

Gestión de bases de datos

Introducción

Acceso a SQLPLUS como sysdba

- Abrir cms
- Acceder a la ruta dbhomeXE y conectarse a sqlplus como sysdba:

```

```
cd C:\app\[USUARIO]\product\[VERSION]\dbhomeXE
sqlplus / as sysdba
```

```

Habilitar scripts

Desde SQLPLUS como sysdba:

```

```
sql> alter session set "_ORACLE_SCRIPT"=true;
```

```

Crear un nuevo usuario

Desde SQLPLUS como sysdba:

```

```
sql> create user [NOMBRE_USUARIO] identified
by [CLAVE_USUARIO]
```

```

Dar permisos a un usuario

Desde SQLPLUS como sysdba:

```

```
sql> grant connect, create session, create table to
[NOMBRE_USUARIO]
```

```

Tipos de datos

- ❑ CHAR / CHAR(n). Cadena de caracteres de ancho fijo (1 - 2000 bytes o caracteres) (n=1 por defecto).
- ❑ VARCHAR2(n). Cadena de caracteres de ancho variable (1 - 4000 bytes o caracteres).
- ❑ NCHAR / NCHAR(n). Cadena Unicode de ancho fijo (1 - 2000 caracteres) (n=1 por defecto).
- ❑ NVARCHAR2(n). Cadena Unicode de ancho variable (1 - 4000 caracteres).

Tipos de datos: NUMBER

NUMBER, NUMBER(p), NUMBER(p,s)

Un número decimal con hasta 38 dígitos.

- ❑ **p (precisión):** El número total máximo de dígitos que se almacenarán. Este número incluye los dígitos a ambos lados de la coma. La precisión debe ser un valor comprendido entre 1 y la precisión máxima de 38. La precisión predeterminada es 18.
- ❑ **s (escala):** El número de dígitos que se almacenarán a la derecha de la coma. Está relacionado con p, por tanto debe ser un valor comprendido entre 0 y p, y solo se puede especificar si se especifica la precisión. La escala predeterminada es 0, por lo que $0 \leq s \leq p$. Los tamaños de almacenamiento máximo varían según la precisión.

Tipos de datos

date

Almacene una fecha en formato dd/mm/aaaa.
Desde el 1 de enero de 0001 hasta el 31 de
diciembre de 9999.

Cómo crear una tabla

```
CREATE TABLE nombre_tabla (  
    columna1 datatype,  
    columna2 datatype,  
    columna3 datatype,  
    ....  
);
```

Cómo modificar una tabla para añadir una nueva columna

```
ALTER TABLE nombre_tabla  
    ADD nuevacolumna datatype;
```

Cómo modificar una tabla para borrar una columna

```
ALTER TABLE nombre_tabla  
    DROP COLUMN columna;
```

Cómo modificar el nombre de una columna en una tabla

```
ALTER TABLE nombretabla  
    RENAME COLUMN nombrecolumna TO  
nuevonombrecolumna;
```

Cómo borrar una tabla

```
DROP TABLE nombre_tabla;
```

Ejercicio 1

Crea una tabla alumnos que tenga como campos nombre (varchar2 de tamaño 15), apellido (varchar2 de tamaño 50) y dni (char tamaño 9).

Ejercicio 2

Modifica la tabla anterior para incluir un nuevo campo llamado código postal que sea del tipo number y tamaño 5.

Ejercicio 3

Ahora modifica nuevamente la tabla anterior alumnos para incluir un nuevo campo llamado fecha_nacimiento que sea del tipo date.

Ejercicio 4

Se quiere borrar el campo código postal, cambiar el nombre de la columna fecha_nacimiento por f_nacimiento, y para terminar, se quiere que apellido sea un varchar2 pero de tamaño 200.

Tipos de restricciones

- ☐ NOT NULL - No es posible el valor nulo.
- ☐ UNIQUE - No puede existir un valor duplicado.
- ☐ PRIMARY KEY - Valor único y no nulo.
- ☐ FOREIGN KEY - Relaciona la columna con una PRIMARY KEY de otra tabla.
- ☐ CHECK - Todos los valores cumple esa condición.
- ☐ DEFAULT - Da un valor predeterminado cuando no se especifica ningún valor.

Cómo añadir PK

Para incluir una Primary Key (PK) en una tabla se puede hacer:

- ☐ Añadir PRIMARY KEY tras el tipo de dato del campo en su definición en CREATE TABLE.
- ☐ Hacer un ALTER TABLE nombretabla ADD PRIMARY KEY (campo o campos que la componen).
- ☐ Hacer un ALTER TABLE nombretabla ADD CONSTRAINT nombreRestricción PRIMARY KEY (campo o campos).
- ☐ Incluir PRIMARY KEY (campo) al finalizar la definición de todos los campos dentro de CREATE TABLE.

Ejercicio 5

Vuelve a crear la tabla alumnos anterior haciendo que el campo dni sea primary key de varias formas diferentes. Usa cuatro tablas alumnos (alumnos1, alumnos2, alumnos3 y alumnos4), con una diferente técnica de implementación de la PK en cada caso.

Cómo añadir FK

Para incluir una Foreign Key (FK) en una tabla se puede hacer:

- ☐ Hacer un ALTER TABLE nombretabla ADD FOREIGN KEY (campo) REFERENCES nombretabladondeesPK(columnaPK).
- ☐ Hacer un ALTER TABLE nombretabla ADD CONSTRAINT nombreRestricción FOREIGN KEY (campo) REFERENCES nombretabladondeesPK(columnaPK).
- ☐ Incluir FOREIGN KEY (campo) REFERENCES nombretabladondeesPK(columnaPK) al finalizar la definición de todos los campos dentro de CREATE TABLE.

Cómo borrar PK o FK

Para eliminar una restricción de una tabla hay que conocer el nombre de la restricción, y luego:

- ❑ Hacer un ALTER TABLE nombretabla DROP CONSTRAINT nombredelarestricción.

Cómo cambiar el nombre a una restricción, como PK o FK

Para cambiar el nombre de una restricción de una tabla hay que conocer el nombre de la restricción, y luego:

- ❑ Hacer un `ALTER TABLE nombretabla RENAME CONSTRAINT nombredelarestricción to nuevonombrerestrccion.`

Ejercicio 6

Crea una tabla localidades que tenga:

- ☐ Id int
- ☐ Localidad varchar2(100)

Haz que id sea la PK de localidades.

Crea un campo id_localidad en tu tabla alumnos.

Añade FK a tu id_localidad de la tabla alumnos que apunte al campo id que es PK en localidades (creamos una FK por tanto). La restricción debe tener el nombre id_localidad_alumnos_fk.

Borra la restricción PK de id en localidades.



CEU

*Centro de Estudios
Profesionales*

Fundación San Pablo Andalucía