Relational Algebra

Molina, Ullman, Widom

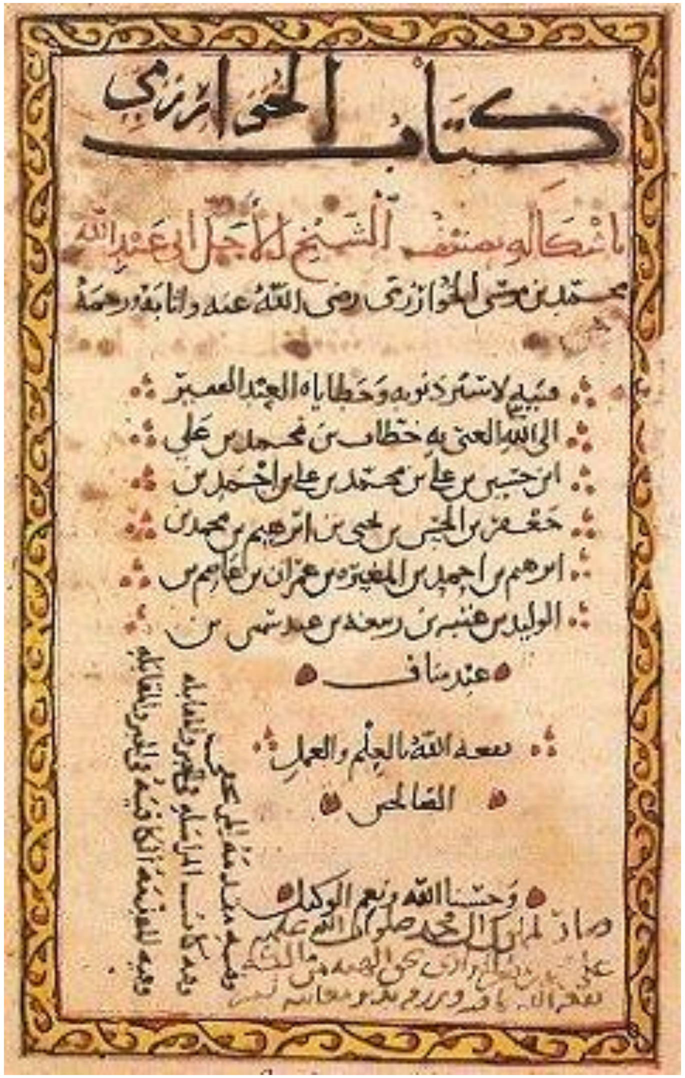
Database Management: Complete Book, Chapters 2 & 5

Algebra

- 2 "fathers of algebra“:
  - where algebra ≡ theory of equations
    → Greek *Diophantus*
  - where algebra ≡ rules for manipulating & solving equations
    → Persian *al-Khwarizmi*

Xorazm, Usbekistan

- Source: Wikipedia



What is “Algebra”?

- Mathematical system consisting of:
  - Operands - variables or values from which new values can be constructed
  - Operators - symbols denoting procedures that construct new values from given values
  - Ex: ((x + 7)/(2 - 3)) + x
- Algebra A = (C,OP)
  -- "simplest" mathematical structure:
  - C nonempty carrier set (=value set)
  - OP nonempty operation set
  - C closed under OP expressions

Selection

- R1 := σ_C (R2)
  - *C* : condition on attributes of R2.
  - R1 is all those tuples of R2 that satisfy *C*.

sid	name	login	gpa
53666	Jones	jones@cs	3.4
53688	Smith	smith@eecs	3.2
53650	Smith	smith@math	3.8

$\sigma_{gpa<3.8}$(Students):

sid	name	login	gpa
53666	Jones	jones@cs	3.4
53688	Smith	smith@eecs	3.2

Selection: Observations

- unary operation: 1 table
- conditions apply to each tuple individually
  - condition cannot span tuples (how to do that?)
- degree of $\sigma_C(R)$ = degree of R
  - Cardinality?
- Select is commutative: $\sigma_{C1}(\sigma_{C2}(R)) = \sigma_{C2}(\sigma_{C1}(R))$
  - Ex: $\sigma_{S.sid=E.sid}(\sigma_{E.cid=C.cid}(R)) = \sigma_{E.cid=C.cid}(\sigma_{S.sid=E.sid}(R))$

Projection

- R1 := π_{attr}(R2)
  - *attr* : list of attributes from R2 schema
- For each tuple of R2:
  - extract attributes from list attr in order specified (!) → R1 tuple
- Eliminate duplicate tuples

sid	name	login	gpa
53666	Jones	jones@cs	3.4
53688	Smith	smith@eecs	3.2
53650	Smith	smith@math	3.8

$\pi_{name,login}$(Students) =

name	login
Jones	jones@cs
Smith	smith@eecs

Projection: Observations

- Unary operation: 1 table
- removes duplicates in result
  - Cardinality?
  - Degree?
- Project is not commutative
- Sample algebraic law: $\pi_{L1}(\pi_{L2}(R)) = \pi_{L1}(R)$ if $L1 \subseteq L2$
  - else incorrect expression, syntax error
  - Ex: $\pi_{name}(\pi_{name,gpa}(R)) = \pi_{name}(R)$

Exercises

- $\pi_{Name,login}(\sigma_{gpa=3.8}(Students)) = ?$

sid	name	login	gpa
53666	Jones	jones@cs	3.4
53688	Smith	smith@eecs	3.2
53650	Smith	smith@math	3.8

- "name and rating for sailors with rating > 8"
  - Note explicit operation sequence!

Cartesian Product

- project, select operators operate on single relation
- Cartesian product combines two: R3 = R1 x R2
  - Pair each tuple t1 ∈ R1 with each tuple t2 ∈ R2
  - Concatenation t1,t2 is a tuple of R3
  - Schema of R3 = attributes of R1 and then R2, in order
  - if attribute *A* of the same name in R1 and R2: use R1.*A* and R2.*A*
- Algebraic laws? Associative; commutative if ignoring attribute order; …

Cross Product ("Cartesian Product")

- Example U := R x S

A	B
1	2
3	4

(a) Relation R

B	C	D
2	5	6
4	7	8
9	10	11

(b) Relation S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

(c) Result $R \times S$

Natural Join

- $T = R \bowtie S$
  - Ex: Reserves $\bowtie_{bid}$ Sailors
- connect two relations:
  - Equate attributes of same name, project out redundant attribute(s)

„natural" = remove duplicate attribute(s)

A	B
1	2
3	4

(a) Relation R

B	C	D
2	5	6
4	7	8
9	10	11

(b) Relation S

A	R.B	S.B	C	D
1	2	2	5	6
1	2	4	7	8
1	2	9	10	11
3	4	2	5	6
3	4	4	7	8
3	4	9	10	11

(c) Result $R \times S$

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

$R \bowtie S$

Generalizing Join

- $T = R \bowtie_C S$
  - First build R x S, then apply σ_C
- Generalization of equi-join: A θ B where θ one of =, <, ...
  - Today, more general: σ_C can be any predicate
- Common join types:
  - Left join, right join, natural join, self join, ...

Relational Algebra: Summary

= Mathematical definition of relations + operators
  - Query = Algebraic expression
- Relational algebra A = (R,OP) with relation $R = A_1 \times \ldots \times A_n$, $OP=\{\pi,\sigma,\times\}$
  - Projection: $\pi_{attr}(R) = \{ r.attr \mid r \in R \}$
  - Selection: $\sigma_p(R) = \{ r \mid r \in R, p(r) \}$
  - Cross product: $R_1 \text{ x } R_2 = \{(r_{11}, r_{12}, \ldots, r_{21}, r_{22}, \ldots) \mid (r_{11}, r_{12}, \ldots) \in R_1, (r_{21}, r_{22}, \ldots) \in R_2 \}$
  - Further: set operations, join, ...
- Set + predicate notation = Relational Calculus
  - Equally powerful as Relational Algebra – proven by E Codd

Relational Calculus

- Tuple variable = variable over some relation schema
- Query Q = { T | T∈R, p(T) }
  - R relation schema, p(T) predicate over T
- Example 1: "sailors with rating above 8"
  - Sailors = sid:int × sname:string × rating:int × age:float

  = $\{ S \mid S \in Sailors \wedge S.rating > 8 \}$
- Example 2: "names of sailors who have reserved boat #103":
  - Reserves = sid:int × bid:int × day:date

  = $\{ S.sname \mid \exists S \in Sailors\ \exists R \in Reserves: R.sid = S.sid \wedge R.bid = 103 \}$

Comparison of Relational Math

- Relational algebra
  - set-based formalization of selection, projection, cross product (no aggregation!)
  - Operation oriented = procedural = imperative; therefore basis of optimization
- Relational calculus
  - Same, but in predicate logic
  - Describing result = declarative; therefore basis of SQL semantics
- Equally powerful
  - proven by E Codd in 1970