

ELS Checkpoint 3

The Preql Language

GROUP 3

Eduardo Luís Tronjo Ramos (up201906732@up.pt);
Fábio Araújo de Sá (up202007658@up.pt);
Pedro Pereira Ferreira (up202004986@up.pt);

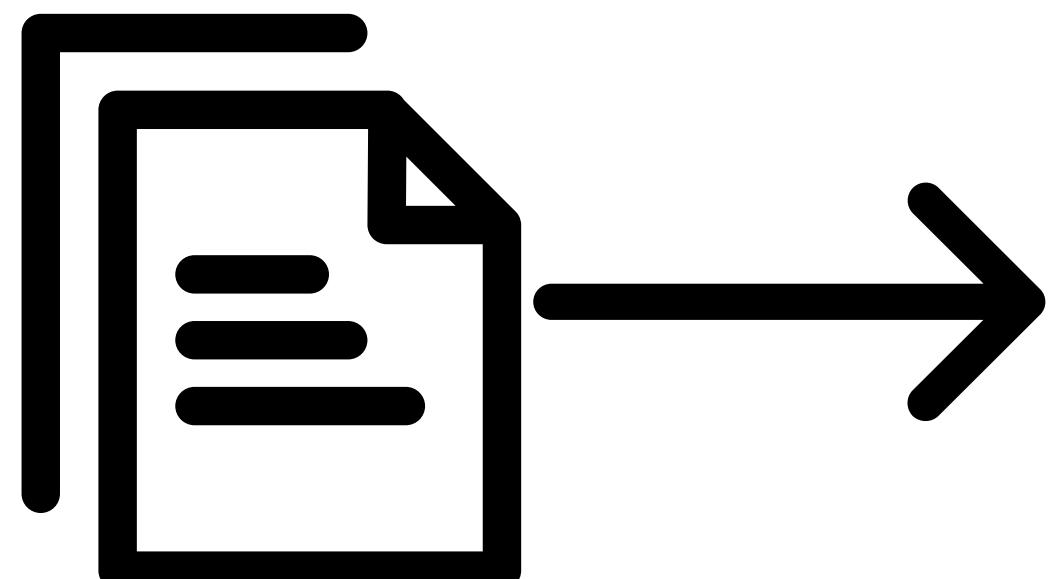
INTRODUCTION

- Last checkpoint we created an internal DSL, so that users could manipulate data from files without requiring any external file. Now, we created **Preql**, the external DSL build upon it.
- With external DSL, users now have a more specific and direct syntax to make their tasks.
- Users are also able to extract and process more data at the same time.

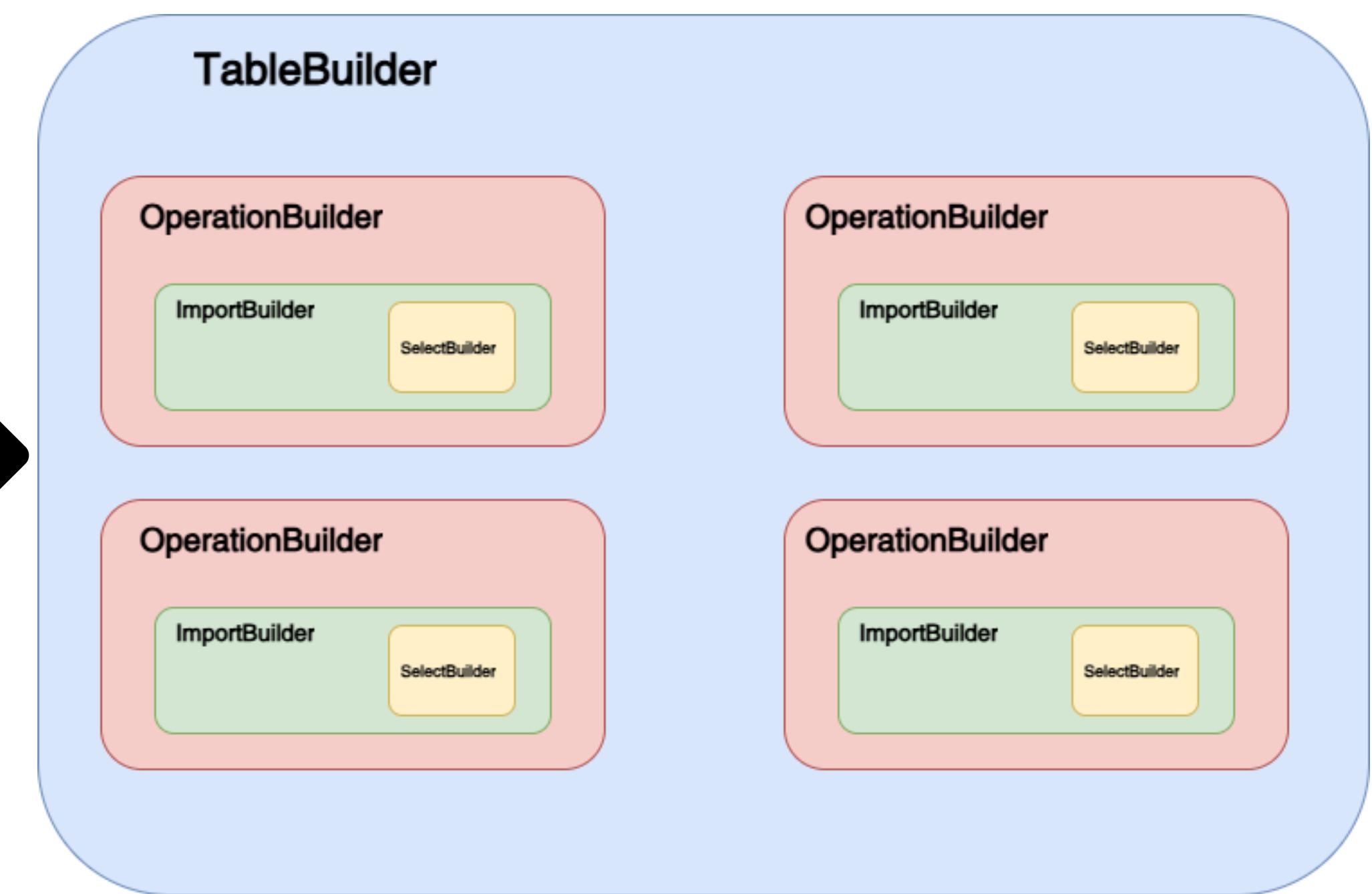


DSL ARCHITECTURE

External DSL



Internal DSL



PREQL FEATURES

- Multiple import types (import a file, a folder or folders):

```
• IMPORT DATA FROM FILE "resources/assignment_2/input/vitis-report.xml" {
    SELECT TABLE "/AreaEstimates/Resources";
};
```

```
• IMPORT DATA FROM FOLDER "resources/assignment_1/input/" ONLY YAML {
    SELECT COLUMN "params/criterion";
};
```

```
• IMPORT DATA FROM FOLDERS "resources/assignment_3/input/" {
    WITH EXTENSIONS {
        EXTENSION YAML {
            SELECT TABLE "/total/results/dynamic";
        };
    }
};
```

PREQL FEATURES

- Multiple tables extracted from different input file formats (JSON, YAML, XML):

```
IMPORT DATA FROM FOLDER "resources/assignment_1/input/" WITH EXTENSIONS {  
    EXTENSION YAML {  
        SELECT TABLE "/total/results/dynamic";  
    };  
    EXTENSION XML {  
        SELECT TABLE "/total/results/static";  
    };  
    EXTENSION JSON {  
        FILTER BY ("/functions/time%", MAX, ["name #1", "time% #1", "name #2", "time% #2", "name #3", "time% #3"], 3);  
        ADD COLUMN "Folder" AS "FOLDERNAME";  
        SUB "time% #1" AND "time% #2" TO "time% #4";  
    };  
};
```

```
IMPORT DATA FROM FOLDER "resources/assignment_1/input/" ONLY YAML {  
    SELECT COLUMN "params/criterion";  
};
```

PREQL FEATURES

- Different types of selecting (by table, by column, by filter):

```
6 IMPORT DATA FROM FOLDER "resources/assignment_1/input/" ONLY YAML {  
7     SELECT COLUMN "params/criterion";  
8 };
```

```
6 IMPORT DATA FROM FILE "resources/assignment_2/input/parameters.yaml" {  
7     SELECT TABLE "params";  
8 };
```

```
6 IMPORT DATA FROM FILE "resources/assignment_2/input/profiling.json" {  
7     FILTER BY ("/functions/time%", MAX, ["name", "time%"], 1);  
8 };
```

```
6  
7 IMPORT DATA FROM FILE "resources/assignment_2/input/decision_tree.yaml" {  
8     SELECT TABLE * BY NON-COMPOSITE;  
9 };
```

PREQL FEATURES

- Mathematical operations:

```
SUB "time% #1" AND "time% #2" TO "time% #4"; // (or SUM, MUL, DIV)
```

```
ADD *;  
AVERAGE *;
```

PREQL FEATURES

- Column operations:

```
PICKER BY ( "iterations", "calls", "time", [ "name", "time" ],  
    ADD COLUMN "Folder" AS "FOLDERNAME";  
    SUB "time% #1" AND "time% #2" TO "time% #4"; // (or SUM, MIN, MAX)  
};
```

```
ADD SUFFIX "(Dynamic)" TO "iterations";  
ADD SUFFIX "(Dynamic)" TO "calls";
```

```
ADD PREFIX "Mega" TO "iterations (Dynamic)";
```

```
RENAME COLUMN "Folder" TO "Pasta";
```

```
ADD GLOBAL COLUMN "Folder" AS "/input/";
```

PREQL FEATURES

- Merge of tables & Multiple table exportation in multiple formats (HTML, CSV, JSON, XML):

```
EXPORT TO "resources/assignment_3/output/output_external_test.html";
```

ERROR HANDLING

- Types of errors:
 - User incorrect input;
 - Internal errors (invalid operations);
- Exception layer, which throws the respective error:

```
• IMPORT DATA FROM FOLDERS "resources/assignment_4/input/" { // Inexistent path!
```

```
> Task :run FAILED
Exception in thread "main" pt.up.fe.els2024.exception.DSLEException: DSL Error: Invalid path: resources/assignment_4/input/
        at pt.up.fe.els2024.parser.DSLParser.parse(DSLParser.java:482)
        at pt.up.fe.els2024.Main.main(Main.java:30)
```

KNOWN ISSUES & LIMITATIONS

- Table sort, pivoting not supported;
- User is forced to learn the syntaxics of our external DSL;
- Some column operations missing (mean, median and mode);
- Using a path to select columns and tables can be cumbersome;
- Low number of possible constraints for the SelectorBuilder, such as filtering by number ranges, other types than composite and non-composite data (strings, numbers, etc...).
- Due to the new semantic model we can now do operations on only individual tables and columns, due to efficiency concerns.



USER PROFILE

- Now the user must know how **Preql**'s syntax works, which can be rapidly learnt.
- Having previous knowledge to **SQL** is a nice-to-have to accelerate on the learning process.
- The user can be compared to one who is not very familiar with programming languages but has some technical ability in manipulating data, such as someone familiar with **Excel**.





DEMO



CONCLUSIONS & FUTURE WORK

- As the internal DSL was already developed with the external architecture in mind, its transition to Preql was relatively smoothly.
- The DSL was designed to support all three use cases (Alice, Bob, Cal) effectively, it is feature-complete and exceeds the minimum requirements.
- If future developments were to occur, they would focus on adding functionalities such as table filtering, column sorting, grouping by attributes, column aggregation, and table joins, further extending the DSL's capabilities to address more complex use cases. However, no further enhancements are currently planned.