https://dbis-uibk.github.io/relax/calc.htm

Relational algebra

```
SELECT LNAME, FNAME
FROM   EMPLOYEE
WHERE SALARY > C
```

$$\pi_{LNAME,FNAME}(\sigma_{SALARY>C}(EMPLOYEE))$$

- Relational algebra, first created by Edgar F. Codd while at IBM, is a family of algebras with a well-founded semantics used for modelling the data stored in relational databases, and defining queries on it. The main application of relational algebra is providing a theoretical foundation for relational databases, particularly query languages for such databases, chief among which is SQL.
- Relational Algebra provides a formal foundation for relational model operations. The algebra defines a set of operations for the relational model. It very useful for representing query execution plans, and query optimization techniques. The algebra operations produce new relations. The result of a retrieval is a new relation. A sequence of relational algebra operations forms a **relational algebra expression.**

Question*: It seems to me that formulating expressions in relational algebra is basically the same as formulating queries in SQL and that much the same thought processes underly both tasks. In particular, I can't really see that knowing relational algebra makes it* easier *to write SQL queries or vice-versa. This makes me wonder if the teaching of relational algebra is just some sort of historical hangover, or if there actually specific benefits to knowing it.*
Answer 1: Yes I would agree with that. But I think any introduction to database course needs to teach relational algebra so that students understand the foundations that led to SQL. Teaching just SQL would be like an introduction to industrial design class teaching just a CAD/CAM tool and not first exposing the students to the fundamental principles of industrial design.
Answer 2: If you have formulated a query one way and are getting poor performance, you have the skills to formulate it a different way and know it has the same semantics. Another practical advantage to this is in the specification of database constraints. First, understanding the relational algebra enables you to determine the simplest way to formulate the constraint.
Answer 3: I strongly feel you can be successful as a DBA without taking a formal course in relational algebra, just like you can be a successful programmer without taking a formal course in discrete math. The need *to take a course in relational algebra would very much depend on your career path/goals. If someone wanted to get a Masters in Algorithms etc. I would say it is fairly obvious that he needs to take and master discrete math. Similarly if your goal is to write a database engine or be part of the core team working on a major relational database engine then I would strongly recommend mastering relational algebra, stats, and the like.*

- Six basic operators in relational algebra:

select	σ	selects a subset of tuples from reln
project	π	deletes unwanted columns from reln
Cartesian Product	$\times$	allows to combine two relations
Set-difference	$-$	tuples in reln. 1, but not in reln. 2
Union	$\cup$	tuples in reln 1 plus tuples in reln 2
Rename	ρ	renames attribute(s) and relation

- The operators take one or two relations as input and give a new relation as a result (relational algebra is "closed").

A relation is a set. In mathematics, a set is a collection of distinct objects, considered as an object in its own right.

Operation	My HTML	Symbol
Projection	**PROJECT**	π
Selection	**SELECT**	σ
Renaming	**RENAME**	ρ
Union	**UNION**	$\cup$
Intersection	**INTERSECTION**	$\cap$
Assignment	<-	$\leftarrow$

Operation	My HTML	Symbol
Cartesian product	**X**	$\times$
Join	**JOIN**	$\bowtie$
Left outer join	**LEFT OUTER JOIN**	$⟕$
Right outer join	**RIGHT OUTER JOIN**	$⟖$
Full outer join	**FULL OUTER JOIN**	$⟗$
Semijoin	**SEMIJOIN**	$\ltimes$

Select Operation

- Example: given the relation r

A	B	C	D
α	α	1	7
α	β	5	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

Project Operation

- Notation: $\pi_{A_1,A_2,\ldots,A_k}(r)$

  where $A_1, \ldots, A_k$ are attribute names and
  r is a relation (name).

- The result of the projection operation is defined as the relation that has k columns obtained by erasing all columns from r that are not listed.

- Duplicate rows are removed from result because relations are sets.

- Example: given the relations r

r

A	B	C
α	10	2
α	20	2
β	30	2
β	40	4

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

A	C
α	2
β	2
β	4

Cartesian Product

- Example: relations r, s:

r

A	B
α	1
β	2

s

C	D	E
α	10	$+$
β	10	$+$
β	20	$-$
γ	10	$-$

$r \times s$

A	B	C	D	E
α	1	α	10	$+$
α	1	β	10	$+$
α	1	β	20	$-$
α	1	γ	10	$-$
β	2	α	10	$+$
β	2	β	10	$+$
β	2	β	20	$-$
β	2	γ	10	$-$

Union Operator

- Example: given the relations r and s

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cup s$

A	B
α	1
α	2
β	1
β	3

Set Difference Operator

- Notation: $r - s$ where both r and s are relations

- For $r - s$ to be applicable,
  1. r and s must have the same arity
  2. Attribute domains must be compatible
- Example: given the relations r and s

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r - s$

A	B
α	1
β	1

Rename Operation

- Allows to name and therefore to refer to the result of relational algebra expression.
- Allows to refer to a relation by more than one name (e.g., if the same relation is used twice in a relational algebra expression).
- Example:

  $$\rho_x(E)$$

  returns the relational algebra expression E under the name x

  If a relational algebra expression E (which is a relation) has the arity k, then

  $$\rho_{x(A_1,A_2,\ldots,A_k)}(E)$$

  returns the expression E under the name x, and with the attribute names $A_1, A_2, \ldots, A_k$.

Assume the following relations:

group:BookStore

BOOKS ={ISBN:NUMBER, TITLE:STRING, PUBLISHER:STRING, YEAR:NUMBER}
STUDENTS= {STID:NUMBER, SNAME:STRING, MAJOR:STRING, AGE:NUMBER}
AUTHORS= {ANAME:STRING, ADDRESS:STRING}
BORROWS= {ISBN:NUMBER, STID:NUMBER, DATE:NUMBER}
HAS_WRITTEN= {ISBN:NUMBER, ANAME:STRING}
DESCRIBES= {ISBN:NUMBER, KEYWORD:STRING}

BOOKS (ISBN, Title, Publisher, Year)
STUDENTS (StId, SName, Major, Age)
AUTHORS (AId, AName, Address)
BORROWS (ISBN, StId, Date)
HAS_WRITTEN (ISBN, AId)
DESCRIBES (ISBN, Keyword)

-- 1. List the year and title of each book.
-- 2. List all information about students whose major is CS
-- 3. List all students and all the possible books they can borrow
-- 4. List all books published by Pearson before 1995.
-- 5. List the name of authors who live in Sacramento.
-- 6. List the name of students who are older than 25 and are not studying CS.
-- 7. Rename AName in the relation AUTHORS to Name.

Composition of Operations

- It is possible to build relational algebra expressions using multiple operators similar to the use of arithmetic operators (nesting of operators)
- Example: $\sigma_{A=C}(r \times s)$

$r \times s$

A	B	C	D	E
α	1	α	10	+
α	1	β	10	+
α	1	β	20	−
α	1	γ	10	−
β	2	α	10	+
β	2	β	10	+
β	2	β	20	−
β	2	γ	10	−

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

A	B	C	D	E
α	1	α	10	+
β	2	β	10	+
β	2	β	20	−

Additional Operators

These operators do not add any power (expressiveness) to the relational algebra but simplify common (often complex and lengthy) queries.

Set-Intersection	$\cap$	
Natural Join	$\bowtie$	
Condition Join	$\bowtie_C$	(also called Theta-Join)
Division	$\div$	
Assignment	$\longleftarrow$	

Set-Intersection

- Notation: $r \cap s$
  Defined as $r \cap s := \{t \mid t \in r \textbf{ and } t \in s\}$
- For $r \cap s$ to be applicable,
  1. r and s must have the same arity
  2. Attribute domains must be compatible
- Derivation: $r \cap s = r - (r - s)$
- Example: given the relations r and s

r

A	B
α	1
α	2
β	1

s

A	B
α	2
β	3

$r \cap s$

A	B
α	2

Natural Join

- Notation: $r \bowtie s$
- Let r, s be relations on schemas R and S, respectively. The result is a relation on schema $R \cup S$. The result tuples are obtained by considering each pair of tuples $t_r \in r$ and $t_s \in s$.
- If t_r and t_s have the same value for each of the attributes in $R \cap S$ ("same name attributes"), a tuple t is added to the result such that
  - t has the same value as t_r on r
  - t has the same value as t_s on s
- Example: Given the relations $R(A, B, C, D)$ and $S(B, D, E)$
  - Join can be applied because $R \cap S \neq \emptyset$
  - the result schema is (A, B, C, D, E)
  - and the result of $r \bowtie s$ is defined as

    $\pi_{r.A,r.B,r.C,r.D,s.E}(\sigma_{r.B=s.B \wedge r.D=s.D}(r \times s))$
- Example: given the relations r and s

r

A	B	C	D
α	1	α	a
β	2	γ	a
γ	4	β	b
α	1	γ	a
δ	2	β	b

s

B	D	E
1	a	α
3	a	β
1	a	γ
2	b	δ
3	b	τ

$r \bowtie s$

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

- Natural join is associative
  - (*instructor* $\bowtie$ *teaches*) $\bowtie$ *course* is equivalent to *instructor* $\bowtie$ (*teaches* $\bowtie$ *course*)
- Natural join is commutative
  - *instruct* $\bowtie$ *teaches* is equivalent to *teaches* $\bowtie$ *instructor*

Condition Join

- Notation: $r \bowtie_C s$

  C is a condition on attributes in $R \cup S$, result schema is the same as that of Cartesian Product. If $R \cap S \neq \emptyset$ and condition C refers to these attributes, some of these attributes must be renamed.
  Sometimes also called *Theta Join* ($r \bowtie_\theta s$).

- Derivation: $r \bowtie_C s = \sigma_C(r \times s)$

- Note that C is a condition on attributes from both r and s

- Example: given two relations r, s

r

A	B	C
1	2	3
4	5	6
7	8	9

s

D	E
3	1
6	2

$r \bowtie_{B<D} s$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

If C involves only the comparison operator "=", the condition join is also called *Equi-Join.*

- Example 2:

r

A	B	C
4	5	6
7	8	9

s

C	D
6	8
10	12

$r \bowtie_{C=SC} (\rho_{S(SC,D)}(s))$

A	B	C	SC	D
4	5	6	6	8

Outer Join – Example

- Join

  instructor $\bowtie$ *teaches*

- Left Outer Join

  instructor $⟕$ *teaches*

- Right Outer Join

  instructor $⟖$ *teaches*

- Full Outer Join

  instructor $⟗$ *teaches*

Assignment

- Operation ($\longleftarrow$) that provides a convenient way to express complex queries.
  Idea: write query as sequential program consisting of a series of assignments followed by an expression whose value is "displayed" as the result of the query.

- Assignment must always be made to a temporary relation variable.
  The result to the right of $\longleftarrow$ is assigned to the relation variable on the left of the $\longleftarrow$. This variable may be used in subsequent expressions.

BOOKS (ISBN, Title, Publisher, Year)
STUDENTS (StId, SName, Major, Age)
AUTHORS (AId, AName, Address)
BORROWS (ISBN, StId, Date)
HAS_WRITTEN (ISBN, AId)
DESCRIBES (ISBN, Keyword)
(1, 'database')
(1,'programming')
(2,'database')

-- 1. List each book with its keywords
-- 2. List each student with the books s/he has borrowed
-- 3. List the title of books written by the author 'Ullman'
-- 4. List the names of all students who have borrowed a book and who are CS majors
-- 5. Redo query 4 using assignments
-- 6. List the authors of the books the student 'Smith' has borrowed
-- 7. Which books have both keywords 'database' and 'Programming?
-- 8. List the title of books written by the author 'Expert' but not books that have the
-- keyword 'database'.

COLLEGE = {CID:NUMBER, CNAME:STRING, STATE:STRING, ENROLLMENT:STRING}
STUDENT = {SID:NUMBER, SNAME:STRING, GPA:NUMBER}
APPLY = {SID:NUMBER, CID:NUMBER, MAJOR:STRING, DECISION:STRING}

-- 1) SID and sName of all Students with GPA > 2.7
-- 2) All students who applied to CSUS and have been accepted
-- 3) Name and GPA of all students who are computer science majors, applied to colleges in
-- California and were rejected (Cross product, Natural Join and Theta join)
-- 4) ID and name of students who did not apply anywhere
-- 5) Names that are both college names and student names
-- 6) Pairs of colleges in the same state

Some Relational Algebraic Equivalence Transformation rules

$$\sigma_{p \wedge q \wedge r}(R) = \sigma_p(\sigma_q(\sigma_r(R)))$$

Example:

$$\sigma_{deptno=10 \wedge sal>1000}(Emp) = \sigma_{deptno=10}(\sigma_{sal>1000}(Emp))$$

==========

$$\Pi_{A_i, \ldots, A_m}(\sigma_p(R)) = \sigma_p(\Pi_{A_i, \ldots, A_m}(R))$$

where $p \in \{A_1, A_2, \ldots, A_m\}$

Example: Selection predicate (p) is only made up of projected attributes

$$\Pi_{name, job}(\sigma_{name='Smith'}(Emp)) = \sigma_{name='Smith'}(\Pi_{name, job}(Staff))$$

==========

$$\sigma_p(\sigma_q(R)) = \sigma_q(\sigma_p(R))$$

Example:

$$\sigma_{sal>1000}(\sigma_{deptno=10}(Emp)) = \sigma_{deptno=10}(\sigma_{sal>1000}(Emp))$$

Division

- Notation: $r \div s$

- Example: given the relations r, s:

r

A	B	C	D	E
α	a	α	a	1
α	a	γ	a	1
α	a	γ	b	1
β	a	γ	a	1
β	a	γ	b	3
γ	a	γ	a	1
γ	a	γ	b	1
γ	a	β	b	1

s

D	E
a	1
b	1

$r \div s$

A	B	C
α	a	γ
γ	a	γ

Consider a schema with two relations, *R*(A, B) and *S*(B, C), where all values are integers. Make no assumptions about keys. Consider the following three relational algebra expressions. Which two are the same?

a. $\pi_{A,C}(R \bowtie \sigma_{B=1} S)$

b. $\pi_A(\sigma_{B=1} R) \times \pi_C(\sigma_{B=1} S)$

c. $\pi_{A,C}(\pi_A R \times \sigma_{B=1} S)$

Consider a relation *R*(A, B) that contains r tuples, and a relation *S*(B, C) that contains s tuples; assume r > 0 and s > 0. For each of the following relational algebra expressions, state in terms of r and s the minimum and maximum

number of tuples that could be in the result of the expression.

a. $R \cup \rho_{S(A,B)} S$

b. $\pi_{A,C}(R \bowtie S)$

c. $\pi_B R - (\pi_B R - \pi_B S)$

d. $(R \bowtie R) \bowtie R$

e. $\sigma_{A>B} R \cup \sigma_{A<B} R$