1.1 Motivation

While working at IBM in the early 1970, Edgar F. Codd invented the concept of the Relational Algebra used to model data stored in relational databases. In the paper "A Data Base Sublanguage Founded on the Relational Calculus", he introduced Alpha as the first database language. Also at IBM, Donald D. Chamberlin and Raymond F. Boyce developed SQL, one of the first commercial languages to operate on the relational model. Originally conceived to retrieve and manipulate data on IBM's System R, SQL quickly became the most widely used relational data query language and a de facto standard.

Despite its popularity, SQL has a number of flaws, some of which caused by the fact, that SQL was designed to resemble natural language. This design philosophy lead to arguably absurd constructs such as the IS NOT DISTINCT FROM operator, and, while working well for simple queries, leads to increasingly complex constructs for non-trivial queries. Because of this increase in complexity, tools that partially or completely automate the generation of queries have been investigated by researchers and commercial entities alike.

Due to the very widespread use of SQL, most tools are focused on a specific field of application, therefore limiting the complexity, but also power of the tool. Moreover, the inconsistent implementation of the SQL Standard by most major database vendors (the exception being PostgreSQL) often leads to tools that are tailored to work with a specific implementation only, producing non-standard SQL code that does not work on other implementations.

During my semester break, I worked with Orbis, a clinical information system produced and marketed by Agfa Healthcare. Orbis is built on top of a relational database backend provided by Oracle and includes a visual query builder used to design queries to be executed on this database. While it is a powerful tool, I was never quite satisfied with it, wondering if there was a more fitting approach.

At the same time, I was starting to get interested in Blender, an open-source 3D modelling software. One feature that particularly caught my attention, was Blender's apporach to certain relatively complicated tasks such as texturing and post-processing, using a node or flow-based programming interface. This choice of interface has proven to be vastly superior to the interfaces used in previous iterations and has therefore been applied to more and more parts of the software.

The finding, that the flow-based programming paradigm can be successfully applied to complex tasks combined with the apparent need for a more capable query builder lead to idea of a flow-based visual query builder, which is the subject of this thesis.