Enabling Exploratory Programming for Oracle MLE in Lively4

Jonas Grunert, 14.01.21

Programming Languages: Concepts, Tools, and Environments

Software Architecture Group, WS20/21

Agenda

1. Motivation
2. Related Work
3. Demo
4. Architecture & Implementation
5. Evaluation & Limitations
6. Future Work & Conclusion

Motivation

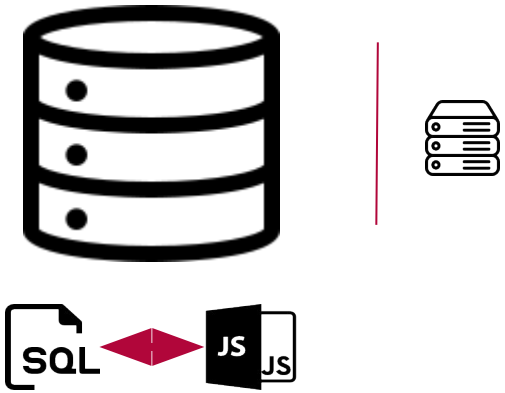
Large volume of data on relational database

Complex setup and knowledge of SQL

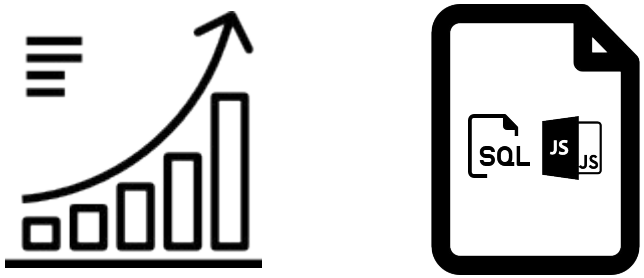
2 Distribution boundaries between data and visualization

Motivation

Executing JS code in the database for data/code locality



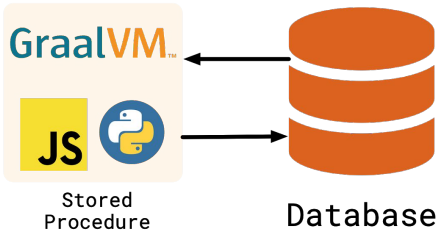
Defining query and visualization in one file in the browser



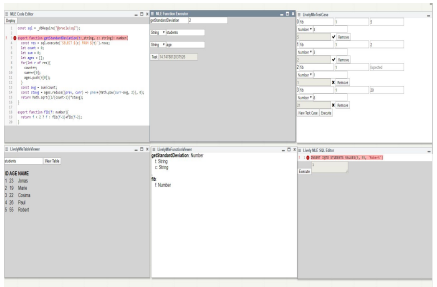
1,5 Distribution Boundaries

Related Work

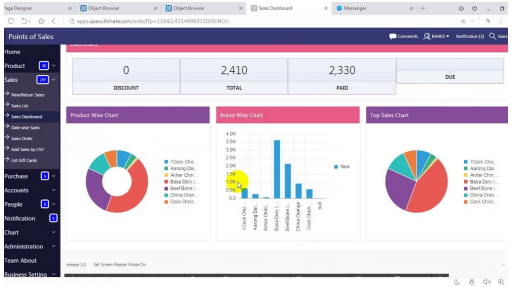
Oracle **M**ulti **L**anguage **E**ngine



Previous Seminar:
Seamless deployment of MLE Code

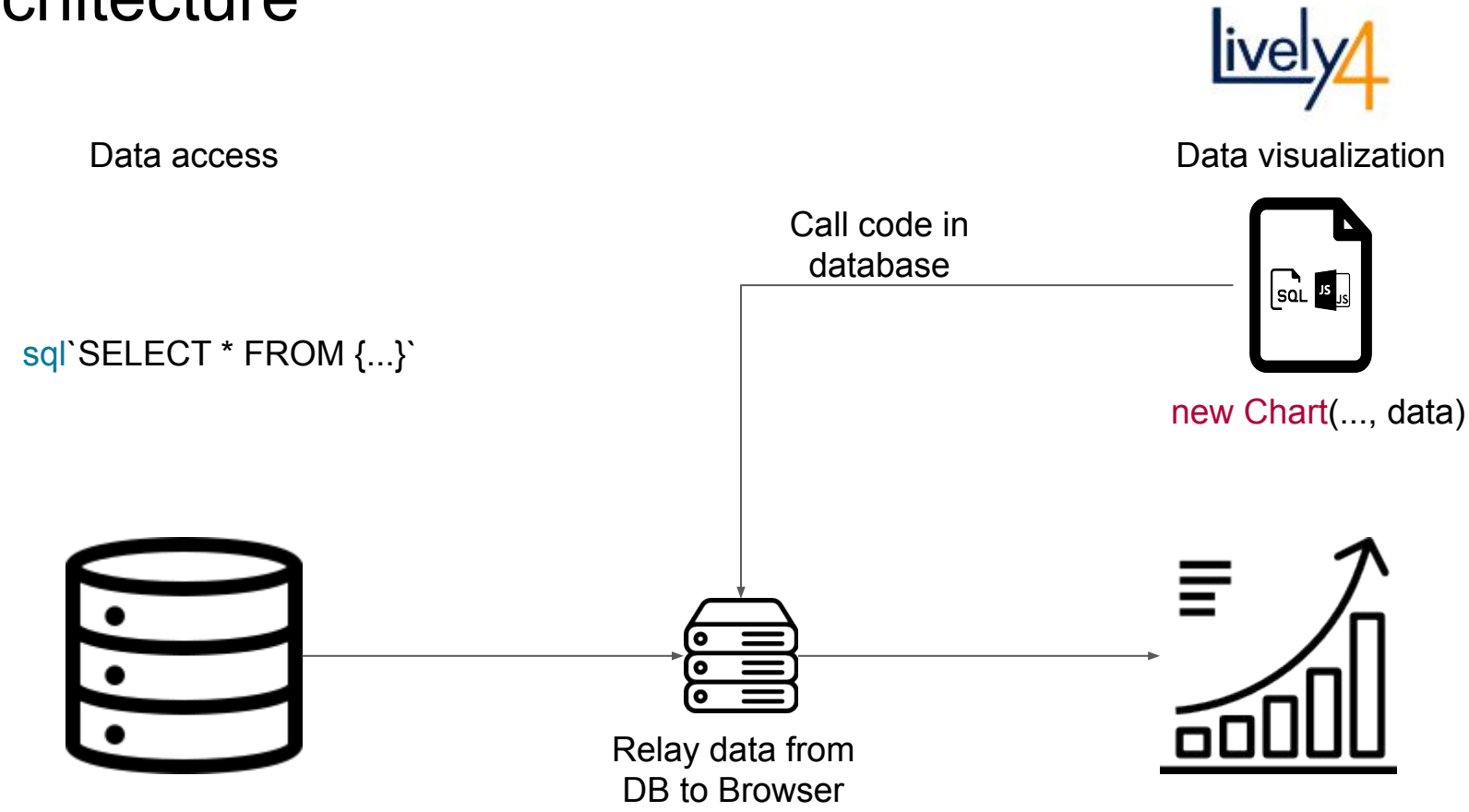


Oracle Apex



Demo

Architecture



Implementation details

Eval for dynamic code

sql`SELECT * FROM {...}`

In MLE:
eval(code)

Utility on global object

sql Tagged Template
SQL Helper Class

In MLE:
global.sql
global.SQL

Execution per element

map
reduce


```
let prev;
for(const element of result){
        prev = reduce(
                prev,
                map(element)
        );
}
return prev;
```

Pilot Benchmark

Example query:

Which GitHub projects takes the longest to merge a PR?

- 3 Tables ~ 48GB Data
- 16GB RAM
- Dockerized Database

MLE Setup: 15 Minutes / 3 GB RAM

Node Setup: 32 Minutes / 15 GB RAM

MLE Setup

Data Processing

Node Setup

Data Processing

Object Oriented Programming in DB

- Allowing for stateful behavior
- ORM-like programming within database
- Able to preemptively exit SQL and resume later on
- Persisting data over different executions without using a table
- Using advanced data structures (e.g. Sets, Maps etc)

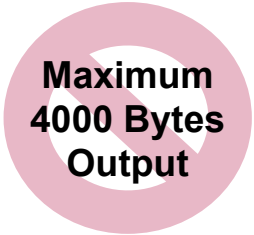
MLE Complexity Abstraction

1. Getting connection
2. Find out data types of table
3. Execute command handling binds
4. Iterate rows
5. Create Object
6. Return ResultSet

sql`.....`

Limitations

Border between
data and visualization
now within file



Mixed approach

Map/Reduce
Paradigm

Better SQL Knowledge

Faster execution time

STILL SLOW EXECUTION

Next Steps

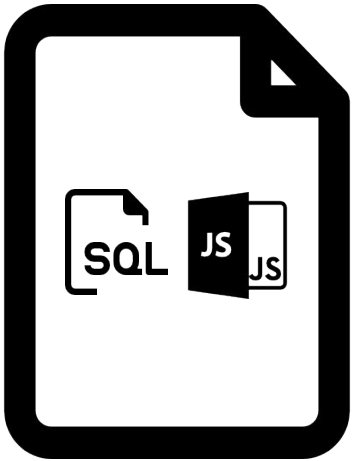
- Increase Outbound Data above 4000 Bytes
- Exploring incremental execution with intermediate results

Future Work

- Bindings support for sql tagged template
- Dynamic In- and Output

Conclusion

Defining query and visualization in **one** file
Executing JS code **within** database
No application server



lively4