Relational Algebra

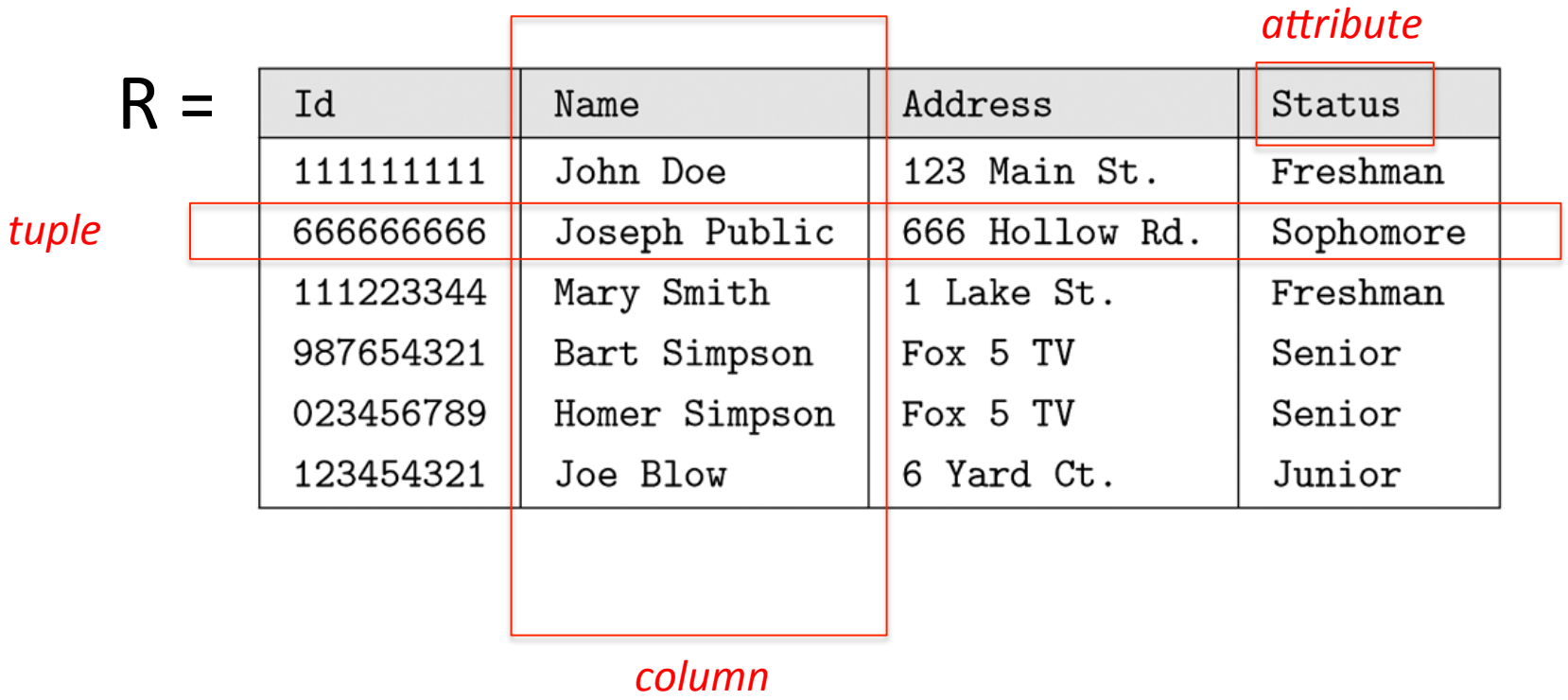
Carsten Schürmann

Formal query languages

- How do we collect information?
- Example: Average age of students in this class?
- Relational algebra [Codd 1970]
  - Set theory
  - Operators
  - What is a minimal set of operators?

Relational Algebra

Domain: Relations



Definitions

Let $R(A_1 \ldots A_n)$ a relational schema

Instance of the schema

A1	…	An
F_{11}	…	F_{1n}
…	…	…
F_{m1}	…	F_{mn}

Attributes $A(R) = A_1 \ldots A_n$

Tuples $T(R) = \{(F_{11}, \ldots, F_{1n}), \ldots, (F_{m1}, \ldots, F_{mn})\}$

RelAlg.js Introduction

Webaddress: http://fangel.github.io/RelAlg.js/

Author: Morten Fangel, former ITU student

Declaration of Relations:

[['attribute_a', 'attribute_b'] -> [1,2], [2,3], [3,4]]

Union

Query: Who is enrolled at ITU or DIKU?

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

R_{DIKU}

cpr	name	address
120492-1234	Claudia	Odense
010299-2345	Peter	Copenhagen
151987-3456	Merete	Odense
250899-4567	Paul	Copenhagen

$R = R_{ITU} \cup R_{DIKU}$

R := R_{ITU} Union R_{DIKU}

[Mathematics]

[RelAlg.js]

Intersection

Query: Who is enrolled at ITU and SDU?

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

R_{SDU}

cpr	name	address
120492-1234	Claudia	Odense
010299-2345	Peter	Copenhagen
041297-5367	Nikoline	Aarhus
250899-4567	Paul	Copenhagen

$R = R_{ITU} \cap R_{SDU}$ [Mathematics]

$R := R_{ITU}$ InterSection R_{SDU} [RelAlg.js]

Difference

Query: Who is enrolled at ITU and not SDU?

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

R_{SDU}

cpr	name	address
120492-1234	Claudia	Odense
010299-2345	Peter	Copenhagen
041297-5367	Nikoline	Aarhus
250899-4567	Paul	Copenhagen

$R = R_{ITU} - R_{SDU}$

$R := R_{ITU} - R_{SDU}$

[Mathematics]

[RelAlg.js]

Minmality

- Do we need all three, or can we get away with fewer connectives?

- Intersection is redundant

$$R \cap S = R - (R - S) = S - (S - R)$$

Selection

Query: Find all ITU students living in Copenhagen

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

$R = \sigma_{address=Copenhagen}(R_{ITU})$ [Mathematics]

R := Select [address == Copenhagen] (R_{ITU}) [RelAlg.js]

Projection

Query: Find all names of ITU students

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

$R = \pi_{name}(R_{ITU})$ [Mathematics]

R := Project[name] (R_{ITU}) [RelAlg.js]

Cartesian Product

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus

R_{grades}

cpr	course	grade
140298-1234	SIDD	10
041297-5367	SIDD	12

R

R1.cpr	name	address	R2.cpr	course	grade
140298-1234	Jesper	Copenhagen	140298-1234	SIDD	10
140298-1234	Jesper	Copenhagen	041297-5367	SIDD	12
041297-5367	Nikoline	Aarhus	140298-1234	SIDD	10
041297-5367	Nikoline	Aarhus	041297-5367	SIDD	12

$R = R_{ITU} \times R_{grades}$ [Mathematics]

$R := R_{ITU} \; X \; R_{grades}$ [RelAlg.js]

Renaming

Query: Replace attribute name by first name

R_{ITU}

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

$R = \varrho_{name/firstname}(R_{ITU})$ [Mathematics]

R := Rename [name/firstname](R_{ITU}) [RelAlg.js]

Fundamental Relational Operators

- Selection $\sigma_{P(A1...An)}(R)$
- Projection $\pi_{A1...An}(R)$
- Cartesian Product $R \times S$
- Set Union $R \cup S$
- Set Difference $R \cap S$
- Renaming $\varrho_{A/B}(R)$

Where P is a logical formula with conjunctions, relations etc.

Derived Operators

Join

Equi-join: $R \bowtie_{A=B} S = \sigma_{A=B}(R \times S)$

Theta-join: $R \bowtie_{\theta} S = \sigma_{\theta}(R \times S)$

Natural join:

Let A be an attribute common to R and S

$R \bowtie S = \pi_{(R \cup S)}(\varrho_{A}(\varrho_{A/R.A}(R) \bowtie_{R.A=S.A} \varrho_{A/S.A}(S)))$

Relation Renaming

R

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

$\varrho_S(R)$ =

S.cpr	S.name	S.address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

Division

Find the suspicious suppliers who supply all parts necessary to build a bomb!

Shipment

supplier	part
s1	p1
s2	p1
s1	p2
s3	p1
s5	p1
s2	p3

÷

Bomb

part
p1
p2

=

Bad

supplier
s1

Division

Idea: Compute the bad (suspicious suppliers) from all by removing the good suppliers

Shipment

supplier	part
s1	p1
s2	p1
s1	p2
s3	p1
s5	p1
s2	p3

All = π_{supplier}(Shipment)

supplier
s1
s2
s3
s5

Division

Idea: Good Suppliers are those who do not deliver one bomb part.

Shipment

supplier	part
s1	p1
s2	p1
s1	p2
s3	p1
s5	p1
s2	p3

All

supplier
s1
s2
s3
s5

Bomb

part
p1
p2

Suspicious = All x Bomb

supplier	part
s1	p1
s1	p2
s2	p1
s2	p2
s3	p1
s3	p2
S5	p1
S5	p2

Division

Idea: Compare shipments and suspicious shipments (didn't happen)

Shipment Suspicious NotHappen

= Suspicious - Shipment

supplier	part
s1	p1
s2	p1
s1	p2
s3	p1
s5	p1
s2	p3

supplier	part
s1	p1
s1	p2
s2	p1
s2	p2
s3	p1
s3	p2
S5	p1
S5	p2

supplier	part
s2	p2
s3	p2
S5	p2

Division

Idea: Good suppliers are those with a bad shipment not happening

All

supplier
s1
s2
s3
s5

NotHappen

supplier	part
s2	p2
s3	p2
S5	p2

Good = π_{supplier}(NotHappen)

supplier
s2
s3
S5

Bad = All - Good

supplier
s1

Putting the Pieces Together

Shipment ÷ Bomb

= Bad

= All – Good

= π_{supplier}(Shipment) - π_{supplier}(NotHappen)

= π_{supplier}(Shipment) - π_{supplier}(Suspicious – Shipment)

= π_{supplier}(Shipment) - π_{supplier}(All x Bomb - Shipment)

= π_{supplier}(Shipment)

- π_{supplier}(π_{supplier}(Shipment) x Bomb - Shipment)

Abstracting from the Example

Step 1:

$R \div S = \pi_{supplier}(R) - \pi_{supplier}(\pi_{supplier}(R) \times S - R)$

But how shall we abstract the supplier?

supplier = A(shipment) – A(bomb)

Step 2:

$R \div S = \pi_{R\text{-}S}(R) - \pi_{R\text{-}S}(\pi_{R\text{-}S}(R) \times S - R)$

Relational Tuple Calculus

Introduction

Query: Give me the set of tuples t that satisfy predicate P

Notation: $\{t \mid P(t)\}$

Example:

$\{t \mid t \in \text{Student}\}$

Student

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus
151197-2352	Claus	Dragør
050596-1142	Martin	Copenhagen

First-Order Logic

Atomic Propositions	(t, s tuples, R relation, a,b attributes)
Tuple equality	t = s
Set membership	t ∈ R
Domain equality	t.a = s.b
Domain inequality	t.a < s.b
	t.a <= s.b
	t.a <> s.b

First-Order Logic

Formulas (Assuming P, Q are propositions)

Conjunction	$P \wedge Q$
Disjunction	$P \vee Q$
Negation	$\neg P$
Implication	$P \supset Q$
Universal Quantification	$\forall t \in R.\ P$
Existental Quantification	$\exists t \in R.\ P$

Examples

Find a student with cpr number 140298-1234

$$\{t \mid t \in \text{Student} \ \wedge \ t.\text{cpr} = 140298\text{-}1234\}$$

Find the name of a student with cpr number 140298-1234

$$\{t \mid \exists s \in \text{Student}.\ t.\text{cpr} = 140298\text{-}1234 \ \wedge \ t.\text{name} = s.\text{name}\}$$

Who is enrolled at ITU or DIKU?

$$\{t \mid t \in \text{ITU} \ \vee \ t \in \text{DIKU}\}$$

Examples

Compute Cartesian product

ITU

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus

Grades

cpr	course	grade
140298-1234	SIDD	10
041297-5367	SIDD	12

$$\{t \mid \exists s \in ITU.\ \exists r \in Grades.\ t.cpr1 = s.cpr \wedge t.name = s.name \wedge t.address = s.address \wedge t.cpr2 = r.cpr \wedge t.course = r.course \wedge t.grade = s.grade\}$$

Fun Facts

Relational Algebra

and

Relational Tuple Calculus are equivalent!

Example

Query: Who received what grade?

Student

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus

Grades

cpr	course	grade
140298-1234	SIDD	10
041297-5367	SIDD	12

Example

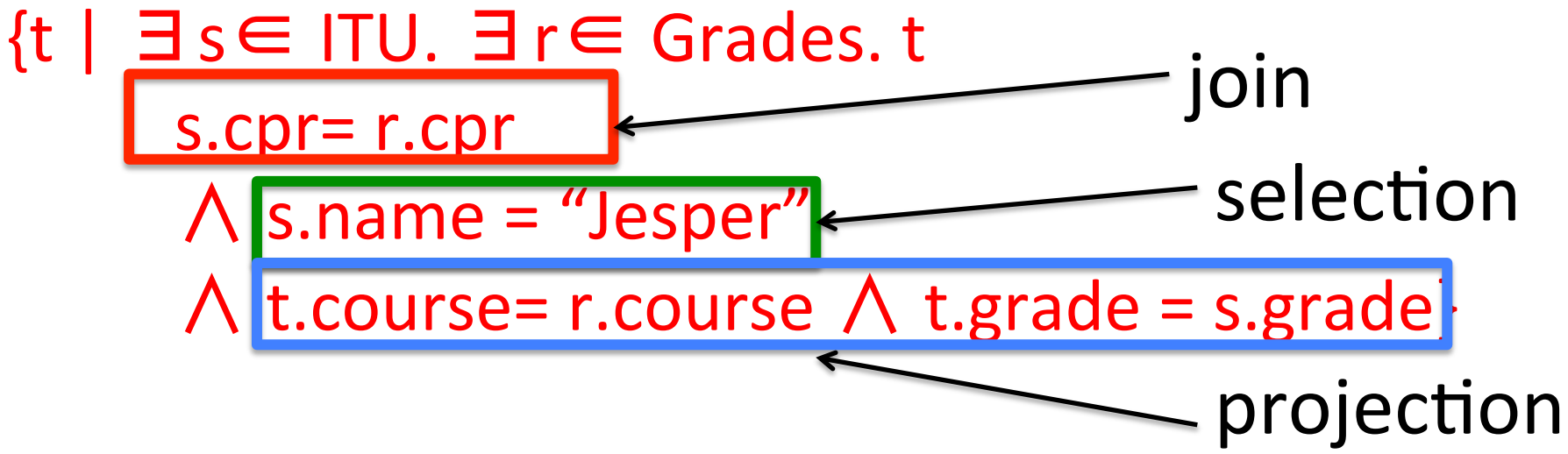
What grade did Jesper receive in which course?

ITU

cpr	name	address
140298-1234	Jesper	Copenhagen
041297-5367	Nikoline	Aarhus

Grades

cpr	course	grade
140298-1234	SIDD	10
041297-5367	SIDD	12



Conclusion

- Relational Algebra
- RelAlg.js tool
- Relational Tuple Calculus

They are all equivalent

Next time: SQL