Review report of a final thesis

Reviewer:	Mgr. Radek Mácha
Student:	Bc. Lucie Procházková
Thesis title:	Data flow analysis of scripts in Databricks SQL dialect
Branch / specialization:	Computer Science
Created on:	5 June 2023

Evaluation criteria

1. Fulfillment of the assignment

▸ [1] **assignment fulfilled**
[2] assignment fulfilled with minor objections
[3] assignment fulfilled with major objections
[4] assignment not fulfilled

From the thesis assignment, I identified the following objectives:
1. Analysis of Databricks SQL dialect
2. Evaluation of required extensions to Databricks extractor
3. Feasibility of using static analysis for Databricks SQL data flow
4. Proposal of an approach to parsing Databricks SQL
5. Description of Databricks SQL statements relevant to data flow
6. Design of parser and data flow generator components
7. Implementation of Databricks parser and data flow generator prototypes
8. Testing of aforementioned prototypes

Objective 1 has been achieved through analysis presented in Chapter 3.
Objective 2 has been addressed in Chapter 2.6.
While the settlement of objectives 3 and 4 could have been addressed more explicitly, they were covered by the design described in Chapter 4.
Objective 5 has been extensively covered by Chapter 3.
Objective 6 is addressed in Chapter 4.
Objective 7 mainly takes form of implementation within the Manta product. Given that a functional and tested codebase, partially integrated with the rest of the product, has been presented, this objective has been completed from my perspective.
Objective 8 pertains mostly to implementing automated tests which make sure the solution to Objective 7 remains functional. The relevant automated tests have been implemented and will serve as regression gates to future enhancements of the Databricks scanner codebase.

All of the aforementioned objectives being achieved, I assess the thesis assignment as fulfilled.

2. Main written part 85/100 (B)

The thesis takes us through the analysis, design, implementation and testing phases of the Databricks SQL data flow analyzer, realized under the Manta project. The text can be roughly split in three sections:
* Chapter 1, covering the general theory fundamental to understanding the work this thesis covers
* Chapters 2-3, dedicated to a general analysis of Databricks and the SQL dialect associated therewith
* Chapters 3-6, describing the design, implementation and testing of the Databricks SQL data flow analyzer component of the Manta product

The thesis follows a logical overarching analysis-design-implementation structure which aims to reflect the goals established in the thesis assignment.

Chapter 1 offers a structured introduction into parser theory and enables the reader to quickly catch up on relevant concepts without the need to look for external sources. While it slightly breaks the natural flow of the thesis, this drawback is easily compensated by the chapter's usefulness.

Chapter 2 serves as a general introduction to the Databricks technology, and is easily the weakest point of the text. While it introduces the concept of Data Lakes and Data Lakehouses well, the chapter feels like it's missing an introduction to Spark. The Databricks Runtime introduction is slightly chaotic as a result. That being said, a few reads of the chapter along with cross-referencing to provided sources does provide the reader with sufficient understanding to put the thesis into context.

Where Chapter 2 might be slightly lacking, Chapter 3 definitely compensates by providing a comprehensive guide to Databricks SQL language features, striking a remarkably good balance between brevity and exhaustiveness. The inclusion of relevant data flow graph cutouts inlined into the text helps the reader quickly grasp the concept, while the full graphs in Appendix B serve as a good anchor point the reader can go back to and cross-reference the code snippets with to solidify their understanding of the topic at hand. The author also makes sure to make a clear discernment between which statements of Databricks SQL are relevant for data flow generation and which ones can be omitted from analysis.

The contents of chapters 4 and 5 are mostly unsurprising to a reader with knowledge of Manta architecture. As mentioned in Chapter 4 preface, most of the design decisions were heavily influenced by the existing architecture of Manta software and the frameworks it leverages. The choices of using Maven, JUnit or Spring being mostly technical in nature, the decision most relevant for the thesis is the use of ANTLR as a framework for parsing. ANTLR being a natural fit for parsing complex languages while interfacing with Java, that choice makes a lot of sense. Additionally, the author takes care to explain why ANTLR 3 was chosen over ANTLR 4. The module layout and workflow design presented in this chapter closely matches patterns adopted in Manta, as well as allows for integration with the existing Databricks extractor.

Chapter 6 proposes 4 main angles on testing:
* Resolver tests to verify AST construction
* Invariants tests to verify constructed AST node can reconstitute original source code
* Regression tests which verify that analysis of corner-case inputs produces expected AST outputs
* Flow tests verifying produced data flow graphs match expected structure

These testing scenarios follow the usual Manta automated testing pattern for parsers/ resolvers, and work well to complement each other in ensuring implementation robustness.

3. Non-written part, attachments 90/100 (A)

While unpublished, the code produced as the implementation of this thesis has been peer-reviewed and is demonstrably functional (as can be observed from data flow graphs attached to this thesis, generated by said code).
The technologies used by the implementation were tailored to maximize the integration potential with existing Manta codebase, making them an adequate choice for the case presented.

4. Evaluation of results, publication outputs and awards 100/100 (A)

The codebase produced as part of this thesis is already being partially deployed in the product. Full-scale adoption will follow as soon as further product integration is addressed. The author has demonstrated the ability to prioritize according to real-world requirements, as well as integrate their work with that of other developers, resulting in a successful prototype and soon-to-be MVP. The work performed by the author addresses real-world client needs and enhances the capabilities of the Manta product.

The overall evaluation 90/100 (A)

From my perspective, the author has met the goals they set out to achieve. The quality of the implementation and its successful integration into the existing solution are my key assessment criteria. The thesis text, while mildly lacking in business context introduction, more than adequately describes the pivotal decisions behind the pilot design and implementation.

Questions for the defense

* Please outline the challenges posed by lambda function data flow processing and possible solutions.
* Elaborate on your solution to resolving delimited vs. non-delimited identifiers (described in Chapter 3.4).

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.