



데이터베이스란?



01-1 데이터베이스란 무엇일까?

01-2 데이터베이스의 종류
알아보기

01-3 SQL이란 무엇일까?

01-4 SQL을 배워야 하는 이유

- **데이터베이스**(database, DB)는 여러 사람이 공유할 목적으로 통합 관리하기 위해 논리적으로 연관된 데이터를 모아 일정한 형태로 저장해 놓은 것을 의미함
- 즉, 여러 시스템 또는 사용자들이 공용(shared)할 목적으로 통합(integrated), 저장(stored)한 데이터의 집합
- 데이터베이스의 가장 큰 목적은 중복 데이터를 최소화하여 조직의 목적에 맞게 데이터를 효율적으로 관리하는 데 있음

■ 데이터베이스란?

- 데이터베이스의 정의는 ISOS로 표현하는데, 각각의 첫 글자는 다음을 의미함

정의	설명
통합된 데이터(Integrated Data)	데이터 중복을 최소화한 데이터이다.
저장된 데이터(Stored Data)	컴퓨터가 접근할 수 있는 저장 매체에 저장된 데이터이다.
운영 데이터(Operational Data)	조직의 고유한 업무를 수행하는 데 반드시 필요한 데이터이다.
공용 데이터(Shared Data)	여러 응용 시스템이 공동으로 소유하고 유지하는 데이터이다.

■ 데이터베이스란?

- 데이터베이스 시스템의 특징은 R1C3으로 표현하는데, 각각의 첫 글자는 다음을 의미함

특징	설명
실시간 접근성(Real Time Accessibility)	사용자 질의에 실시간 응답으로 처리한다.
지속적인 변화(Continuous Evolution)	삽입, 삭제, 수정 작업을 하여 항상 최신 데이터를 동적으로 유지한다.
동시 공유(Concurrent Sharing)	목적이 서로 다른 여러 사용자가 동시에 원하는 데이터를 공유한다.
내용에 의한 참조(Content Reference)	데이터베이스에 있는 데이터를 참조할 때 레코드의 주소나 위치가 아니라 사용자가 요구하는 데이터 내용을 참조한다.

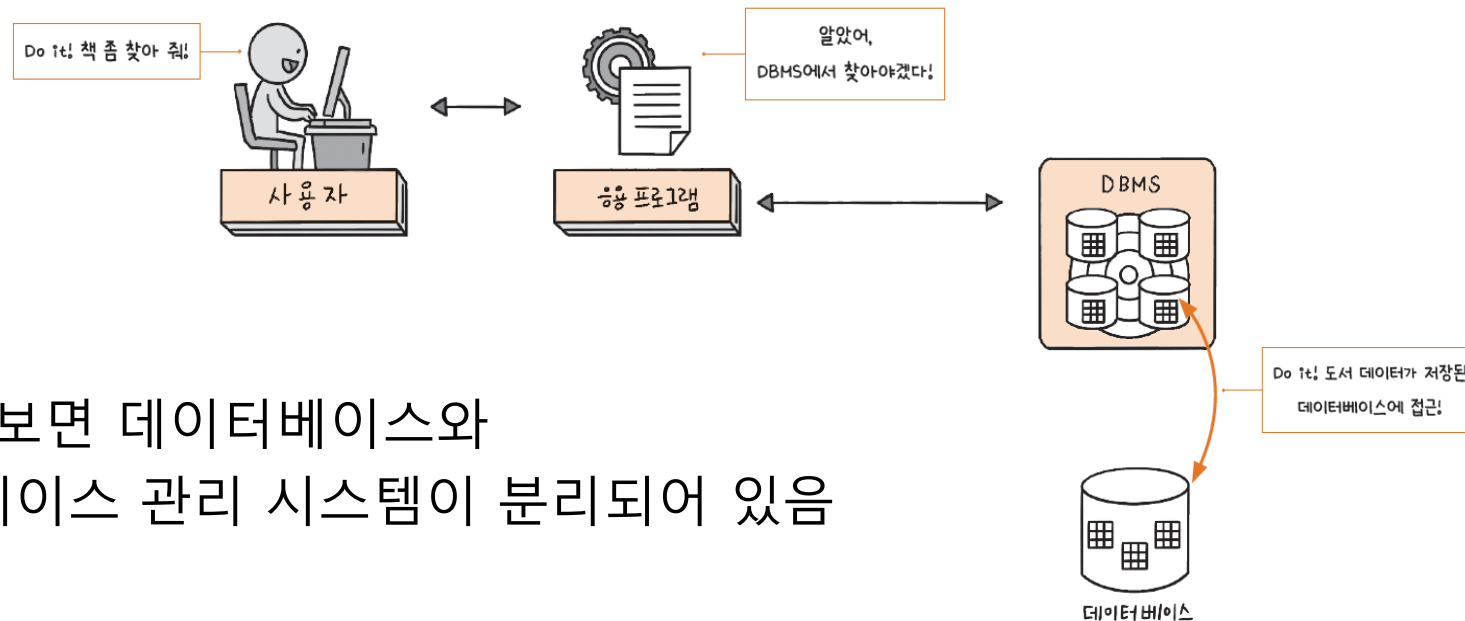
■ 데이터베이스란?

■ 데이터베이스의 장점과 단점

장점	단점
<ul style="list-style-type: none">• 데이터 중복 최소화• 중복 최소화로 데이터 저장 공간 절약• 데이터 공유 가능• 일관성, 무결성, 보안성 유지• 최신 데이터 유지• 데이터 표준화 가능• 데이터의 논리적, 물리적 독립성 확보• 쉬운 데이터 접근	<ul style="list-style-type: none">• 데이터베이스 전문 인력 필요• 시스템 운영 비용 부담• 데이터 백업 및 복구 어려움• 시스템의 복잡함• 대용량 디스크로 액세스가 집중되면서 과부하 발생

■ 데이터베이스 관리 시스템이란?

- **데이터베이스 관리 시스템**(database management system, DBMS)은 데이터 입력, 수정, 삭제 등의 기능을 제공하는 별도의 소프트웨어



- 그림을 보면 데이터베이스와 데이터베이스 관리 시스템이 분리되어 있음

■ 데이터베이스 관리 시스템이란?

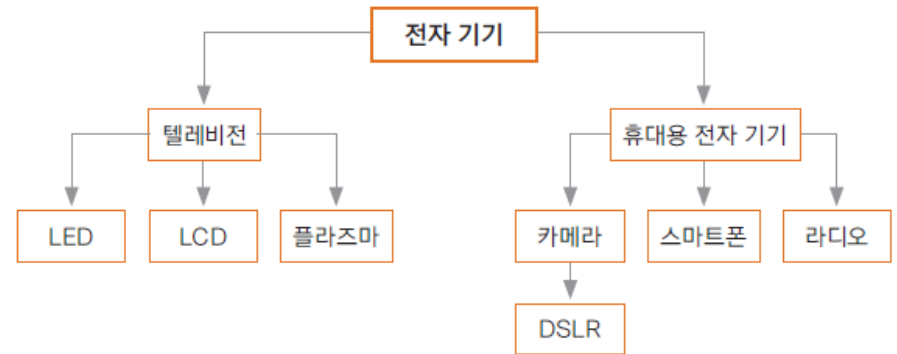
- 데이터베이스는 데이터를 담고 있는 통이고,
데이터베이스 관리 시스템은 이 통을 관리하는 소프트웨어
- 데이터베이스에 데이터가 저장되거나 삭제되는 모든 작업은
데이터베이스 관리 시스템을 통해서만 함

DB ≠ DBMS

- 실무에서는 데이터베이스를 DB, 데이터베이스 관리 시스템을 DBMS라고 함

■ 계층형 데이터베이스

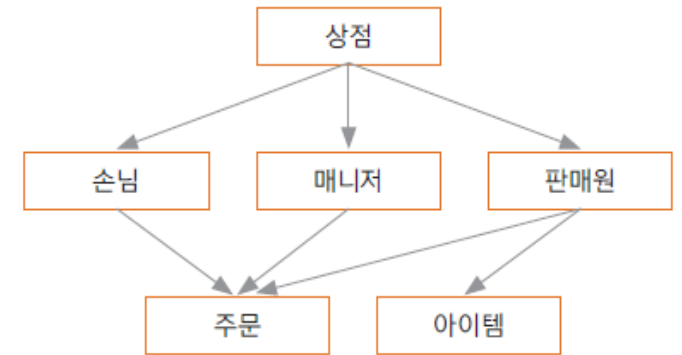
- 데이터가 부모와 자식 관계인 **트리 구조**
- 상위 레코드 아래에 하위 레코드가 여러 개 있는 구조
- 전자 기기가 가장 위에, 아래에 텔레비전, 휴대용 전자 기기가 있으며 텔레비전은 아래에 LED, LCD, 플라즈마가 위치함
- 이처럼 부모 레코드는 자식 레코드가 여러 개이지만, 자식 레코드는 부모 레코드를 하나만 가질 수 있음
- 계층형 데이터베이스는 데이터 중복이 발생하기 쉬우며, 데이터는 상하 종속 관계이므로 초기에 이 방식을 채택하면 이후 프로세스 변경을 하기 어려워서 현재는 거의 사용하지 않음



계층형 데이터베이스의 예

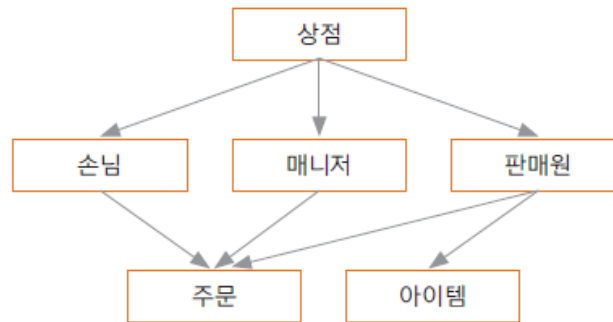
■ 네트워크형 데이터베이스

- 네트워크형 데이터베이스는 데이터를 **노드**로 표현한 모델
- 노드(node)는
기본적으로 컴퓨터 네트워크를 이루는 기초 단위로,
네트워크상에 있으며 서로 대등한 관계임
- 네트워크형 데이터베이스는 계층형 데이터베이스의 단점인
데이터 중복 문제, 상하 종속 관계를 해결했지만
구조가 복잡하여 변경, 운영하기 어렵고 데이터 종속성 문제까지 있음



네트워크형 데이터베이스의 예

■ 네트워크형 데이터베이스

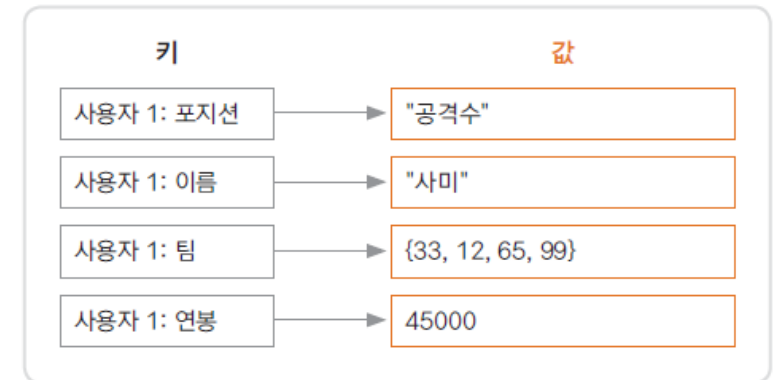


네트워크형 데이터베이스의 예

- 상점은 주인 노드(owner node)이고 손님, 매니저, 판매원은 멤버 노드(member node)
- 이 멤버 노드들은 다시 주인 노드가 되어 주문, 아이템이라는 노드를 가짐
- 멤버 노드는 주인 노드가 여러 개 있는데, 멤버 노드는 반드시 주인 노드와 연결되어야 하며 주인 노드에 있는 레코드와 연관되어야 함
- 이처럼 네트워크형 데이터베이스는 레코드 간의 관계를 일대다 또는 다대다로 표현할 수 있지만, 종속성 문제가 생기기 쉬워 데이터베이스 구조를 변경하기 어려움

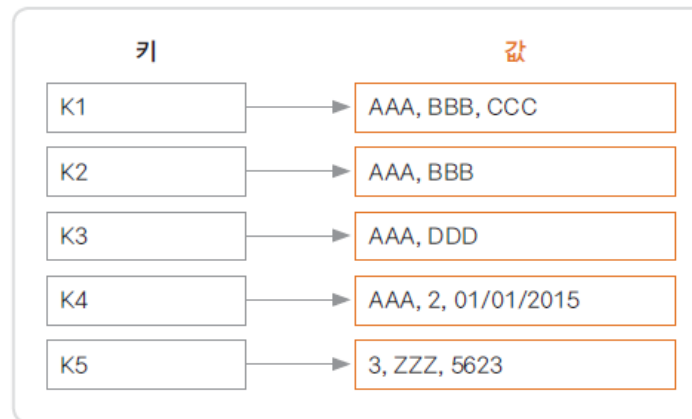
■ 키-값 데이터베이스

- 키-값 데이터베이스(key-value database)는 NoSQL의 한 종류로,
키-값을 일대일 대응해 데이터를 저장함
- 데이터 중복이 발생하며 비정형 데이터 저장에 유리함
- 키-값 데이터베이스는
관계형 데이터베이스와 함께 가장 많이 사용함
- 키-값 스토어(key-value store)라고도 하며,
키와 값 한 쌍으로 데이터를 저장하는 비관계형 데이터베이스 유형임
- 키-값 데이터베이스는 고유한 식별자로 사용하며
단순한 객체에서 복잡한 집합체에 이르기까지 무엇이든 키와 값이 될 수 있음



키-값 데이터베이스의 예

■ 키-값 데이터베이스



데이터가 여러 개인 키-값 데이터베이스의 예

- 그림을 보면 K1이라는 키에 AAA, BBB, CCC라는 값이 저장됨
- K2는 AAA, BBB 데이터가 저장되었는데 K1과 K2를 비교해 보면 데이터의 개수가 다름
- 키-값 데이터베이스는 스키마 없이 작동하므로 데이터 구조를 미리 정의할 필요가 없으며 시간이 지나더라도 언제든지 바꿀 수 있으므로 비정형 데이터를 쉽게 저장할 수 있음

■ 키-값 데이터베이스

■ 데이터 형태에 따른 키-값 작성 예

데이터 형태	작성 예	
데이터 테이블	Key	Value
	K1	AAA,BBB,CCC
	k4	AAA,2,01/01/2015
	k5	3,ZZZ,5623
JSON 형식	{ "k1": "AAA,BBB,CCC", "k4": "AAA,2,01/01/2015", "k5": "3,ZZZ,5623" }	
XML 스키마 표현	<k1>AAA,BBB,CCC</k1> <k4>AAA,2,01/01/2015</k4><k5>3,ZZZ,5623</k5>	

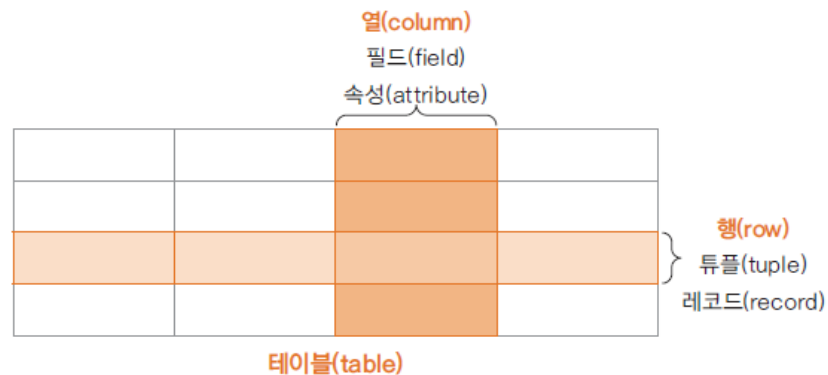
■ 키-값 데이터베이스

- 키-값 데이터베이스를 포함하는 NoSQL의 데이터베이스 종류

NoSQL 유형	특징	종류
키-값 데이터베이스 (Key-Value Database)	키-값 형태로 저장하며 수평 확장이 쉽다. 값의 내용으로 쿼리가 불가능하다.	Memcached, Redis, LevelDB 등
도큐먼트 데이터베이스 (Document Database)	키-값 모델이 진화한 형태이며 키-도큐먼트 형태로 저장된다. 값이 계층적인 형태로 저장된다.	MongoDB, CouchDB, MarkLogic 등
컬럼 데이터베이스 (Column Database)	키에 해당하는 값에 각기 다른 스키마를 가질 수 있으며 대용량 데이터 압축, 분산 처리, 집계 처리 등에 뛰어나다.	HBase, Cassandra, Hypertable 등
그래프 데이터베이스 (Graph Database)	데이터를 노드로 표현하며 노드 사이의 관계를 엣지로 표현한다. 소셜 미디어나 네트워크 다이어그램 등에서 사용할 수 있다.	Neo4j, Blazegraph, OrientDB 등

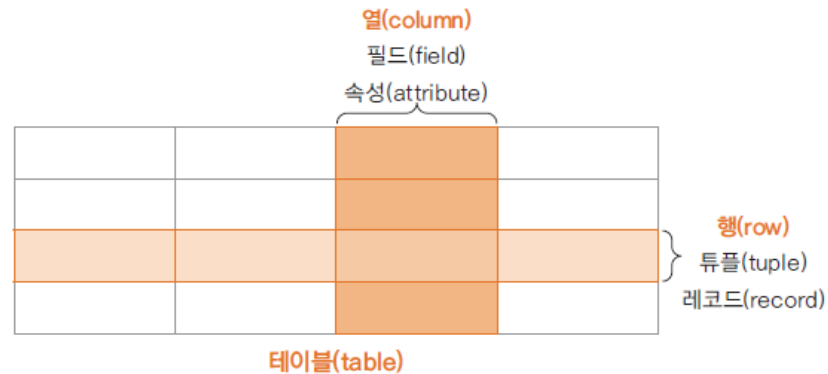
■ 관계형 데이터베이스

- 관계형 데이터베이스는 실무에서 가장 많이 사용하는 데이터베이스
- MySQL도 관계형 데이터베이스
- 관계형 데이터베이스에서는 데이터를 테이블 형태로 저장함



관계형 데이터베이스의 구조

■ 관계형 데이터베이스



관계형 데이터베이스의 구조

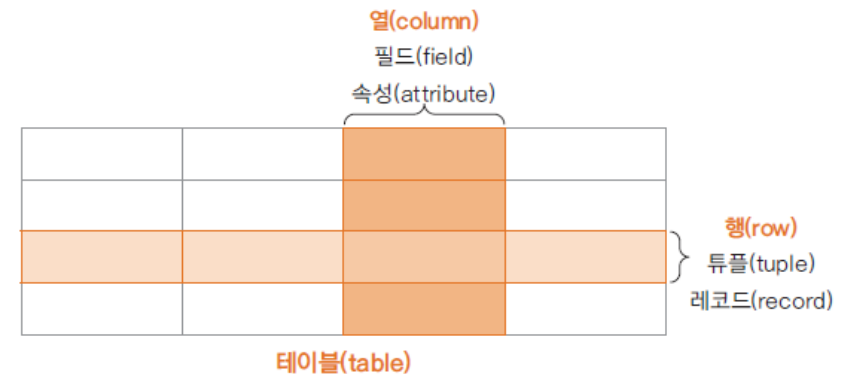
- 그림에서 보듯 관계형 데이터베이스는 데이터를 열과 행으로 구성한 테이블로 정리하며 **기본키**(primary key, PK)가 각 행을 식별함
- 데이터는 행 단위로 저장되며, 각 항목의 속성은 열이라 표현함
- 열 속성에 따라 데이터 유형이 정해짐

■ 관계형 데이터베이스

- 예를 들어 고객 정보를 저장한다고 할 때 [이름, 전화번호, 이메일, 주소]는 열이고

여기에 [강성욱, 000-000-0000-0000, xxxx@xxxx.xxx, 대한민국]으로
데이터가 저장되며 이러한 데이터를 행이라고 함

- 이렇게 고객 정보가 저장된 집합을
테이블이라고 함

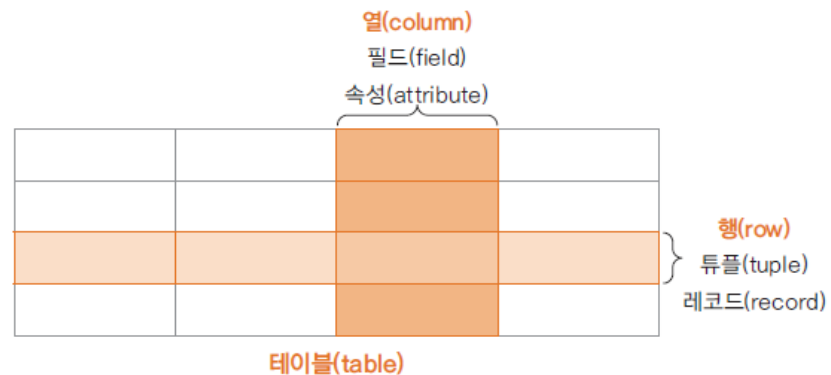


관계형 데이터베이스의 구조

■ 관계형 데이터베이스

■ 관계형 데이터베이스의 구성 요소

구성 요소	설명
열	각 열은 고유한 이름을 가지며 자신만의 타입을 가지고 있다. 열은 필드 ^{field} 또는 애트리뷰트 ^{attribute} 라고도 한다. 이 책에서는 열이라는 용어를 사용한다.
행	관계된 데이터의 묶음을 의미하며 한 테이블의 모든 행은 같은 수의 열을 가지고 있다. 행은 튜플 ^{tuple} 또는 레코드 ^{record} 라고도 한다. 이 책에서는 행이라는 용어를 사용한다.
테이블	행과 열 값들의 모음을 나타내는 것으로, 도메인 특성에 따라 데이터를 논리적으로 그룹화하여 모아 놓은 것이다.



관계형 데이터베이스의 구조

01-2

데이터베이스의 종류 알아보기

■ 관계형 데이터베이스

■ 엑셀로 알아보는 열, 행, 테이블

- 실제 데이터베이스에서 데이터를 조회한 후의 모습과 엑셀의 화면을 비교

[illegible]

	A	B	C	D	E	F	G	H	I	J	K
	Continent	Region	SurfaceArea	Population	LifeExpectancy	GDP	GDPPerCap	Location			
X	Asia	North America	Caribbean	1601.00	16600	78.1	838.00	799.00	Albania		
	Asia	Asia	Asia	32074.00	270000	67.9	507.00	N/A	Algeria		
	Asia	Asia	Central Africa	124470.00	2070000	38.5	6643.00	7094.00	Angola		
	Asia	North America	Caribbean	760.00	401.00	78.1	49.00	49.00	Anguilla		
	Asia	Asia	Southern Europe	3663.00	1992	68.700	71.6	94.00	230.00	Antigua	
Y	Asia	North America	Southern Europe	3663.00	1992	73.00	93.5	1630.00	N/A	Antigua	
Z	Asia	North America	Caribbean	908.00	10.7	81.700	74.7	1601.00	N/A	Aruba	
Y	Asia	Asia	South America	18693.00	19.7	54.00	74.1	2430.00	9949.00	Australia	
	Asia	South America	South America	2740.00	19.7	54.00	75.1	2430.00	223.00	Australia	
	Asia	Asia	South America	2740.00	19.7	53.000	56.4	181.00	167.00	Bahamas	
	Asia	South America	South America	2740.00	19.7	53.000	75.1	234.00	N/A	Bahamas	
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
Z	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
Z	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		
	Asia	Asia	South America	191.0000.00	83.1	8	N/A	0.00	N/A		

데이터베이스와 엑셀의 구조 비교(왼쪽은 데이터베이스 조회 화면, 오른쪽은 엑셀 화면)

- **관계형 데이터베이스**

- **엑셀로 알아보는 열, 행, 테이블**

- 엑셀 화면을 보면 데이터베이스에서 저장된 구조와 동일하게 표현한 것을 볼 수 있음
 - 엑셀의 A, B, C, D, ...가 데이터베이스의 열이고,
엑셀의 1, 2, 3, 4, ... 번호에 채워진 데이터가 데이터베이스의 행
 - 엑셀 시트에 정의된 데이터 집합에 대한 시트 이름이 데이터베이스의 테이블에 해당함

- **관계형 데이터베이스**

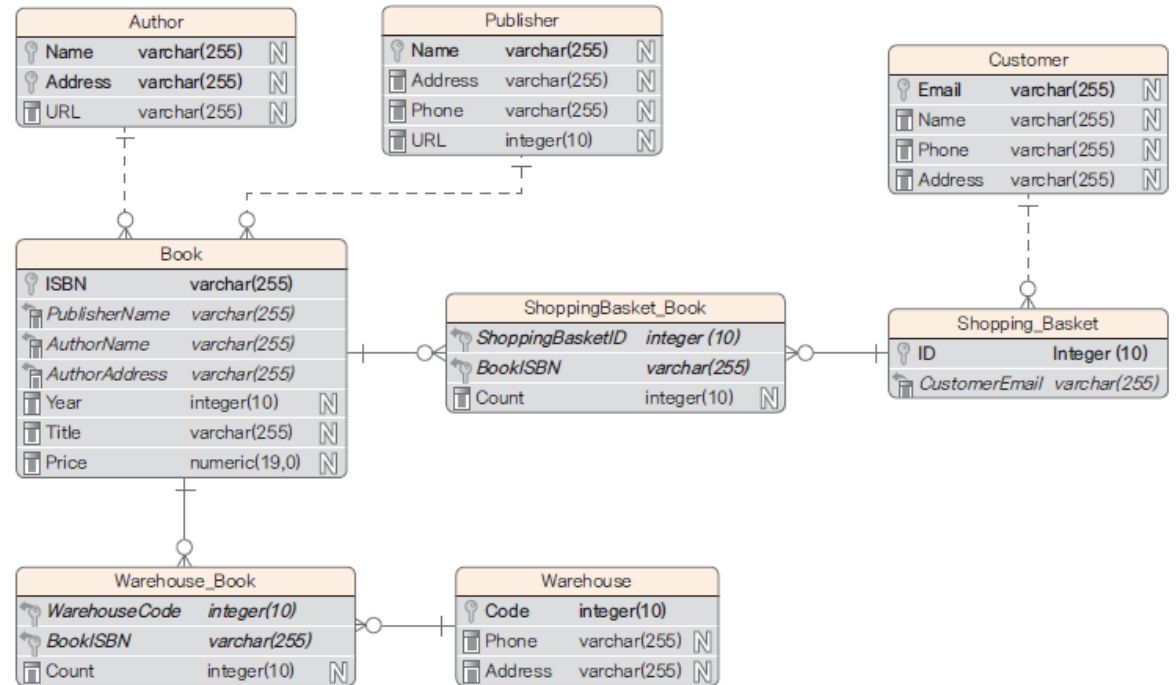
- **ERD로 이해하는 테이블 관계**

- **ERD**(entity relationship diagram)는 테이블 간의 관계를 표현한 그림
 - ERD는 논리 모델과 물리 모델이 있는데
 - **논리 모델**은 데이터 모델의 첫 단계로 고객의 요구 사항을 수집 · 분석해 데이터베이스의 전체 모양을 구성하는 것
 - **물리 모델**은 논리 모델을 바탕으로 실제 데이터를 저장할 수 있는 모델을 표현한 것

■ 관계형 데이터베이스

■ ERD로 이해하는 테이블 관계

- 물리 모델을 표현한 ERD
 - 이 그림에서 네모 박스를 테이블이라고 가정
 - 네모 박스에 테이블 이름과 테이블에 포함되는 열이 적혀 있음

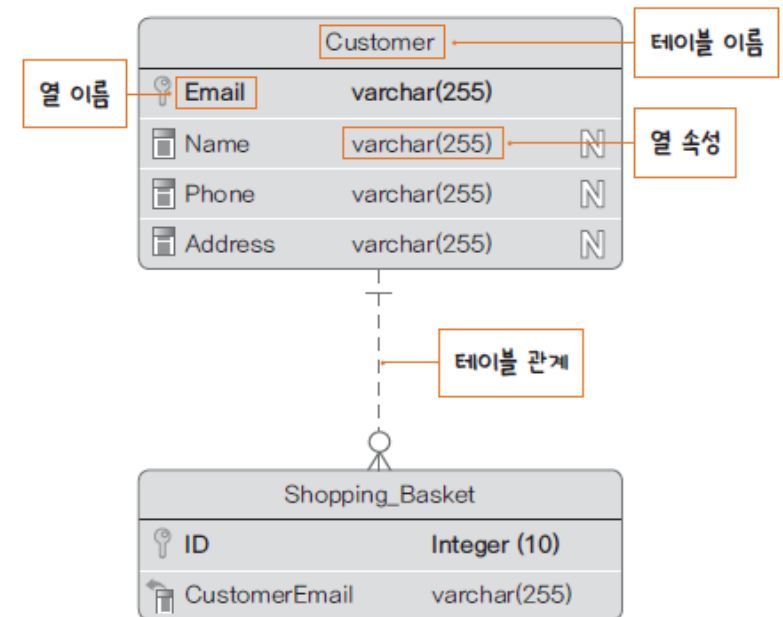


관계형 데이터베이스의 ERD

■ 관계형 데이터베이스

■ ERD로 이해하는 테이블 관계

- ERD의 요소에 주목
- 고객(Customer)과 장바구니(Shopping_Basket)의 관계를 ERD로 나타내고, 이를 확대해 각 요소의 명칭을 표시함
- 장바구니(Shopping_Basket) 테이블이 고객(Customer) 테이블을 참조하는 형태
- 장바구니 테이블의 CustomerEmail 열 요소는 사실 고객 테이블의 Email 열 요소를 참조하는 것



ERD 구성 요소의 명칭

■ 관계형 데이터베이스

■ 도서 주문 테이블로 알아보는 관계형 데이터베이스의 필요성

- 테이블 간에 관계를 가졌을 때 어떤 장점이 있는지 알아보자
- 다음은 고객이 도서 3권을 주문한 경우의 도서 주문 테이블
- 이 주문 테이블의 열에는 주문 번호, 회원 이름, 회원 주소, 주문 상품, 배송 주소가 있음

주문

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

■ 관계형 데이터베이스

■ 도서 주문 테이블로 알아보는 관계형 데이터베이스의 필요성

- 만약 이지스 고객의 회원 주소가 서울에서 부산으로 바뀐다면
주문 테이블에서 회원 이름이 이지스인 데이터를 모두 찾아
회원 주소를 서울에서 부산으로 수정해야 함

주문

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울 부산	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울 부산	Do it! Python	서울

■ 관계형 데이터베이스

■ 도서 주문 테이블로 알아보는 관계형 데이터베이스의 필요성

- 지금은 주문 테이블 하나만 수정하면 되므로 수정 자체가 큰 문제처럼 보이지 않음
- 하지만 주문 테이블뿐만 아니라 다음과 같이 장바구니, 쿠폰, 반품 테이블 등 다른 테이블에도 회원 주소가 포함되어 있다면 이 테이블을 모두 찾아 수정해야 함

주문

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

장바구니

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

쿠폰

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

교환

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

반품

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

멤버십

주문 번호	회원 이름	회원 주소	주문 상품	배송 주소
100	이지스	서울	Do it! SQL	서울
101	퍼블리싱	대전	Do it! SQL	대전
102	이지스	서울	Do it! Python	서울

■ 관계형 데이터베이스

■ 도서 주문 테이블로 알아보는 관계형 데이터베이스의 필요성

- 테이블을 분리하여
각 테이블의 목적에 맞는 데이터만 저장한 후, 참조 관계로 연결하여 이러한 문제를 해결함
- 데이터의 중복 예방과 관리 효율성을 추구함
- 관계형 데이터베이스는 앞에서 예를 든 도서 주문 테이블을 다음과 같이 재구성할 수 있음
- 기존 주문 테이블을 회원 테이블, 주문 테이블로 나눈 후, 회원 번호로 회원 테이블과 주문 테이블을 연결함

회원			주문			
회원 번호	회원 이름	회원 주소	주문 번호	회원 번호	주문 상품	배송 주소
1000	이지스	서울	100	1000	Do it! SQL	서울
1001	퍼블리싱	대전	101	1001	Do it! SQL	대전
1002	주식회사	제주도	102	1000	Do it! Python	서울

■ 관계형 데이터베이스

■ 도서 주문 테이블로 알아보는 관계형 데이터베이스의 필요성

회원			주문			
회원 번호	회원 이름	회원 주소	주문 번호	회원 번호	주문 상품	배송 주소
1000	이지스	서울	100	1000	Do it! SQL	서울
1001	퍼블리싱	대전	101	1001	Do it! SQL	대전
1002	주식회사	제주도	102	1000	Do it! Python	서울

- 이렇게 테이블을 구성하면 회원 주소가 변경되는 경우에는 회원 테이블만 수정하면 됨
- 주문 테이블은 회원 번호로 회원 테이블의 데이터를 참조하므로 주문 테이블의 데이터는 변경이 발생하지 않음

■ 관계형 데이터베이스

■ 대중적으로 알려진 관계형 데이터베이스 제품

제품 이름	상용 유무	특징
SQL 서버 ^{SQL Server}	상용	마이크로소프트에서 개발한 RDBMS
오라클 데이터베이스 ^{Oracle Database}	상용	오라클에서 개발한 RDBMS
DB2	상용	IBM에서 개발한 RDBMS
PostgreSQL	오픈 소스	버클리 대학교에서 개발한 RDBMS
MySQL	오픈 소스	오픈 소스 커뮤니티에서 개발한 RDBMS
SQLite	오픈 소스	오픈 소스 커뮤니티에서 개발한 RDBMS로 임베디드 시스템에 주로 사용

- SQL은 Structured Query Language의 줄임말
- 관계형 데이터베이스 관리 시스템(relational database management system, RDMBS)의 데이터를 관리하는 프로그래밍 언어
- 1970년대에 IBM에서 최초 개발했으며 관계형 모델이라는 이론에서 파생됨
- **SQL은 데이터베이스가 아닌 데이터베이스를 다루는 프로그래밍 언어**

▪ DBMS에 따른 SQL의 종류

- SQL은 국제 표준화 기관에서 표준화한 문법을 **ANSI SQL**이라 명명하여 발표함
- 하지만 SQL은 ANSI SQL만 있는 것이 아니고
DBMS 제조사마다 개발한 고유 SQL도 있음(각 SQL 이름은 다름)

DBMS	SQL 이름
SQL 서버	Transact-SQL(T-SQL)
오라클 데이터베이스	PL/SQL
MySQL	SQL
국제 표준	ANSI SQL

▪ SQL 문법의 종류

- SQL 문법은 데이터 정의 언어(DDL), 데이터 조작 언어(DML), 데이터 제어 언어(DCL)로 크게 3가지로 나눌 수 있음

▪ 데이터 정의 언어(DDL)

- 데이터 정의 언어(Data Define Language, DDL)는 데이터베이스를 정의하는 언어
- 데이터 정의 언어는 테이블 스키마를 관리하는 역할을 하며, 다음 명령어를 포함함

데이터 정의 언어

- CREATE: 데이터베이스 또는 테이블을 생성한다
- ALTER: 테이블을 수정한다.
- DROP: 데이터베이스 또는 테이블을 삭제한다.
- TRUNCATE: 테이블을 초기화한다.

▪ SQL 문법의 종류

▪ 데이터 조작 언어(DML)

- 데이터 조작 언어(Data Manipulation Language, DML)는 데이터베이스에 입력된 데이터를 조회 · 입력 · 수정 · 삭제하며, 다음 명령어를 포함함

데이터 조작 언어

- SELECT: 데이터를 조회한다.
- INSERT: 데이터를 입력한다.
- UPDATE: 데이터를 수정한다.
- DELETE: 데이터를 삭제한다.

▪ SQL 문법의 종류

▪ 데이터 제어 언어(DCL)

- 데이터 제어 언어(Data Control Language, DCL)는 데이터베이스에 접근하거나 객체에 권한을 부여하는 등의 역할을 함
- 데이터 제어 언어는 다음 명령어를 포함함

데이터 제어 언어

- GRANT: 특정 데이터베이스 사용자에게 작업에 대한 특정 수행 권한을 부여한다.
- REVOKE: 특정 데이터베이스 사용자에게 작업에 대한 특정 수행 권한을 삭제한다.
- COMMIT: 트랜잭션 작업을 완료하는 역할을 한다.
- ROLLBACK: 트랜잭션 작업을 취소 및 이전 상태로 복구하는 역할을 한다.

■ 데이터의 홍수 속에서 꼭 필요한 SQL

- 다양한 매체의 등장과 기술의 발달로 과거에 비해 새로 만들어지는 데이터가 많음
 - 동시에 데이터의 형태는 음성, 이미지, 텍스트 등 점점 다양해짐
 - 데이터는 많은 기업에서 중요한 의사결정을 할 때 사용
 - 다양한 부서에서 작은 단위로 데이터를 활용
-
- 다음 질문을 받았을 때, 데이터를 어떻게 추출하고 분석할까?

- 작년 대비 매출이 얼마나 증가했나요?
- 지역별로 어떤 상품이 잘 팔렸나요?
- 버튼 색깔에 따라 구매 비율이 어떻게 달라지나요?

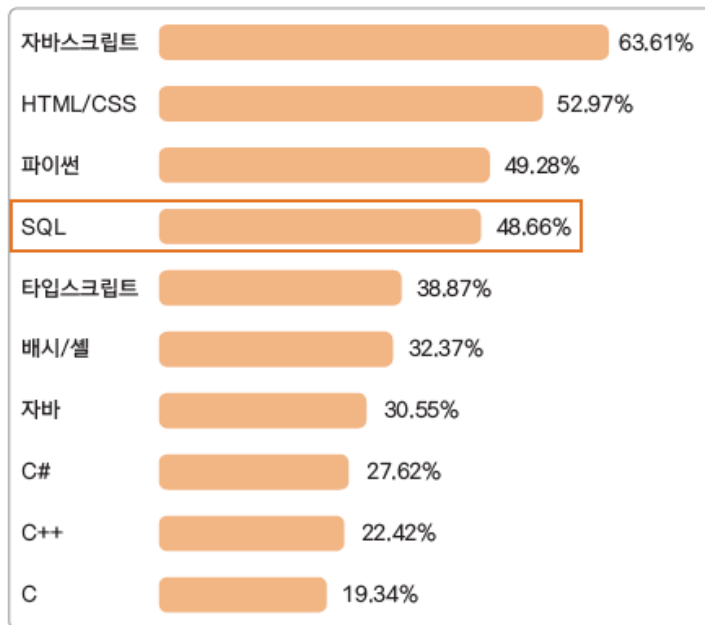
■ 데이터의 홍수 속에서 꼭 필요한 SQL

- 대부분은 엑셀을 활용하지만, 많은 양의 데이터를 처리하기 어려움
- SQL을 배우면 RDBMS, 빅데이터 시스템, NoSQL 시스템과 같은 다양한 형태의 데이터베이스를 다룰 수 있음
- SQL은 투자 대비 활용도가 매우 높은 매력적인 언어
- 프로그래머, 데이터 분석가, 데이터 엔지니어, 마케팅, 기획, 사업 등 다양한 분야에서 사용

01-4 SQL을 배워야 하는 이유

- SQL을 얼마나 사용하고 있을까?

- SQL은 오랫동안 인기 언어로 상위권을 유지하고 있음



스택오버플로에서 2023년 개발자를 대상으로 조사한 프로그래밍 언어 인기 순위

- SQL을 얼마나 사용하고 있을까?

- SQL을 배워야 하는 이유

1. 대부분의 데이터는 SQL을 사용할 수 있는 RDBMS에 저장되어 있다
2. DBMS에 저장되어 있는 데이터를 추출하려면 SQL 문법을 사용하여 컴퓨터에게 명령을 내려야 한다
3. RDBMS가 아닌 시스템에도 SQL과 비슷한 문법을 사용하여 데이터를 추출할 수 있다
4. 데이터 분석의 시작은 데이터 추출이다