



Structured Query Language

Data Definition Language

Administración de Bases de Datos



Departamento de
electrónica e informática

Carlos Juan Martín Pérez

SQL está en continua evolución: Es una evolución del lenguaje SEQUEL de D.D. Chamberlin y R.F. Boyce (1974) y fue implementado por primera vez por IBM en su BDR llamado SYSTEM R.

ISO (*International Standards Organization*) y **ANSI** (*American National Standards Institute*) desarrollaron una versión estándar en 1986, llamada SQL-86 o **SQL1**. Posteriormente, se desarrolló **SQL2** o SQL-92. Actualmente se desarrolla **SQL3**, que incluye conceptos de BD orientadas a objetos.

SQL es un lenguaje estándar para GESTIÓN de BDR:

Está incluido en muchos SGBD (DBMS)

Libres: PostgreSQL, MySQL/MariaDB, Ingres,...

Privativos: DB2, Oracle, Informix, Sybase, Access, SQL Server...

Las mismas sentencias sirven en distintos SGBD.

Si se usan sólo las características estándares facilita la tarea de migrar de SGBD pero casi todos los SGBD incluyen extensiones fuera del estándar que pueden ser una ventaja, pero hay que utilizarlas con cierta precaución porque también pueden secuestrar la decisión de cambiar de SGBD.

Fácil de usar y aprender: Tiene similitudes con el Álgebra Relacional, aunque se parece más al Cálculo Relacional de Tuplas, y definitivamente es más fácil e intuitivo que ambos lenguajes formales.

SQL es un lenguaje COMPLETO: Incluye sentencias para

DDL y DML*: Permite definir esquemas, consultar, actualizar...

Definición de vistas: Para ver la BD de distintas formas.

Seguridad: Permisos de acceso distintos para cada usuario.

Definir restricciones de integridad: Integridad referencial, de clave, unicidad...

Especificar control de transacciones: Para recuperación de errores, archivos históricos, replicación...

Puede incrustarse en lenguajes de alto nivel (C++, JAVA, PHP...).

Permite **operaciones tan complejas** que su total definición es complicada: Sólo veremos un subconjunto de las operaciones posibles.

* DDL: Data Definition Language, DML: Data Manipulation Language

TIPOS de DATOS de los atributos de las relaciones:

Enteros de distintos tamaños: INTEGER o INT y SMALLINT.

Reales de distinta precisión: FLOAT, REAL, DOUBLE PRECISION, NUMERIC(precisión, escala), donde precisión=número total de dígitos y escala=número de decimales.

Caracteres: CHAR(n) o CHARACTER(n), n=longitud fija. También VARCHAR(n), n=longitud máxima.

Cadenas de bits: BYTE para almacenar datos binarios (imágenes, archivos, etc.).

Fecha y Hora: DATE (año, mes, día: YYYY-MM-DD). TIME (horas, minutos, segundos: HH:MM:SS). TIMESTAMP incluye ambos.

Definiciones de DOMINIOS, TIPOS COMPUESTOS, ETC.:

Ejemplo: CREATE DOMAIN T_CEDULA AS CHAR(12);

Utilidades:

Hacer las definiciones más legibles o funcionales.

Hacer más fáciles los cambios de dominio de ciertos atributos.

Comando CREATE TABLE: Crea una nueva relación/tabla **base** con su nombre, atributos (nombres y dominios) y restricciones:

```
CREATE TABLE <NombreTabla> (  
    <NombreA1> <TipoA1> <restA1>, ...  
    <RestriccionesTabla>);
```

Restricciones de Atributos (puede haber varias por cada uno):

NOT NULL

UNIQUE

DEFAULT <valor> (el valor por defecto es NULL)

CHECK (<condición>)

Restricciones de Tabla: Pueden tener un nombre, que se asigna al principio, con el formato: CONSTRAINT <nombreR>

PRIMARY KEY (<lista de atributos de la llave primaria>)

UNIQUE (<atributos de llave candidata o secundaria>)

FOREIGN KEY (<llave ext.>) REFERENCES <tabla>(<atrib.>)

[ON DELETE (CASCADE | SET NULL | SET DEFAULT | RESTRICT)]

[ON UPDATE (CASCADE | SET NULL | SET DEFAULT | RESTRICT)]

Structured Query Language

Data Definition Language

Borrar Tablas: DROP

DROP TABLE <NombreTabla> [CASCADE | RESTRICT]

CASCADE: Borra la tabla (y su contenido). Si hay referencias sobre ella (llaves externas en otras tablas), dichas restricciones son borradas.

RESTRICT: Borra la tabla si no hay referencias sobre ella.

Modificar Tablas:

ALTER TABLE <NombreTabla> <ACCIÓN>

Añadir columna: **ADD <nombreA, tipo, restric.>;**

Borrar columna: **DROP <nombreA> [CASCADE|RESTRICT];**

Añadir restricción de Atributo: **ALTER <nombreA> SET <restricA>;**

Borrar restricción: **ALTER <nombreA> DROP <tipoRA: DEFAULT...>;**

Borrar restricción de tabla (debe tener un nombre):
DROP CONSTRAINT <NombreC> CASCADE;

Añadir restricción de tabla: **ADD <restric-T>;**

Structured Query Language

Data Definition Language

```
CREATE DOMAIN TIPO_DUI AS CHAR(10);  
CREATE TABLE CLIENTE (  
    DUI TIPO_DUI NOT NULL,  
    Nombre VARCHAR(80) NOT NULL,  
    CONSTRAINT Pk_Cliente PRIMARY KEY(DUI));  
CREATE TABLE MASCOTA (  
    DUI_Cliente TIPO_DUI NOT NULL,  
    Nombre VARCHAR(30) NOT NULL,  
    Fecha_Nacimiento DATE,  
    Especie VARCHAR(30) NOT NULL DEFAULT 'Perro',  
    CONSTRAINT Pk_Mascota PRIMARY KEY (DUI_Cliente,Nombre),  
    CONSTRAINT Fk_Mascota_Cliente FOREIGN KEY (DUI_Cliente)  
        REFERENCES CLIENTE(DUI) ON DELETE CASCADE ON UPDATE  
        CASCADE);  
ALTER TABLE CLIENTE ADD TELEFONO VARCHAR(20);  
ALTER TABLE CLIENTE DROP DUI CASCADE;  
ALTER TABLE MASCOTA DROP CONSTRAINT Fk_Mascota_Cliente  
    CASCADE;
```

Ahora conozcamos la cruda realidad:

- Tipos de datos:
<https://www.postgresql.org/docs/current/datatype.html>
- Generalidades de DDL
<https://www.postgresql.org/docs/11/ddl.html>
- Comandos DDL-SQL en profundidad
<https://www.postgresql.org/docs/11/sql-createtable.html>
<https://www.postgresql.org/docs/11/sql-droptable.html>
<https://www.postgresql.org/docs/11/sql-altertable.html>