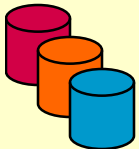


Práctica 4

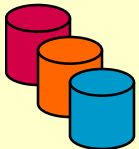
Lenguaje de Definición de Datos

- Tipos de Datos
- Tipos de Operadores
- Sentencias
 - ▶ INSERT (LMD)
 - ▶ DELETE (LMD)
 - ▶ CREATE (LDD)
 - ▶ DROP (LDD)



TIPOS DE DATOS (LDD)

- **NUMBER(p,s)** Representa números
 - ▶ p
 - Precisión, número de dígitos
 - máx 38, mín 1
 - ▶ s
 - Escala, número de cifras decimales
 - máx 127, mín -84)
- **VARCHAR2(size)** Representa cadena de caracteres de longitud variable
 - ▶ size
 - Longitud
 - máx 4000, mín 1
- **DATE** Tipo Fecha
- **LONG** Cadena de caracteres de longitud variable de hasta 2 gigabytes



TIPOS DE OPERADORES (LMD)

■ Sumar, Restar, Concatenar

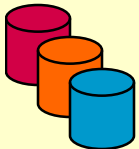
- +
- -
- ||

■ Multiplicar, Dividir

- *
- /

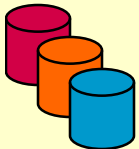
■ Comparadores clásicos

- =
- ==>
- <=
- <
- >



TIPOS DE OPERADORES (LMD)

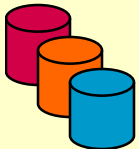
- Comparadores especiales
 - IS NULL
 - BETWEEN
 - IN
 - LIKE
- Operadores Lógicos Clásicos
 - NOT
 - AND
 - OR



TIPOS DE OPERADORES (LMD)

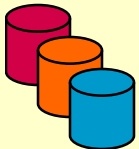
■ Comparadores especiales

- ▶ **IS NULL** → Detecta valores nulos
(A=10), A IS NULL \Rightarrow false, A IS NOT NULL \Rightarrow true
- ▶ **BETWEEN** → Detecta valores entre dos constantes
A BETWEEN X and Y \Leftrightarrow A \geq X y A \leq Y
- ▶ **IN** → Detecta pertenencia a conjunto
A IN (1,2,3) \Rightarrow true si A=1 o A=2 o A=3
- ▶ **LIKE** → Sirve para utilizar “máscaras” en cadenas de caracteres
 - **_** sustituye cualquier carácter
x LIKE ‘_A_’ \Rightarrow true si x=‘1A23’ \Rightarrow false si x=‘1A234’
 - **%** sustituye cualquier cadena
x LIKE ‘%A%’ \Rightarrow true si x=‘1AX’ \Rightarrow true si X=‘ABLA’



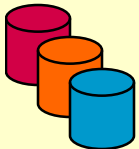
SENTENCIAS

- **INSERT (LMD)**
- **DELETE (LMD)**
- **CREATE TABLE (LDD)**
- **DROP TABLE (LDD)**



CREATE TABLE (LDD)

```
CREATE TABLE [usuario.]nombre_tabla (  
    Nombre_atributo1    tipo1,  
    Nombre_atributo2    tipo2,  
  
    ....  
    Nombre_atributon    tipon,  
    [CONSTRAINT nombre_de_restricción Tipo de  
    restricción (atributo[s]) afectados]  
);
```



■ PRIMARY KEY Identifica la Clave Primaria

► Sintaxis

CONSTRAINT *nombre_de_restricción* **PRIMARY KEY** (atributo o atributos implicados)

Ejemplos:

CONSTRAINT CP_ejemplo **PRIMARY KEY** (nombre_ejemplo),

CONSTRAINT PK_empleados **PRIMARY KEY** (nombre,apellido1,apellido2),

■ UNIQUE El atributo que tenga esta restricción ha de tomar valores únicos. Si hay más de un atributo, la combinación ha de ser única.

► Sintaxis

CONSTRAINT *nombre_de_restricción* **UNIQUE** (nombre de atributo o atributos)

Ejemplos:

CONSTRAINT AU_ejemplo **UNIQUE** (nom_dep),

CONSTRAINT unicos **UNIQUE** (nomprof,nomasig,numaula),



Restricciones de tabla

- **CHECK** Comprueba si el atributo implicado pertenece a un conjunto o rango de valores

- ▶ **Sintaxis**

CONSTRAINT *nombre_de_restricción* **CHECK** (*nombre_atributo* **IN** (conjunto de valores)),

CONSTRAINT *nombre_de_restricción* **CHECK** (*nombre_atributo* **BETWEEN** valor1 **AND** valor2)

Ejemplos:

CONSTRAINT CHIN_valor_crédito **CHECK** (numcred **IN** (3,6,9,12);

CONSTRAINT Chequea_nota **CHECK** (nota **BETWEEN** 0 **AND** 10);

- **FOREIGN KEY** Obliga a la existencia de los valores de ese atributo en el atributo clave primaria de otra tabla a la que referencia

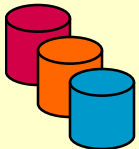
- ▶ **Sintaxis**

CONSTRAINT *nombre_de_restricción* **FOREIGN KEY** (*nombre_atributo/s*) **REFERENCES** *nombre_tabla_ref* (*nombre_atributo/s*)

Ejemplos:

CONSTRAINT FK_matricula **FOREIGN KEY** (*cod_est*) **REFERENCES** *estudiantes* (*dni*),

En la tabla que se está creando existe un atributo llamado *cod_est* que hace referencia al atributo *dni* de la tabla *estudiantes*, esto permite unir la tabla actual con la tabla *estudiantes*.

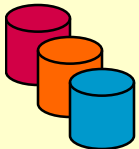


Restricciones de columna

- Hay un tipo de restricciones, que sólo afectan a un atributo y se indican en la sentencia de creación a continuación del atributo correspondiente:

```
CREATE TABLE [usuario.]nombre_tabla (  
    ...,  
    Nombre_atributok   tipok [TIPO RESTRICCIÓN],  
    ....  
);
```

- Las restricciones específicas de columna son **NOT NULL**, **DEFAULT**.
- No obstante cualquier otra restricción que afecte a un único atributo puede ser descrita en la sentencia de creación como si fuera una restricción de columna. No es recomendable, impide darle nombre a la restricción.



Restricciones de columna

- **NOT NULL** defecto **NULL** Impide insertar valores **NULOS** en el atributo correspondiente. Por

- ▶ **Sintaxis**

Nombre_atributo tipok **NOT NULL**,

Ejemplo:

```
CREATE TABLE estudiante(  
    .....  
    apellido    VARCHAR2(15) NOT NULL,  
    .....  
);
```

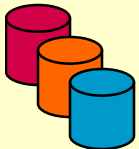
- **DEFAULT** *valor* Si no se inserta ningún valor en ese atributo, por defecto valdrá *valor*

- ▶ **Sintaxis**

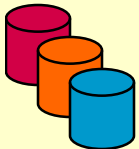
Nombre_atributo tipok **DEFAULT** *valor_tipok* ,

Ejemplo:

```
CREATE TABLE estudiante(  
    .....  
    curso                      NUMBER(1) DEFAULT 1,  
    .....  
);
```



```
CREATE TABLE <Nombre tabla> (  
  <Nombre columna> <tipo de dato> [NOT NULL]  
    [DEFAULT <valor_defecto> ],  
  
  [CONSTRAINT <nombre_restricción>]  
    [PRIMARY KEY (<lista columnas>)],  
    [FOREIGN KEY (<lista columnas>) REFERENCES (<nombre tabla>)],  
    [UNIQUE (<lista columnas>)],  
    [CHECK (condición de búsqueda)] );
```

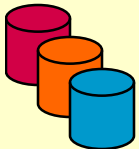


INSERT INTO ... VALUES (LMD)

- Esta sentencia se utiliza para insertar valores en las tablas creadas

```
INSERT INTO nombre_tabla  
  [(atributo1, atributo2,..., atributon)]  
VALUES (valor1,valor2,...,valorn);
```

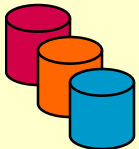
- ▶ *nombre_tabla* Nombre de la tabla en la que se quieren insertar los datos.
- ▶ [*(atributo1, atributo2,..., atributon)*] es **optativa** cuando se inserta una fila completa y ordenada. Cuando sólo se insertan valores para algunos atributos, o estos atributos están desordenados es **obligatoria**.
- ▶ (*valor1,valor2,...,valorn*) valores del mismo tipo que los atributos correspondientes.



► Ejemplos:

```
INSERT INTO estudiante VALUES  
('1342567',3344,'Juan','Pérez',2);
```

```
INSERT INTO estudiante (dni,nia,apellido) VALUES  
('723564',3443,'López');
```

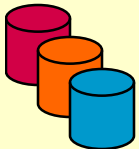


DELETE FROM (LMD)

- Esta sentencia se utiliza para borrar todas o algunas filas de una tabla

DELETE FROM *nombre_tabla* [*WHERE condición_de_selección*];

- ▶ Sin el corchete se borran todas las filas de la tabla.
- ▶ *WHERE condición_de_selección* En este caso sólo se borrarán las filas seleccionadas.



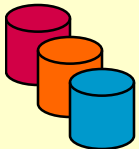
► Ejemplos:

DELETE FROM estudiante;

Borra todas las filas de la tabla estudiante.

DELETE FROM estudiante
WHERE curso=4;

Borra de la tabla estudiante las filas que en el atributo curso tienen el valor 4.

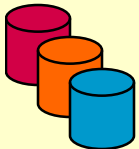


DROP TABLE (LDD)

- Esta sentencia se utiliza para borrar una tabla (y su contenido).

```
DROP TABLE NombreTabla  
[CASCADE CONSTRAINTS] /;
```

- ▶ **CASCADE CONSTRAINTS** Se utiliza para poder borrar una tabla que es referenciada por otra u otras para las que su clave primaria es clave ajena.



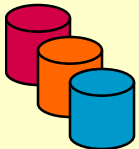
► Ejemplos:

DROP TABLE notas;

Los atributos: *cod_est*, *cod_asig*, *convoc* que constituyen la clave primaria de la tabla *notas* no están referenciados por ninguna otra tabla.

DROP TABLE estudiante **CASCADE CONSTRAINTS**;

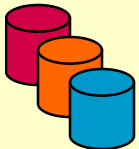
Como el atributo *dni* de *estudiante* es referenciada por la clave ajena *cod_est* en la tabla *notas* es obligado añadir la cláusula **CASCADE CONSTRAINTS**



Resumen: CREATE/DROP TABLE (LDD)

```
CREATE TABLE <Nombre tabla> (  
  <Nombre columna><tipo de dato> [NOT NULL]  
    [DEFAULT <valor_defecto> ],  
  
  [CONSTRAINT <nombre_restricción>  
    [PRIMARY KEY (<lista columnas>)],  
    [FOREIGN KEY (<lista columnas>) REFERENCES (<nombre tabla>)],  
    [UNIQUE (<lista columnas>)],  
    [CHECK (condición de búsqueda)] );
```

```
DROP TABLE NombreTabla  
  [CASCADE CONSTRAINTS] ;
```



Resumen: INSERT / DELETE (LMD)

INSERT INTO <nombre tabla> [(<lista-de columnas>)]
VALUES (<lista-de valores>)

INSERT INTO <nombre tabla> [(<lista-de columnas>)]
<sentencia SELECT>

DELETE FROM <nombre tabla>
[WHERE <condición>]

Si no se pone condición, se borran todas las filas de la tabla

