

# TytusDB

---

## Manual de Usuario OLC2 fase 2

**Universidad San Carlos  
de Guatemala**

**Grupo 17**

Pablo Rodrigo Barillas 201602503  
Christopher Jhoanis Soto 201602569  
Edgar Jonathan Arrecis 201602633  
Nery Eduardo Herrera 201602870

```
/home/jonathan/Documentos/2S2020VACAS/Compi2/tytus/pa
<InterpreteF2.Arbol.Arbol object at 0x7f45923efe80>
si llegoas aca
si llegoas aca
C3D generado:
Nombre: index1
Tipo: index
-----
Nombre: idx_califica
Tipo: index
-----
Nombre: index3
Tipo: index
-----
Nombre: idx_producto
Tipo: index
-----
Nombre: tbbodega_idx
Tipo: index
-----
Nombre: idx_califica
Tipo: index
-----
SIntactico realizado con exito

CREATE INDEX mytable_cat_1 ON mytable (data) WHERE category = 1;
CREATE INDEX test3_desc_index ON test3 (id DESC NULLS LAST, id D
ESC NULLS LAST);
CREATE UNIQUE INDEX idx_califica ON tbCalificacion (idcalifica);
CREATE UNIQUE INDEX index2 ON tbCalificacion (idcalifica);
CREATE UNIQUE INDEX index3 ON tbCalificacion (idcalifica);
CREATE UNIQUE INDEX index4 ON tbCalificacion (idcalifica);

drop INDEX      index4,test3_desc_index, index4;
drop INDEX if exists index2 cascade;

ALTER INDEX mytable_cat_1 RENAME TO index1;

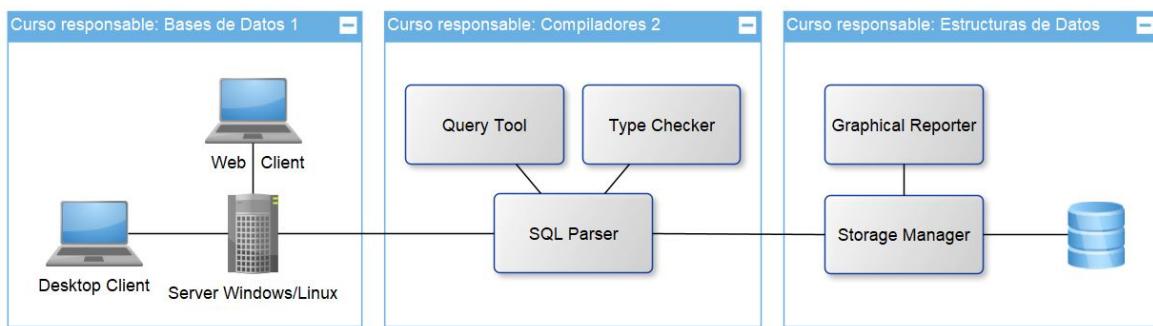
ALTER INDEX distributors SET (fillfactor = 75);
ALTER INDEX coord_idx ALTER COLUMN 3 SET STATISTICS 1000;
ALTER INDEX distributors RENAME TO suppliers;

CREATE UNIQUE INDEX idx_producto ON tbProducto (idproducto);
CREATE INDEX ON tbbodega ( ( lower(bodega) ) );
CREATE UNIQUE INDEX idx_califica ON tbCalificacion (idcalifica);
```

## Interfaz General

La interfaz cuenta con una entrada de texto donde el usuario podra colocar la sintaxis del codigo que desea compilar para luego traducir y posteriormente optimizarlo, y cuenta tambien con una terminal de salida donde podremos ir viendo los resultados del codigo traducido, cuenta tambien con dos botones uno de analizar que corre todo el codigo y otro que es analizar seccion que analiza la porcion de codigo que este seleccionado y el diseno cuenta tambien con dos menus desplegables para que podamos elegir reportes a generar y podamos ver el reporte generado.

## FLUJO DEL PROGRAMA



El proyecto esta dividido en varias etapas este manual se enfocara en el parser de dicho programa es decir la parte de compilacion, traduccion y optimizacion y de analisis del texto de entrada, a continuacion les explicaremos la sintaxis de dicho codigo.

## SINTAXIS DE CODIGO

El codigo se basa en la sintaxis de sql

### Numeric Types

Name	Description Range
Storage Size	
smallint 2 bytes	small-range integer -32768 to +32767
integer 4 bytes	typical choice for integer -2147483648 to +2147483647
bigint 8 bytes	large-range -9223372036854775808 to integer +9223372036854775807
decimal variable	user-specified precision, exact up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
numeric variable	user-specified precision, exact

	up to 131072 digits before the decimal point; up to 16383 digits
Name	Description Range
Storage Size	
	after the decimal point
real 4 bytes	variable-precision, 6 decimal digits precision inexact
double precision 8 bytes	variable-precision, 15 decimal digits precision inexact
money 8 bytes	currency -92233720368547758.08 to amount +92233720368547758.07

## character types

Name	Description
character varying(n), varchar(n)	variable-length with limit
character(n), char(n)	fixed-length, blank padded
text	variable unlimited length

## Comments

Para realizar un comentario utilizaremos la siguiente notacion

```
-- This is a standard SQL comment /* multiline comment
 * with nesting: /* nested block comment */
 */
```

---

## Creacion de tabla

Para crear una tabla se utilziara la siguiente sintaxis, colocandole el nombre deseado

```
CREATE TABLE my_first_table (
    column1 type [PRIMARY KEY]
    [, column2 type [REFERENCES table]] [, column3...]
);
```

## Eliminacion de tabla

Para eliminar una tabla se utilziara la siguiente sintaxis, colocandole el nombre de la tabla que se desea eliminar

```
DROP TABLE my_first_table;
```

## Alterar o modificar la tabla

Para agregar modificar y quitar campos de una tabla se utilziara la siguiente sintaxis, colocandole el nombre de la tabla que se desea eliminar y la accion deseada

```
ALTER TABLE table ADD COLUMN column type;
```

```
ALTER TABLE products DROP COLUMN description;
```

```
ALTER TABLE table DROP CONSTRAINT some_name;
```

## Manipulacion de datos

Para insertar una tupla de la tabla utilizaremos la siguiente sintaxis

```
INSERT INTO products VALUES (1, 'Cheese', 9.99);
```

Para modificar una tupla de la tabla utilizaremos la siguiente sintaxis

```
UPDATE products SET price = 10, costo = 9 WHERE price between 5 and 8;
```

Para eliminar una tupla de la tabla utilizaremos la siguiente sintaxis

```
DELETE FROM products WHERE price = 10;
```

## Trigonometric Functions

ACOS	<pre>SELECT ACOS(0) AS "Acos (0)";</pre>	La función se usa para devolver el coseno inverso de un argumento dado.	Únicamente en el Select
ACOSD	<pre>SELECT acosd(0.5);</pre>	La Función matemática que devuelve el coseno inverso de la expresión especificada, medido en grados.	Únicamente en el Select, update
ASIN	<pre>SELECT ASIN(0) AS "Asin (0)";</pre>	Se usa para devolver el seno inverso de un argumento dado.	Únicamente en el Select, update
ASIND	<pre>SELECT ASIN(0) AS "AsinD (0)";</pre>	Se usa para devolver el seno inverso de un argumento dado, especificada mediante grados	Únicamente en el Select
ATAN	<pre>SELECT ATAN(0) AS "Atan (0)";</pre>	Tangente inversa	Únicamente en el Select
ATAND	<pre>SELECT ATAND(0) AS "Atan (0)";</pre>	Trangente inversa especificada mediante grados	Únicamente en el select
ATAN2	<pre>SELECT ATAN2(0,1) AS "Atan2 (0,1)";</pre>	Se usa para devolver la tangente inversa de una división dada en el argumento	Únicamente en el select
atan2d	<pre>SELECT ATAN2D(0,1) AS "Atan2 (0,1)";</pre>	Se usa para devolver la tangente inversa de una división dada en el argumento, expresada en grados	Únicamente en el select
COS	<pre>SELECT COS(0) AS "Cos (0)";</pre>	Retorna el coseno de un argumento.	Únicamente en el select
COSD	<pre>SELECT COSD(0) AS "Cosd (0)";</pre>	Retorna el coseno de un argumento,	Únicamente en el select

---

Para realizar comparaciones en las condiciones utilizaremos los siguientes

LIKE

NOT LIKE

Como por ejemplo

```
string LIKE pattern [ESCAPE escape-character] string NOT LIKE pattern [ESCAPE  
escape-character] substring(string, pattern, escape-character)
```

## Subqueries

Esto quiere decir que podremos enlazar un query con otro en alguna parte del código como podemos ver en el ejemplo

```
FROM table_reference [AS] alias  
FROM (SELECT * FROM table1) AS alias_name
```

## Limit and offset

Estas condiciones nos permiten limitar el número de tuplas que devolverá la consulta

```
SELECT select_list  
FROM table_expression  
[ ORDER BY ... ]  
[ LIMIT { number | ALL } ] [ OFFSET number ]
```

## Index

```
CREATE INDEX test1_id_index ON test1 (id);
CREATE INDEX name ON table USING HASH (column);
CREATE INDEX test2_mm_idx ON test2 (major, minor);
CREATE INDEX test2_info_nulls_low ON test2 (info NULLS FIRST);
CREATE INDEX test3_desc_index ON test3 (id DESC NULLS LAST);
CREATE UNIQUE INDEX name ON table (column [, ...]);
CREATE INDEX test1_lower_col1_idx ON test1 (lower(col1));
CREATE INDEX access_log_client_ip_ix ON access_log (client_ip
    WHERE NOT (client_ip > inet '192.168.100.0' AND
    client_ip < inet '192.168.100.255'));

CREATE INDEX mytable_cat_1 ON mytable (data) WHERE category = 1;
CREATE INDEX mytable_cat_2 ON mytable (data) WHERE category = 2;
CREATE INDEX mytable_cat_3 ON mytable (data) WHERE category = 3;

CREATE INDEX name ON table (column opclass [ ( opclass_options ) ] [sort options] [, ...]);
```

## Raise

Para utilizar Raise se utiliza la siguiente estructura

```
raise notice 'Variable';
```

The screenshot shows a web-based database editor interface titled "Editor ML WEB". At the top, there are three colored dots (red, yellow, green) and a menu bar with buttons for "Analizador", "Seleccion", "Reporte", "Report Select", and "Genera 3d". The main area is divided into two panes. The left pane contains the SQL code:

```
raise notice 'hola';
```

```
The right pane displays the execution results:
```

```
t0 = 'hola'
print(str(t0))
```

## EJEMPLO DE UTILIZACION

SQL Tools

File Reportes

Analizador Seleccion Reporte Report Select Genera 3d

```
select Fun2('1', '2');

CREATE FUNCTION Fun2 (n integer, m integer)
declare ide1 integer = 'primer ide';
BEGIN
raise notice 'hola';
END;
```

```
t0 = Fun2('1','2')

def Fun2(n, m):
    t1 = n
    t2 = m
    t3 = 'primer ide'
    t4 = 'hola'
    print(str(t4))
```

## REPORTES

### REPORTES DE ERRORES

El siguiente reporte le mostrara al usuario los errores que tienen su código y donde encontrarlos, así este podrá ubicarlos y mejorarlos.

### ERRORES LEXICOS

The screenshot shows a Mac OS X desktop with a Safari browser window open. The title bar of the browser says "localhost". The main content area displays a table titled "Reporte de errores Lexicos OLC2- G17". The table has four columns: "#", "Descripción", "Fila", and "Origen". There is one row with the value 1 in the "#" column, "Caracter ilegal @" in "Descripción", 5 in "Fila", and "Lexico" in "Origen". The browser's address bar shows "https://curso: LABORA...". The Mac OS X dock is visible at the bottom, containing icons for various applications like Finder, Mail, and Safari.

#	Descripción	Fila	Origen
1	Caracter ilegal @	5	Lexico

## ERRORES SINTACTICOS

The screenshot shows a Mac OS X desktop with a Safari browser window open. The title bar of the browser says "localhost". The main content area displays a report titled "Reporte de errores Sintacticos OLC2- G17". The report table has columns for "#", "Descripción", "Fila", and "Origen". One row is shown: "# 1 Descripción Error Sintactico se esperaba otro caracter ; Fila 6 Origen Sintactico". The browser's address bar shows the URL "localhost". The system menu bar at the top indicates it's Wednesday at 18:29. The Dock at the bottom contains various application icons.

#	Descripción	Fila	Origen
1	Error Sintactico se esperaba otro caracter ;	6	Sintactico

## ERRORES SEMANTICOS

The screenshot shows a Mac OS X desktop with a browser window open, showing a semantic error report from Tytusdb. The title bar of the browser says "localhost". The main content area displays a report titled "Reporte de errores Semanticos OLC2- G17". The report table has columns for "#", "Descripción", and "Origen". One row is shown: "# 1 Descripción XX00: la base de datos no existe Origen Use database". The browser's address bar shows the URL "localhost". The system menu bar at the top indicates it's Wednesday at 18:28. The Dock at the bottom contains various application icons.

#	Descripción	Origen
1	XX00: la base de datos no existe	Use database

## REPORTE DE OPTIMIZACION

El siguiente reporte muestra las reglas utilizadas para la optimización del código intermedio.

## Reporte de Optimización OLC2-G17

No.	Regla Utilizada	Código Original	Código Optimizado	Fila	Columna
1	Regla 8	$t1 = t1 + 0$	Se elimina la instrucción	1	1
2	Regla 9	$t1 = t1 - 0$	Se elimina la instrucción	1	1
3	Regla 12	$t1 = t0 + 0$	$t1 = t0$	1	1
4	Regla 13	$t0 = t1 - 0$	$t0 = t1$	1	1

## REPORTE DE GRAMATICA

Dicho reporte desplegará al usuario la gramática utilizada en el proyecto.

ROOT -> SETinstrucciones

SETinstrucciones -> SETinstrucciones SETinstrucciones\_paso  
| SETinstrucciones\_paso

SETinstrucciones\_paso -> INSTRUCCION  
| INSTRUCCION ;

INSTRUCCION -> PRINICIPAL

| SELECT  
| UPDATE  
| ...

// -----

SELECT -> 'select' 'distinct' CUERPO\_SELECT  
| 'select' CUERPO\_SELECT

CUERPO\_SELECT -> grouping\_column\_reference 'from' table\_expression condition

condition -> WHERE\_condition condition

| GROUP BY grouping\_column\_reference condition  
| , grouping\_column\_reference condition  
| GROUP BY grouping\_column\_reference  
| , grouping\_column\_reference  
| WHERE\_condition

WHERE\_condition -> 'WHERE' search\_condition

grouping\_column\_reference -> \* , grouping\_column\_reference  
| Iden , grouping\_column\_reference  
| OBJETO , grouping\_column\_reference  
| AGREGACION , grouping\_column\_reference  
| \*  
| Iden  
| OBJETO  
| AGREGACION

IDen -> ID  
| ID as Iden  
| cadena

search\_condition -> search\_condition '=' search\_condition  
| search\_condition '!=' search\_condition  
| search\_condition '>' search\_condition  
| search\_condition '<' search\_condition  
| search\_condition '>=' search\_condition  
| search\_condition '<=' search\_condition  
| search\_condition 'or' search\_condition  
| search\_condition 'and' search\_condition

```
| search_condition 'like' search_condition  
| IDen  
| number  
| boolean  
| '(' search_condition ')'
```

// ----- UPDATE

UPDATE -> 'UPDATE' IDen set L\_search\_condition WHERE\_condition

L\_search\_condition -> search\_condition L\_search\_condition  
| ',' search\_condition L\_search\_condition  
| search\_condition

// ----- INSERT

INSERT -> 'insert' 'into' IDen '(' grouping\_column\_reference ')' INSERT CUERPO  
| 'insert' 'into' IDen INSERT CUERPO

INSERT CUERPO -> 'values' '(' grouping\_column\_reference ')'

// ---- DELETE

DELETE -> 'DELETE' 'from' IDen WHERE\_condition

// -----

## REPORTE DE ARBOL

Mostrara el árbol que se ira formando con forme la entrada del usuario sea recorrida y compilada.

