

PRÁCTICA: 1. DEFINICIÓN DE DATOS.

1. SOBRE LA PRÁCTICA.....	2
1.1. OBJETIVOS	2
1.2. MATERIAL	2
1.3. CONTENIDOS	2
1.4. BIBLIOGRAFÍA:	2
2. DEFINICIÓN DE DATOS.....	3
2.1. SQL: EL LENGUAJE DE DEFINICIÓN DE DATOS (DDL).....	3
2.2. SENTENCIAS SOBRE CREACIÓN DE TABLAS.....	3
2.2.1. SINTAXIS GENERAL DE LA SENTENCIA CREATE TABLE.....	3
2.2.2. TIPOS DE DATOS	4
2.2.3. RESTRICCIONES DE COLUMNAS.....	4
2.2.4. RESTRICCIONES DE TABLAS	5
2.2.5. EJEMPLOS	5
2.3. RENOMBRAR UNA TABLA.....	6
2.4. ELIMINAR UNA TABLA DE LA BASE DE DATOS.....	6
2.4.1. EJEMPLOS:	6
2.5. MODIFICAR UNA TABLA.....	7
2.5.1. ALTER TABLE <NOMBRE TABLA>	7
2.5.2. EJEMPLOS	7
2.6. SENTENCIAS SOBRE SINÓNIMOS	7
2.6.1. CREAR UN SINÓNIMO.....	7
2.6.2. BORRAR UN SINÓNIMO	8
2.7. GESTIÓN DE DOMINIOS (SI EN SQL – NO EN ORACLE).....	8
2.7.1. CREACIÓN DE DOMINIOS:.....	8
2.7.2. EJEMPLOS:	8
2.7.3. BORRADO DE DOMINIOS:	8
2.7.4. MODIFICADO DE DOMINIOS:	8
3. DICCIONARIO DE DATOS	9
4. EJERCICIOS SOBRE DEFINICIÓN DE DATOS.....	9

1. Sobre la Práctica

1.1. Objetivos

- Enseñar al alumno las sentencias que forman el lenguaje de definición de datos (DDL – Data Definition Language) de SQL, es decir, el subconjunto de órdenes que nos permitirán crear o editar tablas de la base de datos.

1.2. Material

ORACLE versión 9

1.3. Contenidos

- Definición de Datos
 - Creación de Tablas
 - Restricciones de Columnas
 - Restricciones de Tablas
 - Renombrar Tablas
 - Eliminar Tablas
 - Modificar Tablas
 - Crear y Borrar Sinónimos
 - Crear Dominios
 - Ejercicios

1.4. Bibliografía:

- Básica
 - Oracle 8, Guía de aprendizaje. Michael Abbey, Michael Corey, McGraw-Hill 1997.
 - Guía de SQL, James R. Groff, Paul N. Weinberg, McGraw-Hill, 1998.
- Complementaria
 - La biblia de Oracle 8. Ediciones Anaya Multimedia, 1998.
 - Understanding the new SQL: A Complete Guide, Jim Melton, Alan R. Simon, Morgan Kaufmann Publisher, 1993.
 - A Guide to SQL Standard (fourth edition). Date, C.J. and Darwen, H. Addison-Wesley, 1997.
 - SQL. Manual de Referencia del Programador. Freeze, W.S. Paraninfo, 1998.

2. Definición de Datos

2.1. SQL: El lenguaje de definición de datos (DDL)

El lenguaje de definición de datos permite:

- Definir y crear una nueva tabla.
- Suprimir una tabla que ya no se necesita.
- Cambiar la definición de una tabla existente.
- Definir una tabla virtual (o vista) de datos.
- Construir un índice para hacer más rápido el acceso a una tabla.
- Controlar el almacenamiento físico de los datos por parte del SGBD.

2.2. Sentencias sobre Creación de Tablas

2.2.1. Sintaxis General de la sentencia CREATE TABLE.

```
CREATE TABLE <nombre de tabla>
(nombre_columna1 tipo [restricción de columna],
.....
nombre_columnaN tipo [restricción de columna],
[restricción_de_tabla]);
```

La sentencia **CREATE TABLE** se utiliza para crear una tabla dentro de la cual habrá columnas que contienen datos y restricciones

2.2.1.1. Más en detalle...

```
CREATE TABLE < nombre tabla >
( < nombre columna > < tipo de dato >
[NOT NULL] [UNIQUE] [CONSTRAINT < nombre restricción >][PRIMARY KEY]
[REFERENCES][DEFAULT][CHECK
| [PRIMARY KEY (< lista columnas >)]
| [FOREIGN KEY (< lista columnas >) REFERENCES (nombretabla)]
| [UNIQUE (< lista columnas >)] [CONSTRAINT < nombre
restricción > ],[,...] )
| [CHECK (condición de búsqueda)]
```

2.2.2. Tipos de datos

Tipo de dato	Descripción
Char(tamaño)	Almacena datos de tipo carácter de longitud fija, con un máximo de 2000 caracteres)
varchar2(tamaño)	Almacena datos de tipo carácter de longitud variable, con un tamaño máximo de 4000
Varchar	Actualmente es igual que char
Long	Almacena datos de tipo carácter de longitud variable, hasta 2 gigabytes. Solo se permite un Long por tabla. Una columna de tipo Long no puede utilizarse como parte de un índice. Una función almacenada no puede devolver un Long. Las cláusulas Where, Group By, Order By, Unique, o Connect By no pueden referenciar a una columna Long.
Blob	Es un objeto binario de gran tamaño, siendo el tamaño máximo 4 GB (gigabytes). Normalmente un blob se utiliza para almacenar una imagen, datos de voz, o cualquier otro bloque de datos grande no estructurado.
Date	Almacena fechas desde el 1 de enero del 4712 a.C. hasta el 31 de diciembre del 4712 d.C.
Integer	Un número entero que no tiene parte fraccionaria. Normalmente un Integer será un valor de 32 bits con un rango de -2147483648 a +2147483647
Smallint	Representa un número entero que no contiene parte fraccionaria. Su precisión nunca será mayor que la de un Integer. Es un valor de 16 bits entre -32768 y +32767
Number(1,d)	Almacena datos de tipo numérico, siendo “1” la longitud y “d” el número de dígitos decimales
Raw(tamaño)	Datos binarios puros con una longitud máxima de 2000 bytes. Sirven para almacenar datos de tipo binario como sonido e imágenes digitalizadas.

2.2.3. Restricciones de columnas

NOT NULL. La columna no permitirá valores nulos.

CONSTRAINT. Permite asociar un nombre a una restricción.

DEFAULT valor. La columna tendrá un valor por defecto. El SBGD utiliza este valor cuando no se especifica un valor para dicha columna.

PRIMARY KEY. Permite indicar que esta columna es la clave primaria.

REFERENCES. Es la manera de indicar que este campo, es clave ajena y hace referencia a una clave candidata de otra tabla. Esta **foreign key** es sólo de una columna.

UNIQUE. Obliga a que los valores de una columna tomen valores únicos (no puede haber dos filas con igual valor). Se implementa creando un índice para dicha(s) columna(s).

CHECK (condición). Permite indicar una condición que debe de cumplir esa columna.

2.2.4. Restricciones de tablas

PRIMARY KEY (columna1, columna2...). Permite indicar las columnas que forman la clave primaria.

FOREIGN KEY (columna1, columna2....) REFERENCES NombreTabla. Indica las columnas que son clave ajena referenciando a una clave candidata de otra tabla.

UNIQUE (columna1, columna2...). El valor combinado de una o varias columnas es único.

CHECK (condición). Permite indicar una condición que deben cumplir las filas de la tabla. Puede afectar a varias columnas.

La cláusula **Foreign Key** tiene unas opciones que se explican a continuación (no soportadas en su totalidad por Oracle):

- Tratamiento de nulos: Se puede indicar cómo debe tratar el SGBD un valor NULL en una o más columnas de la clave ajena, cuando lo compare con las filas de la tabla padre.
- Modo de borrado: Para determinar la acción que se debe realizar cuando se elimina una fila referenciada, se debe utilizar una regla de supresión opcional para la relación (CASCADE, SET NULL, SET DEFAULT, NO ACTION).
- Modo de modificación: Una regla de actualización para la relación, que determina la acción que se debe realizar cuando se modifica la clave candidata de la fila referenciada (CASCADE, SET NULL, SET DEFAULT, NO ACTION).

2.2.5. EJEMPLOS

areas(código, nombre, departamento) (código es la clave primaria)

departamentos(código_dpto, nombre) (código_dpto es la clave primaria)

La tabla areas tiene una clave ajena

areas.departamento → *departamentos*

```
CREATE TABLE areas
(
  codigo char(3) not null,
  nombre char(55) not null,
  departamento char(3) not null,
  Primary key(codigo),
  Foreign key(departamento) REFERENCES departamentos
  ON DELETE SET NULL ON UPDATE CASCADE);
```

ON DELETE Set Null Significa que si se borra algún departamento de la tabla departamentos el campo departamento de las filas de la tabla areas que le referenciaban se pone como Null.

ON UPDATE CASCADE Significa que si se modifica el código_dpto de una fila de la tabla departamento, también se modificara en las filas de la tabla áreas que le referencian.

```
CREATE TABLE departamentos
(
  codigo_dpto char(3) not null,
  nombre char(40) not null,
  Primary key(codigo_dpto)
);
```

2.3. Renombrar una tabla

```
RENAME TABLE <nombre tabla existente> TO <nuevo nombre tabla>
```

2.4. Eliminar una tabla de la base de datos

```
DROP TABLE <nombre tabla> [CASCADE, RESTRICT]
```

2.4.1. Ejemplos:

- **DROP TABLE DEPARTAMENTOS CASCADE**

(La tabla se borra, así como las posibles restricciones relativas a esta tabla)

- **DROP TABLE DEPARTAMENTOS RESTRICT**

(La tabla se borra sólo si no se hace referencia a ella en ninguna restricción, p.e. en la definición de claves ajenas)

2.5. Modificar una tabla

2.5.1. ALTER TABLE <nombre tabla>

La sentencia **Alter Table** se utiliza para cambiar una tabla existente. Dentro de la tabla podemos **Add** (añadir) o **Drop** (borrar) columnas y restricciones (PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK CONSTRAINT).

```
{ ADD <nombre columna nueva> <tipo de dato> [NOT NULL]
  MODIFY <nombre columna> [DEFAULT | DROP DEFAULT] valor
  DROP <nombre columna> [CASCADE | RESTRICT]
  ADD [PRIMARY KEY (nombre columna) |
  FOREIGN KEY (nombre columna) REFERENCES nombre_tabla | UNIQUE (nombre
  columna) | CHECK (condición)
  DROP CONSTRAINT nombre-restricción [CASCADE| RESTRICT]
```

2.5.2. EJEMPLOS

Agregar a la tabla areas el campo Responsable de tipo char(30)

```
alter table areas
  ADD responsable char(30) not null;
```

Modificar el campo nombre de la tabla departamentos a char(50)

```
alter table departamentos
  MODIFY nombre char(50);
```

2.6. Sentencias sobre Sinónimos

Un sinónimo es un nombre que puede utilizarse como sustituto o alias del nombre real de una tabla. Puede ser útil para simplificar algunas expresiones.

2.6.1. Crear un sinónimo

```
CREATE SYNONYM <nombre sinónimo> FOR <nombre tabla>
```

```
create synonym are for areas;
```

```
create synonym dep for departamentos;
```

2.6.2. Borrar un sinónimo

```
DROP SYNONYM <nombre sinónimo>
```

Drop synonym are

2.7. Gestión de Dominios (SI en SQL – NO en ORACLE)

Los dominios se usan como tipos de datos en la definición de columnas al crear tablas. Permiten definir los valores aceptados en las columnas, así como la definición de valores por defecto.

2.7.1. Creación de Dominios:

```
CREATE DOMAIN <Nombre Dominio> AS <Tipo Datos>  
[DEFAULT <valor defecto>]  
[{{CONSTRAINT <nombre restricción>} CHECK (<condición check>)}...];
```

2.7.2. Ejemplos:

```
CREATE DOMAIN SeisDigitos AS CHAR(6) DEFAULT '000000';
```

```
CREATE DOMAIN SeisDigitos AS CHAR(6) DEFAULT '000000' CHECK (VALUE IS  
NOT NULL) CHECK (CHAR_LENGTH(TRIM(VALUE)) = 6 AND VALUE  
BETWEEN '000000' AND '500000');
```

2.7.3. Borrado de Dominios:

```
DROP DOMAIN <Nombre Dominio>
```

2.7.4. Modificado de Dominios:

```
ALTER DOMAIN <Nombre Dominio> { SET DEFAULT <Valor Defecto | DROP  
DEFAULT }  
ALTER DOMAIN <Nombre Dominio> ADD <Restricción Dominio>  
ALTER DOMAIN <Nombre Dominio> DROP CONSTRAINT <Nombre Restricción>  
[RESTRICT | CASCADE ]
```

3. Diccionario de Datos

Conjunto de tablas que contiene las definiciones de los elementos de las bases de datos creadas por el usuario. Estas tablas son creadas y actualizadas directamente por el SGBD. Las tablas del diccionario de datos se pueden consultar con SELECTs

SELECT TABLE_NAME FROM USER_TABLES

Obtenemos el nombre de las tablas existente en la base de datos.

DESCRIBE nombre_tabla | DESC nombre_tabla

Ofrece la descripción completa de una tabla existente en la base de datos.

4. Ejercicios sobre Definición de datos

- Utilizando SQL*Plus crear las tablas correspondientes al siguiente esquema de base de dato relacional relativo a la gestión de los préstamos de una biblioteca:

Libro (signatura, autor, titulo, editor, clase)

Usuario (carnet, nombre, direccion)

Clase (clave, tiempo_de_prestamo)

Prestamo (signatura, carnet, fecha_inicio, fecha_fin)

Con las siguientes claves ajenas:

Libro.clase → Clase

Prestamo.signatura → Libro

Prestamo.carnet → Usuario

- Agregar a la tabla Usuario el campo Fecha_Ingreso (que sea obligatorio)
- Agregar a la tabla Libro el campo Prestado (que sea obligatorio), y asignarle por defecto el valor 1.
- Definir el dominio “TIPO_EDAD”, como un entero entre 0 y 120, y añadir el atributo Edad del usuario
- Al finalizar borrar las tablas creadas.