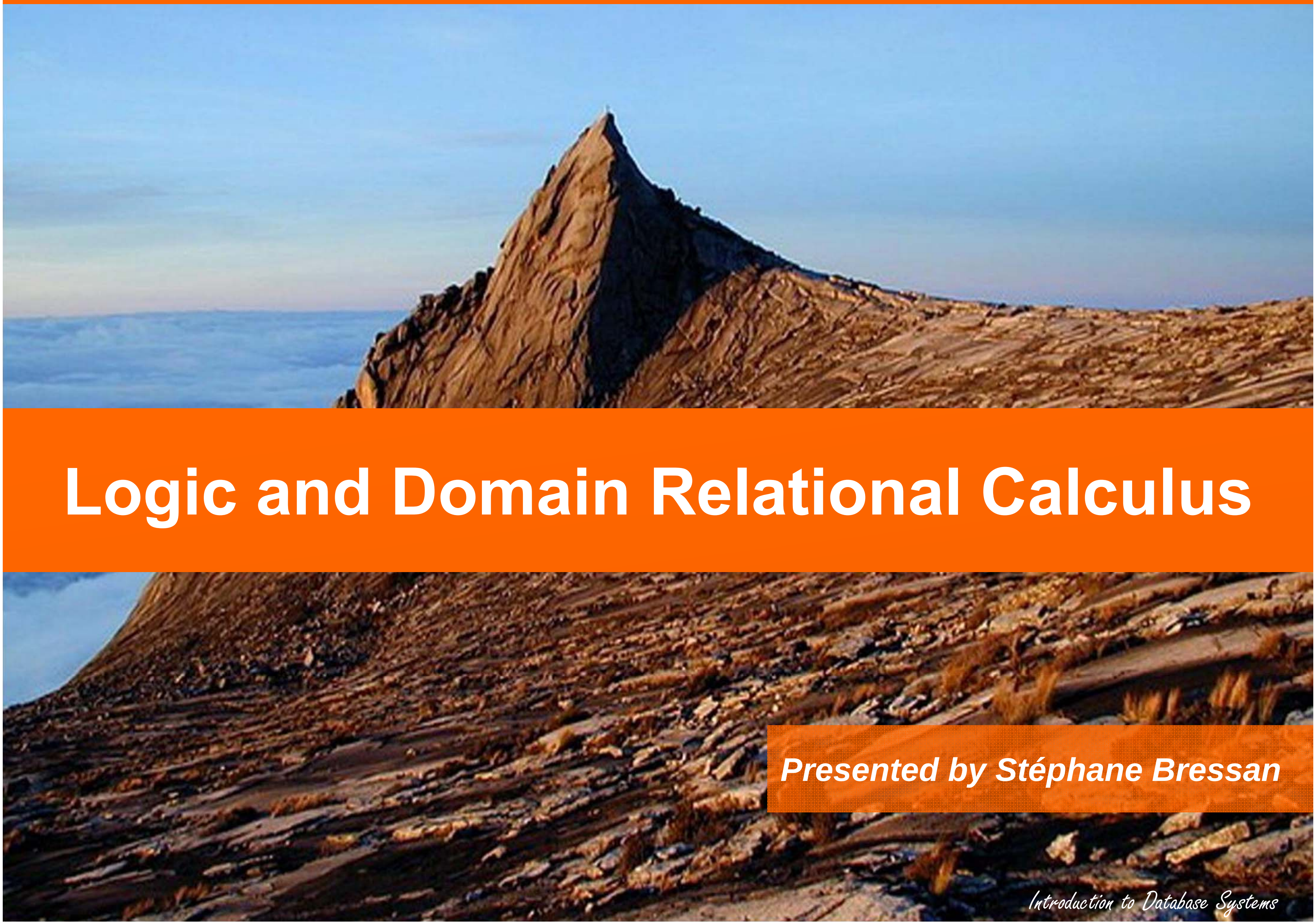
In the Lecture Series Introduction to Database Systems



Logic and Domain Relational Calculus

Presented by Stéphane Bressan

Relational Query Languages

- Two mathematical Query Languages form the basis for practical languages like SQL:
  - Relational Calculus: Declarative, describe what you want, rather than how to compute it.
  - Relational Algebra: Procedural (operational), useful for representing execution plans

- Query languages are NOT programming languages:
  - Not designed to be Turing complete

Relational Calculi

There are two calculi:

- Domain relational calculus (DRC).
- Tuple relational calculus (TRC).

- Both are based on logic
- They differ on what variables represent

Learning Objectives

- Understand and write formula in logic
- Understand and write queries in Domain Relational Calculus

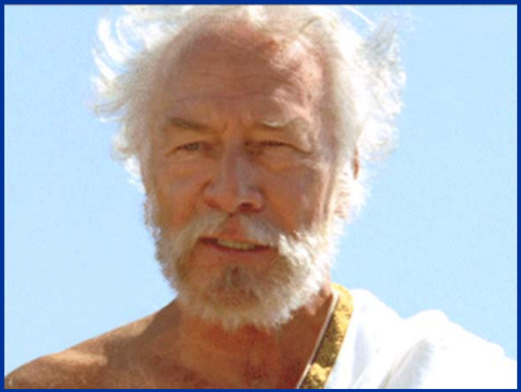
Propositional Logic

"Aristotle is Greek"

"Aristotle is Greek and Alexander is Persian"

"Aristotle is not Greek"

"Alexander is Macedonian or Persian"



Christopher Plummer

Colin Farrel

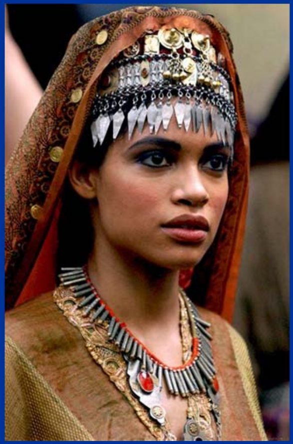
Propositional Logic

"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"



Rosario Dawson

Angelina Jolie

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

$(\neg A \vee B)$

First Order Logic: Predicates

greek(aristotle)

greek(X)

mother(olympias, alexander)

mother(X, Y)

First Order Logic: Quantification

$\exists X \; greek(X)$

$\exists X \; mother(olympias, X)$

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

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

First Order Logic: Quantification

$\forall X \; greek(X)$

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

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

First Order Logic

$$\forall X \, \forall Y \, ((\text{mother}(X, Y) \land \text{greek}(X)) \Rightarrow \text{greek}(Y))$$

Syntax of First Order Logic

First order logic consists of formulae built from

- **predicates**
- **constants** *(lower case)*
- **variables** *(upper case, quantified or free),*
- **quantifiers**: $\forall$ and $\exists$
- **logical operators**: $\wedge$, $\vee$, $\neg$ and $\Rightarrow$

Remarks

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.

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

Different scopes

$$\exists Y\ (mother(X, Y)) \wedge father\ (Z, Y)$$

Remarks

$\neg \forall X F$

is equivalent to

$\exists X \neg F$

Proof by counterexample: Prove $\neg \forall X F$ (i.e., $\forall X F$ is false) by showing $\exists X \neg F$

$\neg \exists X F$

is equivalent to

$\forall X \neg F$

$\neg \exists X$ (greek(X) $\wedge$ mother(X, Alexander)) ➔
$\forall X \neg$(greek(X) $\wedge$ mother(X, Alexander)) ➔
$\forall X$ ($\neg$greek(X) $\vee \neg$ mother(X, Alexander))

*(*Here F represents a formula)*

Calculus

- A Calculus defines formulae and their meaning
- Domain Relational Calculus (DRC): variables range over values
- Tuple Relational Calculus: (TRC) variables range over tuples

Calculus

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

In extension:

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

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

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

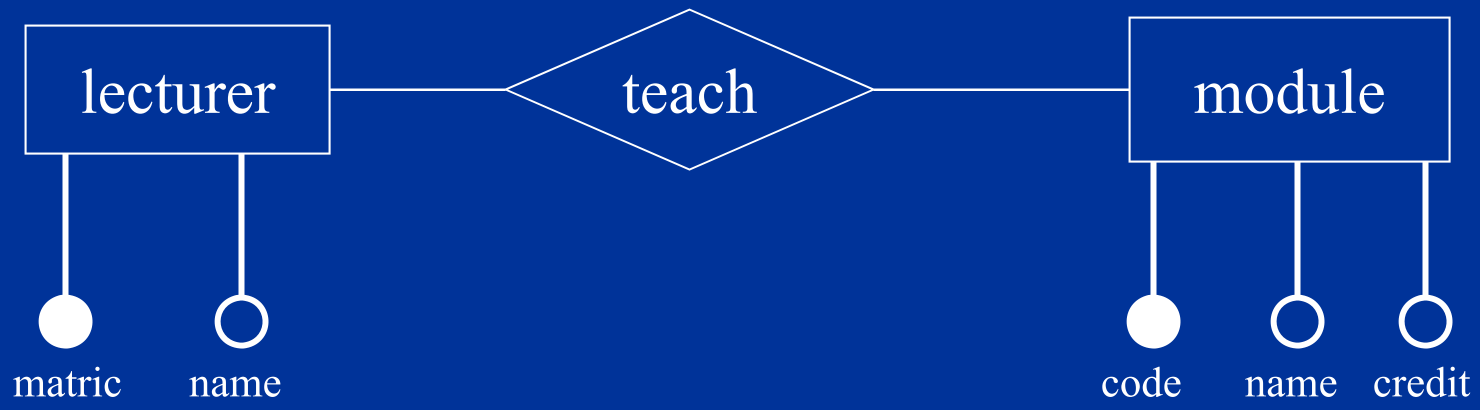
Calculus: Where is the Truth?

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

Domain Relational Calculus

Example

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



Example

- $\{\langle X\rangle \mid \exists Y\ \text{lecturer}(X, Y)\}$

- $\{\langle X\rangle \mid \exists Y\ \text{lecturer}(X, Y) \wedge Y = \text{“john”}\}$

- $\{\langle X\rangle \mid \text{lecturer}(X, \text{“john”})\}$

- $\{\langle X, Y\rangle \mid \text{lecturer}(X, Y)\}$

  -- SELECT * FROM lecturer

Syntax of Domain Relational Calculus

- {Head | Body}   {variableList | formula}
- Head:
  - Variable list: <X1, X2, …>
  - **Variables** in the head are **free**
- Body:
  - Formula in first order logic
  - **Variables** in the body that are not in the head are **quantified**

Semantics of Domain Relational Calculus

- {variableList | formula}

- Head:
  - Return **the set** of **tuples of values**
  - such that if we **replace the variables** in the variable list **by the values** in one such tuples,

- Body:
  - Then **the formula** in the body **is true**.

Example

- $\{\langle X, Y \rangle \mid lecturer(X, Y)\}$
- $\{\langle X \rangle \mid \exists Y\ lecturer(X, Y)\}$

Matric	Name
123	John
214	Peter
555	Magdalena

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

$$\{\langle LN, MN\rangle \mid \exists M \exists C \exists Cr \; lecturer(M, LN) \wedge module(C, MN, Cr) \wedge teach(M, C) \wedge Cr < 2\}$$

Example

$$
\begin{aligned}
\{\langle LN, MN\rangle \mid\ & \exists M1\ \exists M2\ \exists C1\ \exists C2\ \exists Cr \\
& \quad \mathrm{lecturer}(M1, LN) \\
& \wedge \mathrm{module}(C1, MN, Cr) \\
& \wedge \mathrm{teach}(M2, C2) \\
& \wedge C1 = C2 \wedge M1 = M2 \wedge Cr < 2\}
\end{aligned}
$$

Example SQL

- SELECT
  lecturer.lecName,
  module.moduleName
  FROM lecturer, module, teach
  WHERE lecturer.matric=teach.matric
  AND module.code = teach.code
  AND module.credit < 2

Example

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

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

Example

Find the names of the lecturers teaching all modules

$$\{\langle N \rangle \mid \exists M \, \forall C \, \forall MN \, \forall Cr \; (lecturer(M, N) \wedge module(C, MN, Cr) \wedge teach(M, C))\}$$

Incorrect! (Why?)

Example

Find the names of the lecturers teaching all modules

$\{<N> \mid \exists\, M\ \forall\, C\ \forall\, MN\ \forall\, Cr$

$\quad lecturer(M, N) \wedge$

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

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

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