

# Gestión de bases de datos

Diseño físico de bases  
de datos

# Actividad 1

Crea una tabla **piezas**:

- ☐ Id int
- ☐ Nombre varchar2(100)
- ☐ Id\_fabricante int

Crea otra tabla **fabricantes**:

- ☐ Id int
- ☐ Marca varchar(50)

Incluye las siguientes **restricciones**:

- ☐ Id en piezas es PK.
- ☐ Id\_fabricante en piezas es FK referenciado a id PK en fabricantes.
- ☐ Id en fabricantes es PK.

# RESTRICCIÓN TIPO CHECK

Permite validar el valor de un campo mediante una expresión.

Ej.:

```
ALTER TABLE ALUMNOS  
ADD CONSTRAINT CHKALUMNOEDAD  
CHECK (EDAD >= 18)
```

# Restricción CHECK

Se emplea para indicar una condición que debe cumplirse para el valor de un campo que se desea añadir o editar.

Si se quiere incluir la condición dentro de CREATE TABLE, debe realizarse respecto del propio campo donde se aplica la restricción (no se pueden añadir otros campos a la restricción).

# Restricción CHECK

Si se quieren añadir varios campos a una restricción CHECK, se debe realizar en un ALTER TABLE una vez creada la tabla a priori.

# Actividad 2

Incluye un nuevo campo en la tabla piezas de la actividad 1 que se llame precio y sea number con cuatro dígitos, de los cuáles dos serán decimales.

Ahora crea una restricción CHECK en el campo precio que permita solo valores de precio mayores que cero.

# EXPRESIONES REGULARES

Sirven para indicar que un campo tenga una estructura de valores concreta, como una máscara.

Se emplea

`REGEXP_LIKE(campo,'MÁSCARA')`

Se suele usar dentro de una restricción `CHECK` para que un campo tenga esa estructura concreta en sus valores.

# EXPRESIONES REGULARES

La máscara puede contener ser:

- Caracteres: 'abcd'
- Números: 1234...
- Letra concreta entre una serie posible de valores: [a-z], [A-D], etc.
- Número concreto entre una serie posible de valores: [0-2], [0-9], etc.



# EXPRESIONES REGULARES

Ejemplo campo DNI:

```
check(regex_like(dni,'[0-9] [0-9] [0-9]  
[0-9] [0-9] [0-9] [0-9] [0-9][A-Z]'))
```

# Actividad 3

Incluye un nuevo campo en la tabla piezas de la actividad 1 que se llame código, tipo char con 5 caracteres con la siguiente estructura (restricción):

- ☐ Primero la letra P.
- ☐ Segundo, tres números que van de 0 a 9 cada uno de ellos.
- ☐ Final una letra cualquiera del abecedario en mayúsculas.

Ej.: P123A, P000B, P999Z.

# CHECK COMPARANDO CADENAS DE CARACTERES

Se puede emplear la restricción check para asegurarnos que un campo tiene un valor concreto, o bien un valor de una lista posible de valores.

Para ello tan solo hay que indicar dicho valor dentro del check, entre comillas simples, y añadir la condición OR para permitir varios valores posibles. También se puede usar el comando IN.

# **CHECK COMPARANDO CADENAS DE CARACTERES**

Ejemplo:

CHECK (campo = 'UNO' OR campo = 'DOS')

CHECK (campo IN ('UNO','DOS'))

# Actividad 4

Añade una restricción al campo marca de la tabla fabricantes, permitiendo que solo se puedan guardar los siguientes valores en ese campo:

- ☐ FABRIC1
- ☐ FABRIC2
- ☐ FABRIC3

Puedes usar tanto IN como encadenar diferentes condiciones con OR.

# CHECK CON FUNCIONES

También se pueden incluir funciones dentro de nuestro CHECK, como por ejemplo para requerir que el campo tenga todos los caracteres en mayúsculas).

Ej.: `CHECK(NOMBRE = UPPER(NOMBRE))`

# CHECK CON FECHAS

También se pueden incluir restricciones de fechas en un CHECK.

Ej.:

```
CHECK(FECHA BETWEEN DATE '2022-01-15'  
AND DATE '2022-01-18')
```

```
CHECK(FECHA NOT BETWEEN DATE '2022-  
01-15' AND DATE '2022-01-18')
```

# CONSULTA DE TABLAS

Para saber las tablas que tiene una base de datos en Oracle se puede consultar la vista `user_tables`.

Ej.:

```
SELECT TABLE_NAME FROM USER_TABLES;
```



# **CONSULTA DE ESTRUCTURA DE UNA TABLA**

Para saber la estructura de una tabla se emplea el comando DESCRIBE.

Ej.:

**DESCRIBE ALUMNOS;**

# RENOMBRAR TABLA

Se puede modificar el nombre de una tabla haciendo uso de RENAME.

Ej.:

```
RENAME ALUMNOS TO ESTUDIANTES;
```



# CEU

*Centro de Estudios  
Profesionales*

---

*Fundación San Pablo Andalucía*