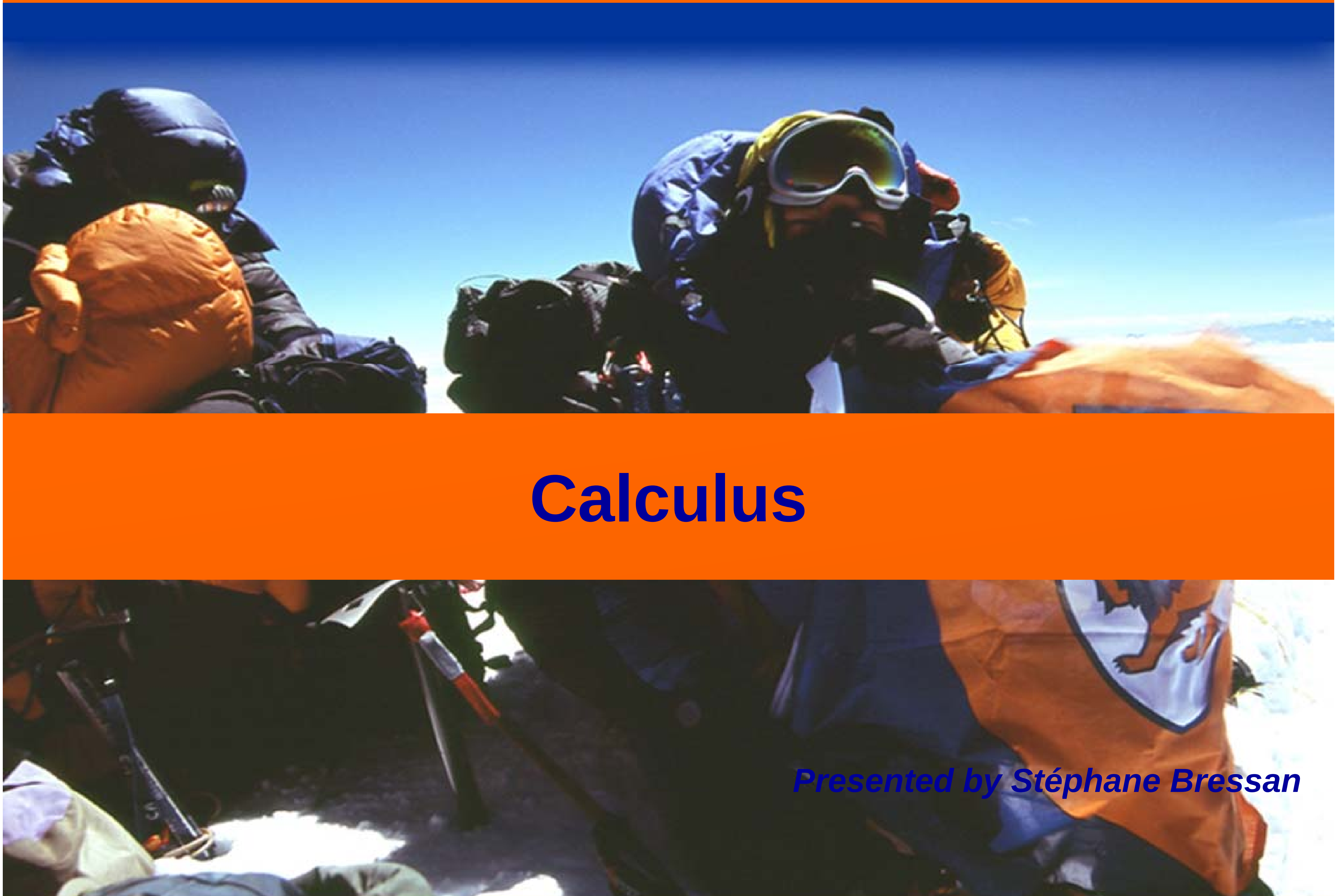


Calculus

Presented by Stéphane Bressan

Relational Calculi

There are two calculi:

- Domain relational calculus (DRC);
- T-uple relational calculus (TRC).

- DRC and TRC are query languages
- They are both based on logic

Learning Objectives

- Write and understand queries in Domain Relational Calculus
- Write and understand queries in T-uple Relational Calculus

Propositional Logic

"Aristotle is Greek"

"Aristotle is Greek and Alexander is Persian"

"Aristotle is not Persian"

"Alexander is Macedonian or Persian"

Propositional Logic

“Roxane is Bactrian or not Bactrian”

“Olympias is Greek and is not Greek”

“Olympias is Greek implies Alexander is Greek”

“Roxane is Bactrian implies Roxane is Bactrian”

Propositional Logic

"Olympias is Greek implies Alexander is Greek"

"Alexander is not Greek implies Olympias is not Greek"

Semantics of Propositional Logic

The semantic of propositional logic is defined by truth tables

A	B	$(A \vee B)$	$(A \wedge B)$	$(A \Rightarrow B)$	$\neg (A)$
T	T	T	T	T	F
F	T	T	F	T	T
T	F	T	F	F	F
F	F	F	F	T	T

First Order Logic: Predicates

greek(aristotle)

greek(X)

mother(olympias, alexander)

mother(X, Y)

First Order Logic

$\exists$ X greek(X)

$\exists$ X mother(olympias, X)

$\exists$ X $\exists$ Y mother(Y, X)

$\exists$ Y $\exists$ X mother(Y, X)

First Order Logic

$\forall X \; greek(X)$

$\forall Y \; \exists X \; mother(X, Y)$

$\exists X \; \forall Y \; mother(X, Y)$

First Order Logic

$$\forall X \, \forall Y \, ((mother(X, Y) \wedge greek(X)) \Rightarrow greek(Y))$$

Syntax of First Order Logic

First order logic consists of formulae built from predicates, constants *(lower case)* and variables *(upper case)*, and connectives:*

- $(F \wedge G)$
- $(F \vee G)$
- $\neg (F)$
- $(F \Rightarrow G)$

And **quantifiers:** $\forall$ and $\exists$

Variables can be quantified (bound) or free

Semantics of Predicate Logic

To avoid confusion we agree that:

- A variable is quantified once at most.
- If a variable is quantified in a formula, it cannot appear outside of the scope of its quantifier.

$$\forall X \, (\underbrace{\exists X \, \overbrace{(\text{mother}(X, Y))}})$$

Different scopes

$$\exists Y \, \overbrace{(\text{mother}(X, Y))} \wedge \text{father}\,(Z, Y)$$

Semantics of Predicate Logic

$\neg \forall X F$

is equivalent to

$\exists X \neg F$

$\neg \exists X F$

is equivalent to

$\forall X \neg F$

*(*Here F represents a formula)*

Calculus

- A Calculus defines formulae and their meaning
- T-uple Relational Calculus: variables range over t-uples (TRC)
- Domain Relational Calculus: variables range over values (DRC)

Calculus

How to represent the set of integers 2, 3, and 4?

In extension:

$$\{2, 3, 4\}$$

In Intention (set-builder notation, comprehension, abstraction):

$$\{ X \mid X \in N \wedge 1 < X \wedge X < 5\}$$

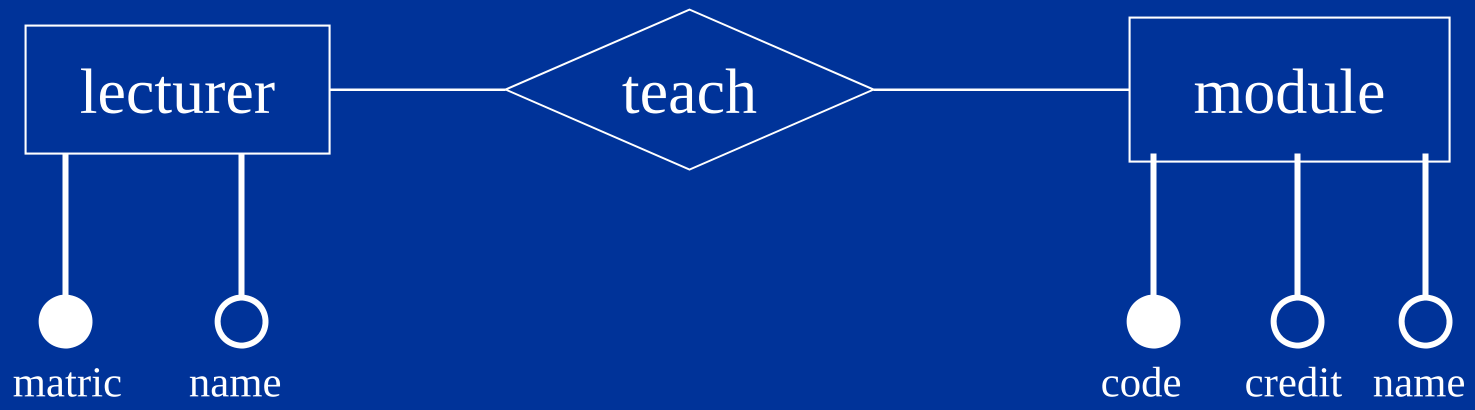
Calculus: Where is the Truth?

- The truth is in the database
- If a relation Mother in the database has a t-uple mother(olympias, alexander) then Olympias is the mother of Alexander
- Otherwise it is not *(closed world assumption)*

T-uple relational Calculus

Example

- lecturer(<u>matric</u>, name)
- module(<u>code</u>, name, credit),
- teach(<u>matric</u>, <u>code</u>)



Example

- $\{T \mid T \in \text{lecturer}\}$

- $\{T \mid \exists\, T1\ (T1 \in \text{lecturer} \wedge T = T1)\}$

- $\{T \mid \exists\, T1\ (T1 \in \text{Lecturer}$
  $\wedge\ T.matric = T1.matric$
  $\wedge\ T.name = T1.name)\}$

  by CONVENTION!

Syntax of T-uple Relational Calculus

- Parenthesis can be omitted if non ambiguous

Example

- $\{T \mid \exists\ T1\ T1 \in \text{lecturer}$
  $\wedge\ T.name = T1.name\}$

Example

- $\{T \mid \exists\ T1\ T1 \in \text{lecturer}$
  $\wedge$ T1.name = “Smith”
  $\wedge$ T.matric = T1.matric$\}$

Example

- $\{T \mid \exists\, T1\ \exists\, T2\ \exists\, T3$

  $T1 \in lecturer$

  $\wedge\ T2 \in module$

  $\wedge\ T3 \in teach$

  $\wedge\ T1.matric = T3.matric$

  $\wedge\ T2.code = T3.code$

  $\wedge\ T2.credit < 3$

  $\wedge\ T.lec\_name = T1.name$

  $\wedge\ T.mod\_name = T2.name\}$

Example

```
SELECT
   T1.name as lec_name,
   T2.name as mod_name
FROM  lecturer T1, module T2,teach t3
WHERE T1.matric = T3.matric
        AND T2.code = T3.code
        AND T2.credit < 3
```

Example (incorrect)

- $\{T \mid \exists\, T1$

  $T1 \in lecturer$

  $\wedge\ ((\exists\, T2\ T2 \in module) \Rightarrow (\exists\, T3$

  $T3 \in teach$

  $\wedge\ T1.matric = T3.matric$

  $\wedge\ T2.code = T3.code))$

  $\wedge\ T.name = T1.name\}$

Example (correct)

- $\{T \mid \exists\, T1\ \forall\, T2\ \exists\, T3$

  $T1 \in lecturer$

  $\wedge\ (T2 \in module \Rightarrow ($

  $T3 \in teach$

  $\wedge\ T1.matric = T3.matric$

  $\wedge\ T2.code = T3.code))$

  $\wedge\ T.name = T1.name\}$

Attention!

- $\exists\, T \in r\ (F)$

  means $\exists\, T\ (T \in r \land F)$

- $\exists\, T1 \in r\ \exists\, T2 \in s\ (F)$

  means $\exists T1\ \exists\, T2\ (T1 \in r \land T2 \in s \land F)$

*(*Here F represents a formula)*

Attention!

- $\forall\ T \in r\ (F)$

  means $\forall\ T\ (T \in r \Rightarrow F)$

- $\forall\ T1 \in r\ \forall\ T2 \in s\ \ F$

  means

  $$\forall\ T1\ \forall\ T2\ ((T1 \in r \wedge T2 \in s) \Rightarrow F)$$

*(*Here F represents a formula)*

Semantics of T-uple Relational Calculus

- Example
  - $\forall T \in module\ (T.credit > 1)$

  means
  - $\forall T\ (T \in module \Rightarrow T.credit > 1)$

  and does not mean
  - $\forall T\ (T \in module \wedge T.credit > 1)$

Semantics of T-uple Relational Calculus

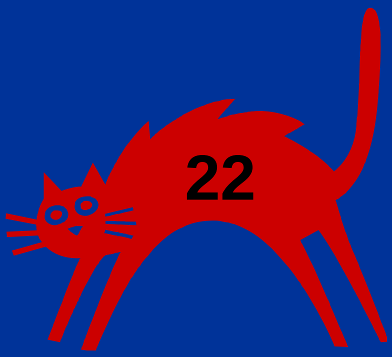
{T | F(T)}

- An **interpretation** I is a mapping of a formula to {true, false}
- An interpretation is defined by a mapping I of the free variable (T) of a formula (F(T)) to a t-uple t of constants
- t $\in$ R is true if and only if the t_uple t is in the instance of R in the database
- A **model** is an interpretation for which the formula is true

Safety of Queries in T-uple Relational Calculus

$\{T \mid T \notin lecturer\}$

("mycat", 22, "red") is not a lecturer, any t-uple in the world maybe an answer if it is not already in the lecturer relation.



Safety of Queries in T-uple Relational Calculus

$\{T \mid \exists\, T1\ T1 \in \text{lecturer}$
$\quad \wedge\ T1.\text{matric} \neq \text{'1234'}$
$\quad \wedge\ T.\text{name} = T1.\text{name}\}$

Safety of Queries in T-uple Relational Calculus

A query is **safe** if the set of t-uples in the answer is a subset of the set of t-uples that can be constructed from the constants explicitly referenced directly (they appear in the query) or indirectly (they appear in a relation mentioned in the query) in the query.

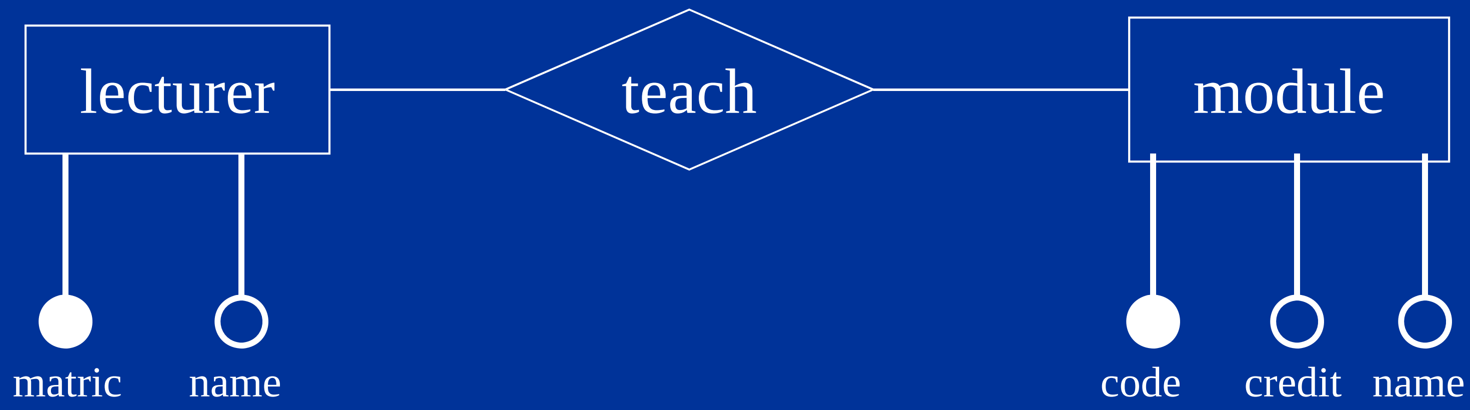
Safety

We consider only safe queries

Domain Relational Calculus

Example

- lecturer(<u>matric</u>, name)
- module(<u>code</u>, name, credit),
- teach(<u>matric</u>, <u>code</u>)



Example

- $\{<X> \mid \exists Y \ \text{lecturer}(X, Y)\}$
- $\{<X> \mid \exists Y \ \text{lecturer}(X, Y) \wedge Y = \text{“john”}\}$
- $\{<X> \mid \text{lecturer}(X, \text{“john”})\}$
- How do you express

  SELECT * FROM Lecturer

Example

Find the names of lecturers teaching a module with less than 2 credits. Print the names of the lecturers and the names of the corresponding modules.

Example

$$\{\langle LN, MN\rangle \mid \exists M \; \exists C \; \exists Cr$$
$$\quad Lecturer(M, LN)$$
$$\quad \wedge \; Module(C, MN, Cr)$$
$$\quad \wedge \; Teach(M, C)$$
$$\quad \wedge \; Cr < 2\}$$

Example SQL

- $\{\langle LN, MN \rangle \mid \exists M1\ \exists M2\ \exists C1\ \exists C2\ \exists Cr$

  $Lecturer(M1, LN)$

  $\wedge\ Module(C1, MN, Cr)$

  $\wedge\ Teach(M2, C2)$

  $\wedge\ C1 = C2 \wedge M1 = M2 \wedge Cr < 2\}$

Example SQL

- SELECT

```
SELECT
    Lecturer.lecName,
    Module.moduleName
FROM Lecturer, Module, Teach
WHERE Lecturer.matric=Teach.matric
AND Module.code = Teach.code
AND Module.credit < 2
```

Example

$$\{\langle M1, M2, N\rangle \mid lecturer(M1, N) \wedge lecturer(M2, N) \wedge M1 <> M2\}$$

Example

$$\{\langle M1, M2, N1\rangle \mid \exists N2 \; lecturer(M1, N1) \wedge lecturer(M2, N2) \wedge M1 <> M2 \wedge N1 = N2\}$$

Example

Find the names of the lecturers teaching all modules

$$\{<N> \mid \exists\, M\ \forall\, C\ \forall\, MN\ \forall\, Cr$$
$$(\text{lecturer}(M, N) \wedge$$
$$(\text{Module}(C, MN, Cr) \Rightarrow \text{teach}(M, C)))\ \}$$

Semantics of Domain Relational Calculus

$$\{<X_1, \ldots, X_n> \mid F(X_1, \ldots, X_n)\}$$

- An **<u>interpretation</u>** **I** is a mapping of each formula to **{true, false}**
- An interpretation is defined by a mapping I of the free variables ($X_1, \ldots, X_n$) of a formula ($F(X_1, \ldots, X_n)$) to constants
- $R(c_1, \ldots, c_n)$ is true if and only if the t_uple $<c_1, \ldots, c_n>$ is in the instance of R in the database
- A **<u>model</u>** is an interpretation for which the formula is true

Example

Find the names of the lecturers teaching all modules:

$\{\langle N \rangle \mid \exists\ M\ \forall\ C\ \forall\ MN\ \forall\ Cr$

$(lecturer(M, N) \land$

$(Module(C, MN, Cr) \Rightarrow teach(M, C)))\ \}$

Example

$$\exists\, M\ \forall\, C\ \forall\, MN\ \forall\, Cr$$

We are looking for values of N such that the formula below is true for **SOME** value of M and, for that value of M, for **ALL** values of C, MN and Cr and :

$(\text{lecturer}(M, N) \wedge$

$(\text{Module}(C, N, Cr) \Rightarrow \text{teach}(M, C)))\ \}$

Example

If <M, N> if a lecturer and

<C, N, Cr> is not a module the formula is true!!!!

$$(\text{lecturer}(M, N) \wedge$$
$$(\text{Module}(C, N, Cr) \Rightarrow \text{teach}(M, C))) \}$$

Example

If <M, N> if a lecturer and

<C, N, Cr> is a module, and M teaches C
the formula is true

$(\text{lecturer}(M, N) \wedge$

$(\text{Module}(C, N, Cr) \Rightarrow \text{teach}(M, C)))\ \}$

Example

The formula is false only if

<M,N> is not a lecturer or if

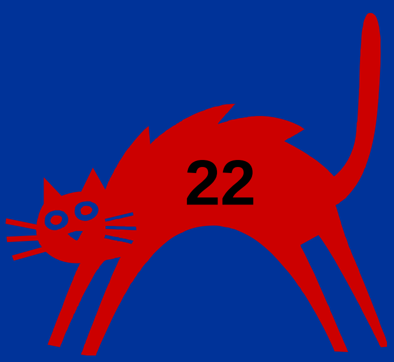
<C, N, Cr> is a module, and M does not teach C

(lecturer(M, N) $\wedge$

(Module(C, N, Cr) $\Rightarrow$ teach(M, C))) }

Safety of Queries in the Domain Calculus

$\{<M,N> \mid \neg \text{ lecturer}(M,N)\}$



Safety of Queries in the Domain Calculus

- A query is **safe** if the set of t-uples in the answer is a subset of the set of t-uples that can be constructed from the constants explicitly referenced directly (they appear in the query) or indirectly (they appear in a relation mentioned in the query) in the query.

Safety

- We consider only safe queries

Credits

The content of this lecture is based on chapter 3 of the book "Introduction to database Systems"
By
S. Bressan and B. Catania,
McGraw Hill publisher

Animated characters are animated using VocaliseTTS under license from Digital Curiosty

Clipart and media are licensed from Microsoft Office Online Clipart and Media

Copyright © 2012 by Stéphane Bressan