DBMS

UNIT-2

Relational Model

feedback/corrections: vibha@pesu.pes.edu

VIBHA MASTI

Relation

- Mathematical concept based on concept of sets
- E F Codd, "A Relational Model of Data for Large Shared Data Banks", 1970

https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf

Flat File

- Any file stored on local file system
- Stored sequentially without metadata (indexes, keys, relationships)
- Common formats: XML, CSV
- Disadvantage: need to know delimiter

CSV

```
Mr,Smith,Ottawa,ON
Mrs,Jones,Winnipeg,MB
```

Relation

- Table of values
- Set of rows (entities), called tuples
- Column headers called attributes or fields
- Domain: type of field

Relation Name

STUDENT

Attributes

Tuples

Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1
The attributes and tuples of a relation STUDENT.

DOMAIN

- Datatype or format defined for it
- Eg: phone numbers (+91 ddddd ddddd), dates (dd-mm-yyyy)

Schema

- Description of a Relation R with attributes $A_1, A_2, \ldots, A_n$
- Represented as $R(A_1, A_2, \ldots, A_n)$
- Each attribute has a domain; no. of fields = degree
- Eg: CUSTOMER(CustId, CustName, Addr, PhoneNo
- Relational schema used to describe relation

```
STUDENT(Name: string, Ssn: string, Home_phone: string, Address: string,
Office_phone: string, Age: integer, Gpa: real)
```

Relation State

- Set of n-tuples $r = \{t_1, t_2, \ldots, t_m\}$; snapshot of table at a given time of m rows and n columns
- Relation extension of Relation R = r(R)
- Relation intension = schema R

Tuple

- Ordered set of values enclosed in angle brackets < >
- Each value derived from its appropriate domain
- Eg: customer relation with 4 fields

  <62341, "Brent", "#1 ABC Street", "+91 86212 12121">
- Relation: set of valid tuples (set ⇒ unique rows)

Key of Relation

- Unique identifying value of a data item
- A column/field that uniquely identifies every row of a relation
- Mandatory for every relation
- Eg: SRN for student
- Artificial/surrogate keys: row # artificially generated

Cartesian Product & State

- Given $R(A_1, A_2, \ldots, A_n)$,

$$r(R) \subset dom(A_1) \times dom(A_2) \times \ldots \times dom(A_n)$$

- Eg: $R(A_1, A_2)$

  Let $dom(A_1) = \{0, 1\}$
  Let $dom(A_2) = \{a, b, c\}$

  $dom(A_1) \times dom(A_2)$

  $= \{\langle 0,a\rangle, \langle 0,b\rangle, \langle 0,c\rangle, \langle 1,a\rangle, \langle 1,b\rangle, \langle 1,c\rangle\}$

  – $r(R)$ could be $\{\langle 1,a\rangle, \langle 1,c\rangle\}$ (one possible extension)

CHARACTERISTICS of RELATION

1. Ordering of tuples in $r(R)$
   - Tuples are not considered to be order (set)

2. Ordering of attributes in schema R and of values within each tuple
   - Order of attributes in schema does not matter
   - Values in tuple must be ordered corresponding to the order of attributes

     eg: R(name, ID) and t = <"Bob", 123>

   - General: t = {<name, "Bob">, <ID, 123>}
     - called self-describing
     - order does not matter

3. Values in a tuple

- All values considered atomic (indivisible)
- Flat relational model
- How to represent composite & multivalued attributes?
- Each value of an attribute must be from domain of the attribute

eg: $t = \langle v_1, v_2, \ldots, v_n \rangle$ in the relation state $r(R)$ of $R(A_1, A_2, \ldots, A_n)$

each value $v_i \in dom(A_i)$ (+NULL if allowed)

- Special NULL value to represent unknown values/inapplicable data
- should be avoided as much as possible
- Can make fields mandatory to prevent NULL entries

- Interpretations of NULL
  - value unknown – missing
  - exists but unavailable – height
  - does not apply – office address

- Should not consider NULL while aggregating or comparing

Constraints

- Which values are permissible

1. Inherent model based or Implicit Constraints
2. Schema Based or Explicit Constraints
3. Application Based or Semantic Constraints

1. Inherent model based or Implicit Constraints

- Based on data Model itself
- Relational data model example:
  - set of rows (no repetition)
  - Order does not matter

2. Schema Based Constraints

Domain | Key | Null | Entity Integrity | Referential Integrity

(a) Domain Constraints

- Value v_i in column A_i must belong to the domain of A_i

$$v_i \in dom(A_i)$$

- Domain may or may not contain NULL
- Atomic value in $dom(A_i)$
- Datatypes or subrange of values from a datatype
- Eg: US phone numbers: (ddd) ddd-dddd

(b) Key constraints

- Superkey: set of attributes that can uniquely identify a tuple in a relation R is called the superkey of R
- No two tuples in any valid relation state r(R) will have the same value for SK

$$\text{if } t_1 \neq t_2, \; SK(t_1) \neq SK(t_2)$$

where t_1 & t_2 are tuples

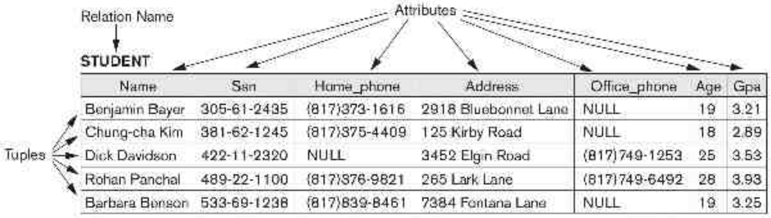
- **Key or Minimal Superkey:** a superkey K such that removal of any attribute from K results in a set of attributes K' that is not a superkey

- Both SK and K must be time invariant

- Default SK: set of all attributes

- Eg: {ssn} is a key

  {ssn, name, age} is superkey, not key



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 5.1
The attributes and tuples of a relation STUDENT.

Candidate key: each key in a relational schema is a candidate key

Primary key: chosen candidate key to uniquely identify the tuples and reference tuples from other relations
- typically candidate key with least no. of attributes

(c) Constraints on NULL

- Whether an attribute can have NULL values (mandatory values)

- Eg: in a STUDENT relation, the Name attribute is specified as NOT NULL

(d) Entity Integrity constraints

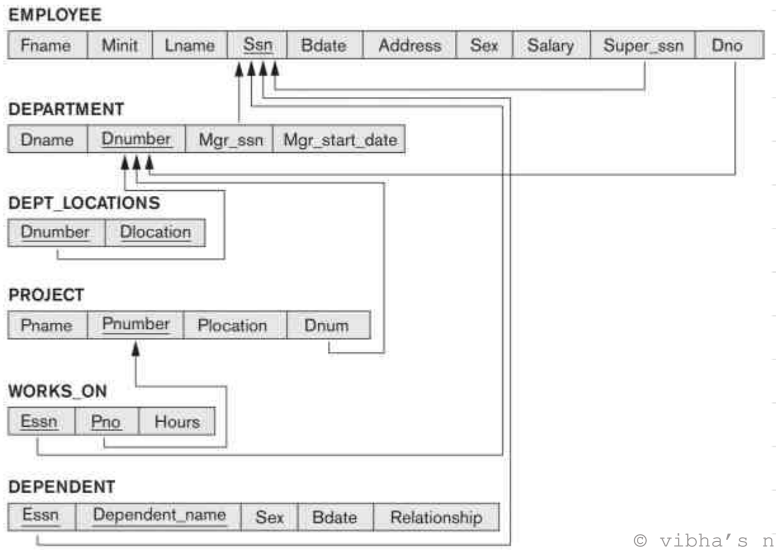
- Primary key of any relation cannot be NULL
- PK attributes of each relation schema R in S cannot have NULL values in any tuple in r(R)

(e) Referential Integrity constraints

- Foreign key : attribute that refers to another relation by utilising the primary key of the foreign attribute
- Dom(FK) = Dom(referencing attribute) ∪ NULL
- Used to specify relationship between two relation
- Tuples in referencing relation R1 have attributes FK that reference PK attributes of the referenced relation R2

$$t1[FK] = t2[PK]$$

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



- FK can refer to its own relation
  - Eg: Super-ssn in EMPLOYEE refers to supervisor

3. Application Based or Semantic Constraints

- Semantic constraints: defined in DB

DATABASE OPERATIONS

1. Update
2. Retrieval

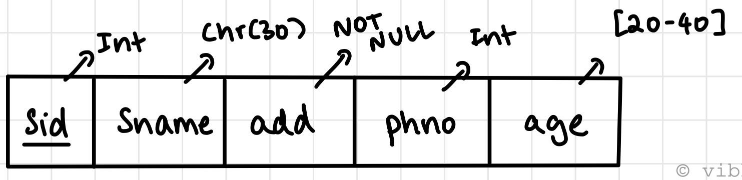
Update Operations

- INSERT
- DELETE
- MODIFY

- Languages: Relation algebra and relational calculus
  - no commercial DB uses anymore
  - only SQL used (underlying: RA and RC)

- If operation violates constraints
  (a) Cancel operation (REJECT or RESTRICT)
  (b) Execute but inform user
  (c) Trigger additional updates to correct violation (CASCADE or SET NULL)
  (d) Execute user-specified error-correction routine

INSERT OPERATION



STUDENT

insert <10 A x... 912 22 > ✓

insert <10 A x... 912 21 > ✗

violates entity integrity

- Default behaviour of INSERT during constraint violation is to reject

- Violation of referential integrity: can choose to either prompt user to enter into the foreign relation or reject the insert

- Can violate the following constraints (default: reject)
  - Domain constraint
  - Key constraint
  - Referential Integrity constraint
  - Entity Integrity constraint

- Eg (EMPLOYEE DB on next page)

1. Insert <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.

   violates entity integrity (NULL)

2. Insert <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> into EMPLOYEE.

   violates key constraint (SSN)

3. Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windswept, Katy, TX', F, 28000, '987654321', 7> into EMPLOYEE.

   violates referential integrity (Dept)

4. Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.

   satisfies all

Figure 5.6
One possible database state for the COMPANY relational database schema.

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

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

DELETE OPERATION

- Can violate referential integrity constraint if PK of tuple is being referenced
- Solutions
  - RESTRICT
  - CASCADE
  - SET NULL
- Eg:

1. Delete the WORKS_ON tuple with Essn = '999887777' and Pno = 10.

   no violations, 1 deleted

2. Delete the EMPLOYEE tuple with Ssn = '999887777'.

   violates referential integrity, rejected

3. Delete the EMPLOYE'E tuple with Ssn = '333445555'.

   violates many referential constraints, rejected

UPDATE OPERATION

- Updating PK
  - like DELETE followed by INSERT
  - same problems| solutions
- Updating FK
  - violation of referential integrity
- Updating any other key
  - Domain
  - Null

- Eg:

1. Update the salary of the EMPLOYEE tuple with Ssn = '999887777' to 28000.

acceptable

2. Update the Dno of the EMPLOYEE tuple with Ssn = '999887777' to 7.

violates referential integrity, rejected

3. Update the Ssn of the EMPLOYEE tuple with Ssn = '999887777' to '987654321'.

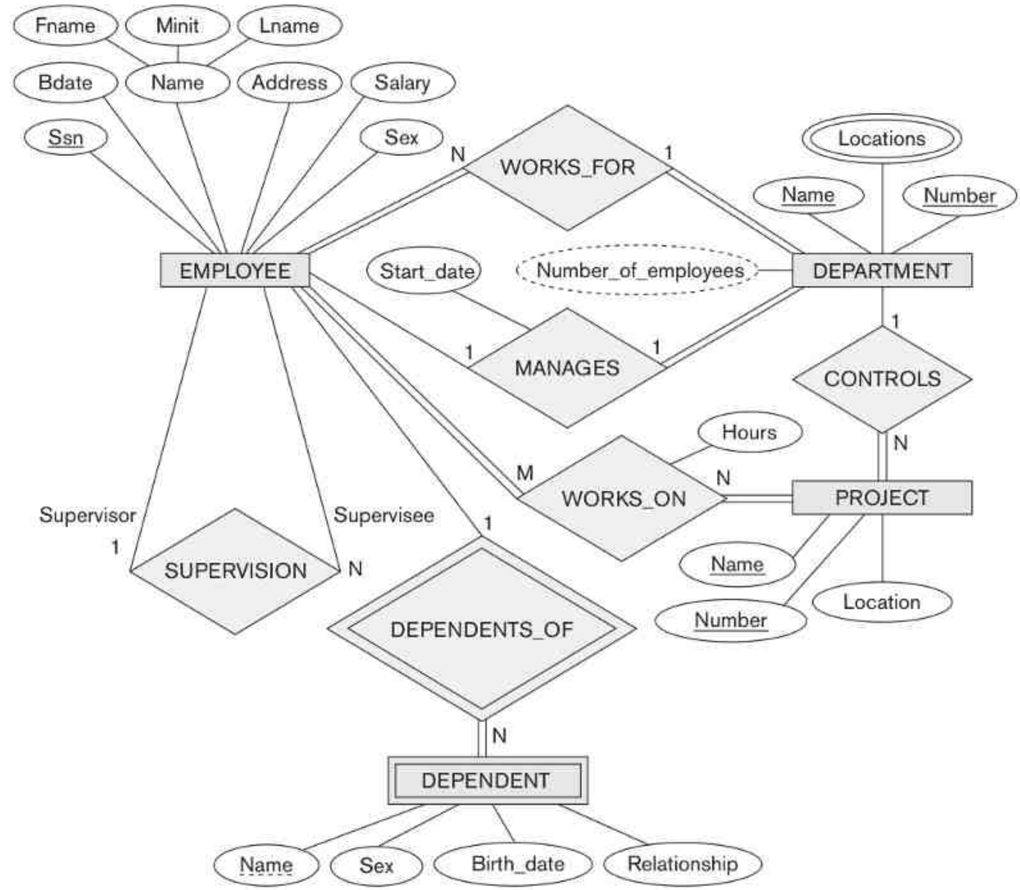
violates PK, referential integrity, rejected

ER to Relational Mapping

- Step 1: Mapping of Regular Entity Types
- Step 2: Mapping of Weak Entity Types
- Step 3: Mapping of Binary 1:1 Relationship Types
- Step 4: Mapping of Binary 1:N Relationship Types
- Step 5: Mapping of Binary M:N Relationship Types
- Step 6: Mapping of Multivalued Attributes
- Step 7: Mapping of N-ary Relationships

- Constraints must be enforced
- Nulls should be minimised
- All information must be retained

Figure 9.1
The ER conceptual schema diagram for the COMPANY database.



Step 1: Mapping of Regular Entity Types

- For each strong entity type E in the ER schema, create a relation R that includes all the simple attributes of E
- choose PK
- If chosen key is composite, set of keys forms PK
- Create EMPLOYEE, PROJECT, DEPARTMENT

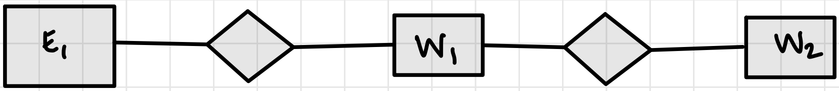
Step 2: Mapping of Weak Entity Types

- For each weak entity W in ER schema with owner entity E, create relation R and include all simple attributes or simple components of composite attributes of W as attributes of R

- FK attributes of R should be the PK attributes of E

- PK of R is combination of PK of E and partial keys of W

- Hierarchically resolve weak entities: W_1 before W_2

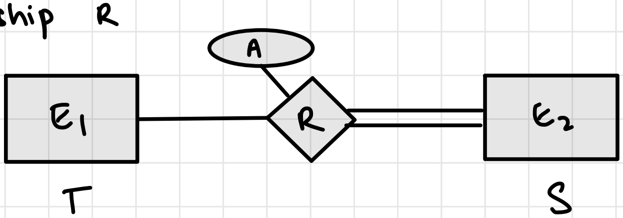


Step 3: Mapping of Binary 1:1 Relation Types

1. Foreign Key Approach

- Choose one of the relations (S) and include the PK of T as FK of S

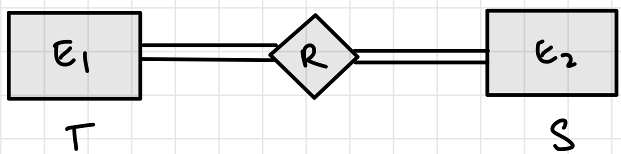
- Choose S such that it is a total participator in the relationship R



- Include all simple (not multivalued) attributes or simple components of composite attributes of the 1:1 relationship type R as attributes of S

2. Merged Relation Approach

- Merge the two entity types and the relationship into a single relation
- Both entities must have total participation in the relationship R (tables will have same no. of tuples)



3. Cross-Reference / Relationship Relation Approach

- Set up relation R that contains PKs of S and T as FKs in R
- R is called a relationship relation or lookup table where each tuple is a relationship instance
- PK will be any one of the FKs

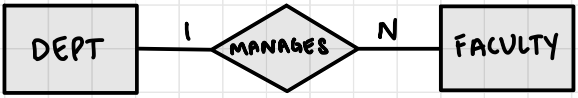
Step 4: Mapping of Binary 1:N Relationship Types

1. Foreign Key Approach

- Choose N side relation as S and the 1 side relation as T
- Include PK of T as FK in S (each instance on N side related to at most one instance on 1 side)

2. Cross-Reference / Relationship Relation Approach

- Set up relation R that contains PKs of S and T as FKs in R
- R is called a relationship relation or lookup table where each tuple is a relationship instance
- PK of R will be PK of S

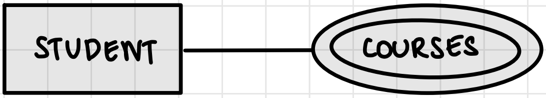


Step 5: Mapping of Binary M:N Relationships

- Cross-reference model
- Combination of PKs of S and T is PK of R
- Include all simple (not multivalued) attributes or simple components of composite attributes of the M:N relationship type R as attributes of S

Step 6: Mapping of Multivalued Attributes

- For each multivalued attribute, create relation R with PK as combination of E's PK and an attribute corresponding the multivalued attribute A



Step 7: Mapping of N-ary Relationships

- Relationship Relation

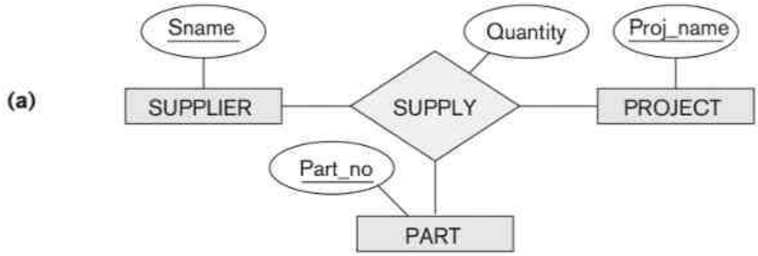


Figure 9.4
Mapping the *n*-ary relationship type SUPPLY from Figure 3.17(a).

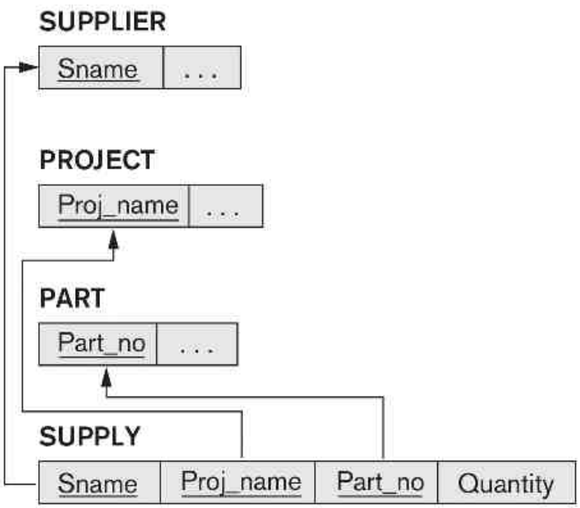


Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

RELATIONAL ALGEBRA & RELATIONAL CALCULUS

Relational Algebra
- Procedural
- Optimise queries

Relational Calculus
- Implemented in SQL
- Next unit
- Higher level declarative language
- RC – no order of operations

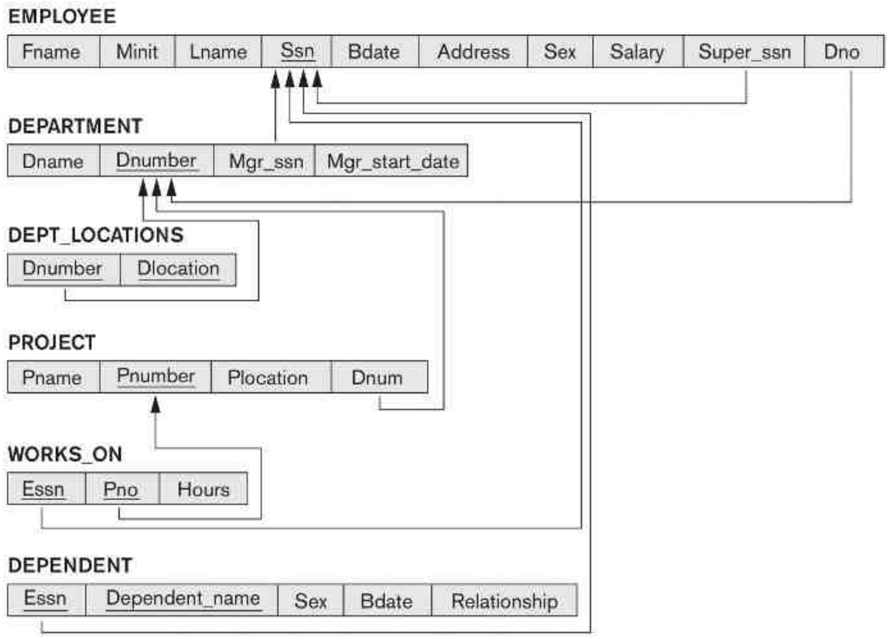
Relational Algebra

- Closed operations ; input a relation and produce a new relation

- Unary operations
- Set theory operations
- Binary operations
- Additional operations

Schema

Figure 5.7
Referential integrity constraints displayed on the COMPANY relational database schema.



(1) UNARY OPERATIONS

1.1 SELECT OPERATION

- Unary (R) : $\sigma_{\langle condition \rangle}(R)$ — sigma
- Input: relation, Output: relation with same schema (attributes)
- Choose a subset of tuples in the relation R based on a condition
- Eg: $\sigma_{Salary > 25000}$ (EMPLOYEE)
- Can use combination of logical operators (AND, OR, NOT)

- Eg: $\sigma_{Dno = 2 \text{ AND } Salary > 30000}$ (EMPLOYEE)
- Commutative operation
- Cascade of σ operations can be replaced with a single σ operation with conditions combined with ANDs

1.2 PROJECT OPERATION

- $\pi_{\langle attributes \rangle}(R)$ — pi
- Display subset of all attributes with all unique tuples (≤ tuples in original)
- Eg: $\pi_{Lname, salary, sex}$ (EMPLOYEE)
- If one of attributes is a superkey, no. of tuples will be the same
- Not commutative

Figure 8.1
Results of SELECT and PROJECT operations. (a) $\sigma_{(Dno=4 \text{ AND } Salary>25000) \text{ OR } (Dno=5 \text{ AND } Salary>30000)}$(EMPLOYEE). (b) $\pi_{Lname, Fname, Salary}$(EMPLOYEE). (c) $\pi_{Sex, Salary}$(EMPLOYEE).

(a)

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
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

(b)

Lname	Fname	Salary
Smith	John	30000
Wong	Franklin	40000
Zelaya	Alicia	25000
Wallace	Jennifer	43000
Narayan	Ramesh	38000
English	Joyce	25000
Jabbar	Ahmad	25000
Borg	James	55000

(c)

Sex	Salary
M	30000
M	40000
F	25000
F	43000
M	38000
M	25000
M	55000

Relational Algebra Expressions

- Can either store intermediate results or write a single statement
- Eg: $R' \leftarrow \sigma_{Dno=5}$

  $Result \leftarrow \pi_{Fname,\ Lname}(R')$

- Single line:

  $Result \leftarrow \pi_{Fname,Lname}(\sigma_{Dno=5}(R))$

(a)

Fname	Lname	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

(b)

TEMP

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
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

R

First_name	Last_name	Salary
John	Smith	30000
Franklin	Wong	40000
Ramesh	Narayan	38000
Joyce	English	25000

Figure 8.2
Results of a sequence of operations. (a) $\pi_{Fname,\ Lname,\ Salary}(\sigma_{Dno=5}(EMPLOYEE))$.
(b) Using intermediate relations and renaming of attributes.

1.3 RENAME OPERATION

- $\rho_{S(B1, B2, B3, \ldots, Bn)}(R)$ to rename R to S and the attributes to B1, B2, ..., Bn — rho
- $\rho_S(R)$ changes only relation name
- $\rho_{(B1, B2, \ldots, Bn)}(R)$ changes only all column names
- $\rho_{(A1 \rightarrow B1)}(R)$ changes column name A1 to B1
- https://towardsdatascience.com/a-quick-guide-to-relational-algebra-operators-in-dbms-1ff2ddecaad7
- Can use with set theory operations to make more meaningful

(2) SET THEORY OPERATIONS

2.1 UNION OPERATION

- $R \cup S$
- Conditions: union compatible
  - degree is same (same no. of attributes)
  - domain is same
- Eg: SSN of all employees who either work in department 5 or directly supervise an employee who works in department 5
- Resultant relation attributes of R by default (first relation)

$DEP5\_EMP \leftarrow \sigma_{Dno=5}(EMPLOYEE)$

$RESULT1 \leftarrow \pi_{SSN}(DEP5\_EMPS)$

$RESULT2_{(SSN)} \leftarrow \pi_{Superssn}(DEP5\_EMPS)$

$RESULT \leftarrow RESULT1 \cup RESULT2$

RESULT1

Ssn
123456789
333445555
666884444
453453453

RESULT2

Ssn
333445555
888665555

RESULT

Ssn
123456789
333445555
666884444
453453453
888665555

Figure 8.3
Result of the UNION operation RESULT ← RESULT1 ∪ RESULT2.

[4] As a single relational algebra expression, this becomes Result ← $\pi_{Ssn}(\sigma_{Dno=5}(EMPLOYEE)) \cup \pi_{Super\_ssn}(\sigma_{Dno=5}(EMPLOYEE))$.

2.2 INTERSECTION OPERATION

- $R \cap S$
- Union and intersection are commutative
- Same dimensions ; resultant relation attributes of R by default (first relation)

2.3 SET DIFFERENCE OPERATION

- $R - S$

$$R \cap S = ((R \cup S) - (R - S)) - (S - R)$$

Figure 8.4
The set operations UNION, INTERSECTION, and MINUS. (a) Two union-compatible relations. (b) STUDENT ∪ INSTRUCTOR. (c) STUDENT ∩ INSTRUCTOR. (d) STUDENT − INSTRUCTOR. (e) INSTRUCTOR − STUDENT.

(a) STUDENT

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

(b)

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

(c)

Fn	Ln
Susan	Yao
Ramesh	Shah

(d)

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

(e)

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

2.4 CARTESIAN PRODUCT

- Combine tuples from two relations together
- $R(A1, A2, \ldots, An) \times S(B1, B2, \ldots, Bm)$
- Result is $Q(A1, A2, \ldots, An, B1, B2, \ldots, Bm)$
- One tuple for each combination of tuples
- No. of tuples in Q = no. of tuples in R * no. of tuples in S
- Not a very meaningful operation
- Tuples may not always have meaning
- PK is combination of PKs from each participating relation

STUDENT

Sid	Name	Dno
1	A	d_1
2	B	d_1
3	C	d_2
4	D	d_3

DEPARTMENT

dno	dname
d_1	CS
d_2	IS
d_3	EC

STUDENT x DEPARTMENT

Sid	Name	Dno	dno	dname
1	A	d_1	d_1	CS
1	A	d_1	d_2	IS
1	A	d_1	d_3	EC
2	B	d_1	d_1	CS
2	B	d_1	d_2	IS
2	B	d_1	d_3	EC
3	C	d_2	d_1	CS
3	C	d_2	d_2	IS
3	C	d_2	d_3	EC
4	D	d_3	d_1	CS
4	D	d_3	d_2	IS
4	D	d_3	d_3	EC

spurious tuples

Relation: Singers

Singer-id	Singer-name	Address	Age
0126	Helen Drummond	1 Thorley Street	42
0243	Katerina Christou	12 High Road	37
0247	Desmond Venables	27 Long Lane	55
0259	Anne Freeman	5 Tower Hill	40
0594	Alphonse Trieste	20 Longchamps	34
0628	Tamanna Patel	9 Crown Hill	23
0855	Swee Hor Tan	4 Long Lane	54
0876	Panos Constantinou	32 Mallet Road	49

Relation: Roles

Role-id	Role-name
0101	Figaro
0175	Mimi

New relation: Singers-Roles

Singer-id	Name	Address	Age	Role-id	Role-name
0126	Helen Drummond	1 Thorley Street	42	0101	Figaro
0126	Helen Drummond	1 Thorley Street	42	0175	Mimi
0243	Katerina Christou	12 High Road	37	0101	Figaro
0243	Katerina Christou	12 High Road	37	0175	Mimi
0247	Desmond Venables	27 Long Lane	55	0101	Figaro
0247	Desmond Venables	27 Long Lane	55	0175	Mimi
0259	Anne Freeman	5 Tower Hill	40	0101	Figaro
0259	Anne Freeman	5 Tower Hill	40	0175	Mimi
0594	Alphonse Trieste	20 Longchamps	34	0101	Figaro
0594	Alphonse Trieste	20 Longchamps	34	0175	Mimi
0628	Tamanna Patel	9 Crown Hill	23	0101	Figaro
0628	Tamanna Patel	9 Crown Hill	23	0175	Mimi
0855	Swee Hor Tan	4 Long Lane	54	0101	Figaro
0855	Swee Hor Tan	4 Long Lane	54	0175	Mimi
0876	Panos Constantinou	32 Mallet Road	49	0101	Figaro
0876	Panos Constantinou	32 Mallet Road	49	0175	Mimi

Q: Consider the following relational DB schema consisting of the four relation schemas:

passenger (pid, pname, pgender, pcity)
agency (aid, aname, acity)
flight (fid, fdate, time, src, dest)
booking (pid, aid, fid, fdate)

Answer the following questions

(i) Get complete details of all flights to New Delhi

$$\sigma_{dest = New\ Delhi}(flight)$$

(ii) Get complete details of all flights from Chennai to New Delhi

$$\sigma_{dest = New\ Delhi\ and\ Chennai}(flight)$$

(iii) Get complete details of all flights scheduled on both dates 01/12/2020 and 02/12/2020 at 16:00

$$(\sigma_{fdate = 01/12/2020\ AND\ time = 16:00}(flight)) \cap (\sigma_{fdate = 02/12/2020\ AND\ time = 16:00}(flight))$$

(iii) Get complete details of all flights scheduled on either of the dates or both dates 01/12/2020 and 02/12/2020 at 16:00

$$(\sigma_{fdate = 01/12/2020\ AND\ time = 16:00}(flight)) \cup (\sigma_{fdate = 02/12/2020\ AND\ time = 16:00}(flight))$$

$$\sigma_{(fdate = 01/12/2020\ OR\ fdate = 02/12/2020)\ AND\ time = 16:00}(flight)$$

(3) BINARY RELATIONAL OPERATIONS

3.1 JOIN OPERATION

- Performs cartesian product followed by select in a single step

- Theta join $R \bowtie_{A_i \theta B_i} S$

  from S

  from R

  operator (=, >=, <=, <> etc)
  or expression with AND, OR, NOT

- Equijoin $R \bowtie_{A_i = B_i} S$

- Tuples for which join attributes are NULL do not appear in the result

Figure 8.6
Result of the JOIN operation DEPT_MGR ← DEPARTMENT $\bowtie_{Mgr\_ssn=Ssn}$ EMPLOYEE.

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

3.2 NATURAL JOIN

- One attribute name must be common to both relations

- Equijoin followed by project (remove redundant columns A_i and B_i)

- $R * S$

- can join / natural join as many times

(a)

PROJ_DEPT

Pname	Pnumber	Plocation	Dnum	Dname	Mgr_ssn	Mgr_start_date
ProductX	1	Bellaire	5	Research	333445555	1988-05-22
ProductY	2	Sugarland	5	Research	333445555	1988-05-22
ProductZ	3	Houston	5	Research	333445555	1988-05-22
Computerization	10	Stafford	4	Administration	987654321	1995-01-01
Reorganization	20	Houston	1	Headquarters	888665555	1981-06-19
Newbenefits	30	Stafford	4	Administration	987654321	1995-01-01

(b)

DEPT_LOCS

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

Figure 8.7
Results of two natural join operations. (a) proj_dept ← project * dept.
(b) dept_locs ← department * dept_locations.

Complete Set of Relational Operations

- $\sigma, \pi, \cup, -, \rho, \times$
- Any operation can be expressed as a combination of these 6
- Natural join * using this set : $\rho \to \times \to \sigma \to \pi$

3.3 DIVISION OPERATION

- For special queries ; eg: "Fetch names of employees who work on all the projects that Rohan Swamy works on"
- $R(Z) \div S(X)$ where $X \subseteq Z$

- $Y = Z - X \Rightarrow Z = X \cup Y$
- Result = T(Y) that includes a tuple t if tuples t_R appear in R with $t_R[Y] = t$ and with $t_R[X] = t_S$ for every tuple t_S is S
- Values in t must appear in R in combination with every tuple in S
- Sequence of π, $\times$, $-$

Q: Retrieve the names of employees who work on all the projects that 'John Smith' works on.

$\text{SMITH} \leftarrow \sigma_{\text{Fname='John' AND Lname='Smith'}}(\text{EMPLOYEE})$
$\text{SMITH\_PNOS} \leftarrow \pi_{\text{Pno}}(\text{WORKS\_ON} \bowtie_{\text{Essn=Ssn}} \text{SMITH})$

$\text{SSN\_PNOS} \leftarrow \pi_{\text{Essn, Pno}}(\text{WORKS\_ON})$

$\text{SSNS(Ssn)} \leftarrow \text{SSN\_PNOS} \div \text{SMITH\_PNOS}$
$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname}}(\text{SSNS} * \text{EMPLOYEE})$

Figure 8.8
The DIVISION operation. (a) Dividing SSN_PNOS by SMITH_PNOS. (b) $T \leftarrow R \div S$.

(a)

SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

Pno
1
2

SSNS

Ssn
123456789
453453453

(b)

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

Q: Return dept details that control the project

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 5.5
Schema diagram for the COMPANY relational database schema.

$$PROJECT * \rho_{Dnumber \rightarrow Dnum}(DEPARTMENT)$$

specify all column names for test

Join Selectivity

- $\frac{\text{Expected size of join result}}{\text{maximum size}} = \frac{[0, n_R * n_S]}{n_R * n_S}$

- Ratio ≤ 1

Table 8.1 Operations of Relational Algebra

OPERATION	PURPOSE	NOTATION
SELECT	Selects all tuples that satisfy the selection condition from a relation R.	$\sigma_{\text{<selection condition>}}(R)$
PROJECT	Produces a new relation with only some of the attributes of R, and removes duplicate tuples.	$\pi_{\text{<attribute list>}}(R)$
THETA JOIN	Produces all combinations of tuples from R_1 and R_2 that satisfy the join condition.	$R_1 \bowtie_{\text{<join condition>}} R_2$
EQUIJOIN	Produces all the combinations of tuples from R_1 and R_2 that satisfy a join condition with only equality comparisons.	$R_1 \bowtie_{\text{<join condition>}} R_2$, OR $R_1 \bowtie_{(\text{<join attributes 1>}),(\text{<join attributes 2>})} R_2$
NATURAL JOIN	Same as EQUIJOIN except that the join attributes of R_2 are not included in the resulting relation; if the join attributes have the same names, they do not have to be specified at all.	$R_1 *_{\text{<join condition>}} R_2$, OR $R_1 *_{(\text{<join attributes 1>}),(\text{<join attributes 2>})} R_2$ OR $R_1 * R_2$
UNION	Produces a relation that includes all the tuples in R_1 or R_2 or both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cup R_2$
INTERSECTION	Produces a relation that includes all the tuples in both R_1 and R_2; R_1 and R_2 must be union compatible.	$R_1 \cap R_2$
DIFFERENCE	Produces a relation that includes all the tuples in R_1 that are not in R_2; R_1 and R_2 must be union compatible.	$R_1 - R_2$
CARTESIAN PRODUCT	Produces a relation that has the attributes of R_1 and R_2 and includes as tuples all possible combinations of tuples from R_1 and R_2.	$R_1 \times R_2$
DIVISION	Produces a relation $R(X)$ that includes all tuples $t[X]$ in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$, where $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

QUERY TREE

- Representation of relational algebraic structure
- Input relations: leaf nodes

Q: *For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birth date.*

Create query tree for the above query

Query:

$$DEPTS \leftarrow \rho_{Dnumber = Dnum}(DEPARTMENT) * \sigma_{Plocation = Stafford}(PROJECT)$$

$$ALL \leftarrow \rho_{Ssn = Mgr\text{-}ssn}(EMPLOYEE) * DEPTS$$

$$RESULT \leftarrow \pi_{Pnumber, Dnum, Lname, Address, Bdate}(ALL)$$

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 5.5
Schema diagram for the COMPANY relational database schema.

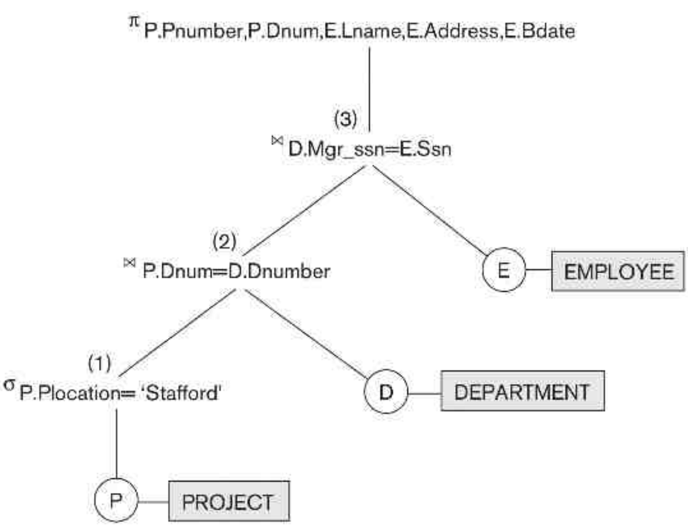


Figure 8.9
Query tree corresponding to the relational algebra expression for Q2.

(4) ADDITIONAL RELATIONAL OPERATIONS

4.1 FINALIZED PROJECT

- Requirement is to generate a report on a relation

As an example, consider the relation

EMPLOYEE (Ssn, Salary, Deduction, Years_service)

A report may be required to show

Net Salary = Salary – Deduction,
Bonus = 2000 * Years_service, and
Tax = 0.25 * Salary

Then a generalized projection combined with renaming may be used as follows:

$$\text{REPORT} \leftarrow \rho_{(\text{Ssn, Net\_salary, Bonus, Tax})}(\pi_{\text{Ssn, Salary – Deduction, 2000 * Years\_service, 0.25 * Salary}}(\text{EMPLOYEE}))$$

$\pi_{F_1, F_2, F_3}(R)$

function π operator

4.2 AGGREGATE FUNCTIONS AND GROUPING

- $\mathfrak{F}$: script F symbol
- $_{\text{<group attributes>}} \mathfrak{F}_{\text{<aggregate function list>}}(R)$

4.1.1 Aggregate

- $\mathfrak{F}_{\text{MAX salary}}(\text{EMPLOYEE})$
- max, min, average, sum, count

4.1.2 Grouping

- List of attributes

- $Dno\ \Im\ _{COUNT\ Ssn,\ AVERAGE\ Salary}(EMPLOYEE)$

R

(a)

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

(b)

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

(c)

Count_ssn	Average_salary
8	35125

Figure 8.10
The aggregate function operation.

a. $\rho_{R(Dno,\ No\_of\_employees,\ Average\_sal)}(_{Dno}\ \Im\ _{COUNT\ Ssn,\ AVERAGE\ Salary}(EMPLOYEE))$.

b. $_{Dno}\ \Im\ _{COUNT\ Ssn,\ AVERAGE\ Salary}(EMPLOYEE)$.

c. $\Im\ _{COUNT\ Ssn,\ AVERAGE\ Salary}(EMPLOYEE)$.

- Duplicates are not eliminated when aggregated
- NULL values are not considered
- Result of aggregation is a relation
- In SQL, we can eliminate duplicates while applying aggregate functions

Q: Consider the following schema

Sailors (sid: integer, sname: string, rating: integer, age: real)
Boats (bid: integer, bname: string, color: string)
Reserves (sid: integer, bid: integer, day: date)

1. Retrieve rows corresponding to expert (rating > 8) sailors

$\sigma_{rating>8}\ (Sailors)$

2. Retrieve rating of each sailor

$$\pi_{rating, sid}(Sailors)$$

3. Retrieve age of sailor

$$\pi_{sname, age}(Sailor)$$

4. Retrieve name and ratings of highly rated sailors

$$\pi_{sname, rating}(\sigma_{rating>8}(Sailors))$$

5. Find names of sailors who have reserved boat 103

$$\pi_{sname}(Sailors * \sigma_{bid=103}(Reserves))$$

6. Find names of sailors who have reserved a red boat

$$\pi_{sname}(\sigma_{color=red}(Boats) * Reserves * Sailors)$$

7. Find colors of boats reserved by Lubber

$$\pi_{color}(\sigma_{sname=Lubber}(Saliors) * Reserves * Boats)$$

8. Find names of sailors who have reserved at least one boat

$$\pi_{sname}(Sailors * Reserves)$$

9. Find the names of sailors who have reserved a red or a green boat

$$\pi_{sname}(Sailors * Reserves * \sigma_{color=red \text{ OR } color=green}(Boats))$$

10. Find the names of sailors who have reserved a red and a green boat

$$\pi_{sname}(Sailors * (\underbrace{\pi_{sid}(Reserves * \sigma_{color=red}(Boats))}^{\text{only sids}} \cap$$

$$\underbrace{\pi_{sid}(Reserves * \sigma_{color=green}(Boats))}_{\text{only sids}}))$$

NOTE:

$$\pi_{sname}(Sailors * Reserves * (\underbrace{\sigma_{color=red}(Boats) \wedge \sigma_{colour=green}(Boats)}_{\text{will give empty set}}))$$

IS WRONG!

11. Find sids of Sailors with age over 20 who have not reserved a red boat

$$\pi_{sid}(\sigma_{age>20}(Sailors)) - \pi_{sid}((\sigma_{color=red}(Boats) * Reserves * Sailors)$$

NOTE:

$$\pi_{sid}(\sigma_{age>20}(Sailors) * Reserves * \sigma_{color <> red}(Boats))$$

IS WRONG!

Sailors who have not reserved any boats at all will be excluded

12. Find the names of sailors who have reserved all the boats

$$\pi_{sname}((\pi_{bid,sid}(Reserves) \div \pi_{bid}(Boats)) * Sailors)$$

13. Find the names of sailors who have reserved all boats called Interlake

$$\pi_{sname}(\pi_{bid,sid}(Reserves) \div \pi_{bid}(\sigma_{bname=Interlake}(Boats)) * Sailors)$$

4.3 RECURSIVE CLOSURE

- Eg: employees supervising other employees
- Self-join where there is a recursive relationship

Q: Return all the SSNs of people directly supervised by John Smith and supervised by Smith's subordinates

$$JOHN\_SSN \leftarrow \pi_{ssn}(\sigma_{ssn='John\ Smith'}(Employee))$$

$$SUBS \leftarrow \pi_{ssn}(JOHN\_SSN \bowtie_{ssn=superssn} Employee)$$

$$SUBSUBS \leftarrow \pi_{ssn}(SUBS_{ssn} \bowtie_{=superssn} Employee)$$

$$RESULT \leftarrow SUBS \cup SUBSUBS$$

Q: Return all the SSNs of people directly supervised by James Borg and supervised by Smith's subordinates

$$BORG\_SSN \leftarrow \pi_{Ssn}(\sigma_{Fname='James' \text{ AND } Lname='Borg'}(EMPLOYEE))$$

$$SUPERVISION(Ssn1, Ssn2) \leftarrow \pi_{Ssn,Super\_ssn}(EMPLOYEE)$$

$$RESULT1(Ssn) \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} BORG\_SSN)$$

$$RESULT2(Ssn) \leftarrow \pi_{Ssn1}(SUPERVISION \bowtie_{Ssn2=Ssn} RESULT1)$$

$$RESULT \leftarrow RESULT2 \cup RESULT1$$

SUPERVISION

(Borg's Ssn is 888665555)

(Ssn) Ssn1	(Super_ssn) Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

RESULT1

Ssn
333445555
987654321

(Supervised by Borg)

RESULT2

Ssn
123456789
999887777
666884444
453453453
987987987

(Supervised by Borg's subordinates)

RESULT

Ssn
123456789
999887777
666884444
453453453
987987987
333445555
987654321

(RESULT1 ∪ RESULT2)

Figure 8.11
A two-level recursive query.

4.4 OUTER JOIN

- So far, all joins have been inner joins
- Excludes tuples that do not match the condition and tuples that have NULL values in the joining attribute
- Eg: details of all employees and their dependents, if there are dependents
  - Cannot use inner join as those with NULL are excluded
  - Must show all employees and show NULL for dependents (padding with NULL values)
- $E \underset{ssn = essn}{⟕} D$ — left outer join
- $R \underset{c_1}{⟖} S$ — right outer join

- $R \underset{c_1}{⟗} S$ — full outer join

Q: List of all employees and if they manage a department, the name of the department

Figure 8.12
The result of a LEFT OUTER JOIN operation.

RESULT

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

$$\text{TEMP} \leftarrow (\text{EMPLOYEE} ⟕_{\text{Ssn}=\text{Mgr\_ssn}} \text{DEPARTMENT})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Minit, Lname, Dname}}(\text{TEMP})$$

Q:

T_1

P	Q	R
10	a	5
15	b	8
25	a	6

T_2

A	B	C
10	b	6
25	c	3
10	b	5

(a) $T_1 \underset{P = A}{⟕} T_2$

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	b	5
25	a	6	25	c	3
15	b	8	NULL	NULL	NULL

(b) $T_1 \underset{Q=B}{⟖} T_2$

P	Q	R	A	B	C
15	b	8	10	b	6
15	b	8	10	b	5
NULL	NULL	NULL	25	c	3

(c) $T_1 \underset{Q=B}{⟗} T_2$

P	Q	R	A	B	C
15	b	8	10	b	6
15	b	8	10	b	5
NULL	NULL	NULL	25	c	3
10	a	5	NULL	NULL	NULL
25	a	6	NULL	NULL	NULL

(d) $T_1 \underset{P=A}{\bowtie} T_2$

P	Q	R	A	B	C
10	a	5	10	b	6
10	a	5	10	b	5
25	a	6	25	c	3

(e) $T_1 \cup T_2$

P	Q	R
10	a	5
15	b	8
25	a	6
10	b	6
25	c	3
10	b	5

4.5 OUTER UNION

- If tables are partially compatible
- Two relations R(X, Y) and S(X, Z)
  - Result T(X, Y, Z)
- Eg: STUDENT(Name, Ssn, Department, Advisor)
  INSTRUCTOR(Name, Ssn, Department, Rank)

STUDENT_OR_INSTRUCTOR(Name, Ssn, Department, Advisor, Rank)

Q: Find the names of the employees who work on projects controlled by dept 5

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Figure 5.5
Schema diagram for the COMPANY relational database schema.

$$\pi_{fname}\left(employee \bowtie_{ssn=essn} \pi_{essn,\,dnum}\left(works\text{-}on \bowtie_{pno=pnumber} project\right)\right)$$