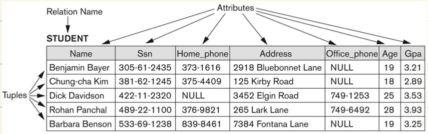
Relational Model Concepts

used for conceptual schema

Implementation data model

Relation	: mathematical concept
Informal definition	· relation ≈ table of values – contains set of rows (tuples) · relation attribute name = column header
key of relation	: uniquely identifies each row
artificial key	: row-ids or sequential numbers (surrogate key)
formal definition	· relation ≈ schema · $R(A_1, A_2, \cdots, A_n)$: relation R has $A_1, A_2, \cdots, A_n$ attributes = Table R은 $A_1, A_2, \cdots A_n$ attribute로 구성
domain	: set of valid values for attribute · data type, format, range 다양한 x
tuple	: ordered set of values · $\langle A_1, A_2, A_3 \rangle : \langle$"Jason", 26, 185$\rangle$ · $\langle A_1$: "Jason"$\rangle$, $\langle A_3 : 185 \rangle$, $\langle A_2 : 26 \rangle$ · n-tuple = tuple with n attributes · relation = set of tuples = Table


Relation Name
STUDENT
Attributes
Tuples

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

– $r(R)$: specific of relation R : Relation set (data)

– $r(R) = \{t_1, t_2, \cdots, t_n\}$ = set of tuples

– $t_i = \langle v_1, v_2, \cdots, v_n \rangle$ = n attributes

$R(A_1, \cdots, A_n)$

$r(R) = \{t_1, \cdots t_n\}$ ← 각 Table(Relation) R은 N개의 Attribute (tn) 구성

$t_i = \langle v_1, v_2, \cdots, v_n \rangle$

↖ 각 tuple은 attributes에 대한 vector임!

Characteristic of Relation

ordering of tuples	: not considered to be ordered
values in tuple	• all values are atomic (indivisible) ★ • each value should be form of domain of attribute • null value – unknown / not available / inapplicable – key attribute can't be null ★ key attribute ≠ null ★

Relation Constraints

Table

Constraint : determines which values are permissible

types

- Inherent / Implicit Constraints
- schema-based / Explicit Constraints
  - domain constraints : Attribute의 값범위를 rule
  - relational integrity constraints : Entity set 간에의 값범위를 rule
- Application-based / semantic constraints ··· application specific

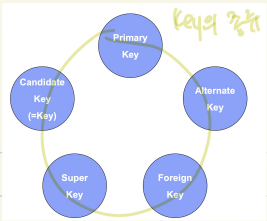
Relational Integrity Constraints : Constraints are conditions that must hold all valid relation states

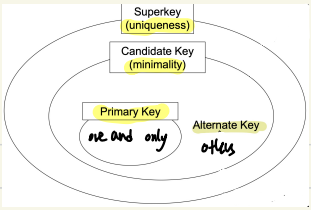
types

- key constraints
- Entity Integrity Constraints : Entity 무결성 (key ≠ NULL)
- referential Integrity Constraints : Reference 무결성 (참조 열? Partial? Total?)

key constraints

one of relational integrity constraints





Super key of R
outer most boundary

: set of attributes sk of R

- sk는 unique value임!
- $t_1[sk] \neq t_2[sk]$ … tuple 1의 sk value랑 tuple 2는 다를 것

Example

PName	Price	Category	Manufacturer
iPhone 13	1090000	Phone	Apple
iPad Pro	1269000	Tablet	Apple
Galaxy S22	999,900	Phone	Samsung
P11	319,000	Tablet	Lenovo

- PName : SK
- {PName, Price} : SK

key of R

: minimal super key ← candidate key

- key ⊂ super key
- **candidate key** … candidate for primary key

Example

- {PName, Price} is not key
- PName can be key … super key 중에서 제일 최소 단위들

Primary key of R

: key of key 중 represent relation 할

- **underlined** ★
- uniquely identify each tuple
- used as **reference** for another tuple/relation
- smallest one in candidate keys ★

Alternative key

: candidate key(key) − primary key

Foreign key

: foreign key in one table points to primary key in other table

- **referenced** primary key of other relation
- 항상 referential integrity 만족되어야 함

다른 relation의 primary key 가 해당.

Entity Integrity

각 entity가 가져야할 rule

- primary key of each relation R in database S **can't be null values**
  - $t[PK] \neq null$ for $\forall$ tuple t in $r(R)$

Referential Integrity

to specify relationship among two relations

referencing relation

: referencing relation R_1 has foreign key attributes

value

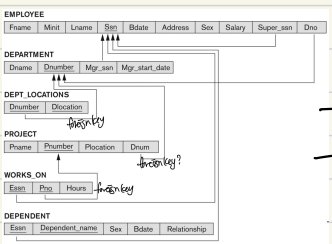
- same with pk of referenced relation
- null
  - ex) "Student with no phone" is okay

primary key는 Null X
Foreign key는 Null O

referenced relation

: has primary key

- $t_1[FK] = t_2[PK]$ ... $R_1 \xrightarrow{\text{reference}} R_2$



→ foreign key 는 보통 underlined 아님

→ 보통 primary key가 foreign key로 들어간다고 함

Relational database Schema

Relational DB schema

- set S of relation schemas that belongs to same database
  - S : name of whole DB schema
  - R_i : relation schema within database S
  - $S = \{R_1, R_2, \cdots, R_n\}$ ← relation schema
    - database S는 (S)
    - $R_1, R_2, \cdots, R_n$의 Relations(Tables)로 이루어져있다.

Relational DB state

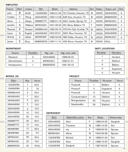
: current state of database

- $DB = \{r_1, r_2, \cdots, r_m\}$ ← Database state
  - r_i = relation state
  - $DB = \{r_1, r_2, \cdots, r_m\}$
- Invalid state : constraints 만족 안할 경우!

Populated DB state

↑ 전체 DB snapshot

: each relation with current relation state



★ upper case = Schema

lower case = state at specific time

= real data at specific time

Update Operations · Dealing with Constraint Violations

Operations	· Insert / Delete / Modify
Characteristics	· Integrity Constraints (key Integrity, Entity Integrity, Referential Integrity)를 Violate 하지 X
When Integrity Violation	① Reject / Restrict : Cancel operation that causes violation ② Perform ⊕ Inform : 일단 violate 한채로 실행하고, user에게 Inform ③ Cascade / Set null : trigger additional update to correct violation

Update Operations · Dealing with Constraint Violations (Cont.)

Insert	① domain constraint : attribute domain에 맞는 type 아닐경우 insert
	② key constraint : Already exist primary key Insert
	③ Referential Constraint : foreign key가 non-exist primary key일 때!
	④ Entity Constraint : Primary key가 NULL
DELETE	• Primary key delete - Restrict : reject deletion - Cascade : propagate into foreign keys of referencing tuples - Set NULL : set foreign key to NULL
UPDATE	: 두가지 update 타입의 경우 ① Update Primary key : Need to delete referential foreign key
	② update Foreign key : Violate referential integrity
	③ update ordinary attribute (Neither PK, FK) : Violate domain constraints