Software Engineering Techniques Applied to databases

ASE doctoral symposium

Julien Delplanque

julien.delplanque@inria.fr

Université de Lille

Inria
inventeurs du monde numérique

RMod
https://rmod.inria.fr/web/

Overview

Software evolves to meet requirements of the real world.

As any software, relational databases evolve and their complexity increase with time.

Changing database entity

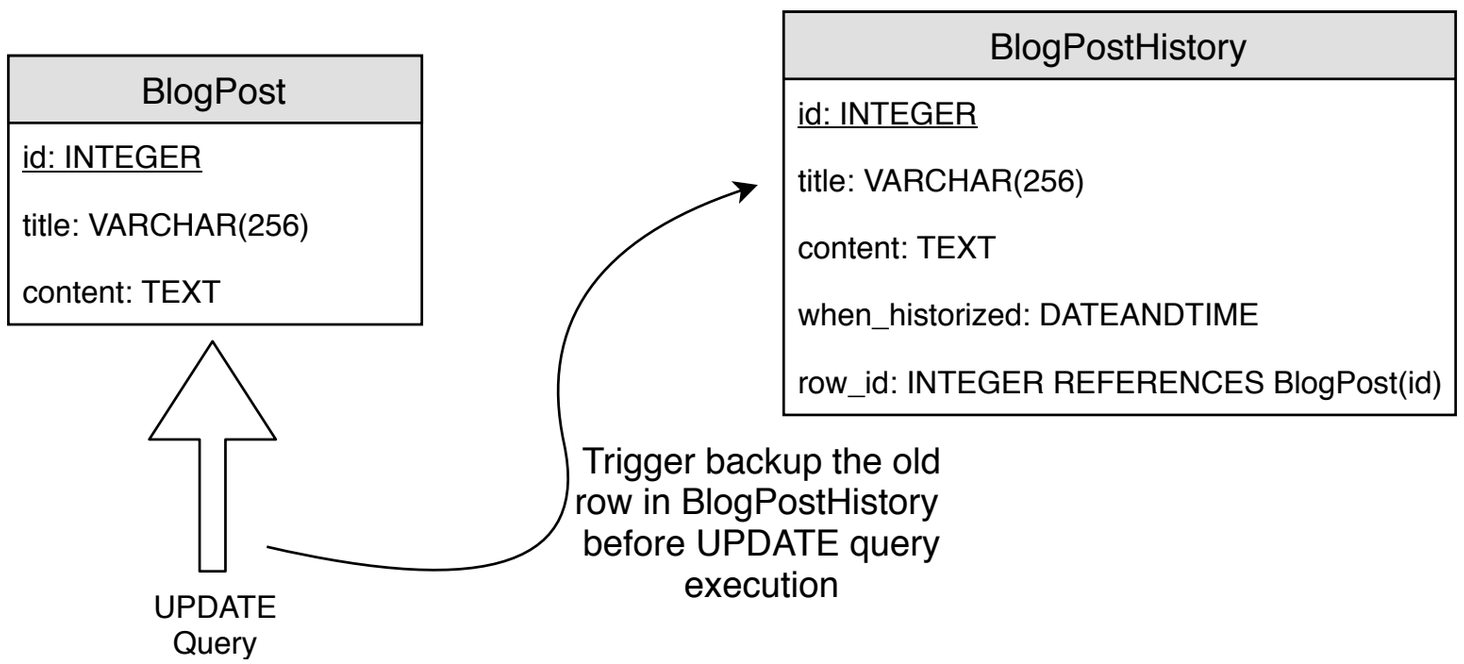
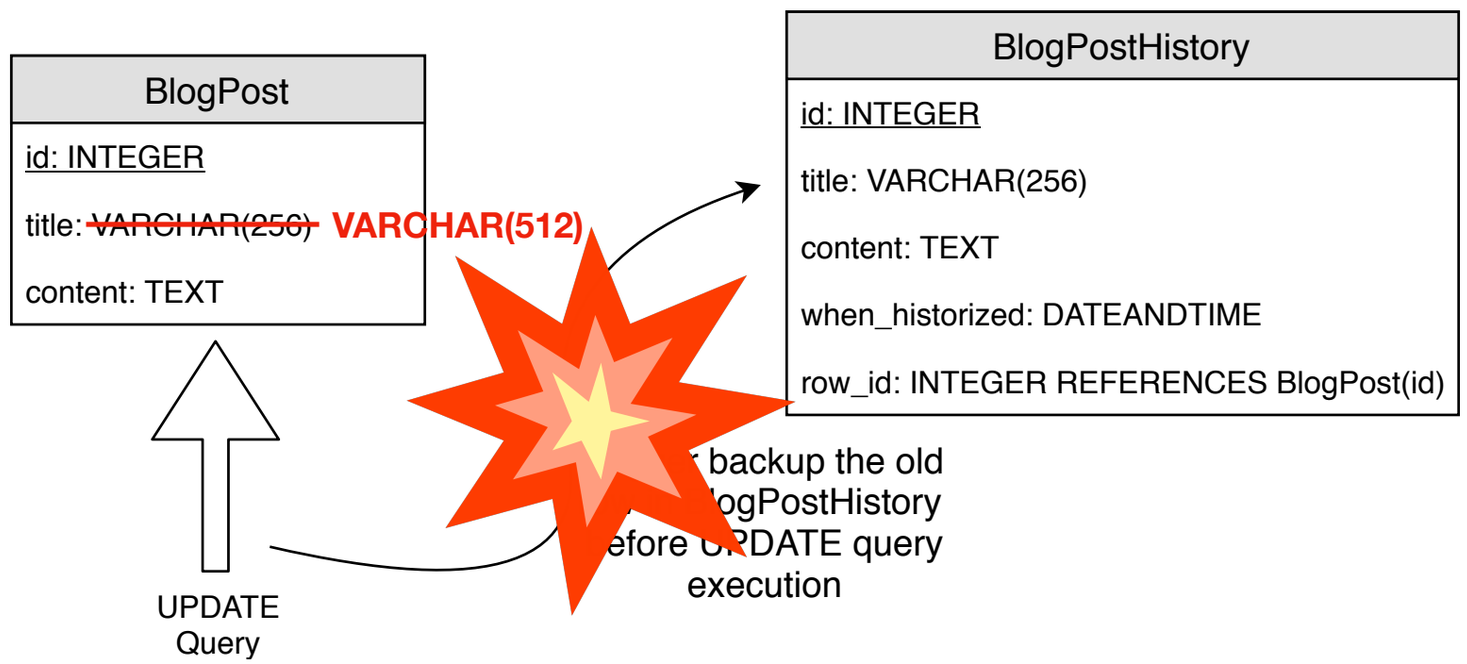
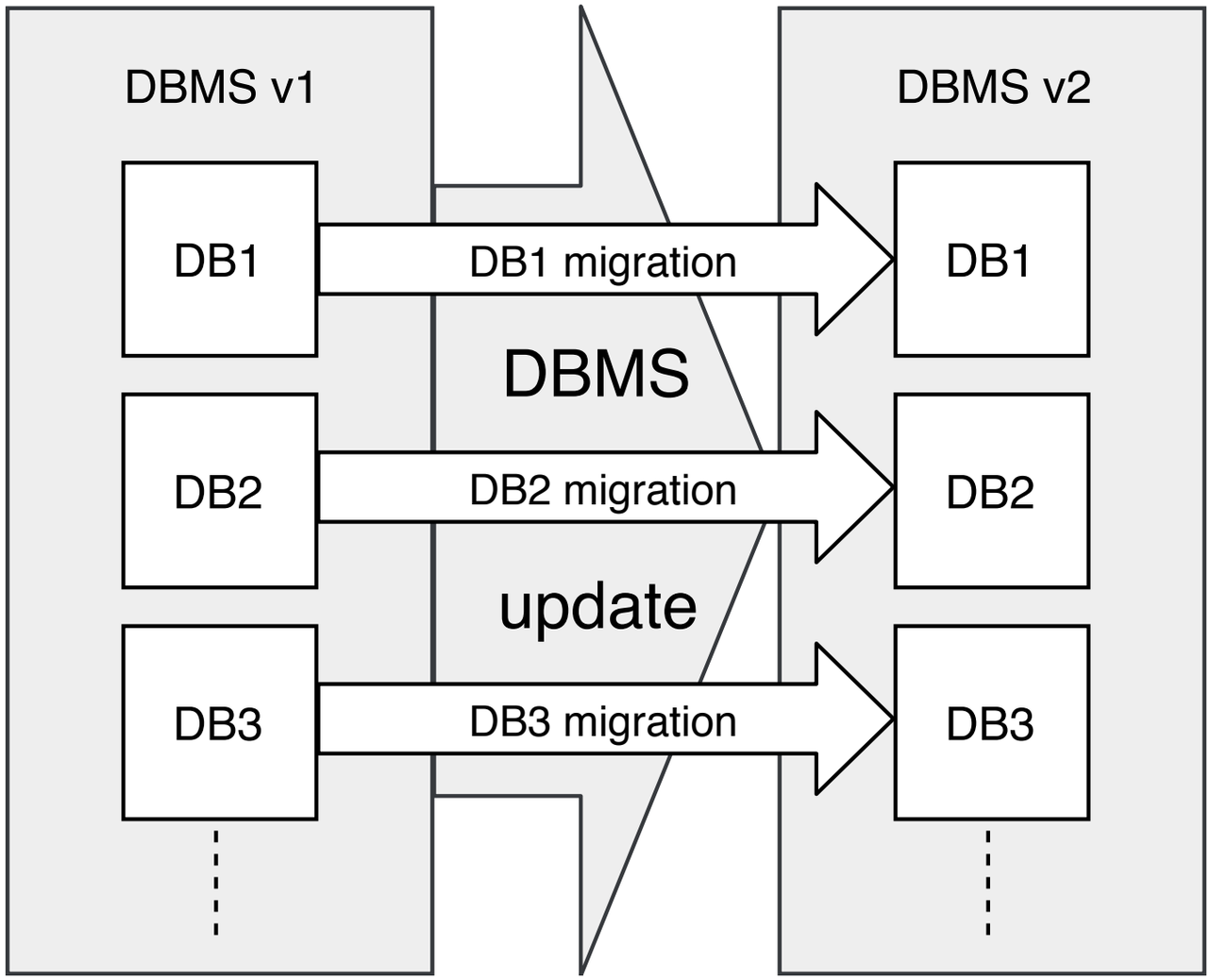


Table historization example

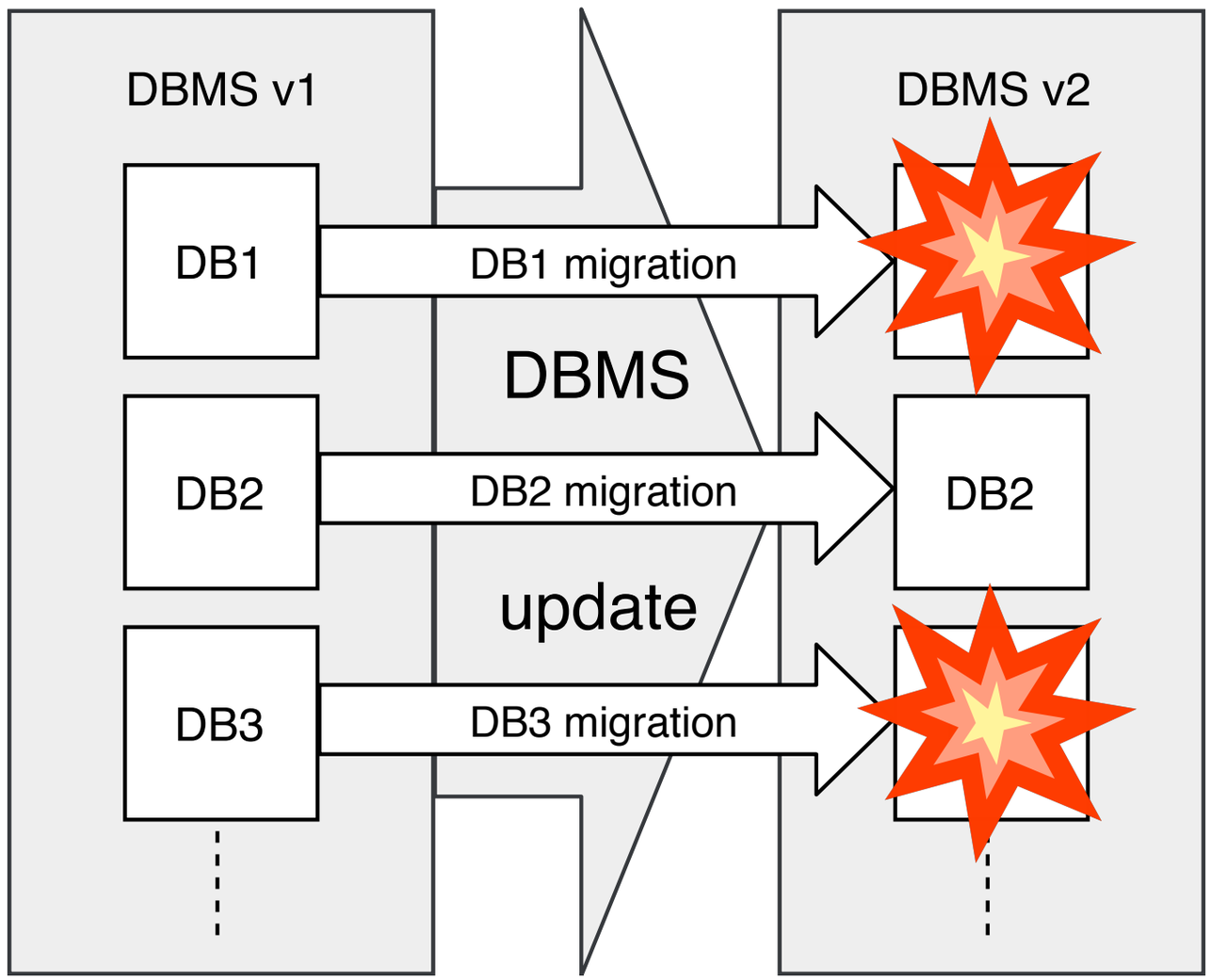
Changing database entity



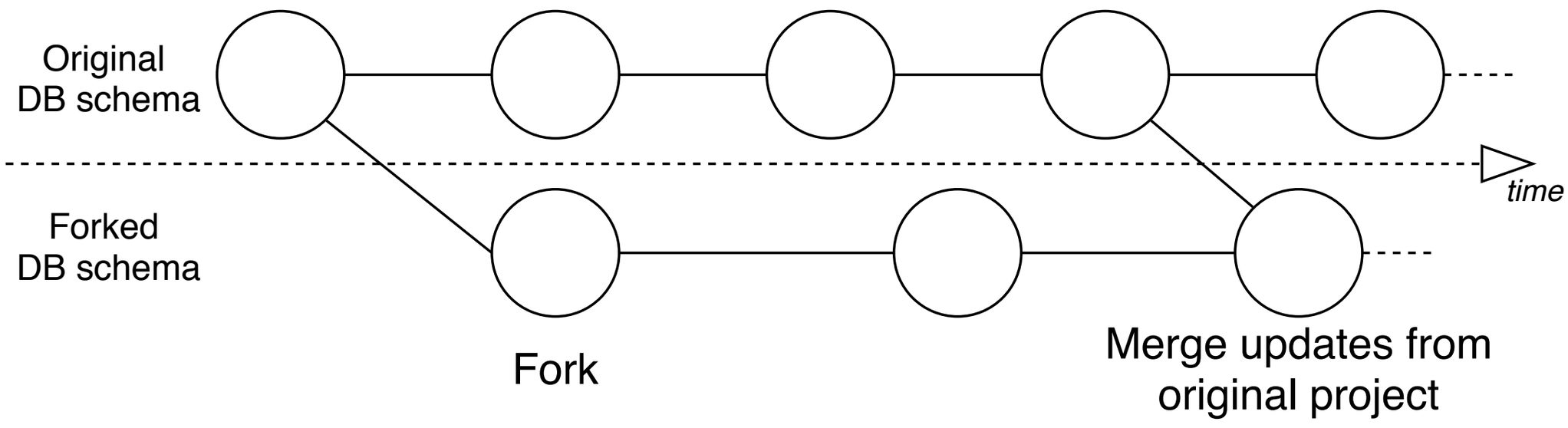
Migrating DBMS version



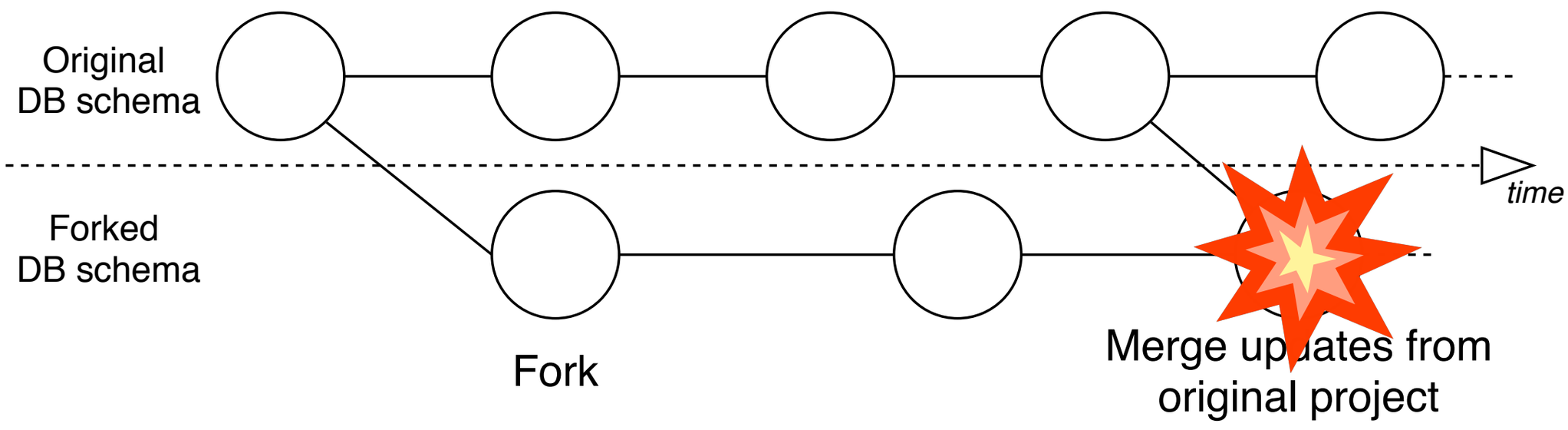
Migrating DBMS version



Maintaining consistency between forks of a schema



Maintaining consistency between forks of a schema



Challenges

- Impact Analysis — *How to know which entities potentially need to be change when an entity is changed?*

- Recommandations — *How to adapt an entity to a change on an entity it depends on?*

- Embedded Behaviour Management — *How to keep the behaviour embedded inside the database working after a change?*

Impact Analysis

- A lot of work in the literature;
- Notably, (Lehnert, 2010) provides a meta-analysis of the research field.

Steffen Lehnert. 2011. A taxonomy for software change impact analysis. In Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution. ACM, 41–50.

Recommandations

- ≠ refactorings in Software Engineering
- The goal is **not** to keep the database structure / behaviour the same but to **adapt it** to new requirements
- Some work was started by (Meurice et. Al, 2016)
- Two main questions:
  - How to combine recommandation operators?
  - What is the impact / recommandation of combined operators?

Loup Meurice, Csaba Nagy, and Anthony Cleve. 2016. Detecting and preventing program inconsistencies under database schema evolution. In Software Quality, Reliability and Security (QRS), 2016 IEEE International Conference on. IEEE, 262–273.

Embedded Behaviour Management

- Databases can hold behaviour interacting with its data and can be used by its clients.
- Views, stored procedures, triggers, rewrite rules, …
- How to adapt those behavioural entities when the database needs to evolve?

Research plan

1. Observe database architects
   — Get a good understanding of the problem

2. Explore and develop theoretical solutions;
   — Design the meta-model / change model / impact-model

3. Implement them;
   — Implement the meta-model / change model / impact-model

4. Validate the implementation against real databases from open-source projects and companies
   — Compare the prediction of the impact-model with patch files of our laboratory's database. Comparison of low-level changes used by the DBA against those used by the SQL generator to implement the high-level changes.

Previous work

- “CodeCritics applied to databases: Challenges and first results”
  Julien Delplanque, Anne Etien, Olivier Auverlot, Tom Mens, Nicolas Anquetil and Stéphane Ducasse *(published)* — Industrial Track of Software Analysis, Evolution and Reengineering 2017

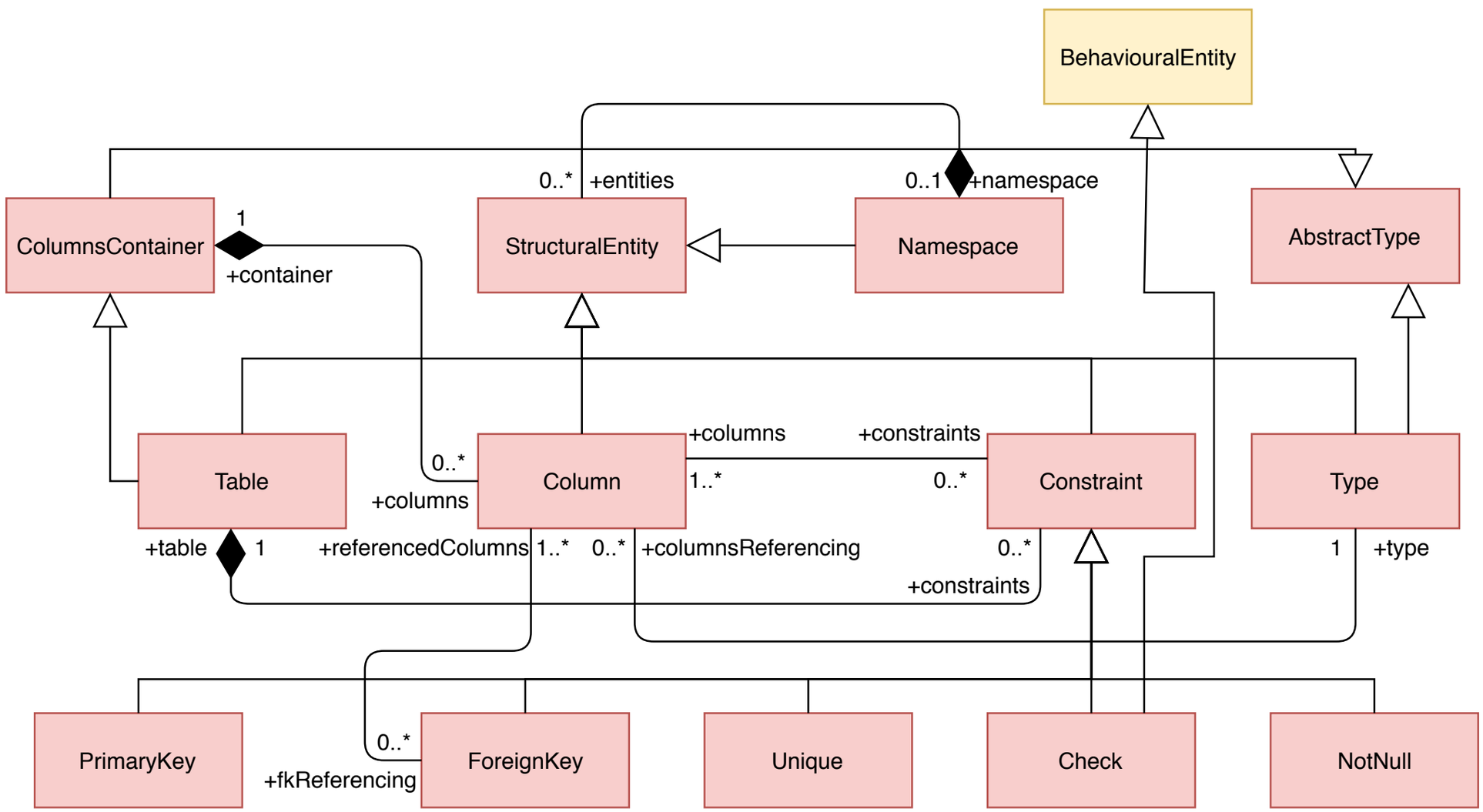
- “Software Engineering Issues in RDBMS, a Preliminary Survey”
  Julien Delplanque *(published)* — Belgian-Netherlands software evolution symposium 2017

- “Définition et identification des tables de nomenclatures”
  Julien Delplanque, Olivier Auverlot, Anne Etien and Nicolas Anquetil *(published)* — Informatique des Organisations et Systèmes d’Information et de Décision 2018

- “Relational Database Schema Evolution: An Industrial Case Study”
  Julien Delplanque, Anne Etien, Nicolas Anquetil and Olivier Auverlot *(accepted)* — Industrial Track of International Conference on Software Maintenance and Evolution 2018

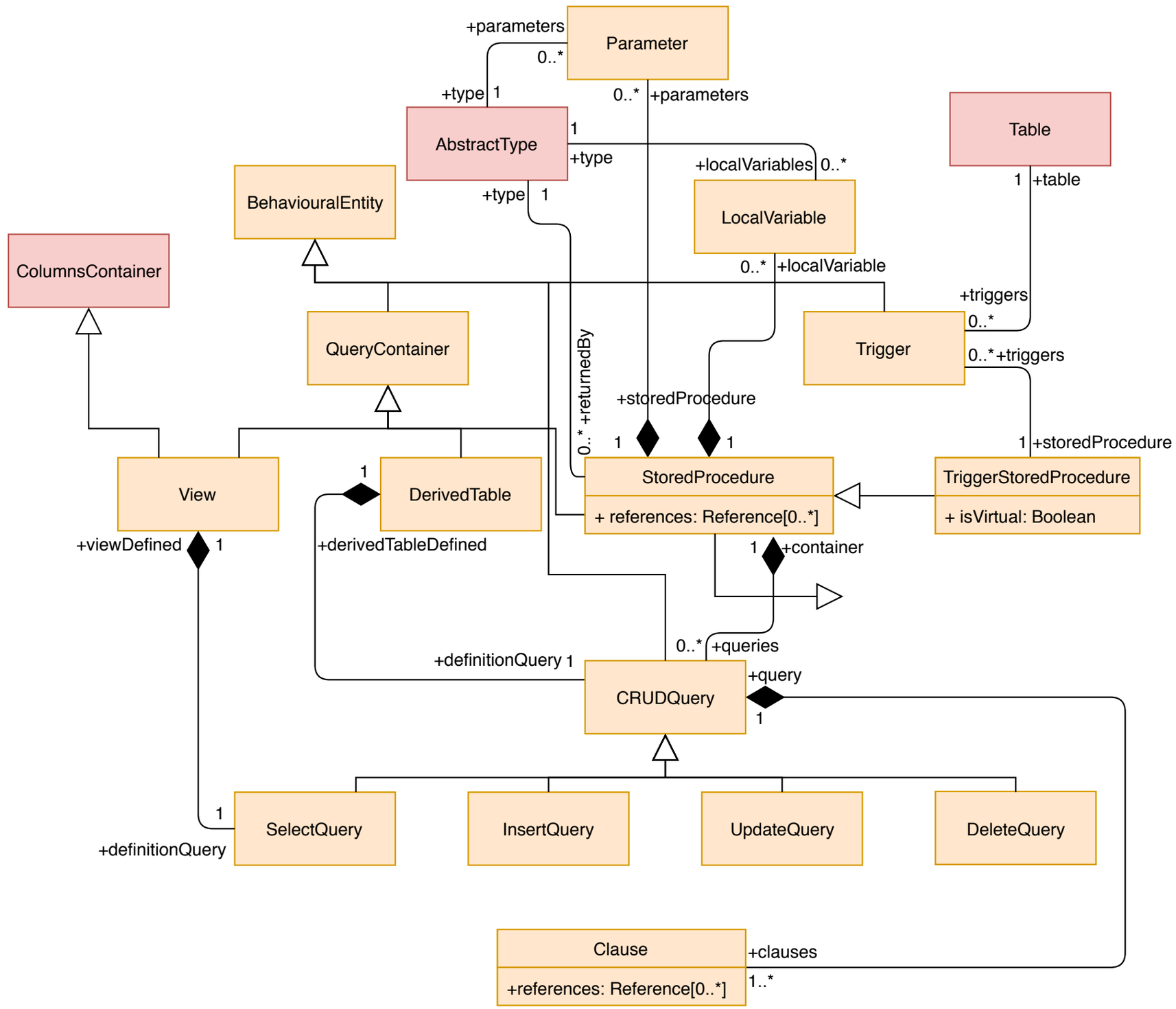
Progress - Not yet reported

SQL meta-model



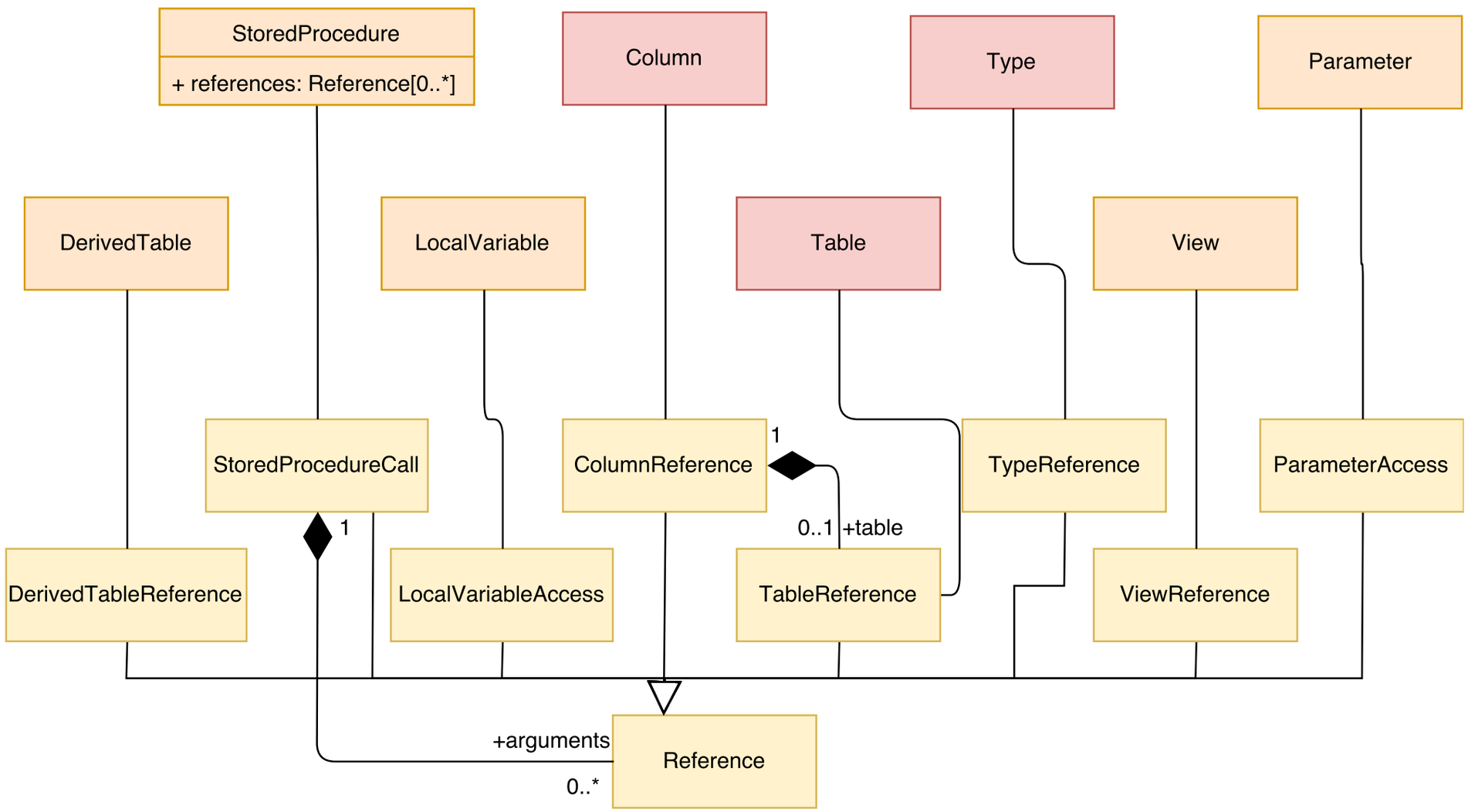
Progress - Not yet reported

SQL meta-model



Progress - Not yet reported

SQL meta-model



Conclusion

✓ Observations / Problem understanding

✓ Meta-model design / implementation

- Dependency analysis
- Change Model
- Recommandations
- Validation on real databases schemas

julien.delplanque@inria.fr

juliendelplanque.be/phd.html