MANUAL TECNICO

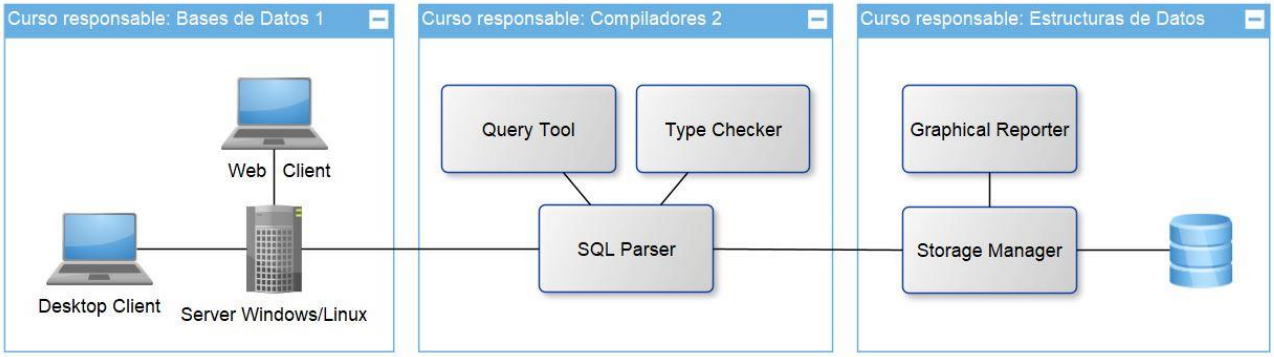
TytusDB-SQL Parser

Que es TytusDB yu

Es un proyecto Open Source para desarrollar un administrador de bases de datos. Está compuesto por tres componentes interrelacionados: el administrador de almacenamiento de la base de datos, que estará a cargo del curso de Estructuras de Datos; el administrador de la base de datos, que estará a cargo del curso de Sistemas de Bases de Datos 1, este administrador se compone a su vez de un servidor y de un cliente; y el SQL Parser, que estará a cargo del curso de Organización de Lenguajes y Compiladores 2.

TytusDB – SQl Parser

La siguiente figura muestra la integración entre los tres cursos desarrollando Tytus. Este manual está dedicado solo al TytusDB-SQL Parser



Licencias Adicionales utilizadas:

Tabulate: Licencia MIT

Enumerable : Licencia MIT

Tkinter: Licencia MIT

Graphiz: Licencia MIT

Listado de Clases utilizadas en el proyecto con una breve descripción de cada uno:

- Principal
- Analizador_lexico:
- Analizador_Sintáctico: calcularDelete
- calcularInsertCampos
- calcularInsertValores
- calcularSelectCampos
- calcularSelectGrupos
- calcularSelectWhere
- claseArbol
- ejecución
- error
- expresiones
- graficarAST
- instrucciones
- listarAST
- Nodo
- recorredorAst
- parsetab
- SelectCampos
- SelectRecorrido
- Sentencias
- Ts

Durante la ejecución se genera el reporte del AST en un archivo PDF, utilizando la librería Graphiz.

SINTAXIS SQL permitida:

Definición de las instrucciones a utilizar:

Para manejo de Bases de Datos:

```
CREATE [OR REPLACE] DATABASE [IF NOT EXISTS] name
    [ OWNER [=] user_name ]
    [ MODE [=] mode_number ]


SHOW DATABASES [LIKE regex]

ALTER DATABASE name RENAME TO new_name

ALTER DATABASE name OWNER TO { new_owner | CURRENT_USER | SESSION_USER }

DROP DATABASE [ IF EXISTS ] name

USE databasename
```

Manipulación de Tablas:

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

DROP TABLE my_first_table;

ALTER TABLE table ADD COLUMN column type;

DELETE FROM [ ONLY ] table_name [ * ] [ [ AS ] alias ]
    [ USING from_item [, ...] ]
    [ WHERE condition | WHERE CURRENT OF cursor_name ]
    [ RETURNING * | output_expression [ [ AS ] output_name ] [, ...] ]
```

Manipulación de Datos:

```
INSERT INTO [table] VALUES (valor1, valor2, valor3);

UPDATE table SET expresion WHERE expresion;
```

```
DELETE FROM table WHERE expresion;
```

Estructura de los queries

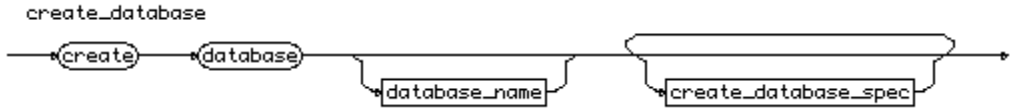
```
SELECT [DISTINCT] select_list FROM table_expression
[WHERE search_condition]
[GROUP BY grouping_column_reference [, grouping_column_reference]...]
```

Tipos permitidos:

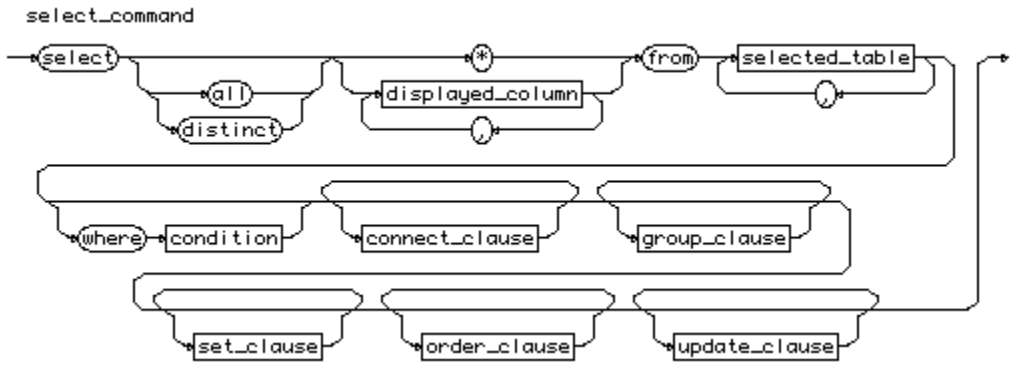
- Numéricos
- Carácter
- Fecha/Hora
- Booleano

DIAGRAMAS DE REFERENCIA PARA LA CREACION DE SENTENCIAS[1]

Crear base de datos:

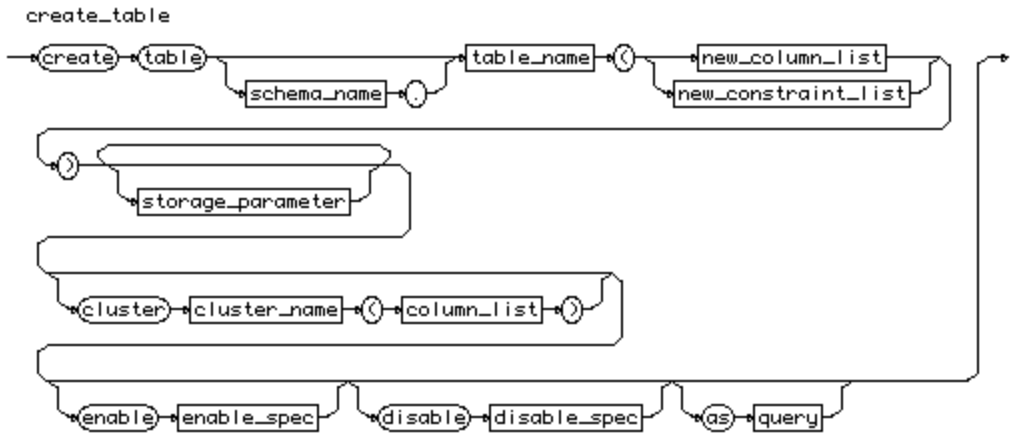


Instrucción SELECT



[1] http://cui.unige.ch/isi/bnf/SQL7/

Crear Tabla



Base de la gramática generada en el proyecto:

<S> ::= <Init>

<Init> ::= <Statement_list>

<Stament_list> ::= <Statement_List> <Statement>

<Statement_list> ::= <Statement>

<Statement> ::= <Insert_statement>

| <Update_statement>

| <Delete_statement>

| <Enumtype>

<Enumtype> ::= CREATE TYPE <un_idx> AS ENUM PARIZQ <list_enum> PARDER PTCOMA

| <Un_idx>

<list_enum> ::= <list_enum> COMA <otro_id>

| <otro_id>

<otro_id> ::= CADENACOMILLASIMPLE

<un_idx> ::= ID

<insert_statement> ::= INSERT INTO <table_name> <insert_columns_and_source> PTCOMA

<table_name> ::= ID

<insert_columns_and_source> ::= <PARIZQ insert_column_list> PARDER VALUES <query_expression_insert>

| VALUES <query_expression_insert>

| <insert_default_values>

<insert_default_values> ::= DEFAULT VALUES

<insert_column_list ::= <insert_column_list> COMA <column_name>

| <column_name>

<column_name> ::= ID

<query_expression_inset> ::= <insert_list>

<insert_list> ::= <insert_list> COMA <insert_value_list>

| <insert_value_list>

<insert_value_list> ::= PARIZQ <value_list> PARDER

<value_list> ::= <value_list> COMA <insert_value>

| <insert_value>

<insert_value> ::= ENTERO

| DECIMAL

| CADENACOMSIMPLE

| DEFAULT

| NOW PARIZQ PARDER

<update_statement ::= UPDATE <table_name> SET <set_clause_list> WHERE <search_condition> PTCOMA

<update_statement ::=UPDATE <table_name> SET <set_clause_list> PTCOMA

<set_clause_list> ::= <set_clause_list> COMA <set_clause>

| <set_clause>

<set_clause> ::= <column_name> IGUAL <update_source>

<update_source> ::= <value_expression>

| NULL

<delete_statement> :=DELETE <table_name_decimal> FROM <table_name_d> PTCOMA

<delete_statement> := DELETE <table_name_d> PARDER <tname_ent> FROM <table_name_d> WHERE <search_condition> PTCOMA

<tname_ent> ::= ENTERO

<tname_ent> ::= VARCHAR

<table_name_decimal> ::= DECIMAL

<table_name_d> ::= ID

<search_condition> ::= <search_condition> OR <boolean_term>

| <boolean_term>

<boolean_term> ::= <boolean_term> AND <boolean_factor>

| <boolean_factor>

<boolean_factor> ::= NOT <boolean_test>

| <boolean_test>

<boolean_test> ::= <boolean_primary>

<boolean_primary> ::= PARIZQ <search_condition> PARDER

<value_expression> ::= ENTERO

| DECIMAL

| CADENACOMSIMPLE

| DEFAULT

| ID

| NOW PARIZQ PARDER