CS354: Database

Relational Query Languages

- Manipulation and retrieval of data from a database
- Relational Calculus: Declarative
  - describe what a user wants rather than how to compute it
- Relational Algebra: Procedural
  - operational and useful for representing execution plans

Query vs Programming Languages

- Query languages:
  - Are not “Turing complete” (some languages derived from SQL are, but not all)
  - Are not intended to be used for complex calculations
  - Supports easy, efficient access to large data sets

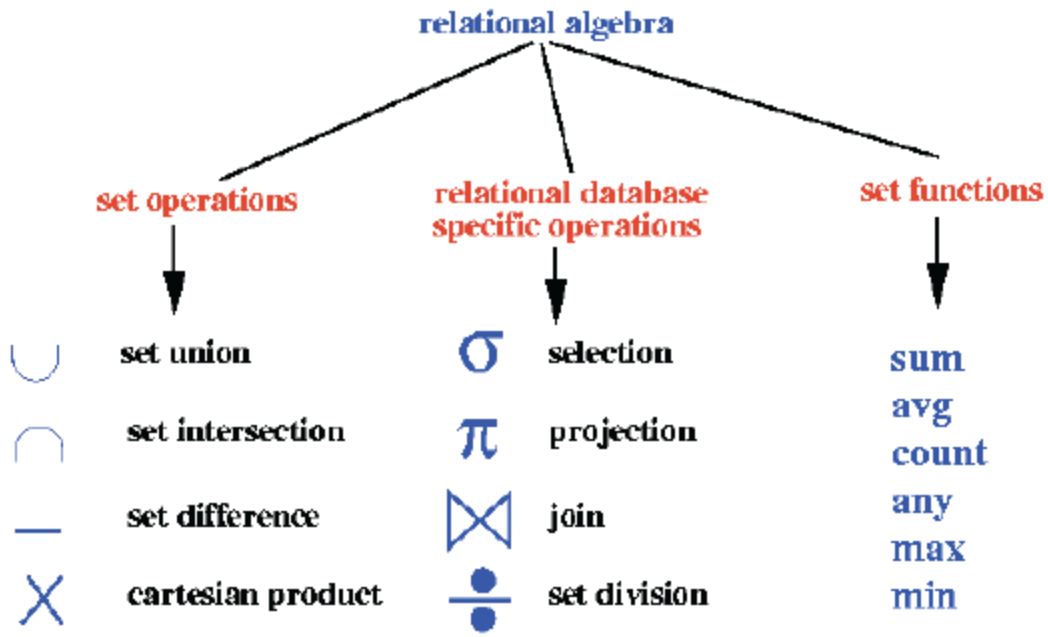
Relational Algebra Overview

- Formal foundation for relational model operations
  - Operators do the most common things needed for relations in a database
- Basis for implementing and optimizing queries in RDBMS
- Basis for practical query languages such as SQL

Relational Algebra Terms

- **Relational algebra** is a mathematical language with a basic set of operations for manipulating relations
- A **relational algebra operation** operates on one or more relations and results in a new relation, which can be further manipulated using operations of the same algebra
- A **relational algebra expression** is a sequence of relational algebra operations

Relational Algebra Language



Set Operations (Set Theory Recap)

- Union: set of all distinct elements in the collection of sets

$$A = \{i, j, k\}, B = \{k, x, y\}, A \cup B = \{i, j, k, x, y\}$$

- Intersection: set of elements that belong in both sets

$$A = \{i, j, k\}, B = \{k, x, y\}, A \cap B = \{k\}$$

- Difference: the set of elements in A, but not in B

$$A = \{i, j, k\}, B = \{k, x, y\}, A - B = \{i, j\}$$

- Cartesian product: set of all ordered pairs from each set

$$A = \{i, j, k\}, B = \{k, x, y\}, A \times B = \{(i, k), (i, x), \cdots, (k, y)\}$$

Set Operations Notes ($\cup, \cap, -$)

- The two operands must be type compatible
  - A and B must have the same number of attributes and the domains of corresponding attributes must be compatible (i.e., $dom(A_i) = dom(B_i)$)
  - Resulting relation has the same attribute names as A
- Union and intersection are commutative and associative operations $A \cup B = B \cup A, A \cup (B \cup C) = (A \cup B) \cup C$
- Minus is not commutative $A - B \neq B - A$

Example: Union

STUDENT

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FN	LN
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

STUDENT ∪ INSTRUCTOR

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

Example: Intersection

STUDENT

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FN	LN
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

STUDENT ∩ INSTRUCTOR

FN	LN
Susan	Yao
Ramesh	Shah

Example: Difference

STUDENT

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FN	LN
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

STUDENT – INSTRUCTOR

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

Example: Cartesian Product

PERSON

SSN	Name
932-45-1789	Yao
123-12-3645	Ramesh

DEPENDENT

depSSN	depName
934-55-1234	Helen
936-58-1578	John

PERSON × DEPENDENT

SSN	Name	depSSN	depName
932-45-1789	Yao	934-55-1234	Helen
123-12-3645	Ramesh	936-58-1578	John
932-45-1789	Yao	936-58-1578	John
123-12-3645	Ramesh	934-55-1234	Helen

Selection Operation

- Select a subset of tuples from a relation that satisfy a selection condition
  - Selection condition is a boolean expression containing clauses in the form <attribute name> <comparison op> <constant value / attribute name>
- Notation: $\sigma_C(R)$
  - C is condition that refers to attributes of R
  - Outputs rows of R that satisfy C

Selection Operation Properties

- Operation produces a relation that has the same schema as R
- Operation is commutative

$$\sigma_{<C1>}(\sigma_{<C2>}(R)) = \sigma_{<C2>}(\sigma_{<C1>}(R))$$

- Cascaded operation may be applied in any order

$$\sigma_{<C1>}(\sigma_{<C2>}(\sigma_{<C3>}(R))) = \sigma_{<C2>}(\sigma_{<C3>}(\sigma_{<C1>}(R)))$$

- Cascaded operation may be replaced by a single selection with a conjunction of all conditions

$$\sigma_{<C1>}(\sigma_{<C2>}(\sigma_{<C3>}(R))) = \sigma_{<C1>AND<C2>AND<C3>}(R)$$

Example: Selection

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

$$\sigma_{(\mathrm{dno}=4)}(\mathrm{employee})$$

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000

Example: Selection (2)

- Retrieve all employees earning more than 50,000
- Retrieve all female employees
- Retrieve all employees working for department number 4 and earning more than 50,000

Projection Operation

- Select certain columns from the tables and discards the other columns
- Notation: $\pi_{<\text{attribute list}>}(R)$
  - Outputs only the columns listed in the attribute list
  - Removes any duplicate tuples because the output is a set and sets to not have duplicate elements

Projection Properties

- Number of tuples after projection operator is always less or equal to the number of tuples in R
- If the list of attributes includes a key of R, then the number of tuples is equal to the number of tuples in R
- Cascade of two projection operators will be equivalent if the second attribute list contains the attributes in the first list

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

Example: Projection

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

$$\pi_{\mathrm{SSN}}(\mathrm{employee})$$

Example: Projection (2)

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

$$\pi_{\text{sex}}(\text{employee})$$

Example: Projection (3)

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

$$\pi_{\text{LName, FName, Salary}}(\text{employee})$$

Renaming Operation

- Name conflicts can arise in some situations
- Does not change the instance, only the schema!
- Notation: $\rho_{A_1, A_2, \cdots, A_n}(R)$
  - Rename the columns of R to the fields provided
- Example:

$$\rho_{\mathrm{DEPT5}}(\sigma_{(\mathrm{DNO}=5)}(\mathrm{employee}))$$

Join (Theta Join) Operation

- A sequence of a cartesian product followed by a selection
- Notation: $R \bowtie_C S = \sigma_C(R \times S)$
  - C can be any boolean-valued condition with each condition of the form: $x\ \theta\ y$
  - x, y are attributes or constants
  - θ operation is one of the following relational operators: ==, !=, <, <=, >=, >
  - Result may have less tuples than cross product

Example: Join

EMPLOYEE

FName	LName	DNo	Salary
John	Smith	5	30000
Frank	Wong	5	40000
Alicia	Zelaya	4	25000
Jennif	Wallace	4	43000
James	Borg	1	55000

DEPARTMENT

DName	DNumber
Research	5
Administration	4
Headquarters	1

$$\text{EMPLOYEE} \bowtie_{\text{DNo}=\text{DNumber}} \text{DEPARTMENT}$$

FName	LName	DNo	Salary	DName	DNumber
John	Smith	5	30000	Research	5
Frank	Wong	5	40000	Research	5
Alicia	Zelaya	4	25000	Administration	4
Jennif	Wallace	4	43000	Administration	4
James	Borg	1	55000	Headquarters	1

Example: Join (2)

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	Walace	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

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

Variation of Joins

- EQUIJOIN
  - Join conditions with equality comparisons only (x ==y)
  - One or more pairs of attributes have identical values in every tuple
- NATURAL JOIN (*)
  - EQUIJOIN operation followed by a projection operation that removes one of the duplicate attributes
  - Requires the two join attributes to have the same name in both relations, otherwise renaming operation is necessary first

Example: Natural Join

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

$$\text{DEPARTMENT} * \text{DEPT\_LOCATIONS}$$

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

Outer Join Operation

- Does not require each record in the two joined tables to have matching values
- Retains records from one or both tables even if no matches are found
- 3 flavors of outer joins
  - Left outer-join
  - Right outer-join
  - Full outer-join

Left Outer-Join: $A \ltimes B$

- Keeps every tuple in the first or left relation A
- If no matching tuple is found in the second relation B, then it is combined with a row of NULL values (filled or padded with NULL values)

A

X	Y
1	7
2	5
3	4

B

U	V
1	8
2	3
4	9

$A \ltimes_{A.X=B.U} B$

X	Y	U	V
1	7	1	8
2	5	2	3
3	4	NULL	NULL

Right Outer-Join: $A \rtimes B$

- Keeps every tuple in the second or right relation B
- If no matching tuple is found in the left relation A, then it is combined with a row of NULL values (filled or padded with NULL values)

A

X	Y
1	7
2	5
3	4

B

U	V
1	8
2	3
4	9

$A \rtimes_{A.X=B.U} B$

X	Y	U	V
1	7	1	8
2	5	2	3
NULL	NULL	4	9

Full Outer-Join $A \mathbin{⟗} B$

- Keeps all tuples from both the left and right relations
- Unmatched tuple in either table is combined with a row of NULL values

A

X	Y
1	7
2	5
3	4

B

U	V
1	8
2	3
4	9

$A \mathbin{⟗}_{A.X=B.U} B$

X	Y	U	V
1	7	1	8
2	5	2	3
3	4	NULL	NULL
NULL	NULL	4	9

Division Operation

- Suppose R_1(A,B) and R_2(B), the output contains all A-tuples such that for every B-tuple in R_2, there exists an (A,B) tuple in R1
- Notation: $R_1 \div R_2$
- Example: Find first name and last name of all employees who work on all projects that "John Smith" works on

Example: Division

A

S	P
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

$B1$

P
p2

$B2$

P
p2
p4

$B3$

P
p1
p2
p4

$A \div B1$

S
s1
s2
s3
s4

$A \div B2$

S
s1
s4

$A \div B3$

S
s1

Division as a Sequence of Other Operations

- Construct a "minimum qualifying" relation which contains the tuples that is necessary to be selected by the query. Suppose $R_1(A, B)$ and $R_2(B)$, then "qualify" relation is constructed as: $\mathbf{qualify} = \pi_A(R_1) \times R_2$
- Compute the difference between "qualify" and R_1. This gets all the disqualified tuples: $\mathbf{unqualify} = \mathbf{qualify} - R_1$
- Last step is to obtain the qualified tuples using another set difference: $\pi_A(R_1) - \pi_A(\mathbf{unqualify})$

Relational Algebra: Recap

- Relational Algebra
  - Query language for relations
- Basic Operations
  - Set Operations
  - Database-specific operations (selection, projection, join, division)