The Relational Data Model and Relational Database Constraints

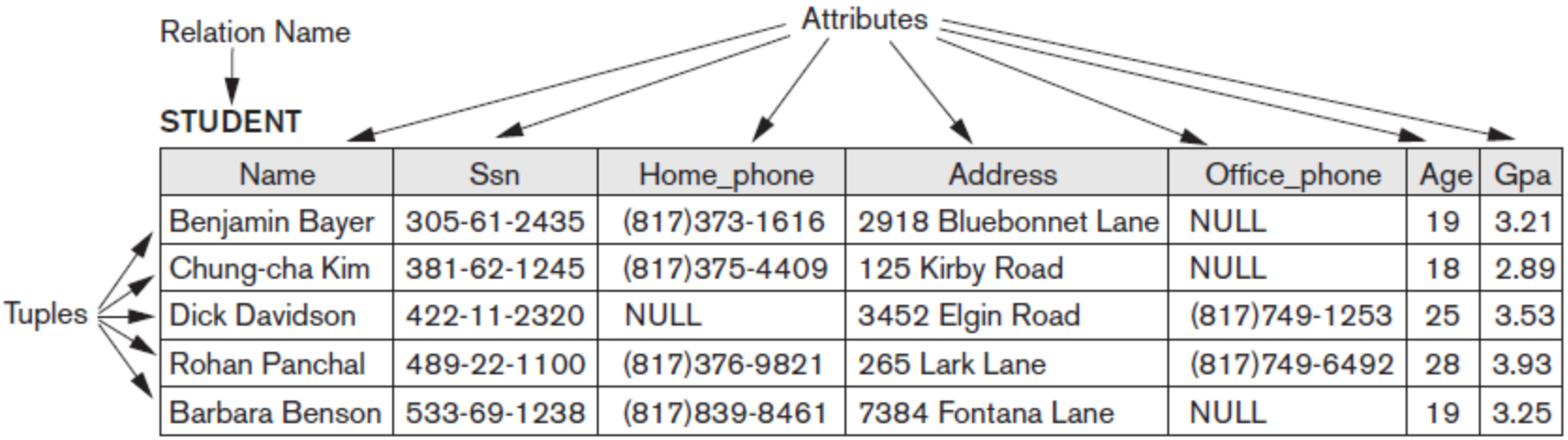
Lecture 4 Outline

- The Relational Data Model
- Relational Model Constraints
- Update Operations
- Relational Algebra

Relational Model Concepts

- Represents data as a collection of relations
- **Table** of values
  - Row
    - Represents a collection of related data values
    - Fact that typically corresponds to a real-world entity or relationship
    - *Tuple*
  - Table name and column names
    - Interpret the meaning of the values in each row
    - *Attribute*

Relational Model Concepts



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 3.1
The attributes and tuples of a relation STUDENT.

Domains, Attributes, Tuples, and Relations

- **Domain** D
  - Set of atomic values (Usa_phone_numbers (10 digits), Names (characters), Employee_ages (number from 15 to 80))
- Specifying a domain
  - **Data type (format)** specified for each domain
    - E.g: telephone_numbers as character string in the form (ddd)ddddddd.

Domains, Attributes, Tuples, and Relations

- **Relation schema** *R*
  - Denoted by $R(A_1, A_2, ..., A_n)$ e.g.
    - STUDENT(Name, Ssn) *or*
    - STUDENT(Name: string, Ssn: string))
  - Made up of a relation name *R* and a list of attributes, $A_1, A_2, ..., A_n$
- **Attribute** $\boldsymbol{A_i}$
  - Name of a role played by some domain *D* in the relation schema *R*
  - *D is called the domain of* A_i *and is denoted by* $dom(A_i)$*.*
- **Degree** of a relation
  - Number of attributes ***n*** of its relation schema

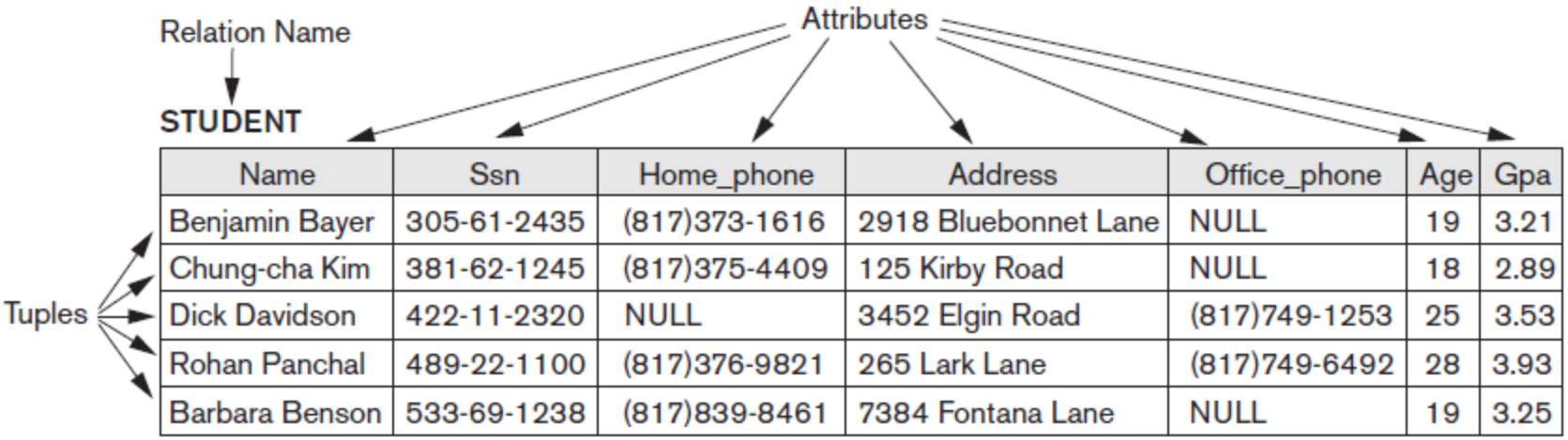
Domains, Attributes, Tuples, and Relations

- **Relation** (or **relation state**)
  - Set of ***n*-tuples** $r = \{t_1, t_2, ..., t_m\}$
  - Each n-tuple t
    - Ordered list of n values $t = <v_1, v_2, ..., v_n>$
    - Each value v_i, $1 \leq i \leq n$, is an element of dom(A_i) or is a special NULL value

Characteristics of Relations

- **Ordering of tuples in a relation**
  - Relation defined as a set of tuples
  - Elements have no order among them
    - When saved as a file or displayed as a table, we may specify a particular ordering
- **Ordering of values within a tuple**
  - Order of attributes and values is not that important
  - As long as correspondence between attributes and values maintained
  - However, for simplicity, we will consider *ordered attributes*

Relational Model Concepts



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 3.1
The attributes and tuples of a relation STUDENT.

Characteristics of Relations

Figure 3.2
The relation STUDENT from Figure 3.1 with a different order of tuples.

STUDENT

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

Characteristics of Relations

- Values and NULLs in tuples
  - Each value in a tuple is atomic
  - **Flat relational model**
    - Composite and multivalued attributes not allowed
  - Multivalued attributes
    - Must be represented by separate relations
  - Composite attributes
    - Represented only by simple component attributes in basic relational model

Characteristics of Relations

- NULL values
  - Represent the values of attributes that may be unknown or may not apply to a tuple
  - Meanings for NULL values
    - *Value unknown*
    - *Attribute does not apply to this tuple (also known as value undefined)*

Relational Model Notation

- Relation schema R of degree n
  - Denoted by $R(A_1, A_2, \ldots, A_n)$
- Uppercase letters Q, R, S
  - Denote relation names
- Lowercase letters q, r, s
  - Denote relation states
- Letters t, u, v
  - Denote tuples

Relational Model Notation

- *n-tuple t* in a relation *r*(*R*)
  - Denoted by $t = <v_1, v_2, ..., v_n>$
  - v_i is the value corresponding to attribute A_i
- Component values of tuples:
  - $t[A_i]$ and $t.A_i$ refer to the value v_i in t for attribute A_i
  - $t[A_u, A_w, ..., A_z]$ and $t.(A_u, A_w, ..., A_z)$ refer to the subtuple of values $<v_u, v_w, ..., v_z>$ from t corresponding to the attributes specified in the list

Relational Model Notation

- Example, consider the tuple *t* = <‘Barbara Benson’, ‘533-69-1238’, ‘(817)839-8461’, ‘7384 Fontana Lane’, NULL, 19, 3.25> from the STUDENT relation
- we have *t*[Name] = <‘Barbara Benson’>, and *t*[Ssn, Gpa, Age] = <‘533-69-1238’, 3.25, 19>.

Relational Model Constraints

Constraints

- Restrictions on the actual values in a database state
- Derived from the rules in the miniworld that the database represents

- **Inherent model-based constraints** or **implicit constraints**
  - Inherent in the data model (e.g. a relation cannot have duplicate tuples)

Relational Model Constraints

- **Inherent model-based constraints** or **implicit constraints:** Constraints that are inherent in the data model (Data Types, Primary Key, Forigen Key).
- **Schema-based constraints** or **explicit constraints:** Can be directly expressed in schemas of the data model (A manager can not manage two departments)
- **Application-based** or **semantic constraints** or **business rules:** Cannot be directly expressed in schemas. Expressed and enforced by application program (this year's salary increase can be no more than last year's)

1)Domain Constraints

- The data types associated with domains typically include Typically include:
  - Numeric data types for integers and real numbers
  - Characters
  - Booleans
  - Fixed-length strings
  - Variable-length strings
  - Date, time, timestamp
  - Money
  - Other special data types

2)Key Constraints and Constraints on NULL Values

- By definition, **no** two tuples can have the same combination of values for all their attributes.
- **Superkey (SK)**
  - No two distinct tuples in any state *r* of *R* can have the same value for SK
  - A combination of all attributes is a SK
  - t1[SK]≠ t2[SK]

2)Key Constraints and Constraints on NULL Values

- Key satisfies two properties:
  - Two distinct tuples in any state of relation cannot have identical values for (all) attributes in key
  - Minimal superkey: Cannot remove any attributes and still have uniqueness constraint in above condition hold
- A key is not just a property for a relation state but, rather, is a *time-invariant* feature

2)Key Constraints and Constraints on NULL Values

- **Candidate key**
  - Relation schema may have more than one key
- **Primary key** of the relation
  - Designated among candidate keys
  - Underline attribute
- Other candidate keys are designated as **unique keys**
- **Another constraint** on attributes specifies whether NULL is permitted or not for an attribute

2)Key Constraints and Constraints on NULL Values

CAR

License_number	Engine_serial_number	Make	Model	Year
Texas ABC-739	A69352	Ford	Mustang	02
Florida TVP-347	B43696	Oldsmobile	Cutlass	05
New York MPO-22	X83554	Oldsmobile	Delta	01
California 432-TFY	C43742	Mercedes	190-D	99
California RSK-629	Y82935	Toyota	Camry	04
Texas RSK-629	U028365	Jaguar	XJS	04

Figure 3.4
The CAR relation, with two candidate keys: License_number and Engine_serial_number.

3)Entity Integrity, Referential Integrity, and Foreign Keys

- **Entity integrity constraint**
  - No primary key value can be NULL
- **Referential integrity constraint**
  - Specified between two relations
  - Maintains consistency among tuples in two relations

3)Entity Integrity, Referential Integrity, and Foreign Keys

- **Foreign key** rules:
  - The attributes in FK have the same domain(s) as the primary key attributes PK
  - Value of FK in a tuple t_1 of the current state $r_1(R_1)$ either occurs as a value of PK for some tuple t_2 in the current state $r_2(R_2)$ or is NULL

  If both these conditions hold, a **referential integrity constraint** from R_1 to R_2 is said to hold

3)Entity Integrity, Referential Integrity, and Foreign Keys

- Diagrammatically display referential integrity constraints
  - Directed arc from each foreign key to the relation it references

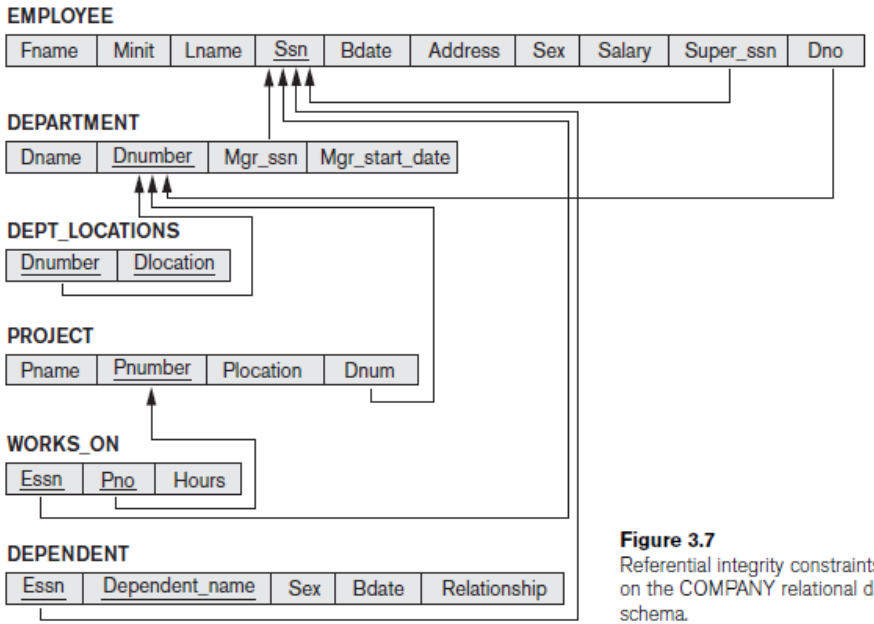


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

Other Types of Constraints

- Semantic integrity constraints
  - May have to be specified and enforced on a relational database
  - Use **triggers** and **assertions**
  - More common to check for these types of constraints within the application programs

Update Operations, and Dealing with Constraint Violations

- Operations of the relational model can be categorized into **retrievals** and **updates**
- Basic operations that change the states of relations in the database:
  - Insert
  - Delete
  - Update (or Modify)

Figure 3.6
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

Figure 3.6
One possible database state for the COMPANY relational database schema.

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

The Insert Operation

- Provides a list of attribute values for a new tuple t that is to be inserted into a relation R
- Can violate what?
  - Can violate any of the four types of constraints
    - Domain constraints, key constraints, entity integrity constraints, referential integrity constraints
  - If an insertion violates one or more constraints
    - Default option is to reject the insertion

The Insert Operation

Operation:
Insert <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.
Result: This insertion violates the entity integrity constraint (NULL for the primary key Ssn), so it is rejected.

Operation:
Insert <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> into EMPLOYEE.
Result: This insertion violates the key constraint because another tuple with the same Ssn value already exists in the EMPLOYEE relation, and so it is rejected.

Operation:
Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windswept, Katy, TX', F, 28000, '987654321', 7> into EMPLOYEE.
Result: This insertion violates the referential integrity constraint specified on Dno in EMPLOYEE because no corresponding referenced tuple exists in DEPARTMENT with Dnumber = 7.

Operation:
Insert <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> into EMPLOYEE.
Result: This insertion satisfies all constraints, so it is acceptable.

The Delete Operation

- Can violate what?
- Can violate only referential integrity
  - If tuple being deleted is referenced by foreign keys from other tuples
  - Solutions:
    - **Restrict:** Reject the deletion
    - **Cascade:** Propagate the deletion by deleting tuples that reference the tuple that is being deleted
    - **Set null** or **set default:** Modify the referencing attribute values that cause the violation
    - **Combinations are possible**
      - e.g. deleting an EMPLOYEE tuple may automatically mean deleting her/his dependents but not the department they manage

The Delete Operation

Operation:
Delete the WORKS_ON tuple with Essn = '999887777' and Pno = 10.
Result: This deletion is acceptable and deletes exactly one tuple.

Operation:
Delete the EMPLOYEE tuple with Ssn = '999887777'.
Result: This deletion is not acceptable, because there are tuples in WORKS_ON that refer to this tuple. Hence, if the tuple in EMPLOYEE is deleted, referential integrity violations will result.

Operation:
Delete the EMPLOYEE tuple with Ssn = '333445555'.
Result: This deletion will result in even worse referential integrity violations, because the tuple involved is referenced by tuples from the EMPLOYEE, DEPARTMENT, WORKS_ON, and DEPENDENT relations.

The Update Operation

- Necessary to specify a condition on attributes of relation
  - Select the tuple (or tuples) to be modified
- If attribute not part of a primary key nor of a foreign key
  - Usually causes no problems
- Updating a primary/foreign key
  - Similar issues as with Insert/Delete

The Relational Algebra

The Relational Algebra

- **Formal languages** for the relational model
- **Basis** of SQL
- **Models** aspects we learnt in the previous part in a formal way
  - Relations, constraints, operations, domains, etc.

The Relational Algebra

- **Relational algebra**
  - Basic set of operations for the relational model
- **Relational algebra expression**
  - Sequence of relational algebra operations that lead to *a new relation*

The Relational Algebra groups of operations

- **Relational Databases operations (unary operations (**operate on single relations**))**
  - SELECT, PROJECT, etc.
- **Set operations (binary operations (**operate on two tables**))**
  - UNION, INTERSECTION, etc.
- **Aggregate functions, additional types of set operations**

Unary Relational Operations: SELECT and PROJECT

- The SELECT Operation
  - Subset of the tuples from a relation that satisfies a selection condition:

$$\sigma_{<\text{selection condition}>}(R)$$

- Boolean expression contains clauses of the form <attribute name> <comparison op> <constant value>

  or

- <attribute name> <comparison op> <attribute name>

Unary Relational Operations: SELECT and PROJECT

Example:

$$\sigma_{Dno=4}(EMPLOYEE)$$
$$\sigma_{Salary>30000}(EMPLOYEE)$$

- R (here, EMPLOYEE) is a relation OR the result of a relational algebra expression
- The resulting relation of the SELECT operation is a **new relation** with the same attributes as the initial one

Unary Relational Operations: SELECT and PROJECT

- Comparison operators (=,<,≤,>,≥,≠) can only be applied on attributes whose domains are *ordered values*
- In case of un-ordered values, only the (=,≠) operators can be used (or some additional like SUBSTRING_OF)
- <selection condition> applied independently to each individual tuple *t* in *R*
  - If condition evaluates to TRUE, tuple selected
- Boolean conditions **AND**, **OR**, and **NOT**
- Example:

$$\sigma_{(\text{Dno}=4 \textbf{ AND } \text{Salary}>25000) \textbf{ OR } (\text{Dno}=5 \textbf{ AND } \text{Salary}>30000)}(\text{EMPLOYEE})$$

Unary Relational Operations: SELECT and PROJECT (cont'd.)

- SELECT operation commutative

$$\sigma_{<\text{cond1}>}(\sigma_{<\text{cond2}>}(R)) = \sigma_{<\text{cond2}>}(\sigma_{<\text{cond1}>}(R))$$

- **Cascade** SELECT operations into a single operation with **AND** condition

$$\sigma_{<\text{cond1}>}(\sigma_{<\text{cond2}>}(\ldots(\sigma_{<\text{cond}n>}(R))\ \ldots)) = \sigma_{<\text{cond1}>\ \textbf{AND}<\text{cond2}>\ \textbf{AND}\ldots\textbf{AND}\ <\text{cond}n>}(R)$$

The PROJECT Operation

- Selects columns from table and discards the other columns:

$$\pi_{<\text{attribute list}>}(R)$$

- **Degree**
  - Number of attributes in <attribute list>
- **Duplicate elimination**
  - Result of PROJECT operation is a set of distinct tuples

The PROJECT Operation

$$\pi_{\text{Sex, Salary}}(\text{EMPLOYEE})$$

- R (here, EMPLOYEE) is a relation OR the result of a relational algebra expression
- The resulting relation of the PROJECT operation is a **new relation** with ONLY those attributes specified in the <attribute_list>

The PROJECT Operation

- What can happen if the attribute list contains only non-key attributes?

The PROJECT Operation

- What can happen if the attribute list contains only non-key attributes?
  - Duplicates are possible but the PROJECT operation removes them (**duplicate elimination**)

The PROJECT Operation

- What can we say about *commutativity?*
- Under what circumstances, is the following expression correct?

$$\pi_{<\text{list1}>}\left(\pi_{<\text{list2}>}(R)\right) = \pi_{<\text{list1}>}(R)$$

Sequences of Operations and the RENAME Operation

- Single relational algebra expression(**In-line** expression):

$$\pi_{\text{Fname, Lname, Salary}}(\sigma_{\text{Dno}=5}(\text{EMPLOYEE}))$$

- Sequence of operations:

$$\text{DEP5\_EMPS} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEE})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Salary}}(\text{DEP5\_EMPS})$$

- **Rename** attributes in intermediate results
  - RENAME operation

$$\rho_{S(B1, B2, \ldots, Bn)}(R) \quad \text{or} \quad \rho_{S}(R) \quad \text{or} \quad \rho_{(B1, B2, \ldots, Bn)}(R)$$

Sequences of Operations and the RENAME Operation

- RENAME examples:

$$\text{TEMP} \leftarrow \sigma_{\text{Dno}=5}(\text{EMPLOYEE})$$
$$R(\text{First\_name}, \text{Last\_name}, \text{Salary}) \leftarrow \pi_{\text{Fname, Lname, Salary}}(\text{TEMP})$$

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

- If no name is specified, resulting relations do not have any names!

Relational Algebra Operations from Set Theory

- **UNION**, **INTERSECTION**, and **MINUS**
  - Merge the elements of two sets in various ways
  - Binary operations
  - Relations must have the same type of tuples (*type compatibility)*. Two relations $R(A1, A2, ..., An)$ and $S(B1, B2, ..., Bn)$ are said to be **type compatible**, if they have the same degree n and if $dom(Ai) = dom(Bi)$.

Relational Algebra Operations from Set Theory

- **UNION**
  - $R \cup S$
  - Includes all tuples that are either in *R* or in *S* or in both *R* and *S*
  - Duplicate tuples eliminated

$$DEP5\_EMPS \leftarrow \sigma_{Dno=5}(EMPLOYEE)$$
$$RESULT1 \leftarrow \pi_{Ssn}(DEP5\_EMPS)$$
$$RESULT2(Ssn) \leftarrow \pi_{Super\_ssn}(DEP5\_EMPS)$$
$$RESULT \leftarrow RESULT1 \cup RESULT2$$

retrieve the SSN of all employees who either work in dept 5 or directly supervise an employee who works in dept 5

RESULT1

Ssn
123456789
333445555
666884444
453453453

RESULT2

Ssn
333445555
888665555

RESULT

Ssn
123456789
333445555
666884444
453453453
888665555

Figure 6.3
Result of the UNION operation RESULT ← RESULT1 ∪ RESULT2.

Relational Algebra Operations from Set Theory

- **INTERSECTION**
  - $R \cap S$
  - Includes all tuples that are in both *R* and *S*
- **SET DIFFERENCE (or MINUS)**
  - $R - S$
  - Includes all tuples that are in *R* but not in *S*
- **Exercise:** Find **INTERSECTION** in terms of union and set difference.
  - $R \cap S = ((R \cup S)-(R-S))-(S-R)$

Relational Algebra Operations from Set Theory

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

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

- **CARTESIAN PRODUCT**
  - **CROSS PRODUCT** or **CROSS JOIN** (R×S)
  - $R(A_1, A_2, ..., A_n) \times S(B_1, B_2, ..., B_m)$
    - $= Q(A_1, A_2, ..., A_n, B_1, B_2, ..., B_m)$
  - The resulting relation Q has one tuple for each combination of tuples—one from R and one from S. Hence, if R has n_R tuples (denoted as $|R| = n_R$), and S has n_S tuples, then $R \times S$ will have $n_R * n_S$ tuples.

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

$$\text{FEMALE\_EMPS} \leftarrow \sigma_{\text{Sex}='F'}(\text{EMPLOYEE})$$
$$\text{EMPNAMES} \leftarrow \pi_{\text{Fname, Lname, Ssn}}(\text{FEMALE\_EMPS})$$
$$\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$$
$$\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{Ssn}=\text{Essn}}(\text{EMP\_DEPENDENTS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Dependent\_name}}(\text{ACTUAL\_DEPENDENTS})$$

retrieve a list of names of each female employee's dependents

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

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

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

EMPNAMES

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

$$\text{FEMALE\_EMPS} \leftarrow \sigma_{\text{Sex='F'}}(\text{EMPLOYEE})$$
$$\text{EMPNAMES} \leftarrow \pi_{\text{Fname, Lname, Ssn}}(\text{FEMALE\_EMPS})$$
$$\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$$
$$\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{Ssn=Essn}}(\text{EMP\_DEPENDENTS})$$

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Joyce	English	453453453	333445555	Alice	F	1986-04-05	...
Joyce	English	453453453	333445555	Theodore	M	1983-10-25	...
Joyce	English	453453453	333445555	Joy	F	1958-05-03	...
Joyce	English	453453453	987654321	Abner	M	1942-02-28	...
Joyce	English	453453453	123456789	Michael	M	1988-01-04	...
Joyce	English	453453453	123456789	Alice	F	1988-12-30	...
Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	...

The CARTESIAN PRODUCT (CROSS PRODUCT) Operation

$$\text{FEMALE\_EMPS} \leftarrow \sigma_{\text{Sex='F'}}(\text{EMPLOYEE})$$
$$\text{EMPNAMES} \leftarrow \pi_{\text{Fname, Lname, Ssn}}(\text{FEMALE\_EMPS})$$
$$\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$$
$$\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{Ssn=Essn}}(\text{EMP\_DEPENDENTS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Dependent\_name}}(\text{ACTUAL\_DEPENDENTS})$$

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULT

Fname	Lname	Dependent_name
Jennifer	Wallace	Abner

Binary Relational Operations: JOIN

- The **JOIN** Operation
  - Denoted by ⋈
  - Combine related tuples from two relations into single "longer" tuples
  - General join condition of the form <condition> **AND** <condition> **AND**...**AND** <condition>
  - Example:

retrieve the name of the manager of each department.

$$\text{DEPT\_MGR} \leftarrow \text{DEPARTMENT} \bowtie_{\text{Mgr\_ssn=Ssn}} \text{EMPLOYEE}$$
$$\text{RESULT} \leftarrow \pi_{\text{Dname, Lname, Fname}}(\text{DEPT\_MGR})$$

Binary Relational Operations:JOIN

Difference between JOIN and CARTESIAN PRODUCT:

- Both result in relations of n+m attributes (n from relation Q and m from R)
- But: In JOIN, only combinations of tuples satisfying the joint condition appear in the result

$$\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$$
$$\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{Ssn}=\text{Essn}}(\text{EMP\_DEPENDENTS})$$

Is equivalent to:

$$\text{ACTUAL\_DEPENDENTS} \leftarrow \text{EMPNAMES} \bowtie_{\text{Ssn}=\text{Essn}} \text{DEPENDENT}$$

- Note that combination is done on a foreign and a primary key!

Binary Relational Operations: JOIN and DIVISION

- **THETA** JOIN
  - Each <condition> of the form $A_i \, \theta \, B_j$
  - A_i is an attribute of R
  - B_j is an attribute of S
  - A_i and B_j have the same domain
  - θ (theta) is one of the comparison operators:
    - {=, <, ≤, >, ≥, ≠}

Tuples whose join attributes are NULL or for which, the join condition is FALSE, do not appear in the result

Variations of JOIN: The EQUIJOIN and NATURAL JOIN

- **EQUIJOIN**
  - Only = comparison operator used
- **NATURAL JOIN**
  - Denoted by *
  - Removes second (superfluous) attribute in an EQUIJOIN condition

  Always have one or more pairs of attributes that have identical names in every tuple
  - If they are not identical, we can rename:

$$\text{PROJ\_DEPT} \leftarrow \text{PROJECT} * \rho_{(\text{Dname, Dnum, Mgr\_ssn, Mgr\_start\_date})}(\text{DEPARTMENT})$$

  - The attribute **Dnum** is called the join attribute for the NATURAL JOIN operation, because it is the only attribute with the same name in both relations

The DIVISION Operation

- Denoted by ÷
- Apply to relations: $R \div S$
  - Attributes of *S* are a subset of the attributes of *R*
  - The result consists of the restrictions of tuples in *R* to the attribute names unique to *R*, i.e., in the header of *R* but not in the header of *S*, for which it holds that all their combinations with tuples in *S* are present in *R*

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

$T \leftarrow R \div S$

T

B
b1
b4

The DIVISION Operation

Example: retrieve the names of employees who work on all the projects that 'John Smith' works on

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

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

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

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

Operations of Relational Algebra

Table 6.1 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 $R_1 \bowtie_{\text{(<join attributes 1>), (<join attributes 2>)}} R_2$
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 $R_1 *_{\text{(<join attributes 1>), (<join attributes 2>)}} R_2$ OR $R_1 * R_2$

Operations of Relational Algebra

Table 6.1 Operations of Relational Algebra

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 $Z = X \cup Y$.	$R_1(Z) \div R_2(Y)$

Additional Relational Operations

- **Generalized projection**
  - Allows functions of attributes to be included in the projection list
  - Given: EMPLOYEE (Ssn, Salary, Deduction, Years_service)
    - Calculate:
      - Net Salary = Salary – Deduction,
      - Bonus = 2000 * Years_service, and
      - Tax = 0.25 * Salary.

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

Additional Relational Operations

- **Aggregate functions** and **grouping**
  - Common functions applied to collections of numeric values
  - Include SUM, AVERAGE, MAXIMUM, and MINIMUM

Additional Relational Operations

- Group tuples by the value of some of their attributes
  - Apply aggregate function independently to each group

$$<\text{grouping attributes}>\ \Im\ <\text{function list}>\ (R)$$

  - where <grouping attributes> is a list of attributes of the relation specified in *R*, and <function list> is a list of (<function> <attribute>) pairs. In each such pair, <function> is one of the allowed functions—such as SUM, AVERAGE, MAXIMUM, MINIMUM,COUNT—and <attribute> is an attribute of the relation specified by *R*.

Figure 6.10
The aggregate function operation.

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

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

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

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

Recursive Closure Operations

- Operation applied to a **recursive relationship** between tuples of same type
  - Ex: Find all employees supervised by “James Borg”

$$\text{BORG\_SSN} \leftarrow \pi_{\text{Ssn}}(\sigma_{\text{Fname='James' AND Lname='Borg'}}(\text{EMPLOYEE}))$$
$$\text{SUPERVISION}(\text{Ssn1}, \text{Ssn2}) \leftarrow \pi_{\text{Ssn,Super\_ssn}}(\text{EMPLOYEE})$$
$$\text{RESULT1}(\text{Ssn}) \leftarrow \pi_{\text{Ssn1}}(\text{SUPERVISION} \bowtie_{\text{Ssn2=Ssn}} \text{BORG\_SSN})$$

SUPERVISION

(Borg's Ssn is 888665555)

(Ssn) (Super_ssn)

Ssn1	Ssn2
123456789	333445555
333445555	888665555
999887777	987654321
987654321	888665555
666884444	333445555
453453453	333445555
987987987	987654321
888665555	null

RESULT1

Ssn
333445555
987654321

(Supervised by Borg)

OUTER JOIN Operations

- **Outer joins**
  - Keep all tuples in *R*, or all those in *S*, or all those in both relations regardless of whether or not they have matching tuples in the other relation
  - **Types**
    - **LEFT OUTER JOIN, RIGHT OUTER JOIN, FULL OUTER JOIN**
  - Example:

$$TEMP \leftarrow (EMPLOYEE \bowtie_{Ssn=Mgr\_ssn} DEPARTMENT)$$

$$RESULT \leftarrow \pi_{Fname,\ Minit,\ Lname,\ Dname}(TEMP)$$

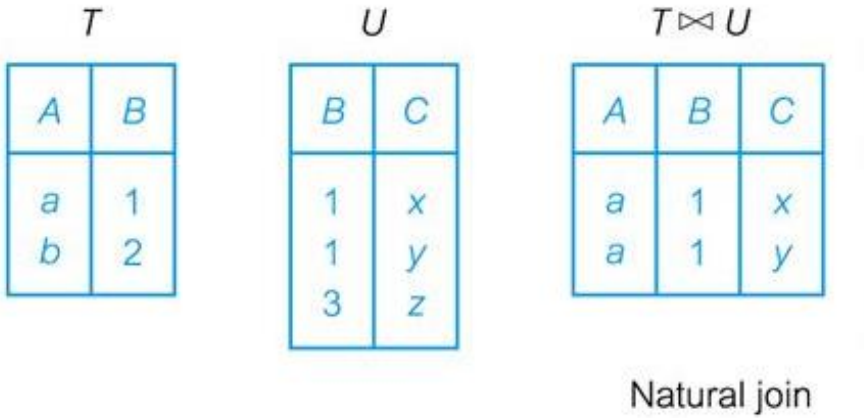
RESULT

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

Important!

- **The cartisian product** gives M*N tuples (i.e. finds all possible combinations). There is no condition in cartisian product.
- **The inner join** will not bring nulls or includes tuples that don’t have match.
- A **Natural Join** is where 2 tables are joined on the basis of all common columns while **Inner Join** is where 2 tables are joined on the basis of common columns mentioned in the ON clause
- **The outer join** includes all from left (**left outer**) or all from right (**right outer**) or both (**full outer**)

Important!



T

A	B
a	1
b	2

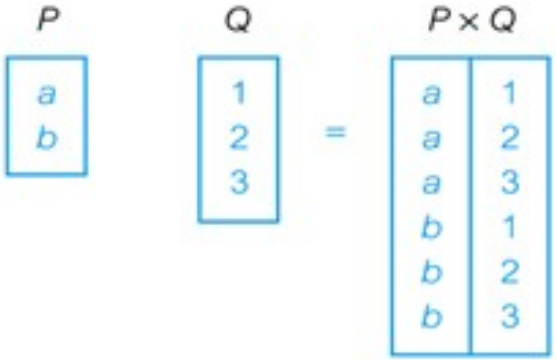
U

B	C
1	x
1	y
3	z

$T \bowtie U$

A	B	C
a	1	x
a	1	y

Natural join



P

a
b

Q

1
2
3

$=$

$P \times Q$

a	1
a	2
a	3
b	1
b	2
b	3

Important!

R

ColA	ColB
A	1
B	2
D	3
F	4
E	5

S

SColA	SColB
A	1
C	2
D	3
E	4

R LEFT OUTER JOIN R.ColA = S.SColA S

A	1	A	1
D	3	D	3
E	5	E	4
B	2	-	-
F	4	-	-

R RIGHT OUTER JOIN R.ColA = S.SColA S

A	1	A	1
D	3	D	3
E	5	E	4
-	-	C	2

Important!

R

ColA	ColB
A	1
B	2
D	3
F	4
E	5

S

SColA	SColB
A	1
C	2
D	3
E	4

R FULL OUTER JOIN S R.ColA = S.SColA

A	1	A	1
D	3	D	3
E	5	E	4
B	2	-	-
F	4	-	-
-	-	C	2

Examples of Queries in Relational Algebra

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

Examples of Queries in Relational Algebra

Query 1. Retrieve the name and address of all employees who work for the 'Research' department.

Query 2. 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.

Query 3. Find the names of employees who work on *all* the projects controlled by department number 5.

Query 6. Retrieve the names of employees who have no dependents.

Query 7. List the names of managers who have at least one dependent.

Examples of Queries in Relational Algebra

Query 1. Retrieve the name and address of all employees who work for the 'Research' department.

$$\text{RESEARCH\_DEPT} \leftarrow \sigma_{\text{Dname='Research'}}(\text{DEPARTMENT})$$
$$\text{RESEARCH\_EMPS} \leftarrow (\text{RESEARCH\_DEPT} \bowtie_{\text{Dnumber=Dno}} \text{EMPLOYEE})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Address}}(\text{RESEARCH\_EMPS})$$

As a single in-line expression, this query becomes:

$$\pi_{\text{Fname, Lname, Address}} (\sigma_{\text{Dname='Research'}}(\text{DEPARTMENT} \bowtie_{\text{Dnumber=Dno}} (\text{EMPLOYEE}))$$

Examples of Queries in Relational Algebra

Query 2. 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.

$$\text{STAFFORD\_PROJS} \leftarrow \sigma_{\text{Plocation}=\text{'Stafford'}}(\text{PROJECT})$$
$$\text{CONTR\_DEPTS} \leftarrow (\text{STAFFORD\_PROJS} \bowtie_{\text{Dnum}=\text{Dnumber}} \text{DEPARTMENT})$$
$$\text{PROJ\_DEPT\_MGRS} \leftarrow (\text{CONTR\_DEPTS} \bowtie_{\text{Mgr\_ssn}=\text{Ssn}} \text{EMPLOYEE})$$
$$\text{RESULT} \leftarrow \pi_{\text{Pnumber, Dnum, Lname, Address, Bdate}}(\text{PROJ\_DEPT\_MGRS})$$

Query 3. Find the names of employees who work on *all* the projects controlled by department number 5.

$$\text{DEPT5\_PROJS} \leftarrow \rho_{(\text{Pno})}(\pi_{\text{Pnumber}}(\sigma_{\text{Dnum}=5}(\text{PROJECT})))$$
$$\text{EMP\_PROJ} \leftarrow \rho_{(\text{Ssn, Pno})}(\pi_{\text{Essn, Pno}}(\text{WORKS\_ON}))$$
$$\text{RESULT\_EMP\_SSNS} \leftarrow \text{EMP\_PROJ} \div \text{DEPT5\_PROJS}$$
$$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}}(\text{RESULT\_EMP\_SSNS} * \text{EMPLOYEE})$$

Examples of Queries in Relational Algebra

Query 6. Retrieve the names of employees who have no dependents.

This is an example of the type of query that uses the MINUS (SET DIFFERENCE) operation.

$$\text{ALL\_EMPS} \leftarrow \pi_{\text{Ssn}}(\text{EMPLOYEE})$$
$$\text{EMPS\_WITH\_DEPS(Ssn)} \leftarrow \pi_{\text{Essn}}(\text{DEPENDENT})$$
$$\text{EMPS\_WITHOUT\_DEPS} \leftarrow (\text{ALL\_EMPS} - \text{EMPS\_WITH\_DEPS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}}(\text{EMPS\_WITHOUT\_DEPS} * \text{EMPLOYEE})$$

Query 7. List the names of managers who have at least one dependent.

$$\text{MGRS(Ssn)} \leftarrow \pi_{\text{Mgr\_ssn}}(\text{DEPARTMENT})$$
$$\text{EMPS\_WITH\_DEPS(Ssn)} \leftarrow \pi_{\text{Essn}}(\text{DEPENDENT})$$
$$\text{MGRS\_WITH\_DEPS} \leftarrow (\text{MGRS} \cap \text{EMPS\_WITH\_DEPS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Lname, Fname}}(\text{MGRS\_WITH\_DEPS} * \text{EMPLOYEE})$$