Introduction to Relational Model

Dr. Shrutilipi Bhattacharjee, Assistant Professor, Dept. of IT, NIT Karnataka, India

Relational Algebra

- 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:

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

- Result:

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

Select Operation

- We allow comparisons using the following operators in the selection predicate

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

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

$$\wedge \ (\textbf{and}),\ \vee \ (\textbf{or}),\ \neg \ (\textbf{not})$$

- 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 **department**s whose name is the same as their **building** name*:

$$\sigma_{\,dept\_name=building}\ (department)$$

Project Operation

- A unary operation that returns its argument relation, with certain attributes left out
- Notation:

$$\prod_{A_1, A_2, A_3 \ \ldots . A_k} (r)$$

- where $A_1, A_2, \ \ldots, A_k$ are attribute names and r is a relation name
- The result is defined as the relation of k columns obtained by erasing the columns that are not listed
- Duplicate rows removed from result, since relations are sets

Project Operation

- Example: *Eliminate the* dept_name *attribute of* ***instructor***
- Query*:*

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

- Result:

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

Select and Project Operations

- Relation ***r***

A	*B*	*C*	*D*
α	α	1	7
α	β	1	7
β	β	12	3
β	β	23	10

- $\sigma_{A=B \wedge D > 5}(r)$

A	*B*	*C*	*D*
α	α	1	7
β	β	23	10

$\Pi_{A,C}(r)$

A	*C*
α	1
α	1
β	12
β	23

=

A	*C*
α	1
β	12
β	23

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**
- Consider the query: *Find the* names *of all* ***instructors*** *in the Physics* ***department***

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

- Instead of giving the name of a relation as the argument of the projection operation, we give an expression that evaluates to a relation

Cartesian-Product Operation

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

  $$\textbf{\textit{instructor}} \times \textbf{\textit{teaches}}$$

- We construct a tuple of the result out of each possible pair of tuples: one from the ***instructor*** relation and one from the ***teaches*** relation (see next slide)
- Since the *instructor ID* appears in both relations we distinguish between these attribute by attaching to the attribute the name of the relation from which the attribute originally came
  - instructor.ID
  - teaches.ID

The *instructor* × *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
...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...

Cartesian-Product Operation

- Relation ***r, s***

A	*B*
α	1
β	2

C	*D*	*E*
α	10	a
β	10	a
β	20	b
γ	10	b

- ***r*** × ***s***

A	*B*	*C*	*D*	*E*
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

Composition of Relational Operations

- $\sigma_{A=C}(r \times s)$
  - $r \times s$

A	B	C	D	E
α	1	α	10	a
α	1	β	10	a
α	1	β	20	b
α	1	γ	10	b
β	2	α	10	a
β	2	β	10	a
β	2	β	20	b
β	2	γ	10	b

  - $\sigma_{A=C}(r \times s)$

A	B	C	D	E
α	1	α	10	a
β	2	β	10	a
β	2	β	20	b

Join Operation

- The Cartesian-Product associates every tuple of ***instructor*** with every tuple of ***teaches***

  $$\textit{\textbf{instructor}} \times \textit{\textbf{teaches}}$$

- Most of the resulting rows have information about ***instructors*** who did NOT teach a particular course

- To get only those tuples of "***instructor*** × ***teaches***" that pertain to ***instructor***s and the courses that they taught, we write:

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

- We get only those tuples of "***instructor*** × ***teaches***" that pertain to ***instructors*** and the courses that they taught

- The result of this expression, shown in the next slide

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

Join Operation

- The **join** operation allows us to combine a **select** operation and a **Cartesian-Product** operation into a single operation
- Consider relations ***r(R)*** and ***s(S)***
- Let "theta" be a predicate on attributes in the schema ***R*** "union" ***S***
- The join operation $\boldsymbol{r \bowtie_\theta s}$ is defined as follows:

$$r \bowtie_\theta s = \sigma_\theta(r \times s)$$

- Thus,

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

- It can equivalently be written as:

$$instructor \bowtie_{instructor.id\ =\ teaches.id} teaches$$

Outer Join

- An extension of the join operation that avoids loss of information
- Computes the join and then adds tuples form one relation that does not match tuples in the other relation to the result of the join
- Uses ***null*** values:
  - ***null*** signifies that the value is unknown or does not exist
  - All comparisons involving ***null*** are (roughly speaking) *false* by definition
  - Will study precise meaning of comparisons with nulls later
- Example:
  - Relation ***loan***
  - Relation ***borrower***

loan-number	***branch-name***	***amount***
L-170	Downtown	3000
L-230	Redwood	4000
L-260	Perryridge	1700

customer-name	***loan-number***
Jones	L-170
Smith	L-230
Hayes	L-155

Outer Join

- Inner Join: ***loan*** $\bowtie$ ***borrower***

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith

- Left Outer Join: ***loan*** ⟕ ***borrower***

loan-number	*branch-name*	*amount*	*customer-name*
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*

Outer Join

- Right Outer Join: ***loan* ⟖ *borrower***

loan-number	***branch-name***	***amount***	***customer-name***
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-155	*null*	*null*	Hayes

- Full Outer Join: ***loan* ⟗ *borrower***

loan-number	***branch-name***	***amount***	***customer-name***
L-170	Downtown	3000	Jones
L-230	Redwood	4000	Smith
L-260	Perryridge	1700	*null*
L-155	*null*	*null*	Hayes

Next Lecture

Introduction to Relational Model

Thank you for your attention...

Any question?

Contact:
Department of Information Technology, NITK Surathkal, India
6th Floor, Room: 13
Phone: +91-9477678768
E-mail: shrutilipi@nitk.edu.in