COS221
L11 - Relational Algebra 2
(Chapter 6 in Edition 6 and Chapter 8 in Edition 7)

Linda Marshall

23 March 2023

Recap

Set operations — **Relational Algebra** — Relational database specific operations

UNION
INTERSECTION
SET DIFFERENCE

CARTESIAN/CROSS PRODUCT

SELECT
PROJECT

JOIN

Generalised projection
Aggregate
Recursive closure

Recap

Question: Write intersection ($\cap$) in terms of the other set operations

Figure 6.4
The set operations UNION, INTERSECTION, and MINUS. (a) Two union-compatible relations. (b) STUDENT $\cup$ INSTRUCTOR. (c) STUDENT $\cap$ INSTRUCTOR. (d) STUDENT – INSTRUCTOR. (e) INSTRUCTOR – STUDENT.

(a) **STUDENT**

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

Fname	Lname
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah

(b)

Fn	Ln
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

(c)

Fn	Ln
Susan	Yao
Ramesh	Shah

(d)

Fn	Ln
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

(e)

Fname	Lname
John	Smith
Ricardo	Browne
Francis	Johnson

Cartesian/Cross product

- The CARTESIAN PRODUCT ($\times$) operation is a binary operation where the two relations do not have to be union compatible.
- The operation produces a new relation by combining every tuple in the first relation with every tuple in the second relation.
- The cartesian product of relations R (of degree n) and S (of degree m) given by: $R(A_1, A_2, ..., A_n) \times R(B_1, B_2, ..., B_m)$ results in the relation Q (with degree $m+n$) : $Q(A_1, A_2, ..., A_n, B_1, B_2, ..., B_m)$

Cartesian/Cross product

- The order of the attributes in Q is the concatenation of the attributes in order of R followed by the attributes in the same order as in S.
- The number of tuples in Q will be $n_R * n_S$, where $n_R = |R|$ and $n_S = |S|$
- Applying the CARTESIAN PRODUCT operation by itself is usually meaningless. A combination of the CARTESIAN PRODUCT with PROJECTs and SELECTs provides a powerful sequence of operations.
- In SQL, the Cartesian product is realised using the CROSS JOIN.

Cartesian/Cross product

Figure 8.5
The CARTESIAN PRODUCT (CROSS PRODUCT) operation.

FEMALE_EMPS

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Alicia	J	Zelaya	999887777	1968-07-19	3321Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291Berry, Bellaire, TX	F	43000	888665555	4
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

EMPNAMES

Fname	Lname	Ssn
Alicia	Zelaya	999887777
Jennifer	Wallace	987654321
Joyce	English	453453453

EMP_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	...
Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	...
Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	...
Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	...
Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	...
Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	...
Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	...
Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	...
Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	...
Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...
Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	...
Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	...
Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	...
Joyce	English	453453453	333445555	Alice	F	1986-04-05	...
Joyce	English	453453453	333445555	Theodore	M	1983-10-25	...
Joyce	English	453453453	333445555	Joy	F	1958-05-03	...
Joyce	English	453453453	987654321	Abner	M	1942-02-28	...
Joyce	English	453453453	123456789	Michael	M	1988-01-04	...
Joyce	English	453453453	123456789	Alice	F	1988-12-30	...
Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	...

ACTUAL_DEPENDENTS

Fname	Lname	Ssn	Essn	Dependent_name	Sex	Bdate	...
Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	...

RESULT

Fname	Lname	Dependent_name
Jennifer	Wallace	Abner

$$\text{FEMALE\_EMPS} \leftarrow \sigma_{\text{Sex}=\text{'F'}}(\text{EMPLOYEE})$$
$$\text{EMPNAMES} \leftarrow \pi_{\text{Fname, Lname, Ssn}}(\text{FEMALE\_EMPS})$$
$$\text{EMP\_DEPENDENTS} \leftarrow \text{EMPNAMES} \times \text{DEPENDENT}$$
$$\text{ACTUAL\_DEPENDENTS} \leftarrow \sigma_{\text{Ssn}=\text{Essn}}(\text{EMP\_DEPENDENTS})$$
$$\text{RESULT} \leftarrow \pi_{\text{Fname, Lname, Dependent\_name}}(\text{ACTUAL\_DEPENDENTS})$$

The JOIN Operation

- The JOIN operation ($\bowtie$) is used to combine related tuples from two relations, resulting in "longer" tuples.
- By joining, relationships among relations can be processed

Figure 8.6
Result of the JOIN operation DEPT_MGR $\leftarrow$ DEPARTMENT $\bowtie_{\text{Mgr\_ssn=Ssn}}$ EMPLOYEE.

DEPT_MGR

Dname	Dnumber	Mgr_ssn	...	Fname	Minit	Lname	Ssn	...
Research	5	333445555	...	Franklin	T	Wong	333445555	...
Administration	4	987654321	...	Jennifer	S	Wallace	987654321	...
Headquarters	1	888665555	...	James	E	Borg	888665555	...

- A JOIN can be written as a CARTESIAN PRODUCT followed by a SELECT.
  $RESULT \leftarrow DEPARTMENT \times EMPLOYEE$
  $DPT\_MGR \leftarrow \sigma_{Mgr\_ssn=Ssn}(RESULT)$

The JOIN Operation

- The general form of a JOIN $Q \leftarrow R \bowtie_{<join\_condition>} S$ where $R(A_1, A_2, ..., A_n)$ and $R(B_1, B_2, ..., B_m)$ results in $Q(A_1, A_2, ..., A_n, B_1, B_2, ..., B_m)$
- A join condition is of the form: $< condition > AND < condition > AND...AND < condition >$
- Each $< condition >$ is of the form: $A_i\ \theta\ B_j$ where A_i is in R and B_j is in S is one of the comparison operators
- This is referred as a THETA join.
- Specialisations of the THETA join are EQUIJOIN and NATURAL JOIN.

The JOIN Operation

- EQUIJOIN is when the only comparison operator is $=$.
  One or more pairs of attributes have identical values in a EQUIJOIN. One of these attributes in each of the pairs are superfluous in the results and can be dropped. To make provision for this, a special operator, the NATURAL JOIN is defined.
- A NATURAL JOIN is denoted by $*$.
  A natural join requires that the pair of attributes from each relation participating in the join have the same name in both relations. Renaming of attributes is therefore necessary if they are not the same.
  $DEPT \leftarrow \rho_{(Dname, Dnum, Mgr\_ssn, Mgr\_start\_date)}(DEPARTMENT)$
  $PROJ\_DEPT \leftarrow PROJECT * DEPT$
  $Dnum$ is a join attribute.

The JOIN Operation

- If no combination of tuples satisfies the join condition the result of the join is an empty relation.
- If there is no join condition the join is a Cartesian product.
- A join combining two relations to form a single relation is also called an inner join. That is a join formed by combining a CARTESIAN PRODUCT followed by a SELECT operation.
- In contrast outer joins were developed when all tuples in either R or S or both relations are to be kept regardless of whether or not they have matching tuples in the other relation. Two types of outer joins exist, a LEFT OUTER JOIN (⟕) and a RIGHT OUTER JOIN (⟖).

The JOIN Operation

- Left outer join example:
  $TEMP \leftarrow EMPLOYEE \mathbin{⟕}_{Ssn=Mgs\_ssn} DEPARTMENT$
  $RESULT \leftarrow \pi_{Fname,Minit,Lname,Dname}(TEMP)$
- Every tuple in EMPLOYEE is kept. If no matching tuple is found in DEPARTMENT, the attributes are padded with NULL values.

Figure 8.12
The result of a LEFT OUTER JOIN operation.

RESULT

Fname	Minit	Lname	Dname
John	B	Smith	NULL
Franklin	T	Wong	Research
Alicia	J	Zelaya	NULL
Jennifer	S	Wallace	Administration
Ramesh	K	Narayan	NULL
Joyce	A	English	NULL
Ahmad	V	Jabbar	NULL
James	E	Borg	Headquarters

The Complete Set of Relational Algebra Operations

The set of relational operations $\{\sigma, \pi, \cup, \rho, -, \times\}$ is a complete set. That is, any of the other original relational algebra operations can be expressed as a sequence of operations from this set.

- The INTERSECTION operation can be expressed using UNION and SET DIFFERENCE.
  $R \cap S \equiv (R \cup S) - ((R - S) \cup (S - R))$
- The JOIN operation is a CARTESIAN PRODUCT followed by a SELECT.
  $R \bowtie_{<condition>} S \equiv \sigma_{<condition>}(R \times S)$
- A NATURAL JOIN is a RENAME of attributes followed by a CARTESIAN PRODUCT, a SELECT and a PROJECT.
  $(\; R1 \leftarrow \rho_{(A_1, A_2, \ldots, A_n)}(R)$
  $S1 \leftarrow \rho_{(B_1, B_2, \ldots, B_n)}(S)$
  $TEMP \leftarrow S1 \times S2$
  $RESULT \leftarrow \pi_{(C_1, C_2, \ldots, C_k)}(\sigma_{<condition>}(TEMP))$
  $) \equiv R * S$

The Complete Set of Relational Algebra Operations

- The DIVISION ($\div$) operation is a combination of PROJECT, CARTESIAN PRODUCT and SET DIFFERENCE. The DIVISION operation is applied to two relations
  (
  $T1 \leftarrow \pi_Y(R)$
  $T2 \leftarrow \pi_Y((S \times T1) - R)$
  $T \leftarrow T1 - T2$
) $\equiv (R \div S)$
  Where the attributes of R are a subset of the attributes of S. The result of the division is a relation T that for every tuple t in T, the values of t must appear in R in combination with every tuple in S.

The Complete Set of Relational Algebra Operations

- For example: *Retrieve the names of all employees who work on all the projects that John Smith works on.*

$$SMITH \leftarrow \sigma_{Fname='John' \text{ AND } Lname='Smith'}(EMPLOYEE)$$

$$SMITH\_PNOS \leftarrow \pi_{Pno}(WORKS\_ON \bowtie_{Essn=Ssn} SMITH)$$

$$SSN\_PNOS \leftarrow \pi_{Essn, Pno}(WORKS\_ON)$$

$$SSNS(Ssn) \leftarrow SSN\_PNOS \div SMITH\_PNOS$$

$$RESULT \leftarrow \pi_{Fname, Lname}(SSNS * EMPLOYEE)$$

Figure 6.8
The DIVISION operation. (a) Dividing SSN_PNOS by SMITH_PNOS. (b) $T \leftarrow R \div S$.

(a)

SSN_PNOS

Essn	Pno
123456789	1
123456789	2
666884444	3
453453453	1
453453453	2
333445555	2
333445555	3
333445555	10
333445555	20
999887777	30
999887777	10
987987987	10
987987987	30
987654321	30
987654321	20
888665555	20

SMITH_PNOS

Pno
1
2

SSNS

Ssn
123456789
453453453

(b)

R

A	B
a1	b1
a2	b1
a3	b1
a4	b1
a1	b2
a3	b2
a2	b3
a3	b3
a4	b3
a1	b4
a2	b4
a3	b4

S

A
a1
a2
a3

T

B
b1
b4

Additional Operations

- Generalised projection adds functions on attributes to be included in the projection list. The general form is given by: $\pi_{F_1, F_2, \ldots, F_n}(R)$ where F_i is a function over the attributes. These functions may include arithmetic operations and constants

  For example:

  Assume EMPLOYEE (Ssn, Salary, Deduction, Years_service)
  Require the following:
  - NetSalary = Salary - Deduction
  - Bonus = 2000 * Years_service; and
  - Tax = 0.25 * Salary

$$REPORT \leftarrow \rho_{(Ssn,\ Net\_salary,\ Bonus,\ Tax)}(\pi_{Ssn,\ Salary-Deduction,\ 2000*Years\_service,\ 0.25*Salary}(EMPLOYEE))$$

Additional Operations

- Aggregate functions ($\mathscr{F}$ - Script F) on collections, for example average or total salary. Common functions included are: SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT. The general form of the function is given by:
  $_{<grouping\_attributes>}\mathscr{F}_{<function\_list>}R$
  Where $<grouping\_attributes>$ is the list of attributes in R to which the aggregate is to be applied, $<function\_list>$ the list of pairs ($<function><attribute>$) specifying the attributes and functions to be calculated with respect to the grouping attributes. The resulting relation will include the grouping attributes and an attribute per pair in the function list.

Additional Operations

- Aggregate example: *Retrieve each department number, the number of employees in each department and their average salary*

R

(a)

Dno	No_of_employees	Average_sal
5	4	33250
4	3	31000
1	1	55000

(b)

Dno	Count_ssn	Average_salary
5	4	33250
4	3	31000
1	1	55000

(c)

Count_ssn	Average_salary
8	35125

Figure 8.10
The aggregate function operation.

a. $\rho_{R(\text{Dno, No\_of\_employees, Average\_sal})}(_{\text{Dno}}\,\mathfrak{I}\,_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE}))$.

b. $_{\text{Dno}}\,\mathfrak{I}\,_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

c. $\mathfrak{I}\,_{\text{COUNT Ssn, AVERAGE Salary}}(\text{EMPLOYEE})$.

Additional Operations

- Recursive closure operations which are applied between tuples of the same type, for example: the supervisor-employee relationship. In many cases, this type of relationship is not just on one level. An employee may indirectly manage another employee because of a management hierarchy. To identify multiple levels of relationship, a transitive closure is computed. The SQL3 standard includes recursive closure.

Find all employees directly supervised by "James Borg".

$$\text{BORG\_SSN} \leftarrow \pi_{Ssn}(\sigma_{Fname='James' \text{ AND } Lname='Borg'}(\text{EMPLOYEE}))$$

$$\text{SUPERVISION}(Ssn1, Ssn2) \leftarrow \pi_{Ssn, Super\_ssn}(\text{EMPLOYEE})$$

$$\text{RESULT1}(Ssn) \leftarrow \pi_{Ssn1}(\text{SUPERVISION} \bowtie_{Ssn2=Ssn} \text{BORG\_SSN})$$

Now, find all employees supervised by an employee who is supervised by "Borg".

$$\text{RESULT2}(Ssn) \leftarrow \pi_{Ssn1}(\text{SUPERVISION} \bowtie_{Ssn2=Ssn} \text{RESULT1})$$

$$\text{RESULT} \leftarrow \text{RESULT2} \cup \text{RESULT1}$$

Examples

Using the COMPANY database:

- Query 1: Retrieve the name and address of all employees who work for the 'Research' department.
- Query 2: For every project located in 'Stafford', list the project number, the controlling department number, and the departments manager's last name, address and birth date.
- Query 3: Find the names of employees who work on all the projects controlled by department number 5.
- Query 4: Make a list of project numbers for projects that involve an employee who's last name is 'Smith', either as a worker or as a manager of the department that controls the project.
- Query 5: List the names of all employees with two or more dependents.
- Query 6: Retrieve the names of employees who have no dependents.
- Query 7: List the names of managers who have at least one dependent.