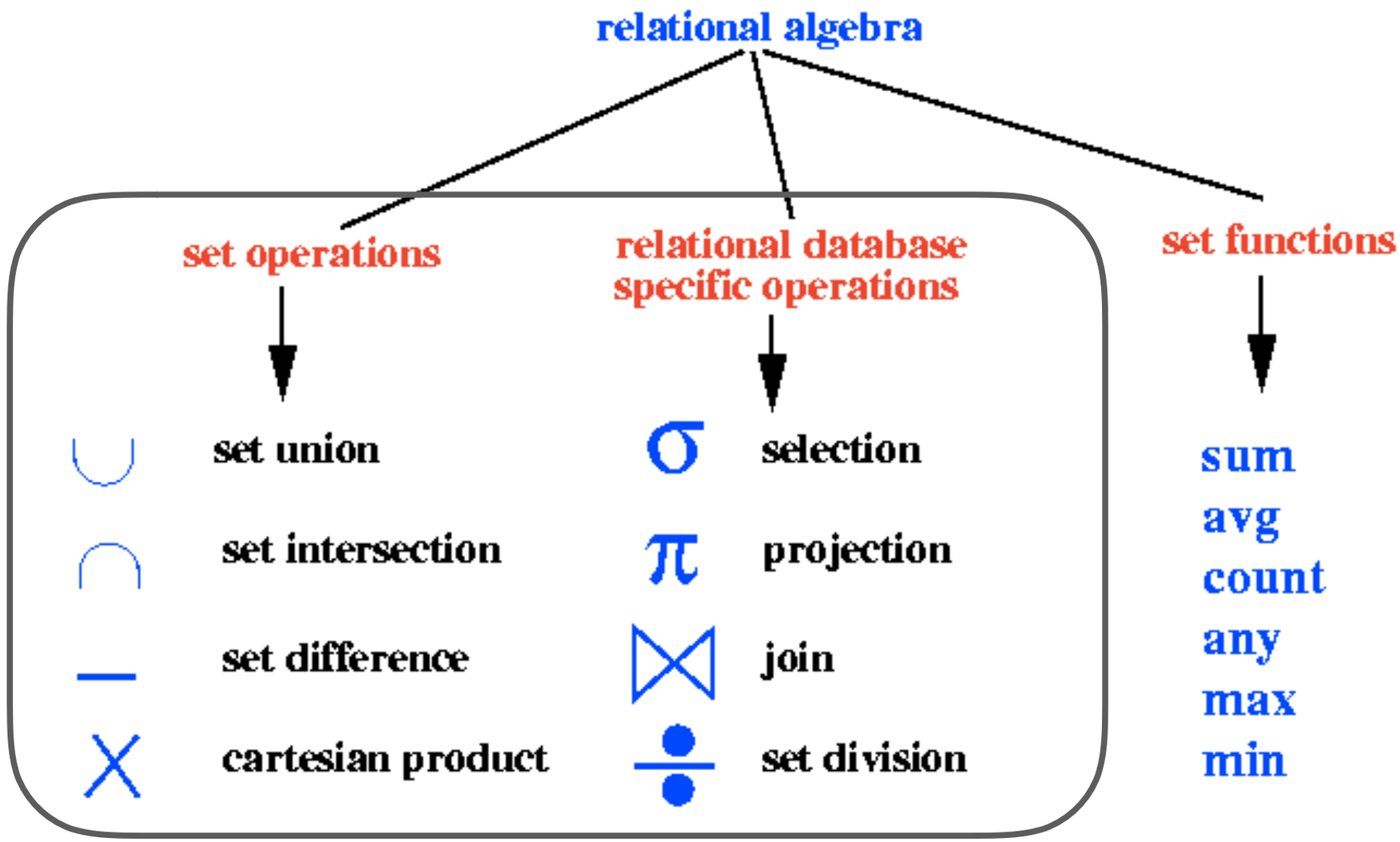
Relational Algebra (II) & Calculus

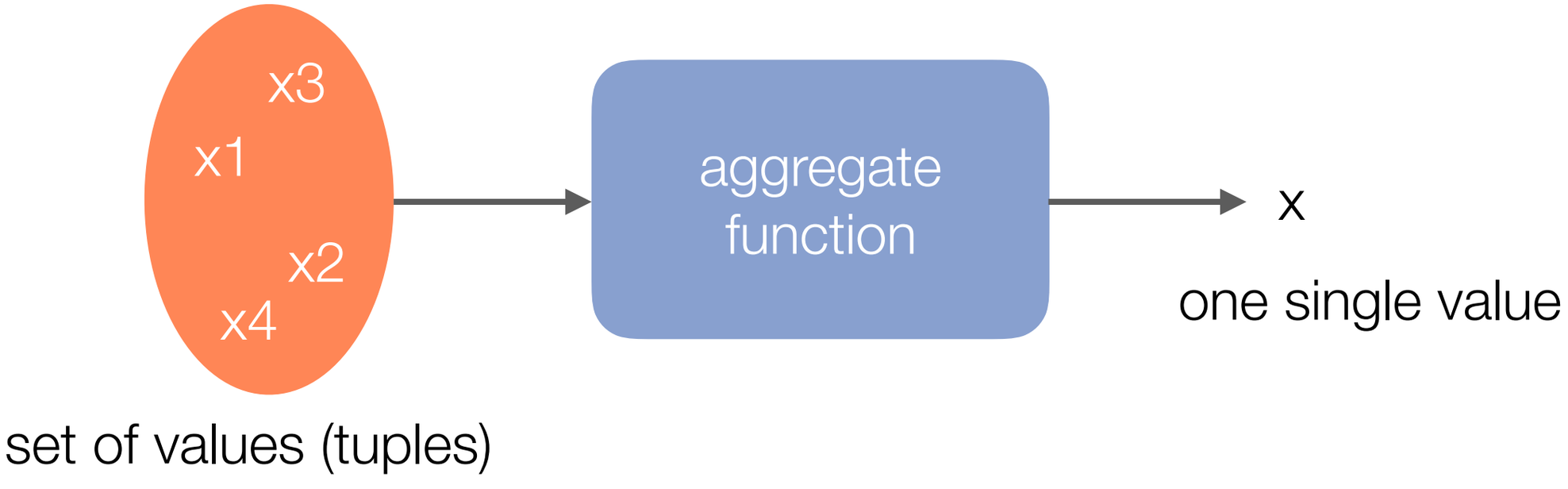
CS 377: Database Systems

Recap: Relational Algebra Part I



Set (Aggregate) Functions

- Operates on a set of values and produce a single value
- Can also be known as aggregate functions
- Common functions include SUM, AVERAGE, MAXIMUM, MINIMUM, and COUNT



Example: Set Functions

A = {1, 4, 5, 10, 15}

Function	Description	Value
sum(A)	sum of all values in the (numeric) set	35
avg(A)	average of all values in the (numeric) set	7
max(A)	maximum value of all values in the set	15
min(A)	minimum value of all values in the set	1
any(A)	TRUE if set is not empty, otherwise FALSE	TRUE
count(A)	cardinality (number of elements) of set	5

Additional Operations: Generalized Projection

- Allows functions of attributes to be included in the projection list

$$\pi_{f_1(a_1),f_2(a_2),\cdots,f_n(a_n)}(R)$$

- Examples:

$$\pi_{\text{LNAME,FNAME,SALARY}*1.03}(\text{EMPLOYEE})$$

$$\pi_{\text{SSN,FNAME,AGE}/2+7,\text{SEX}}(\text{EMPLOYEE})$$

Additional Operations: Group By Aggregate

- Groups are formed using one more attribute value(s)
- Aggregate functions applied independently to each group
- Examples:
  - How many people bought an iPad?
  - What is the average age of students in the Database Systems class?
  - What is the average salary of the different departments?

Example: Group By Aggregate

SSN	FName	Other	Sex	DNo	Salary
111-11-1111	John	…	M	4	50,000
242-12-2340	Mary	…	F	5	60,000
222-22-2222	James	…	M	5	80,000
333-33-3333	Jake	…	M	4	60,000

Group by DNO

111-11-1111	John	…	M	4	50,000
333-33-3333	Jake	…	M	4	60,000

avg(salary) = 55,0000

242-12-2340	Mary	…	F	5	60,000
222-22-2222	James	…	M	5	80,000

avg(salary) = 70,0000

Group By Aggregate Operation

- Notation: $_{a_1,a_2,\cdots,a_N}\mathcal{F}_{f_1(a_1),f_2(a_2),\cdots,f_M(a_M)}(R)$
  - a_1, a_2, …, a_N = attributes used to form groups
  - $f_1(a_1)$, $f_2(a_2)$, …, $f_M(a_M)$ = set functions applied on each group
- Result is always a relation with the following attributes:
  - Grouping attributes (to differentiate the tuples)
  - Set function values (attributes named after function name)

a1	a2	…	aN	f1	f2	…	fM

Example: Group By Aggregate (2)

SSN	FName	Other	Sex	DNo	Salary
111-11-1111	John	…	M	4	50,000
242-12-2340	Mary	…	F	5	60,000
222-22-2222	James	…	M	5	80,000
333-33-3333	Jake	…	M	4	60,000

$$_{\mathrm{DNO,Sex}}\mathcal{F}_{\mathrm{avg(Salary),count(SSN)}}(\mathrm{EMPLOYEE})$$

DNo	Sex	Avg	Count
4	M	55,000	2
5	M	80,000	1
5	F	60,000	1

No tuple with DNO=4, Sex='F' because group (set) is empty!

Example: Group By Aggregate (3)

SSN	FName	Other	Sex	DNo	Salary
111-11-1111	John	…	M	4	50,000
242-12-2340	Mary	…	F	5	60,000
222-22-2222	James	…	M	5	80,000
333-33-3333	Jake	…	M	4	60,000

$$\mathcal{F}_{\text{avg(Salary)},\text{count(SSN)}}(\text{EMPLOYEE})$$

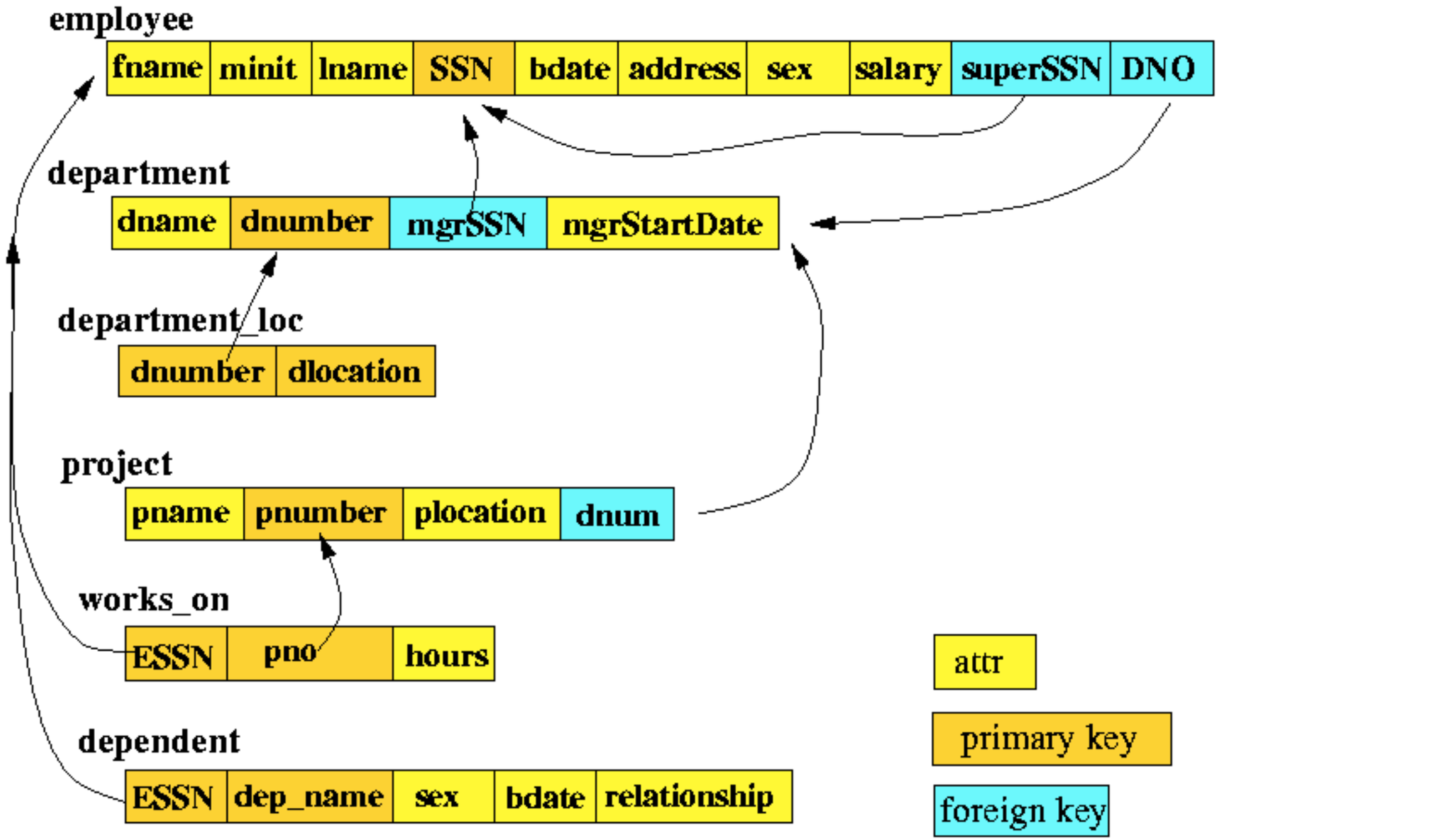
Avg	Count
62,500	4

When no grouping attributes are specified, the set function is applied on ONE group with all the tuples in the relation!

Relational Algebra Operations

Operation	Notation	Purpose
SELECT	$\sigma_{<\text{selection condition}>}(R)$	Selects all tuples that satisfy the selection condition from a relation R
PROJECT	$\pi_{<\text{atttribute list}>}(R)$	New relation with subset of attributes of R and removes duplicate tuples
THETA_JOIN	$R_1 \bowtie_{<\text{join condition}>} R_2$	All combinations of tuples from R_1 and R_2 that satisfy the join condition
EQUIJOIN	$R_1 \bowtie_{<\text{join condition}>} R_2$	Theta join with only equality join comparisons
NATURAL JOIN	$R_1 *_{<\text{join condition}>} R_2$	Equijoin except join attributes of R_2 are not included in the resulting relation
UNION	$R_1 \cup R_2$	Relation that includes all tuples in R_1 or R_2
INTERSECTION	$R_1 \cap R_2$	Relation that includes all tuples in both R_1 and R_2
DIFFERENCE	$R_1 - R_2$	Relation that includes all tuples in R_1 that are not in R_2
CARTESIAN PRODUCT	$R_1 \times R_2$	Relation with attributes of R_1 and R_2 and includes tuples with all possible combinations of tuples of R_1 and R_2
DIVISION	$R_1(Z) \div R_2(Y)$	Relation that includes all tuples t[X] in $R_1(Z)$ that appear in R_1 in combination with every tuple from $R_2(Y)$ where $Z = X \cup Y$

Example: Company Database



http://www.mathcs.emory.edu/~cheung/Courses/377/Syllabus/3-Relation/rel-db-design2.html

Example: RA Queries (1)

Find the name and address of all employees who work in the Research department

Example: RA Queries (2)

Find fname and lname of employees who earn more than ‘John Smith’

Example: RA Queries (3)

Find fname and lname of employees who have 2 or more dependents

Example: RA Queries (4)

Find fname and lname of employees who have the most number of dependents

Example: RA Queries (5)

Retrieve the names of employees who have no dependents

Example: RA Queries (6)

List the names of managers who have at least one dependent

Example: RA Queries (7)

Find fname and lname of employees who work on more projects than ‘John Smith’

Example: RA Queries (8)

For each department, show the department name, number of employees, minimum employee salary and maximum employee salary

Example: RA Queries (9)

Find fname and lname of all employees who work on 2 or more projects controlled by the Research department

Example: RA Queries (10)

Find fname and lname of all employees who work on all projects controlled by the Research department

Example: RA Queries (11)

Find fname and lname of all employees who do not work on any projects controlled by the Research department

Example: RA Queries (12)

Find fname and lname of all employees that only work on projects controlled by the Research department

Relational Calculus

- Declarative query language that describes what is to be retrieved rather than how to retrieve it (nonprocedural)

- Two flavors of relational calculus: Tuple relational calculus (TRC) and Domain relational calculus (DRC)

- Relational calculus and relational algebra are logically equivalent (same logical content)

Relational Calculus

- Calculus has variables, constants, comparison operations, logical connectives, and quantifiers

    - TRC: Variables range over (i.e., get bound to) tuples. Similar to SQL

    - DRC: Variables range over domain elements (field values)

    - Both are simple subsets of first-order Logic

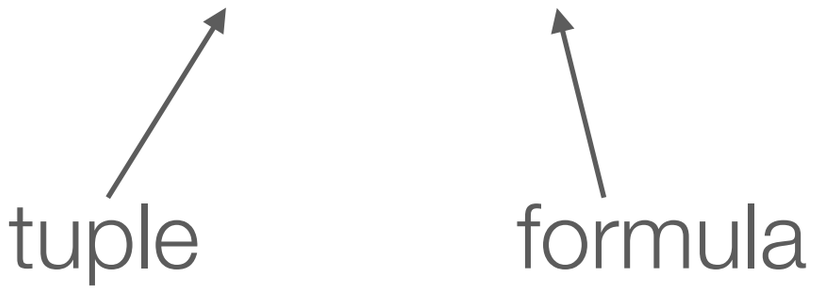
- Expression in calculus are called formulas

Tuple Relational Calculus (TRC)

- Tuple variable: a variable name that represents data tuples in the database
  - Typically denoted using a lower case letter
- Range relation: the relation that is the range for a tuple variable
  - Expression R(t) is evaluated as follows:
    R(t) = true if tuple t is a tuple from the relation R
    R(t) = false if tuple t is not a tuple from the relation R

TRC

- A query in TRC has the form: {t | CONDITION(t) }



- Returns all tuples for which the condition or formula evaluates to true

- Formula is recursively defined, starting with simple atomic formulas and building more complex operators using the logical operators

TRC Formula

- An atomic formula is one of the following:
  - $t \in R$
  - R.a *op* S.b $\quad <, >=, \geq, \leq, \neq$
  - R.a *op* constant
- A formula can be:
  - An atomic formula
  - NOT p, p AND q, p OR q, where p and q are formulas
  - Special quantifiers

TRC Simple Examples

- {t | Employee(t) AND t.salary > 50000}
  - Retrieve all tuples t such that t is a tuple of the relation EMPLOYEE and their salary amount is greater than 50000
- {t.fname, t.lname | Employee(t) AND t.salary > 50000}
  - Retrieve the first and last name of employees whose salary is greater than 50000
- {t.salary | Employee(t) AND t.fname = ‘John’ AND t.lname=‘Smith’}
  - Retrieve the salary of the employee “John Smith”

Special Formula Quantifiers

Two special quantifiers can appear in formulas

- Universal quantifier $(\forall t)\ (\text{Condition}(t))$
  evaluates to true if all tuples t satisfies Condition(t)
  otherwise false

- Existential quantifier $(\exists t)\ (\text{Condition}(t))$
  evaluates to true if there is some (at least one) tuple t
  that satisfies Condition(t)

Free and Bound Variables

- The use of special quantifiers in a formula binds the variable t
    - A variable that is not bound is free
- The variable t that appears to the left of | must be the only free variable in the formula

TRC Example (2)

SAILORS (sid, sname, rating, age)
RESERVES (sid, bid, day)
BOATS (bid, bname, color)

- $\pi_{\text{sname}}(\sigma_{\text{rating}>1}(\text{SAILORS}))$

$$\{t \mid (\exists s)\ (\text{SAILORS}(s)\ \text{AND}\ s.\text{rating} > 1\ \text{AND}\ t.\text{sname} = s.\text{sname})\}$$

free

bound

CONVENTION: the attributes of the free variable t are exactly the ones mentioned in the formula!

TRC Example (3)

Find the department number of the Research department

. $\{d.\text{dno} \mid \text{Department}(d) \text{ AND } d.\text{dname} = \text{‘Research’}\}$

$\{d.\text{dno} \mid \text{Department}(d)$

$\text{AND } ($

$(\exists\ t)$

$(\text{Department}(t)$

$\text{AND } t.\text{dname} = \text{‘Research’}$

$\text{AND } t.\text{dno} = d.\text{dno})$

$)\}$

TRC Example (4)

- List the name and address of all employees who work for the ‘Research’ department

$$\{t.\text{Fname}, t.\text{Lname}, t.\text{Address} \mid \text{EMPLOYEE}(t) \text{ AND } (\exists d)(\text{DEPARTMENT}(d) \text{ AND } d.\text{Dname} = \text{‘Research’ AND } d.\text{Dnumber} = t.\text{Dno})\}$$

- List the names of employees who work on some projects controlled by department number 5

$$\{e.\text{fname}, e.\text{lname} \mid \text{Employee}(e) \text{ AND}((\exists p)\ (\exists w)\ (\text{Project}(p) \text{ AND } \text{Works\_on}(w) \text{ AND } p.\text{dnum} = 5 \text{ AND } p.\text{pnumber} = w.\text{pnum} \text{ AND } w.\text{essn} = e.\text{ssn}))\}$$

TRC Example (4)

Employee(e)

e1, John
e2, Kate
e3, Ann

Works_on(w)

e1, p1
e2, p3
e3, p2

Project(p)

p1, 5
p2, 5
p3, 4

- Run through the employee tuples and make the second condition true, we must find tuples such that p is a Project tuple, w is a Works_on tuple, and it matches the 3 conditions with employee number matching.

- e1 is good since you can find it in all 3 tables and meets the conditions

- e2 is problematic because p3 = 4, which doesn't match our condition

- e3 is also output because the combination exists that can make the second condition true

TRC Example (5)

- List the names of employees who work on all the projects controlled by department number 5

- Solution 1: Projects that are either not controlled by department 5 of e is working on

$$\{e.\text{fname}, e.\text{lname} \mid \text{Employee}(e)$$
$$\text{AND } ((\forall x)\ (\text{NOT}(Project(x))$$
$$\text{OR NOT } (x.dnum = 5)$$
$$\text{OR } ((\exists w)\ (\text{Works\_on}(w)$$
$$\text{AND } w.\text{essn} = e.\text{ssn}$$
$$\text{AND } x.\text{pnumber} = w.\text{pno})))\}$$

TRC Example (5)

- List the names of employees who work on all the projects controlled by department number 5

- Solution 2: There is no project controlled by department 5 that e is not working on

$$\{e.\text{fname}, e.\text{lname} \mid \text{Employee}(e)$$
$$\text{AND } (\text{NOT}(\exists x)(\text{Project}(x)$$
$$\text{AND } (x.\text{dnum} = 5)$$
$$\text{AND } (\text{NOT}(\exists w)(\text{Works\_On}(w)$$
$$\text{AND } w.\text{essn} = e.\text{ssn AND } x.\text{pnumber} = w.\text{pno})))))\}$$

Relational Algebra & Relational Calculus

- (Definition) Expressive power of a query language is the set of all queries that can be written using that query language

- Query language A is more expressive than query language B if the set of all queries written in A is a superset of all queries that can be written in B

- Codd's Theorem: Every relational algebra query can be expressed as a "safe" query in TRC/DRC; the converse is true

  - Relational Algebra and Relational Calculus are equally expressive

Relational Algebra & Calculus: Recap

- Relational Algebra
  - Set Functions
  - Group By Aggregate
- Relational Calculus