(again) Please signup to this canvas

https://canvas.instructure.com/enroll/GH6XJ7

Or using the canvas join code: **GH6XJ7**

Recap: Relational Model

branch	acct_no	balance
Downtown	A-101	500
Brighton	A-201	900
Brighton	A-217	500

Tables → Relations
Columns → Attributes
Rows → Tuples

Schema → e.g., **Account**(branch: Branches, acct_no: GenAccounts, balance: Balances)

Domain → set of all possible values for an attribute.

e.g., Branches = dom(branch) = { Downtown, Brighton, … }
GenAccounts = dom(acct_no) = { A-101, A-201, A-217, … }
Balances = $\mathbb{R}$ = real numbers

Account $\subseteq$ Branches $\times$ GenAccounts $\times$ Balances

```
{ (Downtown, A-101, 500),
  (Brighton, A-201, 900),
  (Brighton, A-217, 500) }
```

Recap: 3 Parts of Relational Model

- **Structure**: The definition of relations and their contents.
- **Integrity:** Ensure the database's contents satisfy constraints.
- **Manipulation:** How to access and modify a database's contents.

Recap: Relational Algebra

- Selection: $\sigma_C(R) = \{t \mid t \in R, C(t)\}$
- Projection: $\Pi_{f_1(A_1),\ldots,f_n(A_n)}(R) = \{(f_1(t[A_1]), \ldots, f_n(t[A_n])) \mid t \in R\}$

  where $t[A_i]$ is the value of t for attribute A_i

A(a_id, b_id)

a_id	b_id
a1	101
a2	102
a2	103
a3	104

$B = \sigma_{\text{a\_id}='a2'}(A)$

a_id	b_id
a2	102
a2	103

$\pi_{b\_id-100,a\_id}(B)$

b_id	a_id
2	a2
3	a2

```
SELECT b_id - 100, a_id from A
where a_id = 'a2'
```

ICCS240 Database Management

Relational Algebra (cont.)

But which representation then?

Students(sid,sname,gpa)

```
SELECT DISTINCT
  sname,
  gpa
FROM Students
WHERE gpa > 3.5;
```

$$\Pi_{sname,gpa}(\sigma_{gpa>3.5}(Students))$$

or?

$$\sigma_{gpa>3.5}(\Pi_{sname,gpa}(Students))$$

How do we represent this query in RA?

Are these (always) logically equivalent?

Some algebraic properties (for queries optimization)

Selection commutes with projection

if and only if

the attributes referenced in the selection condition are a subset of the attributes in the projection

$$\pi_{A_1,\ldots,A_n}\left(\sigma_C(R)\right) = \sigma_C\left(\pi_{A_1,\ldots,A_n}(R)\right)$$

where attributes in $C \subseteq \{A_1, \ldots, A_n\}$

Which expression is preferred then?

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	10000
12121	Wu	Finance	5000
32343	El Said	History	200
45565	Katz	Comp. Sci.	30000
98345	Kim	Elec. Eng.	1000
76766	Crick	Biology	2000
10101	Srinivasan	Comp. Sci.	30000
58583	Califieri	History	200
83821	Brandt	Comp. Sci.	35000
15151	Mozart	Music	1000
33456	Gold	Physics	5000
76543	Singh	Finance	5000

Assume cost of projection for each tuple is much less than cost of testing for the given condition in a selection operation of each tuple.

$$\pi_{deptname,salary}\left(\sigma_{deptname="Comp.Sci."}(T)\right)$$

or

$$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$$

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple is much less than cost of testing for the given condition in a selection operation of each tuple.

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname="Comp.Sci."}(T)\right)$

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	10000
12121	Wu	Finance	5000
32343	El Said	History	200
45565	Katz	Comp. Sci.	30000
98345	Kim	Elec. Eng.	1000
76766	Crick	Biology	2000
10101	Srinivasan	Comp. Sci.	30000
58583	Califieri	History	200
83821	Brandt	Comp. Sci.	35000
15151	Mozart	Music	1000
33456	Gold	Physics	5000
76543	Singh	Finance	5000

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	10000
12121	Wu	Finance	5000
32343	El Said	History	200
45565	Katz	Comp. Sci.	30000
98345	Kim	Elec. Eng.	1000
76766	Crick	Biology	2000
10101	Srinivasan	Comp. Sci.	30000
58583	Califieri	History	200
83821	Brandt	Comp. Sci.	35000
15151	Mozart	Music	1000
33456	Gold	Physics	5000
76543	Singh	Finance	5000

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple **(X)** is much less than cost of testing for the given condition in a selection operation of each tuple **(Y).**

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname="Comp.Sci."}(T)\right)$

Run $|T|$ operations of σ
Outputs T' where $|T'| = 3$

ID	name	dept_name	salary
45565	Katz	Comp. Sci.	30000
10101	Srinivasan	Comp. Sci.	30000
83821	Brandt	Comp. Sci.	35000

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

ID	name	dept_name	salary
22222	Einstein	Physics	10000
12121	Wu	Finance	5000
32343	El Said	History	200
45565	Katz	Comp. Sci.	30000
98345	Kim	Elec. Eng.	1000
76766	Crick	Biology	2000
10101	Srinivasan	Comp. Sci.	30000
58583	Califieri	History	200
83821	Brandt	Comp. Sci.	35000
15151	Mozart	Music	1000
33456	Gold	Physics	5000
76543	Singh	Finance	5000

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple **(X)** is much less than cost of testing for the given condition in a selection operation of each tuple **(Y).**

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname="Comp.Sci."}(T)\right)$

Run $|T'|$ operations of π

ID	name	dept_name	salary
83821	Brandt	Comp. Sci.	35000

Total cost of Case 1 is $|T| \cdot Y + |T'| \cdot Y$

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

ID	name	dept_name	salary
22222	Einstein	Physics	10000
12121	Wu	Finance	5000
32343	El Said	History	200
45565	Katz	Comp. Sci.	30000
98345	Kim	Elec. Eng.	1000
76766	Crick	Biology	2000
10101	Srinivasan	Comp. Sci.	30000
58583	Califieri	History	200
83821	Brandt	Comp. Sci.	35000
15151	Mozart	Music	1000
33456	Gold	Physics	5000
76543	Singh	Finance	5000

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple **(X)** is much less than cost of testing for the given condition in a selection operation of each tuple **(Y).**

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname="Comp.Sci."}(T)\right)$

ID	name	dept_name	salary
83821	Brandt	Comp. Sci.	35000

Total cost of Case 1 is $|T| \cdot Y + |T'| \cdot Y$

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

ID	name	dept_name	salary
22222	Einstein	Physics	10000
12121	Wu	Finance	5000
32343	El Said	History	200
45565	Katz	Comp. Sci.	30000
98345	Kim	Elec. Eng.	1000
76766	Crick	Biology	2000
10101	Srinivasan	Comp. Sci.	30000
58583	Califieri	History	200
83821	Brandt	Comp. Sci.	35000
15151	Mozart	Music	1000
33456	Gold	Physics	5000
76543	Singh	Finance	5000

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple **(X)** is much less than cost of testing for the given condition in a selection operation of each tuple **(Y).**

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname\ \ "Comp.Sci."}(T)\right)$

ID	name	dept_name	salary
83821	Brandt	Comp. Sci.	35000

Total cost of Case 1 is $|T| \cdot Y + |T'| \cdot Y$

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

Run $|T|$ operations of π

Outputs $|T''| = |T| - 1$ (duplicates eliminated)

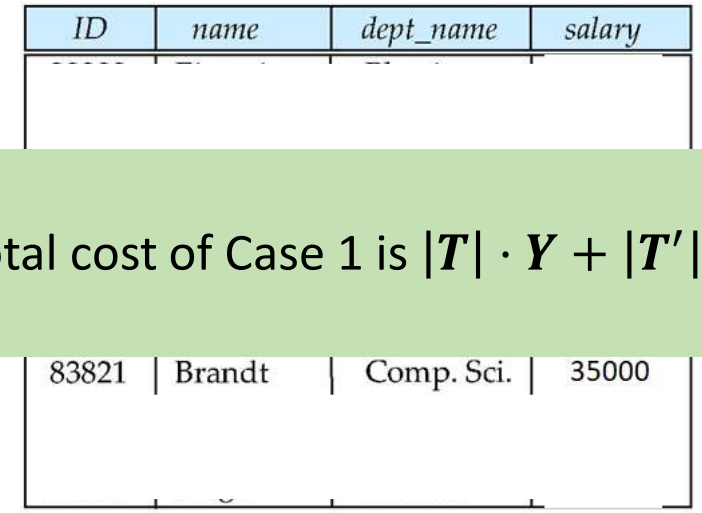
Physics	10000
Finance	5000
History	200
Comp. Sci.	30000
Elec. Eng.	1000
Biology	2000
History	200
Comp. Sci.	35000
Music	1000
Physics	5000
Finance	5000

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple **(X)** is much less than cost of testing for the given condition in a selection operation of each tuple **(Y).**

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname="Comp.Sci."}(T)\right)$



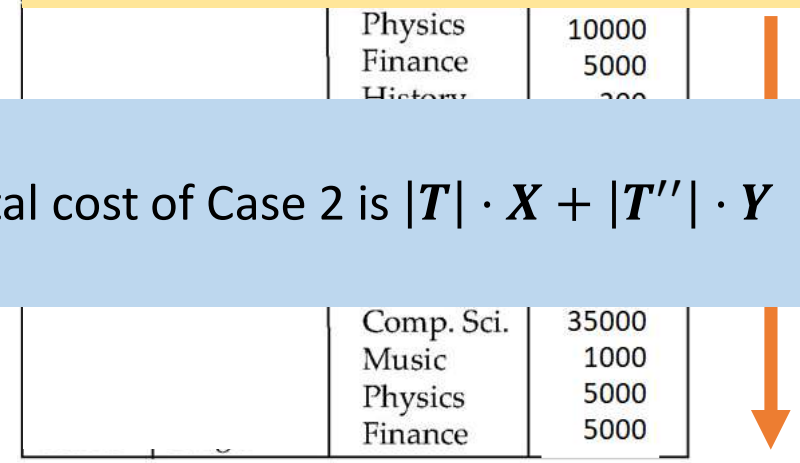
ID	name	dept_name	salary
83821	Brandt	Comp. Sci.	35000

Total cost of Case 1 is $|T| \cdot Y + |T'| \cdot X$

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

Run $|T''|$ operations of σ



Physics	10000
Finance	5000
History	300
Comp. Sci.	35000
Music	1000
Physics	5000
Finance	5000

Total cost of Case 2 is $|T| \cdot X + |T''| \cdot Y$

Which expression is preferred then? (Sol.)

Assume cost of projection for each tuple **(X)** is much less than cost of testing for the given condition in a selection operation of each tuple **(Y).**

Case 1: select then project

$\pi_{deptname,salary}\left(\sigma_{deptname\ \ "Comp.Sci."}(T)\right)$

$$|T| \cdot Y + |T'| \cdot X = 3 \cdot X + |T| \cdot Y$$

$\neq$

Case 2: project then select

$\sigma_{deptname="Comp.Sci."}\left(\pi_{deptname,salary}(T)\right)$

$$|T| \cdot X + |T''| \cdot Y = |T| \cdot X + (|T| - 1) \cdot Y$$

If X is very very small, Case 2 seems to be a better option?
If $X \approx Y$, Case 1 seems to be a better option?
Or maybe any other assumption on $|T|$, ???

Will cover "optimization plan" later in the course…

So far in Relational Algebra …

- Selection: $\sigma_C(R) = \{t \mid t \in R, C(t)\}$
- Projection: $\Pi_{f_1(A_1),\ldots,f_n(A_n)}(R) = \{(f_1(t[A_1]), \ldots, f_n(t[A_n])) \mid t \in R\}$

  where $t[A_i]$ is the value of t for attribute A_i

What's more on today plate?

- Rename
- Product (selection from two or more relations)
- Union, Intersection, Difference
- **`Join`**
- *Views*

Renaming ($\boldsymbol{\rho}$)

- Changes the schema, ***not the instance***
- Notation: $\rho_{R'(B_1,\ldots,B_n)}(R)$

- Note: this is shorthand for the proper form (since names, not order matters!): $\rho_{A_1/B_1}(R)$ or $\rho_{A_1 \to B_1}(R)$

Students(sid,sname,gpa)

SQL:

```
SELECT
  sid AS studId,
  sname AS name,
  gpa AS gradePtAvg
FROM Students;
```

RA:

$\rho_{studId,name,gradePtAvg}(Students)$

We care about this operator *because* we are working in a *named perspective!*

Another example:

Students

sid	sname	gpa
001	John	3.4
002	Bob	1.3

$\rho_{studId,name,gradePtAvg}(Students)$

Students

studId	name	gradePtAvg
001	John	3.4
002	Bob	1.3

Cross-Product (×)

- Generates a relation that contains all possible combinations of tuples from the input relations
- Notation: $R_1 \times R_2$
- Example:
  - Employee × Dependents

Students(sid,sname,gpa)
People(ssn,pname,address)

SQL:

```
SELECT *
FROM Students, People;
```

or

```
SELECT *
FROM Students CROSS JOIN People;
```

RA:

$$Students \times People$$

Another example:

People

ssn	pname	address
1234545	John	216 Rosse
5423341	Bob	217 Rosse

$\times$

Students

sid	sname	gpa
001	John	3.4
002	Bob	1.3

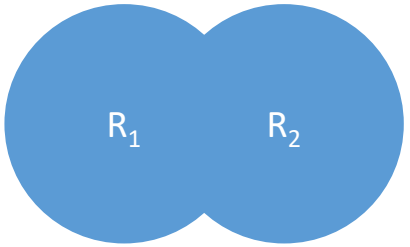
$$Students \times People$$

ssn	pname	address	sid	sname	gpa
1234545	John	216 Rosse	001	John	3.4
5423341	Bob	217 Rosse	001	John	3.4
1234545	John	216 Rosse	002	Bob	1.3
5423341	Bob	216 Rosse	002	Bob	1.3

Reminder: Union (∪) and Difference (–)

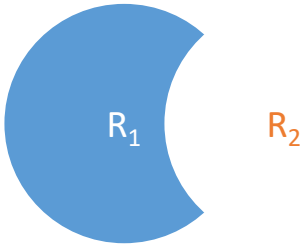
- R1 ∪ R2
- Example:

  ActiveEmployees ∪ RetiredEmployees



- R1 – R2
- Example:

  AllEmployees -- RetiredEmployees



Union (U)

- Generate a relation that contains all tuples that appear in at least one of the input relations.
- Notation: $R \cup S$
- ***Compatible*** only if (1) they have the same attributes and (2) the domain of each attribute matches.

R(a, b)

a	b
a1	11
a2	12
a3	13

S(a, b)

a	b
a4	14
a3	13
a5	15

$R \cup S$

R U S

a	b
a1	11
a2	12
a3	13
a4	14
a5	15

SQL:

```
(SELECT * FROM R)
UNION
(SELECT * FROM S)
```

Intersection (∪)

- Generate a relation that contains all tuples that appear in **both** the input relations.
- Notation: $R \cup S$
- ***Compatible*** only if (1) they have the same attributes and (2) the domain of each attribute matches.

R(a, b)

a	b
a1	11
a2	12
a3	13

S(a, b)

a	b
a4	14
a3	13
a5	15

$R \cap S$

R ∩ S

a	b
a3	13

SQL:

```
(SELECT * FROM R)
INTERSECT
(SELECT * FROM S)
```

DIFFERENCE (−)

- Generate a relation that contains all tuples that appear in the first input relation but not in the second one.
- Notation: $R - S$
- ***Compatible*** only if (1) they have the same attributes and (2) the domain of each attribute matches.

R(a, b)

a	b
a1	11
a2	12
a3	13

S(a, b)

a	b
a4	14
a3	13
a5	15

$R - S$

R - S

a	b
a1	11
a2	12

SQL:

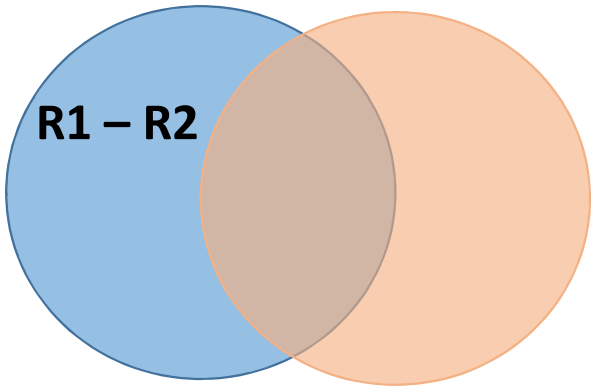
```
(SELECT * FROM R)
EXCEPT
(SELECT * FROM S)
```

INTERSECTION is a luxury

- It is a derived operator

  e.g., convenient but **redundant**

- R1 ∩ R2 = R1 – (R1 – R2)



Quick summary:

Relational Algebra: Theoretical Query Language

1. Select (σ)
2. Project (π)
3. Union ($\cup$)
4. Intersect ($\cap$)
5. (Set) Difference ($-$)
6. (Cartesian/Cross) Product ($-$)
7. Rename (ρ)

Together, they give semantics to practical query languages such as SQL.

EXCERCISE

Schema:

```
Customer(cid, cname, cstreet, ccity)
Borrower(cid, loan_no)
Depositor(cid, acct_no)
```

Find the names of customers who have both a (deposit) account and a loan (account).

- Write a relational algebra expressing this query
- *Write a corresponding SQL query?