DATA BASE SYSTEMS

LECTURE 4

PROPOSED BY

DR: ALSHAIMAA MOSTAFA MOHAMMED

CHAPTER 5

RELATIONAL ALGEBRA AND CALCULUS

Introduction

- The relational model is an abstract Theory of data that is based on the mathematical concept of relations.
- Information is stored in tabular form or in tables.
- So, it consists of collection of tables.
- Use mathematical terms in relational model.

Introduction

Two mathematical Query Languages form the basis for "real" languages (e.g. SQL), and for implementation:

- **Relational Algebra**
- **Relational Calculus**
- Understanding Algebra & Calculus is key to understanding SQL, query processing!

Introduction

- **Relational Algebra:**
  - Specifies the operations to be performed on existing relations to derive result relations.
  - A sequence of relational algebraic operations forms a relational algebraic expression.
  - The result of the relational algebraic expression is also a relation.
- **Relational Calculus:** A formal query language where the queries are expressed as variables and formulas on these variables.

Relational Algebra

- Relational algebra is a procedural query language.
- It uses a collection of operators to compose the queries.
- Every operator in the algebra accepts either one or two relation instances as arguments and output a resultant relation instance.
- Operators can be composed easily to construct complex queries.
- Each relational algebra query describes a step-by-step procedure for computing the desired answer, based on the order in which the operators are applied in the query.

Relational Algebra

- The relational algebra uses various logical connectives [∧(and), ∨(or), ¬(not)] and comparison operators (<, <=, =, ≠, >=, >) to construct composite and more complex queries.

Operations in Relational Algebra

- The relational algebraic operations can be divided into basic set-oriented operations (union, intersection, set difference, and cartesian product) and relational-oriented operations (selection, projection, division and joins).

The union operation

- The union operation is a binary operation that is used to find union of relations.
- Here relations are considered as sets. So, duplicate values are eliminated.
- It is denoted by (∪).
- There are two necessary conditions for union operation:

(*I*) both the relations have same number of attributes.

(*Ii*) data types of their corresponding attributes must be same.

The UNION operation

- Two relations are said to be union compatible if they follow the above two conditions.
- ***Ex.*** If you want to find the names of all employees and names of all students together
- Then the query is
- π name (employee) ∪ πname (student)

Employee

EID	Name	Salary
1E	John	10,000
2E	Ramesh	5,000
3E	Smith	8,000
4E	Jack	6,000
5E	Nile	15,000

Student

SID	Name	Fees
1S	Smith	1,000
2S	Vijay	950
3S	Gaurav	2,000
4S	Nile	1,500
5S	John	950

Name
John
Ramesh
Smith
Jack
Nile
Vijay
Gaurav

Set Intersection operation

- Used to find common tuples between two relations.
- It is denoted by (∩).
- If you want to find all the employees from relation employee those are also students.

Employee

EID	Name	Salary
1E	John	10,000
2E	Ramesh	5,000
3E	Smith	8,000
4E	Jack	6,000
5E	Nile	15,000

Student

SID	Name	Fees
1S	Smith	1,000
2S	Vijay	950
3S	Gaurav	2,000
4S	Nile	1,500
5S	John	950

Set Difference operation

- Is a binary operation which is used to find tuples that are present in one relation but not in other relation.
- It is denoted by (—).
- It removes the common tuples of two relations and produce a new relation having rest of the tupels of first relation.
- ***Ex.*** If you want the names of those employees that are not students, then the query, is
- Πname (employee) — πname (student)

Employee

EID	Name	Salary
1E	John	10,000
2E	Ramesh	5,000
3E	Smith	8,000
4E	Jack	6,000
5E	Nile	15,000

Student

SID	Name	Fees
1S	Smith	1,000
2S	Vijay	950
3S	Gaurav	2,000
4S	Nile	1,500
5S	John	950

Cartesian Product operation

- Binary operation which is used to combine information of any two relations.
- Suppose a relation R1 is having *m* tuples and other relation R2 is having *n* tuples then R1 × R2 has *m* × *n* tuples.
- It is denoted by (x).

Employee

EID	Name	JID
1E	Manoj	1J
2E	Deepak	2J
3E	Vinay	1J

Job

JID	Job
1J	Tester
2J	Manager

FIGURE 5.5. Employee and Job relation.

EID	Name	Employee JID	Job JID	Job
1E	Manoj	1J	1J	Tester
1E	Manoj	1J	2J	Manager
2E	Deepak	2J	1J	Tester
2E	Deepak	2J	2J	Manager
3E	Vinay	1J	1J	Tester
3E	Vinay	1J	2J	Manager

FIGURE 5.6. Result of Cartesian product operation.

Selection or Restriction operation

- The selection operation is a unary operation.
- This is used to find horizontal subset of relation or tuples of relation.
- It is denoted by sigma (σ)**.**
- ***Ex.*** If you want all the employees having salary more than 9,000 from relation employee. The query, is
- **σ salary > 9,000 (employee)**

Employee

EID	Name	Salary
1E	John	10,000
2E	Ramesh	5,000
3E	Smith	8,000
4E	Jack	6,000
5E	Nile	15,000

Student

SID	Name	Fees
1S	Smith	1,000
2S	Vijay	950
3S	Gaurav	2,000
4S	Nile	1,500
5S	John	950

Projection operation

- Unary operation which applies only on a single relation at a time.
- Project operation is used to select vertical subset of relation (*i.E.,* Columns of table).
- It is denoted by pi (π).
- If you want all the names of employees and their salary from relation employee. Then query, is
- **Π name, salary (employee)**

Employee

EID	Name	Salary
1E	John	10,000
2E	Ramesh	5,000
3E	Smith	8,000
4E	Jack	6,000
5E	Nile	15,000

Student

SID	Name	Fees
1S	Smith	1,000
2S	Vijay	950
3S	Gaurav	2,000
4S	Nile	1,500
5S	John	950

Division operation

- Division operation is useful in special kind of queries that include the phrase “for all”.
- It is denoted by (÷).
- It is like the inverse of cartesian product.
- Let R be the relation with schema R and S be the relation with schema S. Let S ⊆ R. Then tuple t is in $R \div S$ if and only if:

(I) t is in $\pi_{r-s}(r)$

(Ii) if tuple is present in the cartesian product of S and R.

A

X	Y
X1	Y1
X1	Y3
X1	Y2
X4	Y
X5	Y5
X2	Y3
X3	Y4
X4	Y1

B1

Y
Y1
Y3

B2

Y
Y1
Y5

B3

Y
Y5
Y2
Y4

A ÷ B1 gives

X
X1
X4
X2

A ÷ B2 gives

X
X1
X4
X5

A ÷ B3 gives

X
X5
X1
X3

FIGURE 5.10. Result of division operation.

Natural-join operation

- Used to join two relations having any number of attributes. It is denoted by symbol (∞).
- it also optimize the query as cartesian product gives unnecessary results and set-union and set - intersection operations are applicable only on those relations that have equal number of attributes with same data-type.
- Find the names of all employees from relation employee with their respective department names.

Employee

EID	Name	Salary	Dept-ID
1	Amit	5,000	10
2	Sachin	8,000	20
3	Vinay	2,000	20
4	Vivek	6,000	10

Department

Dept_ID	Dept_Name
10	Sales
20	Purchase

$$\pi_{\text{Name, Dept\_name}} \left[\begin{array}{c} \text{Employee} \bowtie \text{Department} \\ \text{Employee.EID} = \text{Department.Dept\_ID} \end{array} \right]$$

Name	Dept-Name
Amit	Sales
Sachin	Purchase
Vinay	Purchase
Vivek	Sale

Outer-join operation

- Is an extension of natural join operations.
- It deals with the missing information caused by natural join operation.
- Suppose you need all information of all employees and all students in a single relation.
- There are **three** types of outer joins:
- ***Left outer join***
- ***Right outer join***
- ***Full outer join***

Left Outer-join

- It is used to take all tuples of relation that are on the left side whether they are matching with tuples of right-side relation or not.
- Employee relation is at left side so table consists all information of employee relation but still missing information about Vijay and Gaurav.

(Employee ⟕ Student) gives

EID	SID	Name	Salary	Fees
1E	5S	John	10,000	950
2E	NULL	Ramesh	5,000	NULL
3E	1S	Smith	8,000	1,000
4E	NULL	Jack	6,000	NULL
5E	4S	Nile	15,000	1,500

FIGURE 5.14. Result of left outer join.

Right Outer-join

- It is used to take all tuples of relation that are on the right side whether they are matching with tuples of left side relation or not.
- Student relation is at right side so table consists all information of student relation but still missing information about Ramesh and Jack.

(Employee ⋉ Student)gives

EID	SID	Name	Salary	Fees
3E	1S	Smith	8,000	1,000
NULL	2S	Vijay	NULL	950
NULL	3S	Gaurav	NULL	2,000
5E	4S	Nile	15,000	1,500
1E	5S	John	10,000	950

FIGURE 5.15. Result of right outer join.

Full Outer-join

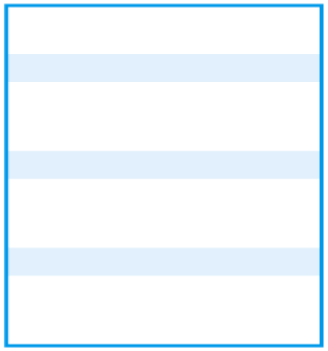
- It is used to take all tuples from left and right relation whether they match with each other or did not match.
- Table consist all information of employee and student relation.
- Here, no information is missing.

(Employee ⟗ Student) gives

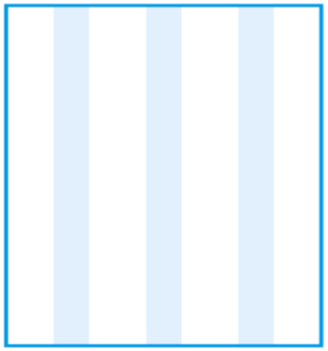
EID	SID	Name	Salary	Fees
1E	5S	John	10,000	950
2E	NULL	Ramesh	5,000	NULL
3E	1S	Smith	8,000	1,000
4E	NULL	Jack	6,000	NULL
5E	4S	Nile	15,000	1,500
NULL	2S	Vijay	NULL	1,000
NULL	3S	Gaurav	NULL	2,000

FIGURE 5.16. Result of full outer join.

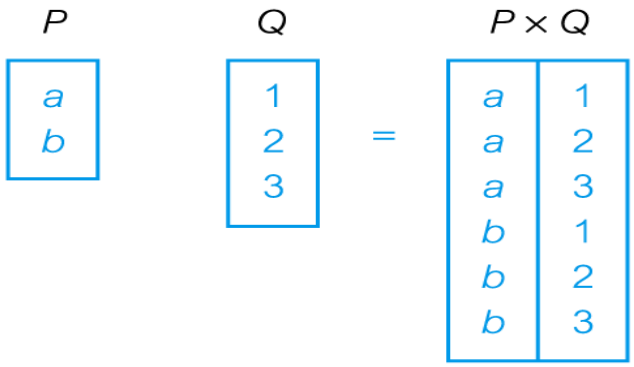
RELATIONAL ALGEBRA OPERATIONS



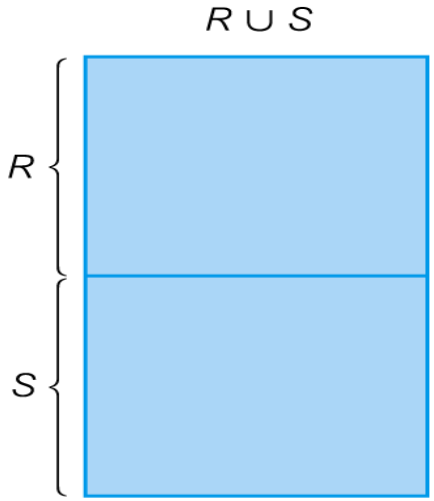
(a) Selection



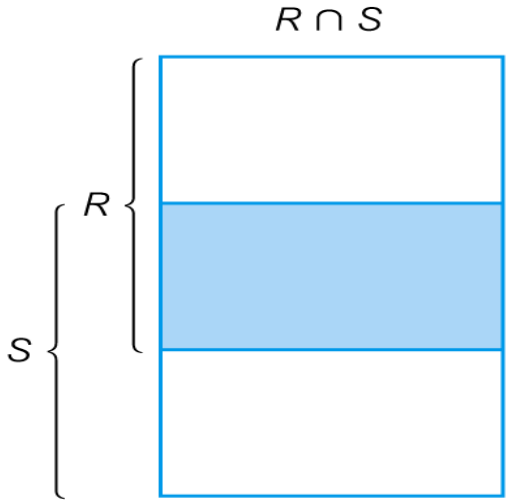
(b) Projection



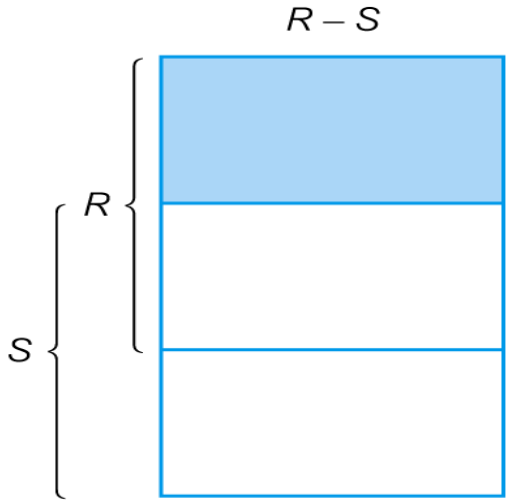
(c) Cartesian product



(d) Union

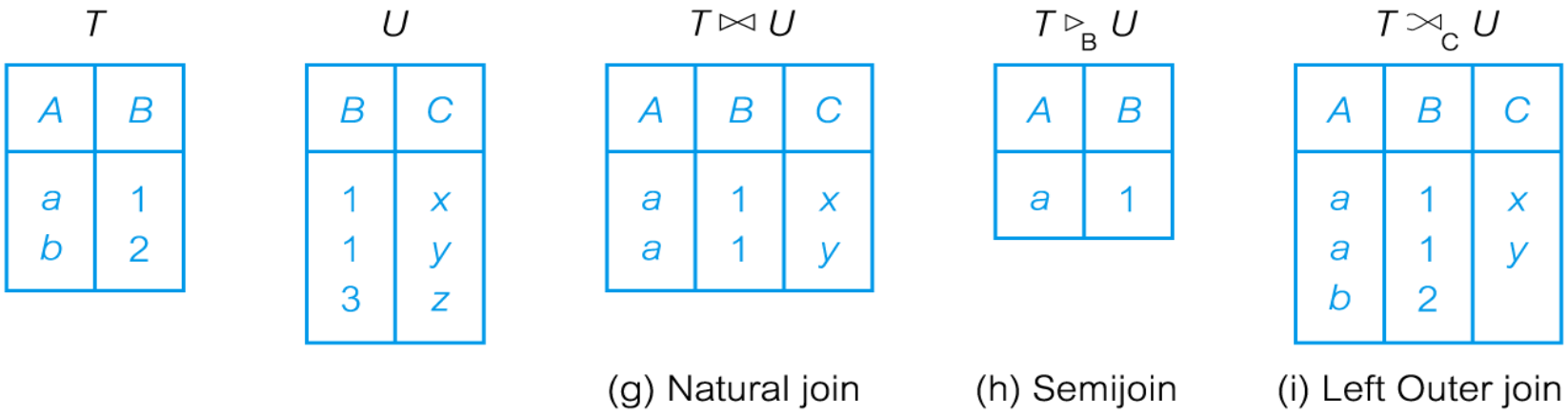


(e) Intersection



(f) Set difference

RELATIONAL ALGEBRA OPERATIONS



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*

(g) Natural join

$T \triangleright_B U$

A	*B*
a	1

(h) Semijoin

$T \bowtie_C U$

A	*B*	*C*
a	1	*x*
a	1	*y*
b	2	

(i) Left Outer join

R

Remainder

S

$R \div S$

(j) Divis on (shaded area)

V

A	*B*
a	1
a	2
b	1
b	2
c	1

W

B
1
2

$V \div W$

A
a
b

Example of division