Relational Algebra

Database Query Languages

- Use “Campus” schema
- Given a database, ask questions, get data as answers
  - Ex: Get all students with GPA > 3.7 who applied to Berkeley and Stanford and nowhere else
  - Ex: Get all humanities departments at campuses in Florida with < 1000 applicants
  - Ex: Get the campus with highest average accept rate over the last five years
- Some questions are easy to pose, some are not
- Some questions are easy for DBMS to answer, some are not.
- "Query language" also used to update the database

Relational Query Languages

- Formal: relational algebra, relational calculus, Datalog
- Actual: SQL, Quel, Query-by-Example (QBE)
- In **ALL** languages, a query is executed over a set of relations, <u>get single relation as the result</u>

Relational Algebra

- Notation for describing queries in the relational model
- Relational model has concrete set of “standard” operations
- Operations are not “Turing Complete”
  - Not a defect, helps with query processing and optimization
  - FYI, a language is Turing Complete if it is powerful enough to implement any Turing machine. It's widely believed that Turing machines can do any calculation that can be performed by a modern computer program
- Start by introducing operations of relational algebra, SQL next
- Algebra applies to sets of tuples, i.e., relations
  - Commercial DBMS use different notation of relations which are multisets

Relational Algebra

- Construct new relations from old ones
  - Set of operators
  - Relations are operands
- Build progressively more complex expressions by applying operators to relations or to rela. algebra expressions (which are relations as well)
- Query is an expression of relational algebra
  - First concrete example of a query language
- Four broad classes of operations
  - Set operations, selection operations, operations that combine data from two relations, rename operation

Sample Relational Schema

Movie (<u>Title,Year</u>,length,filmType, studioName,producerC#)

StarsIn (<u>MovieTitle,MovieYear,StarName</u>)

MovieStar(<u>Name</u>,address,gender,birthdate)

MovieExec(name,address,<u>Cert#</u>,netWorth)

Studio (<u>Name</u>,address,presC#)

Basics

- Operations of traditional relational algebra fall into four broad classes:

1. Set operations
2. Operations that remove parts of a relation
3. Operations that combine tuples of two relations
4. Renaming

Set Operations

`Union` (binary, commutative, associative)

- $R \cup S$

`Intersection` (binary, commutative, associative)

- $R \cap S$

`Set Difference` (binary)

- R - S
  - Set of elements in R but not in S
  - $R\text{-}S \neq S\text{-}R$!!
- $R(A_1,A_2,\ldots,A_n)$, $S(B_1,B_2,\ldots,B_n)$ must be union compatible
  - R and S are of the same degree
  - for each i, $dom(A_i) = dom(B_i)$
  - Columns of R and S must be ordered so that order of attributes is same for both relations

Example

R			
name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

S			
name	*address*	*gender*	*birthdate*
Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77

Sample Operations

$R \cap S$	*name*	*address*	*gender*	*birthdate*
	Carrie Fisher	123 Maple St., Hollywood	F	9/9/99

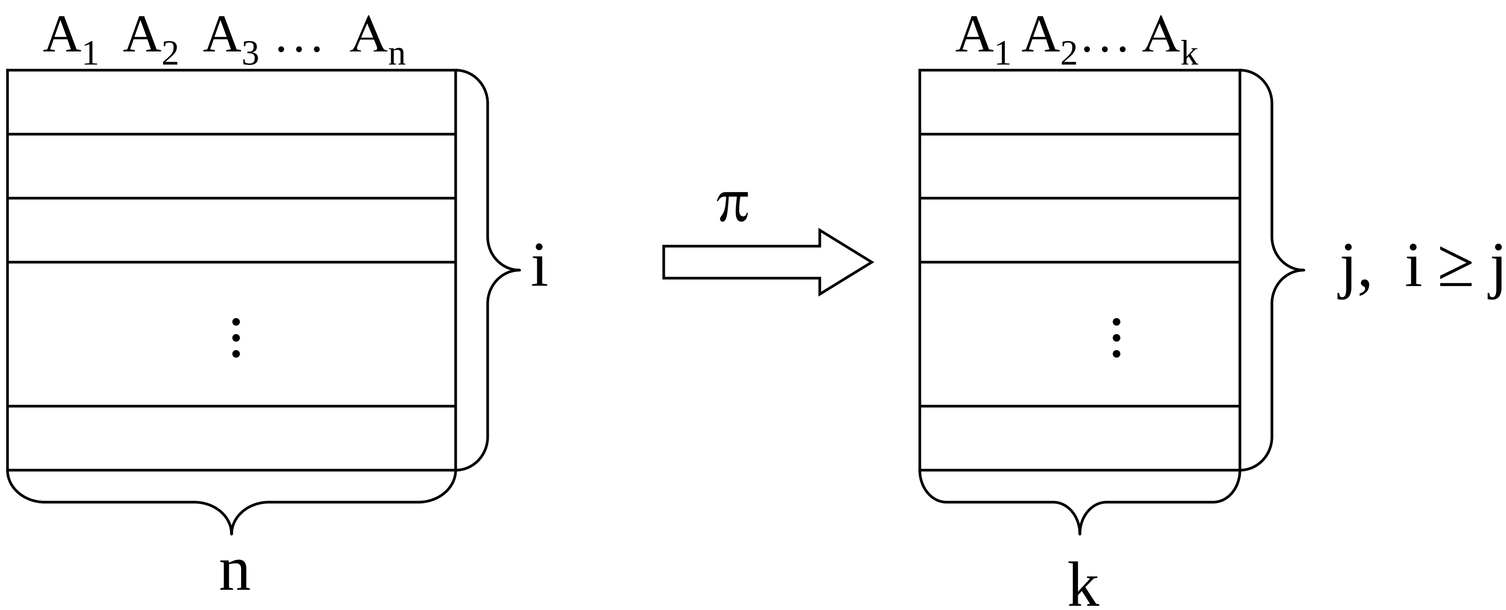
$R \cup S$	*name*	*address*	*gender*	*birthdate*
	Carrie Fisher	123 Maple St., Hollywood	F	9/9/99
	Harrison Ford	789 Palm Dr., Beverly Hills	M	7/7/77
	Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

$R - S$	*name*	*address*	*gender*	*birthdate*
	Mark Hamil	456 Oak Rd., Brentwood	M	8/8/88

Relational Operator: Project

`Project` (unary)

- $\pi_{\text{<attr list>}}$ (R)
- <attr list> is a list of attributes (columns) from R only
- Ex: $\pi_{\text{title, year, length}}$ (Movie) “horizontal restriction”



Project

- PROJECT can produce many tuples with same value
  - Relational algebra semantics says remove duplicates
  - SQL does not -- one difference between formal and actual query languages

Relational Operator: Select

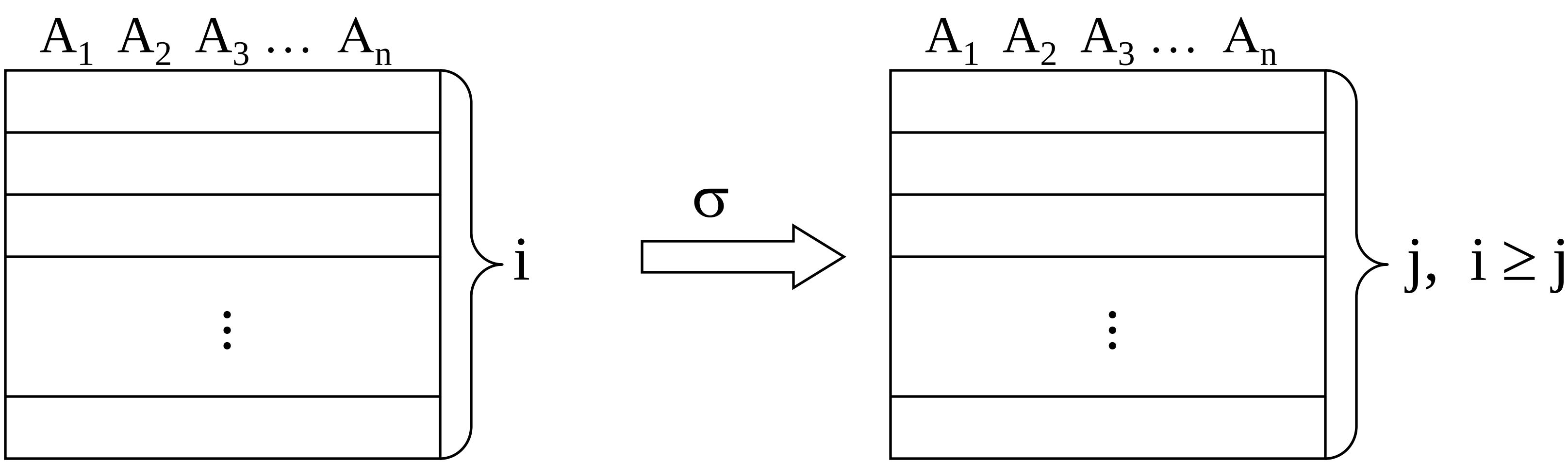
`Select` or `Restrict` (unary, commutative)

- $\sigma_{\text{<predicate>}}$ (R)
- <predicate> is a conditional expression of the type that we are familiar with from conventional programming languages
  - <attribute> <op> <attribute>
  - <attribute> <op> <constant>
  - attribute in R
  - op $\in \{=, \neq, <, >, \leq, \ldots, \text{AND}, \text{OR}\}$
- Ex: $\sigma_{\text{length} \geq 100}$ (Movie) vertical restriction"

Pictorially

Movie	*title*	*year*	*length*	*filmType*
	Star Wars	1977	124	color
	Mighty Ducks	1991	104	color
	Wayne's World	1992	95	color

result set



# of selected tuples is referred to as the selectivity of the condition

Cartesian Product

`Cartesian Product` (binary, commutative, associative)

- R x S
- Sets of all pairs that can be formed by choosing the first element of the pair to be any element of R, the second any element of S
- Relation schema is union of schemas for R and S
- Resulting schema may be ambiguous
  - Use R.A or S.A to disambiguate an attribute that occurs in both schemas

Example

R

A	B
1	2
3	4

x

S

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

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

Join Operations

`Natural Join` (binary)

- R join S
- Match only those tuples from R and S that agree in whatever attributes are common to the schemas of R and S
  - If r and s from r(R) and s(S) are successfully paired, result is called a *joined tuple*
- This join operation is the same we used in earlier section to recombine relations that had been projected onto two subsets of their attributes (e.g., as a result of a BCNF decomposition)

Example

R

A	B
1	2
3	4

join

S

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

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

- Resulting schema has attributes from R, either R or S (i.e., joining attribute(s)), and S
- Tuples that fail to pair with any tuple of the other relation are called *dangling tuples*

Join Operations

`Theta Join` (binary)

- R $join_C$ S, where C is an arbitrary join condition
- Step 1: take the product of R and S
- Step 2: Select from the product only those tuples that satisfy condition C
- As with the product operation, the schema for the result is the union of the schemas of R and S

Example

U

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

$\text{join}_{A<D \text{ AND } U.B \neq V.B}$

V

B	C	D
2	3	4
2	3	5
7	8	10

⇒

A	U.B	U.C	V.B	V.C	D
1	2	3	7	8	10

Final Word on Join

- DBMS often implements theta-join as basic operation
- Use of term "join" in implementation circles usually refers to theta-join or sometimes to cross-product