International Islamic University Chittagong (IIUC)

Department of Computer and Communication Engineering

Program: B.sc (Eng.)
Course Code: CCE–3505
Type: Theory
Segment: Mid Term-Segment 3

Baizid MD Ashadzzaman
Designation: Adjunct Lecturer
Email: baizid.md.ashadzzaman@gmail.com
Phone: 8801862420119

Course Contents:

Contents	Note
Relational Algebra and SQL: Overview of the SQL Query, Language, SQL Data Definition, Basic Structure of SQL, Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Sub-queries, Modification of the Database, Join Expressions, Views, Triggers, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization.	

The Relational Algebra and SQL

The Relational Algebra

The relational algebra is a procedural query language. It consists of a set of operations that take one or two relations as input and produce a new relation as output.

Relational algebra mainly provides a theoretical foundation for relational databases and SQL.

The fundamental operations of relational algebra are as follows

- Select(**σ**)
- Project
- Union
- Set different
- Cartesian product
- Rename

- In addition to the fundamental operations, there are several other operations—namely, set intersection, natural join, and assignment.
- The select, project, and rename operations are called unary operations, because they operate on one relation.
- The other three operations operate on pairs of relations and therefore, they called binary operations.

Select Operation (**σ**):

The select operation selects tuples/rows that satisfy a given condition or predicate. It is used to select required tuples of the relations. We use the lowercase Greek letter sigma (**σ**) to denote selection. The predicate appears as a subscript to (**σ**).

Structure: $\sigma_{predicate}$ (Table)

Table
instructor(id, name, dept_name, salary) **department**(dept_name, building, budget) **section**(course_id, sec_id, semester, year, building, room_number, time_slot_id) **teaches**(ID , course_id, sec_id, semester, year)

1. To select those tuples of the instructor relation where the instructor is in the "Physics" department.

$$\sigma_{\text{dept\_name = "Physics"}}(\text{instructor})$$

2. To find all instructors with salary greater than $90,000

$$\sigma_{\text{salary} > 90000}(\text{instructor})$$

3. To find the instructors in Physics with a salary greater than $90,000

$$\sigma_{\text{dept\_name = "Physics"} \wedge \text{salary} > 90000}(\text{instructor})$$

Project Operation (π)

It is used to project required column data from a relation. Suppose we want to list all instructors' ID , name, and salary, but do not care about the dept name. The project operation allows us to produce this relation.

Since a relation is a set, any duplicate rows are eliminated. Projection is denoted by the uppercase Greek letter pi (π)

Structure: $\prod_{A1, A2, An}(r)$

Where A_1, A_2 , A_n are attribute names of relation r.

1. Show the ID, name, salary.

$$\prod_{\text{ID, name, salary}}(\text{instructor})$$

2. Show the all instructor name who are take the "Physics"

$$\prod_{\text{name}}(\sigma_{\text{dept\_name = "Physics"}}(\text{instructor}))$$

The Union Operation(∪)

Union operation in relational algebra is the same as union operation in set theory. Suppose there are two tuples R and S. The union operation contains all the tuples that are either in R or S or both in R & S. It eliminates the duplicate tuples. It is denoted by ∪.

1. Find the set of all courses assigned in the Fall 2009 semester and the Spring 2010 semester, or both.
To find the set of all courses assigned in the Fall 2009 semester

$$\prod_{\text{course id}} (\sigma_{\text{semester} = \text{“Fall”} \wedge \text{year} = 2009} (\text{section}))$$

2. To find the set of all courses assigned in the Spring 2010 semester

$$\prod_{\text{course id}} (\sigma_{\text{semester} = \text{“Spring”} \wedge \text{year} = 2010} (\text{section}))$$

Final Query:

$$\prod_{\text{course id}} (\sigma_{\text{semester} = \text{“Fall”} \wedge \text{year} = 2009} (\text{section})) \cup \prod_{\text{course id}} (\sigma_{\text{semester} = \text{“Spring”} \wedge \text{year} = 2010} (\text{section}))$$

Therefore, for a union operation r ∪ s. This union operation must hold the following condition:

1. R and S must have the same number of attributes.
2. Duplicate tuples are eliminated automatically.

Set Difference (–)

The result of set difference operation is tuples, which are present in one relation but are not in the second relation

Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in R but not in S..It is denoted by intersection minus (-).

Structure: $r - s$

- Finds all the tuples that are present in Books table but not in Articles Table.

$$\prod_{\text{author}} (\text{Books}) - \prod_{\text{author}} (\text{Articles})$$

- Find all the courses assigned in the Fall 2009 semester but not in Spring 2010 semester

$$\prod_{\text{course id}} (\sigma_{\text{semester} = \text{“Fall”} \wedge \text{year} = 2009} (\text{section})) - \prod_{\text{course id}} (\sigma_{\text{semester} = \text{“Spring”} \wedge \text{year} = 2010} (\text{section}))$$

The Set-Intersection Operation (∩)

An intersection is defined by the symbol ∩ that describe a relation consisting of a set of all tuple that are in both table.

Suppose there are two tuples R and S. The set intersection operation contains all tuples that are in both R & S. It is denoted by intersection ∩.

- To find the set of all courses assigned in both the Fall 2009 and the Spring 2010 semesters.

$\prod_{\text{course id}} (\sigma_{\text{semester = "Fall"} \wedge \text{year =2009}} (\text{section})) \cap \prod_{\text{course id}} (\sigma_{\text{semester = "Spring"} \wedge \text{year =2010}} (\text{section}))$

The Cartesian-Product Operation

The Cartesian-product operation is a relational algebra that allows us to combine information from any two relations. We write the Cartesian product of relations r1 and r2 as r1 × r2.

Instructor Table

Id	Name	Dept_name	Salary(k)
10101	Salam	CSE	56
10102	Rafiq	EEE	65
10103	Jabbor	CSE	67

Teaches Table

Id	course_id	sec_id	semester	year
10101	CSE-2423	A	4	2022
10102	EEE-1221	B	1	2022

r = instructor × teaches is

Ins.Id	Ins.Name	Ins.Dept_name	Inst.Salary	Tea.ID	Tea.Course_id	Tea.sec_id	Tea.semester	Tea.year
10101	Salam	CSE	56	10101	CSE-2423	A	4	2022
10101	Salam	CSE	56	10102	EEE-1221	B	1	2022
10102	Rafiq	EEE	65	10101	CSE-2423	A	4	2022
10102	Rafiq	EEE	65	10102	EEE-1221	B	1	2022
10103	Jabbor	CSE	67	10101	CSE-2423	A	4	2022
10103	Jabbor	CSE	67	10102	EEE-1221	B	1	2022

1. Find the teacher name, and course_d who taken course in CSE department

$$\prod_{\text{name, course id}} (\sigma_{\text{instructor.ID = teaches.ID}} (\sigma_{\text{dept name = "CSE"}} (\text{instructor} \times \text{teaches})))$$

For practice see this tutorial: https://www.youtube.com/watch?v=aX2px__lhvg

Rename (ρ)

The results of relational algebra are also relations but without any name. The rename operation allows us to rename the output relation. It is used to assign a new name to a relation and denoted with small Greek letter rho (ρ).

Syntax

ρ_{newname} (tablename or expression)

Consider the student table given below

Regno	Branch	Section
1	CSE	A
2	ECE	B

Regno	Branch	Section
3	CIVIL	B
4	IT	A

1.The student table is renamed with newstudent

$\rho_{newstudent}$ (student)

2.The name, branch column of student table are renamed newname and newbranch respectively

$\rho_{newname,newbranch}(\prod_{name,branch}$(student))

Join

A Join operation combines related tuples from different relations, if and only if a given join condition is satisfied. It is denoted by ⋈.

Different types of join:

Natural Join(⋈):

A natural join is the set of tuples of all combinations in R and S that are equal on their common attribute names. We can perform a Natural Join only if there is at least one common attribute that exists between two relations.

Instructor Table

Id	Name	Dept_name	Salary(k)
10101	Salam	CSE	56
10102	Rafiq	EEE	65
10103	Jabbor	CSE	67

Teaches Table

Id	course_id	sec_id	semester	year
10101	CSE-2423	A	4	2022
10102	EEE-1221	B	1	2022

Instructor ⋈ Teaches

Id	Name	Dept_name	Salary(k)	course_id	sec_id	semester	year
10101	Salam	CSE	56	CSE-2423	A	4	2022
10102	Rafiq	EEE	65	EEE-1221	B	1	2022

Left Outer Join(R ⟕ S):

Instructor ⟕ Teaches

Id	Name	Dept_name	Salary(k)	course_id	sec_id	semester	year
10101	Salam	CSE	56	CSE-2423	A	4	2022
10102	Rafiq	EEE	65	EEE-1221	B	1	2022
10103	Jabbor	CSE	67	- -	- -	- -	- -

Right Outer Join(R ⟖ S):

Instructor ⟖ Teaches

Id	Name	Dept_name	Salary (k)	course_id	Id	course_id	sec_id	semester	year
10101	Salam	CSE	56	CSE-2423	10101	CSE-2423	A	4	2022
10102	Rafiq	EEE	65	EEE-1221	10102	EEE-1221	B	1	2022

Full Outer Join: (R ⟗ S)

Instructor ⟗ Teaches

Id	Name	Dept_ name	Salary (k)	cours e_id	Id	cours e_id	sec_id	seme ster	year
10101	Salam	CSE	56	CSE-2423	10101	CSE-2423	A	4	2022
10102	Rafiq	EEE	65	EEE-1221	10102	EEE-1221	B	1	2022
10103	Jabbor	CSE	67	- -	- -	- -	- -	- -	- -