TECHNISCHE UNIVERSITÄT DRESDEN

SEMINAR SELECTED TOPICS IN DATABASE THEORY

Lecture 1: Introduction / The Relational Model

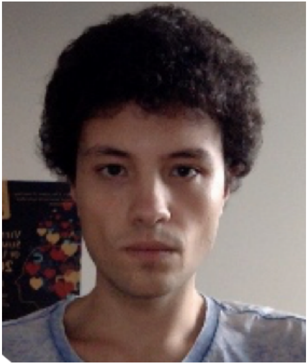
David Carral, Markus Krötzsch

Knowledge-Based Systems

TU Dresden, 10th October 18

Introduction

Course Tutors



David Carral

Markus Krötzsch

Structure of the Seminar and Evaluation

Lectures

- **Wednesday 10th (i.e., today), DS6:** Introductory lecture 1
- **Wednesday 17th, DS6:** Introductory lecture 2
- **Afterwards:** Office hours in 3035 and presentations

Evaluation

- **Paper summary:** self-selected research paper;[a] ~15 pages
- **Presentation:**
  - 20 minutes + discussion
  - Participate in the presentations of other students

[a] See the "Literature" tab at https://iccl.inf.tu-dresden.de/web/Research_Advances_in_Database_Theory_(WS2018).

Other stuff...

Web Page

https://iccl.inf.tu-dresden.de/web/Research_Advances_in_Database_Theory_(WS2018)

Lecture Notes

All slides will be available online.

Reading list

Serge Abiteboul, Richard Hull, Victor Vianu; **Foundations of databases**. Available at http://webdam.inria.fr/Alice/

Acknowledgements

Check out Vim Martens!

On to the content...

What is a database?

A **Database Management System** (DBMS) is a software to manage collections of data. The **architecture of DBMS** consist of three levels:

- **External Level:** Application-specific user views
- **Logical Level:** Abstract data model, independent of implementation, conceptual view
- **Physical Level:** Data structures and algorithms, platform-specific

In this seminar: focus on logical view for relational data model

The Relational Model

Database = collection of tables

Schedule

Movie	Cinema	Date	R-rated
Goodfellas	Thalia	15/10	True
Unforgiven	Thalia	17/10	True
Boogie Nights	Rundkino	21/11	True
Annie Hall	Rundkino	21/11	False

A table has a **schema**:

- Schedule[{Movie, Cinema, Date, R-rated}]

Towards a a formal definition of “table”

A table row has one value for each column.

- That is, a row is a function from the attributes of the table schema to specific values.

Schedule

Movie	**Cinema**	**Date**	**R-rated**
...	...	...	...
Boogie Nights	Rundkino	21/11	True
...	...	...	...

The above row can be represented with the function:

$$f : \{\textbf{Movie} \mapsto \text{Boogie Nights}, \textbf{Cinema} \mapsto \text{Rundkino}, \textbf{Date} \mapsto 21/11, \textbf{R-rated} \mapsto \text{True}\}$$

Database = set of tables

Let **dom** ("domain") be the set of conceivable values in tables.

Definition 1

- A **relation schema** $R[U]$ consists of a relation name R and a finite set U of attributes
- $|U|$ is the arity of $R[U]$
- A **table** for $R[U]$ is a finite set of functions from U to **dom**
- A **database instance** $\mathcal{I}$ is a finite set of tables

Database = set of tables

Let **dom** ("domain") be the set of conceivable values in tables.

Definition 1

- A **relation schema** $R[U]$ consists of a relation name R and a finite set U of attributes
- $|U|$ is the arity of $R[U]$
- A **table** for $R[U]$ is a finite set of functions from U to **dom**
- A **database instance** $\mathcal{I}$ is a finite set of tables

Note: we disregard the order and multiplicity of rows.
Tables are also called relation instances. The table with relation schema $R[U]$ in the database instance $\mathcal{I}$ is written $R^{\mathcal{I}}$.

Database = set of tables

Schedule

Movie	**Cinema**	**Date**	**R-rated**
Goodfellas	Thalia	15/10	True
Unforgiven	Thalia	17/10	True
Boogie Nights	Rundkino	21/11	True
Annie Hall	Rundkino	21/11	False

- The domain **dom** of the above table is the following set: {Goodfellas, Thalia, 15/10, True, Unforgiven, Thalia, 17/10, Boogie Nights, Rundkino, 21/11, Annie Hall, Rundkino, False }
- The above is a table for the relation schema Schedules[{Movie, Cinema, Date, R-rated}]
- Let $\mathcal{I}$ be a database instance. Then, Schedules$^{\mathcal{I}}$ is the set of rows in this table.

Database = set of tables

Schedule

Movie	**Cinema**	**Date**	**R-rated**
Goodfellas	Thalia	15/10	True
Unforgiven	Thalia	17/10	True
Boogie Nights	Rundkino	21/11	True
Annie Hall	Rundkino	21/11	False

The table represented above is the set $\{r_1, r_2, r_3, r_4\}$ where r_1, r_2, r_3, and r_4 are the following functions:

$$r_1 = \{\mathbf{M} \mapsto \text{Goodfellas}, \mathbf{C} \mapsto \text{Thalia}, \mathbf{D} \mapsto 15/10, \mathbf{R} \mapsto \text{True}\}$$

$$r_2 = \{\mathbf{M} \mapsto \text{Unforgiven}, \mathbf{C} \mapsto \text{Thalia}, \mathbf{D} \mapsto 17/10, \mathbf{R} \mapsto \text{True}\}$$

$$r_3 = \{\mathbf{M} \mapsto \text{Boogie Nights}, \mathbf{C} \mapsto \text{Rundkino}, \mathbf{D} \mapsto 21/11, \mathbf{R} \mapsto \text{True}\}$$

$$r_4 = \{\mathbf{M} \mapsto \text{Annie Hall}, \mathbf{C} \mapsto \text{Rundkino}, \mathbf{D} \mapsto 21/11, \mathbf{R} \mapsto \text{False}\}$$

Database = set of relations

Remark: Attribute names do not matter. Instead of the function

$$\{\textbf{M} \mapsto \text{Goodfellas}, \textbf{C} \mapsto \text{Thalia}, \textbf{D} \mapsto 15/10, \textbf{R} \mapsto \text{True}\}$$

we could also use a tuple:

$$\langle \text{Goodfellas}, \text{Thalia}, 15/10, \text{True} \rangle$$

Necessary assumption: Attributes have a fixed order.

Definition 2

- A **relation schema** $R[U]$ is defined as before
- A **table** for $R[U]$ is a finite subset of $\textbf{dom}^{|U|}$
- A **database instance** $\mathcal{I}$ is a finite set of tables

Database = set of relations

Schedule

Movie	Cinema	Date	R-rated
Goodfellas	Thalia	15/10	True
Unforgiven	Thalia	17/10	True
Boogie Nights	Rundkino	21/11	True
Annie Hall	Rundkino	21/11	False

The table represented above is the following set:

$$\{\langle \text{Goodfellas}, 15/10, \text{True}, \text{Thalia}\rangle,$$
$$\langle \text{Unforgiven}, 17/10, \text{True}, \text{Thalia}\rangle,$$
$$\langle \text{Boogie Nights}, 21/11, \text{True}, \text{Rundkino}\rangle,$$
$$\langle \text{Annie Hall}, 21/11, \text{False}, \text{Rundkino}\rangle$$

Database = set of facts

Another convenient way to write databases:

Definition 3

A **fact** is an expression $p(t_1, \ldots, t_n)$ where

- p is an n-ary predicate symbol
- $t_1, \ldots, t_n$ are constan symbols

A **database instance** is a finite set of facts.

Database = set of facts

Schedule

Movie	Cinema	Date	R-rated
Goodfellas	Thalia	15/10	True
Unforgiven	Thalia	17/10	True
Boogie Nights	Rundkino	21/11	True
Annie Hall	Rundkino	21/11	False

The information in the above corresponds to the following facts:

$$\text{Schedule}(\text{Goodfellas}, 15/10, \text{True}, \text{Thalia})$$

$$\text{Schedule}(\text{Unforgiven}, 17/10, \text{True}, \text{Thalia})$$

$$\text{Schedule}(\text{Boogie Nights}, 21/11, \text{True}, \text{Rundkino})$$

$$\text{Schedule}(\text{Annie Hall}, 21/11, \text{False}, \text{Rundkino})$$

Graphical Representation

Director(Scorsese)
DirectedBy(Goodfellas, Scorsese)
ActsIn(De Niro, Goodfellas)
ActsIn(Pesci, Goodfellas)

Summary: Different Perspectives

Perspective	DB Instance	Table	Row
Named	Set of tables	Set of functions	Function
Unnamed	Set of tables	Set of tuples	Tuple
Fact-based	Set of facts	Set of facts	Fact
Graph	Labelled hypergraph	L. hypergraph	L. Edge

The Relational Algebra

Relational Algebra Queries

Query language based on a set of **operations** on databases.
Each operation refers to some tables and produces another table.

Main operations of the named perspective:

- Selection σ
- Projection π
- Join $\bowtie$
- Renaming δ
- Difference $-$
- Union $\cup$
- Intersection $\cap$

Selection

"Find all R-rated movies"

$$\sigma_{\text{R-rated="True"}}\text{Schedule}$$

"Find all connections that begin and end in the same stop"

$$\sigma_{\text{From=to}}\text{Connect}$$

Definition 4

The **selection operator** has the form $\sigma_{n=m}$

- n is an attribute name
- m is an attribute name or a constant value

Consider a table $R^{\mathcal{I}}$ for the relational schema $R[U]$.

- For m constant value: $\sigma_{n=m}(R^{\mathcal{I}}) = \{f \in R^{\mathcal{I}} \mid f(n) = m\}$
- For m constant value: $\sigma_{n=m}(R^{\mathcal{I}}) = \{f \in R^{\mathcal{I}} \mid f(n) = f(m)\}$

Selection

"Find all dates in which some movie is projected."

$$\pi_{\text{Date}}\text{Schedule}$$

Definition 5

The **projection operator** has the form $\pi_{a_1,\ldots,a_n}$ where each a_i is an attribute name.
Consider a table $R^{\mathcal{I}}$ for $R[U]$.

$$\pi_{a_1,\ldots,a_n}(R^{\mathcal{I}}) = \{f_{\{a_1,\ldots,a_n\}} \mid f \in R^{\mathcal{I}}\}$$

where $f_{\{a_1,\ldots,a_n\}}$ is the restriction of f to the domain $\{a_1, \ldots, a_n\}$, i.e., the function $\{a_1 \mapsto f(a_1), \ldots, a_n \mapsto f(a_n)\}$.

Remark: Projection is only defined if $a_i \in U$ for each a_i.

Natural Join

Schedule

Movie	Cinema	Date	R-rated
Unforgiven	Thalia	17/10	True
Boogie Nights	Rundkino	21/11	True

Location

Cinema	Neighborhood
Thalia	Neudstadt
Rundkino	Altstadt

Schedule $\bowtie$ Location

Movie	Cinema	Date	R-rated	Neighborhood
Unforgiven	Thalia	17/10	True	Neudstadt
Boogie Nights	Rundkino	21/11	True	Altstadt

Natural Join

Definition 6

The **natural join** operator has the form $\bowtie$.
Consider tables $R^{\mathcal{I}}$ for $R[U]$ and $S^{\mathcal{I}}$ for $S[V]$.

$$R^{\mathcal{I}} \bowtie S^{\mathcal{I}} = \{f : U \cup V \to \textbf{dom} \mid f_U \in R^{\mathcal{I}} \text{ and } f_V \in S^{\mathcal{I}}\}$$

where f_U (resp. f_V) is the restriction of f to elements in U (resp. V) as before.

Rename

$$\delta_{\text{Movie,Cinema,Date,R-rated}\to\text{Film,Cinema,Date,R-rated}}(\text{Schedule})$$

Definition 7

The **renaming operator** has the form $\delta_{a_1,\ldots,a_n \to b_1,\ldots,b_n}$ with all a_i mutually distinct attribute names, and likewise for all b_i.
Consider a table $R^{\mathcal{I}}$ for $R[\{a_1, \ldots, a_n\}]$

$$\delta_{a_1,\ldots,a_n \to b_1,\ldots,b_n}(R^{\mathcal{I}}) = \{f \circ g \mid f \in R^{\mathcal{I}} \text{ and } g : \{b_i \mapsto a_i\}_{1 \le i \le n}\}$$

where $f \circ g$ is function composition: $(f \circ g)(x) = f(g(x))$

Difference, Union, Intersection

- Binary operators defined like the usual set operations.
- **Remark:** These operators are only defined on tables of the same relational schema. That is, tables with the same set of attributes.

Research on Database Theory

- Relational algebra query answering.
  - Studying combined/data/query Complexity.

Research on Database Theory

- Relational algebra query answering.
  - Studying combined/data/query Complexity.
- Developing and studying novel query languages.
  - Fragments of relational algebra such as **conjunctive queries**. That is, relational algebra expressions that use only select, project, join, and rename.
  - Extensions of relational algebra such as **Datalog**.

Research on Database Theory

- Relational algebra query answering.
  - Studying combined/data/query Complexity.
- Developing and studying novel query languages.
  - Fragments of relational algebra such as **conjunctive queries**. That is, relational algebra expressions that use only select, project, join, and rename.
  - Extensions of relational algebra such as **Datalog**.
- Study the “expressivity” of a query language.

Research on Database Theory

- Relational algebra query answering.
  - Studying combined/data/query Complexity.
- Developing and studying novel query languages.
  - Fragments of relational algebra such as **conjunctive queries**. That is, relational algebra expressions that use only select, project, join, and rename.
  - Extensions of relational algebra such as **Datalog**.
- Study the "expressivity" of a query language.
  - Let's do this!