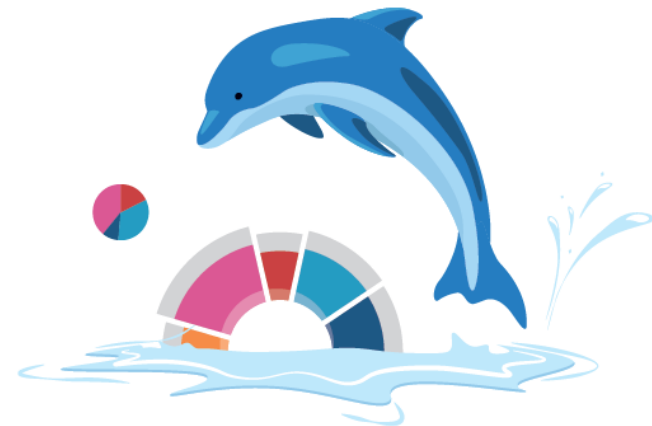


# Chapter 01

## SQL 비긴즈



# Section 01

## 데이터베이스의 개요

# 1. 데이터에 관심을 가져야 하는 이유

## ■ 데이터

- 측정, 관찰, 연구, 분석 등을 통해 수집된 정보
- 활용분야
  - ✓ 금융 부문에서는 위험과 사기 감지
  - ✓ 의료 부문에서는 유전학 및 질병을 분석
  - ✓ 기업에서는 고객의 행동과 선호도 이해
  - ✓ 새로운 시장 기회 발견
  - ✓ 비즈니스 전략 수립

# 1. 데이터에 관심을 가져야 하는 이유

## ■ 폭발적으로 증가하는 데이터

- 정보 기술의 발전과 디지털화로 인해 막대한 양의 데이터가 생성됨
- 전 세계 디지털 정보량 60제타바이트(ZB) 수준(2020년 기준)

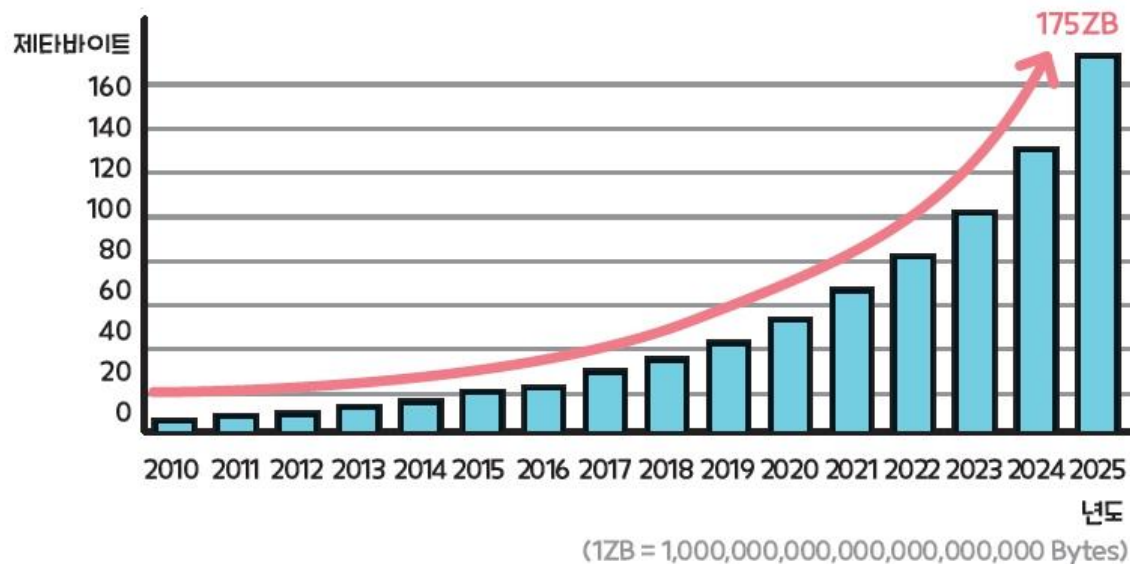


그림 1-1 데이터 예상 증가율

# 1. 데이터에 관심을 가져야 하는 이유

## ■ 4차 산업혁명 기술과 데이터

- 데이터가 핵심 자원으로 적용되는 분야
  - ✓ 인공지능 및 머신러닝
  - ✓ 빅데이터 분석
  - ✓ 사물 인터넷
  - ✓ 블록체인
  - ✓ 로봇공학
  - ✓ 가상 현실 및 증강 현실

## 2. 데이터의 형태

### ■ 정형 데이터(Structured Data)

- 미리 정해진 구조에 따라 저장된 데이터
- 손쉽게 데이터 검색 및 삽입, 수정, 삭제 연산을 수행할 수 있음
- 엑셀의 스프레드시트나 관계형 데이터베이스의 테이블에 저장된 데이터 등

	A	B	C	D
1	사용일자	노선명	역명	승차충승객수
2	20220101	3호선	수서	7370
3	20220101	3호선	학여울	461
4	20220101	3호선	대청	3224
5	20220101	3호선	일원	3321
6	20220101	경원선	창동	1
7	20220101	1호선	신설동	4939
8	20220101	1호선	동대문	6486
9	20220101	1호선	종로5가	7975
10	20220101	1호선	서울역	18398
11	20220101	1호선	동묘앞	7240
12	20220101	1호선	시청	5604
13	20220101	1호선	종각	9977
14	20220101	1호선	종로3가	11017
15	20220101	2호선	구로디지털단지	18363
16	20220101	2호선	신대방	11652
17	20220101	2호선	신림	29564
18	20220101	2호선	봉천	9207
19	20220101	2호선	서울대입구(관악·	20142

그림 1-3 정형 데이터의 예(엑셀 파일)

## 2. 데이터의 형태

### ■ 반정형 데이터(Semi-Structured Data)

- 정형 데이터와는 달리 구조화되어 있지 않아서 연산이 불가능함
- 데이터 내부의 데이터 구조에 대한 메타 정보를 함께 제공하는 파일 형식의 데이터
- HTML, XML, JSON 파일이나 웹 로그, 센서 데이터 등

```
{
  "crewId": "110135",
  "qualificationCode": "737TCE",
  "renewalType": "RECURRENT",
  "fromDateTime": "2099-04-30 00:00:00",
  "toDateTime": "2099-05-30 00:00:00",
  "qualificationStatus": "PLANNED",
  "checkDateTime": "2022-04-07 00:00:00",
  "comments": null,
  "internalCrewID": "108207",
  "odsChangeTimestamp": "2023-08-14 13:37:57.668287"
}
```

그림 1-4 반정형 데이터의 예(JSON 파일)

## 2. 데이터의 형태

### ■ 비정형 데이터(Unstructured Data)

- 정해진 구조나 규칙이 없고, 연산에 사용할 수 없는 형태의 데이터
- 소셜 데이터 및 워드나 PDF 문서, 이미지, 영상 등



(a) 페이스북



(b) 인스타그램



(c) 트위터

그림 1-5 비정형 데이터의 예(SNS)



### 3. 데이터베이스와 데이터베이스 관리 시스템

#### ■ 데이터베이스(Database)

- 여러 사람이 공유하고 운영할 목적으로 관리되는 통합적인 정보의 집합
- 컴퓨터 시스템에 전자적으로 저장되는 구조화된 데이터 모음
- 데이터베이스는 데이터베이스 관리 시스템(DBMS)에 의해 관리됨

#### ■ DBMS

- 최종 사용자 또는 응용 프로그램이 데이터베이스에 접근할 수 있는 인터페이스 역할
- 저장된 데이터를 보다 체계적으로 관리하고 이용할 수 있게 해주는 소프트웨어

### 3. 데이터베이스와 데이터베이스 관리 시스템

#### ■ 사용 방식에 따른 데이터베이스 유형

표 1-1 데이터베이스 유형

데이터베이스 유형	설명
관계형 데이터베이스	열과 행이 있는 테이블 집합으로 구성되며, 정형 데이터에 액세스하는 가장 효율적이고 유연한 방법을 제공한다.
객체 지향 데이터베이스	객체 지향 패러다임을 사용하는 객체 지향 프로그래밍으로부터 영향을 받아 생성된 데이터베이스로 정보를 객체의 형태로 표현한다.
분산 데이터베이스	물리적으로 떨어진 데이터베이스를 네트워크로 연결하여 단일 데이터베이스 이미지를 보여준다. 분산된 작업 처리를 수행하는 데이터베이스로 사용자는 시스템이 분산되어 있는지 인식하지 못한 채 사용할 수 있다.
데이터 웨어하우스	조직 내 서로 다른 다양한 소스들의 정보를 집계하고 저장하는 시스템으로 최종 사용자가 전사적 정보를 분석할 수 있게 하여 의사 결정을 도모할 수 있도록 설계된다.
NoSQL 데이터베이스	관계형 데이터베이스 이외의 형식으로 데이터를 저장하는 데이터베이스로 대량의 분산된 데이터를 저장하고 조회하는 데 특화되어 있다.
클라우드 데이터베이스	클라우드 플랫폼을 통해 구축·접근할 수 있는 데이터베이스로 기존 데이터베이스와 동일한 기능에 클라우드 컴퓨팅의 유연성이 추가되었다. 사용자가 클라우드 인프라에서 DBMS를 선택하여 데이터베이스를 구현할 수 있다.
자율 운영 데이터베이스	클라우드를 기반으로 머신러닝을 사용하여 데이터베이스 튜닝, 보안, 백업, 업데이트 및 기타 데이터베이스 관리자가 전통적으로 수행해 온 일상적인 관리 작업을 자동화한 데이터베이스이다.

## 4. 관계형 데이터베이스

### ■ 관계형 데이터베이스(Relational Database)

- 관계형 데이터 모델을 기반으로 한 데이터베이스

### ■ 테이블(Table)

- 릴레이션(Relation)이라고도 함
- 1개 이상의 열과 0개 이상의 행으로 이루어진 데이터 집합

### ■ 열(Column)

- 속성(Attribute), 필드(Field)라고도 함
- 의미가 더 이상 분리되지 않는 최소의 데이터 단위

## 4. 관계형 데이터베이스

### ■ 행(Row)

- 튜플(Tuple), 레코드(Record)라고도 함
- 관련 있는 열의 묶음
- 한 테이블에 저장된 모든 행은 동일한 수의 열을 가짐

### ■ 관계(Relationship)

- 두 테이블 간에 서로 연결되는 방식
- 데이터의 종속성을 표현

### ■ 기본키(Primary key)

- 주된 식별자
- 한 테이블 내에서 행을 구별할 수 있는 열 또는 열의 묶음

## 4. 관계형 데이터베이스

### ■ 관계형 데이터베이스의 기본 용어

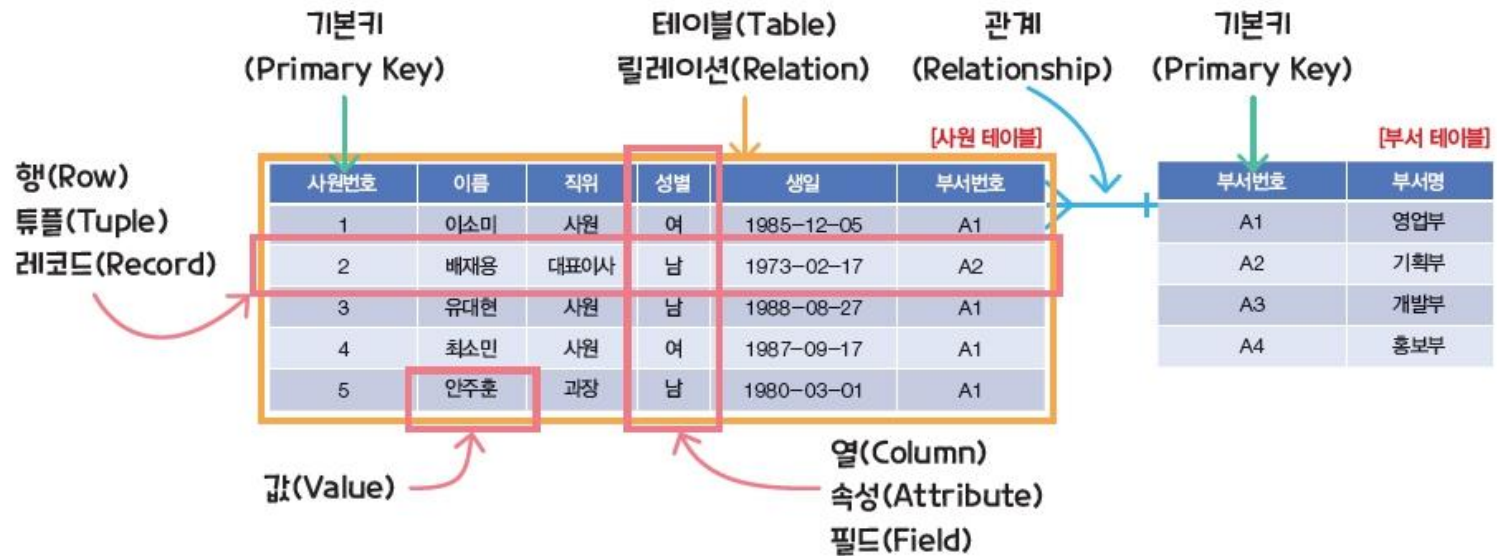


그림 1-6 관계형 데이터베이스의 기본 용어

## 5. MySQL

### ■ MySQL

- 1995년에 발표된 오픈 소스 RDBMS
- 웹 애플리케이션용으로 설계 및 최적화되어 모든 플랫폼에서 실행할 수 있음
- 4대 RDBMS : MySQL, Oracle, MS-SQL, PostgreSQL

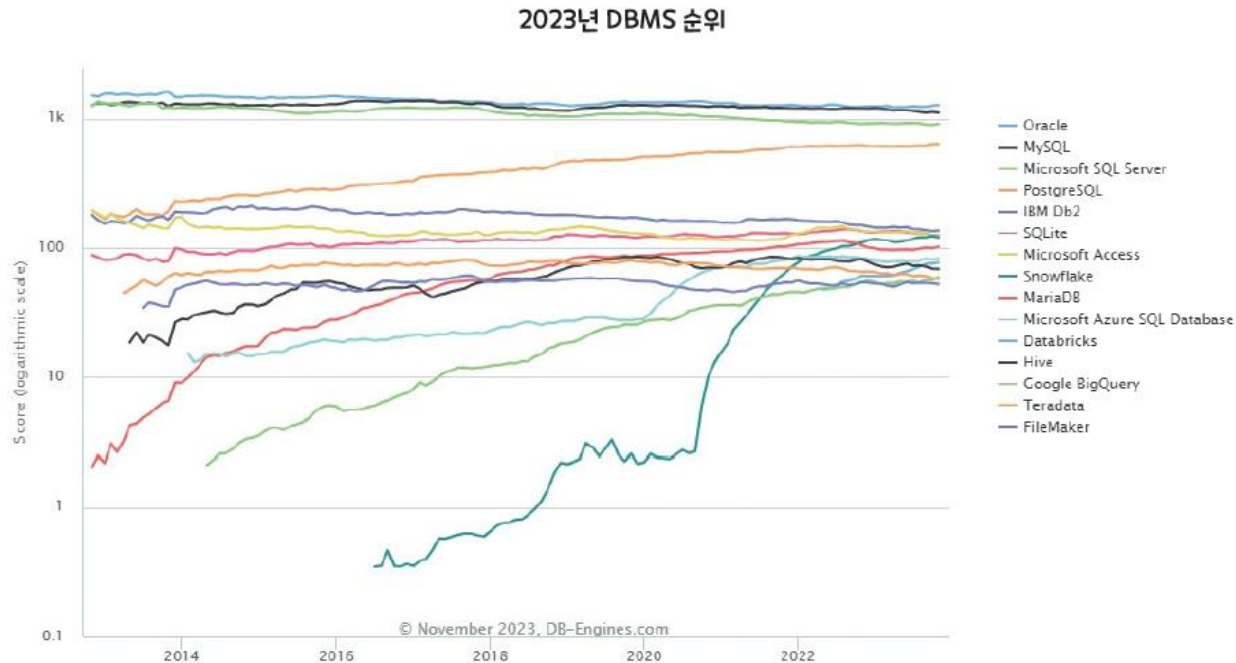


그림 1-7 2023년 기준 DBMS 순위(©[https://db-engines.com/en/ranking\\_trend/relational+dbms](https://db-engines.com/en/ranking_trend/relational+dbms))

# Section 02

## SQL의 개요

# 1. SQL을 배워야 하는 이유

## ■ SQL(Structured Query Language)

- IBM이 1970년대에 개발한 SEQUEL을 기반으로 만들어진 언어
- 데이터에 접근하기 위한 가장 보편적인 수단
- SQL이 사용되는 클라우드 기반의 머신러닝 플랫폼
  - ✓ 아마존 레드시프트 ML
  - ✓ 구글 빅쿼리 ML
  - ✓ 스노플레이크
  - ✓ 오라클 클라우드 인프라 데이터 사이언스
  - ✓ 마이크로소프트 SQL 서버 머신러닝 서비스



## 2. SQL의 개념

### ■ SQL의 역할

- 관계형 데이터베이스 관리 시스템(RDBMS)의 데이터를 관리하고 다양한 데이터 동작을 수행하는 데 사용되는 표준화된 프로그래밍 언어
- 데이터베이스 및 테이블 생성, 데이터 검색, 추가, 수정, 삭제 및 트랜잭션 처리, 보안 및 권한 제어 등 데이터베이스의 모든 측면을 관리하는 데 사용됨

## 2. SQL의 개념

### ■ SQL의 특징

- 배우기 쉽고 사용하기 용이함
- 데이터 연산에 대한 처리 과정이 절차적이지 않으며 집합 단위로 처리됨
- 사용자 편의 중심 언어
- 미국표준협회(ANSI)의 표준에 맞게 SQL을 작성한다면 Oracle, MySQL, MS-SQL, PostgreSQL, MariaDB 등 상용 RDBMS 간에 호환과 전환이 용이함

### 3. SQL문의 종류

#### ■ 데이터 정의어(DDL)

- 데이터베이스의 구조를 정의할 때 사용하는 언어
- 데이터베이스나 테이블, 인덱스와 같은 데이터베이스 객체를 생성, 변경, 삭제할 때 사용함

표 1-2 DDL문의 종류

종류	설명
CREATE	데이터베이스 객체를 생성한다(예 데이터베이스, 테이블, 뷰, 인덱스 등).
ALTER	생성된 객체의 구조를 변경한다.
DROP	생성된 객체를 삭제한다.
TRUNCATE	테이블에 있는 데이터를 모두 삭제한다.

### 3. SQL문의 종류

#### ■ 데이터 조작어(DML)

- 데이터를 관리하는 데 사용하는 언어
- 데이터 조작어는 레코드를 추가, 삭제하거나 데이터를 변경할 때 사용함
- 데이터를 검색하는 데 사용하는 SELECT문은 데이터 조작어에 포함하기도 하고 데이터 질의어(DQL)로 분리하기도 함

표 1-3 DML문의 종류

종류	설명
SELECT	테이블에서 조건에 맞는 데이터를 검색한다.
INSERT	테이블에 새로운 레코드를 삽입한다.
UPDATE	테이블에 있는 데이터 값을 수정한다.
DELETE	테이블에 있는 레코드를 삭제한다.

### 3. SQL문의 종류

#### ■ 데이터 제어어(DCL)

- 데이터에 대한 액세스를 제어하기 위한 언어
- 데이터베이스 객체나 데이터에 대한 권한을 관리하는 데 사용함

#### ■ 트랜잭션 제어어(TCL)

- INSERT, UPDATE, DELETE문에 의해 수행된 변경 사항을 관리하는 데 사용됨

표 1-4 DCL문과 TCL문의 종류

종류		설명
DCL	GRANT	특정 사용자 또는 특정 객체에 대해 생성, 수정 등 특정 작업을 할 수 있도록 권한을 부여한다.
	REVOKE	GRANT문으로 부여한 권한을 철회할 때 사용한다.
TCL	COMMIT	트랜잭션 작업 내용을 실제 데이터베이스에 영구 저장한다.
	ROLLBACK	트랜잭션 처리 과정에서 발생한 변경 사항을 취소한다.