

El Lenguaje SQL

José María Caveró

josemaria.cavero@urjc.es

1. Introducción

2. Sintaxis Básica del SQL

2.1.- Lenguaje de Definición de Datos

2.1.1.- Definición del esquema

2.1.2.- Evolución del esquema

2.2.- Lenguaje de Manipulación de Datos

2.2.1.- Actualizaciones

2.2.2.- Consultas

2.3.- Lenguaje de Control de Datos

2.3.1.- Recuperación y concurrencia

2.3.2.- Seguridad y confidencialidad

- 📖 Tecnología y Diseño de Bases de Datos M.Piattini, E. Marcos, C.Calero y B. Vela Ed.: RA-MA, 2006, Parte II, capítulo 8
- 📖 Fundamentos y Modelos de Bases de Datos. A. de Miguel y M. Piattini Ed.: RA-MA, 1997 Capítulo 7 (Pág. 215-265) y Apéndice A (Pág. 453-468)
- 📖 Introducción al SQL para Usuarios y Programadores Rivero, E. et al. Ed.: Thomson, Segunda Edición, 2003
- 📖 Diseño de Bases de Datos. Problemas Resueltos. A. de Miguel et al. RA-MA 2001
- 📖 Documentación de Oracle
(https://docs.oracle.com/cd/E11882_01/index.htm)

- ❑ SQL y el modelo relacional no son lo mismo
 - ❑ El modelo relacional utiliza los términos formales:
 - Relación (relation), tupla y atributo
 - ❑ SQL utiliza los términos más “user friendly”:
 - Tabla, fila y columna
 - ❑ SQL incorpora conceptos no relacionales (algún tipo de Row ID), los productos SQL incluyen características que no son relacionales ni pertenecen al estándar SQL
 - Índices, habilitar/deshabilitar restricciones...
 - ❑ ... y no incluyen características que sí
 - Dominios, aserciones, opciones en FK...

❑ Recordatorio: Bases de Datos Relacionales

- ❑ En 1970, Edgar F. Codd, matemático y empleado de IBM, publicó [“A Relational Model of Data for Large Shared Data Banks”](#)
 - El artículo comienza: *“Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation)”*
- ❑ Su modelo permitía acceder a la información sin conocer cómo o dónde se almacenaba, permitiendo la independencia de los datos
- ❑ Su propuesta era sencilla y revolucionaria. De hecho, inicialmente fue muy controvertida: los expertos pensaban que el modelo era demasiado simplista y que nunca podría ofrecer un buen rendimiento

- ❑ Las bases de datos relacionales han dominado el mundo de las bases de datos durante décadas
 - ❑ Los productos se han adaptado a almacenar datos en diferentes formatos (objetos, XML, ...)
 - ❑ Uno de los motivos es la existencia de SQL
- ❑ SQL, lenguaje estándar de BD Relacionales
 - ❑ En 1974, Raymond F. Boyce y Donald D. Chamberlin publican “SEQUEL: A Structured English Query Language”, el origen de SQL.
 - Chamberlin: *“Ray Boyce and I wanted to design a query language that had the expressive power of Ted Codd’s relational languages but was easier to understand by users who were not experts in set theory or formal logic”*
 - ❑ El primer producto comercial no fue de IBM, sino Oracle

- ❑ IBM trabajó en SQL y prototipos que lo implementaran
- ❑ El primer producto comercial no fue de IBM, sino Oracle
 - ❑ Fundada en 1977 por Larry Ellison, Bob Miner y Ed Oates
 - ❑ Sacó al mercado el primer SGBDR (como Relational Software, Inc.), Oracle V2, tras ver el potencial de los trabajos de Codd, Chamberlin y Boyce

❑ SQL (Structured Query Language)

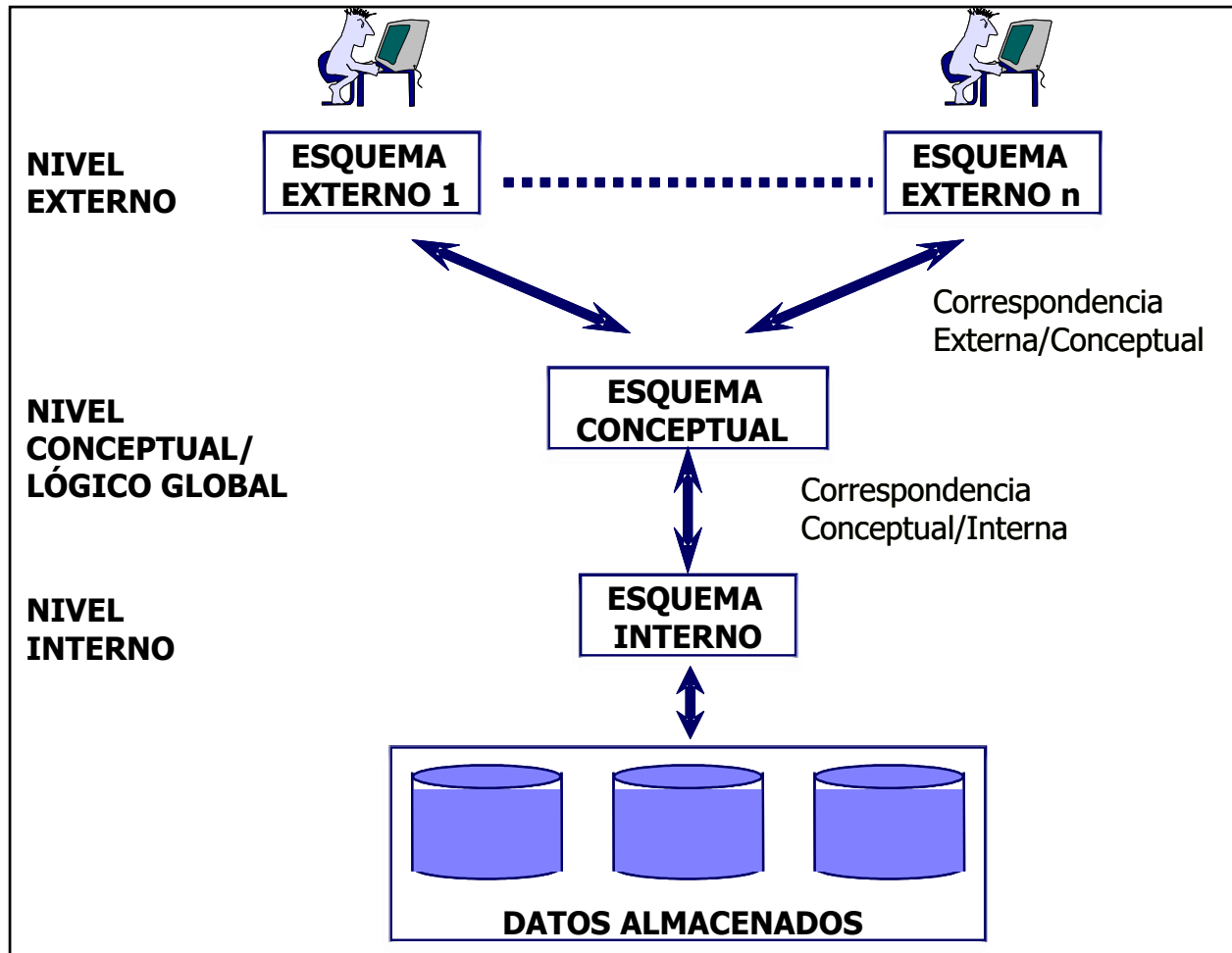
❑ Lenguaje de:

- **Definición**
- **Manipulación**
 - Consulta **interactiva**
 - **Programación**
- **Control**

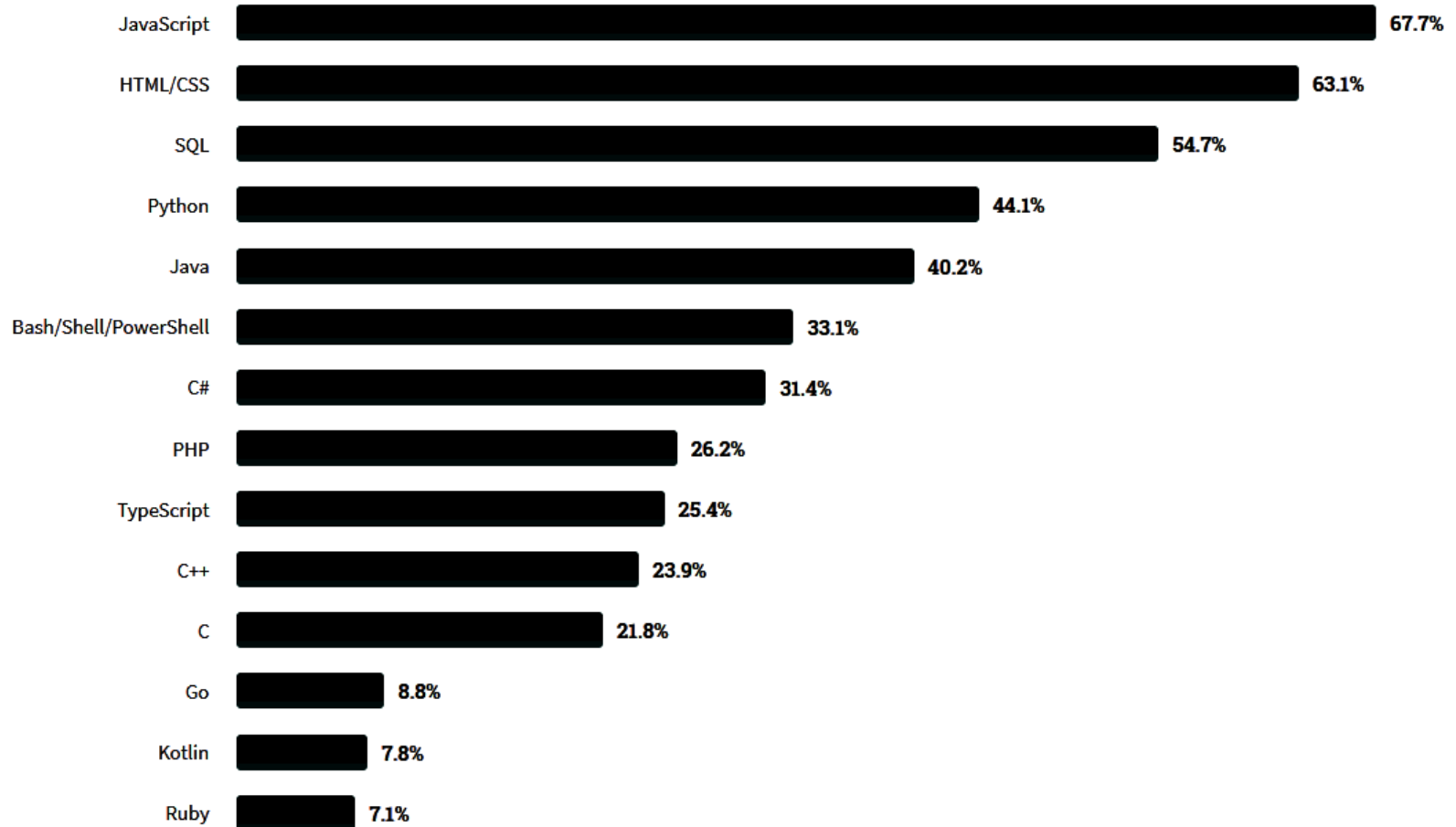
❑ Características:

- OPERA CON **CONJUNTOS DE REGISTROS**
- LENGUAJE **RELACIONAL y DECLARATIVO**

La Arquitectura ANSI/X3/SPARC



Stack Overflow Developer Survey 2020



□ TIOBE index

Oct 2020	Oct 2019	Programming Language	Ratings
1	2	C	16.95%
2	1	Java	12.56%
3	3	Python	11.28%
4	4	C++	6.94%
5	5	C#	4.16%
6	6	Visual Basic	3.97%
7	7	JavaScript	2.14%
8	9	PHP	2.09%
9	15	R	1.99%
10	8	SQL	1.57%
...			

DB-Engines Ranking

DB-Engines Ranking

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

Read more about the [method](#) of calculating the scores.



359 systems in ranking, October 2020

Rank			DBMS	Database Model	Score		
Oct 2020	Sep 2020	Oct 2019			Oct 2020	Sep 2020	Oct 2019
1.	1.	1.	Oracle +	Relational, Multi-model i	1368.77	-0.59	+12.89
2.	2.	2.	MySQL +	Relational, Multi-model i	1256.38	-7.87	-26.69
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	1043.12	-19.64	-51.60
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	542.40	+0.12	+58.49
5.	5.	5.	MongoDB +	Document, Multi-model i	448.02	+1.54	+35.93
6.	6.	6.	IBM Db2 +	Relational, Multi-model i	161.90	+0.66	-8.87
7.	↑ 8.	7.	Elasticsearch +	Search engine, Multi-model i	153.84	+3.35	+3.67
8.	↓ 7.	8.	Redis +	Key-value, Multi-model i	153.28	+1.43	+10.37
9.	9.	↑ 11.	SQLite +	Relational	125.43	-1.25	+2.80
10.	10.	10.	Cassandra +	Wide column	119.10	-0.08	-4.12

❑ Características

- ❑ Sintaxis cercana al inglés:
 - `SELECT nombre FROM empleado WHERE salario > = 3000;`
- ❑ Es capaz de operar con registros individuales y conjuntos de registros
- ❑ No procedimental. Simplemente descriptivo, enumera qué debería hacerse, y no cómo. El SGBD se encarga de elaborar el plan de ejecución
- ❑ *Case insensitive*
- ❑ Habitualmente existen muchas formas de resolver la misma consulta
 - `SELECT nombre FROM empleado WHERE salario NOT BETWEEN 0 AND 3000;`
 - `SELECT nombre FROM empleado WHERE salario > 3000;`

□ El lenguaje estándar

□ El proceso de estandarización se produce a dos niveles:

- Nacional: compañías, universidades, organismos de interés (ANSI, DIN, BSI,...), que son miembros de organismos internacionales
- Internacional: en el caso de SQL, Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 32, Data management and interchange.
 - El comité aprueba los estándares con el nombre ISO/IEC 9075-n:yyyy, donde n es el número de la parte correspondiente, e yyyy es el año de publicación

□ En ocasiones distintas partes se editan en diferentes años

□ Hasta 1996, el NIST certificaba el cumplimiento del estándar de los SGBDR. Actualmente, los propios productos incluyen una especificación del grado de cumplimiento

□ Los SGBDR proporcionan un soporte al estándar y, normalmente, muchas otras características adicionales no estándar.

- En ocasiones, la sintaxis de la parte estándar puede variar

1. Introducción

Versión	Características
SQL-86-87	Primera formalizada por ANSI. Intersección de implementaciones existentes
SQL-89	Integridad Referencial...
SQL-92	SQL2 Mejor tratamiento de restricciones, sintaxis de dominios, tablas temporales, nuevos tipos de datos, Lenguaje de manipulación de esquema, Combinación externa, SQL dinámico...
SQL:1999	SQL3 Disparadores, Orientación al Objeto (TAD, Encapsulamiento, Jerarquías, Herencia, IDO...), OLAP
SQL:2003	XML , nuevos tipos, Sequences, Valores autogenerados...
SQL:2006	Interacción XML-SQL, Xquery,...
SQL:2008	INSTEAD OF en triggers...
SQL:2011	Bases de datos temporales...
SQL:2016	Datos JSON

❑ Partes del lenguaje

- ❑ Aunque el estándar se organiza en diferentes grupos (SQL-schema statements, SQL-data statements...), lo habitual es dividirlos en las siguientes funciones:
 - Data Definition Language (DDL): sentencias de creación, borrado y modificación de los elementos
 - Data Manipulation Language (DML): modificación de los datos
 - Data Query Language (DQL): SELECT, a veces se incluye en el DML
 - Data Control Language (DCL): permisos
- ❑ El estándar también contiene la posibilidad de utilizar lenguaje procedimental, con distintas implementaciones y sintaxis en cada producto (PL/SQL en Oracle)

2. Sintaxis Básica del SQL

		Nivel Lógico Global	Nivel Externo	Nivel Físico
LENGUAJE DE DEFINICIÓN DE DATOS - LDD	Definición del Esquema (CREATE)	DOMAIN, TABLE, ASSERTION...	VIEW	INDEX
	Evolución del Esquema (ALTER, DROP)	DOMAIN, TABLE, ASSERTION...	VIEW	INDEX
LENGUAJE DE MANIPULACIÓN DE DATOS – LMD	Actualizaciones: Altas, Bajas, Modificaciones	INSERT ... DELETE... UPDATE...		
	Consultas	SELECT...		
LENGUAJE DE CONTROL DE DATOS – LCD	Recuperación y control de concurrencia	COMMIT, ROLLBACK...		
	Seguridad y Protección	GRANT, REVOKE...		

2. Sintaxis Básica del SQL

Tipos de datos de ANSI SQL	Tipos de Datos de Oracle
CHARACTER(n), CHAR(n)	CHAR (n)
CHARACTER VARYING(n), CHAR VARYING(n), VARCHAR(n)	VARCHAR2 (n), VARCHAR(n)*
NATIONAL CHARACTER(n), NATIONAL CHAR(n), NCHAR(n)	NCHAR(n)
NATIONAL CHARACTER VARYING(n) NATIONAL CHAR VARYING(n) NCHAR VARYING(n)	NVARCHAR2(n)
NUMERIC(p,s), DECIMAL(p,s)	NUMBER (p,s)
INTEGER, INT, SMALLINT	NUMBER(38)
FLOAT DOUBLE PRECISION REAL	FLOAT(126) FLOAT(126) FLOAT(63)
DATE, TIME, TIMESTAMP, INTERVAL	DATE , TIMESTAMP, INTERVAL

Nomenclatura utilizada para la sintaxis de algunas sentencias del SQL-92:

Extensión de la Forma Normal de Backus (BNF), siendo:

- < > símbolos no terminales del lenguaje
- ::= operador de definición
- [] elementos opcionales
- { } elementos en una fórmula
- | alternativa
- ... repetición

LDD: Definición del esquema

Nivel lógico global

DOMINIOS: Sintaxis

<definición de dominio> ::=

```
CREATE DOMAIN <nombre de dominio> [AS] <tipo de datos>  
[ DEFAULT <opción por defecto> ]  
[ <restricción de dominio> ]
```

literal,
función de valor tiempo o
fecha, o bien USER, SYSTEM
USER o NULL

<restricción de dominio> ::=

```
[ <definición de nombre de restricción> ]  
< definición de restricción de verificación>
```

<definición de nombre de restricción> ::=

```
CONSTRAINT <nombre de restricción>
```

<definición de restricción de verificación> ::=

```
CHECK ( <condición> )
```

2. Sintaxis Básica del SQL

2.1. Lenguaje de Definición de Datos

❑ LDD: Definición del esquema

❑ *Nivel Lógico Global: Dominios*

- CREATE DOMAIN **nomb_valido** AS CHARACTER;
- CREATE DOMAIN **nota_valida** NUMERIC
CHECK (VALUE BETWEEN 0 AND 10);
- CREATE DOMAIN **turno_valido** CHARACTER
DEFAULT 'M'
CONSTRAINT Mañana_o_Tarde
CHECK (VALUE IN ('M','T'));

ORACLE no admite Dominios

2. Sintaxis Básica del SQL

2.1. Lenguaje de Definición de Datos

□ *Nivel Lógico Global: Tablas*

<definición de tabla> ::=

CREATE [TEMPORARY] TABLE <nombre de tabla>

<lista de elementos de tabla>

<lista de elementos de tabla> ::=

(<elemento de tabla> [{ , <elemento de tabla> } ...])

<elemento de tabla> ::= <definición de columna>

| <definición de restricción de tabla>

<definición de columna> ::= <nombre de columna> { <tipo de datos> | <nombre de dominio> } [<cláusula de defecto>] [<definición de restricción de columna> ...]

<definición de restricción de columna> ::=

[<definición de nombre de restricción>]

<restricción de columna> [<atributos de restricción>]

<<restricción de columna> ::=

NOT NULL | <especificación de unicidad> |
<especificación de referencia> | <definición de restricción de verificación>

<definición de restricción de tabla> ::=

[<nombre de definición de restricción>]

<restricción de tabla> [<atributos de restricción>]

<restricción de tabla> ::=

<definición de restricción de unicidad>

| <definición de restricción referencial>

| <definición de restricción de verificación>

<definición de restricción de unicidad> ::=

<especificación de unicidad> (<lista de columnas únicas>)

<especificación de unicidad> ::=

UNIQUE | PRIMARY KEY

<lista de columnas únicas> ::= <lista de nombre de columnas>

<definición de restricción referencial> ::=

FOREIGN KEY (<columnas que ref.>)

<especificación de la referencia>

<especificación de la referencia> ::=

REFERENCES <columnas y tabla referenciadas>

[<acción referencial disparada>]

<columnas que ref.> ::= <lista de columnas de referencia>

<columnas y tabla referenciadas> ::=

<nombre de tabla> [(<lista de columnas de referencia>)]

<lista de columnas de referencia> ::= <lista de nombres de columnas>

<acción referencial disparada> ::=

<regla de modificación> [<regla de borrado>]

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

<regla de modificación> ::= ON UPDATE <acción referencial>

<regla de borrado> ::= ON DELETE <acción referencial>

<acción referencial> ::= CASCADE | SET NULL | SET DEFAULT

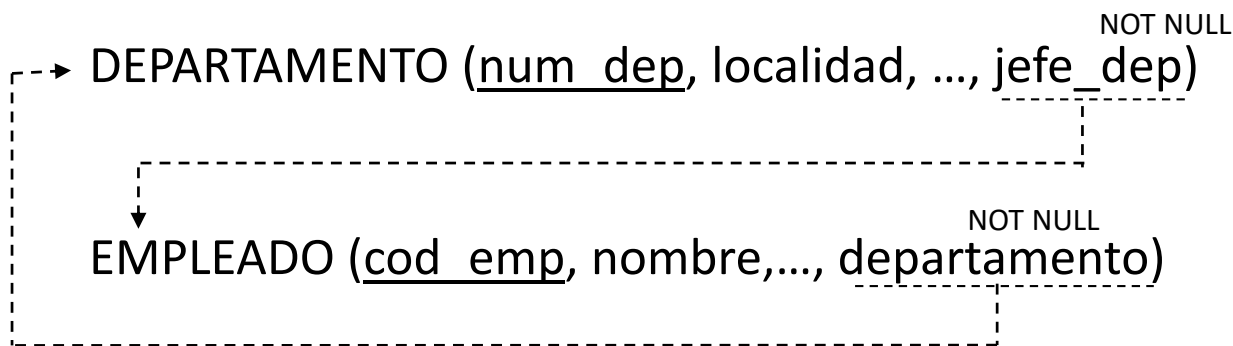
❑ Tiempo de acción de una restricción:

<atributos de restricción> ::= <tiempo de verificación de restricción> [[NOT] DEFERRABLE]

<tiempo de verificación de restricción> ::= INITIALLY DEFERRED | INITIALLY IMMEDIATE

❑ Los atributos de restricción sirven para indicar si la restricción es inmediata o diferida.

- Modo de verificación es inmediato: la restricción se verificará al finalizar cada sentencia
- Modo de verificación diferido: se verificará al finalizar la transacción



☐ **Habilitar/deshabilitar restricciones (No estándar)**

- ☐ Las restricciones pueden habilitarse o deshabilitarse utilizando las sentencias de CREATE TABLE o ALTER TABLE.
- ☐ Además, puede incluirse la opción de incluir los datos existentes o no (VALIDATE/NOVALIDATE).
 - ENABLE VALIDATE = ENABLE
 - ENABLE NOVALIDATE: habilitada para nuevas filas
 - DISABLE NOVALIDATE = DISABLE
 - DISABLE VALIDATE: deshabilitada pero no permite modificaciones.
Pensada para cargas masivas
- ☐ No funciona si la restricción implica tener que crear un índice único

2. Sintaxis Básica del SQL

2.1. Lenguaje de Definición de Datos

❑ Más resumido:

❑ CREATE TABLE nombre (
columna tipodatos | dominio [DEFAULT valorpordefecto] [restricciones de columna],
...
[restricciones de tabla],
...);

❑ Restricciones de columna:

❑ [CONSTRAINT nombrerestricción]
{[NOT] NULL |
UNIQUE/PRIMARY KEY |
REFERENCES tabla [(columna)] [{ON DELETE | ON UPDATE} CASCADE | SET NULL | SET DEFAULT] |
CHECK (condición) }

❑ Restricciones de tabla

❑ [CONSTRAINT nombrerestricción]
{UNIQUE |
PRIMARY KEY (columna/s) |
FOREIGN KEY (columnas)
REFERENCES tabla [(columnas)] [{ON DELETE | ON UPDATE} CASCADE | SET NULL | SET DEFAULT] |
CHECK (condición)}

❑ LDD: Definición del esquema

❑ *Nivel lógico global*

❑ Opciones de integridad referencial:

■ ON UPDATE | ON DELETE

- CASCADE
- SET NULL
- SET DEFAULT
- ***NO ACTION***

- Por defecto, borrado y actualización restringidos (NO ACTION)

¡ORACLE **no** soporta
ON UPDATE ni
ON DELETE SET DEFAULT!

2. Sintaxis Básica del SQL

2.1. Lenguaje de Definición de Datos

EJEMPLO:

ALUMNO (num_mat, nombre, ciudad, cod_grupo)

GRUPO (cod_grupo, curso, turno)

Modificación: Cascada
Borrado: Puesta a nulos

2. Sintaxis Básica del SQL

2.1. Lenguaje de Definición de Datos

LDD: Definición del esquema

Nivel lógico global

Restricción de Tabla

Restricción de Columna

```
CREATE TABLE grupo  
(cod_grupo CHARACTER,  
curso INTEGER NOT NULL,  
turno TURNO_VALIDO,  
PRIMARY KEY (cod_grupo),  
CHECK (curso > 0 AND curso < 4));
```

ó

```
CREATE TABLE grupo  
(cod_grupo CHARACTER PRIMARY KEY,  
curso INTEGER NOT NULL,  
turno TURNO_VALIDO,  
CHECK (curso > 0 AND curso < 4));
```

```
CREATE TABLE alumno (  
    num_mat CHARACTER,  
    nombre NOMB_VALIDO UNIQUE,  
    ciudad CHARACTER NOT NULL,  
    cod_grupo CHARACTER,  
    PRIMARY KEY (num_mat),  
    FOREIGN KEY (cod_grupo) REFERENCES grupo  
        ON UPDATE CASCADE  
        ON DELETE SET NULL);
```

❑ En Oracle

❑ CREATE TABLE grupo

```
(cod_grupo CHAR(3) PRIMARY KEY,  
curso NUMBER(1) NOT NULL,  
turno CHAR(1) DEFAULT 'M'  
CONSTRAINT Manana_o_Tarde CHECK (turno IN ('M','T')),  
CHECK (curso > 0 AND curso < 4)); // o BETWEEN
```

❑ CREATE TABLE alumno

```
(num_mat CHAR(3),  
nombre VARCHAR2(20) UNIQUE,  
ciudad CHAR(25) NOT NULL,  
cod_grupo CHAR(3),  
PRIMARY KEY (num_mat),  
FOREIGN KEY (cod_grupo) REFERENCES grupo ON DELETE SET NULL);
```

LDD: Definición del esquema

Nivel lógico global

Esquema:

Alumno

num_mat	nombre	ciudad	<i>cod_grupo</i>

Grupo

cod_grupo	curso	turno

2. Sintaxis Básica del SQL

2.1. Lenguaje de Definición de Datos

LDD: Definición del esquema

Nivel lógico global

Una forma de ver la estructura
de una tabla en ORACLE:
⇒ **DESC[RIBE] nombre_tabla;**

```
SQL> desc alumno;
```

Nombre	¿Nulo?	Tipo
NUM_MAT	NOT NULL	CHAR(3)
NOMBRE		VARCHAR2(20)
CIUDAD	NOT NULL	CHAR(25)
COD_GRUPO		CHAR(3)

```
SQL> desc grupo;
```

Nombre	¿Nulo?	Tipo
COD_GRUPO	NOT NULL	CHAR(3)
CURSO	NOT NULL	NUMBER(1)
TURN0		CHAR(1)

Avance LMD

Consultas:

```
SELECT columna  
FROM tabla  
WHERE condición;
```

```
SELECT nombre  
FROM ALUMNO  
WHERE ciudad = 'MADRID';
```

```
//nombre  
//de los alumnos  
//de MADRID
```

```
SELECT *  
FROM ALUMNO;
```

```
//todos los atributos  
//de todos los alumnos
```


LDD: Definición del esquema *Nivel lógico global*

ASERCIONES:

¡ORACLE **no** soporta
la creación de aserciones!

*Todos los alumnos de Madrid tienen que estar matriculados en
el turno de tarde.*

```
CREATE ASSERTION madrid_tarde  
CHECK (NOT EXISTS
```

*(Consulta que devuelva los datos
que **incumplen la restricción**, es decir, los
alumnos de Madrid que NO están
matriculados en el turno de tarde));*

LDD: Definición del esquema *Nivel lógico global*

ASERCIONES:

¡ORACLE **no** soporta
la creación de aserciones!

*Todos los alumnos de Madrid tienen que estar matriculados en
el turno de tarde.*

```
CREATE ASSERTION madrid_tarde  
CHECK (NOT EXISTS
```

*Alumnos de
Madrid que NO
están
matriculados en
el turno de tarde*

```
(SELECT *  
FROM ALUMNO, GRUPO  
WHERE alumno.cod_grupo=grupo.cod_grupo  
AND alumno.ciudad = 'MADRID'  
AND grupo.turno <> 'T'));
```

LDD: Definición del esquema

Nivel externo

Vistas:

```
CREATE [OR REPLACE] VIEW nombre_de_vista  
[(lista de columnas)]  
AS <cláusula SELECT>
```

```
CREATE VIEW alumnos_madrid  
AS (SELECT *  
FROM alumno  
WHERE ciudad = 'Madrid');
```

Todos los atributos de los
alumnos de Madrid

```
CREATE VIEW alumnos_madrid  
(nombre_alumno)  
AS (SELECT nombre  
FROM alumno  
WHERE ciudad = 'Madrid');
```

Nombre de los alumnos de
Madrid

LDD: Definición del esquema

Nivel físico

Índices:

```
CREATE INDEX ind_alumno  
      ON alumno (ciudad, cod_grupo);
```

o

```
CREATE UNIQUE INDEX ind_alumno  
      ON alumno (ciudad, cod_grupo);
```

□ LDD: Evolución del esquema

□ *Nivel Lógico Global*

■ ALTER TABLE

- Sirve para modificar una tabla:
 - » añadir (ADD), modificar (MODIFY), borrar (DROP), renombrar (RENAME) columnas, restricciones ...
 - » Habilitar (ENABLE) o deshabilitar (DISABLE) restricciones
 - » ...

■ DROP TABLE, DROP DOMAIN, DROP ASSERTION

□ *Nivel Externo*

■ DROP VIEW, ALTER VIEW

□ *Nivel Físico*

■ DROP INDEX, ALTER INDEX

❑ Ejemplos

❑ LDD: Evolución del esquema

❑ Nivel Lógico Global

- Añadir el precio (atributo) que cada alumno paga al matricularse:
 - `ALTER TABLE alumno ADD (precio INTEGER);`
- Para borrar la columna sería:
 - `ALTER TABLE alumno DROP COLUMN precio;`
- Otros ejemplos:
 - `DROP DOMAIN turno_valido;`
 - `DROP TABLE grupo [CASCADE CONSTRAINTS];`
 - `DROP ASSERTION ciudad_turno;`

❑ Nivel Externo

- `DROP VIEW alumnos_madrid;`

❑ Nivel Físico

- `DROP INDEX ind_alumno;`