



# Bases de Datos

UD 5: Creación y manejo de tablas.  
Incluyendo restricciones

Félix Reyes Fernández

# Índice

SQL\*PLUS

Tipos de Objetos de Bases de Datos

Manipulación de Tablas

Concepto de Constraint

Formato de definición de Constraint

Tipos de Constraints. Chequeando Constraints

Manipulación de Constraints

# SQL\*PLUS

- ▶ SQL\*Plus es un entorno en el cual puedes hacer lo siguiente:
  - Ejecutar sentencias SQL para recuperar, modificar, añadir y eliminar datos de la base de datos.
  - Formatear, realizar operaciones de cálculo, almacenar e imprimir resultados de consultas en forma de informes.
  - Crear ficheros de scripts para almacenar sentencias SQL para usos repetitivos en el futuro.

# Comandos SQL\*Plus

- ▶ **STA[RT] filename [.ext]**: Arranca un fichero de comandos almacenado previamente
- ▶ **@filename**: Igual que START
- ▶ **EXIT**: Abandonar SQL\*Plus
- ▶ **CLEAR SCR**: Limpiar la pantalla
- ▶ **SPOOL y SPOOL OFF**: Sirve para almacenar todo lo que hacemos en ficheros
- ▶ **DESCRIBE**: Lista los atributos de las tablas
- ▶ **SET PAUSE ON**: Hace que SQL\*Plus pare al principio de cada página. Sólo se pasa a la siguiente página cuando el usuario pulsa RETURN.
- ▶ **SET LINESIZE TAMAÑO**: establece la longitud de las líneas en caracteres
- ▶ **SHOW LINESIZE**: muestra el tamaño actual de las líneas
- ▶ **SET PAGESIZE TAMAÑO**: establece el número de líneas por página
- ▶ **SHOW PAGESIZE**: muestra el número de líneas por página
- ▶ **COLUMN columna FORMAT a50**: Darle más longitud a una columna (no numérica)
- ▶ **SHOW USER**: Muestra el usuario que está conectado.

# Tipos de Objetos de una Base de Datos

Objeto	Descripción
Tabla	Unidad básica de almacenamiento, formada por filas y columnas
Vista	Representación lógica de un subconjunto de datos extraídos de una o más tablas
Secuencia	Estructura que genera valores automáticos
Índices	Variables que mejoran la eficiencia de consultas a las bases de datos
Sinónimos	Estructuras que dan nombres alternativos a otros objetos

# Tipos de Objetos de una Base de Datos

- Los nombres de las tablas y de las columnas deben empezar con una letra y no deben superar los 30 caracteres de longitud
- Deben contener solo los caracteres A-Z, a-z, 0-9, \_, \$, #
- Los nombres no deben duplicarse para el mismo usuario, pero si pueden ser igual para distintos (Concepto de esquema)
- No deben ser palabras reservadas de ORACLE.
- Deben usarse nombres descriptivos para cualquier objeto de la base de datos
- En el caso de tener campos relacionados, se debe intentar poner el mismo nombre para los dos.
- No es sensible a las mayúsculas y minúsculas

# Tipos de Objetos de una Base de Datos

Tipo de dato	Descripción
VARCHAR2(n)	Almacena datos alfanuméricos de longitud variable. Si el valor insertado es menor que N, el resto lo ajusta al tamaño del valor → Ganamos espacio de almacenamiento
CHAR(n)	Almacena datos alfanuméricos de longitud fija. Si el valor insertado es menor que N, el resto lo rellena con espacios en blanco.
NCHAR y NVARCHAR2	Almacena datos alfanuméricos de longitud fija-variable usando caracteres Unicode.

# Tipos de Objetos de una Base de Datos

Tipo de dato	Descripción
NUMBER(p, s)	Almacena números. La p representa los dígitos totales, y la s los dígitos decimales.
DATE	Almacena fechas y horas concretas.
TIMESTAMP	Más específico que Date, almacenando incluso fracciones de segundo.
TIMESTAMP WITH TIME ZONE	Almacena fechas relativas a la zona horaria y a la región

# Manipulación de tablas

## ▶ Creación de tablas

```
CREATE TABLE nombre_tabla  
  (nombre_columna1 tipo_dato [DEFAULT expresión],  
   nombre_columna2 tipo_dato [DEFAULT  
   expresión],....)
```

# Manipulación de tablas

- ▶ **Creación de tablas: Ejercicio**
- ▶ Crear una tabla que sirva para almacenar los datos personales de una serie de personas. Concretamente el dni, nombre, apellidos, fecha nacimiento, número de hermanos y estado civil(s:soltero, c:casado, d:divorciado, v:viudo)

# Manipulación de tablas

## ▶ Creación de tablas: Ejercicio

```
CREATE TABLE persona  
    (dni CHAR(9),  
     nombre VARCHAR2(50),  
     apellidos VARCHAR2(50),  
     fecha_nac DATE,  
     numero_hermanos NUMBER(2),  
     estado_civil CHAR(1));
```

## ▶ Describe (desc) Nombre\_tabla

# Manipulación de tablas

- ▶ Creación de tablas: Ejercicio (opción DEFAULT)
  - ▶ Crear una tabla que sirva para almacenar los datos personales de una serie de personas.  
Concretamente el dni, nombre, apellidos, fecha de alta en el sistema, número de hermanos y estado civil(s:soltero, c:casado, d:divorciado, v:viudo).  
Especifica que la fecha de alta en el sistema sea por defecto la fecha actual del sistema. Además, que el estado civil sea por defecto soltero.

# Manipulación de tablas

## ▶ Creación de tablas: Ejercicio

```
CREATE TABLE persona
  (dni CHAR(9),
  nombre VARCHAR2(50),
  apellidos VARCHAR2(50),
  fecha_alta DATE DEFAULT SYSDATE,
  numero_hermanos NUMBER(2),
  estado_civil CHAR(1) DEFAULT 'S');
```

# Manipulación de tablas

## ▶ Añadiendo comentarios

```
COMMENT ON (TABLE nombre_tabla |  
COLUMN nombre_tabla.nombre_columna)  
IS 'comentario a incluir'
```

- ▶ Para borrar un comentario se incluye la cadena vacía
- ▶ Para ver los comentarios acudimos a las siguientes tablas:
  - ALL\_COL\_COMMENTS
  - USER\_COL\_COMMENTS
  - ALL\_TAB\_COMMENTS
  - USER\_TAB\_COMMENTS

# Manipulación de tablas

- ▶ Comentarios: Ejercicio
- ▶ Insertar un comentario a la tabla persona donde se describa cual es el contenido de sus datos.

# Manipulación de tablas

## ► Comentarios: Ejercicio

```
COMMENT ON TABLE persona  
IS 'Datos referentes a los componentes de una  
asociación'
```

# Manipulación de tablas

## ▶ Modificar Estructura Tablas: Alter Table

```
ALTER TABLE nombre_tabla  
ADD (nombre_columna tipo_dato [DEFAULT  
expresion],  
     nombre_columna tipo_dato [DEFAULT  
expresion],...)
```

```
ALTER TABLE nombre_tabla  
MODIFY (nombre_columna tipo_dato [DEFAULT  
expresion],  
        nombre_columna tipo_dato [DEFAULT  
expresion],...)
```

# Manipulación de tablas

## ▶ Modificar Estructura Tablas: Alter Table

```
ALTER TABLE nombre_tabla  
DROP COLUMN nombre_columna
```

# Manipulación de tablas

## ▶ **Modificar Estructura Tablas: Ejercicio**

- ▶ Modifica el sistema para que pueda añadir datos acerca del número de hijos que tuviese cada persona y que por defecto valga 0
- ▶ Modifica el sistema para que pueda se puedan almacenar en el campo apellidos 35 bytes mas de los que tiene.
- ▶ Debido a restricciones del problema, el sistema no tiene porque guardar información acerca del numero de hermanos de cada persona. Realiza los cambios necesarios en el sistema.

# Manipulación de tablas

## ▶ Modificar Estructura Tablas: Ejercicio

```
ALTER TABLE persona  
ADD (numero_hijos NUMBER DEFAULT 0)
```

```
ALTER TABLE persona  
MODIFY (apellidos VARCHAR2(85))
```

```
ALTER TABLE persona  
DROP COLUMN numero_hermanos
```

# Manipulación de tablas

## ▶ Cambiar Nombre Objeto: Rename

```
RENAME nombre_antiguo TO nombre_nuevo
```

## ▶ Ejercicio: Cambiar el nombre de la tabla persona por el de socios

```
RENAME persona TO socios
```

# Manipulación de tablas

- ▶ **Truncate Table:** borra todas las filas de una tabla. Vacía su contenido y libera el espacio de memoria ocupado por esos datos, dejando intacta la estructura de la tabla.

```
TRUNCATE TABLE nombre_tabla
```

- ▶ **Ejercicio:** Dar de baja a todas las personas de la tabla persona

```
TRUNCATE TABLE persona
```

# Manipulación de tablas

- ▶ **Drop Table:** libera el espacio de memoria ocupado tanto por las filas como por las columnas. Borra los datos de una tabla y la tabla en sí.

```
DROP TABLE nombre_tabla
```

- ▶ **Ejercicio:** Dar de baja a la tabla persona

```
DROP TABLE persona
```

# Concepto de CONSTRAINT

- ▶ Mecanismos que hacen cumplir una serie de reglas a la hora de insertar valores dentro de los campos de una tabla. Previenen acerca de datos no válidos en las tablas.
- ▶ Se pueden usar para:
  - Prevenir el borrado de una tabla si existen dependencias con otras tablas de la base de datos
  - Hace cumplir reglas a nivel de tabla ya sea insertando, modificando o borrando datos.

# Concepto de CONSTRAINT

Restricción	Descripción
NOT NULL	La columna no puede tener valores nulos
UNIQUE	La/s columna/s deben tener valores únicos
PRIMARY KEY	Definimos la clave primaria de una tabla
FOREIGN KEY	Relaciona datos de diferentes tablas mediante integridad referencial
CHECK	Una determina condición que debe cumplirse para una columna

Almacenadas en USER\_CONSTRAINTS

# Formato de definición de CONSTRAINT

- ▶ **Nombre restricción**

```
nombretabla_nombrecolumna_abreviaturarestricción
```

- ▶ **Si no se especifica un nombre, Oracle crea uno automáticamente**
- ▶ **Definiciones a nivel de columna**

```
CREATE TABLE nombre_tabla  
  
(columna tipo_dato [DEFAULT expresión],  
...,columna tipo_dato [CONSTRAINT nombre_restricción] tipo_restricción,...)
```

- ▶ **Definiciones a nivel de tabla**

```
CREATE TABLE nombre_tabla  
  
(columna tipo_dato [DEFAULT expresión],  
..., CONSTRAINT nombre_restricción tipo_restricción (columna), ...)
```

# Formato de definición de CONSTRAINT

- ▶ Visualizando restricciones en el Diccionario de Datos

```
SELECT constraint_name, constraint_type  
FROM user_constraints  
WHERE table_name='nombre_tabla';
```

```
SELECT constraint_name, column_name  
FROM user_cons_columns  
WHERE table_name='nombre_tabla';
```

# Tipos de Constraints. Chequeando Constraints

- ▶ NOT NULL: Solo a nivel de columna

```
CREATE TABLE nombre_tabla  
(columna tipo_dato NOT NULL,  
...)
```

- ▶ Ejercicio: Se desea crear una tabla para almacenar los datos de los alumnos de primero de DAM. Concretamente se desea guardar el dni, nombre, apellidos y el teléfono móvil de todos los alumnos, siendo obligatorio que nos lo faciliten.

# Tipos de Constraints. Chequeando Constraints

- ▶ Ejercicio NOT NULL

```
CREATE TABLE alumnos_NN  
  (dni CHAR(9),  
   nombre VARCHAR2(50),  
   apellidos VARCHAR2(50),  
   teléfono CHAR(9) NOT NULL)
```

# Tipos de Constraints. Chequeando Constraints

## ▶ UNIQUE

```
CREATE TABLE nombre_tabla  
(columna tipo_dato,  
CONSTRAINT nombretabla_nombre_colum_u  
UNIQUE (columna), ...)
```

- ▶ **Ejercicio:** Se desea crear una tabla para almacenar los datos de una zapatería. Concretamente se desea almacenar el número identificativo de cada zapato, modelo, color y marca. Nuestra zapatería tiene un convenio con los fabricantes y solo se trae el mejor zapato por cada marca.

# Tipos de Constraints. Chequeando Constraints

## ▶ Ejercicio UNIQUE

```
CREATE TABLE zapateria_U  
  (id_zapato VARCHAR2(9),  
   marca VARCHAR2(50),  
   modelo VARCHAR2(50),  
   color VARCHAR2(9) ,  
   CONSTRAINT zapateria_marca_u  
   UNIQUE(marca));
```

# Tipos de Constraints. Chequeando Constraints

- ▶ **PRIMARY KEY:** Incluye las restricciones UNIQUE y NOT NULL

```
CREATE TABLE nombre_tabla  
(columna tipo_dato,  
CONSTRAINT nombretabla_nombre_colum_pk  
PRIMARY KEY (columna1[, columna2,...]),
```

- ▶ **Ejercicio:** Se desea crear una tabla para almacenar los datos de una zapatería. Concretamente se desea almacenar el número identificativo de cada zapato, modelo, color y marca. Define el campo identificativo de cada zapato como clave primaria de la tabla.

# Tipos de Constraints. Chequeando Constraints

## ▶ Ejercicio PRIMARY KEY

```
CREATE TABLE zapateria_PK  
  (id_zapato VARCHAR2(9),  
   marca VARCHAR2(50),  
   modelo VARCHAR2(50),  
   color VARCHAR2(9) ,  
   CONSTRAINT zapateria_id_pk PRIMARY KEY  
   (id_zapato));
```

# Tipos de Constraints. Chequeando Constraints

## ▶ FOREIGN KEY

```
CREATE TABLE nombre_tabla  
(columna tipo_dato,  
CONSTRAINT tabla_ columna_fk FOREIGN KEY  
(columna[,columna2...])  
    REFERENCES  
tabla_padre(columna[,columna2...])  
    ON DELETE CASCADE, ...)
```

- ▶ ON DELETE CASCADE
- ▶ ON DELETE SET NULL
- ▶ ON DELETE NO ACTION: Por defecto (No se especifica la cláusula). Si se intenta eliminar un registro de la tabla referenciada por una "foreign key", Oracle no lo permita y muestre un mensaje de error

# Tipos de Constraints. Chequeando Constraints

- ▶ FOREIGN KEY
- ▶ **Ejercicio:** Se desea crear almacenar los datos de las motocicletas que hay en Cádiz y sus propietarios. Concretamente de las motocicletas se desea guardar la matricula, marca, modelo, color y propietario. Sobre los propietarios, se desea almacenar su dni, nombre, apellidos y teléfono. Además, para guardar la consistencia entre las tablas, cuando borremos un propietario se deberá borrar todos los registros relacionados en la tabla motocicletas.

# Tipos de Constraints. Chequeando Constraints

- ▶ Ejercicio FOREIGN KEY

```
CREATE TABLE propietarios  
    (dni_propietarios CHAR(9),  
     nombre VARCHAR2(50),  
     apellidos VARCHAR2(50),  
     telefono CHAR(9),  
     CONSTRAINT propietario_dni_pk  
     PRIMARY KEY (dni_propietarios))
```

# Tipos de Constraints. Chequeando Constraints

## ▶ Ejercicio FOREIGN KEY

```
CREATE TABLE motos
  (matricula VARCHAR2(9),
   marca VARCHAR2(50),
   modelo VARCHAR2(50),
   color VARCHAR2(9),
   dni_propietario CHAR(9),
   CONSTRAINT motos_mat_pk PRIMARY KEY
   (matricula),
   CONSTRAINT motos_dni_fk FOREIGN KEY
   (dni_propietario)
   REFERENCES propietarios(dni_propietarios)
   ON DELETE CASCADE);
```

# Tipos de Constraints. Chequeando Constraints

## ▶ CHECK

```
CREATE TABLE nombre_tabla  
(columna tipo_dato,  
CONSTRAINT nombretabla_nombrecolumna_ck  
CHECK expresión,...)
```

- ▶ **Ejercicio:** Se desea crear una tabla para almacenar los datos de una zapatería. Concretamente se desea almacenar el número identificativo de cada zapato, modelo, color y marca. Define el campo identificativo de cada zapato como clave primaria de la tabla y los zapatos solo pueden ser negros, verdes, azules o dorados.

# Tipos de Constraints. Chequeando Constraints

## ▶ Ejercicio CHECK

```
CREATE TABLE zapateria_CK
    (id_zapato VARCHAR2(9),
     marca VARCHAR2(50),
     modelo VARCHAR2(50),
     color VARCHAR2(9) ,
     CONSTRAINT zapateria_marca_pk PRIMARY KEY
     (id_zapato),
     CONSTRAINT zapateria_color_ck CHECK (color
     ='negro'
     OR color ='verde'
     OR color ='azul'
     OR color ='dorado'));
```

# Manipulación de restricciones

- ▶ Añadir restricción
- ▶ A nivel de tabla

```
ALTER TABLE nombre_tabla  
ADD CONSTRAINT nombre_restriccion  
TIPO_RESTRICCIÓN columnna;
```

- ▶ A nivel de columna

```
ALTER TABLE nombre_tabla  
MODIFY (columna tipo_dato NOT NULL);
```

# Manipulación de restricciones

- ▶ **Borrar restricción**

```
ALTER TABLE nombre_tabla  
DROP CONSTRAINT nombre_restriccion ;
```

- ▶ **Deshabilitar restricción**

```
ALTER TABLE nombre_tabla  
DISABLE CONSTRAINT nombre_restriccion ;
```

- ▶ **Habilitar restricción**

```
ALTER TABLE nombre_tabla  
ENABLE CONSTRAINT nombre_restriccion ;  
,
```

# Ejercicios

- ▶ 1.- Crea la tabla *departamento* para que almacene la identificación del departamento y el nombre de cada departamento que constituyen una empresa.
- ▶ 2.- Crea la tabla *empleado* para que contenga información sobre el identificador de empleado, el nombre, los apellidos y el número de departamento en el que trabaja en la empresa.
- ▶ 3.- Modifica la tabla *empleado* para que pueda almacenar nombres más largos.
- ▶ 4.- Confirma que ambas tablas, tanto *departamento* como *empleado*, están almacenadas en el diccionario de datos.
- ▶ 5.- Crea la tabla empleados2 basada en la tabla empleado, pero que solo contenga el número del empleado, en nombre del empleado y el departamento en el que trabaja.
- ▶ 6.- Borra la tabla empleado.
- ▶ 7.- Renombra la tabla empleados2 a empleado.
- ▶ 8.- Añade un comentario a ambas tablas, describiendo su contenido.

# Ejercicios

- ▶ **Restricciones**
- ▶ 1.- Añade a la tabla empleado la restricción PRIMARY KEY usando la columna de ID. La restricción debe especificarse en la creación de la tabla.
- ▶ 2.- Crea una restricción PRIMARY KEY en la tabla departamento usando el campo ID. La restricción debe especificarse en la creación de la tabla.
- ▶ 3.- Añade una clave foránea a la tabla empleado que asegure que los empleados no se asignan a departamentos inexistentes.
- ▶ 4.- Confirma la creación de dichas restricciones consultando el diccionario de datos.

# Trabajo Final

- ▶ Desarrollar un script con la creación de las tablas del trabajo. Incluye todas las restricciones que consideres necesarias

# Ruegos y Preguntas

