

## **Unidad II: Lenguaje de definición de datos (DDL)**

### **2.1 Creación del esquema de la base de datos**

El esquema de una base de datos (en inglés, Database Schema) describe la estructura de una Base de datos, en un lenguaje formal soportado por un Sistema administrador de Base de datos (DBMS). En una Base de datos Relacional, el Esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla.

El esquema es generalmente almacenado en un Diccionario de Datos. Aunque generalmente el esquema es definido en un lenguaje de Base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de base de datos (Diseño de lógico de la base de datos).

Generalmente en la práctica el término esquema de la base de datos se refiere al diseño físico de la base de datos.

Oracle generalmente asocia un 'username' como esquemas en este caso SYSTEM y HR (Recursos humanos).

Por otro lado MySQL presenta dos esquemas information\_schema y MySQL ambos guardan información sobre privilegios y procedimientos del gestor y no deben ser eliminados.

Sintaxis básica para crear una base de datos en Oracle ( No aplica en Oracle express)

```
CREATE DATABASE nombre_baseDatos;
```

Sintaxis básica para crear una base de datos en MySQL

```
CREATE DATABASE IF NOT EXISTS nombre_baseDatos;
```

Para conocer las bases datos creadas use

```
SHOW DATABASES;
```

## 2.2 Actualización, modificación y eliminación del esquema de la base de datos.

### Oracle

Una tabla es un sistema de elementos de datos (atributo - valores) que se organizan que usando un modelo vertical - columnas (que son identificados por su nombre)- y horizontal filas. Una tabla tiene un número específico de columnas, pero puede tener cualquier número de filas. Cada fila es identificada por los valores que aparecen en un subconjunto particular de la columna que se ha identificado por una llave primaria.

Una tabla de una base de datos es similar en apariencia a una hoja de cálculo, en cuanto a que los datos se almacenan en filas y columnas. Como consecuencia, normalmente es bastante fácil importar una hoja de cálculo en una tabla de una base de datos. La principal diferencia entre almacenar los datos en una hoja de cálculo y hacerlo en una base de datos es la forma de organizarse los datos.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	idTorneo	Torneo	Fase	Jorna	idLoca	Local	gLocal	gVisitante	idVisitante	Visitante	Fecha	Hora	Estadio
1464	1	Clausura 2011	Regular	11	14	UNAM	0	0	12	Pachuca	2011-03-20	12:00	Olímpico
1465	5	Clausura 2009	Final	22	14	UNAM	1	0	12	Pachuca	2009-05-28	12:00	Olímpico
1466	6	Apertura 2008	Regular	5	14	UNAM	3	1	12	Pachuca	2008-08-24	12:00	Olímpico
1467	4	Apertura 2009	Regular	8	14	UNAM	1	2	12	Pachuca	2009-09-13	12:00	Olímpico
1468	7	Clausura 2008	Regular	17	14	UNAM	0	0	12	Pachuca	2008-05-04	12:00	Olímpico
1469	2	Apertura 2010	Regular	5	14	UNAM	4	1	13	Puebla	2010-08-22	12:00	Olímpico
1470	1	Clausura 2011	Regular	5	14	UNAM	0	1	13	Puebla	2011-02-06	12:00	Cuauhtém
1471	3	Bicentenario 2010	Regular	9	14	UNAM	4	1	13	Puebla	2010-03-07	12:00	Olímpico
1472	5	Clausura 2009	Semifinales	21	14	UNAM	1	2	13	Puebla	2009-05-23	12:00	Olímpico
1473	5	Clausura 2009	Regular	13	14	UNAM	3	1	13	Puebla	2009-04-12	12:00	Olímpico
1474	1	Clausura 2011	Regular	15	14	UNAM	3	0	7	Queretaro	2011-04-17	12:00	Olímpico
1475	4	Apertura 2009	Regular	6	14	UNAM	3	0	7	Querétaro	2009-08-30	12:00	Olímpico
1476	2	Apertura 2010	Regular	7	14	UNAM	0	1	15	San Luis	2010-09-12	12:00	Olímpico
1477	86	Apertura 2011	Regular	1	14	UNAM	2	0	15	San Luis	2011-07-24	12:00	Olímpico
1478	3	Bicentenario 2010	Regular	12	14	UNAM	0	0	15	San Luis	2010-03-28	12:00	Olímpico
1479	6	Apertura 2008	Regular	9	14	UNAM	0	1	15	San Luis	2008-09-21	12:00	Olímpico
1480	7	Clausura 2008	Regular	13	14	UNAM	2	3	15	San Luis	2008-04-06	12:00	Ciudad Uni

Por lo tanto, la creación de las tablas en el proceso de programación en Oracle juegan un papel muy importante. En el momento de crear las tablas se definen características a dos niveles: Tabla y Columna, como se muestra a continuación:

**A nivel de tabla:** Refieren a una o a varias columnas, donde cada columna se define individualmente.

<b>Nombre:</b>	Nombre de la tabla puede ser de 1 a 30 caracteres. La tabla tiene como propietario al usuario que las crea. Por ejemplo <b>EQUIPO</b> . Hay que tener en cuenta también ciertas restricciones con los nombres de las tablas: longitud máxima de 30 caracteres, no puede haber nombres de tabla duplicados, deben comenzar con un carácter alfabético, permitir caracteres alfanuméricos y el guion bajo '_', y Oracle no distingue entre mayúsculas y minúsculas.
<b>Propietario:</b>	La tabla tiene como propietario al usuario que las crea En nuestro

	caso somos el usuario <b>ALUMNO</b> . Otro usuario que desee usar nuestras tablas debe tener autorización para ello y hacer referencia a la tabla como <b>ALUMNO.EQUIPO (propietario.tabla)</b>
<b>Cantidad de Columnas:</b>	Una tabla puede tener un máximo de 254 columnas.

A **nivel de Columna** el nombre de la columna puede tener un máximo de 30 caracteres.

En Oracle podemos implementar diversos tipos de tablas. A continuación se presenta una recompilación no exhaustiva de ellas.

<b>Tipo Tabla</b>	<b>Descripción</b>
<b>Regular (heap)</b>	Son el mecanismo de almacenamiento de los datos en una base de datos Oracle. Contienen un conjunto fijo de columnas. Las columnas de una tabla describen los atributos de la entidad que se representa con la tabla. Cada columna tiene un nombre y características específicas: tipo de dato y longitud, restricciones, etc.
<b>Clustered</b>	Un cluster proporciona un método opcional de almacenar datos de tabla. Un cluster está compuesto de un grupo de tablas que comparten los mismos bloques de datos. Las tablas son agrupadas mediante columnas comunes.
<b>Index</b>	Aquí una tabla es almacenada en la estructura de un índice. Esto impone orden físico a las filas por si mismas. A diferencia de un heap, donde los datos son almacenados en donde caben, en una tabla IOT (Tabla Organizada por Indices) los datos son almacenados en el orden de la clave primaria.
<b>Particionadas</b>	Es un esquema de organización de los datos con el cual podemos dividirla en múltiples objetos de almacenamientos llamados particiones de datos o rangos, dependiendo los valores puede ser dividido en uno o más columnas de la tabla. Cada particiones de datos es almacenado separadamente. Estos objetos almacenados pueden estar en diferentes tablespaces, en el mismo o en una combinación de ambos.
<b>Temporales</b>	Son tablas cuyos datos permanecerán en el sistema sólo

Tipo Tabla	Descripción
	durante el tiempo que dure la transacción o sesión involucrada. No obstante, al igual que para las tablas permanentes, la definición de las tablas temporales se almacena en las tablas del sistema.

La sintaxis del comando que permite crear un tabla es la siguiente:

```
CREATE TABLE [esquema.]tabla (
  {columna tipocolumna [NOT NULL], }+,
  {CONSTRAINT nombreRestricción
  {UNIQUE ([column, ]+)|}
  DEFAULT expresion|
  CHECK (condicion)|}
  PRIMARY KEY ([column, ]+)|}
  FOREIGN KEY (column) REFERENCES tabla(columna)} , }*
);
```

Del examen de la sintaxis de la sentencia **Create Table** se pueden concluir que necesitamos conocer los distintos tipos de columna y las distintas restricciones que se pueden imponer al contenido de las columnas.

Existen varios tipos de datos en SQL. De esta manera, cada columna puede albergar una información de naturaleza distinta. Los tipos de datos más comunes y sus características en Oracle Express (10 Y 11g) se resumen en la siguiente tabla. Las versiones de Oracle comercial soportan una gama mucho más amplia de tipos de datos.

Tipo de Dato	Descripción
<b>BLOB</b>	Contiene datos binarios con un tamaño máximo de 4 gigabytes. Los datos binarios nos van a permitir guardar en la base de datos archivos, imágenes, sonidos, etc ...  Casi siempre es preferible guardar la ruta del archivo en la base de datos en lugar del propio archivo en modo binario, pero existen ciertas circunstancias en las que no nos queda

	otra solución.
<b>BINARY_DOUBLE</b>	Presición doble
<b>BINARY_FLOAT</b>	Presición simple
<b>CLOB</b>	<p>Un tipo de datos <b>CLOB</b> de Oracle contiene datos de caracteres basados en el juego de caracteres predeterminados del servidor. Su tamaño máximo es de 4 gigabytes. Se asigna a <b>cadena</b>.</p> <p>Use la siguiente expresión para una consulta de un campo CLOB</p> <pre>SELECT DBMS_LOB.substr(campo, DBMS_LOB.getlength(campo),1) FROM tablaprueba;</pre>
<b>CHAR</b>	Almacena datos de tipo carácter alfanumérico de longitud fija, con un tamaño máximo de 2000. caracteres
<b>DATE</b>	Almacena fechas desde el 1-Ene-4712 AC hasta el 31-Dic-4712 DC.
<b>NUMBER(dig dec)</b>	Datos numéricos de <b>n</b> dígitos, de los cuales <b>dec</b> son decimales. El tamaño máximo es de 38 dígitos.
<b>NVARCHAR2</b>	Almacena un valor alfanumérico de longitud variable en caracteres Unicode con las mismas restricciones de varchar.
<b>TIMESTAMP</b>	Fecha y hora (incluidos los segundos), con un tamaño que abarca desde 7 a 11 bytes.
<b>VARCHAR2(tamaño)</b>	Guarda datos de tipo carácter alfanumérico de longitud variable, con un tamaño máximo de 4,000 caracteres.

**Ejemplo:** Considera la siguiente tabla de datos correspondientes a los campeones de Formula 1 (1950 - 2012) y sus escuderías. Y su traducción a sentencias Oracle.

Año	Campeón	Escudería
2012	-	-
2011	Sebastian Vettel	Red Bull Racing

Año	Campeón	Escudería
2010	Sebastian Vettel	Red Bull Racing
2009	Jenson Button	Brawn GP
2008	Lewis Hamilton	McLaren
2007	Kimi Raikkonen	Ferrari
2006	Fernando Alonso	Renault
2005	Fernando Alonso	Renault
2004	Michael Schumacher	Ferrari
2003	Michael Schumacher	Ferrari
2002	Michael Schumacher	Ferrari
2001	Michael Schumacher	Ferrari
2000	Michael Schumacher	Ferrari

```
CREATE TABLE f1 (
    year      INTEGER PRIMARY KEY ,
    campeon   CHAR(30) ,
    escuderia CHAR(20)
);
```

**Ejemplo:** Estados, capitales, densidad de población y superficie de la Republica Mexicana

```
CREATE TABLE estados (
    idEstado      INTEGER PRIMARY KEY ,
    nombreEstado  CHAR(25) NOT NULL ,
    capital       CHAR(25) NOT NULL ,
    densidad      INTEGER NOT NULL ,
    poblacion     INTEGER NOT NULL
);
```

## Tablas Temporales

Oracle permite la creación de tablas temporales para mantener datos propios y exclusivos a una sesión Oracle determinada. Estos datos permanecerán en el sistema sólo durante el tiempo que dure la transacción o sesión involucrada. No

obstante, al igual que para las tablas permanentes, la definición de las tablas temporales se almacena en las tablas del sistema.

La siguiente sintaxis permite crear una tabla temporal personal para cada sesión. Eso significa que los datos no se comparten entre sesiones y se eliminan al final de la misma.

```
CREATE GLOBAL TEMPORARY TABLE [ schema. ]table (
    nombreColumna tipodato [DEFAULT expresión] [NOT NULL],
    [,nombre_columna tipo_dato [DEFAULT expresión]
    [restricción_columna] ...
    [restricción_tabla];
    ON COMMIT { DELETE | PRESERVE } ROWS ]
    [ physical_properties ]
```

Con la opción **ON COMMIT DELETE ROWS** se borran los datos cada vez que se hace **COMMIT** en la sesión.

Con la opción **ON PRESERVE DELETE ROWS** los datos no se borran hasta el final de la sesión.

Sus ventajas son varias, la información contenida en ella está solo disponible para la sesión actual, cualquier **inserción, borrado, actualización** solo se refleja en la sesión activa.

Muchas funcionalidades de cualquier tabla normal se mantienen en ella, como triggers a nivel tabla, vistas, índices, exportar e importar (claro solo la definición de la tabla).

No es posible declarar llaves foráneas en una tabla temporal.



## (DROP) Eliminación

Cuando una tabla ya no es útil y no vamos a volver a necesitarla debe ser borrada. Esta operación se puede realizar con el comando **DROPTABLE**.

**DROP TABLE nombre\_tabla [CASCADE CONSTRAINTS][PURGE]**

Se borra la tabla de la base de datos, borrando toda la información contenida en la tabla, es decir, todas las filas. También se borrará toda la información que sobre la tabla existiera en el diccionario.

Si alguna columna de la tabla a borrar sirve como clave ajena de alguna tabla detalle, impide la eliminación de la tabla, ya que existe una restricción que requiere de la existencia de la tabla maestra. Esto se puede arreglar colocando la sentencia **CASCADE CONSTRAINTS**.

Esto produce que las restricciones de la tabla detalle se borren antes de borrar la tabla maestra. **PURGE** evita que los objetos borrados se vayan a la papelera

La siguiente sentencia produce la eliminación de la tabla BORRAME.

```
drop table BORRAME;  
Resultados Explicar Describir SQL Guardado Historial  
Tabla borrada.
```

## Modificación

Oracle permite modificar las restricciones definidas para una tabla. Esto puede llevar a “**Inconsistencia**” de los datos ya introducidos en la base de datos. Por ello, Oracle tiene definidos mecanismos para modificación de los datos ya existentes.

Esta operación se puede realizar con el comando **ALTER TABLE**.

```
ALTER TABLE [esquema.]tabla  
clausula_constraint [...]  
[ENABLE clausula_activa | DISABLE clausula_disable]  
[{ENABLE|DISABLE} TABLE [LOCK]  
[{ENABLE|DISABLE} ALL TRIGGERS];
```