



Fundamento Bases de Datos
Facultad de Ciencias UNAM
M.I. Gerardo Avilés Rosas
gar@ciencias.unam.mx
Laboratorio:
Efraín Hipólito Chamú <*chamugauss@gmail.com*>

PRACTICA 05

Lenguaje de Definición de Datos (DDL)

1. Introducción

Hasta este momento se cuenta con una representación del modelo lógico de la base de datos, ya sea con un diagrama de clases de UML o con un diagrama Entidad – Relación, sin embargo no se ha instrumentado el diseño en un SMBD.

Para ello, se utilizará SQL que es el lenguaje estándar en cuanto a definición y manipulación de bases de datos, uno de los primeros lenguajes comerciales y que fue propuesto por Donald Chamberlin y Raymond Boyce. Por otra parte el modelo relacional creado por Edgar F. Codd y que él mismo describió como "Un modelo relacional de datos para grandes bancos de datos compartidos". A pesar de no apegarse íntegramente al modelo relacional tal como lo describe Codd , SQL se convirtió en el lenguaje de bases de datos más utilizado.

SQL se convirtió en un estándar de la American National Standards Institute (ANSI) en 1986 y de la Organización Internacional de Normalización (ISO) en 1987. Desde entonces, el estándar se ha mejorado varias veces con características añadidas pero aún no es completamente transferible entre sistemas de bases de datos diferentes, es decir, cada SMBD tiene una combinación específica del código, ya que los diferentes fabricantes no siguen estrictamente el estándar al añadir extensiones.

El alcance de SQL incluye aspectos como la inserción de datos, consulta, actualización y supresión, creación y modificación de esquemas, y el control de acceso a datos.

Dado que se ha elegido Oracle Database en su versión 11g o 12c, con los comandos básicos del estándar se podrá materializar el modelo que se ha diseñado hasta ahora. Para ésta práctica revisaremos la parte correspondiente a la definición de datos, es decir el DDL, donde se define la estructura de la base de datos y creación de tablas. Y solo se hará la definición de las tablas, atributos y sus tipos de datos.

2. DDL en Oracle.

A. Creación de tablas.

Por el momento sólo interesa la traducción del diagrama lógico a tablas. Los temas de integridad, normalización y optimización se tratan en prácticas posteriores.

Para creación de una tabla se define como sigue, figura 1.

```
CREATE TABLE Cliente(  
    Nombre_Cliente VARCHAR(25),  
    Direccion VARCHAR(25),  
    Telefono INTEGER,  
    Fecha_Nacimiento DATE,  
    Edad INTEGER  
)
```

Figura 1.

Si se desea hacer modificación de columnas en la tabla, ver figura 2:

```
ALTER TABLE Cliente RENAME COLUMN Telefono TO Tel;
```

Figura 2.

Para agregar una nueva columna y definir su tipo de dato, se define como sigue, figura 3.

```
ALTER TABLE Cliente ADD Estado_Nacimiento VARCHAR(20);
```

Figura 3.

Si se requiere borrar una columna, figura 4.

```
ALTER TABLE Cliente DROP COLUMN Estado_Nacimiento;
```

Figura 4.

Para la modificación de columnas como creación, renombre o borrado, cabe aclarar que no afectan el estado de la BD porque no se tiene ningún dato almacenado hasta el momento, pero peligros potenciales de estos cambios en caso de que existan datos, Oracle los rechaza, además de tener en cuenta el tipo de rol que está conectado a la base ya que posibles cambios como borrado de columnas no se permitirán.

B. Tipos de datos.

A Continuación se presentan algunos tipos de datos que utiliza Oracle y que puede servir de guía para el esquema que se propone de acuerdo al diagrama de clases, practica 04.

Algunos tipos de datos soportados por Oracle se mencionan a continuación, tabla 1.

TIPO	CARACTERÍSTICAS	OBSERVACIONES
CHAR	Cadena de caracteres (alfanuméricos) de longitud fija	Entre 1 y 2000 bytes como máximo. Aunque se introduzca un valor más corto que el indicado en el tamaño, se rellenará al tamaño indicado. Es de longitud fija, siempre ocupará lo mismo, independientemente del valor que contenga
VARCHAR2	Cadena de caracteres de longitud variable	Entre 1 y 4000 bytes como máximo. El tamaño del campo dependerá del valor que contenga, es de longitud variable.
VARCHAR	Cadena de caracteres de longitud variable	En desuso, se utiliza VARCHAR2 en su lugar
NCHAR	Cadena de caracteres de longitud fija que sólo almacena caracteres Unicode	Entre 1 y 2000 bytes como máximo. El juego de caracteres del tipo de datos (datatype) NCHAR sólo puede ser AL16UTF16 ó UTF8. El juego de caracteres se especifica cuando se crea la base de datos Oracle
NVARCHAR2	Cadena de caracteres de longitud variable que sólo almacena caracteres Unicode	Entre 1 y 4000 bytes como máximo. El juego de caracteres del tipo de datos (datatype) NCHAR sólo puede ser AL16UTF16 ó UTF8. El juego de caracteres se especifica cuando se crea la base de datos Oracle
NCLOB	Almacena datos de tipo carácter	Admiten hasta 8 terabytes. Guarda los datos según el juego de caracteres Unicode nacional.
NUMBER	Almacena números fijos y en punto flotante	Se admiten hasta 38 dígitos de precisión y son portables a cualquier entre los diversos sistemas en que funcione Oracle. Para declarar un tipo de datos NUMBER en un CREATE ó UPDATE es suficiente con: nombre_columna NUMBER

FLOAT	Almacena tipos de datos numéricos en punto flotante	Es un tipo NUMBER que sólo almacena números en punto flotante
DATE	Almacena un punto en el tiempo (fecha y hora)	El tipo de datos DATE almacena el año (incluyendo el siglo), el mes, el día, las horas, los minutos y los segundos (después de medianoche). Oracle utiliza su propio formato interno para almacenar fechas.
TIMESTAMP	Almacena datos de tipo hora, fraccionando los segundos	
TIMESTAMP WITH TIME ZONE	Almacena datos de tipo hora incluyendo la zona horaria (explícita), fraccionando los segundos	
TIMESTAMP WITH LOCAL TIME ZONE	Almacena datos de tipo hora incluyendo la zona horaria local (relativa), fraccionando los segundos	Cuando se usa un SELECT para mostrar los datos de este tipo, el valor de la hora será ajustado a la zona horaria de la sesión actual
INTEGER	Almacena datos de tipo Decimal	Este tipo de datos es un alias del tipo de datos NUMBER (38), y se ha diseñado para que OracleDataReader devuelva System.Decimal o OracleNumber en lugar de un valor entero. El uso del tipo de datos de .NET Framework puede ocasionar un desbordamiento.

Tabla 1.

Si se requiere usar más tipos de datos, consultar:

https://docs.oracle.com/database/121/SQLRF/sql_elements001.htm#SQLRF0021

C. Creación de una base de datos.

Para la creación de una base de datos en Oracle manualmente, es indispensable crear variables, directorios, archivos personalizados de tamaño de almacenamiento, servicios y la instancia de la nueva base de datos. Si se requiere usar el asistente de configuración de Oracle para la creación de una base de datos, solo se deben seguir los pasos del asistente.

Cabe mencionar que Oracle no hace una creación de una base “limpia”, es decir incluye tablas para su configuración, además de que el tiempo para crear la instancia de la base es demasiado.

Para fines prácticos se utilizará la base ya instanciada vista en la práctica 01 y como ejemplo básico de creación de una base de datos se debe definir como sigue, teniendo en cuenta las configuraciones necesarias para su creación, figura 2.

```
Conectado a:  
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production  
SQL> CREATE DATABASE BASEFBD;
```

Figura 2.

Entregables

- Cree un script .sql con el esquema de tablas apoyándose en el esquema gráfico del análisis presentado en la práctica 04 dentro de la base de la Practica01.
- Crea un reporte en PDF donde se justifique la elección del tipo de dato en cada caso del punto anterior.
- Crea un script .sql que permita borrar todas las tablas en caso de que se requieran eliminar.
- Deberán enviar su reporte PDF y sus scripts a chamugauss@gmail.com siguiendo los lineamientos establecidos para la entrega de prácticas.
- **Fecha de entrega domingo 17 de septiembre a las 23:59 hrs.**

Nota: Recuerda que no se debe crear ningún tipo de restricción y/o de llave.