Relational Algebra

CS411 Database Systems

Kevin C. Chang

Announcements

- If you are waiting to register for this class.

Game override,

Question 1:
How to "query" a database?

- Goal: specify what we want from our database

  Find all the employees who earn more than $50,000 and pay taxes in Champaign-Urbana.

What I want

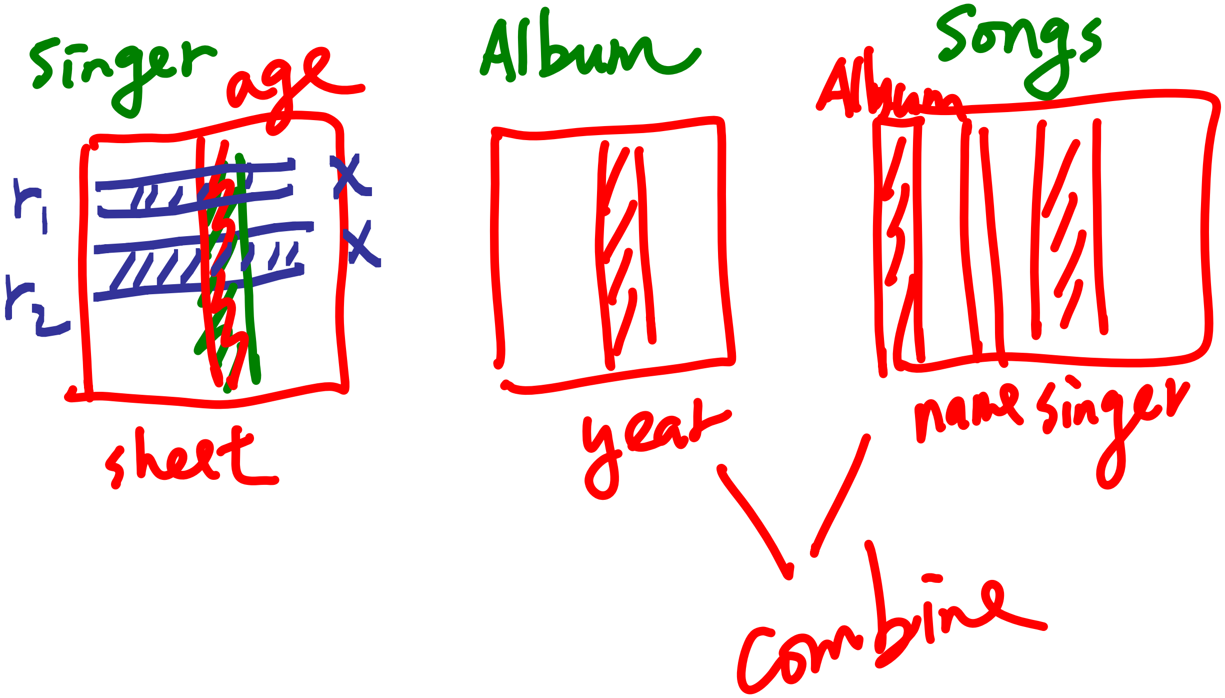
Not How

- Write in C++/Java?

- Use *high-level query languages:*
  - Theoretical: Relational Algebra, Datalog
  - Practical: SQL

- Relational algebra: a basic set of operations on relations that provide the *basic principles*.

Question 2: *What kinds of "computation" can we do on relations?*



What is an "Algebra"

Relations.

+, −, ×, ÷

Operator

Arim. Algebra.

operand.

"numbers"

- 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.

Relations (and only)

Q: Example algebra?

- Arithmetic algebra. Linear algebra.

- What are operands?

- What are operators?

What is Relational Algebra?

- An algebra whose operands are relations or variables that represent relations.
- Operators are designed to do common things that we need to do with relations in a database.
- The result is an algebra that can be used as a *query language* for relations.

Relational Algebra at a Glance

- Operators: relations as input, new relation as output
- Basic RA operators:
  - Basic Set Operations
    - union, difference (no intersection, no complement)
  - Selection: σ
  - Projection: π
  - Cartesian Product: X
  - Renaming: ρ
- Derived operations:
  - Intersection, complement
  - Joins (natural, equi-join, theta join, semi-join)

Set Operations

- Union, difference
- Binary operations

Set Operations: Union

Arbitrary Set ?

Set Set

- Union: all tuples in R1 or R2
- Notation: R1 U R2
- R1, R2 must have the same schema
- R1 U R2 has the same schema as R1, R2
- Example:
  - ActiveEmployees U RetiredEmployees

$R_1 \cup R_2$

Follow Schema

R_1 :

Homogeneous

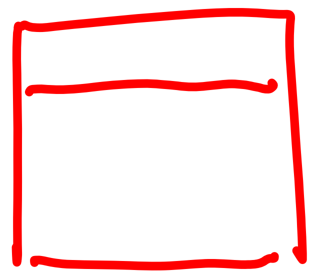
Set

All Employees

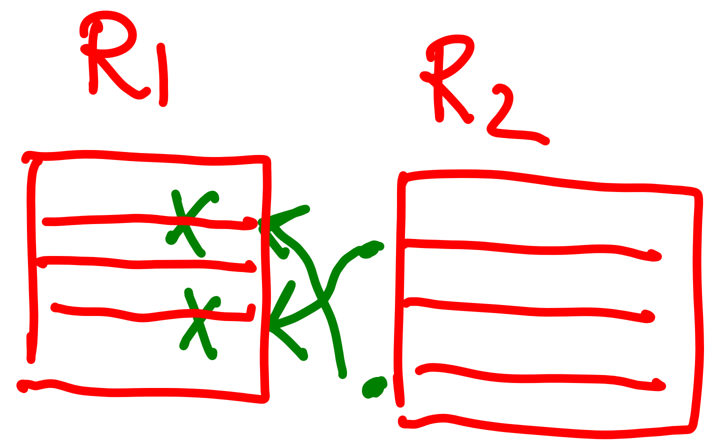
x Tabel

{ "Lady GG",
"ipad 5",
"... "

set of tuples



Set Operations: Difference



- Difference: all tuples in R1 and not in R2
- Notation: R1 – R2
- R1, R2 must have the same schema
- R1 - R2 has the same schema as R1, R2
- Example
  - AllEmployees - RetiredEmployees

R1 R2

Active Employees

store?

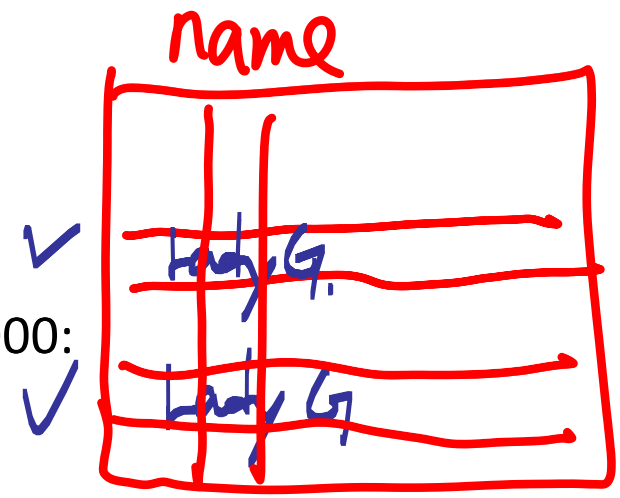
compute?

Selection σ

sigma

$\sigma_{name = \text{"Lady GG"}}$ (Singer)

- Returns all tuples which satisfy a condition
- Notation: $\sigma_c(R)$
- c is a condition: =, <, >, and, or, not
- Output schema: same as input schema



- Find all employees with salary more than $40,000:
  - $\sigma_{salary>40000}(\text{Employee})$

c : salary > 40000

Selection Example

SSN	Name	DepartmentID	Salary
92881	John	1	30,000
32877	Tony	1	32,000
44544	Amy	2	45,000

$$\sigma_{salary>40000}(Employee) = ?$$

Projection π

- Unary operation: returns certain columns
- Eliminates duplicate tuples!
- Notation: $\pi_{A_1,\ldots,A_n}(R)$
- Condition:
  - Suppose input schema $R(B_1, \ldots, B_m)$
  - $\{A_1, \ldots, A_n\} \subseteq \{B_1, \ldots, B_m\}$
- Output schema $S(A_1, \ldots, A_n)$

- Example: $\pi_{SSN,Name}(Employee)$

Projection Example

SSN	Name	DepartmentID	Salary
92881	John	1	30,000
32877	Tony	1	32,000
44544	Amy	2	45,000

$$\pi_{SSN,Name}(Employee) = ?$$

Q: Comparing projection and selection?

- Think of relation as a table.
- How are they similar?
- How are they different?
- Why do you need both?

Cartesian Product

- Each tuple in *R1* with each tuple in *R2*
- Notation: $R_1 \times \ R_2$
- Input schemas $R_1(A_1, \dots, A_n), R_2(B_1, \dots, B_m)$.
- Condition: $\{A_1, \dots, A_n\} \cap \{B_1, \dots, B_m\} = \emptyset$
- Output schema is $S(A_1, \dots, A_n, B_1, \dots, B_m)$.

- Example: **Employee x Dependents**
- Very rare in practice (but joins are very common)

Cartesian Product Example

Employee

Name	SSN
John	999999999
Tony	777777777

2

×

x = 4

Dependents

EmployeeSSN	Dname
999999999	Emily
777777777	Joe

2

Employee x Dependents

Name	SSN	EmployeeSSN	Dname
John	999999999	999999999	Emily
~~John~~	~~999999999~~	~~777777777~~	~~Joe~~
~~Tony~~	~~777777777~~	~~999999999~~	~~Emily~~
Tony	777777777	777777777	Joe