DATABASE SYSTEMS

Dr. Raj Sunderraman

Department of Mathematics and Computer Science Georgia State University

January 29, 1999

Ch 6. Relational Data Model

```
DOMAIN: D is a set of atomic values
```

e.g. USA_Phone_Numbers: set of 10-digit phone numbers valid in USA

Valid_Age: 0 to 150

Grade_Point_Averages: real numbers between 0 and 4

RELATION SCHEMA: R, denoted by R(A1,...,An), consists of a relation name R and list of ATTRIBUTES A1, ..., An n = DEGREE of relation

Each attribute is associated with domain, dom(Ai)

e.g. STUDENT(Name, SSN, HomePhone, Address, OfficePhone, Age, GPA)

is a relation schema of degree = 7

also denoted as r(R) is a finite subset of A RELATION (or RELATION INSTANCE) r on relation schema R(A1,...,An) dom(A1) X ... X dom(An)

Each element of a relation, such as (a1, ..., an) is called a TUPLE

A RELATIONAL DATABASE SCHEMA consists of

മ set of relation schemas R1,...,Rm

A RELATIONAL DATABASE INSTANCE is a set of relation instances satisfying the constraints specified in IC. DB = {r1,...,rn} such that each ri in an instance of Ri a set of Integrity constraints, IC (to be discussed later)

Characteristics of Relations:

- Ordering of tuples in a relation: No order among the tuples since a relation is a set of tuples. (Figures 6.1, 6.2)
- Ordering of values within a tuple: two perspectives (ordered, Skip unordered definition on Page 141. unordered). We shall go with the ordered perspective.
- Atomic values in tuples: First normal form. special value in each domain called NULL.
- A relation can be interpreted as a collection of assertions ENTITIES and others correspond to RELATIONSHIPS or facts about the mini-world. Some facts correspond to

NOTATION:

```
- t[Au, Aw, ..., Az] refers to <vu,vw,...,vz> Upper-case letters Q, R, S, refer to relation names in schemas
                                     Lower-case letters t, u, v refer to tuples
                                                                                                                                                                                                                                                                 n-tuple t in r(R) is <v1,...,vn>, where vi is in dom(Ai)
                                                                                                                                                                                                                                                                                                                       relation schema: R(A1,...,An)
We use R in \mathbb{R}(A1,\ldots,An) to refer to the relation instance
                                                                                 Lower-case letters q, r, s refer to relation instances
                                                                                                                                                                                                                         - t[Ai] refers to vi
```

6.2 Constraints

1) Domain Constraints:

- atomic domains
- data types such as integer, reals, strings, etc.

(2) Key Constraints:

- Super Key: A subset, K, of attributes of R is a super key for R for any two distinct tuples t1 and t2 in a (legal) relation instance of r of R, $t1[K] \iff t2[K]$.
- Candidate key: (or simply a key) is a super key such that its proper subsets is a super key. none 0f
- Primary Key: One of the candidate keys chosen by the database designer

DATABASE SCHEMA (with primary keys) and INSTANCE in Figures 6.5, 6.6

(3) Entity Integrity Constraint: No primary key attributes can have a NULL value.

(4) Referential Integrity Constraint (between relations)

FOREIGN KEY of R1 if set of attributes FK, of a relation schema R1 is a

(a) The attributes in FK have the same domain as the primary FK references or refers to the relation R2. key attributes PK of another relation schema R2

and (b) A value of FK in a tuple t1 of R1 either occurs of PK for some tuple t2 of R2 or is NULL. The tuple t1 references or refers to tuple t2. as a value

Figure 6.7

6.3 Update Operations

Insert, Delete, Modify

i.e. all constraints are statisfied. Database state is assumed to be consistent before updates,

database state (otherwise they should be rejected). Update operations, when completed, must result in a consistent

Constraints we will consider are:

Domain constraints
Key constraints,
Entity integrity constraint
Referential integrity constraints

Insert: (all 4 types of constraints may be violated).

Assume database of Figure 6.6

Examples of inserts:

- (1) Insert <'Cecilia','F','Kolonsky','677678989','05-APR-50', satisfies all constraints; ACCEPT IT into EMPLOYEE '6357 windy Lane, Katy, TX', F, 28000, null, 4>
- (2) Insert <'Alicia','J','Zelaya','999887777','05-APR-50' violates key constraint into EMPLOYEE '6357 windy Lane, Katy, TX', F, 28000, '987654321', 4>

```
(3) Insert <'Cecilia', 'F', 'Kolonsky', null, '05-APR-50',
violates entity integrity constraint;
                                         into EMPLOYEE
                                                                        '6357 windy Lane, Katy, TX',F,28000,null,4>
```

(4) Insert <'Cecilia','F','Kolonsky',677678989,'05-APR-50', violates referential integrity (no dept = 7!!) into EMPLOYEE '6357 Windswept, Katy, TX',F,28000,'987654321',7>

Two options if inserts are unacceptable:

- (1) Reject the insert
- (2) Try to correct the reason for rejection e.g. prompt user for value of key when null was specified or ask user to add dept = 7 then add the employee

Delete: can violate only referential integrity.

- (1) Delete WORKS_ON where ESSN = 999887777 and PNO = 10 ACCEPTABLE
- (2) Delete EMPLOYEE where SSN = '999887777' not ACCEPTABLE because two rows in WORKS_ON refer to this tuple
- (3) Delete EMPLOYEE where SSN = '333445555' not acceptable because it is referenced by tuples in EMPLOYEE, DEPARTMENT, WORKS_ON, DEPENDENT!!

Dr. Raj Sunderraman

12

3 options to deal with unacceptable deletes:

- (1) Reject it
- (2) Attempt to cascade deletes by deleting tuples that reference the two referencing tuples from WORKS_ON the tuple that is being deleted. e.g. in example (2), delete
- (3) Modify the referencing tuples (set to null or some other value); then delete the tuple

Combinations are also possible.

Modify: Can violate all constraints.

- (1) Modify Salary for employee with ssn = '999887777' to 28000 Acceptable
- (2) Modify DNO of Employee with ssn = '999887777' to 1 Acceptable
- (3) Modify DNO of Employee with ssn = '999887777' to 7 not Acceptable; violates referential integrity (not dept=7)
- (4) Modify ssn of employee with ssn='999887777' to '987654321' not acceptable; violates primary key and referential integrity

Modifying = Delete followed by Insert; Issues discussed earlier Modifying non-primary/non-foreign key usually do not create problems apply to modify.

Relational Algebra

Set-theoretic operations:

attributes and the domains of the corresponding attributes in the two relations are the same Two relations are union-compatible if they have the same number of

Consider two relations r(R) and s(S) that are union-compatible (normally ${\sf R}={\sf S}$).

Union: $r \cup s = \{t | t \in r \text{ or } t \in s\}.$

Difference: $r - s = \{t | t \in r \text{ and } t \notin s\}$

Intersection: $r \cap s = \{t | t \in r \text{ and } t \in s\}$

Cartesian Product: r(R) and s(S) on any schemes R and S.

$$r \times s = \{t_1.t_2 | t_1 \in r \text{ and } t_2 \in s\},$$

where, t1.t2 is the concatenation of tuples t_1 and t_2 to form a larger tuple.

Example: set operations

q	а	а	A
d	С	d	α

A B

о в В

q	а	A
р	q	K

а

Ф

ھ

 \circ

۵

മ

р

Ф

ىھ

 \circ

р

Q

ىھ

ര

ര

 $\begin{array}{c|c} r \\ \hline A \\ \hline C \\ \hline \end{array}$

മ	a	а	r.A	
С	Ь	q	r.B	r >
മ	а	В	A.s	$\times s$
C	е	С	s.B	

Relation-theoretic operations

Consider r(R) and s(S), two relations, where $R=(A_1,...,A_n)$ and $S=(B_1,...,B_m)$

Rename: $r(C_1,...,C_n) = \{t | t \in r\}$ with schema $(C_1,...,C_n)$.

Select: $\sigma_F(r) = \{t | t \in r \text{ and } t \text{ satisfies } F\}.$

examples how F is constructed) where F is a selection criteria involving constants and attributes of r. (will discuss in

 $\begin{aligned} \mathbf{Project:} \qquad & \pi_{D_1,...,D_p}(r) = \{t[D_1,...,D_p] | t \in r\} \\ & \text{where } D_i \text{ is one of } A_1,...,A_n. \end{aligned}$

theta-Join: $r \bowtie_F s = \{t | (\exists u \in r) (\exists v \in s) (t = u.v \text{ and } F \text{ is satisfied by } u \text{ and } v) \}$ where F is a conjunction of formulas relating attributes of r with attributes of s. (will discuss in examples how F is constructed)

Natural Join: $r \bowtie s = \{t | (\exists u \in r) (\exists v \in s) (t[R] = u \text{ and } t[S] = v)\}$

Division: Assume $B_1,...,B_m\subset A_1,...,A_n$.

$$r \div s = \{t | (\forall u \in s)(t.u \in r)\}$$

Examples: relation-theoretic operations

7.

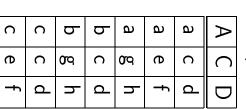
С	С	Ь	Ь	а	В	а	A
е	С	8	С	8	е	0	0
f	d	h	d	h	f	р	D

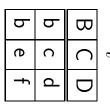
$$\sigma_{A='b'}$$
 or $c='c'(r)$

<u> </u>	b	b	а	А
<u>ر</u>	g	0	0	(
2	h	р	р	D

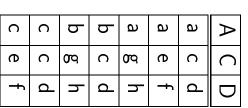
م

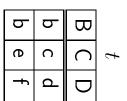
$$\pi_A(r)$$

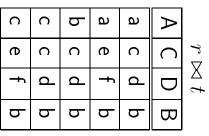


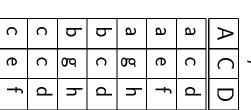


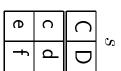
q	b	q	d	A	
ğ	g	С	С	C	7.
٩	h	р	р	D	$\bowtie_{r.A}$
d	Ь	d	q	В	
е	С	е	С	0	$=t.B \ t$
f	р	f	р	D	

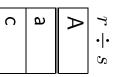












Basic Relational Algebra Operations

- Basic set: union, difference, Cartesian product, rename, select, and project.
- none of them can be expressed in terms of the others.
- Intersection, theta-join, natural join, and division can be expressed in terms of the basic operators as follows:

Intersection: $r \cap s = r - (r - s)$ theta Join: $r \bowtie_F s = \sigma_F(r \times s)$

Natural Join: $r \bowtie s = \pi_{R \cap S}(\sigma_F(r \times s))$

the common attributes of r and s are equal. where F is a selection condition which indicates that the tuple values under

Division: $r \div s = \pi_{R-S}(r) - \pi_{R-S}((\pi_{R-S}(r) \times s) - r)$

equalities for simplicity. Even though relation schemes are defined as sequences, they are treated as sets in these

An explanation for the equality for division is in order!

ullet First, all candidate tuples for the result are calculated by the expression

$$\pi_{R-S}(r)$$

 \bullet Next, these candidate tuples are combined with all tuples of s in the following expression

$$\pi_{R-S}(r) \times s$$

to give a relation containing all combinations of candidate tuples with all tuples of s.

• Since we are looking for tuples under the scheme R-S which combine with all tuples of s and are also present in r, if we subtract r from the previous expression, we will get all the combinations of tuples that are "missing" in r.

$$(\pi_{R-S}(r) \times s) - r$$

 \bullet By projecting these tuples on R-S, we get all those tuples that should not go to the result in the following expression.

$$\pi_{R-S}((\pi_{R-S}(r)\times s)-r)$$

• Finally, we subtract this set from the set of all candidate tuples and obtain the output relation of the division operator.

$$r \div s = \pi_{R-S}(r) - \pi_{R-S}((\pi_{R-S}(r) \times s) - r)$$