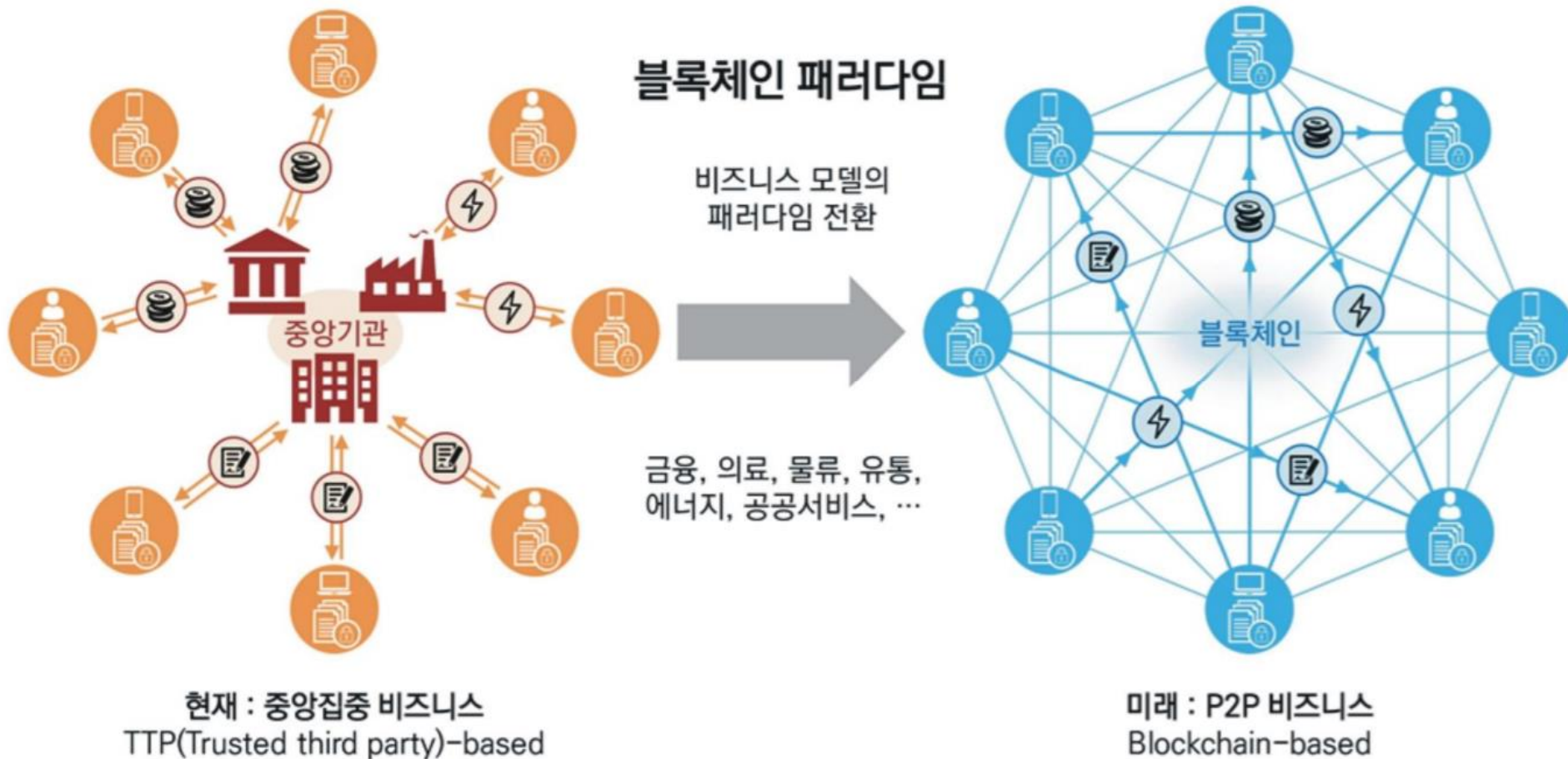
폭 줄여
· 인증

목적

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

▪ 블록체인(Blockchain):



[그림 1] 블록체인 패러다임 전환 : 중앙집중 비즈니스에서 P2P 비즈니스로의 혁신

성공적인 비즈니스 인텔리전스를 위한 데이터 관리 방향은?

➤ 비관계형 데이터베이스, 클라우드 데이터베이스, 블록체인이란?

■ 블록체인(Blockchain):

- 기업

. 블

. 각

. 데

. 거

. 스

. .

. 누

. 원

기존 중앙 집중형 콘텐츠 유통 플랫폼



: 기술

정

(달 등)

행

블록체인 기반 분산형 콘텐츠 유통 플랫폼



대폭 줄여

한 인증

력적

장점: 1. 공정 거래 2. 다양한 지불 방식 지원 3. 사이버 보안 우수 4. 운영 비용 없음

INCHEON NATIONAL UNIVERSITY

THANK YOU

