

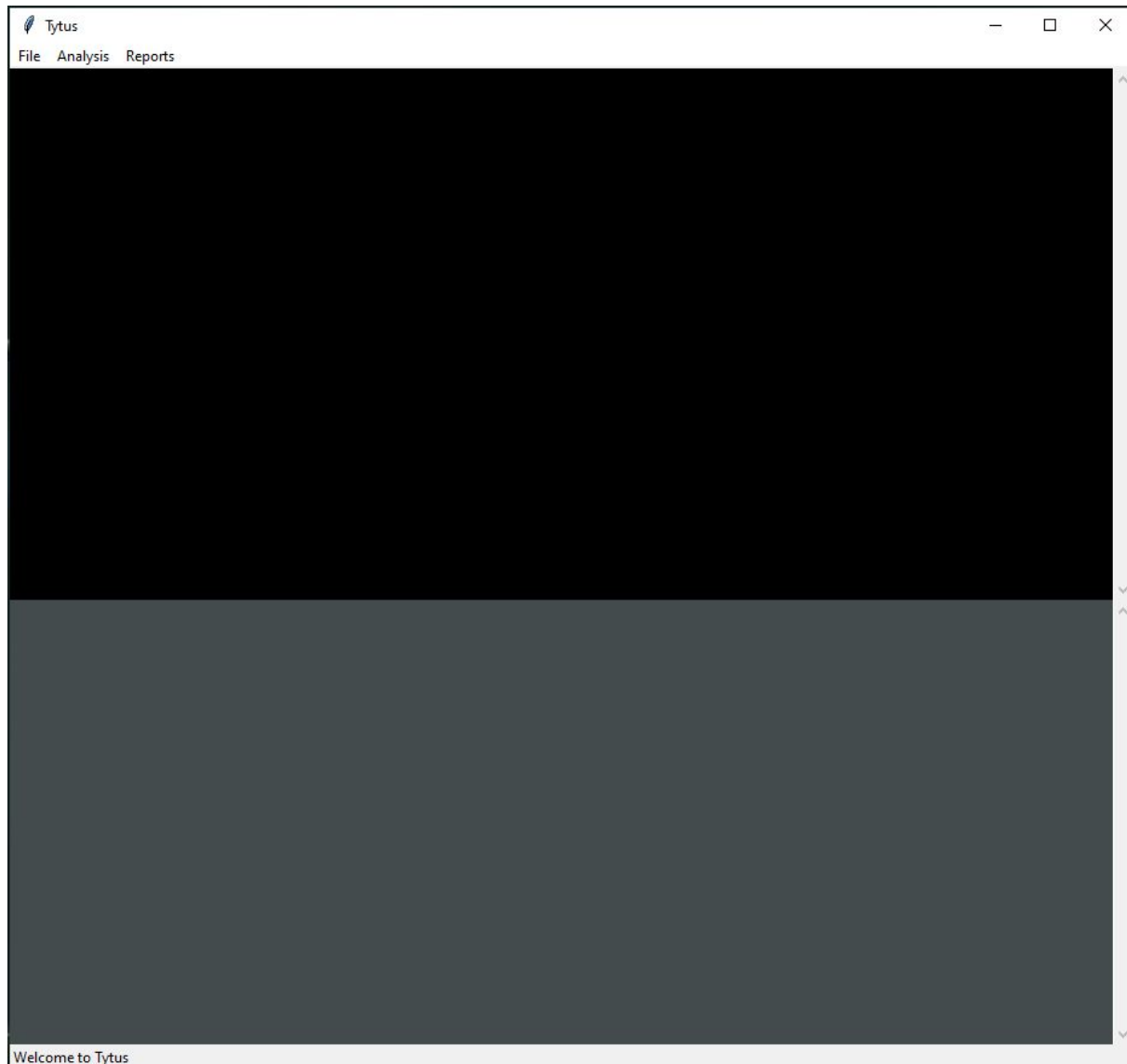
Universidad de San Carlos  
Facultad de Ingeniería  
Organización de Lenguajes y Compiladores 2  
Proyecto 1

## Manual de Usuario TytusDB

Diego Estuardo Gómez Fernández 201612141  
Jeralmy Alejandra de León Samayoa 201612139  
André Mendoza Torres 20161254  
Carlos Manuel García Escalante 201612276

TytusDB es el primer DBMS de código abierto escrito en Python, utiliza la sintaxis de PostgreSQL y permite el almacenamiento de información en distintas estructuras disponibles seleccionadas por el usuario por medio de un query.

El DBMS cuenta con un querytool que permite la escritura de lenguaje SQL, luego pasa por un parser para verificar que todo funcione bien y por último muestra los resultados en la consola que tiene integrada.



El querytool cuenta con las opciones nuevo, abrir, guardar y guardar como muy intuitivas y fáciles de utilizar:



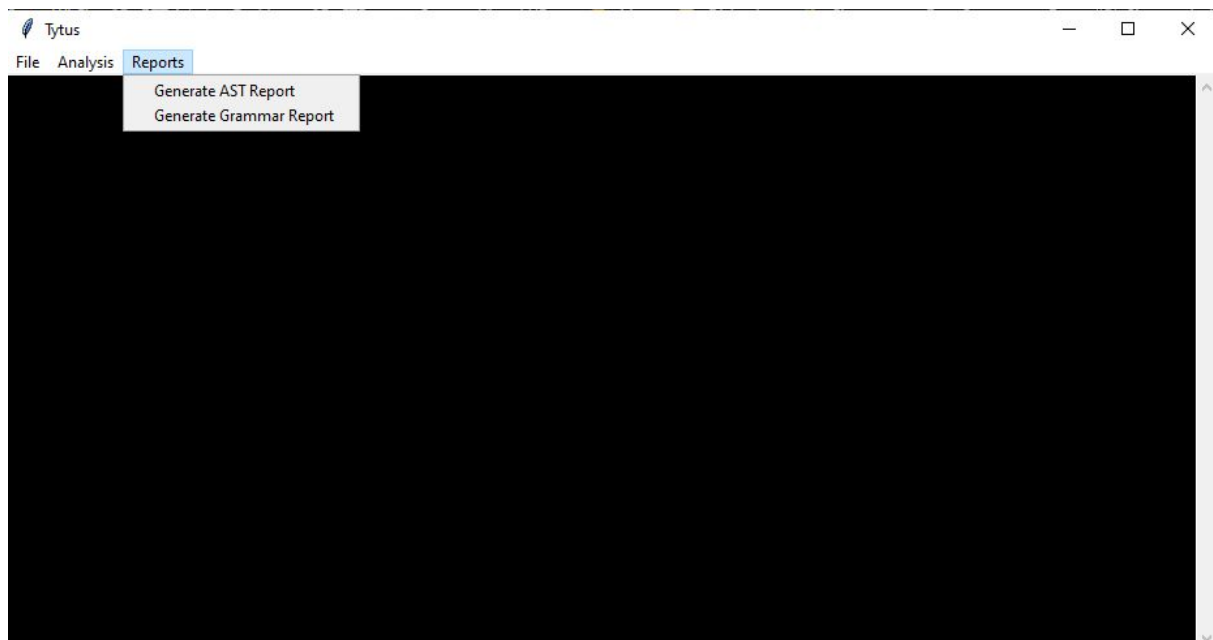
### Análisis:

Para realizar el análisis y ejecución del código SQL cuenta con la función compile.



## Reportes:

Además permite la generación de reportes de árbol de análisis sintáctico y de gramática que son distintos para cada entrada realizada en el querytool y muestran un poco de la estructura que utiliza el SQL Parser.



La consola integrada muestra mensajes de éxito y errores que se puedan tener, así como los resultados de las consultas realizadas.

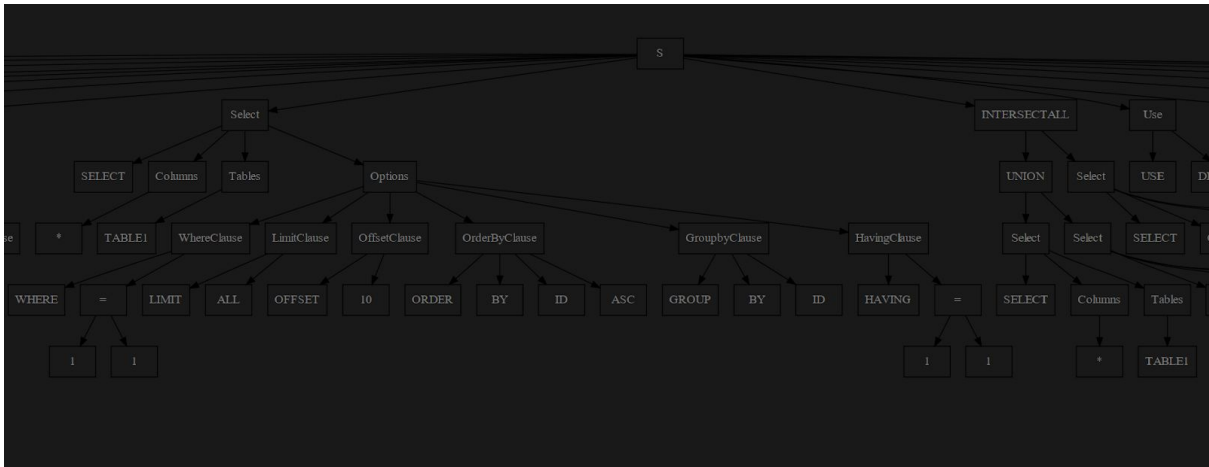
## Mensajes de texto

```
QUERY>> Database BASE1 has been created Query returned successfully
QUERY>> Database BASE1 has been selected
QUERY>> Table USUARIOS has been created
QUERY>> Table TERRENO has been created
QUERY>> Database DBPRUEBA has been selected
```

## Errores

```
ERROR TABLE
Error Semantic: La tabla TABLAPRUEB no existe en Fila 0 , Columna 0
Error Semantic: La tabla TABLAPRUEB no existe en Fila 0 , Columna 0
Error Semantic: El campo NOEXISTE no pertenece a la tabla TABLAPRUEBA en Fila 0 , Columna 0
```

Ejemplo de un reporte de AST



Ejemplo de un reporte gramatical.

# Reporte gramatical

## Terminales

**Palabras reservadas**

'CREATE, DROP, DATABASE, DATABASES, TABLE, SHOW, IF, EXISTS, ALTER, RENAME, OWNER, MODE, TO, COLUMN, CONSTRAINT, UNIQUE, FOREIGN, KEY, REFERENCES, REPLACE, SET, NOT, ADD, NULL, USE, INSERT, INTO, VALUES, TYPE, AS, ENUM, ASC, DESC, HAVING, GROUP, BY, OFFSET, LIMIT, ALL, ORDER, WHERE, SELECT, DISTINCT, FROM, UNION, EXCEPT, INTERSECT, BETWEEN, IN, LIKE, ILIKE, SIMILAR, SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE, PRECISION, MONEY, CHARACTER, VARYING, VARCHAR, TIMESTAMP, TEXT, CHAR, WITH, TIME, ZONE, WITHOUT, INTERVAL, BOOLEAN, DEFAULT, CHECK, PRIMARY, DATE, INHERITS, UPDATE, DELETE, TRUNCATE, ABS, CBRT, CEIL, CEILING, DEGREES, DIV, EXP, FACTORIAL, FLOOR, GCD, LN, LOG, MOD, PI, POWER, RADIANS, ROUND, AND, OR, COUNT, AVG, SUM, ACOS, ACOSD, ASIN, ASIND, ATAN, ATAND, ATAN2, ATAN2D, COS, COSD, COT, COTD, SIN, SIND, TAN, TAND, SINH, COSH, TANH, ASINH, ACOSH, ATANH'

**Simbolos**

; ( ) = + - \* / . ^ % < > <= >= <> !=

**ER**

ID = '[A-Za-z][A-Za-z0-9\_]\*'   NDECIMAL = '\d+\.\d+'   INT = '\d+'   STRING = '".\*?"

## Precedencia

```
precedence = (  
    ('left', 'UNION', 'INTERSECT', 'EXCEPT'),  
    ('left', 'OR'),  
    ('left', 'AND'),  
    ('right', 'NOT'),  
    ('left', 'LESSTHAN', 'GREATERTHAN', 'LESSTHANEQUAL', 'GREATERTHANEQUAL', 'NOTEQUAL'),  
    ('left', 'BETWEEN', 'IN', 'LIKE', 'ILIKE', 'SIMILAR'),  
    ('left', 'PLUS', 'MINUS'),  
    ('left', 'TIMES', 'DIVIDED', 'MODULO'),  
    ('left', 'EXPONENTIATION'),  
    ('right', 'UMINUS', 'UPLUS'),  
    ('left', 'NSEPARATOR'),  
)
```

## Gramatica

```
<start> ::= <sentences> { start.val = sentences.val }  
<sentences> ::= <sentences> <sentence> { sentences.val =  
    sentences.val.append(sentence.val) }  
<sentence> ::= <dml> ';' { sentence.val = dml.val }  
<dml> ::= <select> { dml.val = select.val }  
<select> ::= <selectInstruction> { select.val=selectInstruction }  
<selectInstruction> ::= SELECT <expressionList> FROM <expressionList> {
```