MF QUERIES

Hariom Nayani
Shaunak Kulkarni

MULTI FEATURE QUERIES

- Christmas came early for SQL developers. Ask why?
- Much like Santa clause we have the ‘Such That’ clause.
- SQL made easy.

ISSUES WITH STANDARD SQL

- Standard SQL does not allow succinct expression of queries, especially when grouping/aggregating on the same group of data
- Multiple joins over multiple tables needed in order to do simple operations
- Difficulty in understanding such lengthy queries
- Difficult for query optimizer to identify efficient algorithms

WHY MF QUERIES?

- Allows implementation of succinct queries
- Easier to use for SQL developers, don't have to wrap their head around merging multiple tables/views to get desired output
- Gives the query optimizer a relief by not having to deal with multiple join operations spanned across many tables

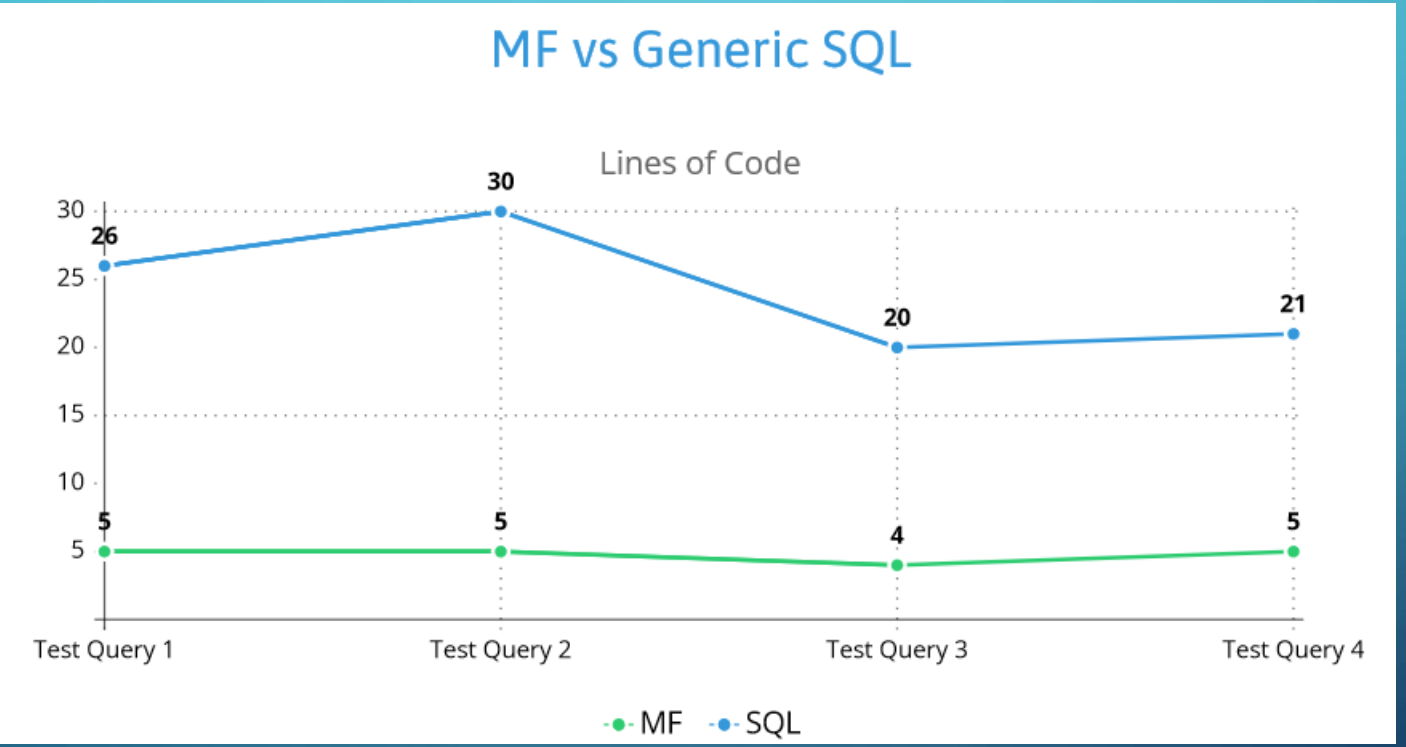
WHAT WE'LL COVER IN THE DEMO

- High Level Architecture of the running of our product
- Query structure – how we take inputs and it's format
- Technology used – which language and tools we used
- Running the project – Live demo of 4 queries and queries of your choice
- Comparison Graphs
- Go over the improvements to a few shortcomings of the project

QUERY STRUCTURE – JSON INPUT

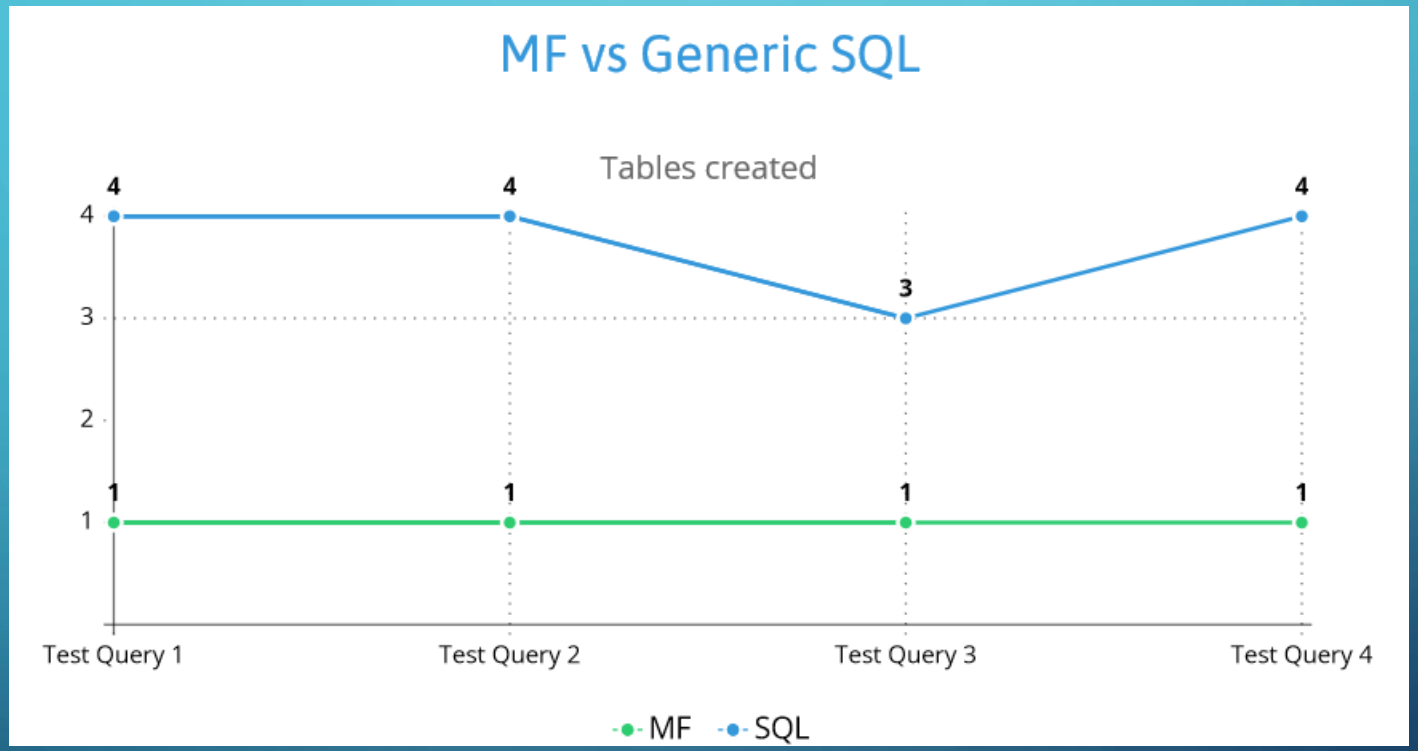
```json
{
  "SelectAttributes": [
    "cust",
    "prod",
    "sum_quant_1",
    "sum_quant_2",
    "sum_quant_3"
  ],
  "NumberOfGroupingVariables": 3,
  "GroupingAttributes": [
    "cust",
    "prod"
  ],
  "AggregateFunctions": [
    "sum_quant_1",
    "sum_quant_2",
    "sum_quant_3"
  ],
  "GroupingVariablePredicate": [
    "state_1 = 'NY'",
    "state_2 = 'NJ'",
    "state_3 = 'CT'"
  ],
  "HavingCondition": "sum_quant_1 > 2 * sum_quant_2 or avg_quant_1 > avg_quant_3"
}
```

RESULTING GRAPHS



(Avg Lines of code)

RESULTING GRAPHS

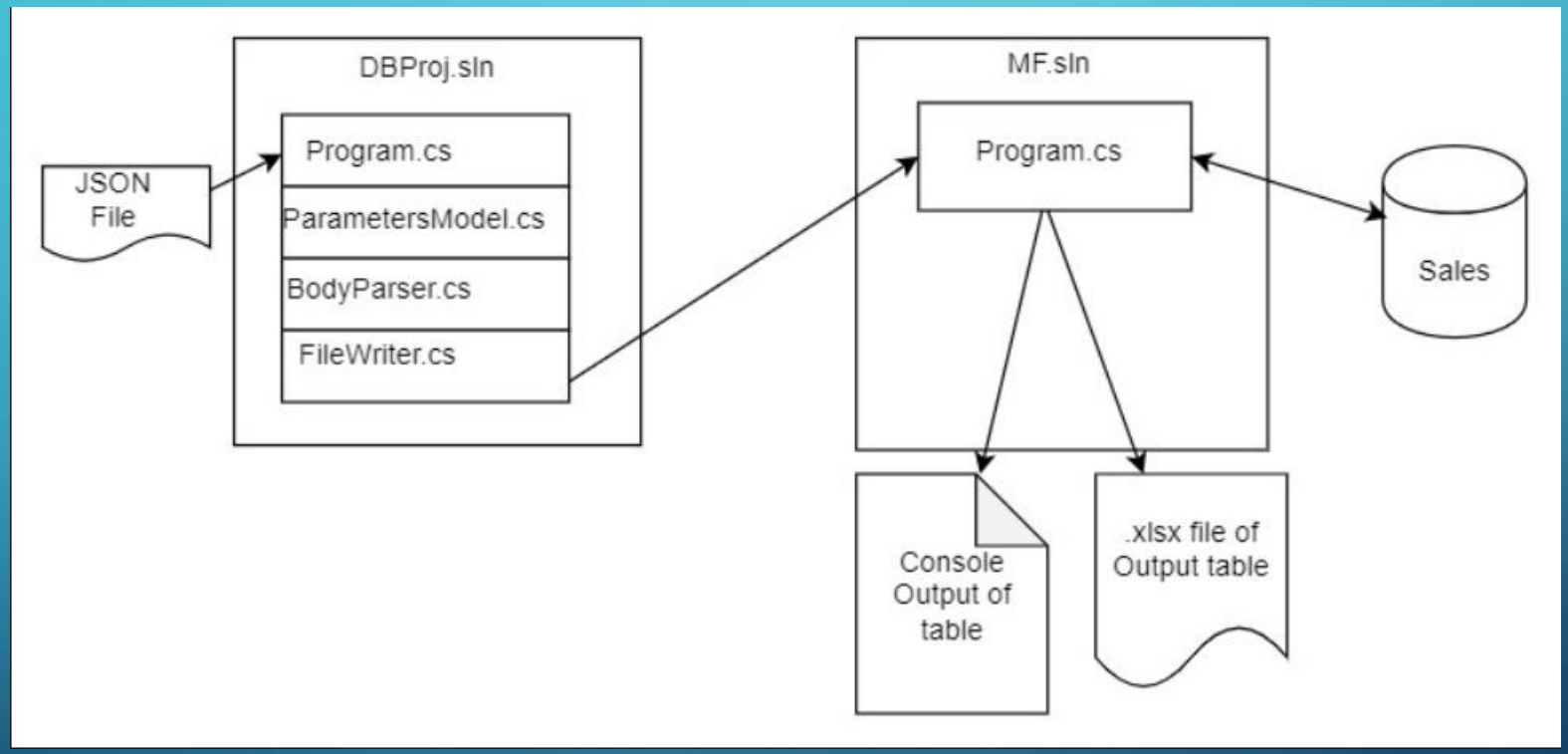


(Tables/Objects created)

TECHNOLOGIES , TOOLS AND EXTERNAL LIBRARIES

- Visual Studio IDE for .NET Development
- .NET Core 6.0 Framework
- PostgreSQL
- NpgSQL (External package for connecting PostgreSQL DB to .NET Application)
- ClosedXML (External package for .xlsx file creation and working on it)

HIGH LEVEL ARCHITECTURE



SCOPE OF THE PROJECT

- Can implement queries for all aggregate functions across any column
- Gives **optimized** output based on interdependencies of grouping variables
- Can perform aggregation on entire table and then consider those values in grouping variable predicates
- We are exporting results to .xlsx file to make it easily readable to human and ready to process
- Around **90%** of project is dynamic and with little touch ups, it can be implemented for any table
- Our code follows best practices of Object Oriented Programming
- Our classes are closed for modification but open for extension
- Provides a base structure to implement Extended Multi Feature Queries

SHORTCOMINGS

- Currently, code is designed to work with '*sales*' table only
- Although the framework we are using(.NET Core 6.0) is open source and platform independent, we need Visual Studio IDE to run our application
- Limited to a Windows file system as our code reads from and writes to Windows directories(eg. 'D:/')
- We assume that the JSON keys-values are always in the correct input format and the table does not have *null* values
- Code does not handle the '*where*' clause

HOW THEY CAN BE IMPROVED

- Error Handling and safety nets can be increased in our codebase
- Application detects device OS and handles file system operations dynamically
- Our application can be packaged and published to run on any platform
- The 10% code that runs statically, is related to '*sales*' , its attributes and their data types. NPGSQL gives us the functionality to read the metadata of the table, which can be leveraged to make this part dynamic.

TASK BREAKDOWN

- Both worked on :
- Writing final output file and reverse engineer it to write a file writer
- Optimization of algorithm
- Making the code

TASK BREAKDOWN

- Shaunak :
- High level architecture including selecting technologies, external libraries and flow of application
- JSON deserializer
- Npgsql connection and data table to model conversion using reflections
- Algorithm structure and operations
- Closedxml for writing excel exporter
- Writing scan on the whole table aggregates and include them in grouping variable predicates

TASK BREAKDOWN

- Hariom :
- JSON Input design
- JSON Input format
- Model creation for JSON Input
- Model creation for sales table
- Model creation of MF structure
- Writing mf structure and its constructor

THANK YOU