Unit II – Relational Model

Relational Data Model - keys, referential integrity and foreign keys, **Relational Algebra** - SQL fundamentals- Introduction, data definition in SQL, table, key and foreign key definitions, update behaviors-Intermediate SQL-Advanced SQL features - Embedded SQL- Dynamic SQL, CASE Studies- Oracle: Database Design and Querying Tools; SQL Variations and Extensions

Relational Algebra

- A procedural language consisting of a set of operations that take one or two relations as input and produce a new relation as their result.
- Six basic operators
  - select: σ
  - project: $\prod$
  - union: $\cup$
  - set difference: –
  - Cartesian product: x
  - rename: ρ

Select Operation

- The **select** operation selects tuples that satisfy a given predicate.
- Notation: $\sigma_{p}(r)$
- *p* is called the **selection predicate**
- Example: select those tuples of the *instructor* relation where the instructor is in the "Physics" department.
  - Query

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

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
33456	Gold	Physics	87000

Select Operation

- comparisons using

  $=, \neq, >, \geq. <. \leq$

  in the selection predicate.
- We can combine several predicates into a larger predicate by using the connectives:

  $\wedge$ (**and**), $\vee$ (**or**), $\neg$ (**not**)
- Example: Find the instructors in Physics with a salary greater $90,000, we write:

$$\sigma_{dept\_name=\text{"Physics"} \wedge salary > 90{,}000}\ (instructor)$$

- The select predicate may include comparisons between two attributes.
  - Example, find all departments whose name is the same as their building name:
  - $\sigma_{dept\_name=building}\ (department)$

σ topic = "Database" (Tutorials)

Assessment

1. **Selects tuples from Tutorials where topic = ‘Database’.**
2. **Selects tuples from Tutorials where the topic is ‘Database’ and ‘author’ is guru99.**
3. **Selects tuples from Customers where sales is greater than 50000**
4. **Select all the students of department ECE whose fees is greater than equal to 10000 and belongs to Team other than A.**

Project operation

- **Project operation** selects (or chooses) certain attributes discarding other attributes. The Project operation is also known as vertical partitioning since it partitions the relation or table vertically discarding other columns or attributes.
- Notations - $\pi_A(R)$

Project Example

- eliminate the *dept_name* attribute of *instructor*
- Query:

$$\Pi_{ID,\ name,\ salary}\ (instructor)$$

ID	*name*	*salary*
10101	Srinivasan	65000
12121	Wu	90000
15151	Mozart	40000
22222	Einstein	95000
32343	El Said	60000
33456	Gold	87000
45565	Katz	75000
58583	Califieri	62000
76543	Singh	80000
76766	Crick	72000
83821	Brandt	92000
98345	Kim	80000

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Composition of Relational Operations

- The result of a relational-algebra operation is relation and therefore of relational-algebra operations can be composed together into a **relational-algebra expression**

Find the names of all instructors in the Physics department.

$$\Pi_{name}(\sigma_{dept\_name = \text{“Physics”}}(instructor))$$

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

Cartesian-Product Operation

- The Cartesian-product operation (denoted by X) allows us to combine information from any two relations.
- Example: the Cartesian product of the relations *instructor* and t*eaches* is written as:

instructor X *teaches*

SNS INSTITUTIONS

The *instructor* X *teaches table*

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	12121	FIN-201	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	15151	MU-199	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
12121	Wu	Finance	90000	10101	CS-101	1	Fall	2017
12121	Wu	Finance	90000	10101	CS-315	1	Spring	2018
12121	Wu	Finance	90000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
12121	Wu	Finance	90000	15151	MU-199	1	Spring	2018
12121	Wu	Finance	90000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
15151	Mozart	Music	40000	10101	CS-101	1	Fall	2017
15151	Mozart	Music	40000	10101	CS-315	1	Spring	2018
15151	Mozart	Music	40000	10101	CS-347	1	Fall	2017
15151	Mozart	Music	40000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
15151	Mozart	Music	40000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...
22222	Einstein	Physics	95000	10101	CS-101	1	Fall	2017
22222	Einstein	Physics	95000	10101	CS-315	1	Spring	2018
22222	Einstein	Physics	95000	10101	CS-347	1	Fall	2017
22222	Einstein	Physics	95000	12121	FIN-201	1	Spring	2018
22222	Einstein	Physics	95000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Join Operation

- The Cartesian-Product

  instructor X *teaches*

  associates every tuple of instructor with every tuple of teaches.

- The **join** operation allows us to combine a select operation and a Cartesian-Product operation into a single operation.

Join Operation

The table corresponding to:

$$\sigma_{instructor.id = teaches.id}(instructor \times teaches))$$

instructor.ID	*name*	*dept_name*	*salary*	*teaches.ID*	*course_id*	*sec_id*	*semester*	*year*
10101	Srinivasan	Comp. Sci.	65000	10101	CS-101	1	Fall	2017
10101	Srinivasan	Comp. Sci.	65000	10101	CS-315	1	Spring	2018
10101	Srinivasan	Comp. Sci.	65000	10101	CS-347	1	Fall	2017
12121	Wu	Finance	90000	12121	FIN-201	1	Spring	2018
15151	Mozart	Music	40000	15151	MU-199	1	Spring	2018
22222	Einstein	Physics	95000	22222	PHY-101	1	Fall	2017
32343	El Said	History	60000	32343	HIS-351	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-101	1	Spring	2018
45565	Katz	Comp. Sci.	75000	45565	CS-319	1	Spring	2018
76766	Crick	Biology	72000	76766	BIO-101	1	Summer	2017
76766	Crick	Biology	72000	76766	BIO-301	1	Summer	2018
83821	Brandt	Comp. Sci.	92000	83821	CS-190	1	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-190	2	Spring	2017
83821	Brandt	Comp. Sci.	92000	83821	CS-319	2	Spring	2018
98345	Kim	Elec. Eng.	80000	98345	EE-181	1	Spring	2017

Union Operation

- The union operation allows us to combine two relations
- Notation: $r \cup s$
- For $r \cup s$ to be valid.

Union operation combines values in R1, R2 by removing duplicate ones.

1. *r, s* must have the *same* **arity** (same number of attributes)
2. The attribute domains must be **compatible** (example: 2^{nd} column of *r* deals with the same type of values as does the 2^{nd} column of *s*)

Example

- Consider two tables R1 and R2

Table R1 is as follows −

Regno	Branch	Section
1	CSE	A
2	ECE	B
3	MECH	B
4	CIVIL	A
5	CSE	B

Table R2 is as follows −

Regno	Branch	Section
1	CIVIL	A
2	CSE	A
3	ECE	B

To display all the regno of R1 and R2

```
∏regno(R1) ∪ ∏regno(R2)
```

Output

Regno
1
2
3
4
5

Example

- Consider two tables R1 and R2

Table R1 is as follows −

Regno	Branch	Section
1	CSE	A
2	ECE	B
3	MECH	B
4	CIVIL	A
5	CSE	B

Table R2 is as follows −

Regno	Branch	Section
1	CIVIL	A
2	CSE	A
3	ECE	B

To Retrieve branch and section of student from table R1 and R2

```
∏ branch, section (R1) ∪ ∏ branch, section (R2)
```

Output

Branch	Section
CSE	A
ECE	B
MECH	B
CIVIL	A
CSE	B

Example

- Find the set of all courses taught in both the Fall 2017 and the Spring 2018 semesters.

$$\Pi_{course\_id}\left(\sigma_{semester=\text{“Fall”}\ \wedge\ year=2017}(section)\right)\ \cup$$
$$\Pi_{course\_id}\left(\sigma_{semester=\text{“Spring”}\ \wedge\ year=2018}(section)\right)$$

course_id
CS-101
CS-315
CS-319
CS-347
FIN-201
HIS-351
MU-199
PHY-101

Set-Intersection Operation

- The set-intersection operation allows us to find tuples that are in both the input relations. – Common values in both the table
- It is denoted by ∩.
- Example
- Consider two sets,
- A={1,2,4,6} and B={1,2,7}
- Intersection of A and B
- A ∩ B ={1,2}

Set – Difference Operation

- The set-difference operation allows us to find tuples that are in one relation but are not in another.
- Notation *r* – *s*
- Set differences must be taken between **compatible** relations.
  - *r* and *s* must have the same arity
  - attribute domains of *r* and *s* must be compatible

Example

- find all courses taught in the Fall 2017 semester, but not in the Spring 2018 semester

$$\Pi_{course\_id}\left(\sigma_{semester=\text{“Fall”}\ \Lambda\ year=2017}(section)\right) - \Pi_{course\_id}\left(\sigma_{semester=\text{“Spring”}\ \Lambda\ year=2018}(section)\right)$$

The Assignment Operation

- The assignment operation is denoted by $\leftarrow$ and works like assignment in a programming language.
- Example: Find all instructor in the “Physics” and Music department.

$$Physics \leftarrow \sigma_{dept\_name=\text{“Physics”}}(instructor)$$
$$Music \leftarrow \sigma_{dept\_name=\text{“Music”}}(instructor)$$
$$Physics \cup Music$$

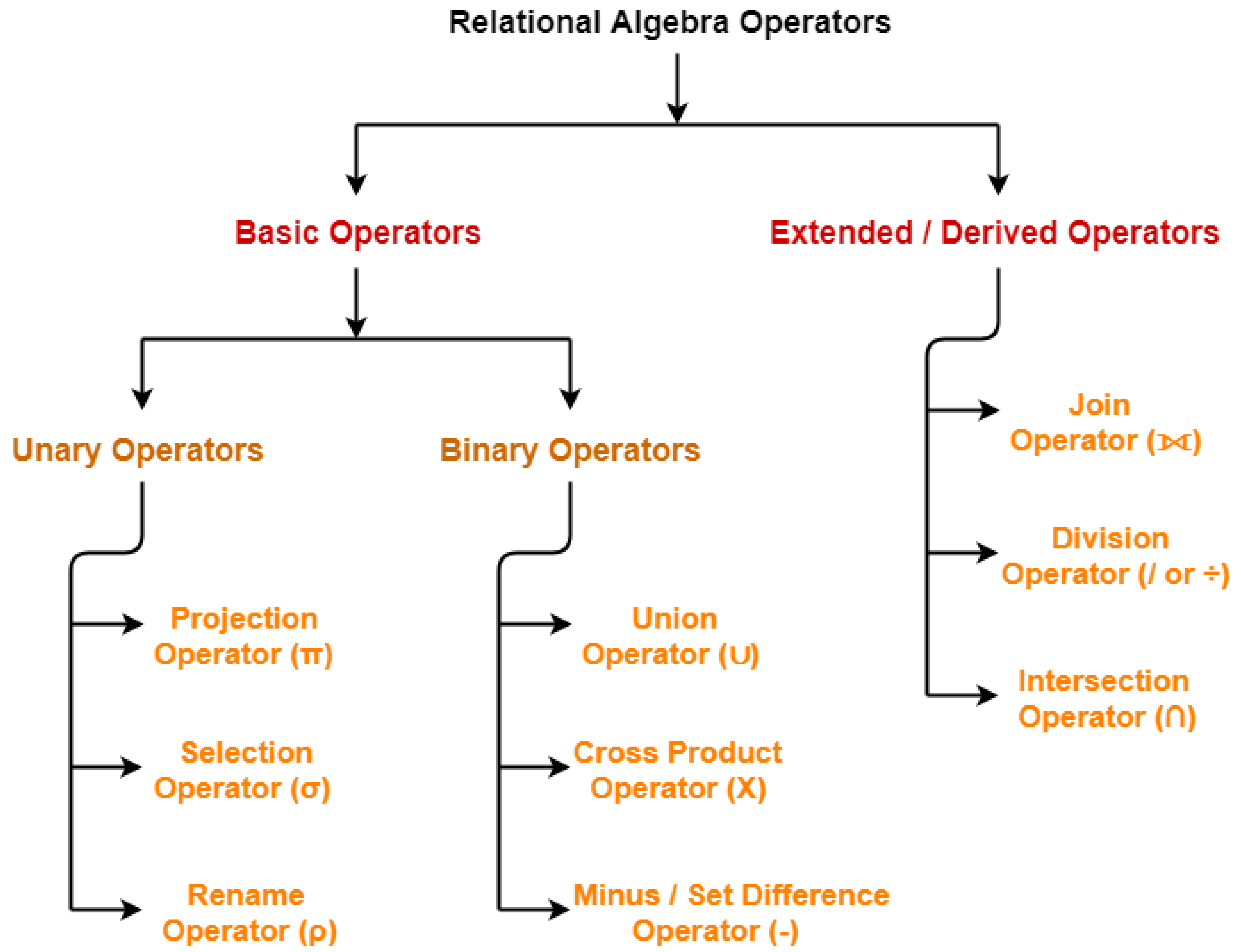
Rename Operation

- The RENAME operation is used to rename the output of a relation.

student table is renamed with newstudent

ρnewstudent (student)

Summary


Relational Algebra Operators
Basic Operators
Extended / Derived Operators
Unary Operators
Binary Operators
Projection Operator (π)
Selection Operator (σ)
Rename Operator (ρ)
Union Operator (∪)
Cross Product Operator (X)
Minus / Set Difference Operator (-)
Join Operator (⋈)
Division Operator (/ or ÷)
Intersection Operator (∩)


Thank You!