

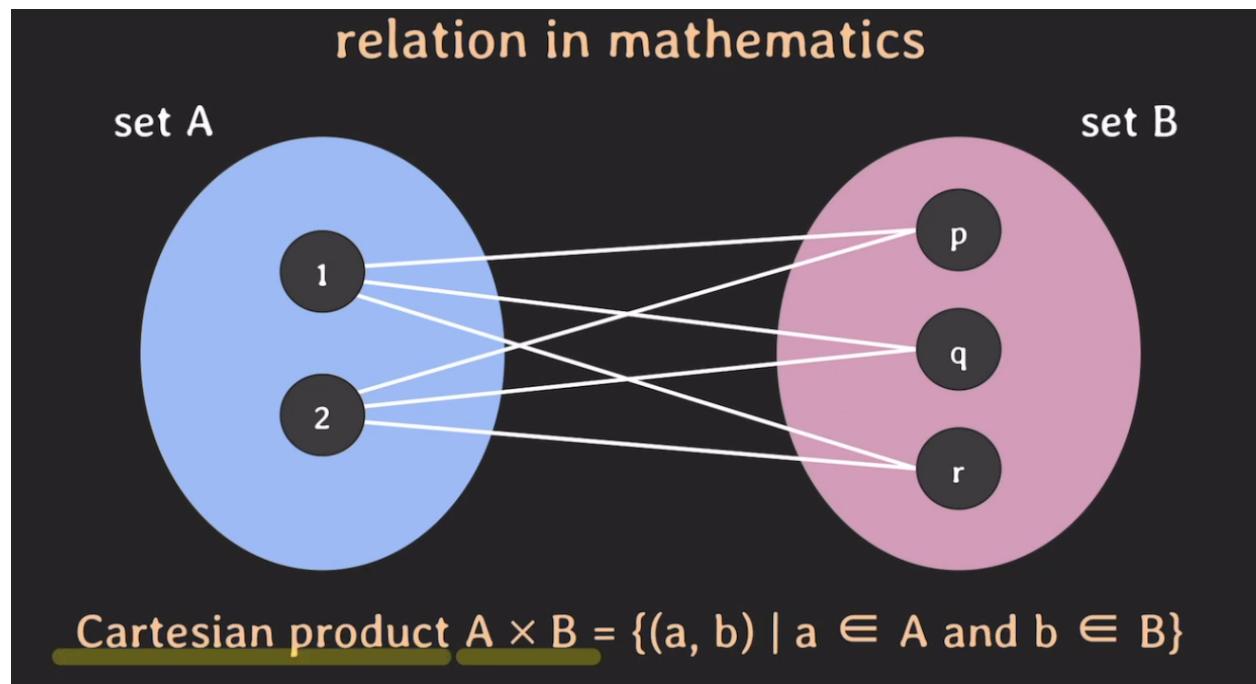
database-2

☰ 태그	
🕒 날짜	@2023년 6월 8일

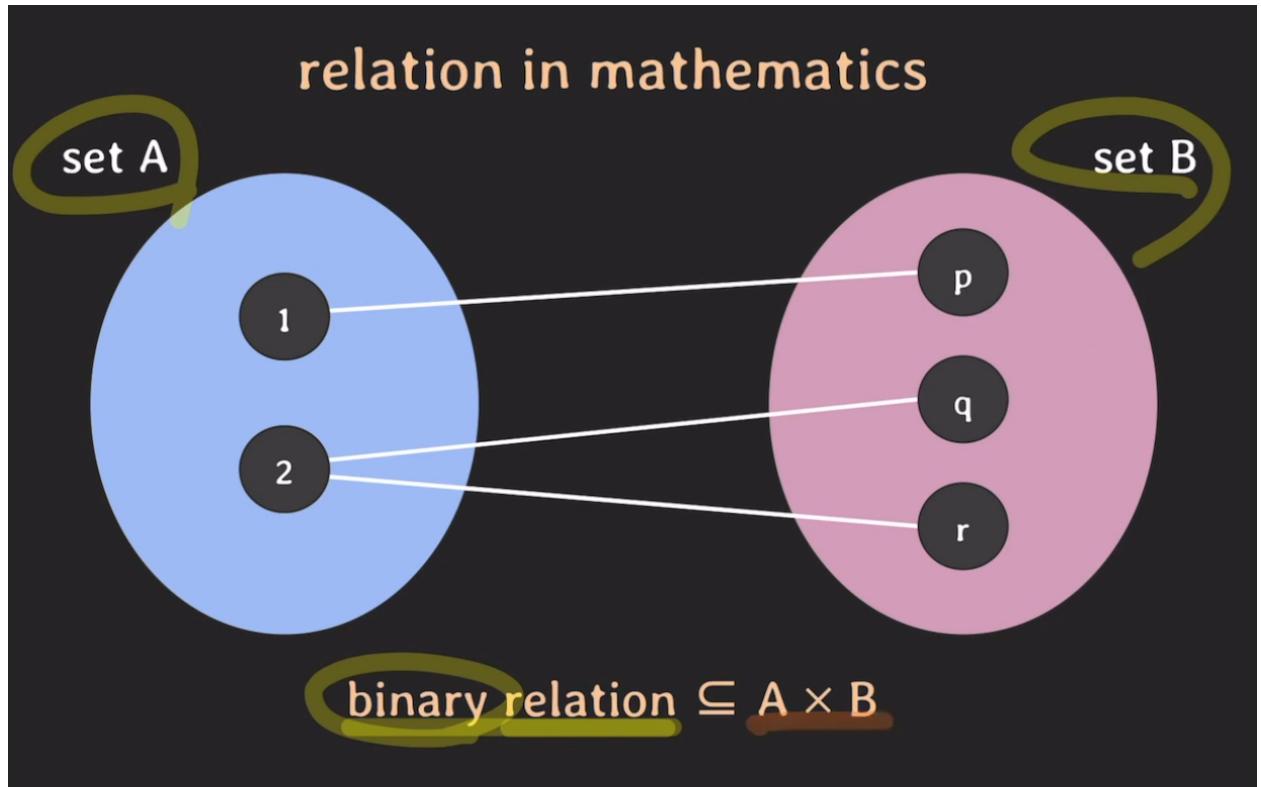
relational data model

set

- 서로 다른 elements를 가진 collection
- 하나의 set에서 elements의 순서는 중요하지 않다.

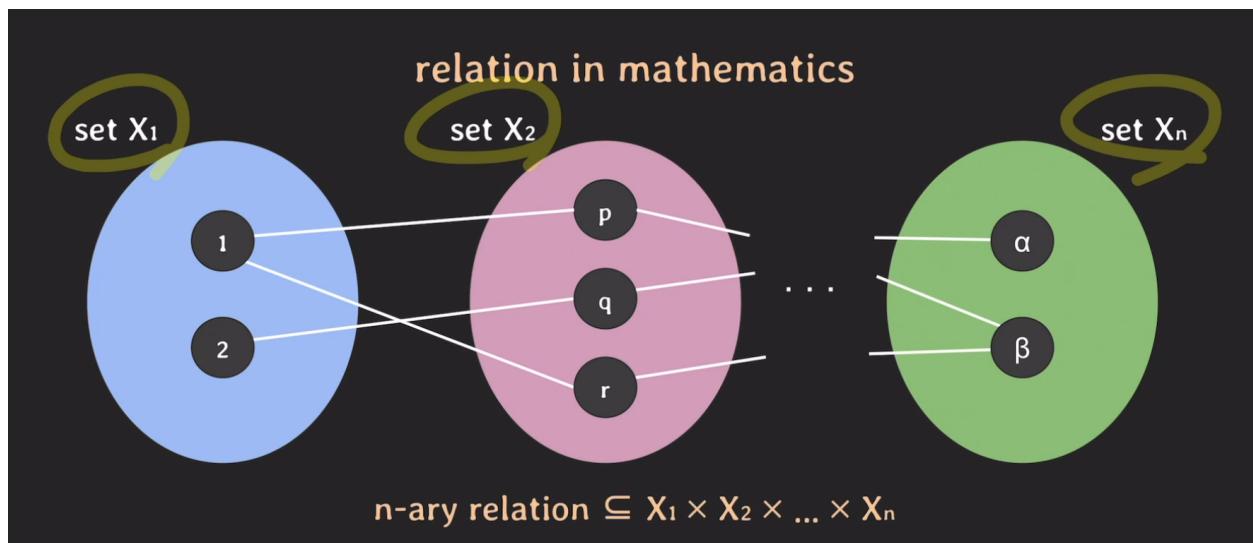


- set a에서 하나를 고르고 set b에서 하나를 골라 페어를 만든다
- 모든 페어의 조합을 수학식으로 나타낼 수 있음 cartesian product
- **Cartesian product** : 모든 페어의 조합을 의미한다



- 이런 경우 binary relation이라고 한다.
- a와 b의 cartesian product 의 부분집합

n개의 집합



- 총 3개의 list가 있고, 각각의 리스트를 tuple이라고 부를 수 있다.
- tuple : 몇개의 엘리먼트로 이루어진 리스트
- n - tuple

relation in mathematics

- subset of Cartesian product : 카르테지안 프로덕트의 부분 집합
- set of tuples : 튜플의 집합

set 은 도메인을 의미.

set 안에 element, 또는 value라고 한다. 이런 값들의 집합을 도메인이라고 한다.

relational data model

student relation을 예를 들어 relational data model을 이해해보자.

domain 정의하기

- students_ids : 학번 집합, 7자리 integer정수
- human_names : 사람 이름 집합, 문자열
- university_grades : 대학교 학년 집합, {1, 2, 3, 4}
- major_names : 전공 이름 집합
- phone_numbers : 핸드폰 번호 집합

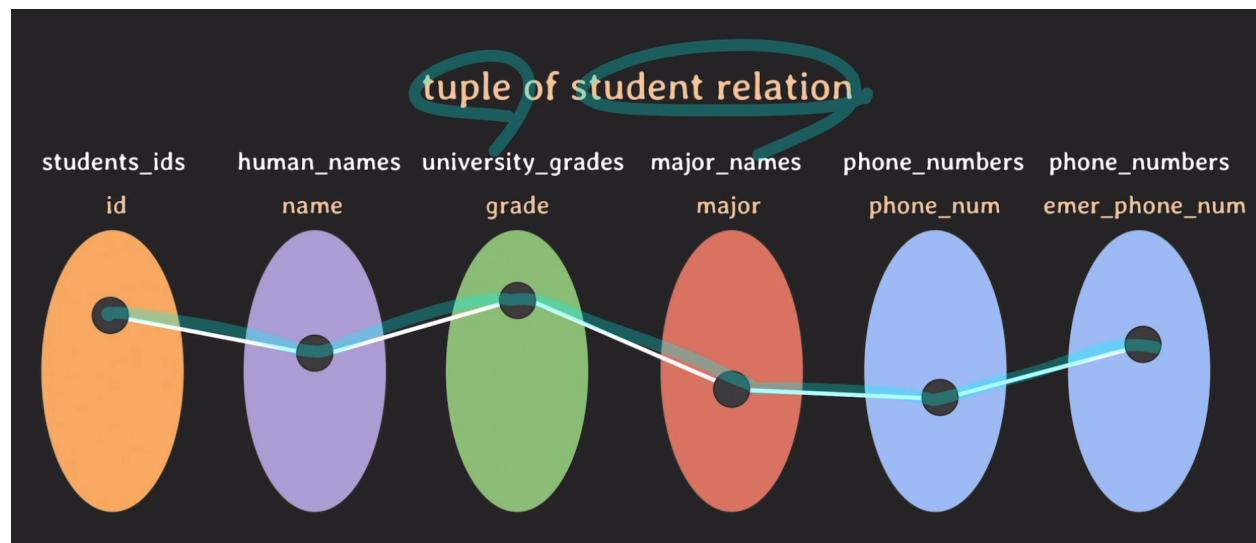
phone_numbers를 두개 만든 이유. : 학생의 번호와 비상 연락망

같은 도메인이 같은 relation에서 두번 사용 되는것 인데 그 역할이 다른 것.

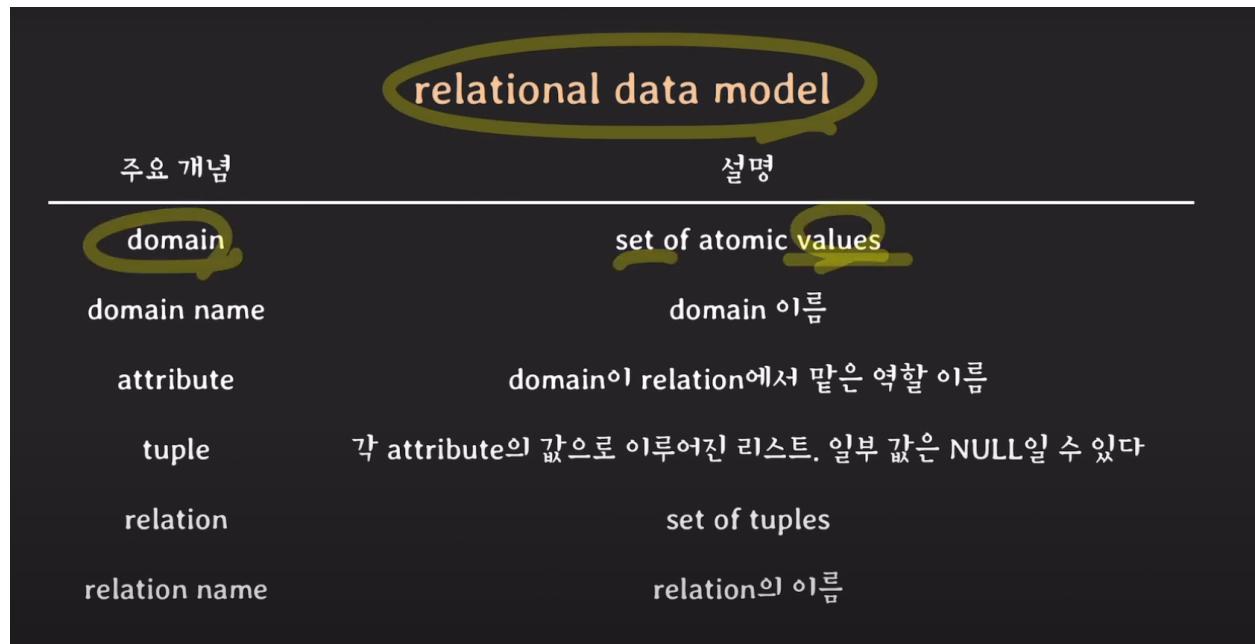
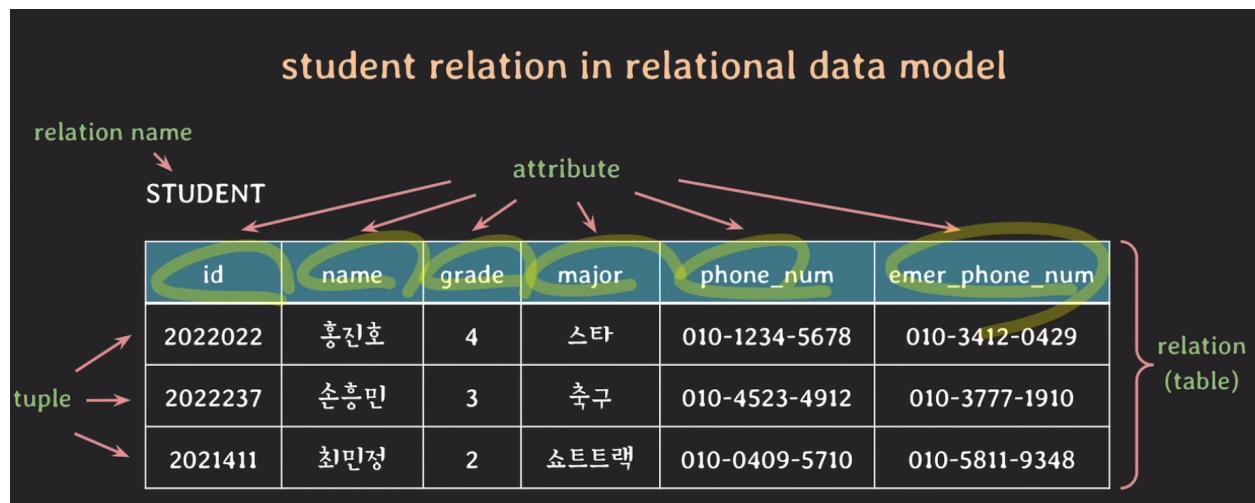
추가적으로 attribute 개념이 등장

attribute : 각각의 도메인 들이 어떤 역할을 수행 하는지 이름을 붙여 준다. id, name, grade, major, phone_num, emer_phone_num

각각의 도메인마다 값들이 있을 것이고 그 값들 중 실제로 데이터로서 존재하는 튜플들이 존재 할것이다.



그림으로 표현을 잘 하지 않는다. 이것을 가장 잘 표현할 수 있는 방법은 테이블이다.
그래서 릴레이션을 테이블이라고도 많이 부른다.



relation schema

- relation의 구조를 나타낸다.
- relation 이름과 attributes 리스트로 표기된다.
- STUDENT(id, name, grade, major, phone_num, emer_phone_num)
릴레이션의 이름(attribute name, attribute name, attribute name...)
- attribute와 관련된 constraints도 포함된다.

degree of a relation

- relation schema에서 attributes의 수
- STUDENT(id, name, grade, major, phone_num, emer_phone_num) → degree 6

relation (or relation state)

- set of tuples

STUDENT

id	name	grade	major	phone_num	emer_phone_num
2022022	홍진호	4	스탱	010-1234-5678	010-3412-0429
2022237	손홍민	3	축구	010-4523-4912	010-3777-1910
2021411	최민정	2	쇼트트랙	010-0409-5710	010-5811-9348

} relation
(relation state)

아까는 전체를 릴레이션이라고 표현했지만, 실제 데이터 실제 튜플들의 집합 또한 릴레이션이라고도 할 수 있다.

relational database

- relational data model에 기반하여 구조화된 database
- relational database는 여러 개의 relation으로 구성된다.

relational database schema : relational schemas set + integrity constraints set

relation의 특징들

- relation은 중복된 tuple을 가질 수 없다. (relation is set of tuples)

id	name	phone_num	grade	major
2022022	홍진호	010-0248-1632	4	스타
2022022	홍진호	010-0248-1632	4	스타
2021411	최민정	010-1015-3030	2	쇼트트랙

impossible

- relation의 개념 자체가 tuple의 집합이기 때문에 set은 중복을 허용하지 않는다.
- relation의 tuple을 식별하기 위해 attribute의 부분집합을 key로 설정한다.

id	name	phone_num	grade	major
2022022	홍진호	010-0248-1632	4	스타
2022237	손홍민	010-2022-2307	3	축구
2021411	최민정	010-1015-3030	2	쇼트트랙

- id를 통해 식별
- relation에서 tuple의 순서는 중요하지 않다.



id	name	phone_num	grade	major
2022022	홍진호	010-0248-1632	4	스타
2022237	손홍민	010-2022-2307	3	축구
2021411	최민정	010-1015-3030	2	쇼트트랙

- 하나의 relation에서 attribute의 이름은 중복되면 안된다.
- 하나의 tuple에서 attribute의 순서는 중요하지 않다.
- attribute는 atomic 해야 한다.(composite or multivalued attribute 허용 안됨)

id	name	address	grade	major
2007777	최고운	서울특별시 강남구 청담동	4	컴공, 디자인
...
...

atomic : 원자적인, 더 이상 나누어 질 수 없는

서울특별시 강남구 청담동은 시, 구, 동으로 쪼개질 수 있는 attribute이다.

이렇게 여럿개가 묶여 있는것을 **composite attribute**라고 한다. 이런 것들을 나누어 주어야 한다.

major 또한 복수 전공이기 때문에 컴공 과 디자인 이런경우를 **multivalued attribute**라고 한다. 이것 또한 잘라주어야 한다.

null의 의미

- 값이 존재하지 않는다.
- 값이 존재하나 아직 그 값이 무엇인지 알지 못한다.
- 해당 사항과 관련이 없다.

STUDENT

id	name	phone_number	grade	major	toeic_score	transfer_from
...	NULL	수운대학교
...	912	NULL
...	990	NULL

toeic_score 의 null은 한번도 시험을 치루지 않아서 점수가 없거나 점수를 업데이트 하지 않았거나

transfer_from 의 null은 편입하지 않았거나 해당사항이 없거나

superkey

- relation에서 tuples를 unique하게 식별할 수 있는 attributes set

PLAYER(id, name, team_id, back_number, birth_date) 의 superkey는 {id, name, team_id, back_number, birth_date}, {id, name}, {name, team_id, back_number},... 등등

candidate key

- 어느 한 attribute라도 제거하면 unique하게 tuples를 식별할 수 없는 super key
- key or minimal super key
- PLAYER(id, name, team_id, back_number, birth_date) 의 candidate key는 {id}, {team_id, back_number}

primary key

- relation에서 tuples를 unique하게 식별하기 위해 선택된 candidate key
- _ 언더바가 표시되어 있는 것이 PK
- PLAYER(id, name, team_id, back_number, birth_date) 의 primary key는 {id} or {team_id, back_number}
- 보통 attribute 숫자가 적은 것을 primarykey로 정한다.

unique key

- primary key가 아닌 candidate keys
- alternate key
- PLAYER(id, name, team_id, back_number, birth_date)에서 {id}가 primarykey가 선택되면
unique key는 {team_id, back_number}

foreign key

- 다른 relation의 PK(primarykey)를 참조하는 attribute set
- PLAYER(id, name, team_id, back_number, birth_date) 와 TEAM(id, name, manager)가 있을 때 foreign key는 PLAYER의 {team_id}
- team_id는 TEAM의 id를 참조한다.

constraints

- relational database의 relations들이 언제나 항상 지켜줘야 하는 제약 사항

1. implicit constraints

- relational data model 자체가 가지는 constraints
- relation은 중복되는 tuple을 가질 수 없다.
- relation 내에서는 같은 이름의 attribute를 가질 수 없다.

2. schema-based constraints

- 주로 DDL을 통해 schema에 직접 명시할 수 있는 constraints
- explicit constraints
- domain constraints

- attribute의 value는 해당 attribute의 domain에 속한 value여야 한다.
- grade의 100은 들어갈 수 없다. {1,2,3,4}

STUDENT				
<u>id</u>	name	phone_num	grade	major
2022022	홍진호	010-0248-1632	4	스타
2022237	손흥민	010-2022-2307	100	축구
2021411	최민정	010-1015-3030	2	쇼트트랙

- key constraints
 - 서로 다른 tuples는 같은 value의 key를 가질 수 없다.

STUDENT				
<u>id</u>	name	phone_num	grade	major
2022022	홍진호	010-0248-1632	4	스타
2022022	손흥민	010-2022-2307	3	축구
2021411	최민정	010-1015-3030	2	쇼트트랙

- null value constraint
 - attribute가 not null로 명시 됐다면 null을 값으로 가질 수 없다.

STUDENT				
<u>id</u>	name	phone_num	grade	major
2022022	홍진호	010-0248-1632	4	스타
2022237	손흥민	010-2022-2307	3	축구
2021411	최민정	NULL	2	쇼트트랙

- entity integrity constraint
 - primary key는 value에 null을 가질 수 없다.
 - primarykey는 tuple을 유니크하게 식별하기 위한 것인데 null이라면 더 이상 tuple을 식별할 수 없다.

STUDENT				
<u>id</u>	name	phone_num	grade	major
NULL	홍진호	010-0248-1632	4	스타
2022237	손흥민	010-2022-2307	3	축구
2021411	최민정	010-1015-3030	2	쇼트트랙

- referential integrity constraint
 - FK(foreign key)와 PK(primary key)와 도메인이 같아야 하고 PK에 없는 values를 FK가 값으로 가질 수 없다.

The diagram illustrates a database schema with two tables: **PLAYER** and **TEAM**. The **PLAYER** table has columns: **id**, **name**, **team_id**, **back_number**, and **birth_date**. The **team_id** column is highlighted with a blue oval. The **TEAM** table has columns: **id**, **name**, and **manger**. A specific row in the **TEAM** table, where **team_id** is **team_023**, is highlighted with an orange oval. A blue arrow points from the **team_id** column in the **PLAYER** table to the highlighted row in the **TEAM** table.

id	name	team_id	back_number	birth_date
...	...	team_017
...	...	NULL
...	...	team_023

id	name	manger
team_011
team_017
team_025

PLAYER 의 team_023이 TEAM에서 참조를 해야하는데 TEAM에 없으면 안된다.