

# **CS-562 Database Management Systems - II**

**Final Project - MF & EMF Query Compiler**

**Sushil Bhandary - 20015528  
Narmit Mashruwala - 20011284**

# Contents

- Introduction
- Project Architecture and Ownership
- Input Files and Query Structure
- Workflow
- Technologies
- Configuration
- Limitations
- Recap and Looking Forward

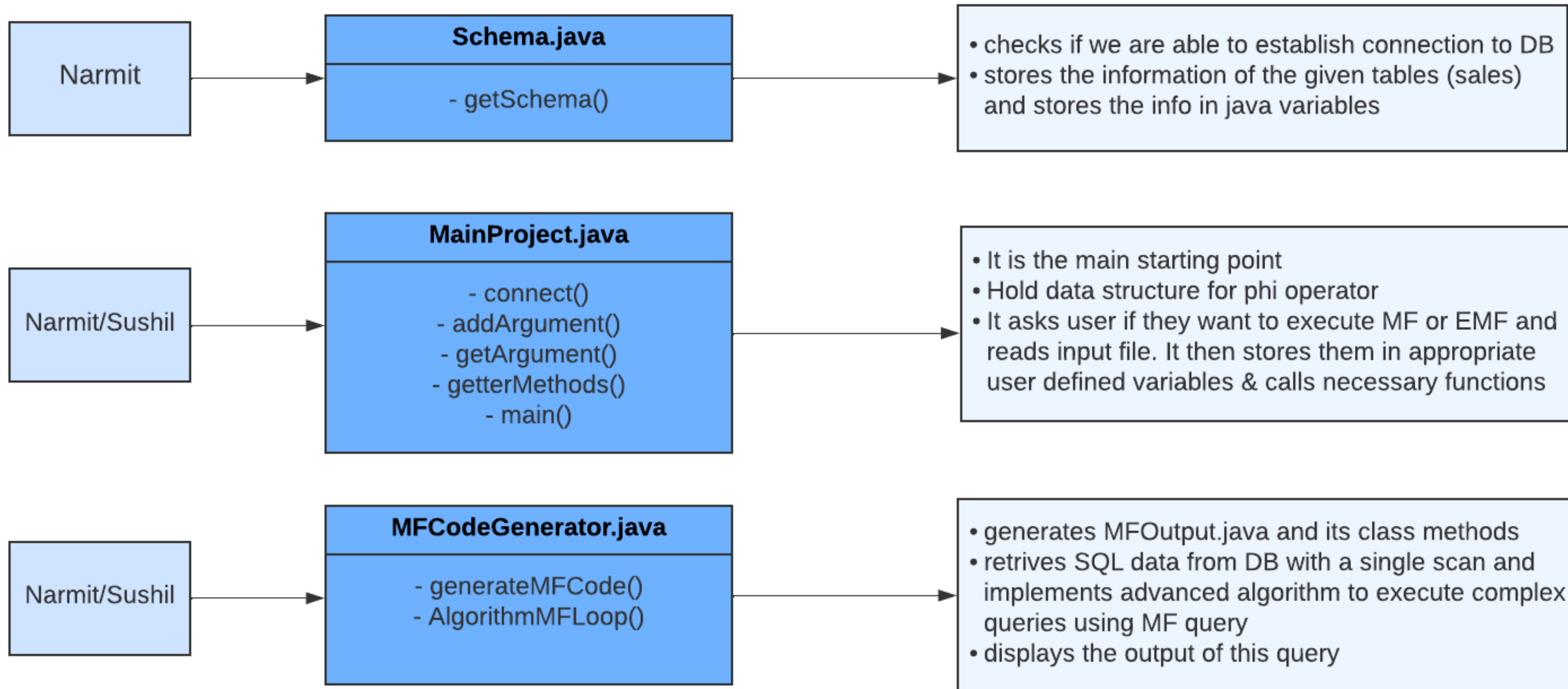
# Introduction

We are aiming to build a Query Processing Engine for Ad-Hoc OLAP queries by extending the group by statement and adding new such that clause

## Advantages

- Succinct
- Easy to maintain
- Easy to understand for non-technical person
- Potential to be more efficient
- Scalable
- Any MF/EMF query can be converted to 6 operands

# Project Architecture

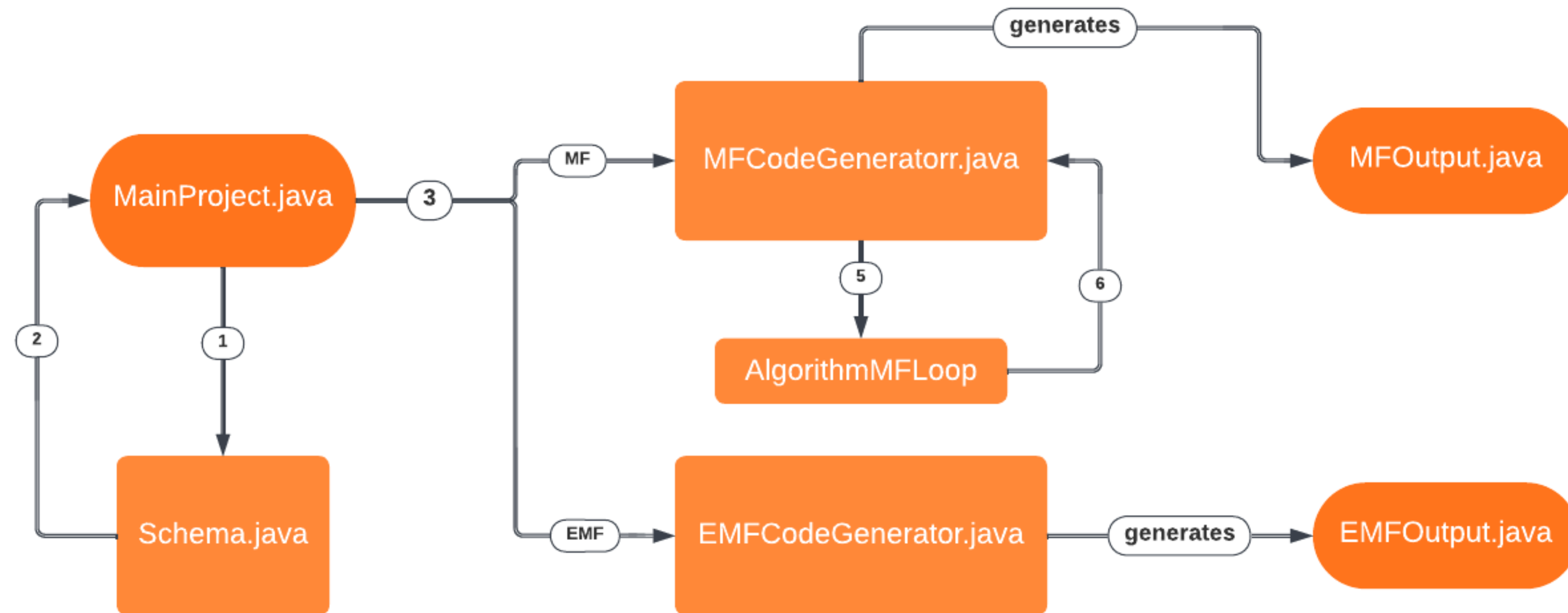


# Input Files and Query Structure

- Grouping Attributes (V)
- Number of Grouping variables (n)
- Projected Attributes (S)
- Aggregate functions (F-Vector) ([F])
- Such that Conditions ([C])
- Where clause
- Having Condition (G)

```
select_attribute:cust, 1_sum_quant,  
1_avg_quant, 2_sum_quant, 2_avg_quant,  
3_sum_quant, 3_avg_quant  
  
no_gv:3  
  
grouping_attributes:cust  
  
fvect:1_sum_quant, 1_avg_quant,  
2_sum_quant, 2_avg_quant, 3_sum_quant,  
3_avg_quant  
  
select:1_state="NY", 1_state="NJ",  
1_state="CT"  
  
where:  
  
having_condition:1_sum_quant > 2_sum_quant,  
1_avg_quant > 3_avg_quant
```

# Workflow



# Technologies

- PostgreSQL
- Java
- External drivers - PostgreSQL JDBC driver
- IDE - IntelliJ IDEA, pgAdmin4

## Configuration

Step 1 - Download the jar file from below url

URL: <http://jdbc.postgresql.org/download/postgresql-8.3-604.jdbc4.jar>

File: postgresql-8.3-604.jdbc4.jar

Version:8.3-604

Step 2 - Click on File -> Project Structure -> Modules -> Add jars or directories. Select the jar file & build.

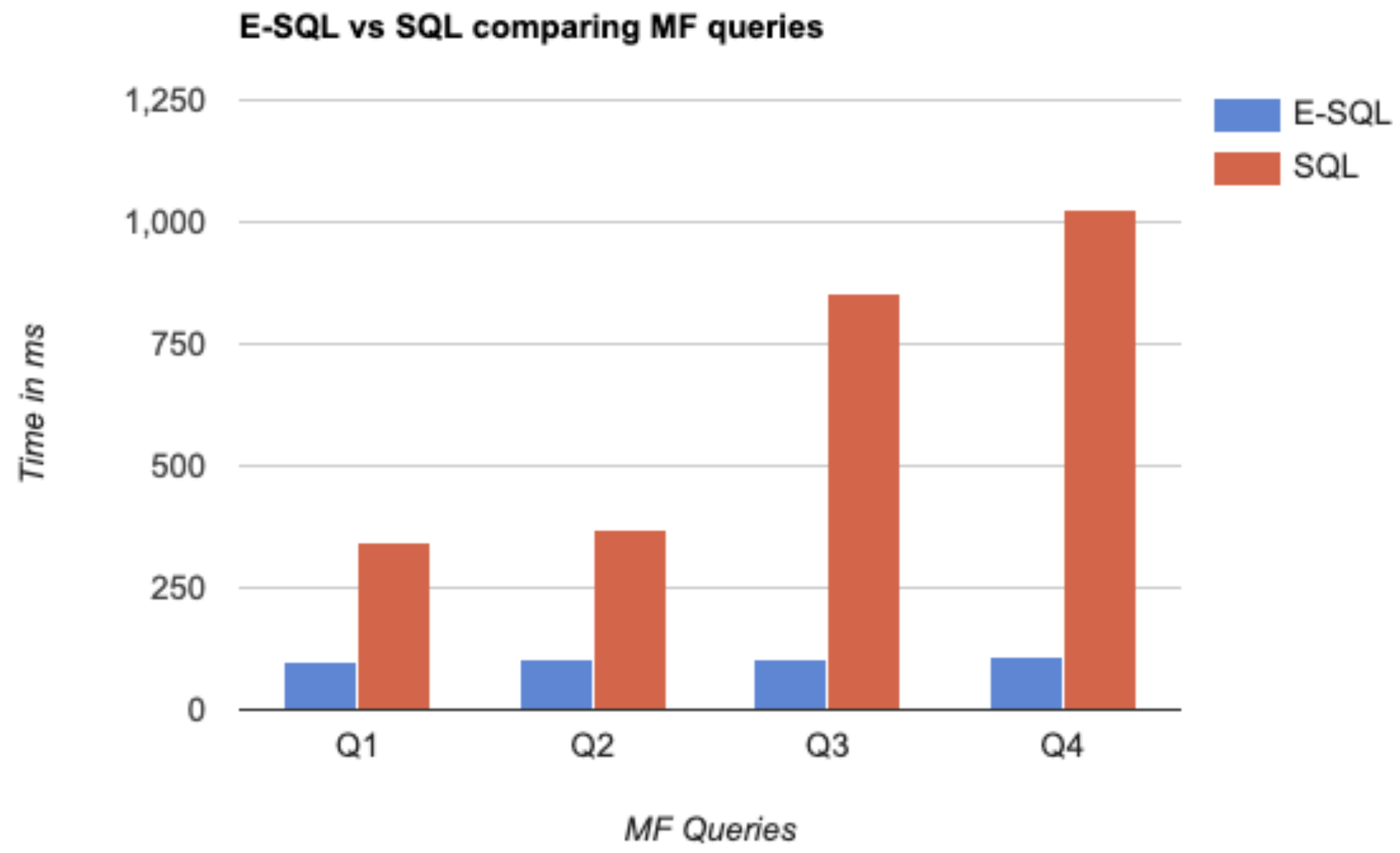


# Limitations

- Limited to phi operator
- The operands are read from a file instead of a UI. The files is structured to take operands in a certain format. Slightest deviation from this format can cause errors and/or incorrect output
- We are using the 6 operands instead of Extended-SQL
- The project scope is only limited to the sales table because it doesn't let the user to input which table to run the queries
- No error checking
- No support for minimal scan, optimization



# Why we love MF/EMF queries



Time taken in seconds by SQL query

```
101 SELECT V1.CUST, V1.PROD, V1.SUM AS SUM1, V2.SUM AS SUM2, V3.SUM AS SUM3, V1.COUNT AS COUNT1, V2.COUNT AS COUNT2, V3.COUNT AS COUNT3
102 FROM MFQ4_View1 AS V1, MFQ4_View2 AS V2, MFQ4_View3 AS V3
103 WHERE V1.CUST=V2.CUST AND V1.CUST = V3.CUST AND V1.PROD = V2.PROD AND V1.PROD = V3.PROD
```

	cust character varying (20)	prod character varying (20)	sum1 bigint	sum2 bigint	sum3 bigint	count1 bigint	count2 bigint
1	Helen	Bread	1637	4200	9050	1	2
2	Bloom	Milk	7150	9520	9352	3	4
3	Knuth	Cookies	1855	12689	10052	2	4
4	Sam	Fruits	10684	9847	6201	3	3
5	Emily	Fruits				6	2
6	Helen	Butter				5	9
7	Bloom	Eggs	3723	5531	9643	1	2
8	Emily	Eggs	6617	15380	2722	4	4
9	Emily	Pepsi	1682	2591	13070	1	1

Total rows: 39 of 39    Query complete: 00:00:01.027    Ln 102, Col 1

Time taken in milliseconds by E-SQL query

Knuth	Pepsi	4928	9578	7138	2	4	4
Bloom	Cookies	4362	8127	10563	2	2	4
Sam	Coke	3053	0	5491	1	0	2
Knuth	Cookies	1855	12689	10052	2	4	3
Sam	Butter	673	12846	0	1	4	0
Sam	Cookies	6198	0	2655	4	0	2
Knuth	Butter	3368	0	2232	2	0	1
Bloom	Bread	3100	10803	7648	1	5	5
Helen	Bread	1637	4200	9050	1	2	3
Bloom	Pepsi	2034	8776	94	1	4	1
Helen	Coke	3132	14523	18539	1	5	7
Bloom	Eggs	3723	5531	9643	1	2	4
Emily	Yogurt	0	2336			1	3
Emily	Coke	0	6807	7369	0	3	3
Bloom	Butter	0	3718	4279	0	1	1
Helen	Cookies	0	1965	2358	0	1	2
Time taken in milliseconds :		109					

# Recap

- Input from structured file
- Collect schema information & create MF struct
- Generate the output files
- Run the generated file to get outcome

# Looking Forward

- Can be extended to multiple tables
- Project architecture can be made dynamic
- Improve syntax parsing from file
- E-SQL queries as input instead of structured file
- Build optimal algorithm

**Thank you**