

Chapter 01.

데이터베이스 시스템의 세계

서동민

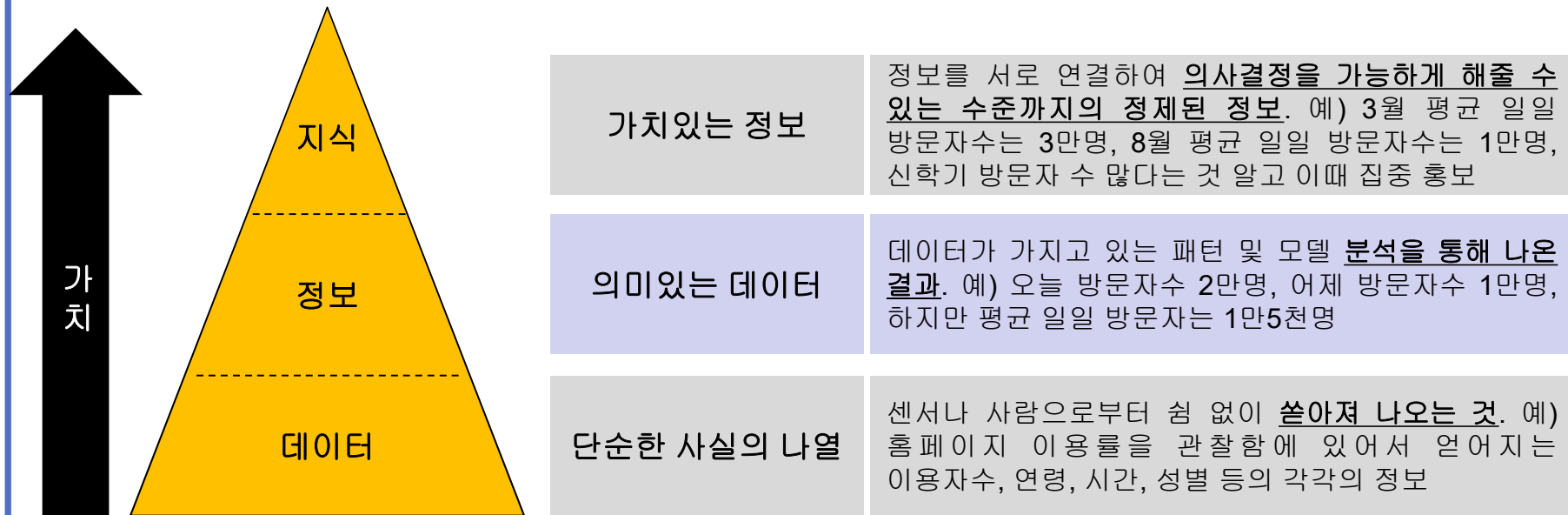
한국과학기술정보연구원

차례

- ❑ 강의 소개
- ❑ 데이터베이스 필요성
- ❑ 데이터베이스 시스템의 발전
- ❑ 데이터베이스 시스템 스터디의 개요
- ❑ MariaDB 설치 및 테스트

데이터베이스 필요성 (1/11)

□ 데이터와 정보의 차이점



강의 소개 (1/3)

교과목명	국문	데이터베이스 시스템 특론
	영문	Advanced Topics in Database System
개설학기	<input checked="" type="checkbox"/> 2022년 1학기	
학점/시수	<input checked="" type="checkbox"/> 3학점 3시수	
강의언어	<input checked="" type="checkbox"/> 한국어강의 <input type="checkbox"/> 영어강의	
화상강의	<input checked="" type="checkbox"/> 오프라인 강의 <input checked="" type="checkbox"/> 원격(화상)강의	
성적평가방식	<input type="checkbox"/> S/U(성패제) <input checked="" type="checkbox"/> Grade(등급제)	
강의요일 및 시간	금요일 10시-13시 (3시간)	
강의실	한국과학기술정보연구원(KISTI) 키움관 세미나2실/온라인(Zoom 이용)	
수업목표	<ul style="list-style-type: none"> - 관계형 데이터베이스 설계 및 활용 능력을 습득한다. - 관계형 데이터베이스 구조 및 성능 최적화 관련 지식을 습득한다. - 빅데이터 및 빅데이터 처리를 위한 NoSQL 기술에 대한 지식 및 활용 능력을 습득한다. 	
수업운영방식	<ul style="list-style-type: none"> - 협동강의(서동민, 이경하, 박경석) - 오프라인/온라인 병행 수업 운영 	
성적평가 세부항목	출석: 20%, 과제: 30%, 기말시험: 50%	
교재 및 참고문헌	강의 슬라이드 및 실습자료 제공 (강의내에 참고자료 소개) 참고문헌: Database System Concepts, 7 th Ed, A. Silberschartz et al., McGraw-Hill	

강의 소개 (2/3)

주별	내용	비고
1	• 강좌 소개 • 데이터베이스 개요 및 모델링 (1) - 데이터베이스 필요성 및 역사 - E/R 데이터 모델링	서동민
2	• 데이터베이스 모델링 (2) - E/R 데이터 모델링 - 관계 모델링	서동민
3	• 데이터베이스 모델링 (3) - 관계 모델링 - 관계형 데이터베이스 설계	서동민
4	• 데이터베이스 언어 SQL (1) - 데이터베이스와 테이블 생성 - 단순 질의	서동민
5	• 데이터베이스 언어 SQL (2) - 부질의	서동민
6	• 데이터베이스 언어 SQL (3) - 연산 및 뷰 - 파이썬에서 SQL 활용	서동민

7	• 저장 장치 관리 및 인덱싱 (1) - 물리적 저장장치 - 파일 시스템과 DB 저장 구조	이경하
8	• 저장 장치 관리 및 인덱싱 (2) - 데이터 인덱싱 - B+, Bitmap, Hash 등 주요 인덱스 구조	이경하
9	• 질의 처리와 최적화 - 질의 처리 개념 및 원리 - 질의 처리 최적화	이경하
10	• 트랜잭션 관리 - 트랜잭션 개념 및 원리 - 동시성 제어 원리	이경하
11	• 복구 시스템 - 장애 유형과 회복 기법 - DBMS의 복구 시스템 원리	이경하

12	• 빅데이터 기술 개요 및 이론 - 빅데이터 기반 기술 주요 구성 요소 - 빅데이터 처리와 분석을 위한 주요 이론	박경석
13	• 하둡과 Spark를 이용한 빅데이터 응용 및 분석 - 하둡/Spark 기술 개요 및 주요 이론 - 하둡/Spark를 이용한 빅데이터 처리와 분석	박경석
14	• NoSQL을 이용한 빅데이터 응용 및 분석 - NoSQL 기술 개요 및 주요 이론 - NoSQL을 이용한 빅데이터 처리와 분석	박경석
15	• 빅데이터 기술을 이용한 고급 분석 기술 응용 - 빅데이터 기반 기술을 이용한 기계학습 분석 - 빅데이터 최신 이론 및 응용 사례	박경석
16	기말고사	박경석

강의 소개 (3/3)

<https://github.com/bart7449/lecture2022a>

The screenshot shows the GitHub interface for the repository 'bart7449/lecture2022a'. The repository is public and has 1 watch, 0 forks, and 0 stars. The main content area displays a commit history table with columns for the commit message, author, and time. The most recent commit is 'Delete 강좌_01(20220301,05,서동민)-Window-제출용.ppt' by 'Dongmin-76' 2 minutes ago. Below the commit history, the 'README.md' file is selected, showing the title 'Advances in Database Systems(Spring 2022)' and a paragraph of text. The right sidebar contains sections for 'About' (Advances in Database Systems), 'Releases' (No releases published), and 'Packages' (No packages published).

Commit Message	Author	Time
Delete 강좌_01(20220301,05,서동민)-Window-제출용.ppt	Dongmin-76	2 minutes ago
LICENSE	Initial commit	10 hours ago
README.md	Update README.md	5 hours ago

Advances in Database Systems(Spring 2022)

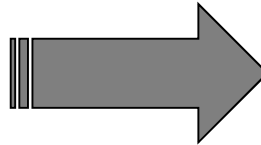
Database systems have become so crucial that it is difficult to think the absence of databases in application developments. In addition, database programming has become one of the basic knowledge that program developers MUST know. However, many developers simply understand and exploit the database system as a means of storing tabular data. In this lecture, students will acquire knowledge about database design techniques and the internals of the database for effective uses. In addition, a big data system for efficiently storing and utilizing a massive volume of data that is difficult to process within a single sited DBMS and how to use it will be studied.

데이터베이스 필요성 (2/11)

□ 데이터베이스 (Database)

- 서로 연관된 데이터의 모임
- DBMS에 의해 관리되는 데이터의 모임
 - 어떤 특정한 의미를 가지는 데이터의 모임
 - 특정한 목적을 위해서 설계, 구축, 운용

※ DBMS (Database Management System)



데이터베이스 필요성 (3/11)

□ 데이터베이스 관리 시스템 (DBMS)

- 또는 간단히, 데이터베이스 시스템
- 사용자가 데이터베이스를 생성하고 관리할 수 있도록 편리한 기능을 제공하는 프로그램들의 모임
 - 다양한 응용에서 필요로 하는 데이터베이스를 정의, 생성, 조작할 수 있도록 편리한 기능을 제공하는 범용 소프트웨어 시스템
- 방대한 데이터의 집합체를 유지관리하고 이용하는데 도움을 주도록 설계된 소프트웨어

데이터베이스 필요성 (4/11)

□ 파일 시스템 개념 및 단점

2018001	홍길동	빅데이터과학	서울	hgd@gmail.com	010-123-1234	...
2018002	이순신	빅데이터과학	대전	lss@naver.com	010-234-2345	...
2018003	박순심	과학기술정책	부산	pss@hatmail.com	010-345-3456	...

.....

<A학교의 학생 신상 정보를 저장하고 있는 학생신상.txt 파일>



질문 1) naver 메일을 사용하는 학생의 이름은?

질문 2) 지역별 학생들 영어 평균 성적은?

수정 1) 학생신상.txt 파일에 우편번호 데이터가 추가된다면?

수정 2) 대전 거주하는 학생들의 영어 점수를 5점씩 올려줘야 한다면?

2018001	홍길동	빅데이터과학	국어:80	영어:70	수학:90	...
2018002	이순신	빅데이터과학	국어:88	영어:90	수학:70	...
2018003	박순심	과학기술정책	국어:68	영어:85	수학:60	...

.....

<A학교의 학생 성적 정보를 저장하고 있는 학생성적.txt 파일>



데이터베이스 필요성 (5/11)

□ 파일 시스템 개념 및 단점

■ 파일 시스템 개념

- 파일 시스템은 **DBMS**가 등장하지 않았을 때인 **1960년대부터** 사용
- 파일의 기본적인 구성요소는 순차적인 레코드들
 - 한 레코드는 연관된 필드들의 모임
- 파일 시스템을 사용하는 경우에는 일반적으로 각각의 응용 프로그램마다 별도의 파일을 유지
- 프로그래머는 데이터의 논리적 구조뿐만 아니라 데이터의 물리적인 구조도 프로그램에 표현

2018001	홍길동	빅데이터과학	hgd@gmail.com
2018002	이순신	빅데이터과학	lss@naver.com
2018003	박순심	과학기술정책	pss@hatmail.com

.....

<파일의 예>



<응용 프로그램>

데이터베이스 필요성 (6/11)

□ 파일 시스템 개념 및 단점

■ 파일 시스템 단점

- 데이터가 많은 파일에 중복해서 저장됨
- 다수 사용자들을 위한 동시성 제어가 제공되지 않음
- 파일을 검색하거나 갱신하는 절차가 상대적으로 복잡하기 때문에 프로그래머의 생산성이 낮음
- 회복 기능이 없음
- 프로그램-데이터 독립성이 없으므로 유지보수 비용이 큼
- 융통성이 부족함 (여러 파일로부터 원하는 데이터 검색 어려움)

EMPLOYEE 파일 (인사 관리 프로그램용)

NAME JUMIN-NO DEPARTMENT ... ADDRESS ...
--

ENROLLMENT 파일 (교육 관리 프로그램용)

NAME JUMIN-NO DEPARTMENT ... COURSE ...

<파일 시스템 단점 예>

데이터베이스 필요성 (7/11)

❑ 데이터베이스 관리 시스템 (DBMS) 구조



데이터베이스 필요성 (8/11)

- 데이터베이스 관리 시스템의 특징 (1/2)
 - 고수준 질의 언어 (high level query language)
 - 트랜잭션 관리 (transaction management)
 - 병렬수행 제어(concurrency control)와 회복(recovery)
 - 지속적 저장 시스템 (persistent storage system)
 - 대용량 데이터 저장
 - 데이터에 효율적인 접근 방법
 - 데이터 모델 (data model)
 - 데이터베이스의 논리적인 구조를 명시
 - 접근 제어 (access control)
 - 권한이 없는 사용자로부터 데이터베이스를 보호



데이터베이스 필요성 (9/11)

□ 데이터베이스 관리 시스템의 특징 (2/2)

■ 중복성의 제어 (redundancy control)

■ 중복성의 문제

- 한번의 변경이 중복된 횟수만큼 반복해서 수행
- 메모리의 낭비
- 데이터의 불일치

- 데이터의 일관성을 보장하기 위해 오직 한번만 저장

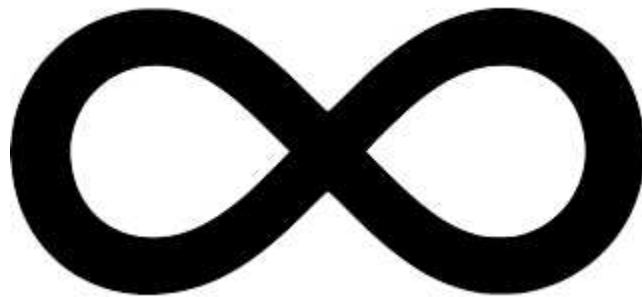
■ 무결성 제약조건

- 각 데이터 항목에 대한 데이터 타입
- 한 테이블의 레코드가 반드시 다른 테이블의 레코드들과 연관성을 가져야 함
- 데이터 항목의 유일 값



데이터베이스 필요성 (10/11)

- 데이터베이스 관리 시스템의 주요 응용
 - 항공 예약 시스템 (airline reservation system)
 - 은행 시스템 (banking system)
 - 기업 업무 (corporate applications)



데이터베이스 필요성 (11/11)

□ 파일 시스템과 DBMS 방식의 비교

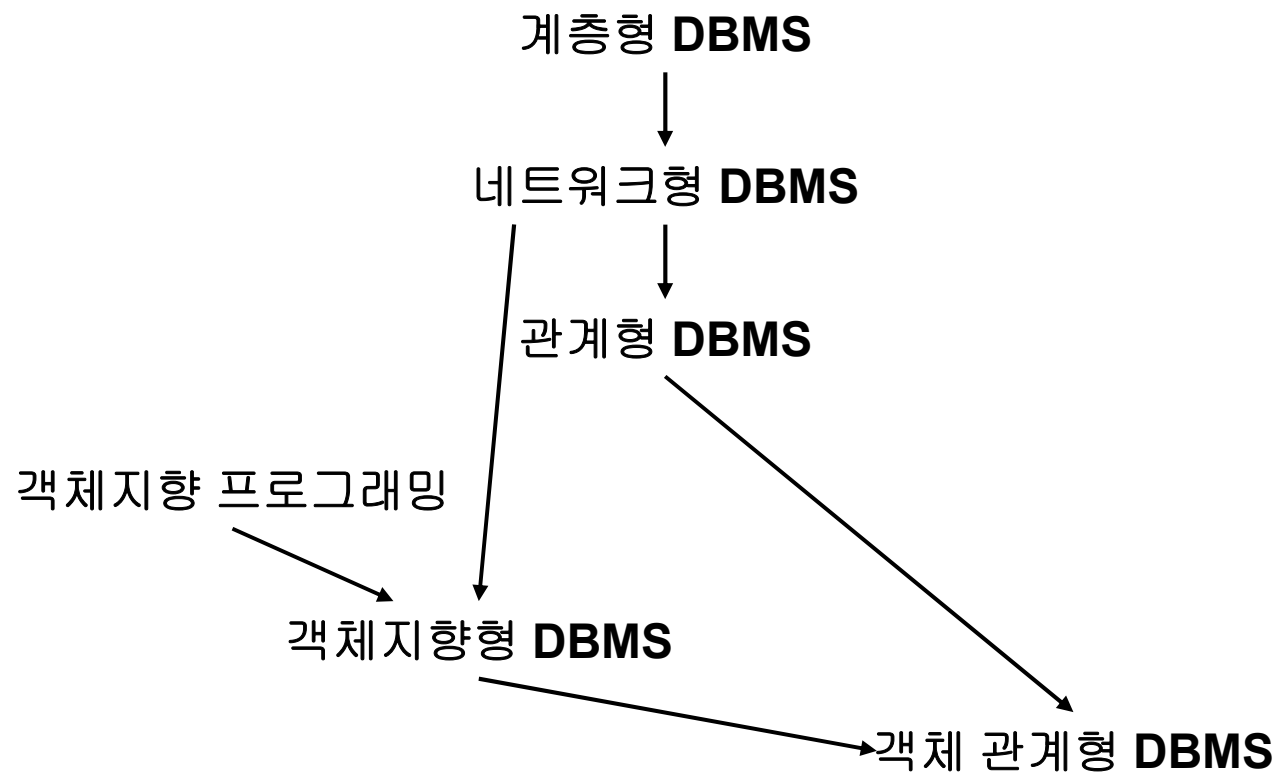
파일 시스템 방식	DBMS 방식
<ul style="list-style-type: none">• 데이터에 대한 물리적 접근만 조정한다.	<ul style="list-style-type: none">• 데이터에 대한 물리적 접근과 논리적인 접근을 모두 조정한다.
<ul style="list-style-type: none">• 동일한 파일을 두 개 이상의 프로그램이 동시에 접근할 수 없다.	<ul style="list-style-type: none">• 동일한 데이터를 다수 사용자가 동시에 접근할 수 있다.
<ul style="list-style-type: none">• 데이터가 비구조적이며, 중복성과 유지보수 비용이 높다.	<ul style="list-style-type: none">• 데이터가 구조화되어 있으며, 중복성과 유지보수 비용이 낮다.
<ul style="list-style-type: none">• 어떤 프로그램이 기록한 데이터는 다른 프로그램에서 읽을 수 없는 경우가 많다.	<ul style="list-style-type: none">• 접근 권한이 있는 모든 프로그램이 데이터를 공유한다.
<ul style="list-style-type: none">• 데이터에 대한 접근은 미리 작성된 프로그램을 통해서만 가능하다.	<ul style="list-style-type: none">• 질의어를 사용하여 데이터에 대한 융통성 있는 접근이 가능하다.
<ul style="list-style-type: none">• 각 응용 프로그램마다 파일이 따로 있으므로 데이터가 통합되어 있지 않다.	<ul style="list-style-type: none">• 데이터가 중복을 배제하면서 통합되어 있다.

데이터베이스 시스템의 발전 (1/10)

- 초기 데이터베이스 관리 시스템
 - 1960년대 후반에 최초의 상용 DBMS 등장
 - 파일 시스템으로부터 발전
 - 계층 모델, 네트워크 모델과 같은 여러 종류의 데이터 모델
 - 고수준 질의 언어 지원 못함
 - 1960년 초반 General Electric 사에서 Integrated Data Store를 개발
 - 네트워크 데이터 모델의 기초
 - 1960년 말 IBM은 IMS (Information Management System)를 개발
 - 계층 데이터 모델

데이터베이스 시스템의 발전 (2/10)

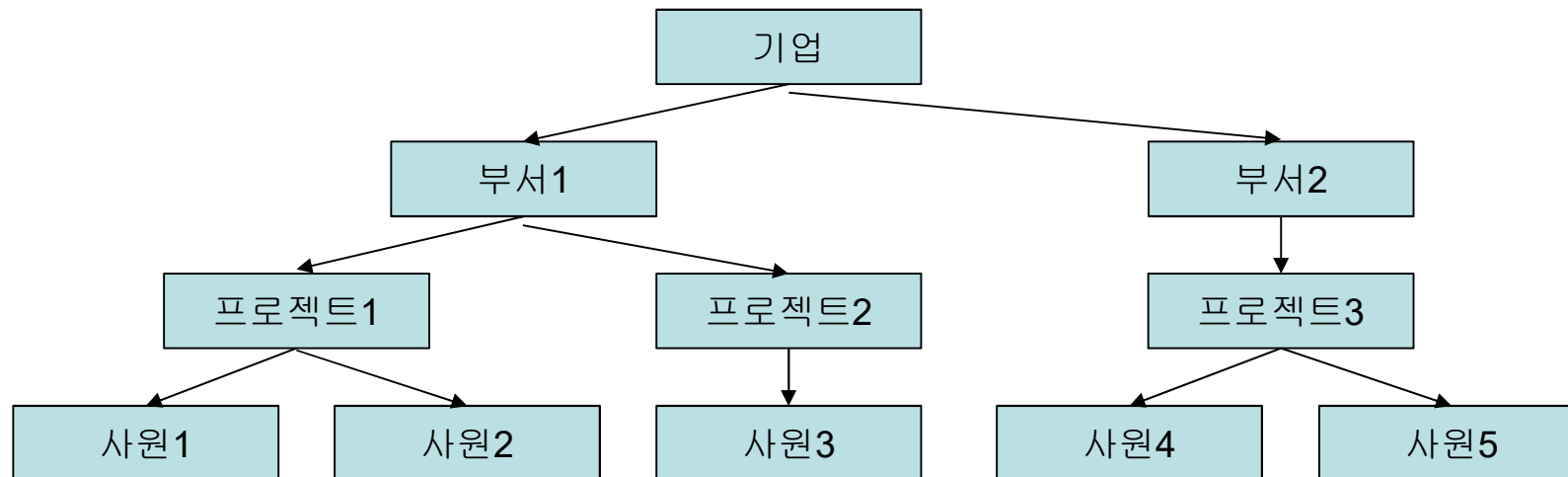
□ DBMS 발전과정



데이터베이스 시스템의 발전 (3/10)

□ 계층형 DBMS

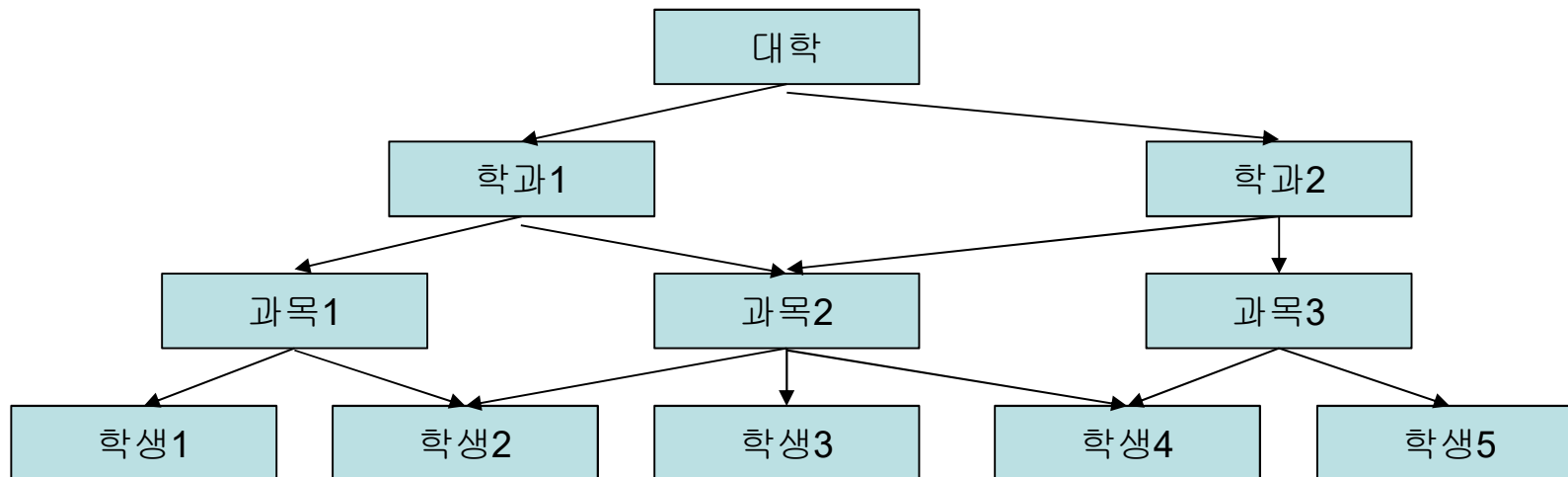
- 트리 구조를 기반으로 하는 계층형 데이터 모델을 사용
- 계층형 데이터 모델에서 데이터는 트리 형태로, 각 데이터 요소(개체)들은 상하관계를 나타내는 링크로 구성
- 개체들이 링크로 연결되어 있어 개체 간의 관계 변경, 추가, 삭제되면 기존 구조 변경 어려움



데이터베이스 시스템의 발전 (4/10)

□ 네트워크형 DBMS

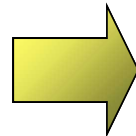
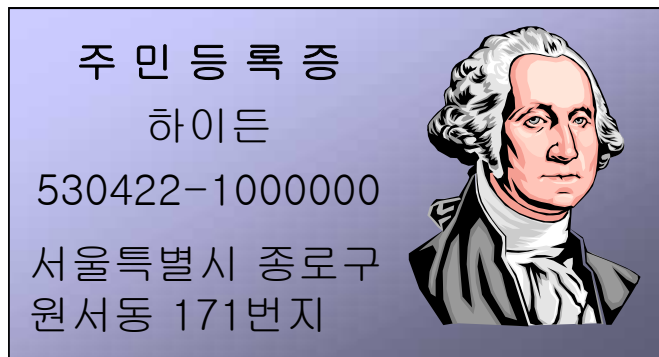
- 그래프 구조를 기반, 네트워크형 데이터 모델을 사용
- 개체와 개체 관계를 그래프 구조로 연결하는 데이터 모델
- 계층형 데이터베이스에 비해 구조가 복잡



데이터베이스 시스템의 발전 (5/10)

□ 관계형 데이터베이스 시스템

- 관계 모델 (relational model)
 - 데이터를 테이블 형태로 바라봄
- 1970년 Ted Codd에 의해 제안
- 고수준 질의 언어 지원



필드(속성)			레코드 (튜플)
성명	주민등록번호	주소	
하이든	530422-1000000	서울특별시 종로구 원서동 171번지	
...	

데이터베이스 시스템의 발전 (6/10)

□ 관계형 데이터베이스 시스템

- 테이블(Table)

- 테이블은 릴레이션(Relation) 혹은 엔티티(Entity, 개체)라고 하며, 열(Column)과 행(Row)으로 구성

- 필드(Field)

- 필드는 속성(Attribute)이라고도 하며, 테이블의 열을 의미

- 레코드(Record)

- 레코드는 튜플(Tuple) 이라고도 하며, 테이블의 행을 의미

- 도메인(Domain)

- 도메인은 필드가 가질 수 있는 값들의 집합을 의미
- 필드를 생성할 때는 데이터 형식(Data Type)을 정의, 도메인은 이러한 데이터 형식에 맞는 값들의 집합

데이터베이스 시스템의 발전 (7/10)

□ 2020년 7월 DBMS 순위

Rank			DBMS	Database Model	Score		
Jul 2020	Jun 2020	Jul 2019			Jul 2020	Jun 2020	Jul 2019
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1340.26	-3.33	+19.00
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1268.51	-9.38	+38.99
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	1059.72	-7.59	-31.11
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	527.00	+4.02	+43.73
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	443.48	+6.40	+33.55
6.	6.	6.	IBM Db2 +	Relational, Multi-model ⓘ	163.17	+1.36	-10.97
7.	7.	7.	Elasticsearch +	Search engine, Multi-model ⓘ	151.59	+1.90	+2.77
8.	8.	8.	Redis +	Key-value, Multi-model ⓘ	150.05	+4.40	+5.78
9.	9.	↑ 11.	SQLite +	Relational	127.45	+2.64	+2.82
10.	10.	10.	Cassandra +	Wide column	121.09	+2.08	-5.91
11.	11.	↓ 9.	Microsoft Access	Relational	116.54	-0.64	-20.77
12.	12.	↑ 13.	MariaDB +	Relational, Multi-model ⓘ	91.13	+1.34	+6.69
13.	13.	↓ 12.	Splunk	Search engine	88.27	+0.19	+2.78
14.	14.	14.	Hive	Relational	76.42	-2.23	-4.45
15.	15.	15.	Teradata +	Relational, Multi-model ⓘ	75.97	+2.69	-1.85
16.	16.	↑ 20.	Amazon DynamoDB +	Multi-model ⓘ	64.58	-0.29	+8.17
17.	17.	↑ 19.	SAP Adaptive Server	Relational	53.87	+0.78	-2.78

<총 358개 중 17개 표시. 출처: <https://db-engines.com/en/ranking>>

데이터베이스 시스템의 발전 (8/10)

□ SQL (Structured Query Language)

- 관계 모델에서 사용되는 질의 언어
 - (예) 계좌번호 67890의 잔고를 알아보자

```
SELECT balance
FROM Accounts
WHERE accountNo = 67890;
```

- (예) 잔고가 0 이하인 저축성 예금 계좌들을 찾아라.

```
SELECT accountNo
FROM Accounts
WHERE type = 'savings' and balance < 0;
```

애트리뷰트 →

<i>accountNo</i>	<i>balance</i>	<i>type</i>
12345	1000.00	savings
67890	2846.92	checking
...

튜플 →

Accounts 테이블 (또는 릴레이션)

데이터베이스 시스템의 발전 (9/10)

□ 소형화 되어가는 시스템

■ 과거

- DBMS는 거대하고 값비싼 소프트웨어 시스템
- 대형 컴퓨터에서 운영

■ 오늘날

- DBMS는 소형 기계에서도 사용 가능
 - 수십 기가바이트의 데이터가 **PC**용 디스크에 저장 가능
 - 데이터베이스 시스템을 **PC**에서도 구동 가능

데이터베이스 시스템의 발전 (10/10)

□ 대형화 되어가는 시스템

- 오늘날 기가바이트 단위도 대용량이 아님
 - terabyte(10^{12} byte) 혹은 그 이상의 정보가 요구되어지기도 함
- 멀티미디어 데이터
 - 이미지, 오디오, 비디오 같은 대용량의 데이터 타입
 - petabyte(10^{15} byte) 데이터를 필요로 하는 응용도 있음
- 대용량 데이터를 위한 경향
 - 3차 저장 매체
 - 디스크보다 큰 접근 시간, 그러나 더 많은 저장 공간
 - 병렬 연산
 - 병렬 디스크 접근
 - 병렬 컴퓨터나 분산된 컴퓨터들에서 병렬 질의 수행

데이터베이스 시스템 스터디의 개요 (1/2)

□ 데이터베이스 설계

- 어떻게 하면 유용한 데이터베이스를 만들 수 있는가?
- 어떤 종류의 정보들이 데이터베이스에 저장되어야 하는가?

□ 데이터베이스 프로그래밍

- 질의와 기타 다른 연산들을 어떻게 표현하는가?
- **DBMS**의 다양한 기능들을 어떻게 사용하는가?
- 프로그래밍 언어와 어떻게 같이 사용되는가?

□ 데이터베이스 시스템 구현

- **DBMS**를 어떻게 만들 것인가?
 - 질의 처리, 트랜잭션 처리, 스토리지 시스템 등

데이터베이스 시스템 스테디의 개요 (2/2)

HR / 경력관리 / 리더십|조직관리 / 이직/채용

© 2018.11.29

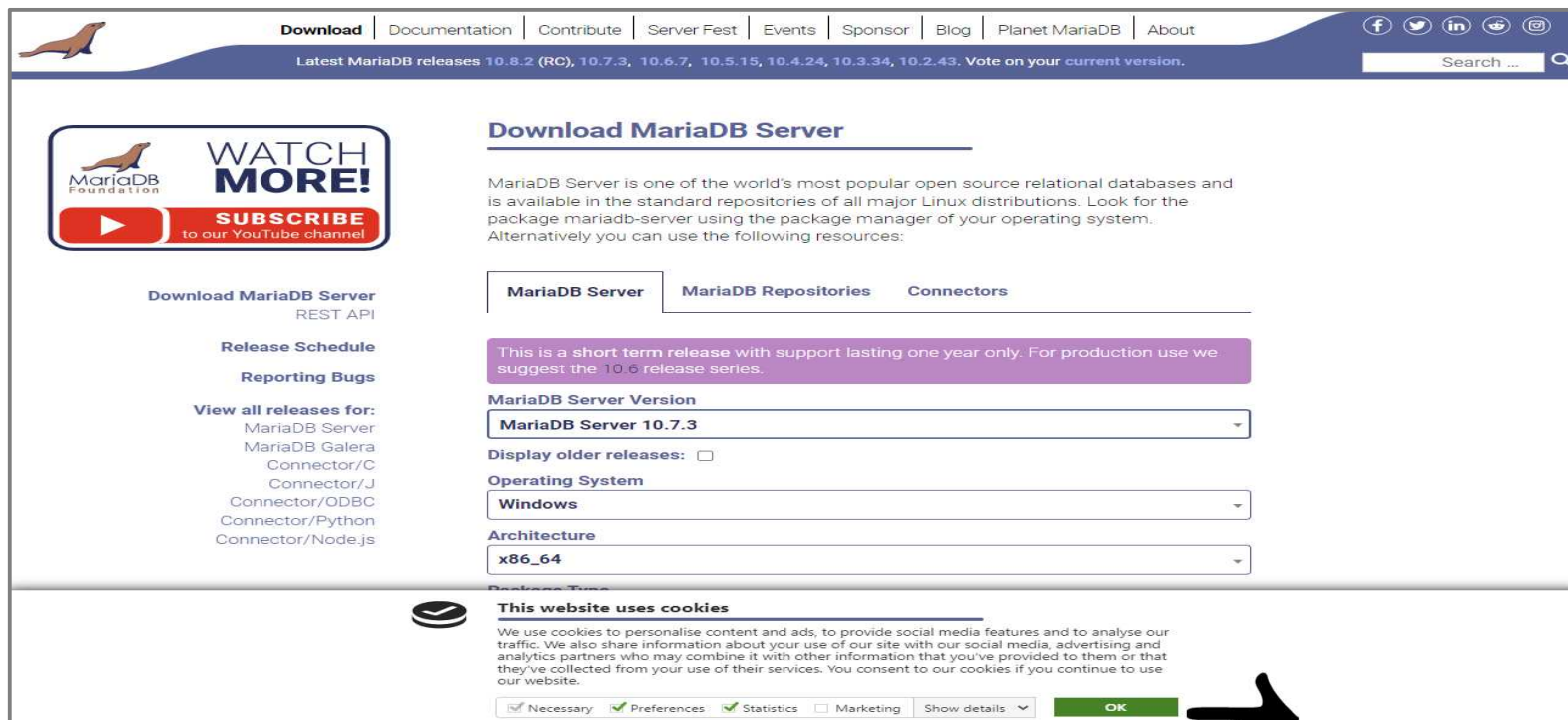
초급 전문가도 연봉 1억 이상... 2019년 수요 높은 IT 역량 13선

2019년 수요 높은 직종 13선

직종	25th percentile	50th percentile	75th percentile	95th percentile
BI 애널리스트	\$85,750	\$106,000	\$132,000	\$178,000
클라우드 아키텍트	\$75,000	\$94,500	\$118,000	\$159,500
클라우드 시스템 엔지니어	\$86,250	\$103,000	\$123,250	\$145,750
데이터 과학자	\$102,750	\$121,500	\$147,500	\$175,000
데이터베이스 엔지니어	\$98,250	\$118,000	\$141,000	\$167,750
개발자(웹, SW, 모바일)	\$83,500 (웹) \$98,250 (SW) \$65,600 (모바일)	\$100,250 (웹) \$117,500 (SW) \$79,000 (모바일)	\$119,750 (웹) \$140,750 (SW) \$93,500 (모바일)	\$142,000 (웹) \$166,500 (SW) \$105,000 (모바일)
데브옵스 엔지니어	\$90,250	\$110,500	\$134,750	\$178,250
풀스택 개발자	\$65,000	\$79,250	\$96,000	\$130,500
헬프데스크, 데스크톱 지원 전문가	\$49,000 (tier 1) \$38,250 (tier 2) \$32,250 (tier 3)	\$58,500 (tier 1) \$45,740 (tier 2) \$54,750 (tier 3)	\$70,000 (tier 1) \$54,750 (tier 2) \$46,000 (tier 3)	\$83,750 (tier 1) \$64,500 (tier 2) \$55,000 (tier 3)
IoT 전문가	\$59,500	\$71,500	\$85,250	\$100,750
네트워크 관리자	\$74,750	\$89,000	\$106,750	\$126,750
보안 전문가 (정보, 데이터, 네트워크, 시스템)	\$116,000 (정보) \$105,000 (데이터) \$93,000 (네트워크) \$93,750 (시스템)	\$139,000 (정보) \$125,250 (데이터) \$111,500 (네트워크) \$112,250 (시스템)	\$167,250 (정보) \$149,500 (데이터) \$134,000 (네트워크) \$134,750 (시스템)	\$199,750 (정보) \$178,250 (데이터) \$158,750 (네트워크) \$159,750 (시스템)
시스템 관리자	\$68,000	\$81,750	\$97,750	\$115,750

MariaDB 설치 및 테스트 (1/20)

- ❑ MariaDB 설치 (1/15)
 - MariaDB 다운로드
 - <https://downloads.mariadb.org>

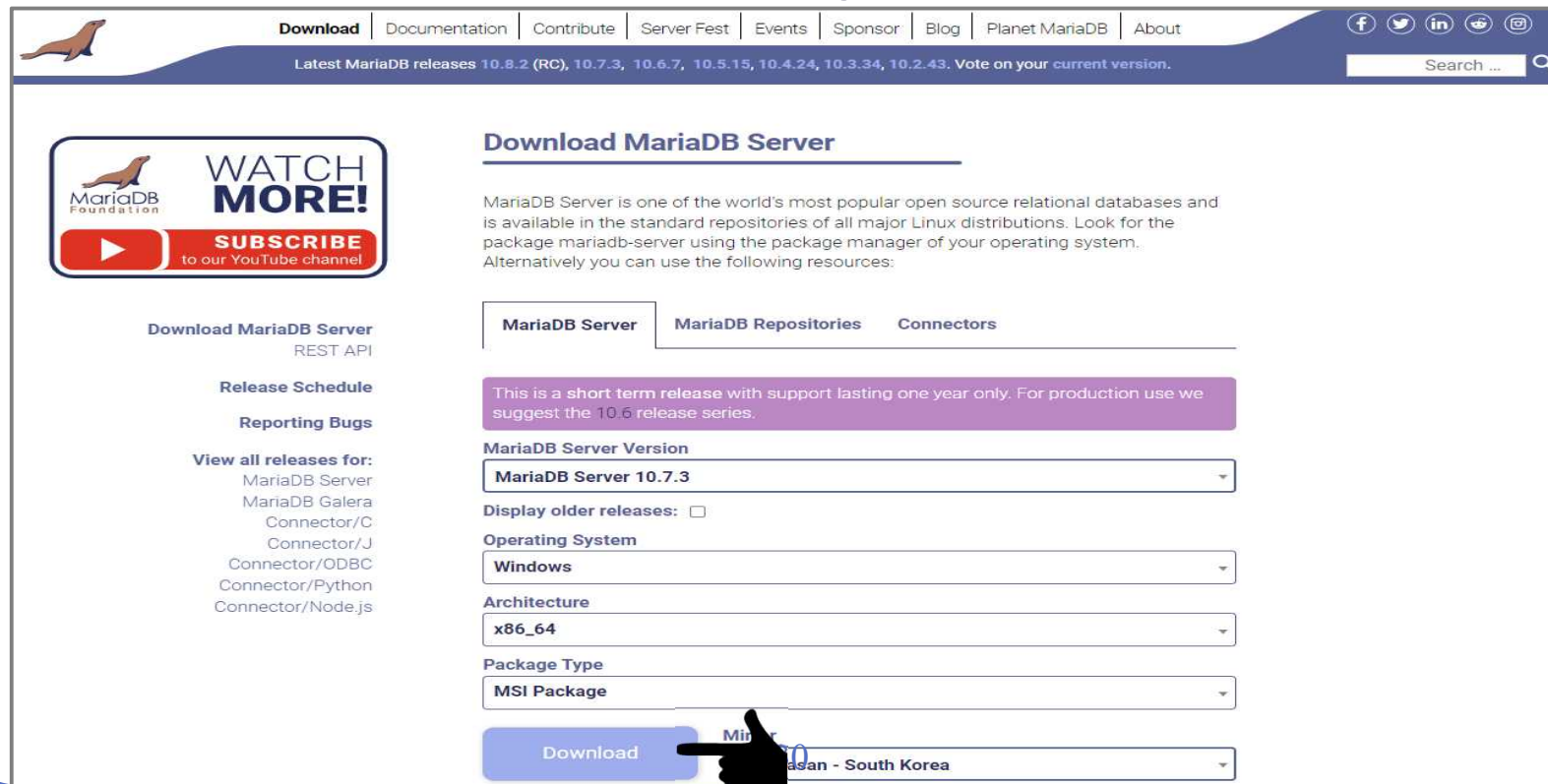


MariaDB 설치 및 테스트 (2/20)

□ MariaDB 설치 (2/15)

■ MariaDB 다운로드

- 운영체제에 맞는 msi Package 선택 후 [Download] 버튼 클릭



MariaDB 설치 및 테스트 (3/20)

□ MariaDB 설치 (3/15)

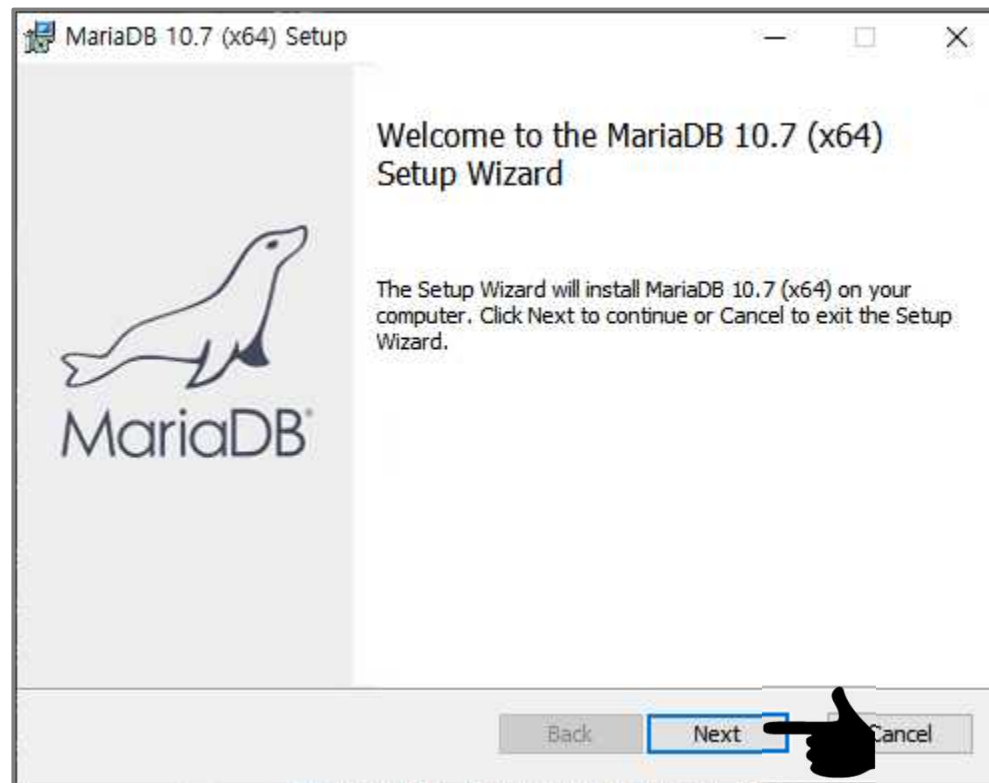
- MariaDB 다운로드

- 다운로드 된 [mariadb-10.7.3-winx64.msi] 파일 실행



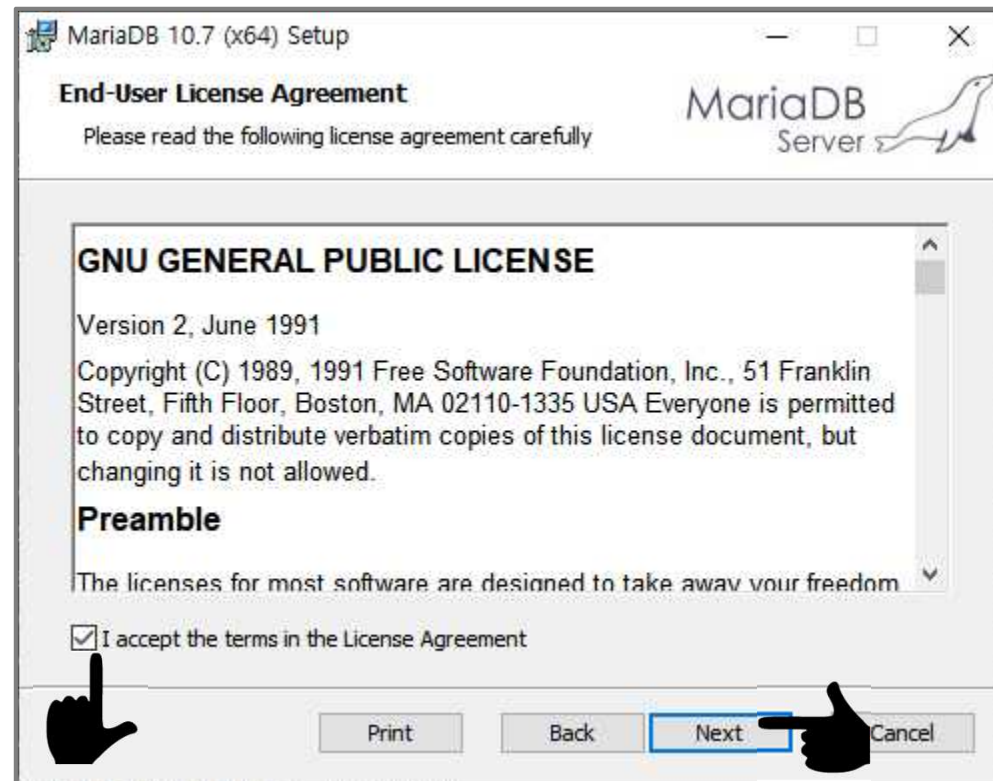
MariaDB 설치 및 테스트 (4/20)

- MariaDB 설치 (4/15)
 - msi 파일 실행



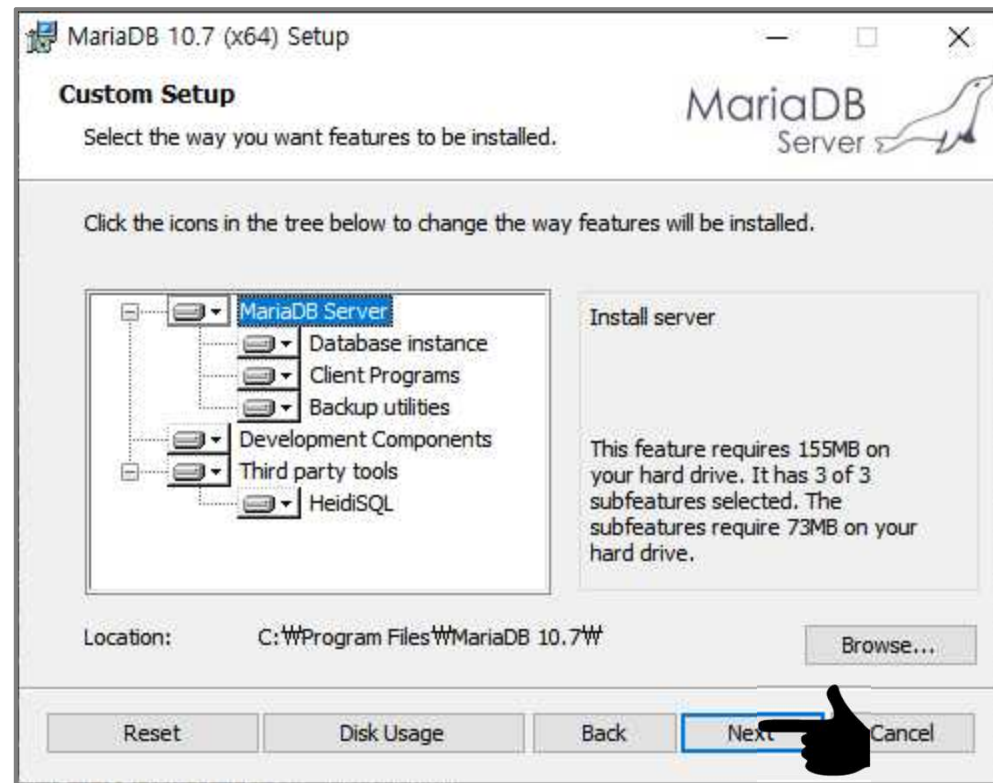
MariaDB 설치 및 테스트 (5/20)

- MariaDB 설치 (5/15)
 - msi 파일 실행



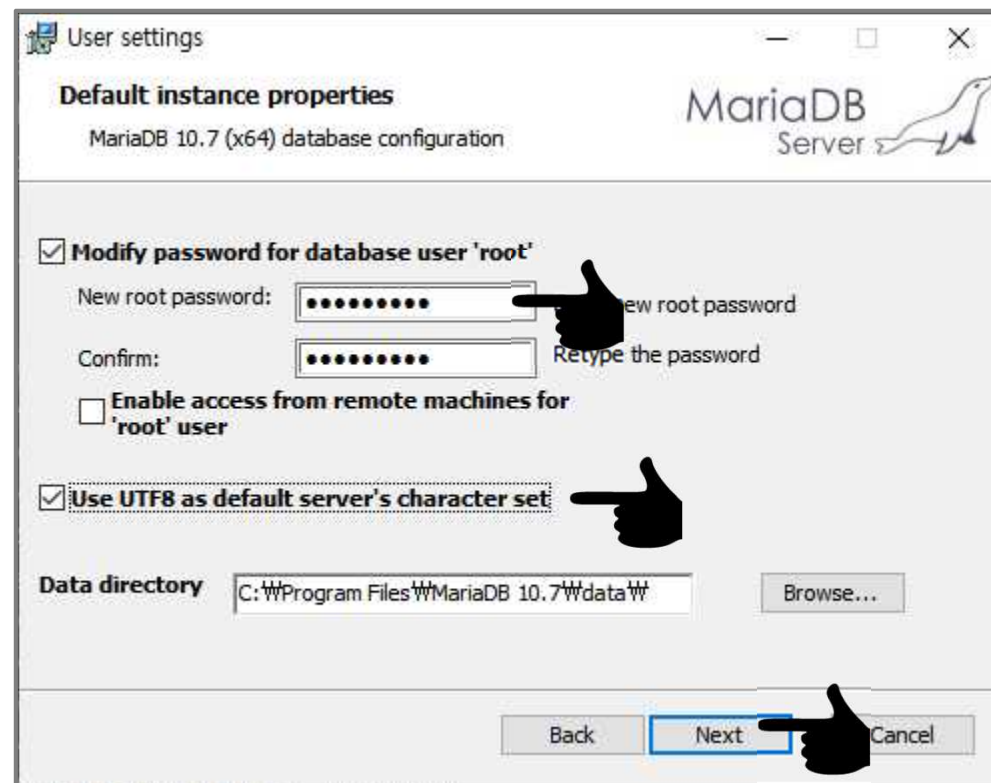
MariaDB 설치 및 테스트 (6/20)

- MariaDB 설치 (6/15)
 - msi 파일 실행



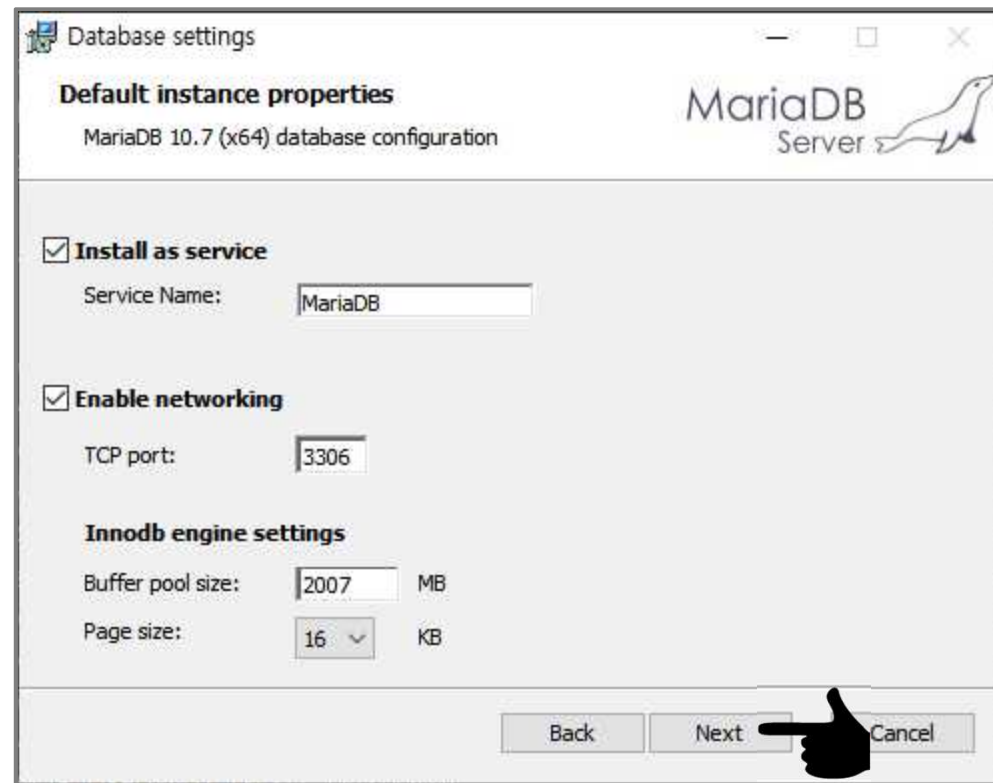
MariaDB 설치 및 테스트 (7/20)

- MariaDB 설치 (7/15)
 - msi 파일 실행



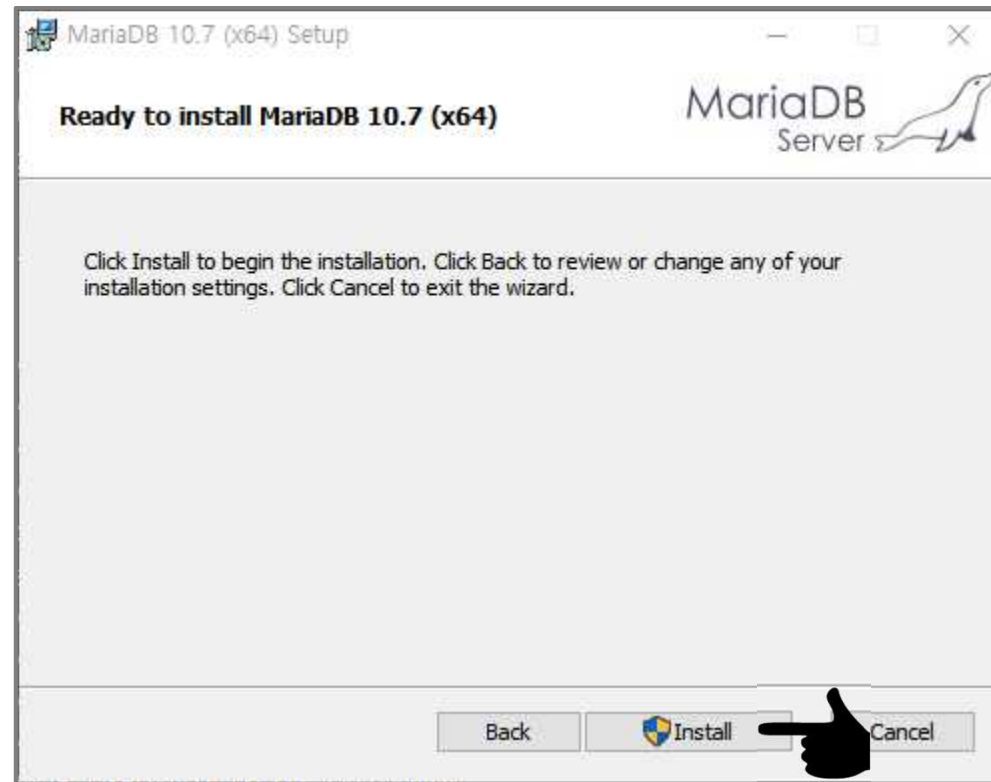
MariaDB 설치 및 테스트 (8/20)

- MariaDB 설치 (8/15)
 - msi 파일 실행



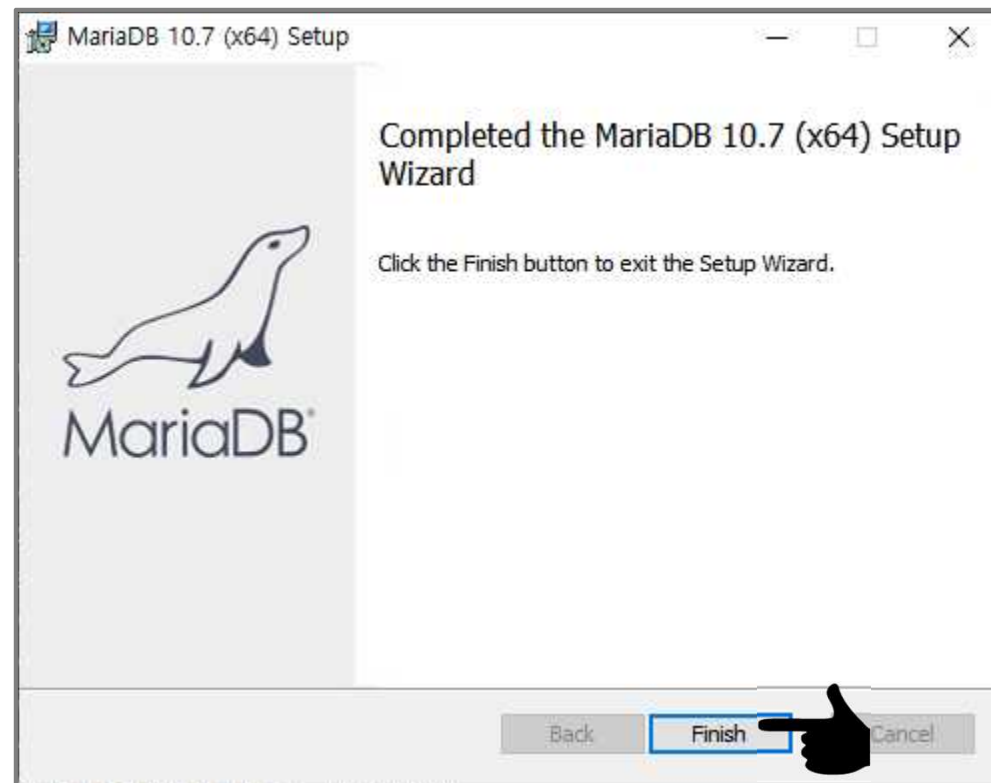
MariaDB 설치 및 테스트 (9/20)

- MariaDB 설치 (9/15)
 - msi 파일 실행



MariaDB 설치 및 테스트 (10/20)

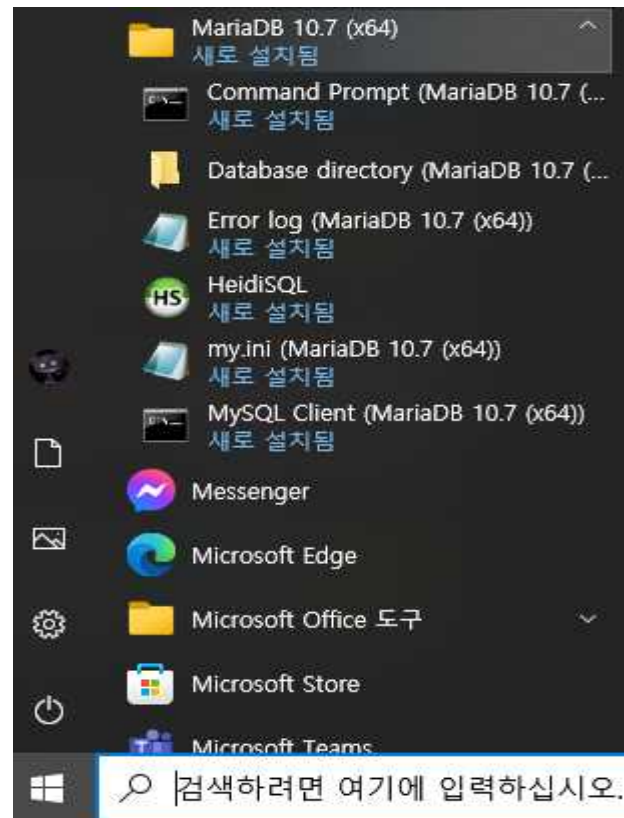
- MariaDB 설치 (10/15)
 - msi 파일 실행



MariaDB 설치 및 테스트 (11/20)

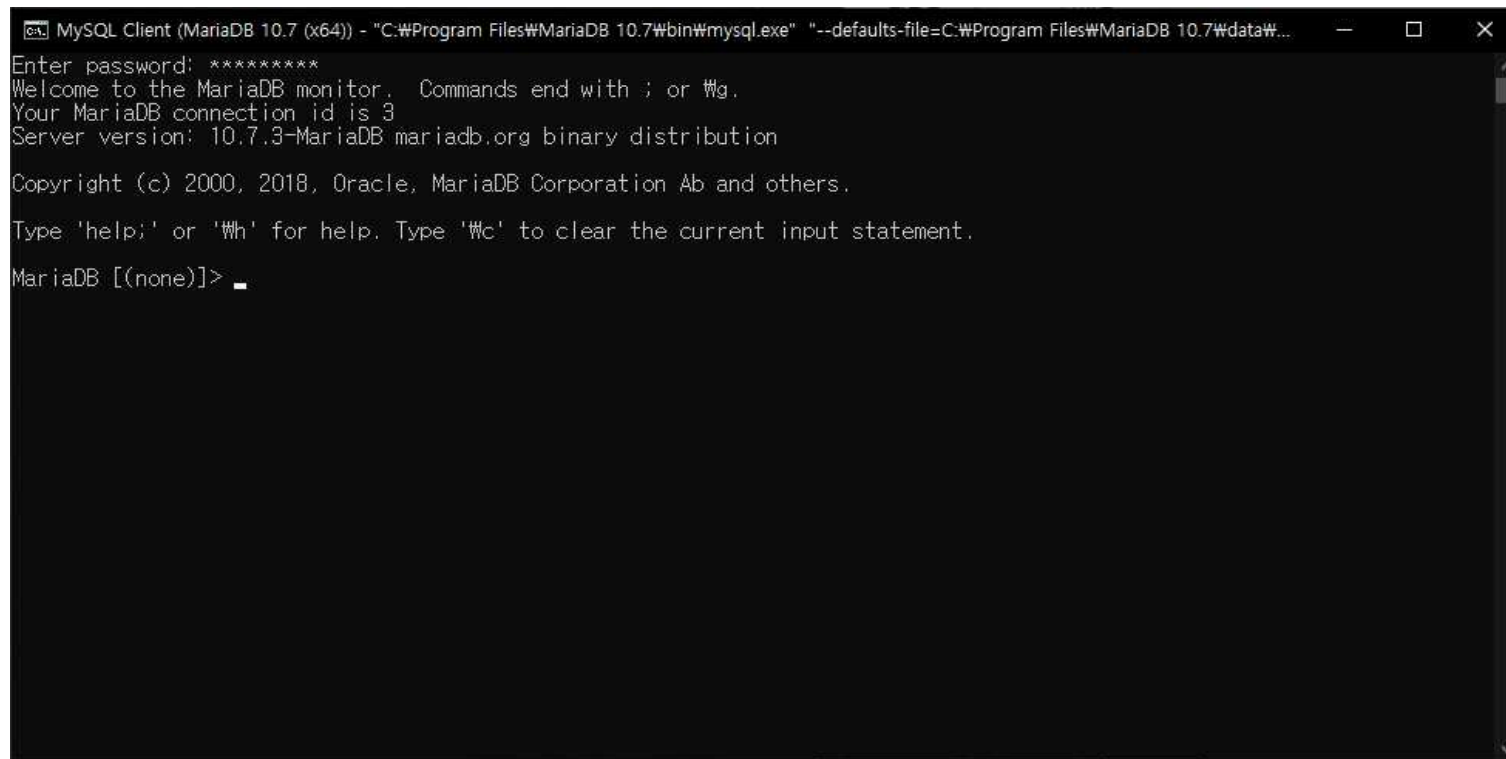
□ MariaDB 설치 (11/15)

- 시작메뉴 - MySQL Client(MariaDB ...) 선택



MariaDB 설치 및 테스트 (12/20)

- MariaDB 설치 (12/15)
 - Password 입력



```
MySQL Client (MariaDB 10.7 (x64)) - "C:\Program Files\MariaDB 10.7\bin\mysql.exe" --defaults-file=C:\Program Files\MariaDB 10.7\data\...
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.7.3-MariaDB mariadb.org binary distribution

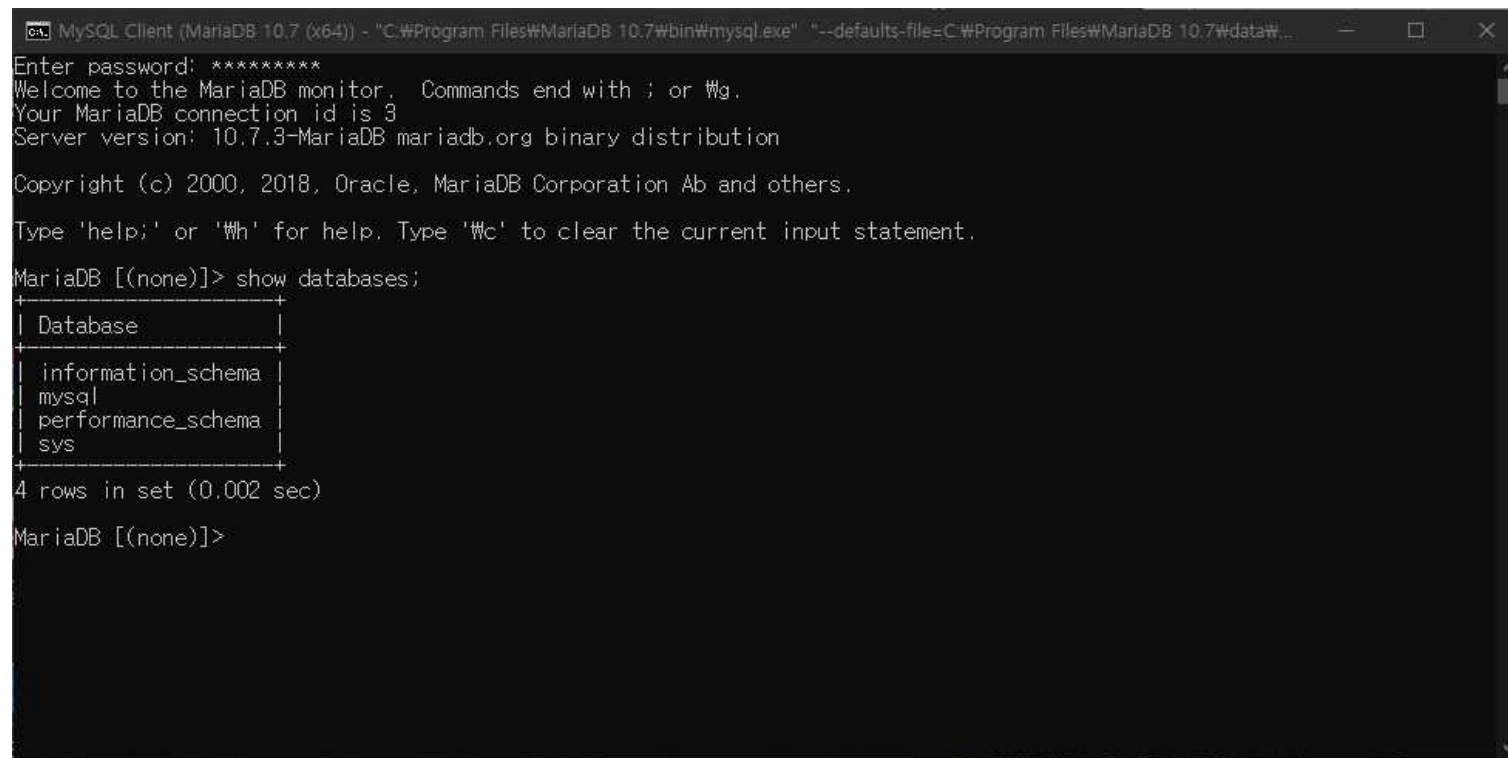
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```


MariaDB 설치 및 테스트 (13/20)

- MariaDB 설치 (13/15)
 - 입력: show databases;



```
MySQL Client (MariaDB 10.7 (x64)) - "C:\Program Files\MariaDB 10.7\bin\mysql.exe" --defaults-file=C:\Program Files\MariaDB 10.7\data\...
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 3
Server version: 10.7.3-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

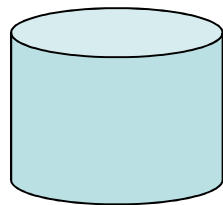
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.002 sec)

MariaDB [(none)]>
```

MariaDB 설치 및 테스트 (14/20)

□ MariaDB 테스트 (1/7)

- 데이터베이스 및 테이블 생성, 데이터 삽입, 질의 테스트



ust_db

데이터베이스 생성

birthday (생년월일)	name (이름)	dept (학과)	hood (거주지역)	eng (영어)	math (수학)
890405	홍길동	빅데이터과학	대전	90	95
891025	이순신	빅데이터과학	서울	80	92
900710	이효리	과학기술경영	광주	70	60
911010	차태현	생명의료융합	부산	80	50

students 테이블 생성



질의 수행

- 1) 서울 거주하는 학생은 누구?
- 2) 빅데이터과학 학생 수는?
- 3) 영어 70점 이상, 수학 90점 이상 학생은?

MariaDB 설치 및 테스트 (15/20)

□ MariaDB 테스트 (2/7)

▪ ust_db 데이터베이스 생성

```
CREATE DATABASE ust_db DEFAULT CHARACTER SET UTF8;  
SHOW DATABASES;
```

```
MariaDB [(none)]> CREATE DATABASE ust_db DEFAULT CHARACTER SET UTF8;  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [(none)]> SHOW DATABASES;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| kcs |  
| kcs_clustering |  
| korea_customs |  
| mysql |  
| performance_schema |  
| test |  
| ust_db |  
+-----+  
8 rows in set (0.00 sec)
```

▪ ust_db 데이터베이스 선정

```
USE ust_db;
```

```
MariaDB [(none)]> USE ust_db;  
Database changed
```

MariaDB 설치 및 테스트 (16/20)

□ MariaDB 테스트 (3/7)

■ students 테이블 스키마 만들기

```
CREATE TABLE students
(
  birthday VARCHAR(6),
  name VARCHAR(10) NOT NULL,
  dept VARCHAR(20) NOT NULL,
  hood VARCHAR(10) DEFAULT '대전',
  eng INT,
  math INT,
  PRIMARY KEY(birthday)
) ENGINE=INNODB;
```

■ 만든 스키마 확인

```
DESC students;
```

```
MariaDB [ust_db]> DESC students;
```

Field	Type	Null	Key	Default	Extra
birthday	varchar(6)	NO	PRI	NULL	
name	varchar(10)	NO		NULL	
dept	varchar(20)	NO		NULL	
hood	varchar(10)	YES		대전	
eng	int(11)	YES		NULL	
math	int(11)	YES		NULL	

```
6 rows in set (0.01 sec)
```

MariaDB 설치 및 테스트 (17/20)

□ MariaDB 테스트 (4/7)

▪ students 테이블에 데이터 삽입

```
INSERT INTO students VALUES('890405', '홍길동', '빅데이터과학', '대전', 90, 95);  
INSERT INTO students VALUES('891025', '이순신', '빅데이터과학', '서울', 80, 92);  
INSERT INTO students VALUES('900710', '이효리', '과학기술경영', '광주', 70, 62);  
INSERT INTO students VALUES('911010', '차태현', '생명의료융합', '부산', 80, 50);
```

```
MariaDB [ust_db]> INSERT INTO students VALUES('890405', '홍길동', '빅데이터과학', '대전', 90, 95);  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [ust_db]> INSERT INTO students VALUES('891025', '이순신', '빅데이터과학', '서울', 80, 92);  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [ust_db]> INSERT INTO students VALUES('900710', '이효리', '과학기술경영', '광주', 70, 62);  
Query OK, 1 row affected (0.00 sec)  
  
MariaDB [ust_db]> INSERT INTO students VALUES('911010', '차태현', '생명의료융합', '부산', 80, 50);  
Query OK, 1 row affected (0.00 sec)
```

MariaDB 설치 및 테스트 (18/20)

□ MariaDB 테스트 (5/7)

- students 테이블에 삽입된 전체 데이터 보기

```
SELECT * FROM students;
```

```
MariaDB [ust_db]> SELECT * FROM students;
```

birthday	name	dept	hood	eng	math
890405	홍길동	빅데이터과학	대전	90	95
891025	이순신	빅데이터과학	서울	80	92
900710	이효리	과학기술경영	광주	70	62
911010	차태현	생명의료융합	부산	80	50

```
4 rows in set (0.00 sec)
```

MariaDB 설치 및 테스트 (19/20)

□ MariaDB 테스트 (6/7)

- 질의 1) 서울 거주하는 학생은 누구?

```
SELECT name FROM students WHERE hood='서울';
```

```
MariaDB [ust_db]> SELECT name FROM students WHERE hood='서울';
+-----+
| name |
+-----+
| 이순신 |
+-----+
1 row in set (0.00 sec)
```

- 질의 2) 빅데이터과학 학생 수는?

```
SELECT COUNT(*) FROM students WHERE dept='빅데이터과학';
```

```
MariaDB [ust_db]> SELECT COUNT(*) FROM students WHERE dept='빅데이터과학';
+-----+
| COUNT(*) |
+-----+
| 2 |
+-----+
1 row in set (0.00 sec)
```

MariaDB 설치 및 테스트 (20/20)

□ MariaDB 테스트 (7/7)

- 질의 3) 영어 70점 이상, 수학 90점 이상 학생은?

```
SELECT name FROM students WHERE eng>=70 AND math>=90;
```

```
MariaDB [ust_db]> SELECT name FROM students WHERE eng>=70 AND math>=90;
+-----+
| name |
+-----+
| 홍길동 |
| 이순신 |
+-----+
2 rows in set (0.00 sec)
```


질의응답

