

IEEE 15th International Symposium on Embedded
Multicore/Many-core Systems-on-Chip (MCSoc-2022)

Algorithm to Interconvert SQL and Procedural Visual Queries

Tomonori SUZUKI, Yutaka WATANOBE, Divij G. SINGH

Department of Computer Science and Engineering,
University of Aizu, Fukushima 965-8580, Japan
m5261168@u-aizu.ac.jp, yutaka@u-aizu.ac.jp, divijsingh@gmail.com

Outlines

- Introduction
- Background
- Related work
- Interfaces
- System Architecture
- Translation Algorithm
 - Algorithm Limitations
- Conclusion


Introduction

- To manipulate a database, the user need to have the following knowledge.
 - Understanding of database structure
 - A specific query language (SQL)

Introduction (cont.)

- SQL (Structured Query Language)
 - SQL is not sequential executable.
 - Subqueries reduce readability.
 - DB-specific operations: JOIN.

Background

- There are situations where SQL novices want to manipulate a database, however, It is difficult because they need to understand two things.
 - In previous study, data retrieve can be done without the need to understand the above, however there are some problems.
 - Program is blackbox.
 - Operation is limited.
- 
- Suggest an interface that allow them to manipulate the database and interconversion algorithms.

Related Work

Assist with database operations

- Query by Icon (QBI)
- Query by Diagram (QBD)
- Query by Object (QBE)
- Query by Example (QBE)

Related Work (Cont.)

Assist with learning SQL

- Evaluate SQL and provide feedback
- Generate SQL exercises

Interface

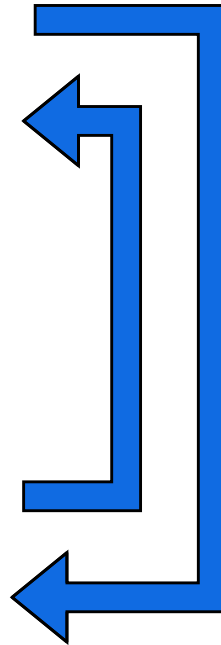
- Query is sequential executable.
- Query can be written with icons.
- Subquery can be folded.
- UI covers most of the features of SQL-DML.

Interface (cont.)

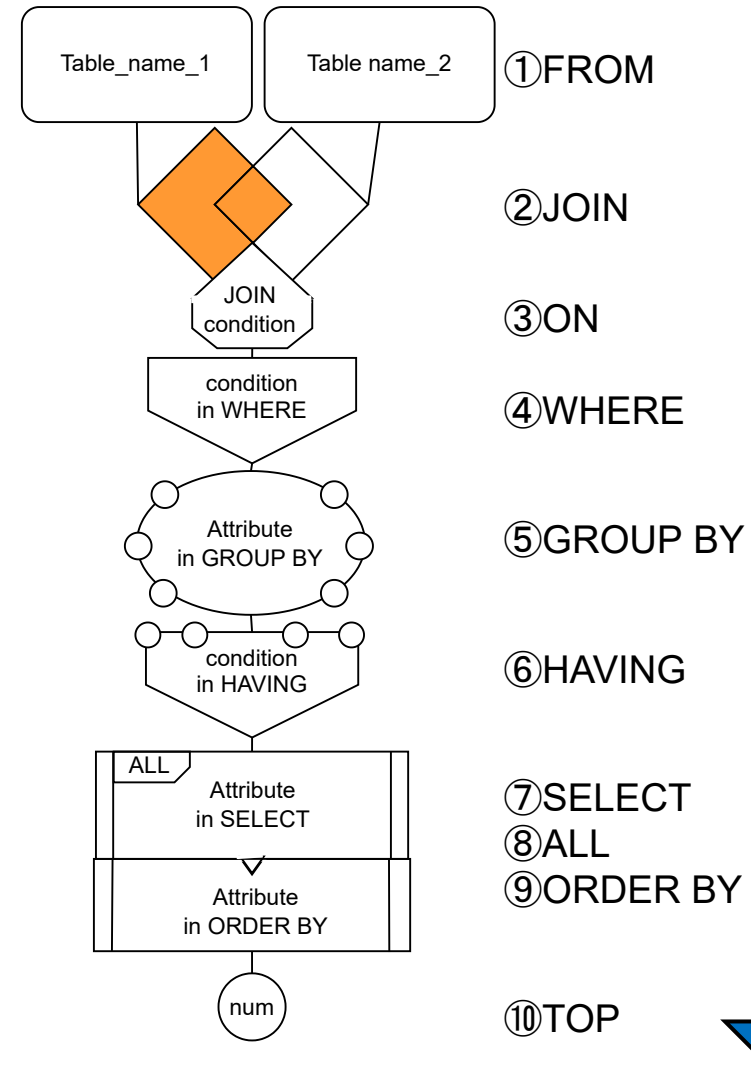
Query is sequential executable.

- SQL (retrieve data)

SELECT ALL TOP num (7)(8)(10)
 output_attribute
FROM table1 (1)
JOIN table2 (2)
ON join_condition (3)
WHERE where_condition (4)
GROUP BY group-by condition (5)
HAVING having_attribute (6)
ORDER BY order-by_attribute (9)



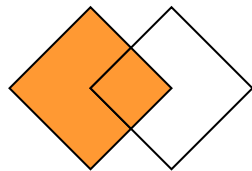
- Visual Query Interface



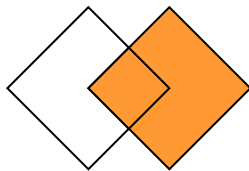
Interface (cont.)

Query can be written with icons.

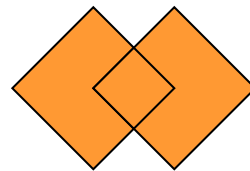
Join kinds



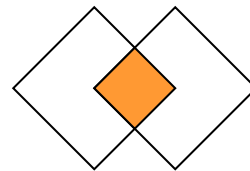
LEFT JOIN



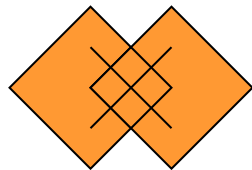
RIGHT JOIN



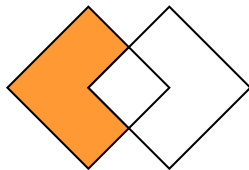
FULL OUTER JOIN



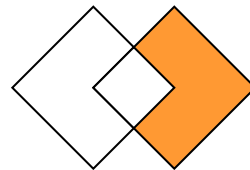
INNER JOIN



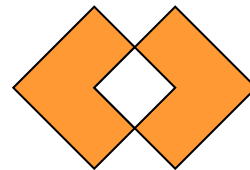
CROSS JOIN



ANTI
LEFT JOIN

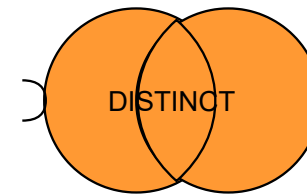


ANTI
RIGHT JOIN

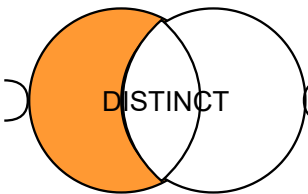


ANTI
OUTER JOIN

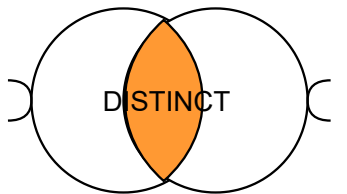
Set operations



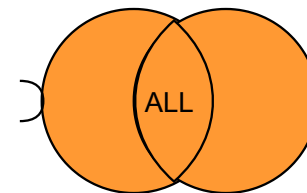
UNION (DISTINCT)



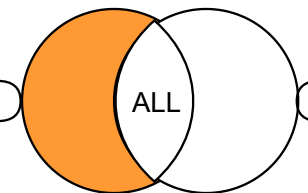
EXCEPT (DISTINCT)
(MINUS DISTINCT)



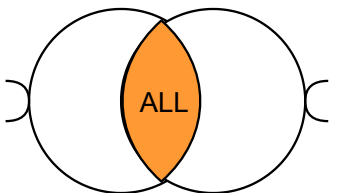
INTERSECT (DISTINCT)



UNION ALL



EXCEPT ALL
(MINUS ALL)

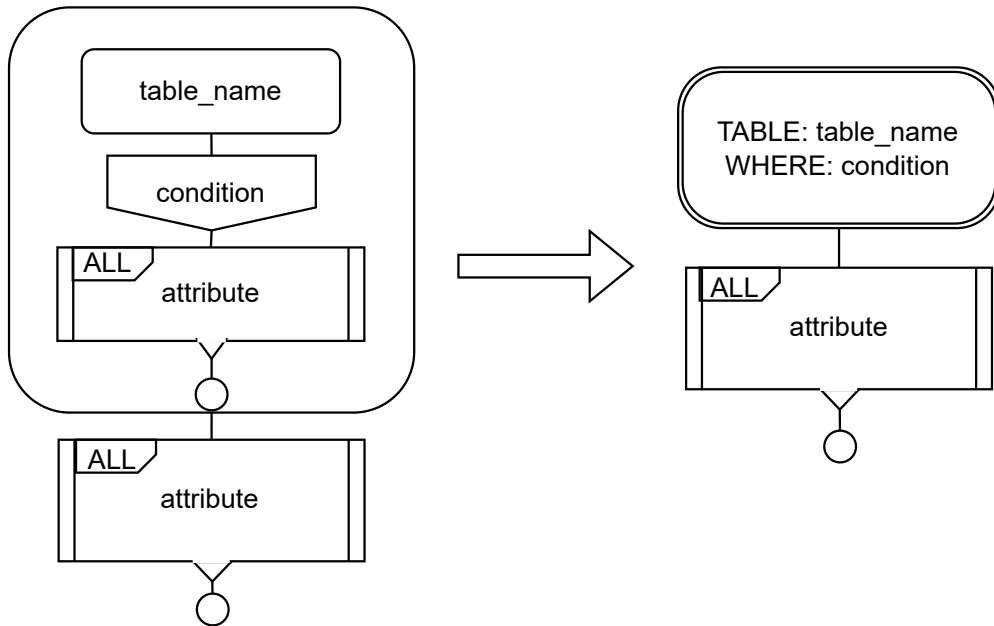


INTERSECT ALL

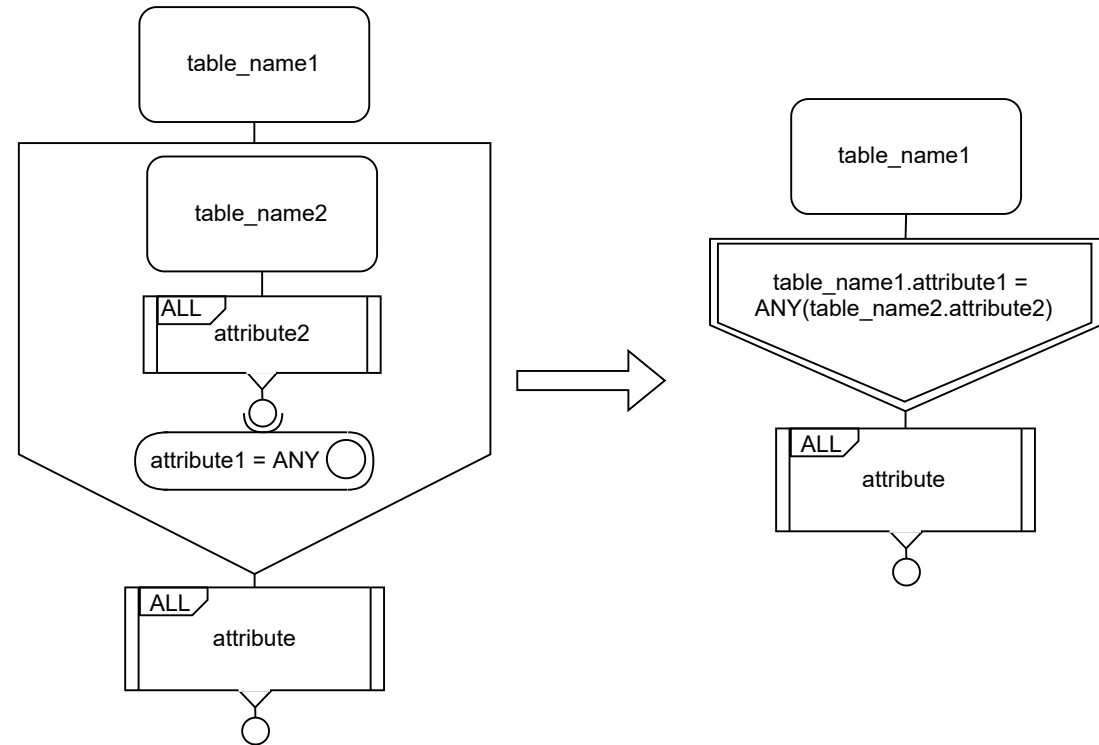
Interface (cont.)

Subquery can be folded.

Folded FROM



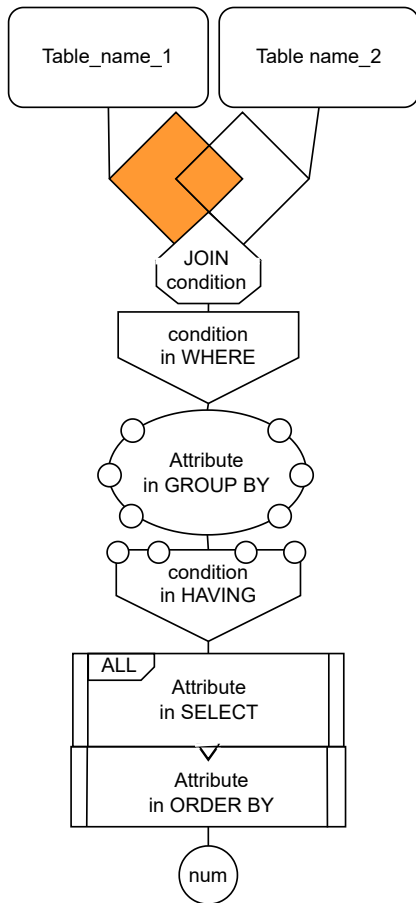
Folded WHERE



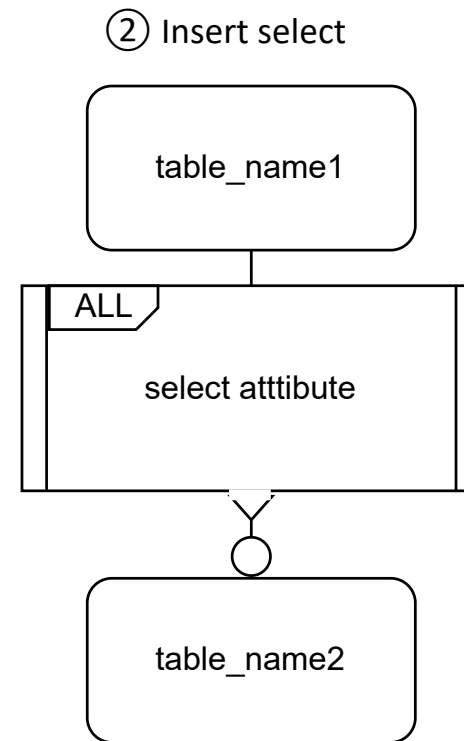
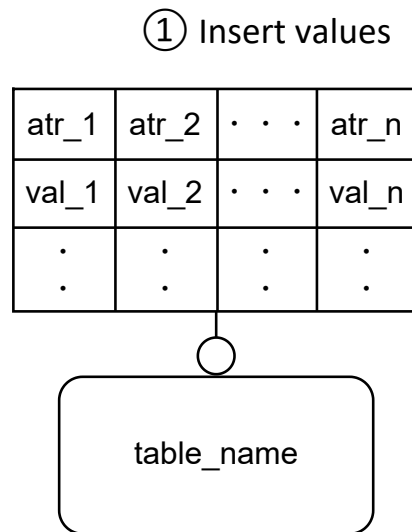
Interface (cont.)

UI covers most of the operation in SQL-DML.

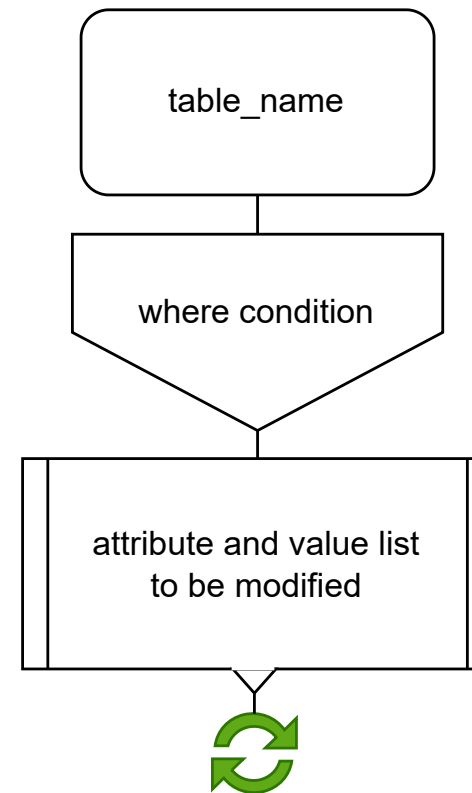
Retrieve data



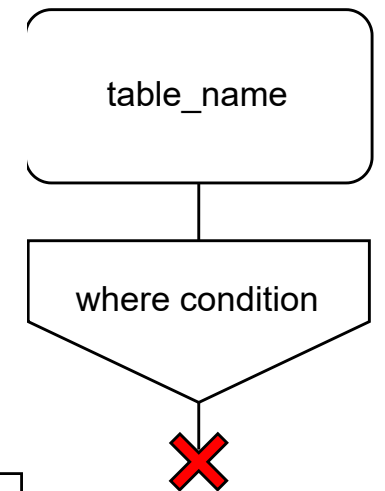
Add data



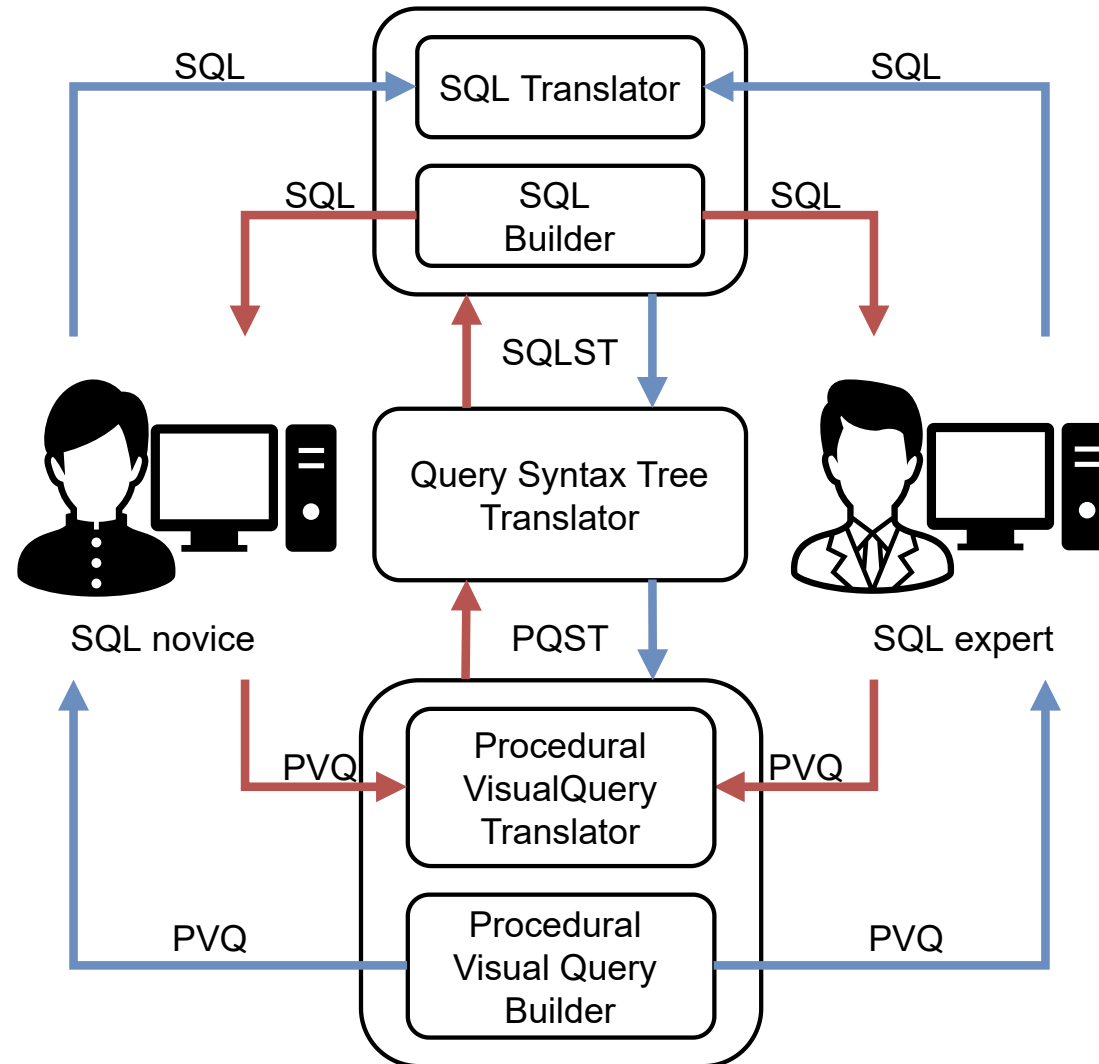
Modify data



Delete data



System Architecture

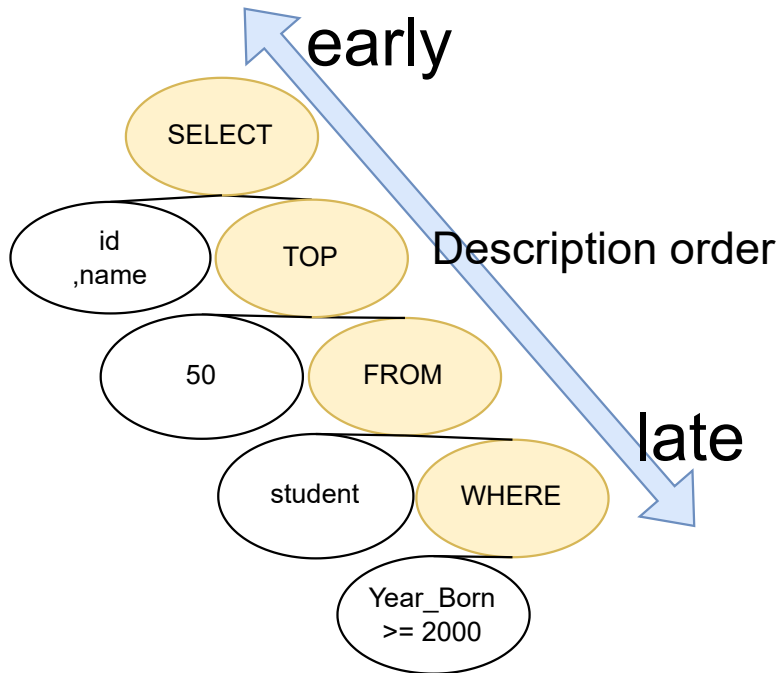


Translation Algorithm

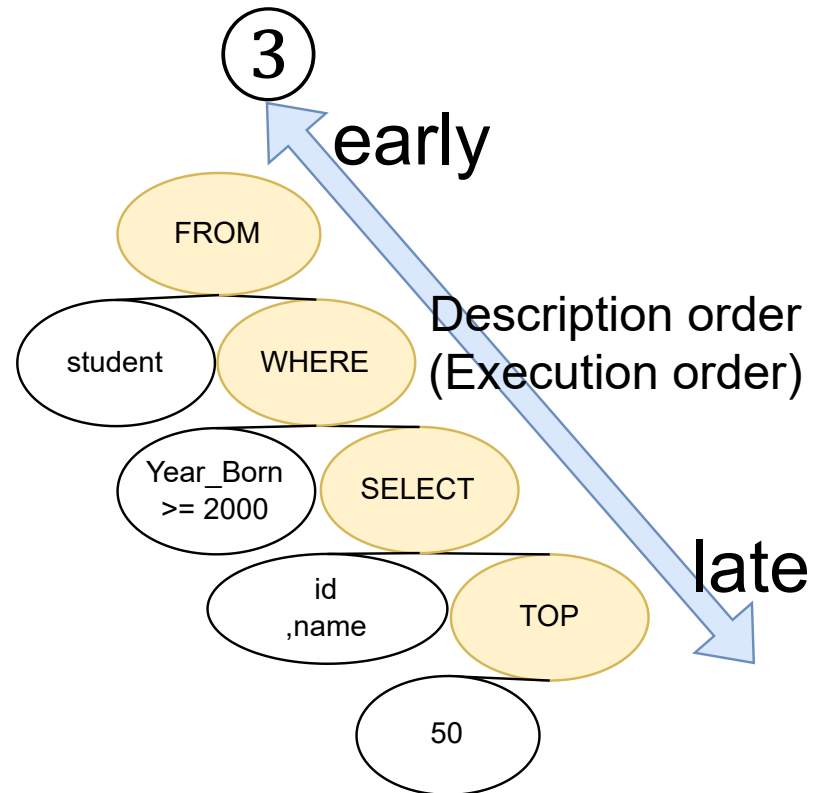
1

- 1 SELECT ALL TOP 50
- 2 id
- 3 ,name
- 4 FROM student
- 5 WHERE Year_Born >= 2000

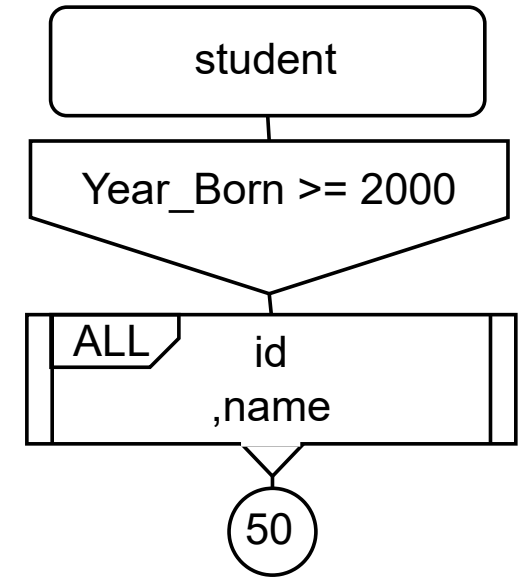
2



3



4



Translation Algorithm (Cont.)

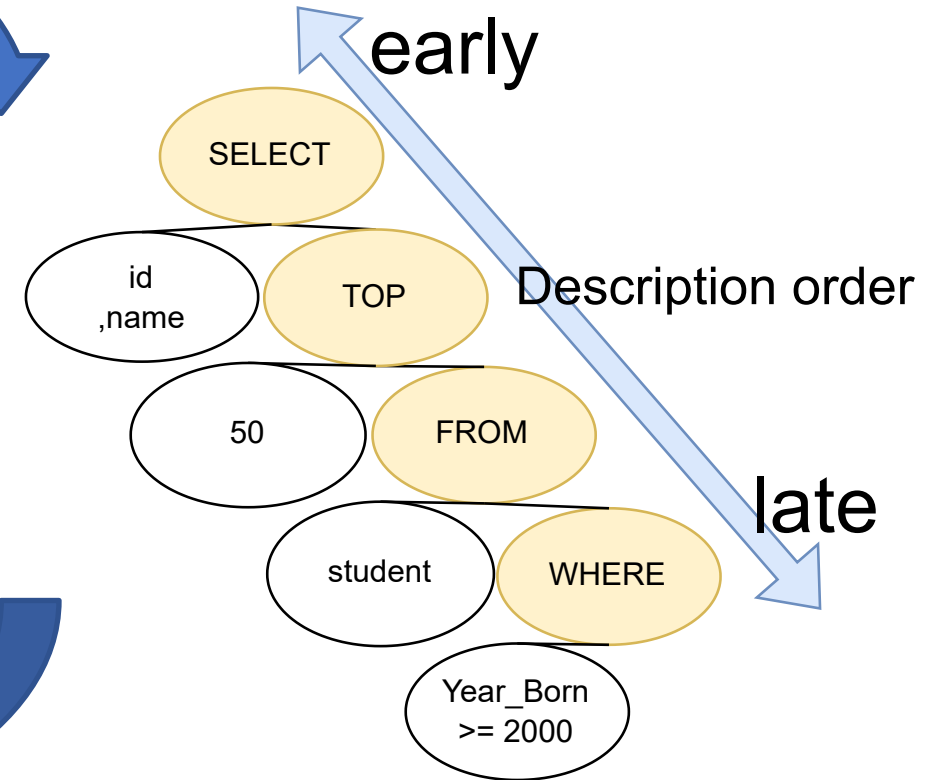
① SQL

```
1 SELECT ALL TOP 50  
2   id  
3   ,name  
4 FROM student  
5 WHERE Year_Born >= 2000
```

SQL Translator



② SQL Syntax Tree

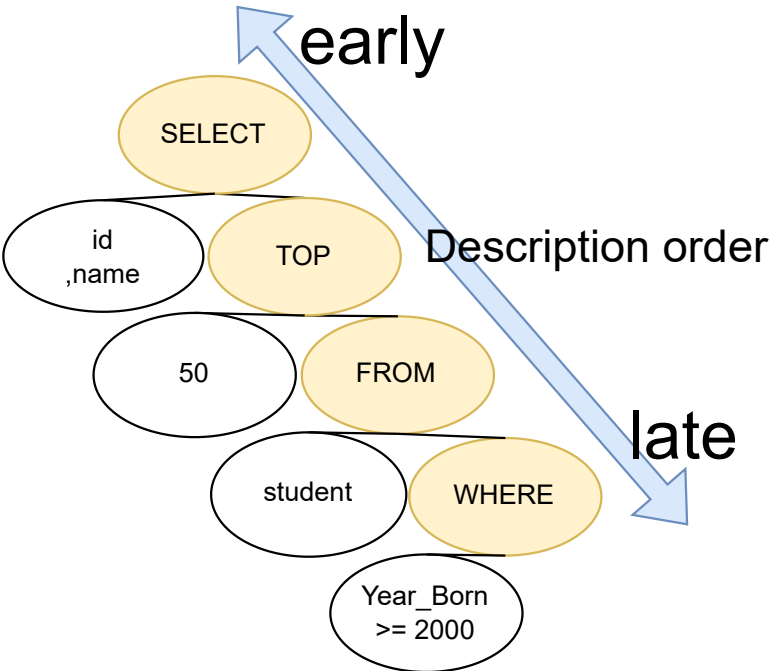


SQL Builder

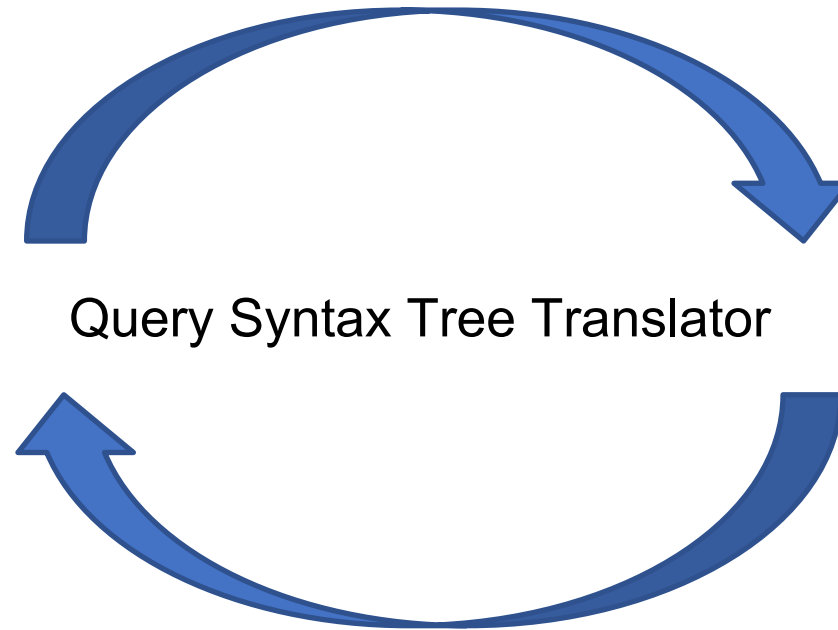
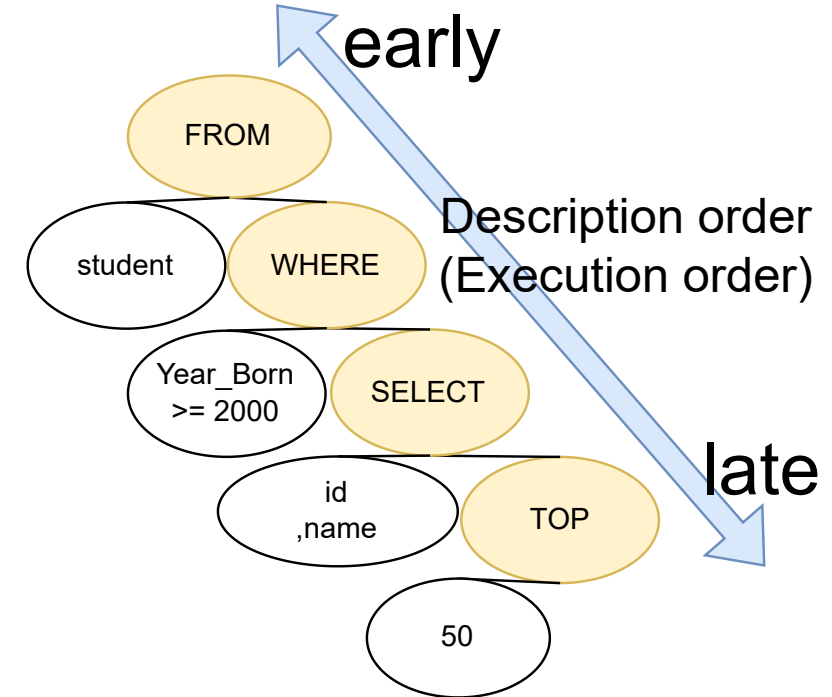


Translation Algorithm (Cont.)

② SQL Syntax Tree

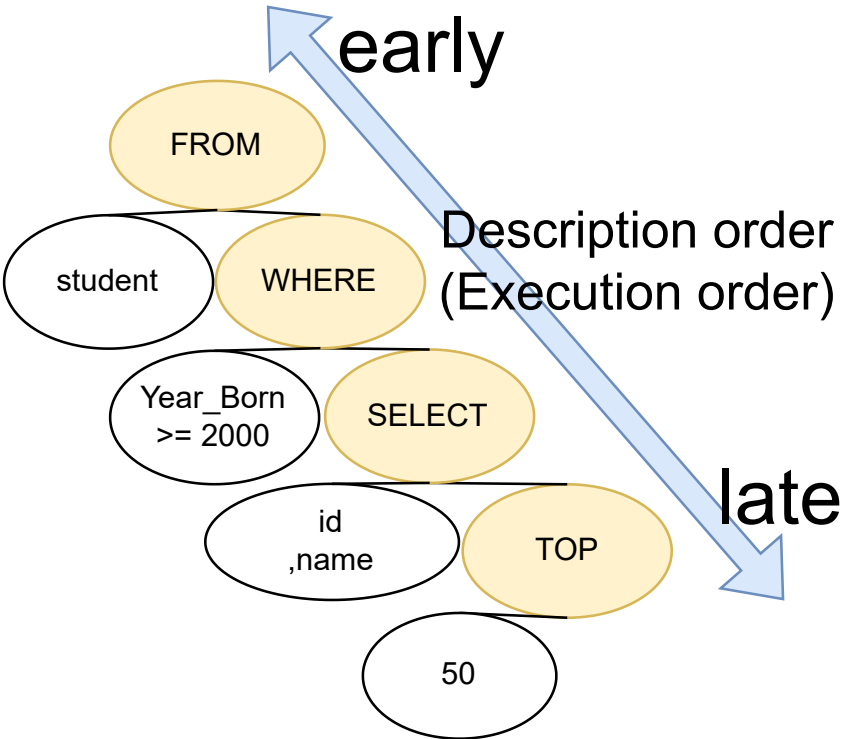


③ Procedural Query Syntax Tree

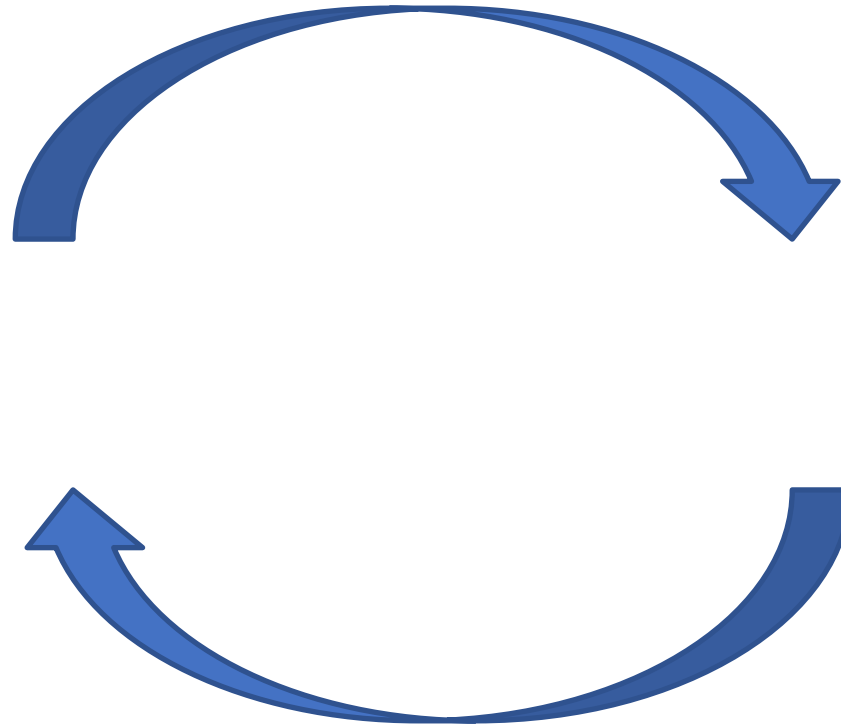


Translation Algorithm (Cont.)

③ Procedural Query Syntax Tree

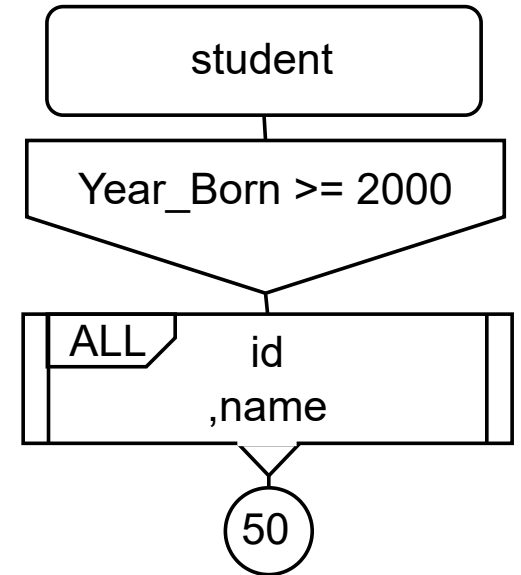


Procedural Visual Query Builder



Procedural Visual Query Translator

④ Procedural Visual Query



Algorithm Limitations

- SELECT INTO is not supported.
- TRUNCATE is not supported.
- The user needs to input SQL into the RDBMS to reference the execution plan.
- The user needs to input SQL into the RDBMS to check for syntax errors.

Conclusion

Procedural visual query features

- Query is sequential executable.
- Query can be written with icons.
- Subquery can be folded.
- UI covers most of the features of SQL-DML.

Future work

- Make end-user to see where bottlenecks are in the query by looking at the execution plan.
- We implement the interface in an online platform.

Thank you for your attention.