



BASES DE DATOS

Grado en Ingeniería Informática

RESUMEN Tema 4 (*parte 1/3*)

LDD

LMD (Instrucciones de Actualización)

Diccionario de datos de Oracle

Curso 2022- 23

1. Lenguaje de Definición de Datos (LDD)

El lenguaje SQL puede utilizarse como DDL (*Data Definición Language*) o Lenguaje de Definición de Datos (LDD). El LDD permite:

- Definir y crear tablas
- Suprimir tablas
- Modificar la definición de las tablas
- Definir tablas virtuales (vistas) de datos
- Construir índices para hacer más rápido el acceso a las tablas

NOTA: la **notación** utilizada para la especificación de los **comandos de SQL** es la siguiente:

- Las *palabras clave* se indican en **mayúsculas**
 - Los corchetes [] indican **opcionalidad**
 - Las llaves { } delimitan **alternativas** separadas por el símbolo |
 - Los puntos suspensivos (...) indica **repetición**
- } **BNF**

1.1 Sentencias sobre Tablas

1.1.1 Crear una Tabla

```
CREATE TABLE nombre_tabla (  
  { nombre_atributo tipo_dato [restricción_atributo] ...  
  | restricción_tabla } [,  
  { nombre_atributo tipo_dato [restricción_atributo] ...  
  | restricción_tabla } ] ...  
);
```

Donde **restricción_atributo** tiene la sintaxis:

```
[CONSTRAINT nombre_restricción]  
{ [NOT] NULL  
| DEFAULT valor  
| { UNIQUE | PRIMARY KEY }  
| REFERENCES nombre_tabla [(nombre_atributo)] [ON DELETE CASCADE]  
| CHECK (condición)  
}
```

Las restricciones **DEFAULT** y **[NOT] NULL** se suelen colocar junto al atributo, sin asociarle nombre de restricción.

La cláusula **CONSTRAINT** sirve para asignarle un nombre a una **restricción**

Ejemplo:

```
CREATE TABLE CURSO (
    nomCurso varchar2(20) NOT NULL,
    codCurso char(3),
    profesor char(9),
    maxAlum number(2, 0),
    fechaInic date,
    fechaFin date,
    horas number(3, 0) NOT NULL,
    CONSTRAINT cursosClave PRIMARY KEY (codCurso),
    CONSTRAINT cursosFechasValidas
        CHECK (fechaFin > fechaInic),
    CONSTRAINT cursosCodigosValidos
        CHECK (codCurso IN ('C01','C02','C03','C04','C05')),
    CONSTRAINT cursosAjena
        FOREIGN KEY (profesor) REFERENCES PROFESOR(dni),
    CONSTRAINT cursosNomUnico UNIQUE (nomCurso));
```

-Restricciones al final

-Usar nombres de restricciones distintos para cada tabla

Y donde **restricción_tabla** tiene la sintaxis:

```
[CONSTRAINT nombre_restricción]
{{UNIQUE | PRIMARY KEY} (nombre_atributo [,nombre_atributo] ...)}
| FOREIGN KEY (nombre_atributo [,nombre_atributo] ...)
REFERENCES nombre_tabla [(nombre_atributo [,nombre_atributo] ...)]
[ON DELETE CASCADE | SET NULL]
| CHECK (condición)
}
```

Ejemplos:

```
CREATE TABLE ASIGNATURA
(
    ...
    CONSTRAINT asig-prof FOREIGN KEY (prof) REFERENCES PROFESOR (nPr)
    ON DELETE SET NULL );
```

Esta sentencia indica que si eliminamos un profesor de la tabla PROFESOR, las tuplas que hacían referencia a dicho profesor en la tabla ASIGNATURA ponen un valor nulo en el atributo prof.

```
CREATE TABLE ASIGNATURA
(
    ...
    CONSTRAINT asig-prof FOREIGN KEY (prof) REFERENCES PROFESOR (nPr)
    ON DELETE CASCADE);
```

Esta sentencia indica que si eliminamos un profesor responsable de alguna asignatura, se eliminará la asignatura correspondiente en la tabla ASIGNATURA.

Tipos de datos

- Numéricos

SQL	Tipo	Oracle
→ INTEGER	Números enteros de distintos tamaños	NUMBER(38, 0)
INT		NUMBER(38, 0)
SMALLINT		NUMBER(38, 0)
FLOAT(<i>n</i>)	Números reales de distinta precisión	NUMBER
REAL		NUMBER
DOUBLE PRECISION		NUMBER
DECIMAL(<i>p</i> , <i>e</i>)	Número real con <i>p</i> dígitos y <i>e</i> decimales	NUMBER(<i>p</i> , <i>e</i>)
DEC(<i>p</i> , <i>e</i>)		NUMBER(<i>p</i> , <i>e</i>)
→ NUMERIC(<i>p</i> , <i>e</i>)		NUMBER(<i>p</i> , <i>e</i>) ←

- Cadenas de caracteres

SQL	Tipo	Oracle
→ CHAR(<i>n</i>)	Cadena de caracteres de longitud fija	CHAR(<i>n</i>)
CHARACTER(<i>n</i>)		CHAR(<i>n</i>)
→ VARCHAR(<i>n</i>)	Cadena de caracteres de longitud variable	VARCHAR2(<i>n</i>)

- Cadenas de bits

SQL	Tipo	Oracle
BIT(<i>n</i>)	Longitud fija con <i>n</i> bits	RAW(<i>n</i>)
BIT VARYING(<i>n</i>)	Longitud variable con <i>n</i> bits como máximo	LONG RAW

- Fecha y hora

SQL	Tipo	Oracle
→ DATE	Año, mes y día	DATE
→ TIMESTAMP	Fecha y hora	TIMESTAMP(<i>p</i>), siendo <i>p</i> el número de dígitos de la parte fraccionaria de los segundos. Por defecto, <i>p</i> = 6.

1.1.2 Eliminar una tabla

```
DROP TABLE nombre_tabla [CASCADE CONSTRAINT];
```

1.1.3 Modificar una tabla

```
ALTER TABLE nombre_tabla
{ADD nuevo_nombre_atributo tipo [NOT NULL] [CONSTRAINT restricción]
| MODIFY nombre_atributo tipo_nuevo [CONSTRAINT restricción]
| DROP COLUMN nombre_atributo [CASCADE CONSTRAINT]
| DROP (nombre_atributo1, nombre_atributo2, ...)
| ADD CONSTRAINT nombre_restricción restricción
| { DROP | ENABLE | DISABLE } CONSTRAINT nombre_restricción [CASCADE]
};
```

**** NO se pueden MODIFICAR restricciones → borrar + crear ****

Ejemplo:

```
ALTER TABLE ASIGNATURA DROP CONSTRAINT cursos;
```

```
ALTER TABLE ASIGNATURA ADD CONSTRAINT cursos
CHECK (curso IN ('1','2','3','4','5'));
```

2.- Lenguaje de Manipulación de Datos (LMD)

A) Instrucciones de actualización

A.1) Comando INSERT

```
INSERT INTO nombre_tabla  
[(nombre_atributo [, nombre_atributo] ...)]  
{VALUES (valor [, valor] ...)}  
};
```

Por **ejemplo**, para añadir una nueva tupla a la tabla ORDENADOR, podemos hacer:

```
INSERT INTO ORDENADOR  
VALUES ('Ord220', 'Ordenador Multimedia', 'Aula 8');
```

Se puede **especificar explícitamente cuáles son los campos** a los que se le va a insertar un valor. A los atributos que no se indican se le asigna el valor NULL o su valor por defecto (si lo tiene). Por ejemplo, si queremos introducir un profesor, pero no conocemos todos sus datos, podemos realizar la siguiente instrucción:

```
INSERT INTO PROFESOR (nPr, dni, nombre)  
VALUES ('30', '29.555.555', 'Antonio Díaz Sotelo');
```

A.2) Comando DELETE

```
DELETE FROM nombre_tabla  
[WHERE condición];
```

Ejemplo:

Borra de la tabla ORDENADOR todos los ordenadores que sean estaciones SUN.

```
DELETE FROM ORDENADOR  
WHERE tipo = 'Estación Sun';
```

A.3) Comando UPDATE

```
UPDATE nombre_tabla
SET {nombre_atr = valor [, nombre_atr = valor, ...]
| nombre_atr [, nombre_atr, ...] = (subconsulta)
}
[WHERE condición];
```

Ejemplo:

Modifica el cuatrimestre y la especialidad de la asignatura 'Análisis Numérico'.

```
UPDATE ASIGNATURA
SET cuat = 2, esp = 'S'
WHERE nombre = 'Análisis Numérico';
```

3.- El Diccionario de Datos de Oracle

El comando `describe nombre_tabla` permite ver la definición de una tabla concreta de la base de datos. En esta descripción aparecerán los nombres de los atributos, el tipo de datos de cada atributo y si tiene o no permitidos valores nulos.

Ejemplo: `describe ALUMNO`

Algunas de las **tablas** del **diccionario de datos** son:

- **user_tables**

Contiene información sobre las tablas de un usuario. Podemos usar el sinónimo **tabs**.

Ejemplo:

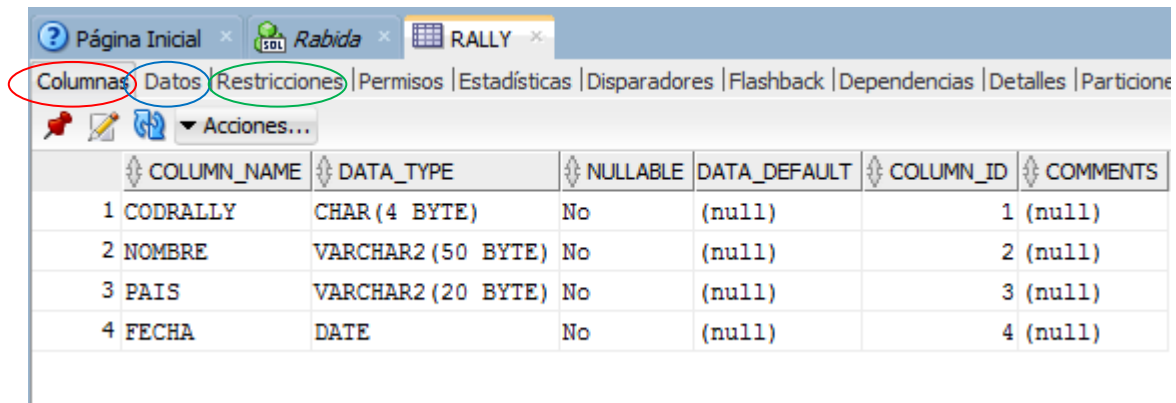
```
select table_name
from user_tables;
```

- **user_constraints**

Contiene información sobre las restricciones definidas por el usuario.

Ejemplo:

```
select constraint_name, constraint_type, search_condition, status
from user_constraints
where table_name='ALUMNO';
```

Consultar Información del diccionario de datos de tablas desde SQLDeveloper

The screenshot shows the SQL Developer interface with the 'Columns' tab selected for a table named 'RALLY'. The 'Columns' tab is circled in red. The 'Datos' tab is circled in blue, and the 'Restricciones' tab is circled in green. The table structure is displayed in a table format with the following columns: COLUMN_NAME, DATA_TYPE, NULLABLE, DATA_DEFAULT, COLUMN_ID, and COMMENTS.

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	CODRALLY	CHAR(4 BYTE)	No	(null)	1	(null)
2	NOMBRE	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	PAIS	VARCHAR2(20 BYTE)	No	(null)	3	(null)
4	FECHA	DATE	No	(null)	4	(null)

Pestañas:

- **Columnas:** Información de los atributos y tipos
- **Datos:** Datos cargados en la tabla (tuplas)
- **Restricciones:** Constraints (restricciones) definidas sobre la tabla