



## SQL Historia

1975: Prototipo creado por IBM, conocido como SEQUEL.

1977: Cambia el nombre a **SQL (Structured Query Language)**.

1979: **Primer SGBDR** comercial basado en el (ORACLE).

1986: Aprobada la norma SQL/ANSI.

1987: **Primer estándar internacional** de **ISO-9075**.

1989: Nueva versión **ISO SQL-89** que añade integridad referencial básica.

1992: Versión **SQL-92** que amplía notablemente la anterior.

1999: Versión **SQL-99** que incorpora aspectos de **orientación a objetos** y amplía notablemente el lenguaje. Ya está incorporado a los principales SGBD-OR. (1126 pgs).

2003: Versión **SQL:2003** (estándar de 2000 páginas).



## SQL Historia

- **SQL-2003: ISO/IEC 9075-n:2003**

- Consta de varias partes. Algunas no están concluidas todavía.
- 1- Framework: Introducción con el marco de trabajo conceptual general.
- 2- Foundation: Define las estructuras de datos y operaciones para trabajar con BD relacionales y especifica la semántica y sintaxis del lenguaje.
- 3- Call-Level Interface (SQL/CLI): SQL embebido, uso de órdenes SQL desde otros lenguajes de programación generales.
- 4- Persistent Stored Modules (SQL/PSM): Procedimientos almacenados.
- 5- Host Language Bindings (SQL/Bindings): entre otros, incluye "On-Line Analytical Processing" (SQL/OLAP).
- 9- Management of External Data (SQL/MED): gestión de datos externos a una BD relacional mediante el uso de "wrappers" y "datalinks".
- 10- Object Language Bindings (SQL/OLB): SQL embebido en Java. También conocido como SQLJ.
- 11- Information and Definition Schemas (SQL/Schemata): Establece las estructuras y contenido del DEFINITION\_SCHEMA, es decir, los metadatos internos.
- 13- SQL Routines and Types Using the Java TM Programming Language (SQL/JRT): utilizar métodos y clases Java como si fuesen rutinas-SQL y tipos estructurados SQL.
- 14- XML-Related Specifications (SQL/XML): Maneras de utilizar SQL en conjunción con XML.



## SQL Principales Conceptos

En SQL-99 y SQL-2003: (sin orientación a objetos)

- Catálogo
- Esquema
- Dominio
- Tabla
- Columna
- Restricción de integridad
  - De tabla
  - De dominio
  - Aserción
- Vista
- Disparador

*Se estudia en detalle en las prácticas de laboratorio.  
Ver apéndice con la sintaxis de SQL.*



## SQL LDD – catálogo y esquema

- **Catálogo** (catalog)
  - Conjunto de esquemas. Proporciona un mecanismo adicional para calificar nombres de elementos:
    - **<catalogo>.<esquema>.<elemento>**
  - En cada catálogo existe un esquema especial (DEFINITION\_SCHEMA) que contiene una colección de tablas base con los **metadatos**:
    - USERS, TABLES, COLUMNS, DOAMINS, KEY\_COLUMN\_USAGE, ....
- **Esquema** (schema)
  - Representa la estructura de una base de datos.
  - Tiene un nombre, un identificador del propietario, y descriptores de elementos de esquema (schema objects).
  - Los elementos son dominios, tablas, vistas, restricciones, disparadores y otros constructores.



<definición de esquema> ::=

CREATE SCHEMA <cláusula de nombre de esquema>  
[ <elemento de esquema> ... ]

<cláusula de nombre de esquema> ::=

<nombre de esquema>  
| AUTHORIZATION <id. de autorización de usuario>  
| <nombre de esquema> AUTHORIZATION <id. de autorización de usuario>

<elemento de esquema> ::=

<definición de tabla>  
| <definición de vista>  
| <definición de dominio>  
| <definición de aserción>  
| <definición de disparador>  
| <asignación de privilegio> | ...

**CREATE SCHEMA Biblioteca  
AUTHORIZATION UCLM;**



## ● Dominio (domain)

- Tienen un nombre, un tipo de datos y, opcionalmente, un valor por defecto y una restricción.
- La restricción limita los valores del tipo de datos.

<definición de dominio> ::=

CREATE DOMAIN <nombre de dominio> [ AS ] <tipo de dato predefinido>  
[ <cláusula de defecto> ]  
[ <restricción de dominio> ]

<cláusula de defecto> ::= DEFAULT

{ <literal>  
| <función de valor tiempo/fecha>  
| USER | CURRENT\_USER  
| CURRENT\_ROLE | SESSION\_USER  
| SYSTEM\_USER | CURRENT\_PATH | ... }

**CREATE DOMAIN Tipos\_Doc CHAR(1)  
CHECK (VALUE("A", "L"));**



- **Tabla Base** (base table)

- Cada descriptor de una tabla base contiene:
  - Nombre de la tabla.
  - Lista de descriptores de columnas.
  - Indicador del tipo de tabla.
  - Opcionalmente, descriptores de restricciones de tabla.
  - Conjunto de una o varias dependencias funcionales.
  - Conjunto de una o varias claves candidatas.
  - La clave candidata principal (clave primaria).
- Los tipos de tablas base son:
  - Persistentes
  - Temporales:
    - Globales (GLOBAL TEMPORARY)
    - Locales (LOCAL TEMPORARY)



<definición de tabla> ::=

```
CREATE [ <alcance> ] TABLE <nombre>
    ( <elemento de tabla> [ {, <elemento de tabla>} ...] )
```

<alcance> ::= { GLOBAL | LOCAL } TEMPORARY

<elemento de tabla> ::=

```
{ <definición de columna>
| <definición de restricción de tabla> | .. }
```

```
CREATE TABLE Editorial
(Codigo_E  Codigos,
Nombre_E  Nombres NOT
NULL,
Direccion Dirs  NOT NULL,
Ciudad Lugares  NOT NULL,
PRIMAY KEY (Codigo_E),
UNIQuE  (Nombre_E));
```



- **Columna** (column)

- Tienen un nombre, un **tipo de dato** o dominio asociado, un valor por defecto (opcional) y un indicador de si pueden admitir valores nulos.
- También existen columnas generadas, cuyo valor se obtiene a partir de las demás columnas de la tabla evaluando una expresión.
- Los tipos de datos predefinidos son:
  - CHARACTER
  - CHARACTER VARYING
  - CHARACTER LARGE OBJECT
  - BINARY LARGE OBJECT
  - NUMERIC
  - DECIMAL
  - SMALLINT
  - INTEGER
  - BIGINT
  - FLOAT
  - REAL
  - DOUBLE PRECISION
  - BOOLEAN
  - DATE
  - TIME
  - TIMESTAMP
  - INTERVAL



<definición de columna> ::=

<nombre> [ <tipo de dato> | <dominio> ]  
[ <cláusula de defecto> | ... | <cláusula de generación> ]  
[ <definición de restricción de columna>... ]

<cláusula de generación> ::=

GENERATED ALWAYS AS ( <expresión valuada> )

<definición de restricción de columna> ::=

[ CONSTRAINT <nombre de restricción> ] <restricción de columna>  
[ <modo de restricción> ]



- **Restricción de Integridad** (integrity constraint)
  - De **tabla**
    - Unicidad (unique).
    - Clave primaria (primary key).
    - Integridad referencial (foreign key - references).
    - Control (check).
  - De **dominio**
    - Se aplican a cada columna (de cualquier tabla) que está basada en ese dominio, actuando como una restricción de tabla.
    - Incluyen una condición de búsqueda referida a los valores del dominio.
  - **Aserción**
    - Una restricción de control (check constraint) definida independientemente de cualquier tabla.
    - Incluyen una condición de búsqueda que puede referir al contenido de filas individuales de una o varias tablas, al contenido de tablas enteras o a un estado determinado por las relaciones entre varias tablas.



- Todas las restricciones son comprobadas con los mismos mecanismos y procedimientos:
  - Cada restricción tiene una condición que se evalúa a cierto o a falso.
  - Cada restricción tiene un **modo** (constraint mode):
    - DEFERRED => diferido
    - IMMEDIATE => inmediato
  - Las restricciones con el modo IMMEDIATE son comprobadas después de la ejecución de cada sentencia SQL.
  - Las restricciones con el modo DEFERRED son comprobadas al final de la ejecución de un grupo lógico de sentencias SQL (transacción).
  - Cada restricción tiene un modo por defecto que es el utilizado al iniciar una nueva sesión SQL o una nueva transacción.

<modo de restricción> ::=

{ INITIALLY DEFERRED | INITIALLY IMMEDIATE }  
[ [ NOT ] DEFERRABLE ]



- **Unicidad** (unique).
  - Dos filas de la tabla no pueden tener el mismo valor no nulo en las columnas indicadas.
  - Puede aplicarse al conjunto completo de columnas ( **UNIQUE (VALUE)** ).
- **Clave primaria** (primary key).
  - Es una restricción de unicidad pero en la cual las columnas no pueden tomar valor nulo (**integridad de entidades**).
- **Integridad referencial** (foreign key- references).
  - Representa una **clave ajena**.
  - Incluye una o más columnas referenciantes (referencing columns) y las correspondientes columnas referenciadas (referenced columns) de una tabla base referenciada (referenced table), que puede ser la referenciante.
  - Las columnas referenciadas deben tener una restricción de unicidad en la tabla referenciada.
  - Se puede establecer un modo de comparación total (**MATCH FULL**) o parcial (**MATCH PARTIAL**).
- **Control** (check).
  - Incluye una condición de búsqueda que sólo puede referir a elementos de la tabla que la incluye.



```
<definición de restricción de tabla> ::=  
[ CONSTRAINT <nombre de restricción> ] <restricción de tabla>  
[ <modo de restricción> ]  
  
<restricción de tabla> ::=  
{ <definición de restricción de unicidad>  
| <definición de restricción referencial>  
| <definición de restricción de control> }  
  
<definición de restricción de unicidad> ::=  
{ { UNIQUE | PRIMARY KEY } ( <lista de columnas> )  
| UNIQUE ( VALUE ) }  
  
<definición de restricción de control> ::=  
CHECK (<condición de búsqueda>)
```



**<definición de restricción referencial> ::=**

FOREIGN KEY ( <lista columnas referenciantes> )  
<especificación de referencia>

**<especificación de referencia> ::=**

REFERENCES <nombre tabla referenciada> [ (<lista columnas referenciadas> )  
[ MATCH { FULL | PARTIAL | SIMPLE } ]  
[ <acciones referenciales> ]

**<acciones referenciales> ::=**

{ <regla de modificación> [ <regla de borrado> ]  
| <regla de borrado> [ <regla de modificación> ] }

**<regla de modificación> ::=**

ON UPDATE { CASCADE | SET NULL | SET DEFAULT | RESTRICT | NO ACTION }

**<regla de borrado> ::=**

ON DELETE { CASCADE | SET NULL | SET DEFAULT | RESTRICT | NO ACTION }



- Las restricciones de columna son restricciones asociadas a una única columna de una tabla:

**<restricción de columna> ::=**

{ NOT NULL  
| { UNIQUE | PRIMARY KEY }  
| <especificación de referencia>  
| <definición de restricción de control> }





SQL

## LDD – restricciones de dominio y aserciones

<restricción de dominio> ::=

[ CONSTRAINT <nombre de restricción> ]

CHECK (<condición de búsqueda>)

[ <modo de restricción> ]

<definición de aserción> ::=

CREATE ASSERTION <nombre de restricción>

CHECK (<condición de búsqueda>)

[ <modo de restricción> ]

UCLM-ESI-BDa



SQL

## LDD – restricciones de dominio y aserciones

CREATE TABLE Documento

(Tipos Tipos\_Doc,

Cod\_Doc CHAR(4),

Titulo CHAR(25) NOT NULL,

Idioma Idiomas,

Nombre\_E Nombres,

Año INTEGER(4) CHECK (Año > 1950),

ISBN INTEGER(10),

PRIMARY KEY (Tipo, Cod\_Doc),

UNIQUE(ISBN)

CHECK ((Tipo = "A" AND

ISBN IS NULL AND

Nombre\_E IS NULL

OR

(Tipo = "L" AND

ISBN IS NOT NULL

NOMBRE\_E IS NOT NULL)),

FOREIGN KEY (Nombre\_E) REFERENCES TO Editorial

ON UPDATE CASCADE

ON DELETE NO ACTION));

UCLM-ESI-BDa



SQL

## LDD – restricciones de dominio y aserciones

CREATE ASSERTION

Idiomas\_No\_Usados\_Por\_Editoriales\_En\_Madrid

CHECK (NOT EXISTS

(SELECT \* FROM

Documento NATURAL JOIN Editorial

WHERE Idioma IN ("F", "A")

AND Ciudad = "Madrid"));

UCLM-ESI-BDa



SQL

## LDD – vistas

- **Vista** (view)

- Las vistas pueden ser **modificables** (WITH CHECK OPTION) dependiendo de diversos factores.
- Un descriptor de vista incluye:
  - El **nombre** de la vista.
  - La **expresión de la consulta**.
  - Un indicador de si es modificable.

<definición de vista> ::=

CREATE [ RECURSIVE ] VIEW <nombre de tabla> ( <lista de columnas> )

AS <expresión de consulta> [ WITH CHECK OPTION ]

- Además de las vistas normales o regulares existe otro tipo de vistas basadas en el uso de tipos de datos definidos por el usuario llamadas **referenciables** (referenceable view).

UCLM-ESI-BDa



```
CREATE VIEW Libro  
AS SELECT *  
FROM Documento  
WHERE Tipo = "L";
```



- **Disparador** (trigger)
  - Está asociado a una única tabla base.
  - Es el concepto clave para implementar **BD activas**.
  - Tiene tres partes:
    - Un **evento**: indica la acción sobre la tabla base que causará que se active el disparador.
      - INSERT, DELETE, o UPDATE
    - Un **tiempo de acción**: indica cuando se activará el disparo.
      - BEFORE => antes del evento.
      - AFTER => después del evento.
    - Una **acción**: Se llevan a cabo si ocurre el evento. Puede ser de dos tipos:
      - Una sentencia SQL ejecutable (SQL executable statement).
      - Un bloque atómico de sentencias SQL ejecutables.