MQL

Daisy Wang, Yiqu Liu, Pitchapa Chantanapongvanij, Peihan Liu

About the language

What is MQL?

- Static and imperative table querying programming language
- Types: int, string, float, boolean, table
- Built-in functions: Import, Create (table), Insert (table), Print (table)
- Special table manipulation functions: Select, Where, Distinct, Delete,
- Can process data extracted from a user-provided source table (via Import table function)
- Easy to read with simplified SQL like syntax
- Additional functional programming features with C-like syntax
  (eg: TABLE B = Buildings.WHERE(name == "Hamilton"))

Evolution of MQL

Iteration 1

- Inspired by SQL
- Vision:
  - Basic table manipulation functions (select, where, join, extend, distinct)
  - Import table (csv files)
  - Other features: string concatenation, functions

Iteration 2

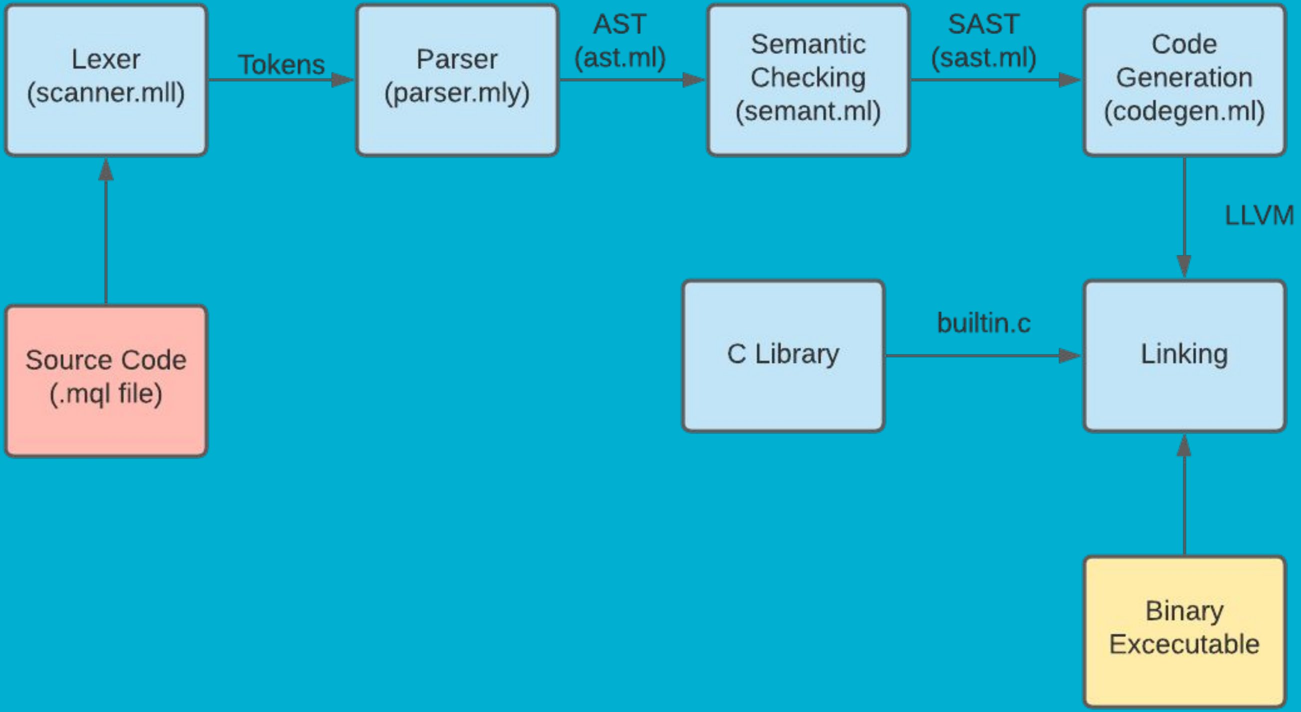
- Discarded functions
- Revised Vision:
  - Basic table manipulation functions + table creation and insertion (create, insert)
  - Discussion of linking codegen with C file (vs. previously tried to implement everything with Ocaml llvm module)

Current Iteration

- Discarded join and extend functions (time constraint)
- Current implementation:
  - Table data structure that can be manipulated with built in manipulation functions
    - Table represented as array of structs in C
  - Additional table feature: Create and insert table

Implementation

Compiler Architecture



MQL vs SQL

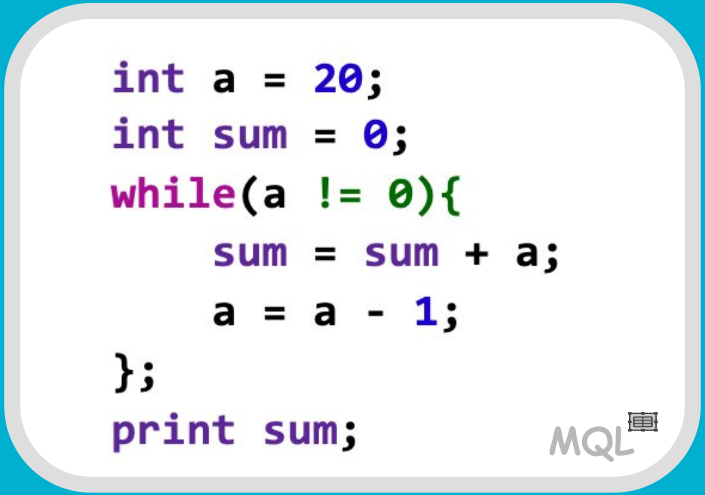
- MQL and MySQL are of different programming paradigms
- MQL is a minimalized version of MySQL with extended features as an imperative programming language
    - MQL extended features include:
      while loops, if-else statements, mathematical operations
- MQL supports the key features, and major built-in functions of MySQL
    - Eg: import, create, insert, select, where, distinct
- MQL has similar syntax to SQL

MQL

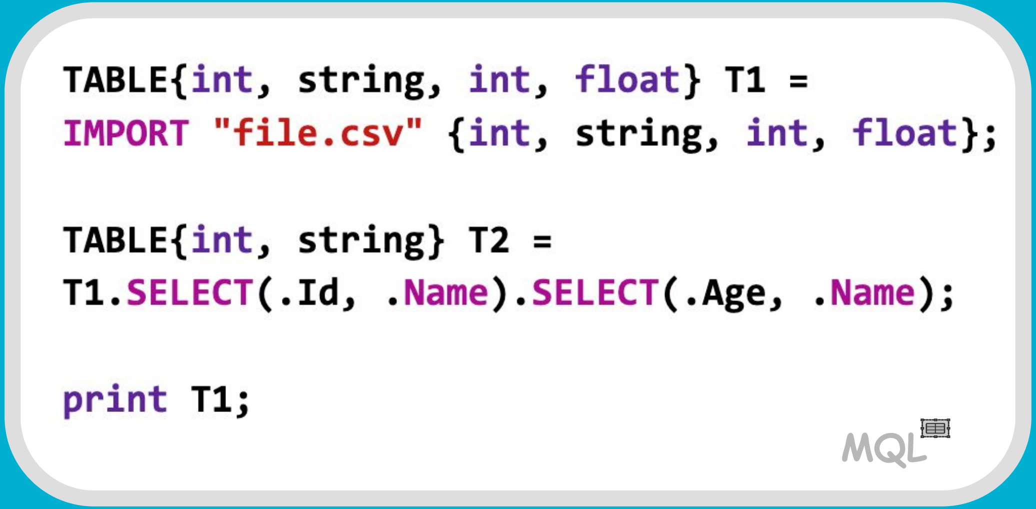
Key Language Features

Syntax & Grammar

- MQL follows imperative programming language style (step-by-step instructions)
- MQL supports *while loops, if else statements, table import, table manipulation*
- Below are some valid code segments written in MQL
- We will look into more details of MQL syntax & grammar in the following slides



```
int a = 20;
int sum = 0;
while(a != 0){
    sum = sum + a;
    a = a - 1;
};
print sum;
```



```
TABLE{int, string, int, float} T1 =
IMPORT "file.csv" {int, string, int, float};

TABLE{int, string} T2 =
T1.SELECT(.Id, .Name).SELECT(.Age, .Name);

print T1;
```

MQL Features

- *TABLE*
- *TABLE operators*
- *Comment*
- *Variable*
- *Expression Operation & Arithmetic*
- *Conditional Statement*
- *While Loop*

```
/*
TABLE is a major data structure provided
by MQL.
It is a collection of data, organized in
terms of rows and columns.
A TABLE can contain multiple columns
with combination of different types.
A set of table operations are provided
in MQL standard library, including
IMPORT, CREATE, SELECT, INSERT, WHERE,
DISTINCT, etc.
*/
```

Built-in Table Operators

/* Below is a complete list of TABLE operations supported and stored in standard library of MQL */

- ❑ IMPORT
- ❑ CREATE
- ❑ SELECT
- ❑ WHERE
- ❑ DELETE
- ❑ DISTINCT
- ❑ INSERT

```
/*import file.csv into TABLE T2*/
TABLE{int, string, int, float} T1 =
IMPORT "file.csv" {int, string, int, float};

/*create a table named new_table */
TABLE{int, string} New_table =
TABLE{int Id, string Name};

/*select column Id and Name from TABLE T2*/
TABLE{int, string} T2 =
T1.SELECT(.Id, .Name);

/*return rows with Id equals to 235 and remove
all duplicate*/
TABLE{int, string} T3 =
T2.WHERE(.Id == 235).DISTINCT(.Id, .Name);

/*delete rows from table T1 where has Name
equals to “Jenny”*/
TABLE{int, string} T4 =
T2.DELETE(.Name == “Jenny”);
```

MQL

Code Segment

MQL

Sample Program

```
float cutoffScore_2020 = 3.3;
float cutoffScore_2021 = 3.0;
int year = 2020;
boolean isUndergrade = true;
TABLE {int, string, int, string, boolean,
float} Students =
IMPORT "sample.csv" {int, string, int,
string, boolean, float} ;

TABLE {int, string, int, string, boolean,
float} ValidStudents =
Students.DELETE(.Id < 0);
Students = Students.INSERT(20, "Brian",
20, "Physics", false, 4.0);
```

```
TABLE{int, string} StudentInfo =
Students.DISTINCT(.Id, .Name);

if(year == 2020){
    StudentInfo = ValidStudents
       .WHERE(.Undergrad == isUndergrade)
       .DELETE(.Grade < cutoffScore_2020)
       .DISTINCT(.Id, .Name);
}else{
    StudentInfo = ValidStudents
       .WHERE(.Undergrad == isUndergrade)
       .DELETE(.Grade < cutoffScore_2021)
       .DISTINCT(.Id, .Name);
};
```

MQL

Demo Time

MQL

Testing

Test Suite

- Positive and negative test files: **test-***, **fail-*** for each language feature
- Positive test files output comparing with ***.out** files , negative test file has err log stored in ***.err** files
- Over 70 tests in MQL final repository

QUESTIONS?

Thank you!