

DataBase

In computing, a database is an organized collection of data or a type of data store based on the use of a database management system (DBMS), the software that interacts with end users, applications, and the database itself to capture and analyze the data.

1. database

개념

여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합 -wiki

RDBMS (ReLational DataBase Management System)

- 관계형 데이터 베이스
- 정형 데이터들의 관계를 정의

NoSQL (Not only SQL)

- 비 정형 데이터 저장
- 빅데이터에 용이 (분산 저장 등)

2.rdb

개념

Entity 데이터베이스에 저장하려고 하는 현실상의 개념 / 객체

Attribute Entity의 속성

Tuple Entity의 값

Entity (table)

EMPNO	ENAME	JOB	SAL
1001	홍길동	developer	2000
1002	이순신	scientist	2000
1003	김선달	engineer	2000

Attribute (column)

Tuple (row)

3.sql

개념

Structured Query Language : 구조화 된 질의 언어

DDL (Data Definition Language)	데이터 정의 언어
DB 스키마 정의, 조작	

DML (data Manipulation Language)	데이터 조작 언어
Data 조작	

DCL (Data Control Language)	데이터 제어 언어
Data 제어	

3.sql

ddl

데이터 정의 언어 (Data Definition Language)

CREATE 데이터베이스, 테이블, 뷰(view), 프로시저(procedure) 등을 생성

ALTER 데이터베이스, 테이블, 뷰, 프로시저 등을 수정

DROP 데이터베이스, 테이블, 뷰, 프로시저 등을 삭제

* View : 다른 테이블/뷰 에 있는 데이터를 보여주기 위해 사용 (수정 불가)
Procedure : 특정 작업에 필요한 Query들을 함수처럼 사용

3.sql

dml

데이터 조작 언어 (Data Manipulation Language)

SELECT 데이터 읽기

INSERT 데이터 삽입

UPDATE 데이터 수정

DELETE 데이터 삭제

3.sql

dcl

데이터 제어 언어 (Data Control Language)

COMMIT 데이터, 트랜잭션 저장

ROLLBACK 데이터, 트랜잭션 취소 (가장 마지막 COMMIT으로 되돌아감)

GRANT DB 권한 부여

REVOKE DB 권한 삭제

* TCL (Transaction Control Language) : COMMIT, ROLLBACK

3.sql

key

키 (key) : 식별자(Identifier) → 튜플을 식별(검색, 정렬 등) 할 때 사용

슈퍼키 (Super key / Composite key)	복합키 라고도 하며, 유일성을 만족하는 속성
---------------------------------	--------------------------

후보키 (Candidate key)	유일성과 최소성을 만족하는 속성
---------------------	-------------------

기본키 (Primary key)	후보키 중 선택된 키
-------------------	-------------

대체키 (Alternate key)	후보키 중 선택되지 않은 나머지 키
---------------------	---------------------

외래키 (Foreign key)	다른 테이블의 컬럼이나 기본키를 참조
-------------------	----------------------

3.sql

정규화

정규화 (Normalization) : 데이터들의 관계를 구조화시켜 테이블을 분리

- | | |
|------|---------------------------|
| 1 NF | 하나의 컬럼엔 하나의 값 |
| 2 NF | 기본키에 종속적이지 않은 컬럼 분리 |
| 3 NF | 기본키를 제외한 나머지 컬럼들 간의 종속 불가 |
| BCNF | 기본키가 여러 개 존재할 경우, 복합키로 |
| 4 NF | 다치종속 제거 |
| 5 NF | 조인이 후보키를 통해서만 성립되도록 |

* 역정규화 (반정규화) : 필요할 경우 정규화를 거꾸로 진행

3.sql

constraint

데이터의 무결성 : 데이터의 정확성과 일관성 유지

제 약 조 건	설 명
NOT NULL	해당 컬럼에 NULL 입력 불가
UNIQUE	해당 컬럼에 중복되는 값 불가
PRIMARY KEY	각 행을 유일하게 식별
FOREIGN KEY	다른 테이블의 컬럼 값 참조
CHECK	해당 컬럼에 특정 값만 허용

3.sql

join

서로 관계가 있는 테이블을 하나로 연결

INNER JOIN

NATURAL JOIN

CROSS JOIN

OUTER JOIN

...

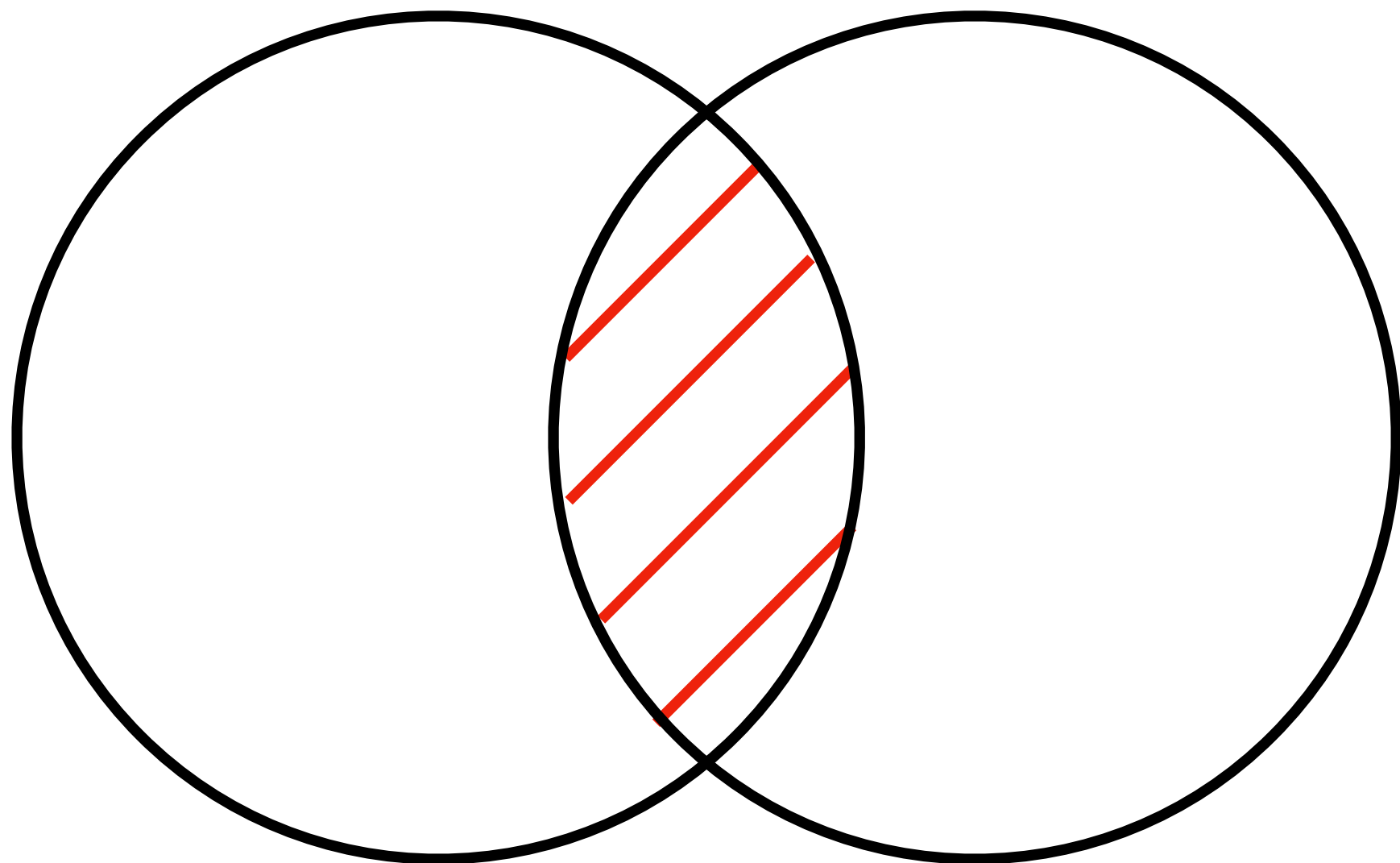
3.sql

inner

SELECT 컬럼 FROM 테이블A INNER JOIN 테이블B ON 테이블A.컬럼 = 테이블B.컬럼

SELECT 컬럼 FROM 테이블A INNER JOIN 테이블B USING(컬럼)

* 테이블A의 컬럼명과 테이블B의 컬럼명이 같으면 USING(컬럼)으로 사용 가능

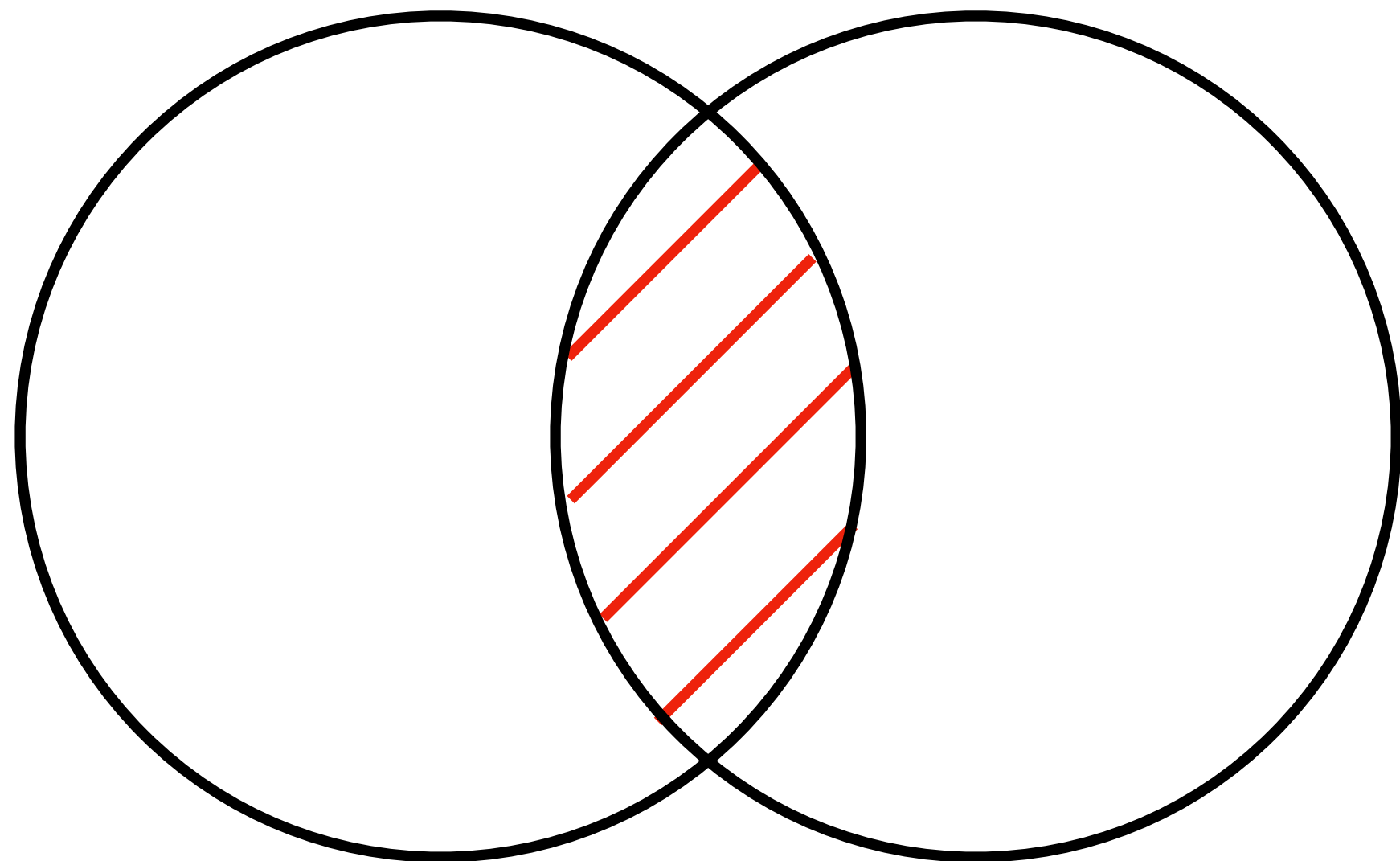


3.sql

natural

SELECT 컬럼 FROM 테이블A NATURAL JOIN 테이블B

* 테이블A의 컬럼명과 테이블B의 컬럼명이 단 하나만 같은 경우에 사용



3.sql

cross

SELECT 컬럼 FROM 테이블A CROSS JOIN 테이블B

SELECT 컬럼 FROM 테이블A JOIN 테이블B

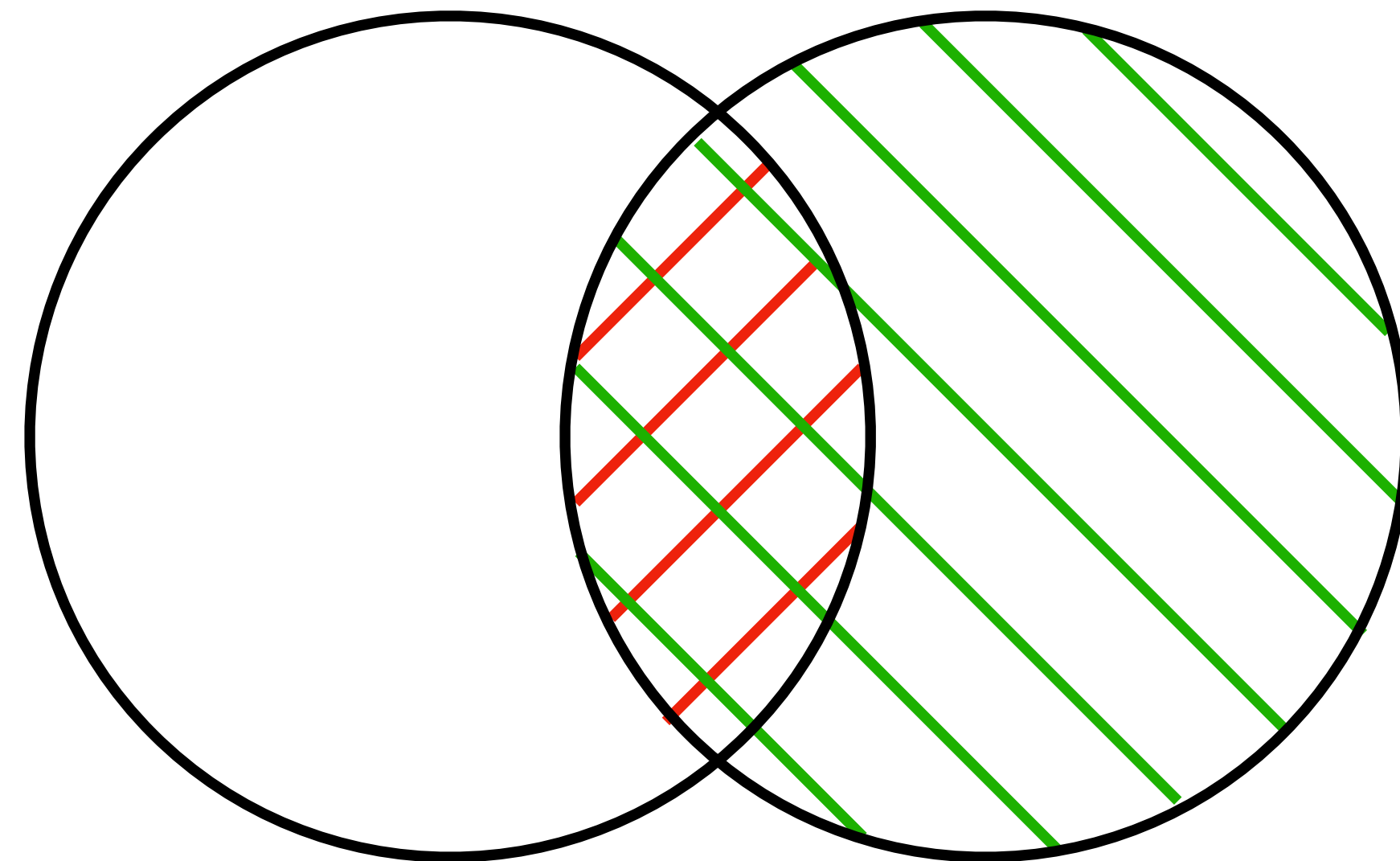
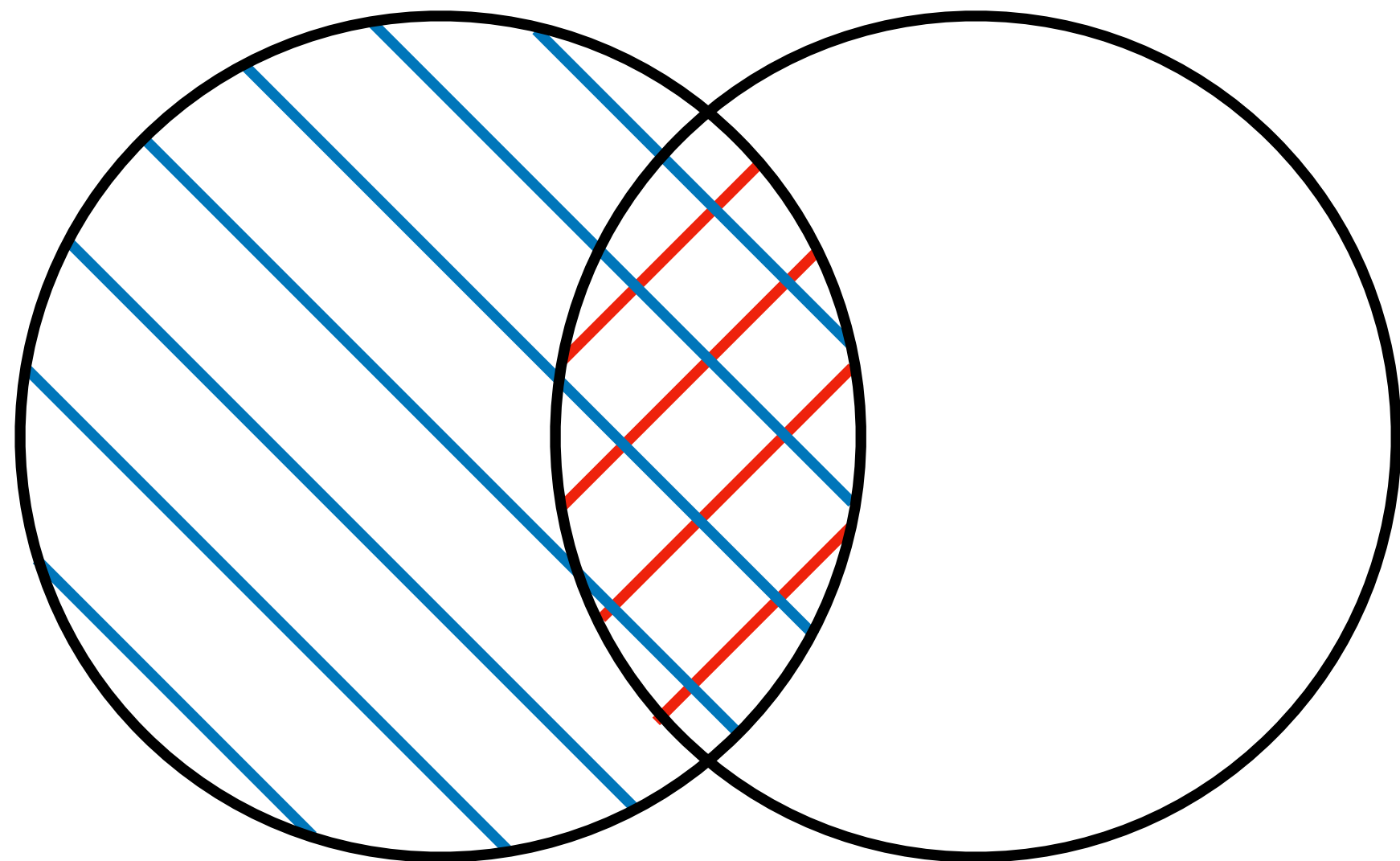
* CARTESIAN PRODUCTS

3.sql

outer

SELECT 컬럼 FROM 테이블A **LEFT JOIN** 테이블B ON 테이블A.컬럼 = 테이블B.컬럼

SELECT 컬럼 FROM 테이블A **RIGHT JOIN** 테이블B ON 테이블A.컬럼 = 테이블B.컬럼



3.sql

subquery

QUERY 내부에서 사용되는 SELECT문

중첩된 SELECT (Nested SELECT) 라고 부르기도 한다

SINGLE ROW SUBQUERY

결과가 1개의 값

MULTI ROW SUBQUERY

결과가 여러 개의 값

MULTI COLUMN SUBQUERY

WHERE 조건절에서 여러 개의 컬럼 값 비교

INLINE VIEW

FROM 절에서 사용 (가상 테이블)

4. ERD

종류

Entity Relationship Diagram

바커 표기법 (Barker Notation)

영국 컨설팅 회사 CACI에 의해 개발, 리차드 바커에 의해 업그레이드

I/E 표기법 (Information Engineering Notation)

CLIVE FINKELSTEIN과 JAMES MARTINO이 공동 저술로 발표

관계의 다(MANY) 를 나타내기 위해 까마귀 발을 사용 → 까마귀 발 모델이라고도 부른다
(CROW'S FOOT MODEL)

4. ERD

barker

Entity

: primary key
* : not null
0 : null



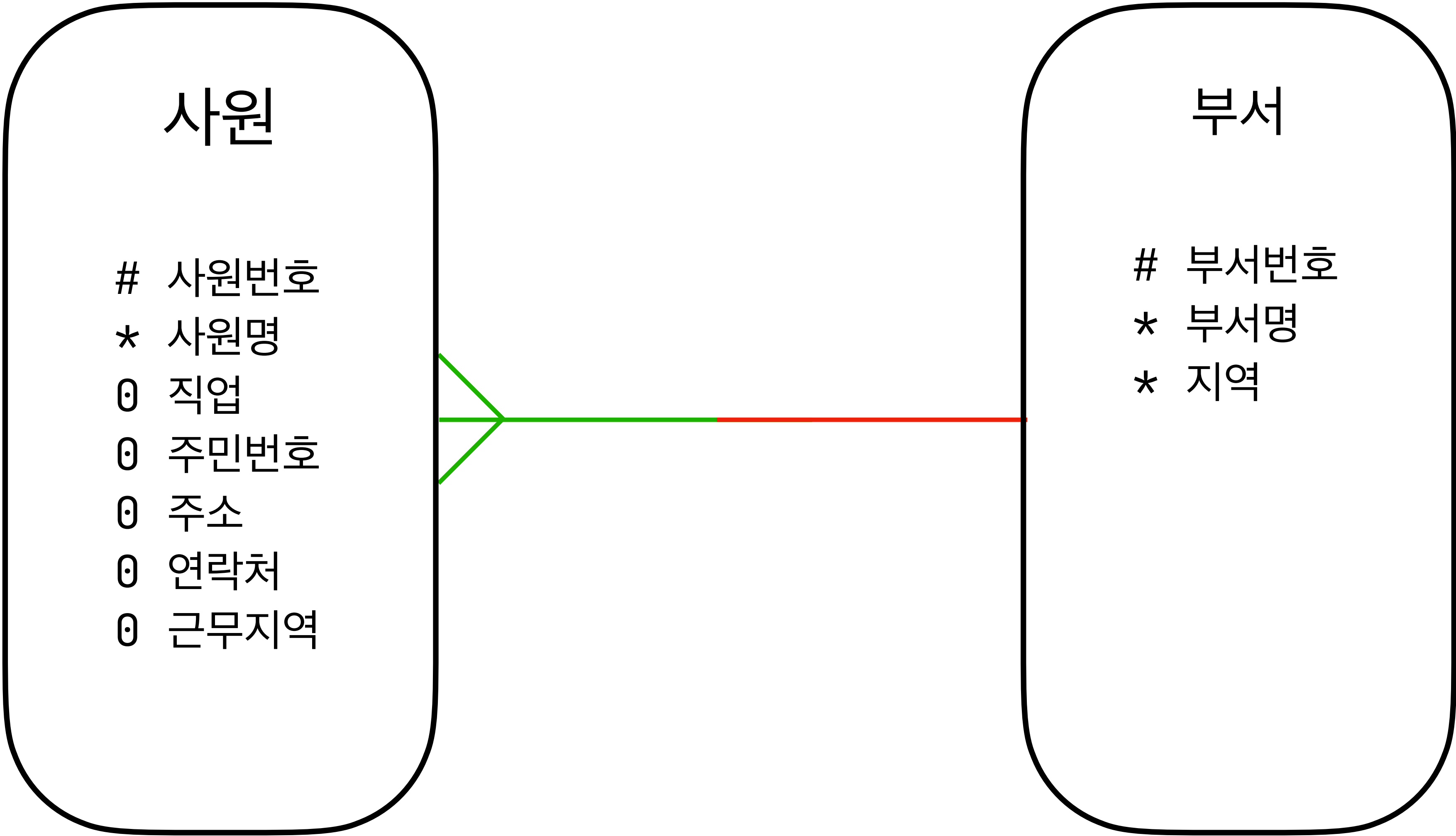
Attribute

4. ERD

barker

사원은 하나의 부서

부서는 한 명 이상의 사원



4 . ERD

barker

A : B 관계

1 : 1 A 에 존재하는 데이터 1개 <—> B 에 존재하는 데이터 1개

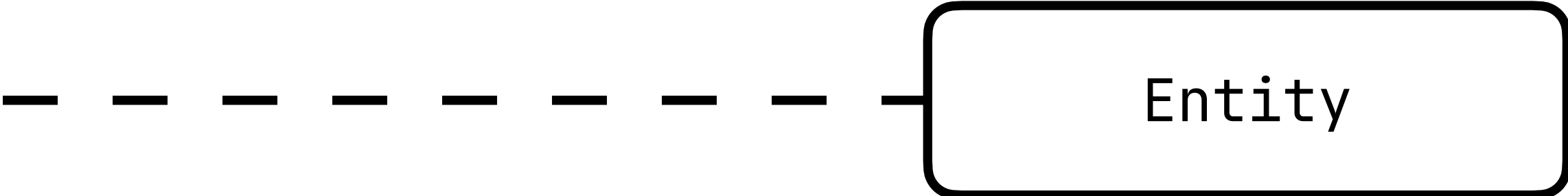
1 : N A 에 존재하는 데이터 1개 <—> B 에 존재하는 데이터 N개

N : 1 A 에 존재하는 데이터 1개 <—> B 에 존재하는 데이터 N개
 A 에 존재하는 데이터 N개 <—> B 에 존재하는 데이터 1개

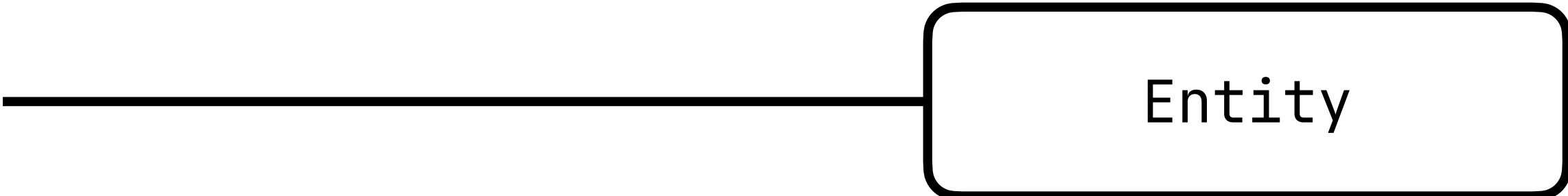
4. ERD

barker

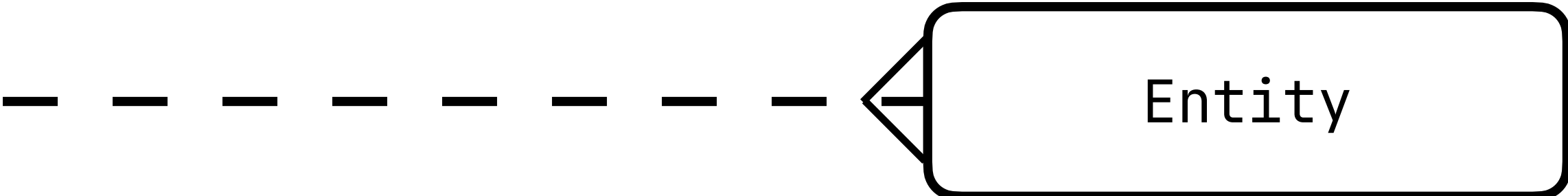
0 or 1



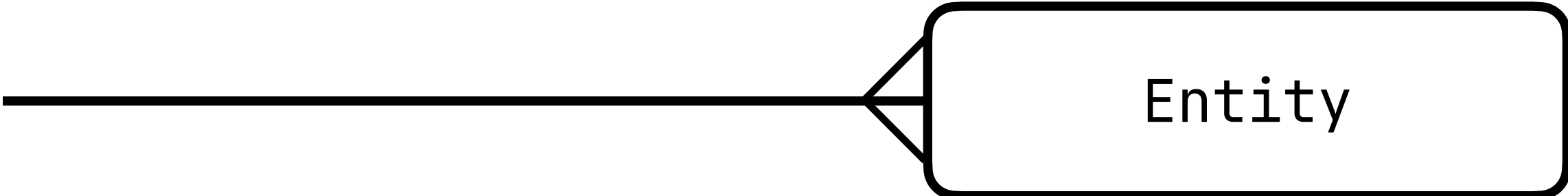
1



0 or more



1 or more



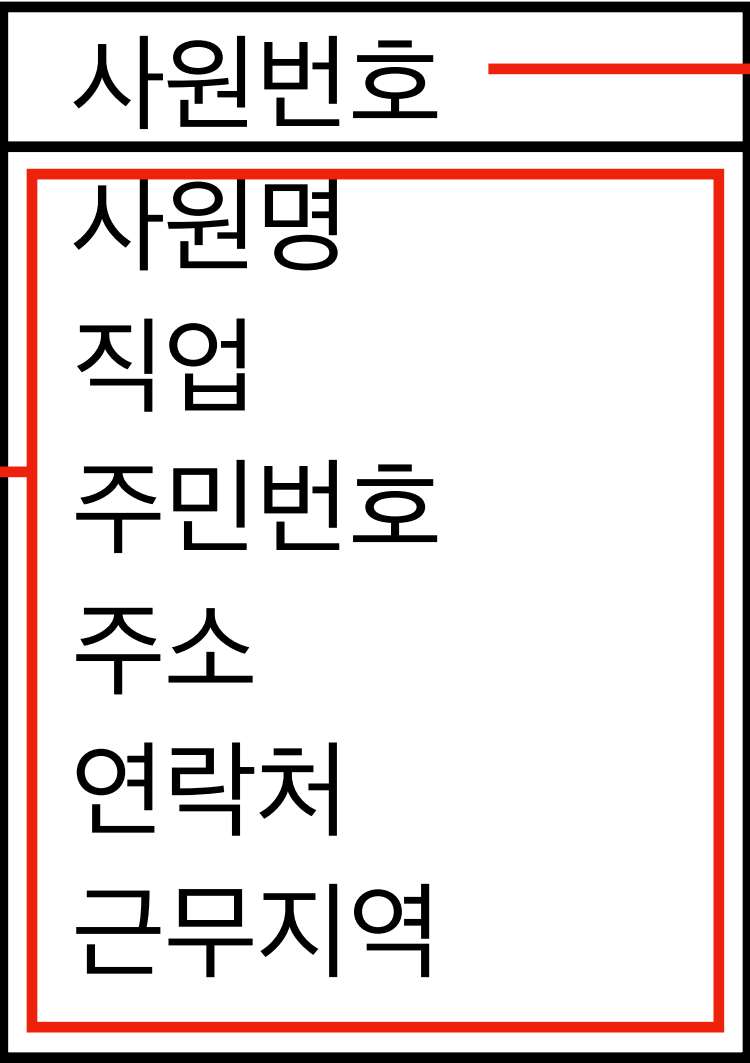
4 . ERD

i/e

Entity

사원

Attribute



primary key

4. ERD

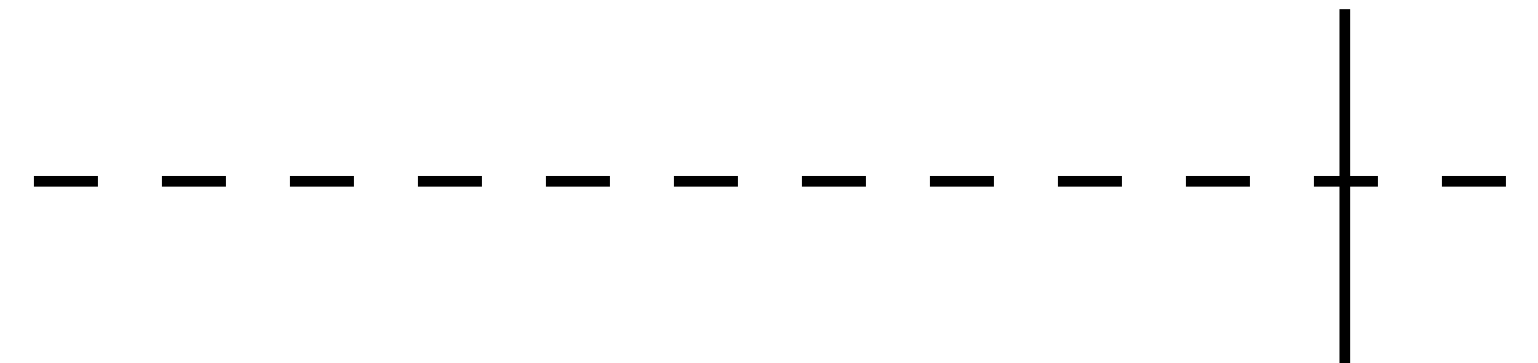
barker

A - B 관계

A 가 없으면 B 가 존재할 수 없다 (종속)



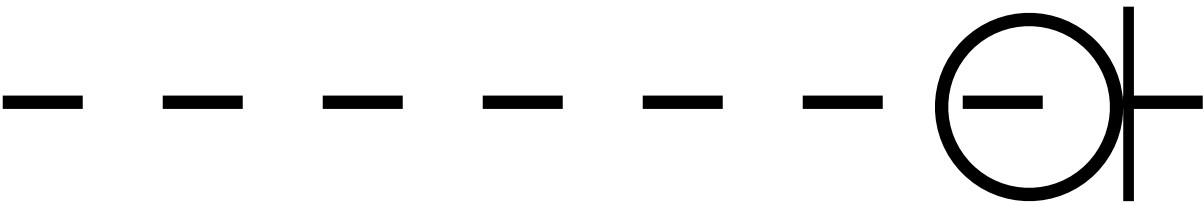
A 가 없어도 B 가 존재할 수 있다 (독립)



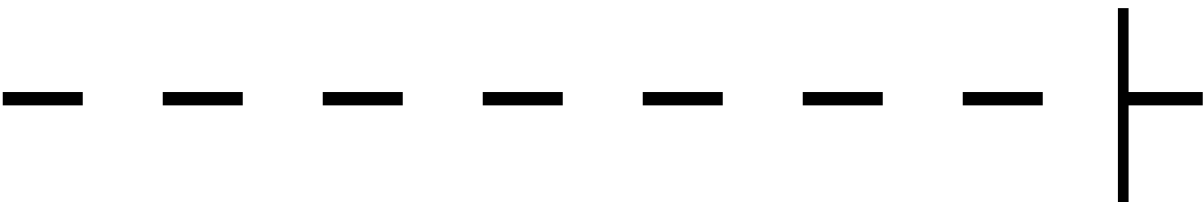
4. ERD

barker

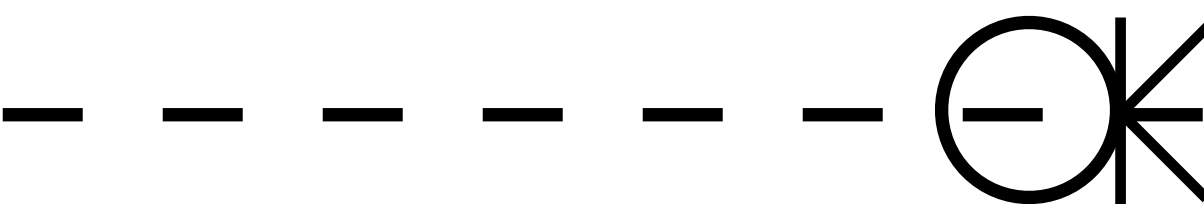
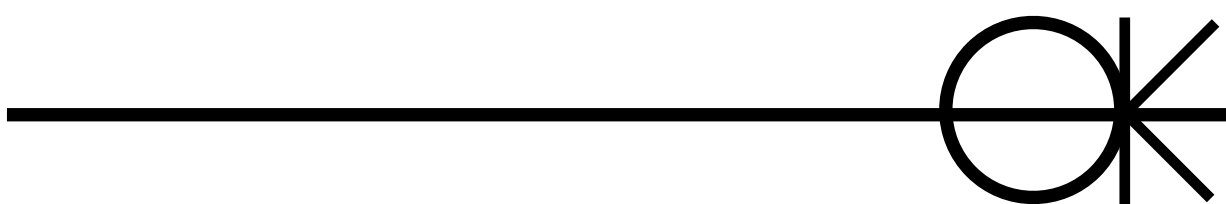
0 or 1



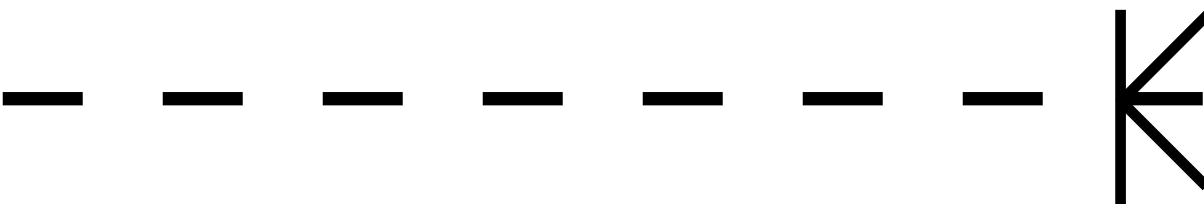
1



0 or more



1 or more



5.nosql

개념

고정되지 않은 스키마

데이터 간의 관계를 정의하지 않는 데이터베이스

분산형 구조 (대용량 데이터 저장 용이)

5.nosql

mongodb data structure

database

독립적인 하나의 권한

각각의 database는 분리된 파일로 저장

collection

document들의 group(rdbms - table)

schema를 가지지 않음(document들의 field가 각각 다름)

document

data record를 BSON(Binary JSON)으로 저장

field(key) 중복 불가

5.nosql

mongodb pipeline

collection이 각 stage를 거치면서 document 처리 및 집계

일부 처리는 shard에 대응 (각 shard에서 처리)

* pipeline
이전 단계의 결과를 다음 단계의 입력으로 사용

Collection

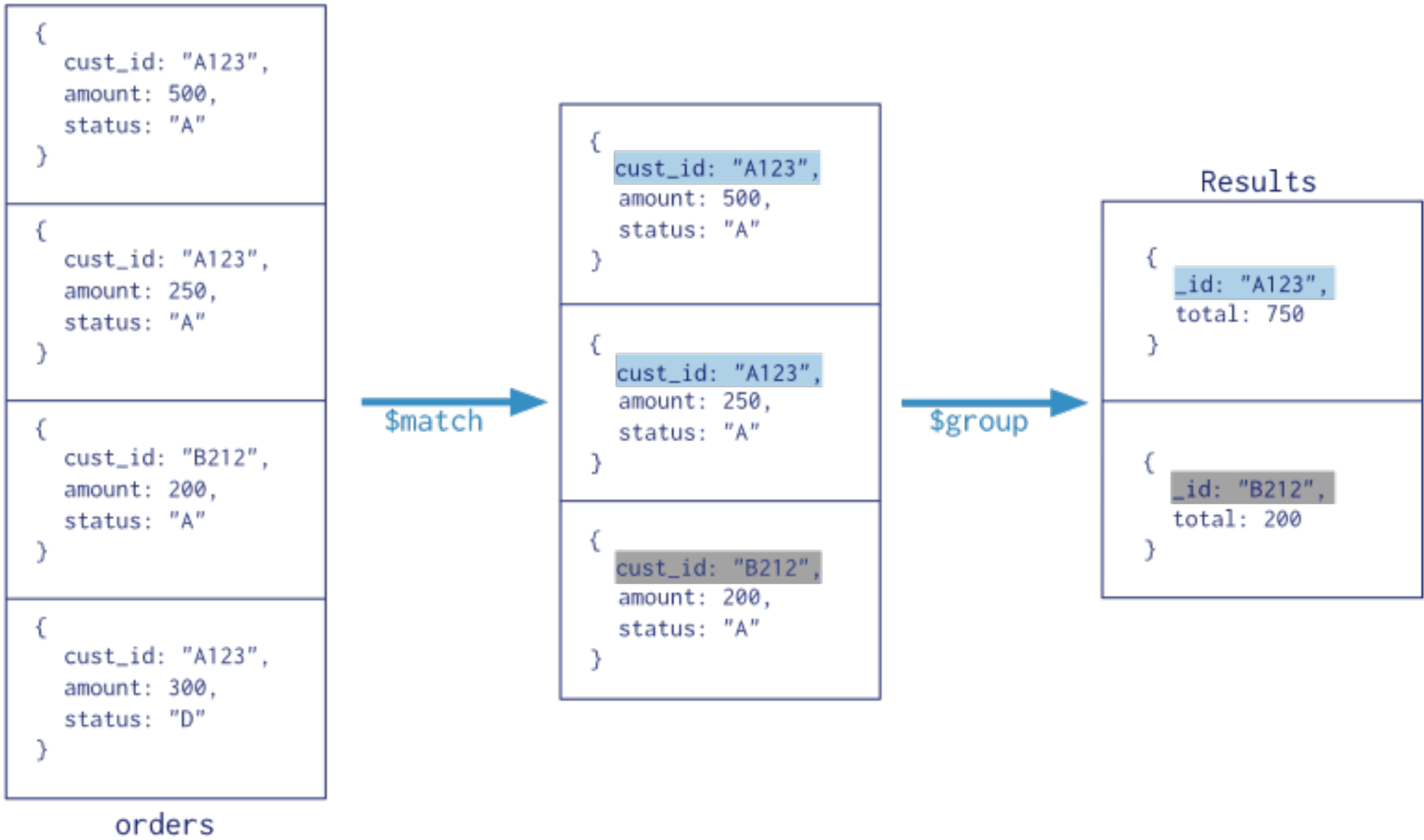
↓

db.orders.aggregate([

\$match stage → { \$match: { status: "A" } },

\$group stage → { \$group: { _id: "\$cust_id", total: { \$sum: "\$amount" } } }

])



5.nosql

redis 사용 예시

caching

session control

message queue

