

# 데이터베이스

SQL

## SQL의 중요성



- 관계형 데이터베이스는 결국 SQL에 의해서만 데이터 에 접근이 가능하기 때문에 데이터베이스를 기반으로 하는 정보시스템은 SQL 사용이 필수
- 정보시스템을 개발하는 수많은 개발자들은 SQL 사용 능력이 있으나 SQL의 수행 원리를 깊이 있게 이해하고 제대로 구사할 수 있는 전문적 지식을 갖춘 인재는 상대적 빈약하여 성능과 품질을 저하시키는 중요 원인
- SQL 개발자(SQL Developer)란 데이터베이스와 데이터 모델링에 대한 지식을 바탕으로 응용 소프트웨어를 개발하면서 데이터를 조작하고 추출하는데 있어서 정확하고 최적의 성능을 발휘하는 DB와 SQL을 작성할수 있는 능력을 필요로 함

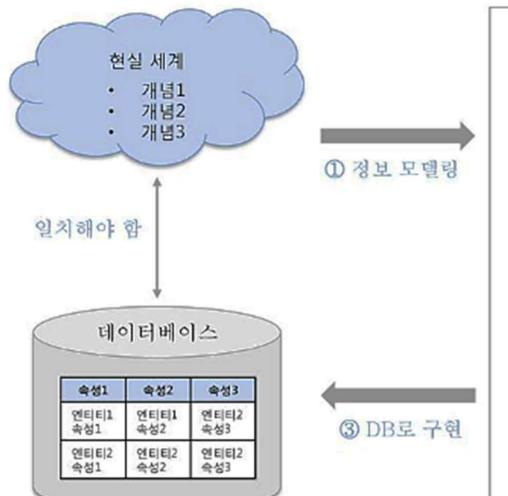
### SQL의 기본

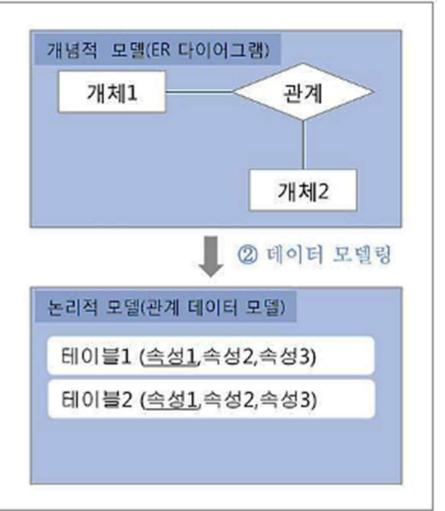


- 관계형 데이터베이스 개요
- DDL
- DML
- TCL
- WHERE 절
- 함수(FUNCTION)
- GROUP BY, HAVING 절
- ORDER BY 절
- 조인(JOIN)

## 데이터베이스의 개요







## ••• 데이터베이스(DataBase)의 개요



- 데이터베이스(DB: database)는 통합하여 관리되는 데이터의 집합체를 의미합니다.
- 이는 중복된 데이터를 없애고, 자료를 구조화하여, 효율적인 처리를 할 수 있도록 관리됩니다.
- 따라서, 여러 업무에 여러 사용자가 데이터 베이스를 사용할 수 있습니다.
- 이러한 데이터베이스는 응용 프로그램과는 다른 별도 의 미들웨어에 의해 관리됩니다.
- 데이터베이스를 관리하는 이러한 미들웨어를 데이터 베이스 관리 시스템(DBMS: Database Management System)이라고 합니다.



## 데이터베이스 모델링



- 데이터의 생성부터 운영, 관리까지 이르는 전체 프로 세스를 명확하고 구체적으로 표현(문서화)하는 것
  - 용량, 관리 인력 등 리소스에 한계
- 단계



### DBMS의 분류



 계층형(Hierarchical), 그래프(Graph), 망형(Network), 문서(Document), 관계형(Relational), 키-값(key-value), 객체지향형(Object-Oriented), 객체관계형(Object-Relational) 등

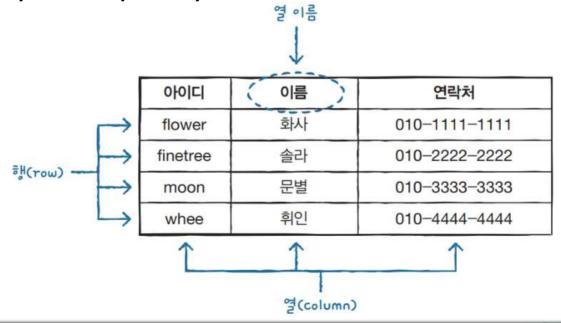
• 관계형 DBMS가 가장 많은 부분을 차지



## 관계형 DBMS



- Relational DBMS, RDBMS
- 대부분의 DBMS가 RDBMS 형태로 사용
- RDBMS의 데이터베이스는 테이블(table)이라는 최소 단위로 구성되며, 이 테이블은 하나 이상의 열 (column)과 행(row)으로 이루어져 있습니다.





### DBMS



- Database Management System
- 데이터베이스를 관리하고 운영하는 소프트웨어
- 종류

DBMS	제작사	작동 운영체제	기타
MySQL	Oracle	Unix, Linux, Windows, Mac	오픈 소스(무료), 상용
MariaDB	MariaDB	Unix, Linux, Windows	오픈 소스(무료), MySQL 초기 개발자들이 독립 해서 만듦
PostgreSQL	PostgreSQL	Unix, Linux, Windows, Mac	오픈 소스(무료)
Oracle	Oracle	Unix, Linux, Windows	상용 시장 점유율 1위
SQL Server	Microsoft	Windows	주로 중/대형급 시장에서 사용
DB2	IBM	Unix, Linux, Windows	메인프레임 시장 점유율 1위
Access	Microsoft	Windows	PC용
SQLite	SQLite	Android, iOS	모바일 전용, 오픈 소스(무료)



### SQL



- Structured Query Language
- 관계형 데이터베이스에서 사용되는 언어
- ISO(국제 표준화 기구, International Organization for Standardization)
  - 업체마다 상이한 점은 있으나 표준에는 맞추어 만들 어짐.
- 다른 개발 언어에 비해 기초 단계 학습은 쉬운 편
  - SQL이 시스템에 미치는 영향이 크므로 고급 SQL이나 SQL 튜닝의 중요성 강조



### SQL



- 단순 스크립트가 아니라 독립된 하나의 개발 언어
- 세미콜론(;)으로 분리되어 있는 SQL 문장 단위로 독립
- 데이터를 집합으로써 취급
- 특정 데이터들의 집합에서 필요로 하는 데이터를 꺼내 서 조회하고 새로운 데이터를 입력/수정/삭제하는 행 위를 통해서 사용자는 데이터베이스와 대화

## SQL의 종류



명령어의 종류	명령어	설명
데이터 조작어 (DML: Data Manipulation Language)	SELECT	데이터베이스의 데이터 조회 검색(매우 중요)
	INSERT / UPDATE / DELETE	데이터베이스의 테이블에 들어있는 데이터를 변형 에 사용되는 명령어 데이터의 추가,수정,삭제에 관련된 명령어
데이터 정의어 (DDL: Data Manipulation Language)	CREATE / ALTER / DROP / RENAME	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어 구조를 생성하거나 변경하거나 삭제하거나 이름을 바꾸는 데이터 구조와 관련된 명령어
데이터 제어어 (DCL: Data Control Language)	GRANT / REVOKE	데이터베이스에 접근 및 객체사용 권한을 부여하거나 회수하는데 사용되는 명령어
트랜잭션 제어어 (TCL: Transaction Control Language)	COMMIT / ROLLBACK	데이터의 조작 결과에 대한 논리적인 작업단위(트랜잭션) 를 제어하는 명령어

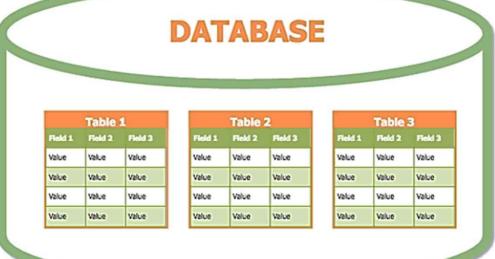


### **TABLE**



- 테이블은 데이터를 저장하는 객체(Object)로서 관계 형 데이터베이스의 기본 단위
- 모든 자료는 테이블에 등록이 되고, 테이블로부터 원 하는 자료를 SQL 이용하여 읽음

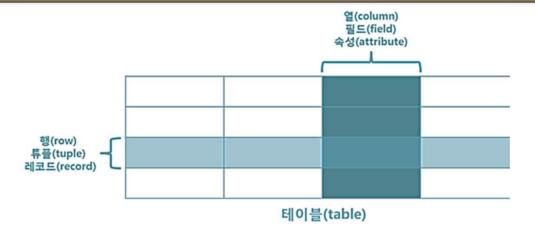
 테이블은 어느 특정한 주제와 목적으로 만들어지는 일 종의 집합





## **TABLE**





용어	설명
테이블	행과 컬럼의 2차원 구조를 가진 데이터 저장 장소
(Table)	데이터베이스의 기본 구성요소
컬럼 / 열	2차원 구조의 테이블에서 세로 방향을 이루는 하나하나의 특
(Column)	정 속성
로우 / 행	2차원 구조의 테이블에서 가로 방향을 이루는 연결된 데이터
(Row)	레코드(Record)라고도 부름

## 정규화(Normalization)



- 이상현상이 있는 릴레이션을 분해하여 이상현상을 없애는 과정
  - 이상현상이 존재하는 릴레이션을 분해하여 여러 개의 릴레이션을 생성
  - 데이터의 중복 속성을 제거하고 결정자에 의해 동일한 의미의 일반 속성이 하나의 테이블로 집약하여 테이블의 데이터 용량이 최소화
  - 단계별로 구분하여 정규형이 높아질수록 이상현상은 축소
- 장점
  - 이상 현상(Anomaly)을 제거
  - 데이터베이스와 연동된 응용 프로그램에 최소한의 영향만을 미 치게 되어 응용프로그램의 생명을 연장
- 단점
  - 릴레이션의 분해로 인해 릴레이션 간의 JOIN연산 증가
    - 질의에 대한 응답시간이 증가
    - 데이터를 처리할 때 속도가 빨라질 수도 있고 느려질 수도 있다
    - 성능저하가 나타나면 반정규화(De-normalization)를 적용

## 제 1정규형 (1NF)



- 1. 각 컬럼이 하나의 속성만을 가져야 한다.
- 2. 하나의 컬럼은 같은 종류나 타입(type)의 값을 가져 야 한다.
- 3. 각 컬럼이 유일한(unique) 이름을 가져야 한다.
- 4. 칼럼의 순서가 상관없어야 한다.

## 제 2정규형 (2NF)



- 1. 1정규형을 만족해야 한다.
- 2. 모든 컬럼이 부분적 종속(Partial Dependency)이 없어야 한다. == 모든 컬럼이 완전 함수 종속을 만족해야 한다.
  - 부분적 종속 : 기본키 중에 특정 컬럼에만 종속
  - 완전 함수 종속 : 기본키의 부분집합이 결정자가 되어선 안된다.

## 제 3정규형 (3NF)



- 1. 2정규형을 만족해야 한다.
- 2. 기본키를 제외한 속성들 간의 이행 종속성 (Transitive Dependency)이 없어야 한다.
  - 이행 종속성: A->B, B->C 일 때 A->C 가 성립하면 이행 종속



## OOO BCNF (Boyce-Codd Normal Form)



- 1. 3정규형을 만족해야 한다.
- 2. 모든 결정자가 후보키 집합에 속해야 한다.



## 4정규형 이상





## 데이터 모델



- 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화 하여 체계적으로 표현한 개념적 모형
- 데이터, 데이터의 관계, 데이터의 의미 및 일관성, 제약조건 등을 기술하기 위한 개념적 도구들의 모임
- 개념적 데이터 모델
  - 현실 세계의 개체를 인간이 이해할 수 있는 정보구조로 표현 정보모델
  - E-R 모델
- 논리적 데이터 모델
  - 개념적 구조를 컴퓨터 처리 가능한 상태로 변환하는 과정
  - 관계모델, 계층모델, 네트워크 모델

## 데이터 모델링의 순서



- 1. 업무 파악/ 요구사항 분석
  - 해당하는 업무 혹은 요구에 대한 분석 및 파악
- 2. 개념적 데이터 모델링
  - 업무와 요구의 데이터 간의 관계를 구상
  - 엔티티-관계 다이어그램(E-R diagram)
- 3. 논리적 데이터 모델링
  - 구체화된 데이터 모델(key,attribute,relation 등)
  - 정규화, 테이블 형태로 구성
- 4. 물리적 데이터 모델링
  - 데이터 베이스를 선택, 코딩



## 업무 파악 /요구 분석



• 게人吐 Plipop RANG HOME SHOP LOOKBOOK REVIEW COMMUNITY



#### Q&A

NOTICE FAQ Q&A 1:1 문의 CLASS 입고정与

No	제목	글쓴이	작성시간
P	2021년 입고정보게시판 신설	PLIPOP	2021-01-26
16	△ 신상입고 기대합니다!	PLIPOP	2021-01-26
15	△ 무통장입금 가능한가요?	PLIPOP	2021-01-26
14	△ 신상입고 언제 되요? ㅠㅠㅠㅠ	PLIPOP	2021-01-26
13	△ 친절한 응대 감사합니다 ^^	PLIPOP	2021-01-26
12	△ 프로모션으로 상품을 구매했는데요.	PLIPOP	2021-01-26
11	☆ 배송 언제 되나요?	PLIPOP	2021-01-26
10	△ 새상품 입고 언제 되나요?	PLIPOP	2021-01-26
9	△ 스키니진 블루블랙 언제 입고되나요?	PLIPOP	2021-01-26
8	△ [문의] 문의드립니다.	익명	2019-09-24
7	△ 이거 이름을 모르겠는데 재입고 언제 되나요?	익명	2019-09-24

Search Q

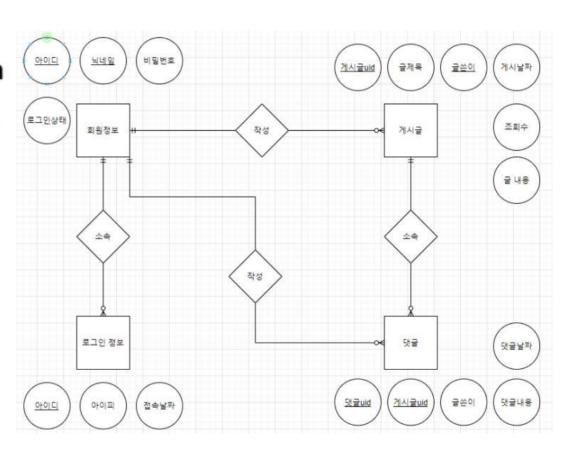
글쓰기



## 개념적 데이터 모델링



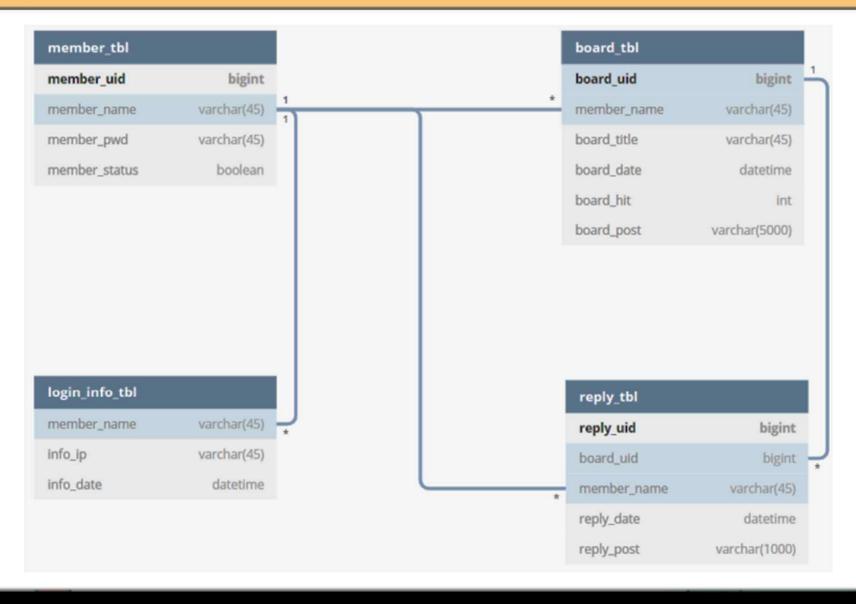
- ERD(Entity Relation Diagram)
  - □ Entity
- → Table
- O Attribute
- → Column
- Relation
- ⇒ PK, FK
- Tuple
- → Row





## 논리적 데이터 모델링







## 물리적 데이터 모델링



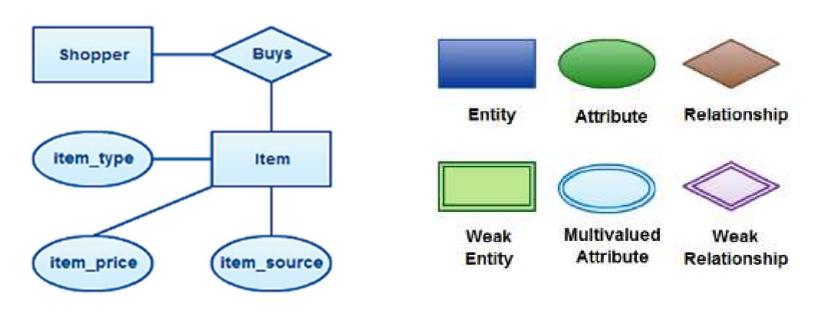
SQL

```
create table member tbl (
  member_uid bigint primary key auto_increment,
  member_name varchar(45) unique not null,
 member pwd varchar(45) not null,
 member status boolean not null
);
create table login_info_tbl(
  member_name varchar(45) not null,
  info ip varchar(45) not null,
  info_date datetime not null,
  constraint fk member name foreign key (member name) references member tbl (member name)
);
create table board_tbl (
  board_uid bigint primary key auto_increment,
 member name varchar(45) not null,
  board title varchar(45) not null,
  board_date datetime not null,
  board hit int not null,
  board_post varchar(5000) not null,
  constraint fk_member_name foreign key(member_name) references member_tbl(member_name)
```

## ••• ERD(Entity Relationship Diagram)



- 테이블 간 서로의 상관 관계를 그림으로 도식화
- 구성 요소
  - 개체(Entity)
  - 관계(Relationship)
  - 속성(Attribute)
- 현실 세계의 데이터는 이 3가지 구성 요소로 모두 표현 가능



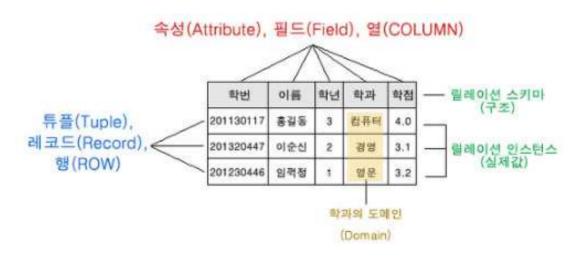


## Entity



- 정의 가능한 사물 또는 개념
- 데이터베이스 설계 시 대개 테이블로 정의
- 속성(Entity Attribute)
  - 엔티티의 포함된 속성 (컬럼명)
  - 도메인(Domain,속성의 값,타입,제약등 범위의 표현)





		0
0	O	O

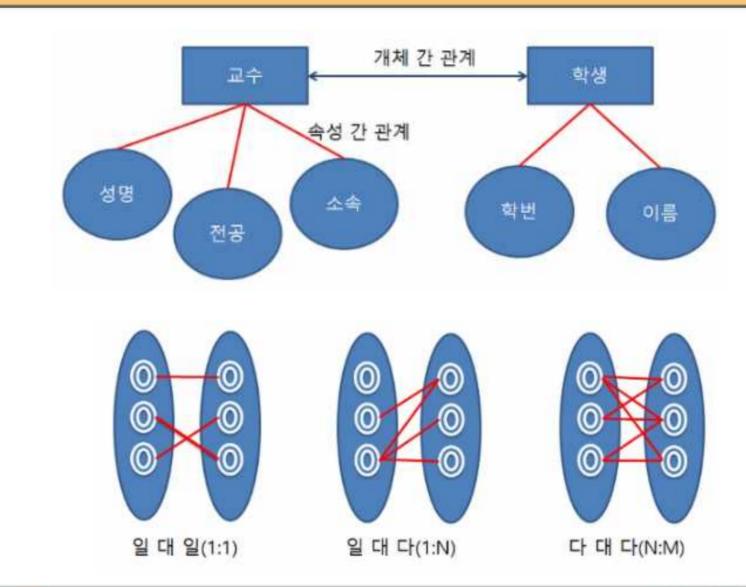
## 엔티티의 분류



구 분	내용
유형 엔티티	물리적인 형태 (예 : 고객, 상품, 거래처, 학생, 교수 등)
무형 엔티티	물리적인 형태가 없고 개념적으로만 존재하는 엔티티 (예 : 인터넷 장바구니, 부서 조직 등)
문서 엔티티	업무 절차상에서 사용되는 문서나 장부, 전표에 대한 엔티티 (거래명세서, 주문서 등)
이력 엔티티	업무상 반복적으로 이루어지는 행위나 사건의 내용을 일자별, 시간별로 저장하기 위한 엔티티 (예:입고이력,출고이력,구매이력 등)
코드 엔티티	무형 엔티티의 일종으로 각종 코드를 관리하기 위한 엔티티 (예:국가코드, 각종 분류 코드)

••• 관계

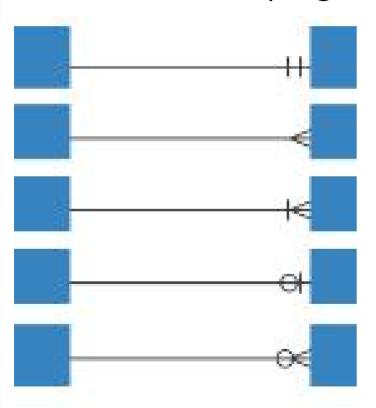




## 기호의 종류



- 1:1개 / 실선은(dash) '1'을 나타낸다.
- ∈: 여러개 / 까마귀 발(crow's foot or Many)은 '다수' 혹은 '그 이상'을 나타낸다.
- ○: 0개 / 고리 (ring or Optional)은 '0'을 나타낸다.

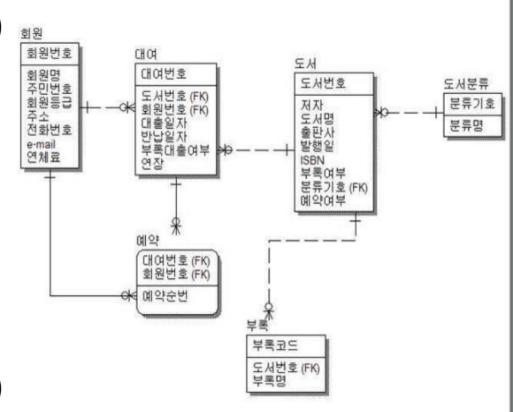


- •실선과 실선 정확히 1:1
- •까마귀발 여러개
- •실선과 까마귀발 1개 이상
- •고리와 실선 0 혹은 1
- •고리와 까마귀발 0개 이상

### ERD 예제



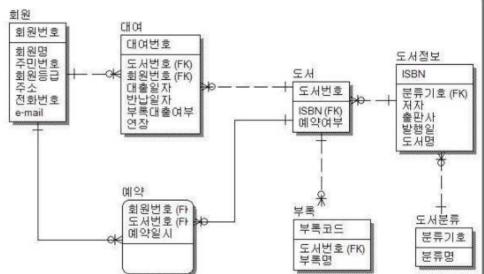
- [회원 ↔ 대여]
  - 회원번호PK가 대여 테이블에서 FK로 일 반속성으로 쓰이고 있다. (점선)
  - 회원은 대여를 여러개 할 수 있다. (1:N)
  - 아예 대여하지 않은 회원이 있을 수 있다. (1:N(선택))
  - 대여를 할땐 반드시 회원 정보가 필수로 존재해야한다. (1[필수]:N[선택])
- [도서 ↔ 대여]
  - 도서번호PK가 대여 테이블에서 FK로 일 반속성으로 쓰이고 있다. (점선)
  - 도서가 과거에 여러번 대여된 기록이 있을 수 있으니. (1:N)
  - 아예 대여하지 않은 도서가 있을 수 있다. (1:N[선택])
  - 대여를 할땐 반드시 도서 정보가 필수로 존재해야한다. (1[필수]:N[선택])
- [회원 ↔ 예약]
  - 회원번호PK가 예약 테이블에서 FK이자 PK로 쓰이고 있다. (실선)
  - 회원은 예약을 여러개 할 수 있다. (1:N)
  - 아예 예약하지 않은 회원이 있을 수 있다. (1:N[선택])
  - 예약을 할땐 반드시 회원 정보가 필수로 존재해야한다. (1[필수]:N[선택])



### ERD 예제



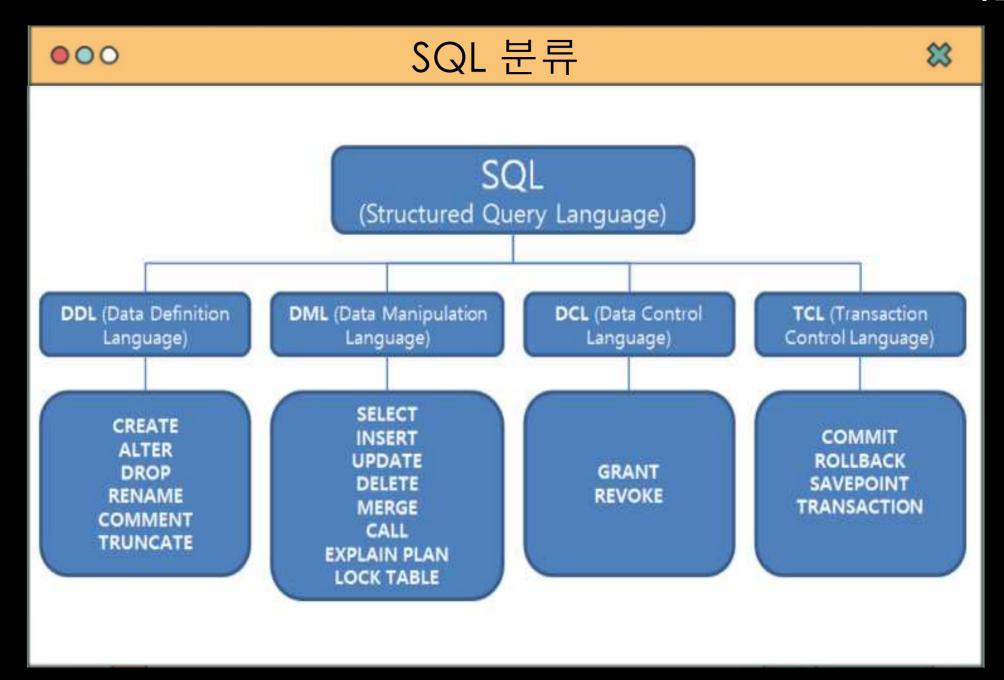
- [도서 정보 ↔ 도서]
  - ISBN(국제표준도서번호) PK가 도서 테 이블에서 FK로 일반속성으로 쓰이고 있다. (점선 )
  - 같은 책이 여러개 있을 수 있다. 도서 정보가 같은 도서가 여러개. (1:N)
  - 유실된 책이 있을수 있다. 도서정보는 무형의 정보일 뿐이고, 도서 엔티티는 실제 물리적 엔티티이다. (1:N[선택])
  - 도서는 반드시 도서 정보가 필수로 존 재해야한다. ( 1[필수]:N[선택] )
- [도서 ↔ 예약]
  - 도서번호 PK가 예약 테이블에서 FK이 자 PK로 쓰이고 있다. (실선)
  - 하나의 도서에 여러개 예약이 걸려 있을수 있다. (1:N)
  - 예약이 없는 도서가 있을 수 있다. (1:N[선택])
  - 예약 정보에는 반드시 도서 정보가 들어 있어야 한다. (1[필수]:N[선택])



## SQL 기본



- 특정 데이터들의 집합에서 필요로 하는 데이터를 꺼내서 조회하고 새로운 데이터를 입력/수정/삭제하는 행위를 통해서 사용자는 데이터베이스와 대화
- SQL(Structured Query Language)은 이러한 대화를 가능하도록 매개 역할
- SQL은 관계형 데이터베이스의 데이터 정의, 데이터 조작, 데이터 제어를 위해 사용하는 언어



## SQL의 종류



명령어의 종류	명령어	설명
DQL(Data Query lang)	SELECT	데이터베이스의 데이터 조회 검색(매우 중요)
데이터 조작어 (DML: Data Manipulation Language)	INSERT / UPDATE / DELETE	데이터베이스의 테이블에 들어있는 데이터를 변형에 사용되는 명령어데이터의 추가,수정,삭제에 관련된 명령어
데이터 정의어 (DDL: Data Manipulation Language)	CREATE / ALTER / DROP / RENAME	테이블과 같은 데이터 구조를 정의하는데 사용되는 명령어 구조를 생성하거나 변경하거나 삭제하거나 이름을 바꾸는 데이터 구조와 관련된 명령어
데이터 제어어 (DCL: Data Control Language)	GRANT / REVOKE	데이터베이스에 접근 및 객체사용 권한을 부여하거나 회수하는데 사용되는 명령어
트랜잭션 제어어 (TCL: Transaction Control Language)	COMMIT / ROLLBACK	데이터의 조작 결과에 대한 논리적인 작업단위 (트랜잭션) 를 제어하는 명령어

# ODL(Data Definition Language)



- 데이터베이스를 정의하는 언어
- 데이터를 생성하거나 수정 삭제 등 데이터의 전체 구
   조를 결정하는 역할의 언어
- CREATE
- ALTER
- DROP
- TRUNCATE

# 테이블 생성 규칙



- 테이블명은 객체를 의미할 수 있는 적절한 이름을 사용
  - 가능한 단수형을 권고
- 테이블명 중복 불가
- 한 테이블 내에서는 칼럼명 중복 불가
- 테이블 이름을 정하고 각 칼럼들은 괄호 "() " 로 묶어 지정
- 각 칼럼들은 콤마","로 구분되고, 항상 끝은 세미콜론 ";"
- 칼럼에 대해서는 다른 테이블까지 고려하여 데이터베이스 내에서는 일관성 있게 사용 권장 (데이터 표준화 관점)
- 칼럼 뒤에 데이터 유형은 반드시 지정
- 테이블명과 칼럼명은 반드시 문자로 시작
  - 벤더별로 길이 제한
- 벤더에서 사전에 정의한 예약어(Reserved word) 사용불가
- A-Z, a-z, 0-9, \_, \$, # 문자만 허용

# 테이블 정보



- 테이블명
  - PLAYER
- 테이블설명
  - 현재 게임을 하는 선수들의 정보
- 컬럼명
  - PLAYER\_ID(선수ID) 문자 고정 자릿수 7자리,
     PLAYER\_NAME(선수명) 문자 가변 자릿수 30자리,...
- 제약조건
  - 기본키(PRIMARY KEY)->PLAYER\_ID / 제약조건명 PLAYER\_PK
  - NOT NULL->PLAYER\_NAME,.....

# 제약조건



- PRIMARY KEY (기본키)
- UNIQUE(고유키)
- NOT NULL
- CHECK
- FOREIGN KEY(외래키)

# Table의 관계



- 기본키 (Primary Key)
  - Data-Table에 있는 유일하게 구분되는 Data-Key
  - 유일한 값이기 때문에 중복 불가
  - Primary Key는 공백 불가
- 외래 키 (Foreign Key)
  - 한 Table과 참조되는 다른 Table 간의 연결되는 Primary Key Column
  - Foreign Key는 다른 Primary Key를 참조하는 속성 또는 속성들의 집합
  - Foreign Key는 참조관계의 기본 키와 같은 속성
- 후보키 (Candidate Key)
  - Primary Key와 Foreign Key와 같은 각 Data를 식별하기 위해 사용되는 속성들의 집합
  - Candidate Key는 유일성과 최소성을 모두 만족
    - 유일성(Unique): 하나의 Key 값으로 하나의 Data를 식별하는 성질
    - 최소성(Minimality) : 모든 Data를 유일하게 식별하는 속성으로만 구성되는 성질

# 기본키 vs 외래키



Foreign key	
foreign key는 두 테이블 사이에 데이 터를 연결 시켜주는 관계형 데이터베 이스 테이블에 컬럼이다.	
다른 테이블의 primary key를 참조 할 수 있다.	
테이블 내에 한개 이상의 foreign key 가 허락된다.	
관계형 데이터베이스 내에 중복 값과 테이블을 포함할 수 있다.	
Null 값이 가능하다.	
상위 테이블에서 해당 값을 삭제 할 수 있다.	

×

Address

Delhi

Delhi

Delhi

#### Table의 관계 000 Enroll No RollNo Address DeptID. DeptID Name Name FK\_Relationship 60391481 Avin AX101 Dehi IT 60391402 Possible Candidate Keys Amit. AX102 Noida Mechanical 3 60391403 Mohan AX103 Ghaziabad 2 **Bectrical** 60391404 Payan AX104 Ghaziabad 2 Foreign Key Dehi 60391405 Deepak AX105 3 60391406 Jitendra AX106 Dehi **Primary Key** copyright dotnet-tricks.com Primary Key Unique Key Alternate Keys Super Key

000	문자 데이터	유형
-----	--------	----



데이터타입	설명		
CHAR(n)	고정길이 문자 / 최대 2000byte / 디폴트 값은 1byte		
VARCHAR2(n)	가변길이 문자 / 최대 4000BYTE / 디폴트 값은 1byte		
NCHAR(n)	고정길이 유니코드 문자(다국어 입력가능) / 최대 2000byte / 디폴트 값은 1byte		
NVARCHAR(n)	가변길이 유니코드 문자(다국어 입력가능) / 최대 2000byte / 디폴트 값은 1byte		
LONG	최대 2GB 크기의 가변길이 문자형		
CLOB	대용량 텍스트 데이터 타입(최대 4Gbyte)		
NCLOB	대용량 텍스트 유니코드 데이터 타입(최대 4Gbyte)		
1글자만 사용할 것이 아니라면 대부분 VARCHAR2를 사용			

# ••• 숫자형 데이터 타입



이터 타입	설명
NUMBER(P,S)	가변숫자 / P (1 ~ 38, 디폴트 : 38) / S (-84 ~ 127, 디폴트 값 : 0) / 최대 22byte
FLOAT(P)	NUMBER의 하위타입 / P (1~128 .디폴트 : 128) / 이진수 기준 / 최대 22byte
BINARY_FLOAT	32비트 부동소수점 수 / 최대 4byte
BINARY_DOUBLE	64비트 부동소수점 수 / 최대 8byte

- NUMBER형을 사용
  - ✓ P는 소수점을 포함한 전체 자릿수를 의미
  - ✓ S는 소수점 자릿수를 의미
- NUMBER는, 가변숫자이므로 P와S를 입력하지 않으면 저장 데이터 의 크게에 맞게 자동으로 조절



# 날짜 데이터 타입



데이터타입	설명
DATE	BC 4712년 1월 1일부터 9999년 12월 31일, 연, 월, 일, 시, 분, 초까지 입력 가능
TIMESTAMP	연도, 월, 일, 시, 분, 초 + 밀리초까지 입력가능

일반적으로 쓰는 데이터타입은 DATE타입



# LOB 데이터 타입



데이터 타입	설명
CLOB	문자형 대용량 객체 고정길이와 가변길이 문자집합 지원
NCLOB	유니코드를 지원하는 문자형 대용량 객체
BLOB	이진형 대용량 객체
BFILE	대용량 이진 파일에 대한 위치,이름 저장

- LOB: Large Object의 약자로 대용량 데이터를 저장할 수 있는 데이터 타입
- 일반적으로 그래픽, 이미지, 사운드 등 비정형 데이터를 저장할 때 사용
- 문자형 대용량 데이터는 CLOB나 NCLOB, 그래픽, 이미지, 동영 상 등의 데이터는 BLOB를 주로 사용

# 테이블 생성



```
CREATE TABLE emp
  컬럼 명 ②
              데이터 타입③
                                기본 값 (생략가능)
                                                NULL(생략가능) 5
                NUMBER(4)
                                 DEFAULT 1000
                                                  NOT NULL,
    empno
                VARCHAR2(10),
    ename
                VARCHAR2(9),
    job
                NUMBER(4),
    mgr
    hiredate
                DATE,
                NUMBER(7,2),
    sal
                NUMBER(7,2),
    comm
    deptno
                NUMBER(2)
);
   1 테이블 명
   2 컬럼명
   3 데이터 타입
   4.기본 값
   5.Nullable
```



# 테이블 생성



```
CREATE TABLE emp
CREATE TABLE emp
                                                                                   NOT NULL.
                                                                    NUMBER(4)
                                                         empno
                                 NOT NULL,
                NUMBER(4)
    empno
                                                                    VARCHAR2(10),
                                                         ename
                VARCHAR2(10),
    ename
                                                         iob
                                                                    VARCHAR2(9),
    job
                VARCHAR2(9),
                                                                    NUMBER(4),
                                                         mgr
                NUMBER(4),
    mgr
                                                         hiredate
                                                                    DATE,
    hiredate
                DATE.
                                                                    NUMBER(7,2),
                                                         sal
    sal
                NUMBER(7,2),
                                                                    NUMBER(7,2),
                                                         comm
                                                         deptno
                                                                    NUMBER(2),
                NUMBER(7,2),
    comm
                                                         CONSTRAINT emp pk PRIMARY KEY (empno)
                NUMBER(2)
    deptno
                                                     );
);
ALTER TABLE emp ADD CONSTRAINT emp_pk PRIMARY KEY (empn COMMENT ON TABLE emp IS '사원정보';
COMMENT ON TABLE emp IS '사원정보';
                                                     COMMENT ON COLUMN emp.empno IS '사원번호';
COMMENT ON COLUMN emp.empno IS '사원번호';
                                                     COMMENT ON COLUMN emp.ename IS '성명';
                                                     COMMENT ON COLUMN emp.job IS '직군';
                                                     COMMENT ON COLUMN emp.mgr IS '직속상사';
                                                     COMMENT ON COLUMN emp.hiredate IS '입사일';
                                                     COMMENT ON COLUMN emp.sal IS '급여';
                                                     COMMENT ON COLUMN emp.comm IS '보너스';
                                                     COMMENT ON COLUMN emp.deptno IS '부서코드';
```



## **CHECK - CK**



- 조건과 일치하는 데이터만 저장하는 제약조건.
- CHECK 다음 괄호 안에는 조건식(IN연산자, AND/OR, 비교연산자)과 함께 사용

```
-- 컬럼레벨
CREATE TABLE DEPARTMENT7
(DEPTNO NUMBER(2),
DNAME VARCHAR2(15) CONSTRAINT DEPARTMENT7_DNAME_CK CHECK(DNAME IN('개발', '인사')),
LOC VARCHAR2(15));
-- 테이블레벨
CREATE TABLE DEPARTMENTS
(DEPTNO NUMBER(2),
DNAME VARCHAR2(15),
LOC VARCHAR2(15),
CONSTRAINT DEPARTMENT8_DNAME_CK CHECK(DNAME IN ('개발', '인사'))
);
```

# 테이블 삭제



- DROP
- 테이블에 저장된 모든 데이터와 관련 인덱스, 제약조 건(FK제외)이 삭제

DROP TABLE 테이블이름;

- TRUNCATE
- 테이블의 데이터를 전부 삭제하고 사용하고 있던 공간을 반납(테이블 자체가 지워지는 것은 아님)

TRUNCATE TABLE 테이블명;

# 테이블 수정



- **ALTER** 
  - 컬럼추가
  - 컬럼 변경

  - 컬럼 삭제

  - 제약조건 삭제

ALTER TABLE 테이블명

ADD (컬럼 데이타타입 [DEFAULT]);

MODIFY (컬럼 데이타타입 [DEFAULT]);

컬럼이름변경 RENAME COLUMN 원래이름 TO 새 이름;

DROP(컬럼);

■ 제약조건 추가 ADD [CONSTRAINT 제약조건명] 제약조건타입(컬럼명);

DROP PRIMARY KEY | UNIQUE(컬럼) | CONSTRAINT 제약조건명 [CASCADE];

# OOD DML(Data Manipulation Language)



- 테이블의 행과 열을 조작하는 언어
- 정의된 데이터베이스의 입력된 레코드를 조회,수정, 삭제하는 역할의 언어
- 사용자의 질의어를 통해 저장된 데이터를 실질적으로 처리하는데 사용하는 언어
- SELECT
- INSERT
- UPDATE
- DELETE

# OCL(Data Control Language)



- 데이터 관리 목적으로 보안/무결성/회복/병행 제어 등을 정의하는데 사용되는 언어
- Transaction(TCL)
  - 하나의 Logical function을 수행하는 단위(작업 단위)
- COMMIT
  - Transaction의 변경내용 최종 반영
- ROLLBACK
  - Transaction의 변경내용 모두 취소하고 이전 상태로 회복
- GRANT
  - 사용자에게 권한을 부여
- REVOKE
  - 사용자에게 권한을 취소

마무리	**
	마무리