CSc 134
Database Management Systems

3. Relational Data Model and Relational Database Constraints

Ying Jin
Computer Science Department
California state University, Sacramento

Relational Model Concepts

- Relational Model presents a database as a collection of relations.
  - Table :- Relation
  - Row :- Tuple
  - Column header :- attribute

Attribute

Tuple

Student

Name	**SSN**	**Home Phone**
Joe Smith	307-88-2907	602-7765543
Barbara Miller	590-38-6654	422-1076031

Relational Model
- Domain

- Domain: A domain D in the relational model is a set of atomic values.
  - Atomic: Each value in the domain is indivisible as far as the relational model is concerned.
- Domain:name, data type, format
- e.g. USA_Phone_numbers: A character string of the form (ddd)ddd-dddd, where each d is a numeric (decimal) digit and the first three digits form a valid telephone area code.
- e.g. employee_age: Possible ages of employee of a company; each must be an integer value between 15 and 80.

Relational Model
- Relation Schema

name of the relation

- A relation Schema R(A1,A2,…An) is made up of a relation name R and a list of attributes A1,A2,…,An
  - E.g. STUDENT(Name,ssn,phoneNumber)
- Domain of Ai is denoted by dom(Ai)

  Degree of a relation: number of attributes n of its relation schema

Relational Model
- relation state

- A relation (or relation state) of the relation schema R(A1,A2,…An) is denoted by r(R)
  - The relation is a set of of n-tuples r={t1,t2,..tm}, where each n-tuple t is an ordered list of values: t=<v1,v2,..vn>
  - each value vi, 1=<i<=n, is an element of dom(Ai) or is a special null value.

unknown or may not apply to a tuple

Relation name

Attributes

Tuples

STUDENT	Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
	Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
	Katherine Ashly	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
	Charles Cooper	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

Relational Model
- relation state (Cont.)

- $r(R) \subseteq (dom(A1) \times dom(A2) \times \ldots \times dom(An))$
- Tuples are unordered in a relation
- A relation cannot have duplicate tuples
- Denote cardinality (number of values) of domain D by |D|.
- Maximum number of tuples in r(R) is
  $|dom(A1)| * |dom(A2)| * \ldots * |dom(An)|$

Relational Model
- Attribute value

- Value v_i in tuple t for attribute A_i
  - t[Ai] or t.A_i
  - E.g. Given tuple t=<'Joe Smith', '307-88-2907', '602-7765543'>
    - t[Name]=<'Joe Smith'>
    - t.Name=<'Joe Smith'>
    - t[SSN,Name]=<'307-88-2907', 'Joe Smith'>
    - t.(SSn,Name)=<'307-88-2907', 'Joe Smith'>
- An attribute A of a relation R can be presented as R.A
  - STUDENT.Name

Constraints
- Category

- Constraints on databases can generally be divided into three main categories:
  - Inherent model-based constraints
    - constraints that are inherent in the data model
    - e.g.
      - Ordering of tuples in a relation
      - Relational model represents facts about both entities and relationship uniformly a relation
      - A relation cannot have duplicate tuples

Constraints
- Category (Cont.)

- Schema-based constraints
  - can be directly expressed in the schemas of the data model, typically by DDL.
- Application-based constraints
  - cannot be directly expressed in the schemas of the data model
  - must be expressed and enforced by application program.

- Another important category of constraints: Data Dependencies
  - functional dependencies and multivalued dependencies.

Schema-based constraints

- Constraints are *conditions* that must hold on *all* valid relation states.
- Domain constraints
- Key constraints
- Constraints on nulls
- Entity integrity constraints
- Referential integrity constraints

Domain constraints

- Within each tuple, the value of each attribute A must be an atomic value from the domain dom(A).
- Data type of domain
  - Integer
  - boolean
  - ...

Key constraints

- SK is a superkey of R, if for any two distinct tuples t1 and t2 in a relation state r of R, we have the constraint that

  t1[SK] ‡ t2[SK]
- Key constraint, Unique constraint
  - No two distinct tuples in any state r or R can have the same value for SK.
- e.g. {SSN, Name, Age}

Key

- A key is a minimal superkey – a superkey such that removal of any attribute from K results in a set of attributes that is not a superkey.
- e.g. {ssn}
- A relation schema may have more than one key, each of the keys is called a candidate key.
- e.g. fig

CAR	LicenseNumber	EngineSerialNumber	Make	Model	Year
	Texas ABC-739	A69352	Ford	Mustang	96
	Florida TVP-347	B43696	Oldsmobile	Cutlass	99
	New York MPO-22	X83554	Oldsmobile	Delta	95
	California 432-TFY	C43742	Mercedes	190-D	93
	California RSK-629	Y82935	Toyota	Camry	98
	Texas RSK-629	U028365	Jaguar	XJS	98

Primary Key

- Designate one of the candidate keys as the primary key of the relation.
- The choice of primary key from candidate keys is arbitrary
- It is better to choose a primary key with *a single attribute* or a *small number* of attributes.
- The primary key attributes are *underlined*.

Constraints on NULL values

- A constraint specifies that null values are or are not permitted
- e.g. employee Name is constrained to be NOT NULL.

Entity integrity constraint

- Entity integrity constraint: No primary key value can be null
- Because the primary key value is used to identify individual tuples in a relation.
- Involve a *single* relation

Referential integrity constraints

- Specify a *relationship* among tuples in two relations: the **referencing relation** and the **referenced relation**.
- Informally:
  - refer to an existing tuple

Foreign Key

- A set of attributes FK in relation schema R1 is a foreign key of R1 that references relation R2 if it satisfies two rules:

1. The attributes in FK have the same domain(s) as the primary key attributes PK of R2
2. A value of FK in a tuple t1 of the current state r1(R1) either occurs as a value of PK for some tuple t2 in the current state r2(R2), or is NULL.

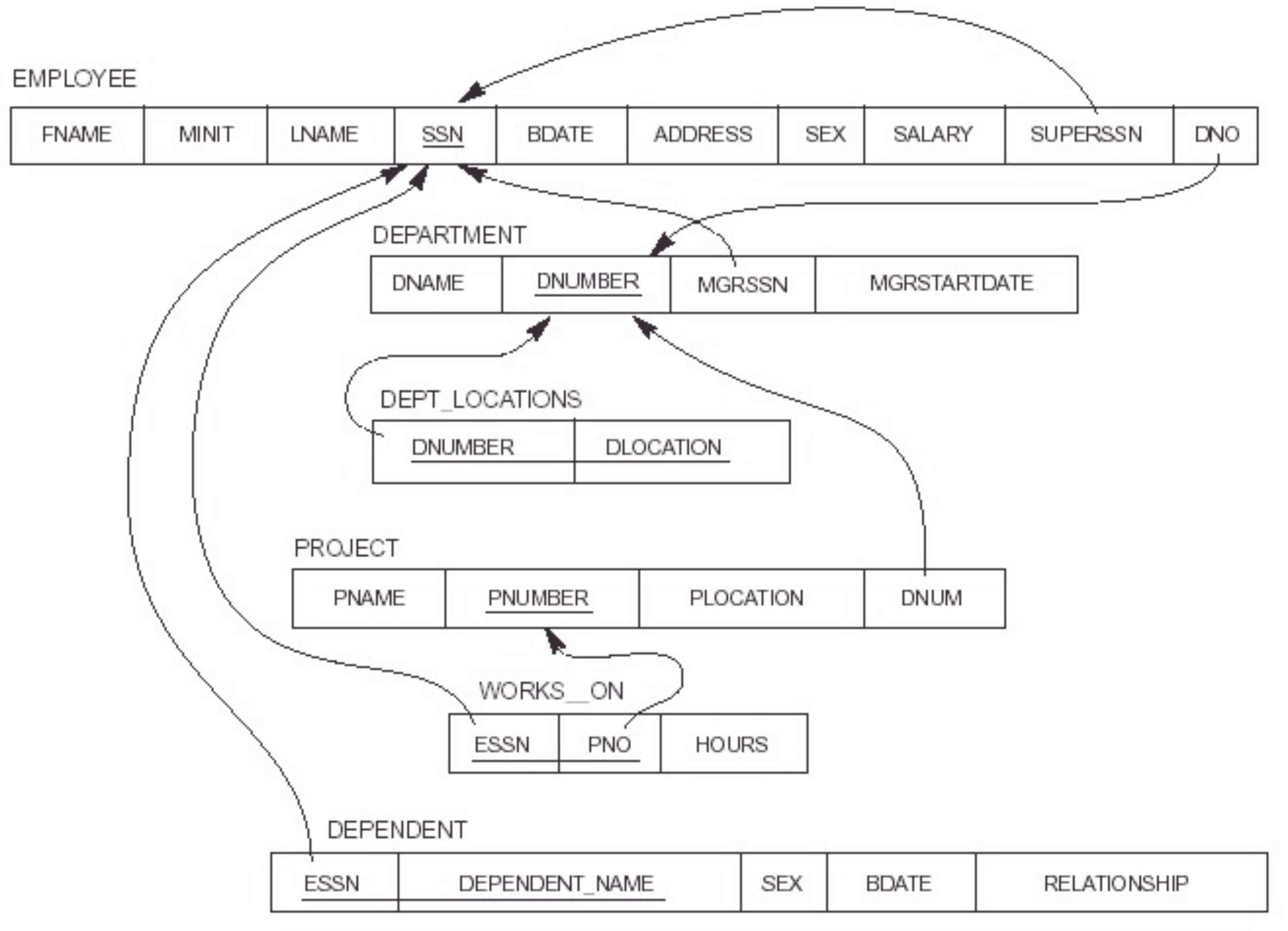
t1[FK] = t2[PK].

t1 references or refers to t2.

Referential integrity constraint definition

- If the two conditions hold, the referencing integrity constraint from R1 to R2 is said to hold.
- A referential integrity constraint can be displayed in a relational database schema as a directed arc from R_1.FK to R_2.

Referential integrity constraint example



dno=1?

dno=null?

Refer to its own relation

- A foreign key can refer to its own relation.
- e.g. superssn

Application-based constraints

- Semantic integrity constraints
- e.g. "The salary of an employee should not exceed the salary of the employee's supervisor"
- Constraint specification language
  - e.g. trigger, assertions
- Check within application programs

Relational Database Schemas

- A relational database schema S is a set of relation schemas

  S = {R1, R2,..., Rm}

  and a set of integrity constraints IC.
- A relational database state DB of S is a set of relation states

  DB = {r1,r2,..., rm}

  such that each ri is a state of Ri and ri satisfy the IC.

Example of relational database schema

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

One possible database state for the company schema

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John		Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin		Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia		Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer		Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh		Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce		English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad		Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James		Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPT_LOCATIONS	DNUMBER	DLOCATION
		Houston
		Stafford
		Bellaire
		Sugarland

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Valid /invalid state

- When we refer to a relational database, we implicitly include its *schema* and its *current state*.
- A database state satisfies all the constraints in IC is called a valid state.
- A database state does not obey all the integrity constraints is called an invalid state.

Update Operations on Relations

- INSERT a tuple.
- DELETE a tuple.
- MODIFY a tuple.

- Integrity constraints should not be violated by the update operations.

- Updates may *propagate* to cause other updates automatically. This may be necessary to maintain integrity constraints.

Update Operations on Relations (Cont.)

- In case of integrity violation, several actions can be taken:
  - Cancel the operation that causes the violation (REJECT option)
  - Perform the operation but inform the user of the violation
  - Trigger additional updates so the violation is corrected (CASCADE option, SET NULL option)
  - Execute a user-specified error-correction routine

Constraint violation
- Insert

- Insert can violate _______
  - Domain constraints
  - Key constraints
  - Constraints on null values
  - Entity integrity constraints
  - Referential integrity constraints
- Reject the insertion in case of constraint violation

Constraint violation
- Delete

- Can violate referential integrity
- In case of violation
  - Reject the deletion
  - Attempt to cascade the deletion
  - Modify the referencing attribute values the cause the violation
    - Set to null
      - foreign key is part of the primary key.
    - Change to reference another valid tuple
  - Specify it in DDL

Constraint violation
- Update

- Modify neither a primary key nor a foreign key
  - Check new value in the correct domain
- Update a primary key or a foreign key
  - Delete + Insert
  - Can use DDL to specify how to handle update

These slides are based on the textbook:

R. Elmaseri and S. Navathe, *Fundamentals of Database Systems*, 7th Edition, Addison-Wesley.