Relational Algebra & Calculus

CS 377: Database Systems

Quiz #1

- Question: What is metadata and why is it important?

- Answer:
  Metadata is information about the data such as name, type, size. It is important because it helps achieve physical dependence as programs that use metadata do not need to change if there is a new attribute that is added.
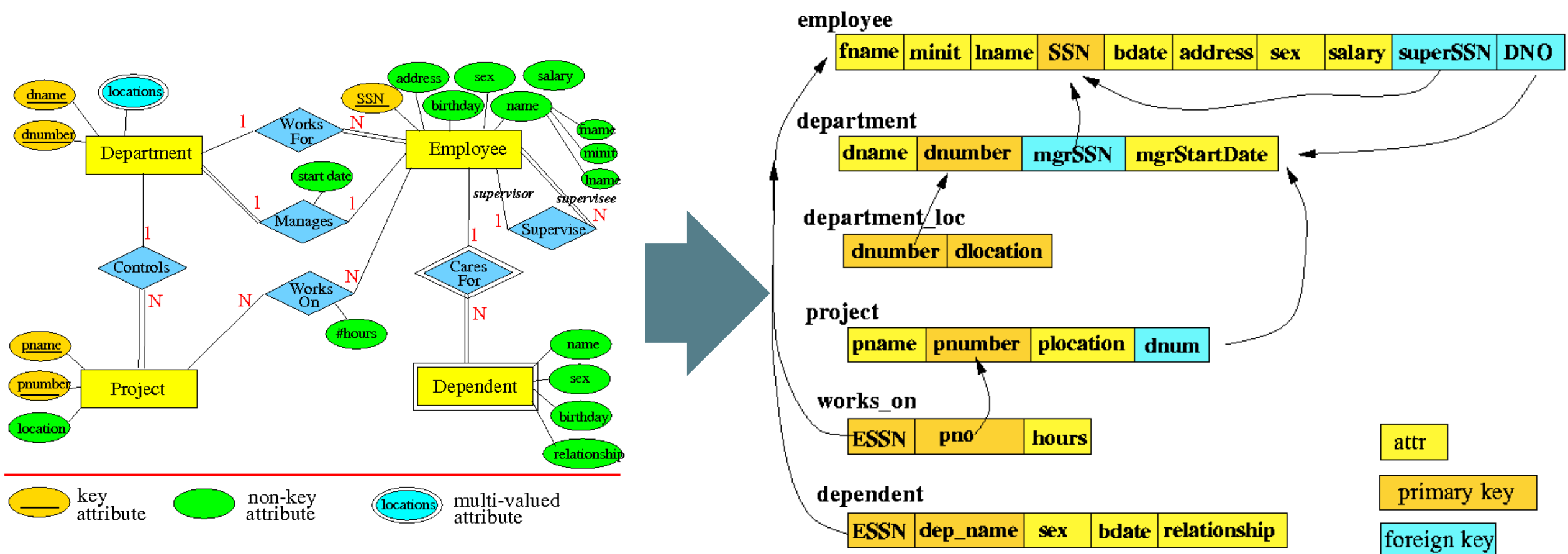
Avg Score: 4.31

ER & Relational Models: Recap

Recap: ER & Relational Model

- ER Model: Design good conceptual models to store information

- Relational Model: Table representation with structures and constraints

- ER model to Relational Model Map

Recap of Last Two Lectures

Now we have a database with relations and data tuples…
What should we do?

Today and Next Class

1. Relational Query Language Overview

2. Relational Algebra

    1. Set operations

    2. Database specific operations

    3. Set functions

3. Relational Calculus

Relational Query Languages

- Manipulation and retrieval of data from a database

- Relational Calculus: Declarative

  - Describe what a user wants rather than how to compute it

- Relational Algebra: Procedural

  - Operational and useful for representing execution plans

Query vs Programming Languages

- Query languages:

  - Are not "Turing complete" (some languages derived from SQL are, but not all)

  - Are not intended to be used for complex calculations

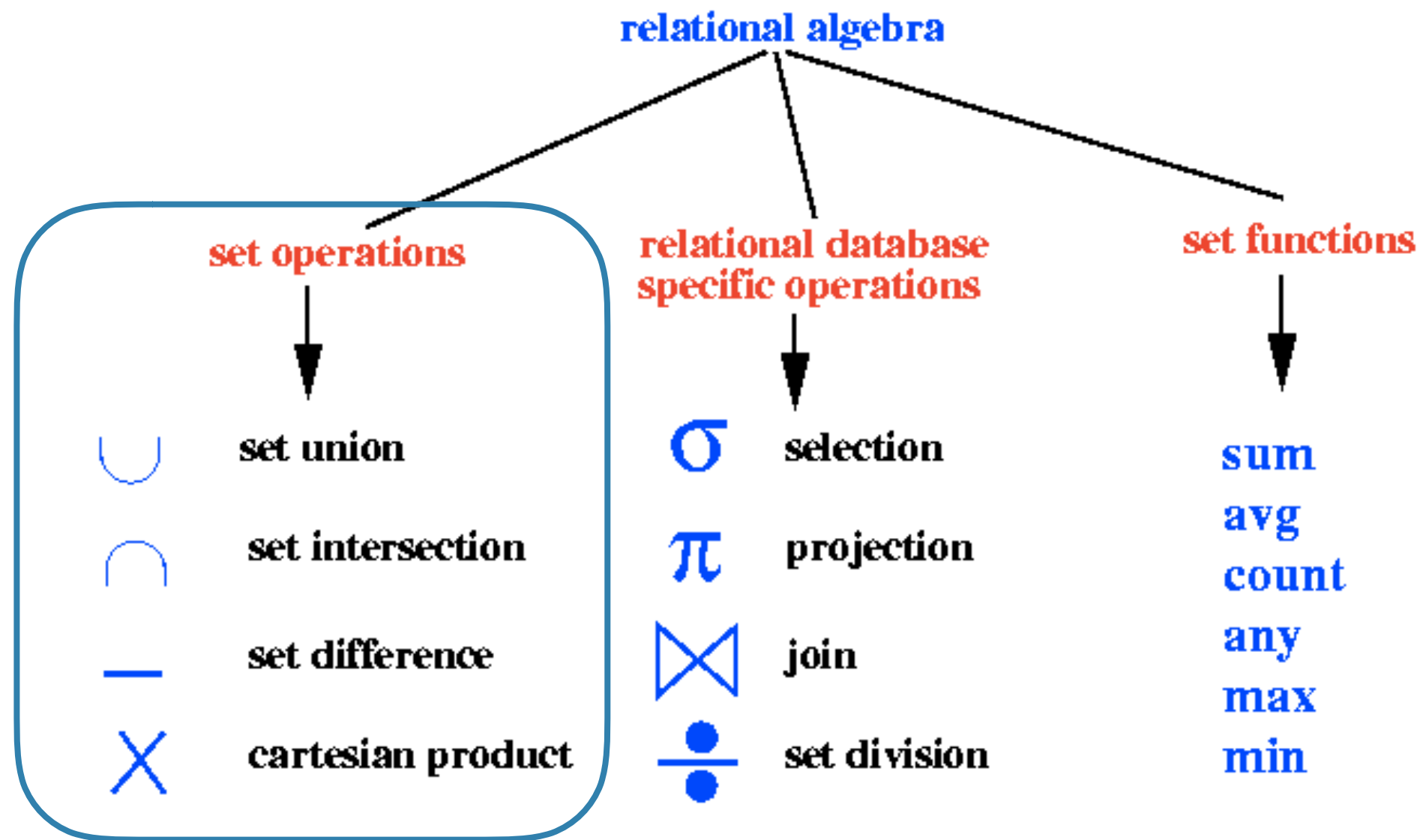 - Supports easy, efficient access to large data sets

Relational Algebra Overview

- Formal foundation for relational model operations

  - Operators do the most common things needed for relations in a database

- Basis for implementing and optimizing queries in RDBMS

- Basis for practical query languages such as SQL

Relational Algebra Terms

- **Relational algebra** is a mathematical language with a basic set of operations for manipulating relations

- A **relational algebra operation** operates on one or more relations and results in a new relation, which can be further manipulated using operations of the same algebra

- A **relational algebra expression** is a sequence of relational algebra operations

Relational Algebra Language

relational algebra

set operations
- ∪ set union
- ∩ set intersection
- — set difference
- ✕ cartesian product

relational database specific operations
- σ selection
- π projection
- ⋈ join
- ÷ set division

set functions
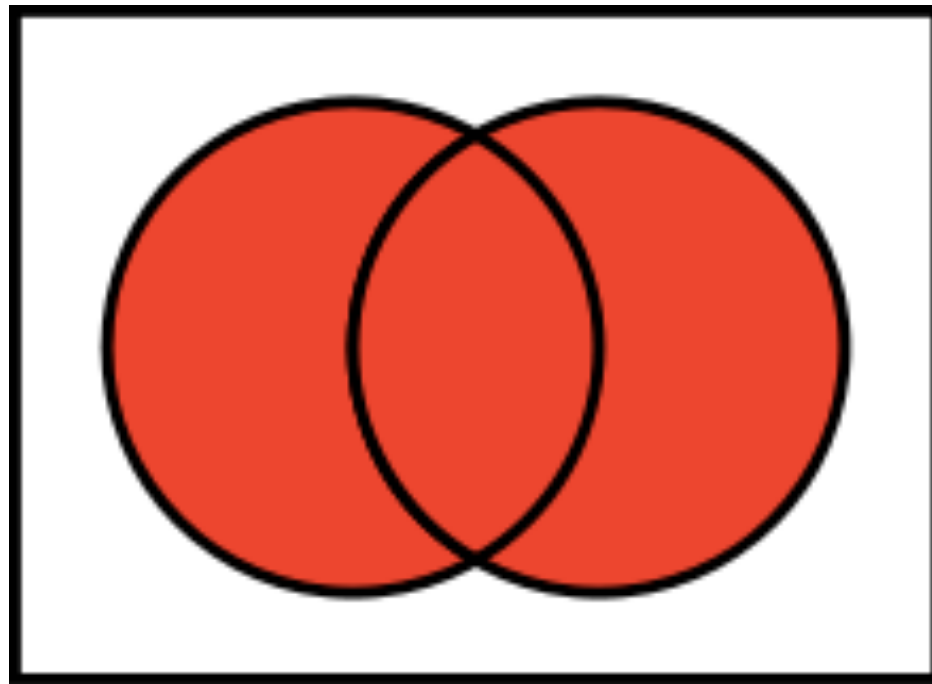- sum
- avg
- count
- any
- max
- min

Set theory

Union Operation

Set of all distinct elements in the collection of sets

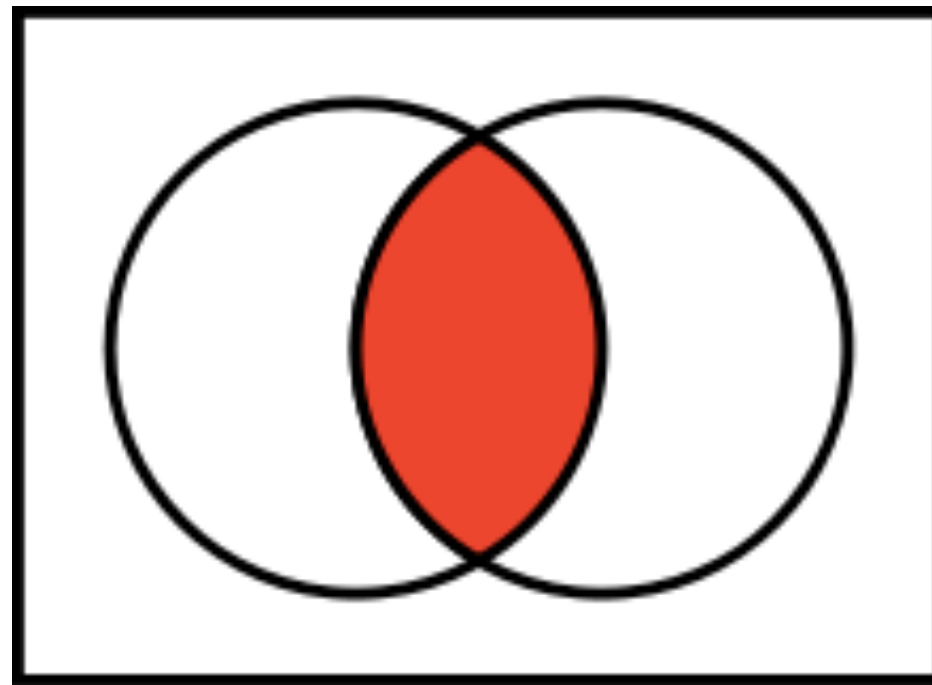$$A = \{i, j, k\}, B = \{k, x, y\}, A \cup B = \{i, j, k, x, y\}$$

Intersection Operation

Set of elements that belong in both sets

$$A = \{i, j, k\}, B = \{k, x, y\}, A \cap B = \{k\}$$

Difference Operation

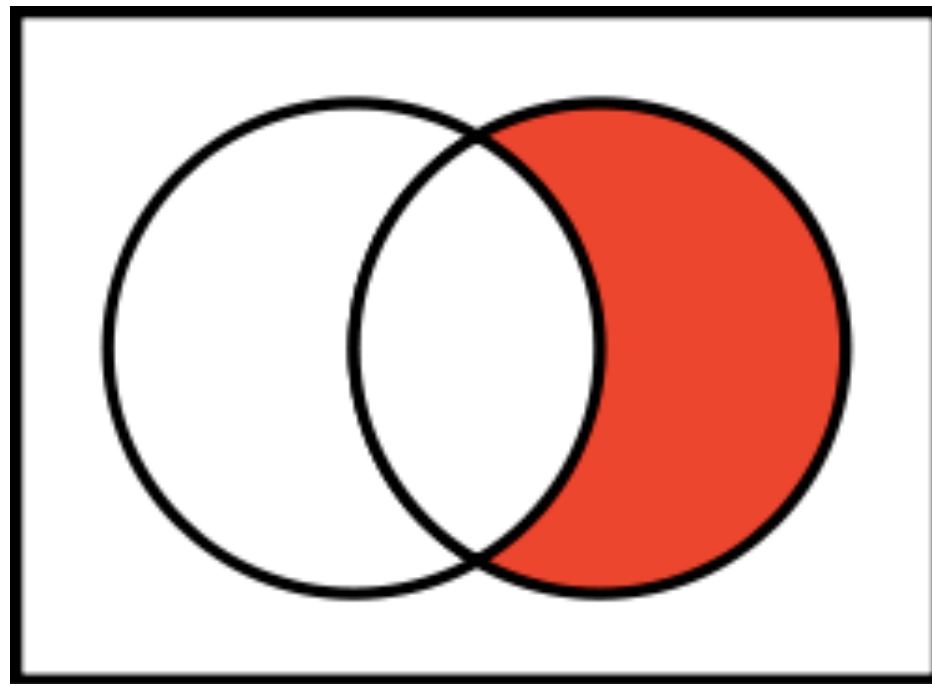Set of elements in A, but not in B

$$A = \{i, j, k\}, B = \{k, x, y\}, A - B = \{i, j\}$$

Cartesian Product Operation

Set of all ordered pairs from each set

$$A = \{i, j, k\}, B = \{k, x, y\}, A \times B = \{(i, k), (i, x), \cdots, (k, y)\}$$

Set Operations Notes $(\cup, \cap, -)$

- Operands must be type compatible

  - A and B must have the same number of attributes and the domains of corresponding attributes must be compatible (i.e., dom(A$_i$) = dom(B$_i$))

  - Resulting relation has the same attribute names as A

- Union and intersection are commutative and associative operations $A \cup B = B \cup A, A \cup (B \cup C) = (A \cup B) \cup C$

- Minus is not commutative $A - B \neq B - A$

Example: Union

STUDENT

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FN	LN
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah
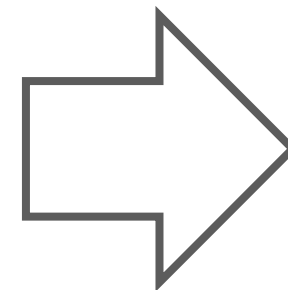
STUDENT ∪ INSTRUCTOR

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert
John	Smith
Ricardo	Browne
Francis	Johnson

Example: Intersection

STUDENT

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FN	LN
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah
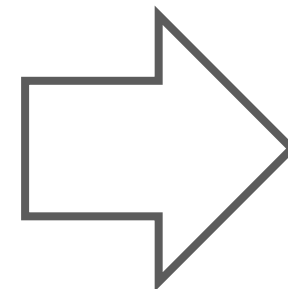
STUDENT ∩ INSTRUCTOR

FN	LN
Susan	Yao
Ramesh	Shah

Example: Difference

STUDENT

FN	LN
Susan	Yao
Ramesh	Shah
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

INSTRUCTOR

FN	LN
John	Smith
Ricardo	Browne
Susan	Yao
Francis	Johnson
Ramesh	Shah
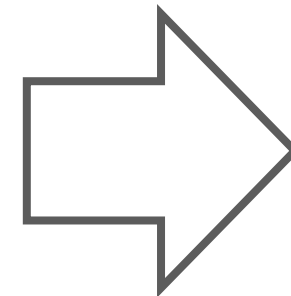
STUDENT − INSTRUCTOR

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

Example: Cartesian Product

PERSON

SSN	Name
932-45-1789	Yao
123-12-3645	Ramesh

DEPENDENT

depSSN	depName
934-55-1234	Helen
936-58-1578	John
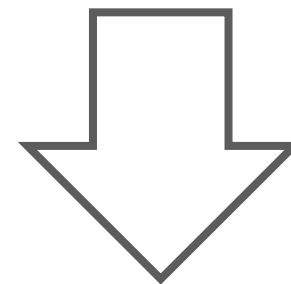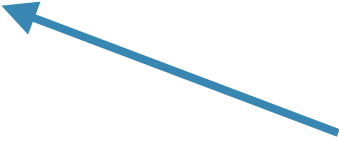
PERSON × DEPENDENT

SSN	Name	depSSN	depName
932-45-1789	Yao	934-55-1234	Helen
123-12-3645	Ramesh	936-58-1578	John
932-45-1789	Yao	936-58-1578	John
123-12-3645	Ramesh	934-55-1234	Helen

Relational Algebra Language
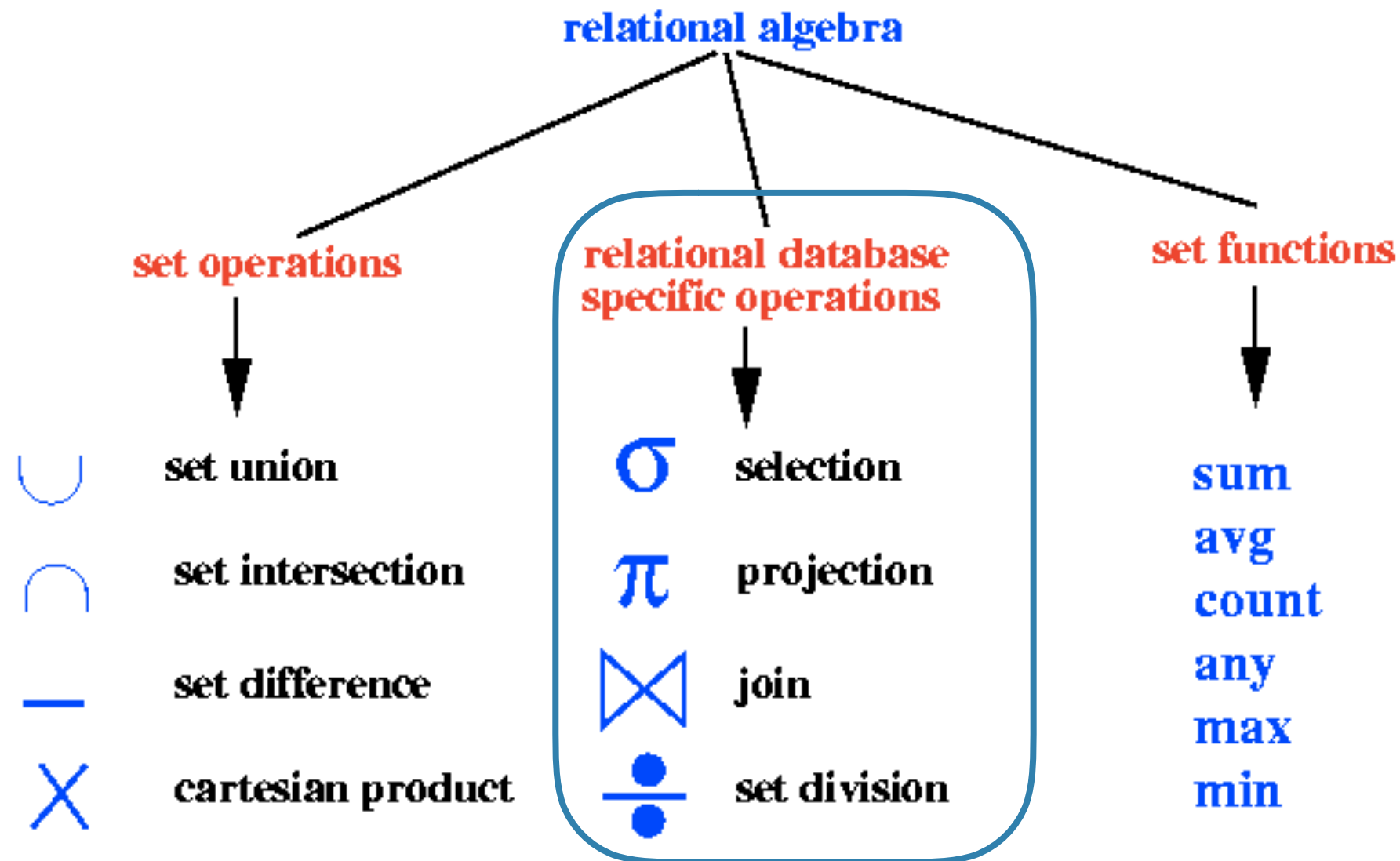
Selection Operation

- Select a subset of tuples from a relation that satisfy a selection condition

- Notation: $\sigma_C(R)$

  - C is condition that refers to attributes of R in the form:
  <attr1> <comparison op> <constant value / attr2>

    boolean expression

  - Outputs rows of R that satisfy C (same schema)

Example: Selection

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

Select employees from department number 4
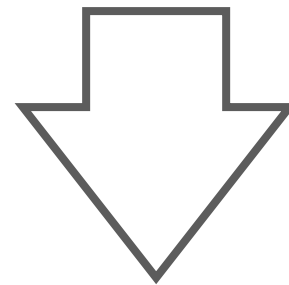
$\sigma_{(\mathrm{dno}=4)}(\mathrm{employee})$

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000

Selection Operation Properties

- Commutative

$$\sigma_{<C1>}(\sigma_{<C2>}(R)) = \sigma_{<C2>}(\sigma_{<C1>}(R))$$

- Order of cascade doesn't matter

$$\sigma_{<C1>}(\sigma_{<C2>}(\sigma_{<C3>}(R))) = \sigma_{<C2>}(\sigma_{<C3>}(\sigma_{<C1>}(R)))$$

- Cascade equivalent to conjunction of all conditions

$$\sigma_{<C1>}(\sigma_{<C2>}(\sigma_{<C3>}(R))) = \sigma_{<C1>AND<C2>AND<C3>}(R)$$

Exercise: Employee Relation Selection

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

1. Retrieve all employees earning more than 50,000

2. Retrieve all female employees

3. Retrieve all employees working for department number 4 and earning more than 50,000

Projection Operation

- Select certain columns from the tables and discards the other columns

- Notation: $\pi_{<\text{attribute list}>}(R)$

  - Outputs only the columns listed in the attribute list

  - Removes duplicate tuples

    Output is a set => no duplicate elements

Projection Properties

- Number of tuples after projection operator is always less or equal to the number of tuples in R

- If the list of attributes includes a key of R, then the number of tuples is equal to the number of tuples in R

- Cascade of two projection operators if the second attribute list contains the attributes in the first list:

$$\pi_{<\text{list}1>}(\pi_{<\text{list}2>}(R)) = \pi_{<\text{list}1>}(R)$$

Example: Projection

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

Return all the social security number of employees
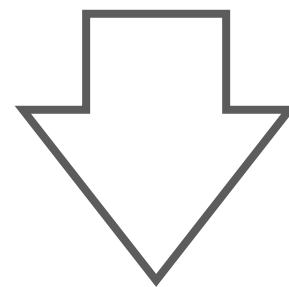
$\pi_{\text{SSN}}(\text{employee})$

SSN
111-11-1111
222-22-2222
333-33-3333

Example: Projection (2)

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

Return all possible genders of employees
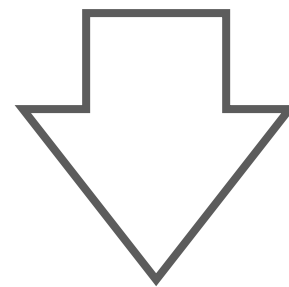
$\pi_{\mathrm{sex}}(\mathrm{employee})$

Sex
M
~~M~~
F

Example: Projection (3)

SSN	FName	LName	Age	Sex	Address	DNo	Salary
111-11-1111	John	Smith	24	M	12 West Lane	4	50,000
222-22-2222	James	Bond	50	M	1 East St	4	80,000
333-33-3333	Jane	Brown	30	F	24 South Lane	3	60,000

Return the last name, first name, and salary of employees
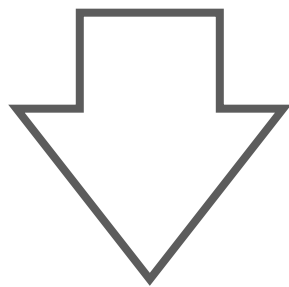
$\pi_{\text{LName, FName, Salary}}(\text{employee})$

LName	FName	Salary
Smith	John	50,000
Bond	James	80,000
Brown	Jane	60,000

Renaming Operation

- Name conflicts can arise in some situations

- Notation: $\rho_{A_1, A_2, \cdots, A_n}(R)$

  - Rename the columns of R to the fields provided

  - Does not change the instance, only the schema!

Example: Rename

- Query: Get the SSN of employees from department 5 and rename the column to Dept5

- Steps:

  - Selection of the employees with department = 5

  - Project only the SSN column

  - Rename

- Answer: $\rho_{\mathrm{DEPT5}}\left(\pi_{\mathrm{SSN}}\left(\sigma_{(\mathrm{DNO}=5)}\left(\mathrm{employee}\right)\right)\right)$

Theta Join Operation

- Composition of relations

- Equivalent to a Cartesian product followed by a selection

- Notation: $R \bowtie_C S = \sigma_C(R \times S)$

==, !=, <, <=, >=, >

Boolean-valued condition of the form $x \; \theta \; y$

x, y are attributes or constants

Example: Join

EMPLOYEE

FName	LName	DNo	Salary
John	Smith	5	30000
Frank	Wong	5	40000
Alicia	Zelaya	4	25000
Jennif	Wallace	4	43000
James	Borg	1	55000

DEPARTMENT

DName	DNumber
Research	5
Administration	4
Headquarters	1

Return employees' info and their department names
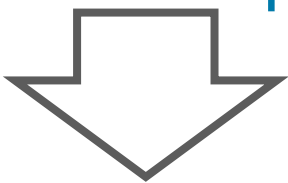
EMPLOYEE $\bowtie_{DNo=DNumber}$ DEPARTMENT

FName	LName	DNo	Salary	DName	DNumber
John	Smith	5	30000	Research	5
Frank	Wong	5	40000	Research	5
Alicia	Zelaya	4	25000	Administration	4
Jennif	Wallace	4	43000	Administration	4
James	Borg	1	55000	Headquarters	1

Join Variation: Equijoin

- Join conditions with equality comparisons only (x ==y)

- One or more pairs of attributes have identical values in every tuple

Example: Equijoin

EMPLOYEE

| Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19
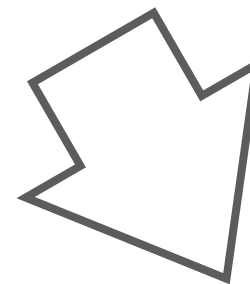
Return the manager's information for all the departments

$$\text{DEPARTMENT} \bowtie_{\text{Mgr\_ssn}=\text{Ssn}} (\text{EMPLOYEE})$$

DEPT_MGR

Dname	Dnumber	Mgr_ssn	$\cdots$	Fname	Minit	Lname	Ssn	$\cdots$
Research	5	333445555	$\cdots$	Franklin	T	Wong	333445555	$\cdots$
Administration	4	987654321	$\cdots$	Jennifer	S	Wallace	987654321	$\cdots$
Headquarters	1	888665555	$\cdots$	James	E	Borg	888665555	$\cdots$

Join Variation: Natural Join

- Denoted using *

- EQUIJOIN operation followed by a projection operation that removes one of the duplicate attributes

- Requires the two join attributes to have the same name in both relations, otherwise renaming operation is necessary first
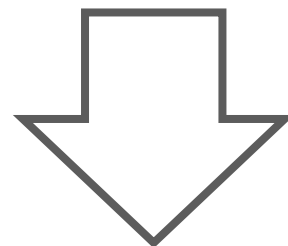
Example: Natural Join

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

$$DEPARTMENT * DEPT\_LOCATIONS$$

DEPT_LOCS

Dname	Dnumber	Mgr_ssn	Mgr_start_date	Location
Headquarters	1	888665555	1981-06-19	Houston
Administration	4	987654321	1995-01-01	Stafford
Research	5	333445555	1988-05-22	Bellaire
Research	5	333445555	1988-05-22	Sugarland
Research	5	333445555	1988-05-22	Houston

Outer Join Operation

- Does not require each record in the two joined tables to have matching values

- Retains records from one or both tables even if no matches are found

- 3 flavors of outer joins

  - Left outer-join

  - Right outer-join

  - Full outer-join

Left Outer-Join: $A ⟕ B$

- Keeps every tuple in the first or left relation A

- If no matching tuple is found in the second relation B, then it is combined with a row of NULL values (filled or padded with NULL values)

A

X	Y
1	7
2	5
3	4

B

U	V
1	8
2	3
4	9

$A ⟕_{A.X=B.U} B$

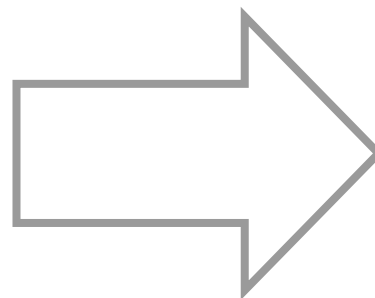| X	Y	U	V
1	7	1	8
2	5	2	3
3	4	NULL	NULL

Right Outer-Join: $A \bowtie B$

- Keeps every tuple in the second or right relation B
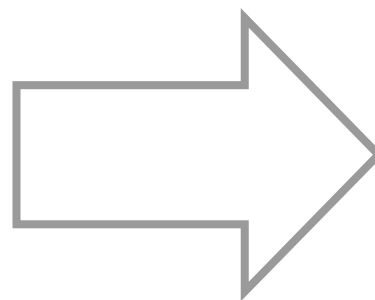
- If no matching tuple is found in the left relation A, then it is combined with a row of NULL values (filled or padded with NULL values)

A

X	Y
1	7
2	5
3	4

B

U	V
1	8
2	3
4	9

$A \bowtie_{A.X=B.U} B$

X	Y	U	V
1	7	1	8
2	5	2	3
NULL	NULL	4	9

Full Outer-Join $A \bowtie B$

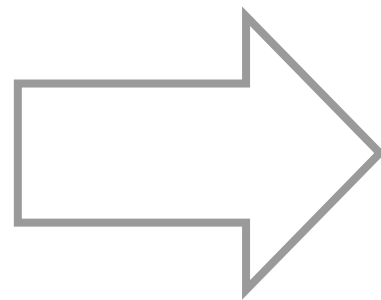- Keeps all tuples from both the left and right relations

- Unmatched tuple in either table is combined with a row of NULL values

A

X	Y
1	7
2	5
3	4

B

U	V
1	8
2	3
4	9

$A \bowtie_{A.X=B.U} B$

X	Y	U	V
1	7	1	8
2	5	2	3
3	4	NULL	NULL
NULL	NULL	4	9

Division Operation

- Suppose $R_1(A,B)$ and $R_2(B)$, the output contains all A-tuples such that for every B-tuple in R_2, there exists an (A,B) tuple in R1

- Notation: $R_1 \div R_2$

- Example: Find first name and last name of all employees who work on all projects that "John Smith" works on

Example: Division

A

S	P
s1	p1
s1	p2
s1	p3
s1	p4
s2	p1
s2	p2
s3	p2
s4	p2
s4	p4

B1

P
p2

B2

P
p2
p4

B3

P
p1
p2
p4

$A \div B1$

S
s1
s2
s3
s4

$A \div B2$

S
s1
s4

$A \div B3$

S
s1

Division as a Sequence of Other Operations

- Construct a "minimum qualifying" relation which contains the tuples that is necessary to be selected by the query. Suppose $R_1(A, B)$ and $R_2(B)$, then "qualify" relation is constructed as: $\text{qualify} = \pi_A(R_1) \times R_2$

- Compute the difference between "qualify" and R_1. This gets all the disqualified tuples: $\text{unqualify} = \text{qualify} - R_1$

- Last step is to obtain the qualified tuples using another set difference: $\pi_A(R_1) - \pi_A(\text{unqualify})$

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

x3
x1
x2
x4

set of values (tuples)

aggregate function

x

one single value
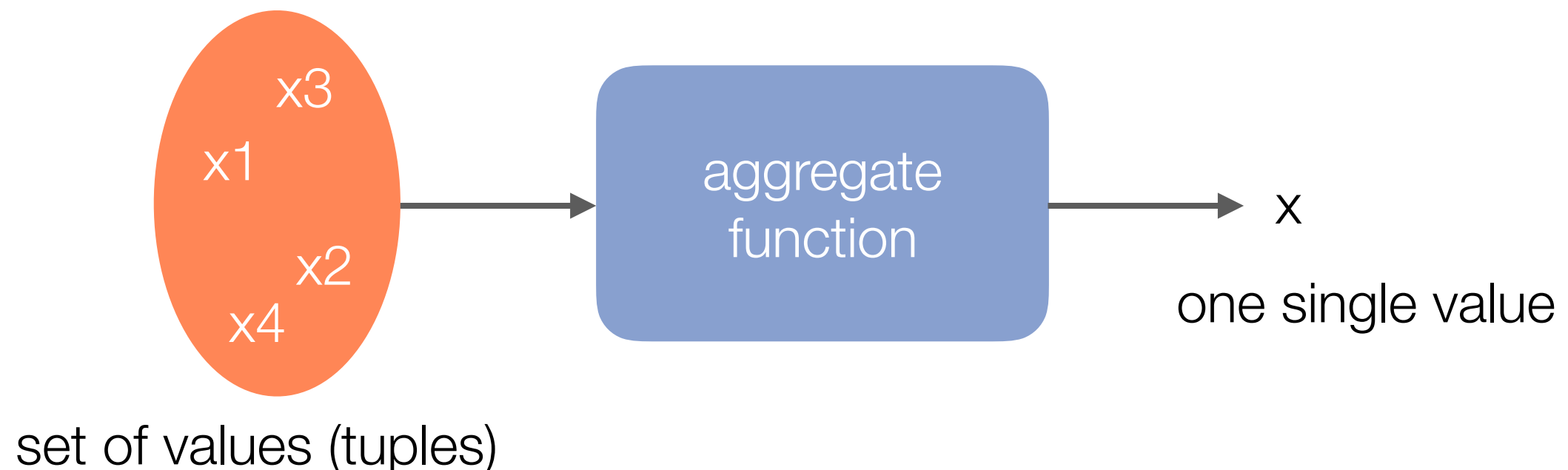
Example: Set Functions

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:

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

$$\pi_{\text{SSN,FNAME,AGE}/2+7,\text{SEX}}(\text{EMPLOYEE})$$
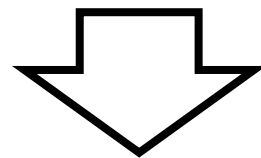
Additional Operations: Group By Aggregate

- 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

avg(salary) = 55,0000

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

avg(salary) = 70,0000

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

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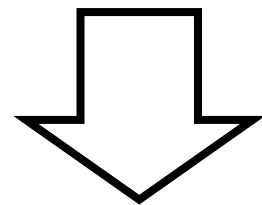
$$\text{DNO,Sex}\mathcal{F}_{\text{avg(Salary),count(SSN)}}(\text{EMPLOYEE})$$

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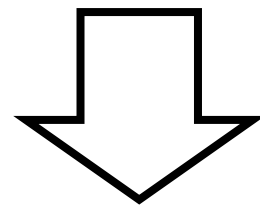
$$\mathcal{F}_{\text{avg(Salary),count(SSN)}}(\text{EMPLOYEE})$$

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
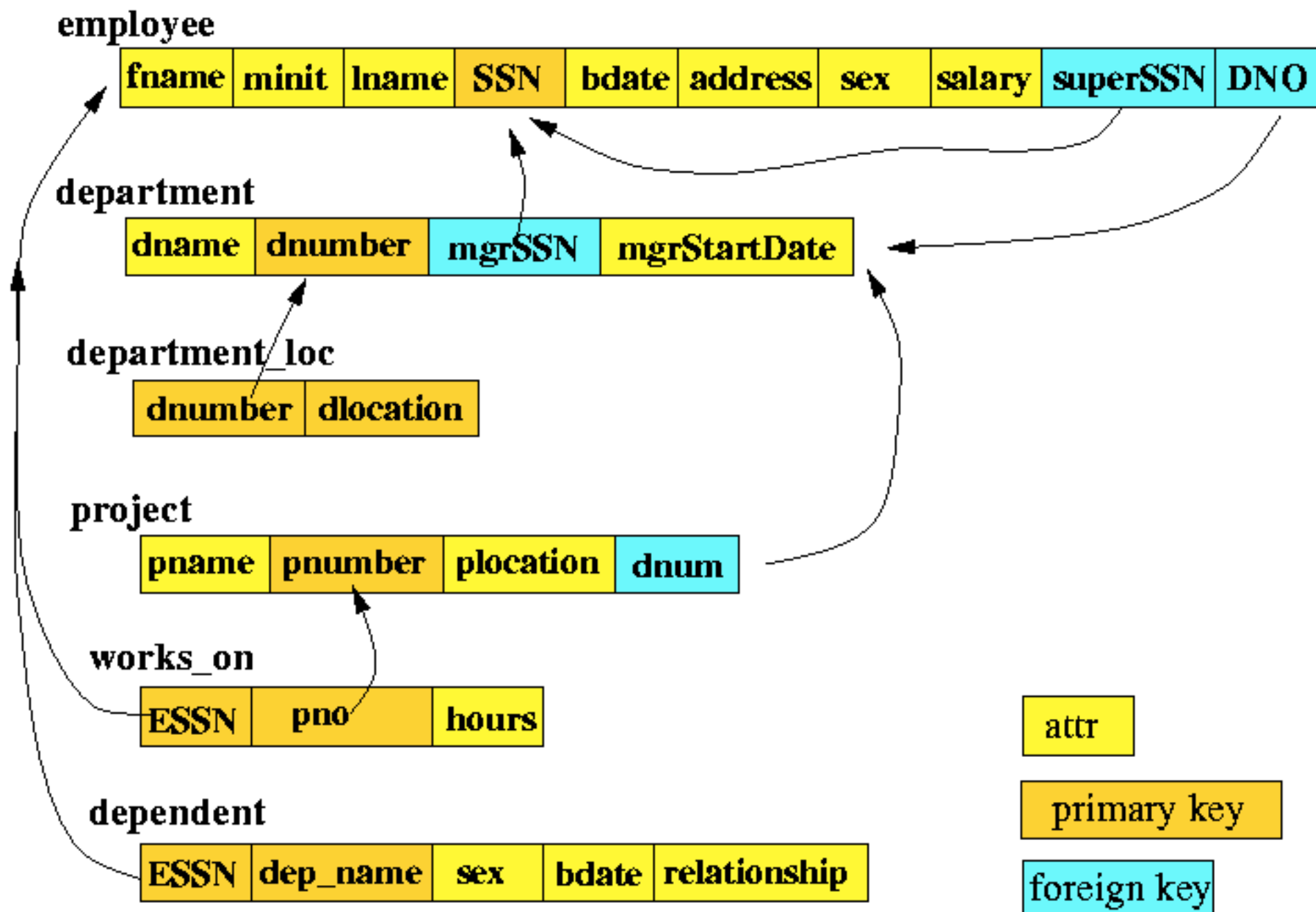
Exercises: RA Queries

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

- Find fname and lname of employees who earn more than 'John Smith'

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

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

Exercises: RA Queries (2)

- Retrieve the names of employees who have no dependents

- List the names of managers who have at least one dependent

- Find fname and lname of employees who work on more projects than 'John Smith'

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

Exercises: RA Queries (3)

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

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

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

- 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 or get bound to tuples (similar to SQL)

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

- Expression in calculus are called formulas

Tuple Relational Calculus

- 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) }

  tuple   formula

- Returns all tuples for which the 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

  - R.a *op* constant

$$\text{op} \in \{>, <, =, \leq, \geq, \neq\}$$

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

- Existential quantifier $(\exists t)\ (\mathrm{Condition}(t))$
  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
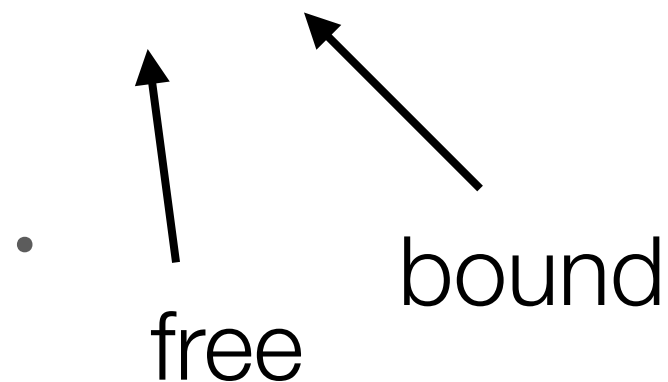
TRC Example: Convention

SAILORS (<u>sid</u>, sname, rating, age)
RESERVES (<u>sid</u>, <u>bid</u>, <u>day</u>)
BOATS (bid, bname, color)

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

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

free

bound

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

TRC Example: Exists

Find the department number of the Research department

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

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

   AND (

        $(\exists \ t)$

        $(\text{Department}(t)$

-       AND $t.\text{dname} = \text{'Research'}$

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

        $)\}$

TRC Example: Complex Exists

- 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

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

TRC Example: Decoding Exists

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: For All

- 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: Not Exists

- 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: Recap

- Relational Algebra

  - Query language for relations

- Basic Operations

  - Set operations & functions

  - Database-specific operations (selection, projection, join, division)

  - Group by aggregate

Relational Calculus: Recap

- Relational Calculus

  - Same expressive power as RA

  - Expression & formulas

  - Special quantifiers