Relational Model

CS 6070 Databases

Kennesaw State University

KENNESAW STATE UNIVERSITY

Relational Data Model

A *relation schema* $R(A_a, ..., A_n)$ is a relation name R and a list of attributes $A_1, ..., A_n$.

Each attribute A_i is the name of a role played by some domain D.

- Example: $AUTHOR(author\_id, first\_name, last\_name)$
  - $dom(A_1)$ (or $dom(author\_id)$) is integer
  - The role played by an integer in A_1 is that of an identifier/key.

A *database schema* is a collection of relation schemas.

- Example: *PUBS* database has relation schemas *BOOK*, *AUTHOR*, and *PUB* (for publication, not public house)

Relations and Databases

A *relation*, or *relation state*, $r(R)$ is a **set** of tuples that conform to a *relation schema* R.

- Example: given $AUTHOR(author\_id, first\_name, last\_name)$, a particular $r(AUTHOR) =$

author_id	first_name	last_name
1	John	McCarthy
4	Claude	Shannon
5	Alan	Turing
6	Alonzo	Church

A *database* is a set of relations.

Tuples

A *tuple* is an **ordered list** of values.

- Example: $t_1 = <1$, 'John', 'McCarthy'$>$

Each value in the tuple is that tuple's value for the corresponding attribute of the relation schema.

Example: (these are equivalent notations):

- $t_1[\mathit{first\_name}] =$ 'John' (bracket notation)
- $t_1.\mathit{first\_name} =$ 'John' (object notation)
- $t_1[2] =$ `'John'` (positional notation)

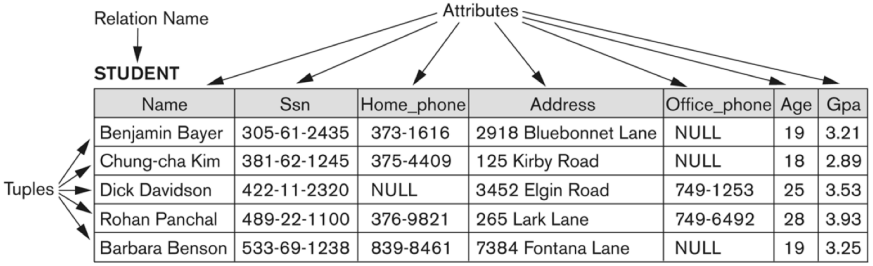
The *degree* or *arity* of a relation schema is the number of attributes it has.

- Example: *AUTHOR* has degree 3.

Attributes and Domains

Each attribute has a name and a *domain*

- The name describes the role played by the attribute
  - Example: the *first_name* attribute of the *AUTHOR* schema plays the role of the first name of an author represented by a tuple in a $r(AUTHOR)$ relation.
- The domain is a set of atomic values that a tuple may have for that attribute.
- A domain has a *logical definition*, e.g., integer or string, and may also have a *format*.
  - Example: `Home_phone` as $ddd - dddd$, where d is a digit



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

Figure 5.1
The attributes and tuples of a relation STUDENT.

KENNESAW STATE UNIVERSITY

Mathematical Definition of Relation

Given $R(A_1, ..., A_n)$,

- $r(R) \subseteq (dom(A_1) \times dom(A_2) \times ... \times dom(A_n))$

The total number of values, or *cardinality*, of a domain D is $|D|$.

So the maximum number of tuples that could possibly be in $r(R)$ is

- $|dom(A_1)| * |dom(A_2)| * ... * |dom(A_n)|$

Example: given

- $R(A_1, A_2)$
- $dom(A_1) = \{1, 2\}$, $dom(A_2) = \{a, b\}$

What are all the possible tuples that could appear in any $r(R)$?

Enumerating Tuples

Example: given

- $R(A_1, A_2)$
- $dom(A_1) = \{1, 2\}$, $dom(A_2) = \{a, b\}$

What are all the possible tuples that could appear in any $r(R)$?

$$dom(A_1) \times dom(A_2) = \{< 1, a >, < 1, b >, < 2, a >, < 2, b >\}$$

Given the definition of a *relation* or *relation state*, what is the maximum size of any $r(R)$?

Properties of Relations

- Atomicity of values, i.e., the First Normal Form assumption
  - Attribute values in tuples are indivisible, e.g., no compound or multivalued attributes as in EER models
- Nulls may appear in tuples for *some* attributes (more later)
  - Unknown, not applicable, not existing
- Closed world assumption
  - Facts not asserted explicitly are assumed to be false

Consider the properties above in the context of the following relation.

author_id	first_name	middle_name	last_name
1	John	NULL	McCarthy
4	Claude	Elwood	Shannon
5	Alan	Mathison	Turing
6	Alonzo	NULL	Church

Kinds of Constraints

- Inherent model-based (or *implicit*) constraints
  - domain constraints, atomic attribute values
- Schema-based (or *explicit*) contstraints
  - keys, referential integrity
- Application-based (or semantic constraints), a.k.a., business rules

Superkeys

A *superkey SK* is a set of attributes of a relation schema R such that

$$t_i[SK] \neq t_j[SK]$$

for any $i \neq j$.

In other words, the values of the superkey attributes of a tuple uniquely identify the tuple within the relation.

By the definition of the relational model, the full attribute set of a relation schema is a *default superkey*.

- Pause for a moment and make sure you understand that last statement.

Keys

A *minimal superkey* is a superkey for which removing an attribute would make it no longer a superkey.

We call a minimal superkey a *key*.

A relation schema may have several keys. We call these *candidate keys* and choose one arbitrarily to be the *primary key*.

We underline the primary key in a relation schema.

- Example: $AUTHOR(\underline{author\_id}, first\_name, last\_name)$

Database Integrity Constraints

- Domain constraints - Attribute values in tuples must be in domain for that attribute
- Key constraints - No two tuples can have the same values for the primary key
- Entity Integrity Constraints - No tuple can have a NULL value for its primary key attribute
- Referential Integrity Constraints - Tuples in one relation referencing tuples in another relation
- Semantic Integrity Constraints - Constraints on values of attributes that cannot be specified in the databases DDL

Referential Integrity Constraints

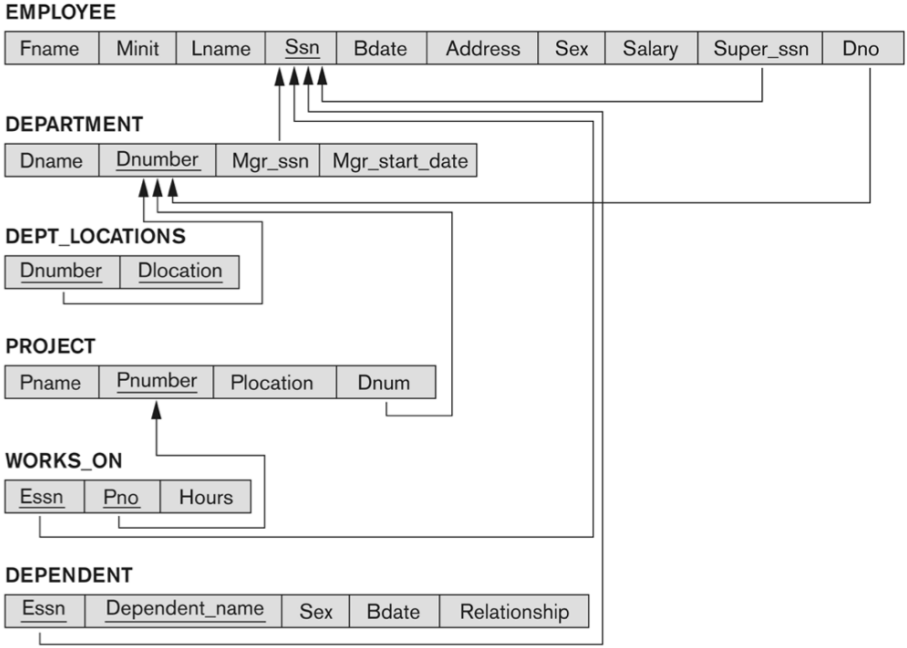
A foreign key value from a tuple in one relation must refer to nothing, or to the primary key for an existing tuple in another relation. Formally:

Given relation schemas R_1 and R_2, a set of attributes FK in R_1 is a foreign key referencing R_2 if

- the attributes in FK in R_1 have same domains as PK in R_2
- Given some t_1 in $r_1(R_1)$ and t_2 in $r_2(R_2)$, either $t_1[FK] = t_2[PK]$ or $t_1[FK]$ is NULL.

R_1 is the referencing relation, R_2 is the referenced relation.

Diagramming FK Relationships



Semantic Integrity Constraints

- Can't be specified in DDL
- Can be checked with triggers and assertions
- Usually checked in application code

Example: salary of an employee cannot exceed the salary of the employee's supervisor.

Constraint Violations on Insert

- Domain constraints
  - Insert a tuple with an attribute value not in attribute's domain
- Key constraints
  - Insert a tuple with a key that's already in the relation state
- Entity integrity constraints
  - Insert a tuple with a NULL value for any part of the primary key
- Referential integrity constraints
  - Insert a tuple in a referring relation whose FK does not appear as a PK value in any tuple of the referenced relation

Constraint Violations on Update

- Domain constraints
  - Update a tuple with an attribute value not in attribute's domain
- Key constraints
  - Update a tuple with a key value that already appears in another tuple in the relation
- Entity integrity constraints
  - Update a tuple with a NULL value for any part of the primary key
- Referential integrity constraints
  - Update a tuple in a refferring relation with a FK does not appear as a PK value in any tuple of the referenced relation
  - Update the primary key for a tuple in a referenced relation for which there are tuples in referring relationships. The tuples in referring relationships would be orphaned or end up referring to the wrong parent tuple.

Domain Integrity Violation Examples

author_id	first_name	last_name
1	John	McCarthy
4	Claude	Shannon
5	Alan	Turing
6	Alonzo	Church

$dom(author\_id) = \text{integer}$, $dom(first\_name) = \text{string}$, $dom(last\_name) = \text{string}$

- Insert `<"Two", "Jenny", "McCarthy">` – `"Two"` is not in $dom(author\_id)$
- Update `<1, "John", "McCarthy">` to `<1, "John", 1>` – `1` is not in $dom(last\_name)$

Key Integrity Violation Examples

author_id	first_name	last_name
1	John	McCarthy
4	Claude	Shannon
5	Alan	Turing
6	Alonzo	Church

- Insert `<1, "Jenny", "McCarthy">` – `1` is an existing primary key
- Update `<1, "John", "McCarthy">` to `<6, "John", "McCarthy">` – `6` is an existing primary key

KENNESAW STATE UNIVERSITY

Entity Integrity Violation Examples

author_id	first_name	last_name
1	John	McCarthy
4	Claude	Shannon
5	Alan	Turing
6	Alonzo	Church

- Insert <NULL, "Jenny", "McCarthy"> – NULL not allowed for primary key
- Update <NULL, "John", "McCarthy"> to <1, "John", 1>– NULL not allowed for primary key

KENNESAW STATE UNIVERSITY

Referential Integrity Violations – Employee - Department Example

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Constraint Violations

One last contraint violation on delete:

- Referential integrity: Delete a tuple in a referenced relationship for which there are tuples in referring relationships. The tuples in referring relationships would be orphaned.

Exercise:

1. For the database depicted in the previous slide, write down some reasonable constrints given the data shown.
2. In the context of the database in the previous slide and your constraints, which of the following are permissible? If an operation is not permissible, why is it not?

- Update `Salary` for `John Smith` to `"100K"`.
- Update `Bdate` for `Jennifer Wallace` to `February 29, 1980`.
- Update `SSN` for `Ahmad Jabbar` to `123456789`.
- Update `Dno` for `Alicia Zelaya` to `2`.
- Insert into `DEPT_LOCATIONS` the tuple `<3, "Marietta">`.
- Update `Super_ssn` for `James Borg` to `8675309`.
- Update `Super_ssn` for `John Smith` to `NULL`.
- Delete the `Research` depatment from the `DEPARTMENT` relation.

KENNESAW STATE UNIVERSITY

Closing Thoughts

- The relational model is a mathematical database model.
- Aside from its rigorous grounding, a strength of the relational model is its modeling (and, in DBMS systems, enforcement) of constraints.
- Relational data model is especially valuable if the integrity of your data is important.