

HENRY

A bright yellow beam of light originates from the left edge of the frame and points towards the letter 'R' in the word 'HENRY'. The beam is wider on the left and tapers as it moves to the right. The word 'HENRY' is written in a bold, black, sans-serif font.



SQL



SQL



SQL

(Structured Query Language) es el lenguaje más utilizado para hacer consultas a una Base de Datos.



SQL

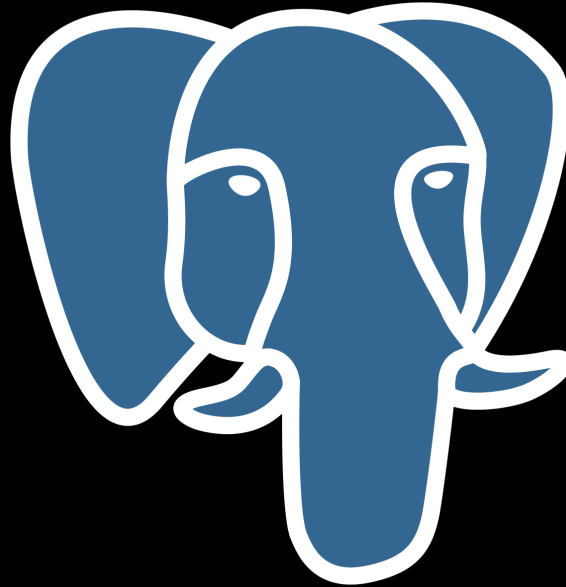
```
UPDATE clause {UPDATE country  
               SET clause {SET population = expression population + 1  
               WHERE clause {WHERE name = expression 'USA' ;  
                               predicate } } statement
```

SQL es un lenguaje de consultas, tiene expresiones y statements igual que JS.

Está compuesto por Cláusulas.



PostgreSQL



Existe muchos motores de SQL. Algunos gratis y otros pagos.
PostgreSQL es un proyecto Open Source, que está siempre a
la vanguardia.

Instalar PostgreSQL



SQL



```
1 // Cada Statement SQL termina en ;
2
3 CREATE DATABASE prueba;
4
5
6 // Con Create vamos a poder crear Bases de Datos, Tablas, índices, etc..
7
8 CREATE TABLE table_name
9 (
10 column_name1 data_type(size),
11 column_name2 data_type(size),
12 column_name3 data_type(size),
13 ....
14 );
```



SQL

```
1 CREATE TABLE ciudades
2 (
3     id serial PRIMARY KEY,
4     nombre varchar(255) UNIQUE
5 );
6
7 CREATE TABLE personas
8 (
9     id serial PRIMARY KEY,
10    apellido varchar(255) NOT NULL,
11    nombre varchar(255) UNIQUE,
12    ciudad integer references ciudades (id)
13 );
```

En SQL los tipos de datos están bien definidos, veamos una lista [acá](#).



SQL

```
1 CREATE TABLE ciudades
2 (
3     id serial PRIMARY KEY,
4     nombre varchar(255) UNIQUE
5 );
6
7 CREATE TABLE personas
8 (
9     id serial PRIMARY KEY,
10    apellido varchar(255) NOT NULL,
11    nombre varchar(255) UNIQUE,
12    ciudad integer references ciudades (id)
13 );
```

Además del tipo de datos, podemos definir **Constraints**. Que son reglas forzadas sobre un atributo.

- **NOT NULL Constraint** – El atributo no puede ser nulo.
- **UNIQUE Constraint** – No puede haber dos atributos iguales en esta tabla.
- **PRIMARY Key** – Identifica unívocamente una fila en una tabla.
- **FOREIGN Key** – La key debería existir sí o sí en otra tabla.



SQL

```
1 // Insertar filas en una tabla
2
3 INSERT INTO table_name (column1,column2,column3,...)
4 VALUES (value1,value2,value3,...);
5
6
7 // Ejemplo
8
9 INSERT INTO ciudades (nombre)
10 VALUES ('Tucuman');
11
12 INSERT INTO ciudades (nombre)
13 VALUES ('Buenos Aires');
14
15 INSERT INTO ciudades (nombre)
16 VALUES ('New York');
17
18 INSERT INTO personas (nombre, apellido, ciudad)
19 VALUES ('Toni', 'Tralice', 1);
20
21 INSERT INTO personas (nombre, apellido, ciudad)
22 VALUES ('Emi', 'Chequer', 3);
23
24 INSERT INTO personas (nombre, apellido, ciudad)
25 VALUES ('Fran', 'Etcheverri', 2);
```



SQL

```
1 // El Select nos permite recuperar filas de una tabla.
2
3 SELECT column_name,column_name
4 FROM table_name;
5
6 // Ejemplo
7
8 SELECT * FROM personas;
9
10 SELECT * FROM ciudades;
11
12 // el * indica que traiga todas las columnas disponibles
13
14
15 // ORDER BY CLAUSE
16
17 SELECT * FROM ciudades
18 ORDER BY nombre, id DESC;
19
20 // WHERE CLAUSE
21
22 SELECT * FROM personas
23 WHERE nombre = 'Toni'
24 AND apellido = 'Tralice';
```



SQL

```
1 // GROUP BY CLAUSE
2
3 # select * from COMPANY;
4 id | name | age | address | salary
5 ---+---+---+-----+-----
6 1 | Paul | 32 | California | 20000
7 2 | Allen | 25 | Texas | 15000
8 3 | Teddy | 23 | Norway | 20000
9 4 | Mark | 25 | Rich-Mond | 65000
10 5 | David | 27 | Texas | 85000
11 6 | Kim | 22 | South-Hall | 45000
12 7 | James | 24 | Houston | 10000
13
14
15 SELECT NAME, SUM(SALARY) FROM COMPANY GROUP BY NAME;
16
17 name | sum
18 ----+----
19 Teddy | 20000
20 Paul | 20000
21 Mark | 65000
22 David | 85000
23 Allen | 15000
24 Kim | 45000
25 James | 10000
```



SQL

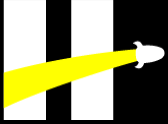


```
1 // SUB-QUERIES
2
3 SELECT column_name [, column_name ]
4 FROM   table1 [, table2 ]
5 WHERE  column_name OPERATOR
6        (SELECT column_name [, column_name ]
7         FROM table1 [, table2 ]
8         [WHERE])
```



SQL





SQL



```
1 // EJoin nos permite unir datos de diferentes tablas según una condición.  
2  
3 SELECT * FROM personas  
4 JOIN ciudades  
5   ON ciudades.id = personas.ciudad;
```

