

BASE DE DATOS

● CREAR UN NUEVO USUARIO EN ORACLE

Desde la conexión de **SYSTEM** nos creamos un usuario.

Nos vamos a CONEXIÓN -> OTROS USUARIOS -> Click derecho (Crear)

Este usuario debe empezar con **C##...** y siempre en mayúsculas.

The 'Crear Usuario' dialog box is shown with the 'Usuario' tab selected. The 'Usuario' field contains 'C##TIEDADINF'. The 'Contraseña Nueva' and 'Confirmar Contraseña' fields are masked with asterisks. The 'Edición Activada' checkbox is checked. The 'Tablespace por defecto' is set to 'USERS' and the 'Tablespace Temporal' is set to 'TEMP'. The 'Aplicar' button is highlighted.

The 'Editar Usuario' dialog box is shown with the 'Roles Otorgados' tab selected. The 'Administrar Todo' button is highlighted. The table below lists various roles and their status.

Nombre del Rol	Otorgado	Administrar	Valor por Defecto
ADM_PARALLEL_EXECUTE_TASK	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
APPLICATION_TRACE_VIEWER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AQ_ADMINISTRATOR_ROLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AQ_USER_ROLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AUDIT_ADMIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AUDIT_VIEWER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
AUTHENTICATEDUSER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CAPTURE_ADMIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CDB_DBA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CONNECT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
CTXAPP	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DATAPATCH_ROLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DATAPUMP_EXP_FULL_DATABASE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DATAPUMP_IMP_FULL_DATABASE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DBA	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DBFS_ROLE	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DBJAVASCRIPT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DBMS_MDX_INTERNAL	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DV_ACCTMGR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
DW_ADMIN	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

CONSULTAS SQL

● CRITERIOS DE NOTACIÓN

Palabras clave de la sintaxis SQL en **MAYÚSCULAS**.

Los “[]” indican **opcionalidad**.

Las “{ }” delimitan **alternativas** separadas por “|” donde solo se puede **elegir una**.

Los puntos suspensivos ... indican **repetición** varias veces de la opción anterior.

Los comentarios de bloque: /* comentario */

Los comentarios de línea: -- comentario

● CONSULTAS

Sintaxis de **SELECT**

```
SELECT * | {[ DISTINCT ] columna | expresión [[AS] alias ], ...}  
FROM nombre_tabla;
```

● CÁLCULOS

Cálculos aritméticos

```
SELECT nombre, precio, precio*1.16 FROM ARTICULOS;
```

Para ponerle un alias a la 3º columna

```
SELECT nombre, precio, precio*1.16 AS "PRECIO + IVA" FROM ARTICULOS;
```

Concatenación

```
SELECT tipo, modelo, tipo || '-' || modelo "Clave Pieza" FROM PIEZAS;
```

● OPERADORES

Operadores de Comparación

<	Menor que	>	Mayor que
<=	Menor o igual que	>=	Mayor o igual que
=		<> ó !=	Distinto

Operadores Lógicos

AND	Verdadero si las dos expresiones, a la izquierda y derecha son verdaderas
OR	Verdadero si una de las expresiones es verdadera
NOT	Invierte el valor de la expresión de la derecha

Ejemplo

```
SELECT nombre, apellidos FROM PERSONAS WHERE edad >= 25 AND edad <= 50;
```

BETWEEN (precio entre 3 y 8) NOT BETWEEN (precio no entre 3 y 8)

```
SELECT tipo, modelo, precio FROM PIEZAS WHERE precio BETWEEN 3 AND 8;
```

```
SELECT tipo, modelo, precio FROM PIEZAS WHERE precio NOT BETWEEN 3 AND 8;
```

IN (precio = 3,5,8) NOT IN (precio != 3,5,8)

```
SELECT tipo, piezas, precio FROM PIEZAS WHERE precio IN (3,5,8);
```

```
SELECT tipo, piezas, precio FROM PIEZAS WHERE precio NOT IN (3,5,8);
```

LIKE

%	Una serie de caracteres
_	Un carácter cualquiera

Ejemplo: % (Selección del nombre de las personas que empiecen por A)

```
SELECT nombre FROM PERSONAS WHERE nombre LIKE 'A%';
```

Ejemplo: _ (Selección del nombre y apellidos cuyo primer apellido sea Jiménez, Giménez o Ximénez)

```
SELECT nombre, apellido1, apellido2 FROM PERSONAS WHERE apellido1 LIKE '_jiménez';
```

IS NULL (es nulo) IS NOT NULL (no es nulo)

```
SELECT nombre, apellido1 FROM PERSONAS WHERE telefono IS NULL;
```

```
SELECT nombre, apellido1 FROM PERSONAS WHERE telefono IS NOT NULL;
```

● SUBCONSULTAS

SINTAXIS

```
SELECT lista expresiones
FROM tablas
WHERE expresión OPERADOR (SELECT lista expresiones FROM tablas);
```

Ejemplo: empleados cuyas pagas sean inferiores a lo que gana Martina.

```
SELECT nombre_empleado, paga
FROM EMPLEADOS
WHERE paga < (SELECT paga FROM EMPLEADOS WHERE nombre_empleado =
'Martina');
```

INSTRUCCIONES ESPECIALES

ANY	Compara con cualquier registro de la subconsulta. La instrucción es válida si hay un registro en la subconsulta que permite que la comparación sea cierta.
ALL	Compara con todos los registros de la consulta. La instrucción resulta cierta si es cierta toda la comparación con los registros de la subconsulta.
IN	No usa comparador, ya que sirve para comprobar si un valor se encuentra en el resultado de la subconsulta.
NOT IN	Comprueba si un valor no se encuentra en la subconsulta.

Ejemplo: (obtener el empleado que más cobra)

```
SELECT nombre, sueldo
FROM EMPLEADOS
WHERE sueldo >= ALL (SELECT sueldo FROM EMPLEADOS);
```

● ORDENACIÓN

SINTAXIS

```
SELECT * | {[DISTINCT] columna | expresión [[AS] alias], ... }
FROM nombre_tabla
[WHERE condición]
[ORDER BY columna1[{ASC|DESC}]][,columna2[{ASC|DESC}]]...;
```

AS

```
SELECT nombre, precio, precio*1.16 AS "Precio+IVA" FROM ARTICULOS;
```

TEMA 4

1. Lenguaje de definición:

- **CHAR(n)** → Cadena de caracteres de longitud fija(n).
- **VARCHAR2(n)** → Cadena de caracteres de longitud variable. Se debe especificar el número de caracteres que tendrá (n).
- **NUMBER(n)** → Dato de tipo numérico de un máximo de 40 dígitos, además del signo y el punto decimal.
- **NUMBER(a,b)** → Números reales. Donde “p” especifica el número total de dígitos (máximo 38 dígitos) y “d” el número total de decimales.
- **DATE** → Para almacenar datos tipo fecha.

2. Creación de bases de datos:

Se creará a partir de la sintaxis:

```
CREATE DATABASE nombre;
```

Para añadir los parámetros se hará de la siguiente forma:

```
CREATE DATABASE nombre  
LOGFILE prueba.log  
MAXLOGFILES 25  
MAXINSTANCES 10  
ARCHIVELOG  
CHARACTER SET AL32UTF8  
NATIONAL CHARACTER SET UTF8  
DATAFILE prueba1.dbf AUTOEXTEND ON MAXSIZE 500MB;
```

FALTARÍA AÑADIR PRIVILEGIOS.

2.1. Eliminación de base de datos:

Para eliminar una base de datos seguiremos la siguiente sintaxis:

```
DROP DATABASE nombre;
```

2.2. Alteración de base de datos:

Para modificar una base de datos seguiremos la siguiente sintaxis:

```
ALTER DATABASE nombre ...;
```

3. Creación de tablas:

Para crear las tablas usaremos **CREATE TABLE** y para añadir columnas:

```
CREATE TABLE nombre_tabla (  
  COLUMNA1 tipo_dato [ restricciones de columna1 ],  
  COLUMNA2 tipo_dato [ restricciones de columna2 ],  
  ...  
  [ restricciones de tabla ]  
);
```

Si queremos añadir restricciones a la tabla usaremos **CONSTRAINT** dentro de la creación de la tabla:

```
{...  
  
CONSTRAINT nombre_constraint  
}
```

Tipos de restricciones:

- **PRIMARY KEY**: establece ese atributo o conjunto de atributos como la clave primaria de la tabla. Esta restricción ya implica las restricciones **UNIQUE** y **NOT NULL**.
- **UNIQUE**: impide que se introduzcan valores repetidos para ese atributo o conjunto de atributos. No se puede utilizar junto con PRIMARY KEY. Se utiliza para claves alternativas.
- **NOT NULL**: evita que se introduzcan filas en la tabla con valor NULL para ese atributo. No se utiliza con PRIMARY KEY.
- **DEFAULT** valor por defecto: permite asignar un valor por defecto al campo que se está definiendo.
- **CHECK** (condición): permite establecer condiciones que deben cumplir los valores de la tabla que se introducirán en dicha columna.
- **FOREIGN KEY**: define una clave externa de la tabla respecto de otra tabla. Esta restricción especifica una columna o una lista de columnas como clave externa de una tabla referenciada. + **REFERENCES nombre_tabla ON DELETE CASCADE**.

4. Creación de vistas:

Para crear vistas usaremos la siguiente sintaxis:

```
CREATE [ OR REPLACE ] VIEW nombre_vista [ (alias1 [, alias2] ...) ]  
AS SELECT ...
```

4.1. Eliminación de vistas:

Para eliminar vistas usaremos la siguiente sintaxis:

```
DROP VIEW nombre_vista;
```

4.2. Alterar vistas:

Para redefinir la vista usaremos la siguiente sintaxis: (también se puede usar replace)

```
ALTER VIEW nombre_vista
```

5. Creación de índices:

Para crear los índices usaremos la siguiente sintaxis:

```
CREATE INDEX nombre  
ON tabla (columna1 [,columna2] ...)
```

5.1. Eliminar índices:

Para eliminar índices usaremos la siguiente sintaxis:

```
DROP INDEX nombre_indice;
```

6. Creación de secuencias:

Para crear secuencias usaremos la siguiente sintaxis:

```
CREATE SEQUENCE secuencia  
[ INCREMENT BY n ]  
[ START WITH n ]  
[ { MAXVALUE n | NOMAXVALUE } ]  
[ { MINVALUE n | NOMINVALUE } ]  
[ { CYCLE | NOCYCLE } ];
```

- **SECUENCIA:** Es el nombre que se le da al objeto de secuencia.
- **INCREMENT BY:** Indica cuánto se incrementa la secuencia cada vez que se usa. Por defecto se incrementa de uno en uno
- **START WITH:** Indica el valor inicial de la secuencia (por defecto 1)
- **MAXVALUE:** Máximo valor que puede tomar la secuencia. Si no se toma NOMAXVALUE que permite llegar hasta 1027
- **MINVALUE:** Mínimo valor que puede tomar la secuencia. Si el incremento es negativo y no se toma NOMINVALUE permite llegar hasta -1026.
- **CYCLE:** Hace que la secuencia vuelva a empezar si se ha llegado al máximo valor.

6.1. Eliminación de secuencias:

Seguiremos la siguiente sintaxis:

```
DROP SEQUENCE nombre_secuencia;
```

6.2. Alteración de secuencias:

Seguiremos la siguiente sintaxis:

```
ALTER SEQUENCE secuencia  
[ INCREMENT BY n ]  
[ START WITH n ]  
[ { MAXVALUE n | NOMAXVALUE } ]  
[ { MINVALUE n | NOMINVALUE } ]  
[ { CYCLE | NOCYCLE } ];
```

7. Expresiones regulares:

Las expresiones regulares deben ir entre los siguientes caracteres: ^...\$

Símbolo	Descripción
*	Se corresponde con cero o más incidencias
	Operador de modificación para especificar correspondencias alternativas
^/\$	Se corresponde con el inicio de línea/fin de línea
[]	Expresión entre corchetes para una lista de correspondencia que se corresponde con cualquiera de las expresiones representadas en la lista
{m}	Se corresponde exactamente <i>m</i> veces
{m,n}	Se corresponde al menos <i>m</i> veces pero no más de <i>n</i> veces
[:]	Especifica una clase de carácter y se corresponde con cualquier carácter de esa clase
\	Puede tener 4 significados diferentes: 1. Se representa a sí mismo. 2. Presenta el siguiente carácter. 3. Introduce un operador. 4. No hace nada.
+	Se corresponde con una o más incidencias
?	Se corresponde con cero o una incidencia
.	Se corresponde con cualquier carácter del juego de caracteres soportado, excepto NULL
()	Expresión de agrupamiento, que se trata como subexpresión única
[=]	Especifica clases de equivalencia
\n	Referencia a expresión anterior
[..]	Especifica un elemento de intercalación como, por ejemplo, un elemento de varios caracteres