DATABASE SYSTEMS

CS – 355/ CE – 373

Instructor: Maria N. Samad

September 18th, 2024

RELATIONAL QUERY LANGUAGE

- A ***query language*** is a language in which a user requests some kind of information from the database
- These are usually on a level higher than that of a standard programming language
- ***Relational Algebra*** is one such procedural query language, which consists of a set of operations that take one or two relations as inputs and produces a new relation
- It provides a precise, mathematical foundation for querying databases, which helps in formalizing operations on data, ensuring correctness and consistency in query processing.

RELATIONAL ALGEBRA

- The main operations used in Relational Algebra are:
  - Select (σ)
  - Project (∏)
  - Union (∪)
  - Set Difference (–)
  - Cartesian Product (x)
  - Rename(ρ)
  - Intersection(∩)
  - Join (⋈)
  - Assignment (←)

RELATIONAL ALGEBRA

- The relational algebra operations can be classified into two categories:
  - ***Unary Operations***: Operations that work on one, single relation
  - ***Binary Operations***: Operations that work on pairs of relations

FUNDAMENTAL OPERATIONS

- There are six fundamental operations as follows:
  - Select (σ)
  - Project (∏)
  - Union (∪)
  - Set Difference (–)
  - Cartesian Product (x)
  - Rename(ρ)

SELECT (σ)

- This operation selects tuples that satisfy a given ***predicate***
- ***Predicate*** refers to the condition to choose the tuple(s) – used as a subscript to the σ symbol
- ***Relation*** is given as argument in brackets
- ***Notation***: $\sigma_p(r)$
- Here ‘p’ refers to the predicate and ‘r’ is the relation
- Example: Select those tuples of the instructor relation where the instructor is in the “Physics” department.
- ***Query:*** $\sigma_{\text{dept\_name=“Physics”}}(\textit{instructor})$

SELECT (σ)

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

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

Result:

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

SELECT (σ)

- We allow 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)

- The select operator is unary; that is, it is applied to a single relation.
- The select operation is commutative; that is, $\sigma_{c1}(\sigma_{c2}(R)) = \sigma_{c2}(\sigma_{c1}(R))$

SELECT (σ)

- Example:
  - Form a query to find the instructors in Physics Department with a salary greater $90,000:

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

- Query:
  - $\sigma_{\text{(dept\_name="Physics"} \wedge \text{ salary > 90,000)}}$ (Instructor)

- Result:

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000

SELECT (σ)

- The select predicate may include comparisons between two attributes.
- For example, find all departments whose name is the same as their building name:

  Query: $\sigma_{\text{dept\_name = building}}$ (department)

  Result: ?

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

SELECT (σ) – EXERCISES

- Activity Sheet:
  - Attempt **Part A**

SELECT (σ) – EXERCISES

- Activity Sheet **<u>Part A</u>** Solution:

1	$\sigma_{dept\_name\ =\ \text{“CS”}\ \wedge\ credits\ =\ 3}$ (*course*)
2	$\sigma_{(semester\ =\ \text{“Fall”}\ \wedge\ year\ =\ 2023)}$ (*section*)
3	$\sigma_{(semester\ =\ \text{“Spring”}\ \wedge\ year\ =\ 2024\ \wedge\ course\_id\ =\ sec\_id)}$ (*takes*)

PROJECT (Π)

- A unary operation that returns its argument relation, with certain attributes left out.
- Notation: $\Pi_{A1,\, A2,\, A3\, \ldots,\, \mathrm{Ak}}\ (r)$

  where $A1$, $A2$, …, Ak 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 (Π)

- Example:
  - Eliminate the *dept_name* attribute of *instructor*
- Query: $\prod_{ID,\ name,\ salary}$ (*instructor*)
- Result: $\rightarrow$

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

PROJECT (Π)

- Example:
  - Find the names of all instructors in the university
- Query: Π_{name} (*instructor*)
- Result: $\rightarrow$

name
Srinivasan
Wu
Mozart
Einstein
El Said
Gold
Katz
Califieri
Singh
Crick
Brandt
Kim

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

- The number of tuples in a relation resulting from a project operation is always less than or equal to the number of tuples in *R*.

PROJECT (Π) – EXERCISES

- Activity Sheet:
  - Attempt **Part B**

PROJECT (Π) – EXERCISES

- Activity Sheet **Part B** Solution:

1	$\Pi_{dept\_name,\ building}$ (*department*)
2	$\Pi_{course\_id,\ title}$ (*course*)
3	Π_{i_ID} (*advisor*)

COMPOSITION OF RELATIONAL OPERATORS

- The result of a relational-algebra operation is relation and therefore two or more relational-algebra operations can be composed together into a **relational-algebra expression**.
- Consider the following 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.

COMPOSITION OF RELATIONAL OPERATORS

- Example:
  - Find names of departments that are located in Watson building

- Query:
  - $\Pi_{dept\_name}(\sigma_{\text{building = "Watson"}}\ \text{(department)})$

- Result: $\rightarrow$

dept_name
Biology
Physics

dept_name	*building*	*budget*
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

COMPOSITION OF RELATIONAL OPERATORS

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

- Example:
  - Find the IDs and names of all instructors in Physics Department with a salary greater $90,000
- Query:
  - $\Pi_{ID,\ name}(\sigma_{(dept\_name=\text{"Physics"}\ \wedge\ salary > 90,000)}\ (Instructor))$
- Result: $\rightarrow$

ID	*name*
22222	Einstein

COMPOSITION OF RELATIONAL OPERATORS – EXERCISES

- Activity Sheet:
  - Attempt **Part C**

COMPOSITION OF RELATIONAL OPERATORS – EXERCISES

- Activity Sheet **<u>Part C</u>** Solution:

1	$\Pi_{name}\ (\sigma_{(dept\_name\ =\ \text{“CS”}\ \vee\ dept\_name\ =\ \text{“EE”})}\ (\textit{student}))$
2	$\Pi_{course\_id}\ (\sigma_{prereq\_id\ =\ \text{“CS-101”}}\ (\textit{prereq}))$
3	$\Pi_{ID,\ sec\_id}\ (\sigma_{(course\_id\ =\ \text{“CS-355”}\ \wedge\ semester\ =\ \text{“Fall”}\ \wedge\ year\ =\ 2024)}\ (\textit{teaches}))$

UNION ($\cup$)

- The union operation allows us to combine two relations

Notation: $r \cup s$

- For $r \cup s$ to be valid.

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

UNION (∪)

- Example:
  - Find all courses taught in the Fall 2017 semester, or in the Spring 2018 semester, or in both:
- Query:
  - $\Pi_{course\_id}\ (\sigma_{semester=\text{“Fall”}\ \wedge\ year=2017}\ (section))\ \cup$
    $\Pi_{course\_id}\ (\sigma_{semester=\text{“Spring”}\ \wedge\ year=2018}\ (section))$
- Result →

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

course_id	*sec_id*	*semester*	*year*	*building*	*room_number*	*time_slot_id*
BIO-101	1	Summer	2017	Painter	514	B
BIO-301	1	Summer	2018	Painter	514	A
CS-101	1	Fall	2017	Packard	101	H
CS-101	1	Spring	2018	Packard	101	F
CS-190	1	Spring	2017	Taylor	3128	E
CS-190	2	Spring	2017	Taylor	3128	A
CS-315	1	Spring	2018	Watson	120	D
CS-319	1	Spring	2018	Watson	100	B
CS-319	2	Spring	2018	Taylor	3128	C
CS-347	1	Fall	2017	Taylor	3128	A
EE-181	1	Spring	2017	Taylor	3128	C
FIN-201	1	Spring	2018	Packard	101	B
HIS-351	1	Spring	2018	Painter	514	C
MU-199	1	Spring	2018	Packard	101	D
PHY-101	1	Fall	2017	Watson	100	A

The *section* relation.

UNION (∪) – EXERCISES

- Activity Sheet:
  - Attempt **<u>Part D</u>**

UNION (∪) – EXERCISES

- Activity Sheet **Part D** Solution:

1	$\prod_{name}$ (instructor) ∪ $\prod_{name}$ (student) • If both instructor and student have the same name, it will remove the duplicates and in turn remove any one entry of instructor or student
2	$\prod_{ID}$ ($\sigma_{(course\_id\ =\ “CS-355”\ \wedge\ semester\ =\ “Fall”\ \wedge\ year\ =\ 2024)}$ (teaches)) ∪ $\prod_{ID}$ ($\sigma_{(course\_id\ =\ “CS-355”\ \wedge\ semester\ =\ “Fall”\ \wedge\ year\ =\ 2024)}$ (takes))
3	$\prod_{ID}$ ($\sigma_{(dept\_name\ =\ “CS”\ \vee\ dept\_name\ =\ “EE”)}$ (*student*)) ∪ $\prod_{ID}$ ($\sigma_{(dept\_name\ =\ “CS”\ \vee\ dept\_name\ =\ “EE”)}$ (*instructor*)) • If both instructor and student have the same IDs, it will remove the duplicates and in turn remove any one entry of student or instructor