

ANEXO AL TEMA 6: CONCEPTOS BÁSICOS SOBRE SQL. INTRODUCCIÓN. ESQUEMA CON EL QUE TRABAJAREMOS.

TRABAJANDO CON SQL

Cuando instalamos el Oracle Express te instala por defecto una base de datos llamada XE. Cuando creamos un usuario (en nuestro caso: clase/clase) tiene su esquema dentro de esa base de datos.

Comandos básicos conexión:

- Línea de comandos de MS-DOS: sqlplus
- Línea de comandos de SQL: connect clase/clase

Si no podemos conectar con la base de datos, fijarnos en servicios (herramientas administrativas/servicios) que tenemos el oracleserviceXE levantado.

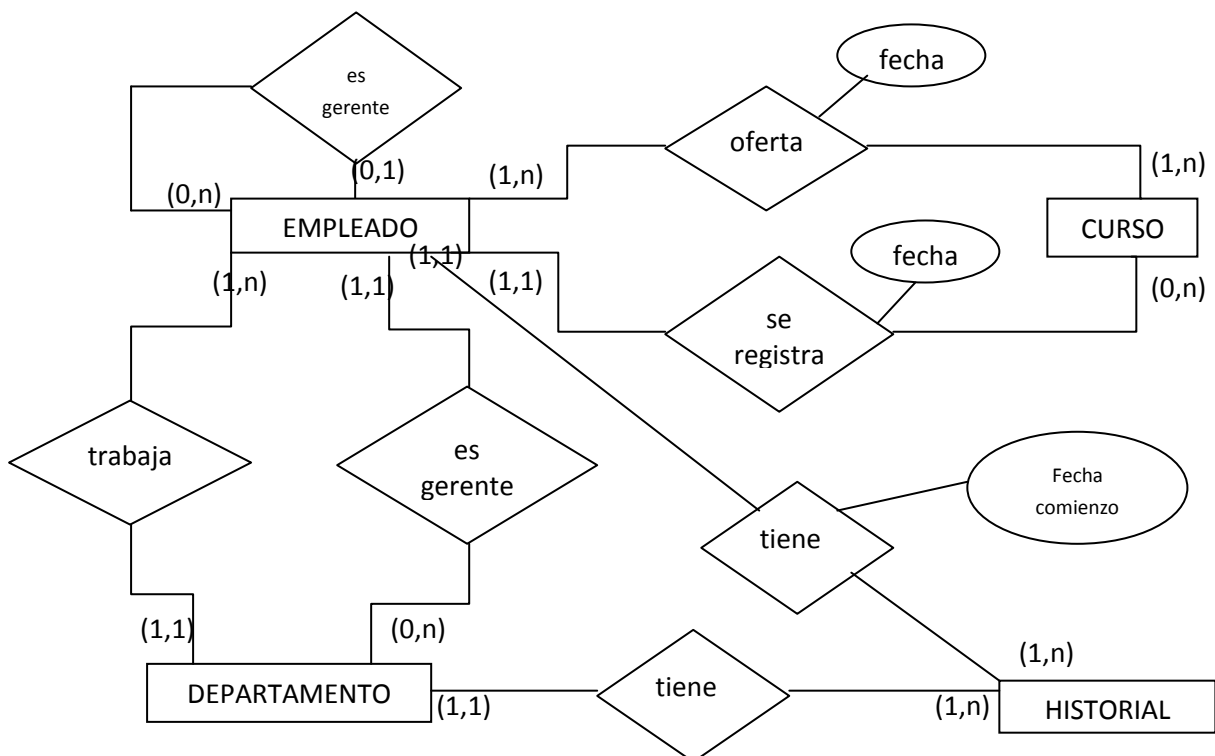
1. Definición de los requisitos, tablas, relaciones.

Diagrama entidad/relación:

Un empleado trabaja en un departamento, en un departamento puede haber varios empleados trabajando. Un empleado puede ser gerente de un departamento, o no serlo de ninguno o a lo largo de su vida laboral haber sido gerente de varios departamentos. Además un departamento siempre tendrá como gerente a uno y sólo un empleado. A su vez, un empleado puede que sea gerente de otros empleados.

Un empleado ofrece cursos para impartirlos, es importante la fecha de dicha oferta. Los empleados se registran en esas ofertas (en una o varias ofertas), a su vez cada oferta de un curso tendrá a varios empleados apuntados. Una oferta de un curso la imparte uno y sólo un empleado. Un empleado puede impartir muchas ofertas de un curso.

Se desea guardar el historial de trabajo de los empleados. Para ello tenemos una entidad historial. Los departamentos tienen muchos historiales, cada historial pertenece a un departamento. Además los empleados tienen también muchos historiales, siendo de relevancia la fecha de comienzo de cada historial, pues marca en qué momento comienza a trabajar en un determinado departamento.



Descripción de las tablas:

Tabla1: *EMPLOYEES*

COLUMNA	DESCRIPCIÓN	CLAVE
EMPNO	Número empleado	PK
ENAME	Nombre	--
INIT	Iniciales (sin puntos)	--
JOB	Trabajo del empleado	--
MGR	Número del gerente del empleado	FK
BDATE	Fecha de nacimiento	--
MSAL	Salario mensual, excluyendo bonos y comisiones	--
COMM	Comisión salario anual (sólo relevante para represent. Ventas)	--
DEPTNO	Número de departamento en el que trabaja el empleado	FK

Tabla2: *DEPARTMENTS*

COLUMNA	DESCRIPCIÓN	CLAVE
DEPTNO	Número departamento	PK
DNAME	Nombre departamento	--
LOCATION	Ciudad	--
MGR	Número del gerente del departamento	FK

Tabla3: *SALGRADES*

COLUMNA	DESCRIPCIÓN	CLAVE
GRADE	Número de la categoría del salario	PK
LOWERLIMIT	Salario más bajo de la categoría	--
UPPERLIMIT	Salario más alto de la categoría	--
BONUS	Bonificación opcional (libre de impuestos)	--

Tabla 4: *COURSES*

COLUMNA	DESCRIPCIÓN	CLAVE
CODE	Código del curso	PK
DESCRIPTION	Pequeña descripción del contenido del curso	--
CATEGORY	Indicador del tipo de curso (valores: GEN, BLD o DSG)	--
DURATION	Duración del curso expresado en días	--

El campo "CATEGORY" tiene 3 posibles valores:

- GEN (general): para cursos de introducción
- BLD (build: para programadores de aplicaciones.
- DSG (design): para analistas y diseñadores de sistemas

Tabla 5: *OFFERINGS*

COLUMNA	DESCRIPCIÓN	CLAVE
COURSE	Código del curso	PK,FK
BEGINDATE	Día de comienzo del curso ofrecido	PK
TRAINER	Número del empleado que imparte el curso	FK
LOCATION	Ciudad donde el curso es impartido	--

Tabla 6: *REGISTRATIONS*

COLUMNA	DESCRIPCIÓN	CLAVE
ATTENDEE	Número del empleado apuntado al curso	PK,FK1
COURSE	Código del curso	PK,FK2
BEGINDATE	Día de comienzo del curso ofrecido	PK,FK2
EVALUATION	Evaluación del curso por alumno (escala del 1 al 5)	--

Tabla 7: *HISTORY*

COLUMNA	DESCRIPCIÓN	CLAVE
EMPNO	Número del empleado	PK,FK1
BEGINYEAR	Año (4 dígitos) de la fecha de comienzo	--
BEGINDATE	Día de comienzo del intervalo de tiempo	PK
ENDATE	Día de finalización del intervalo de tiempo	--
DEPTNO	Número del departamento en el que trabajó durante ese intervalo	FK2
MSAL	Salario mensual durante ese intervalo	
COMMENTS	Comentarios	

En la tabla HISTORY se mantiene información sobre la historia laboral de todos los empleados. Más específicamente, están los datos de los departamentos en los que han trabajado y los salarios que han tenido a lo largo de los años, comenzando desde el día en que fueron contratados. Todos los cambios de departamento y/o cambios de salarios son anotados en esta tabla.

2. SQL

Normalmente SQL se dice que es un lenguaje de consultas, pero podemos dividirlo en cuatro categorías.

- Lenguaje de definición de datos (DDL)
- Lenguaje de manipulación de datos (DML)
- Lenguaje de consulta de datos
- Seguridad y autorización.

Data Definition (DDL)

El DDL sirve para crear, modificar y borrar componentes de la **estructura** de la base de datos. Los componentes típicos de la estructura de una base de datos son: tablas, vistas, índices, constraints (restricciones), sinónimos, secuencias. Comenzaremos por los 3 siguientes comandos:

- CREATE: crea un nuevo objeto de base de datos
- ALTER: sirve para cambiar la estructura de un objeto que ya existe en la BD.
- DROP: borrar un objeto de la base de datos.

Data Manipulation and Transactions (DML)

DML trata de los cambios en los **contenidos** de una base de datos. Tres comandos básicos para dicha manipulación:

- INSERT: para añadir filas a una tabla (registros)
- UPDATE: para cambiar valores en un determinado campo de una fila.
- DELETE: para borrar registros de una tabla

Los comandos de manipulación son tratados siempre como parte de una transacción. Esto significa que todos los cambios realizados por comandos de manipulación dependen del estatus de la BD, hasta que no se confirmen (commit) o se cancelen (rollback).

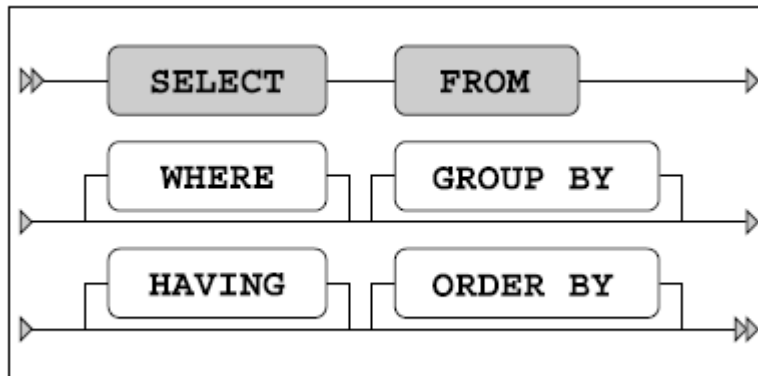
SQL ofrece dos comandos para controlar nuestras transacciones:

- COMMIT: confirma los cambios pendientes
- ROLLBACK: cancela los cambios pendientes y restaura la situación original.

A veces, las transacciones son salvadas implícitamente. Por ejemplo, todos los comandos de definición de datos se guardan sin necesidad de realizar un commit. (Create, Drop, Grant)

Lenguaje de consulta de datos

El único comando usado para realizar consultas en una base de datos es el **SELECT**.



Las líneas de este diagrama representan todas las posibilidades del comando **SELECT**. Se pueden deducir las siguientes reglas:

- El orden de los 6 componentes del comando es fijo
- El **SELECT** y el **FROM** son obligatorios
- Los otros componentes (**WHERE**, **GROUP BY**, **HAVING** y **ORDER BY** son opcionales).

COMPONENTE	DESCRIPCIÓN
FROM	¿Qué table(s) necesitamos para la consulta?
WHERE	¿Cuál es la condición para filtrar filas?
GROUP BY	¿Cómo deberían las filas ser agrupadas?
HAVING	¿Cuál es la condición para filtrar los grupos?
SELECT	¿Qué columnas quieres ver en el resultado?
ORDER BY	¿En qué orden queremos ver los resultados?

Security

SQL ofrece varios comandos para implementar la seguridad y la restricción de acceso a datos. Lo primero de todo: el acceso a la base de datos debe estar definido. La autorización de usuario es implementada a través de usuarios y passwords junto con los privilegios-ocultación sobre la base de datos. Comandos:

- **CREATE USER**: define un nuevo usuario en la base de datos.
- **ALTER USER**: cambia las propiedades (privilegios y passwords) de los usuarios existentes en la base de datos...
- **DROP USER**: borra un usuario de la base de datos.

Privileges and Roles

Si eres un usuario autorizado de la base de datos puedes conceder acceso preciso a los objetos a través de *privileges*.

Oracle ofrece dos tipos de privilegios: los privilegios de sistema y los privilegios de objetos. Los privilegios de sistema son genéricos sobre toda la base de datos, mientras que los privilegios de objeto son específicos sobre un objeto concreto.

GRANT CREATE TABLE TO CLASE; **SISTEMA**

Ahora me conecto a la base de datos como CLASE, y concedo a Pepe el privilegio de consulta sobre MI TABLA DEPARTMENTS:

GRANT SELECT ON DEPARTMENTS TO PEPE; **OBJETOS**

Privilegios más importantes sobre objetos:

Privilegio de objeto	Acción asociada
ALTER	Cambiar la estructura de la tabla
DELETE	Borrar registros
EXECUTE	Ejecutar funciones y procedimientos
FLASHBACK	Volver atrás en el tiempo
INDEX	Crear índices para tablas
INSERT	Insertar nuevos registros
REFERENCES	Crear restricciones de clave ajena haciendo referencia a una tabla no mía.
SELECT	Realizar consultas de tablas o índices
UPDATE	Cambiar columnas

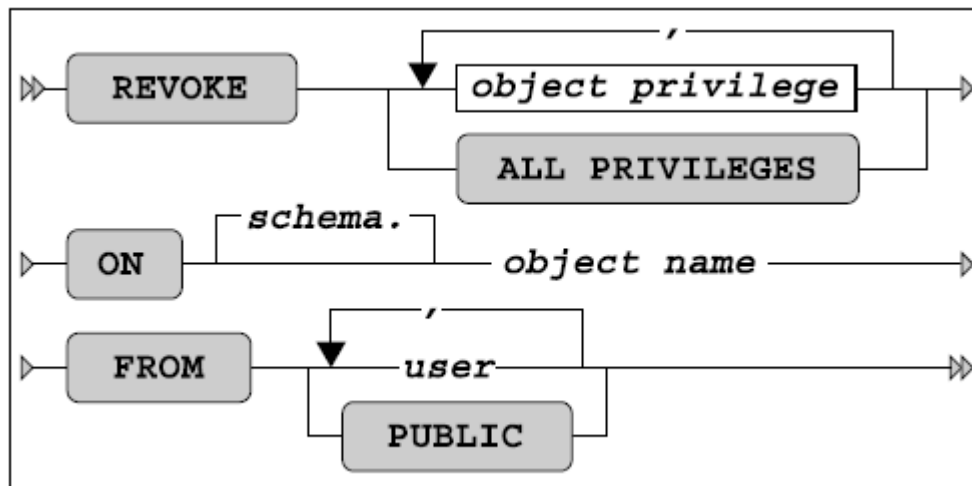
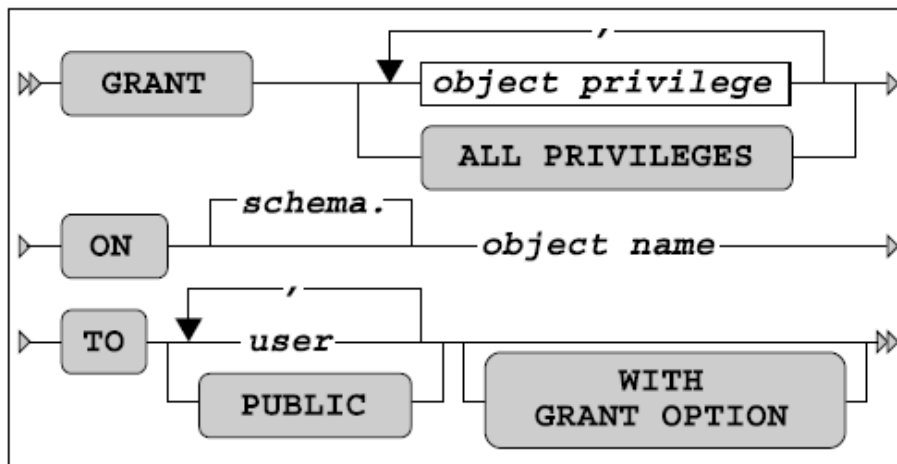
Oracle permite agrupar privilegios en *roles*. Esto hace el manejo de usuario más fácil, flexible y más manejable. Comandos SQL para conceder privilegios y roles:

GRANT: para conceder privilegios y roles

REVOKE: para quitar privilegios y roles

Un escenario típico:

```
CREATE ROLE <role name>
GRANT privileges TO <role name>
GRANT <role name> TO user(s)
```



3. CONCEPTOS Y TERMINOLOGÍA BÁSICA DE SQL

Vamos a ver los siguientes conceptos:

1. Constantes
2. Variables
3. Operadores, operandos, condiciones y expresiones
4. Funciones
5. Nombres de objetos de la base de datos
6. Comentarios
7. Palabras reservadas

3.1 CONSTANTES

Una constante es un valor fijo. Podemos distinguir entre constantes numéricas y constantes de texto. En la jerga de base de datos, las constantes alfanuméricas se denominan "strings".

En SQL las constantes alfanuméricas (cadenas) se deben colocar entre comillas simples. Los números no, si los ponemos entre comillas serán interpretados como un string. Mediante la adición de los sufijos `f` o `d` se indica la precisión simple (float) o doble, respectivamente.

En SQL, las **fechas y la duración de tiempo** (intervalos) son casos especiales. Por lo general se representan como constantes alfanuméricas, pero es necesario algo más para distinguirlas de las cadenas normales.

- Podemos usar como prefijo la palabra clave (`DATE`, `TIMESTAMP` o `INTERVAL`) para ayudar al SGBD a distinguirlas. Si utilizamos **INTERVAL** necesitaremos un sufijo para indicar una dimensión, tales como día, mes o año.
- Podemos especificarlas como constantes alfanuméricas (cadenas) y el uso del **CAST** o la función de conversión **TO_DATE**.

Ejemplos:

`DATE '2011-11-03'`

`TIMESTAMP '2011-09-05 11.42.59.00000'`

`INTERVAL '2' SECOND`

`INTERVAL '2-3' YEAR TO MONTH` (intervalo de 2 a 3 años en meses):
tiene sentido para efectuar operaciones con fechas:

`select sysdate, sysdate+interval '2-3' YEAR TO MONTH from dual;`

3.2 VARIABLES

Una variable es algo que cambia su valor en el tiempo. Todas las variables tienen un nombre para poder referirnos a ellas. SQL soporta dos tipos de variables:

- **Variables de nombre de columna:** el nombre de columna es el mismo, pero su valor varia de fila en fila en la exploración de una tabla.
- **Variables de sistema:** no tienen nada que ver con las tablas, pero son importantes. Se les llama también "pseudo columnas"

VARIABLE	DESCRIPCIÓN
SYSDATE	Fecha actual del sistema
CURRENT_DATE	Fecha actual pero desde el lado de la aplicación del cliente (conexión remota: en otra zona horaria)
SYSTIMESTAMP	Fecha y hora exacta con información de zona horaria
LOCALTIMESTAMP	Igual, pero desde el lado de la aplicación cliente (conexión remota: en otra zona horaria)
USER	El nombre usado para conectarse a la BD

3.3 OPERADORES, OPERANDOS, CONDICIONES Y EXPRESIONES

Operadores Aritméticos:

+	Suma
-	Resta
*	Multiplica
/	Divide

Podemos aplicar los operadores solo con valores numéricos. Excepciones a esto:

- La diferencia entre dos fechas expresadas en días.
- Podemos añadir una fecha a un INTERVAL, el resultado es otra fecha.
- Si sumamos una fecha y un número. Será interpretado como un INTERVAL expresado en días.

Operador alfanumérico:

Operador de concatenación: sirve para concatenar STRINGS: ||

Operadores de comparación:

<	Menor que
>	Mayor que
=	Igual a
<=	Menor o igual que
>=	Mayor o igual que
<> or !=	No igual que

Operadores lógicos:

AND	Y lógico
OR	O lógico
NOT	Negación Lógica

Expresiones:

Una expresión es una cadena que contiene variables, constantes, operadores o funciones.

3 + 4	Numérica
ENAME ', ' INIT	Alfanumérica
LOCATION = 'Utrecht'	Boolean
12*MSAL > 20000 AND COMM >= 100	Boolean
BDATE + INTERVAL '16' YEAR	Fecha
999	Numérica

Cuando las expresiones se vuelven complejas (la última expresión es una simple constante) la precedencia de operadores puede ser un problema, en otras palabras: ¿cuáles son las reglas de prioridad del operador? Por supuesto, SQL tiene algunas reglas de precedencia. Por ejemplo, los operadores aritméticos siempre tienen prioridad sobre los operadores de comparación, y los operadores de comparación tienen prioridad sobre los operadores lógicos. Sin embargo, es recomendable el uso de paréntesis en las expresiones SQL para forzar un orden determinado.

Funciones:

Oracle añade mucha funcionalidad al SQL estándar con las funciones. Las funciones es una de las razones por las que Oracle es tan poderoso. ¿Cómo reconocemos una función?, tienen un nombre, seguido por argumentos entre paréntesis y separados por comas.

Podemos encontrarnos con:

- Funciones numéricas
- Funciones alfanuméricas
- Funciones de grupo
- Funciones de fecha
- Funciones de conversión
- Otras

Por ejemplo:

AVG(MSAL)	Media de salario mensual
LENGTH(INIT)	Número de caracteres en el valor de la columna INIT (longitud)
LOWER(ENAME)	El valor de la columna ENAME en minúscula
SUBSTR(ENDDATE,4,3)	Tres caracteres del valor de la columna ENDDATE desde la cuarta posición.

Nombres de objetos de la base de datos:

Todo objeto de una base de datos necesita un nombre. Esto se aplica a tablas, columnas, vistas, índices, sinónimos, usuarios, roles, restricciones, funciones... En general, para mejorar la legibilidad de nuestro código SQL es recomendable usar caracteres de la A a la Z, dígitos (0-9) y opcionalmente el guión bajo (_).

Comentarios:

Se pondrán entre: /* comentario */

Palabras reservadas:

Como en cualquier otro lenguaje las palabras reservadas no se pueden usar para nombrar objetos de la base de datos. Algunos ejemplos de palabras reservadas: AND, CREATE, DROP, FROM, GRANT, HAVING, INDEX, INSERT, MODIFY, NOT, NULL, NUMBER, OR, ORDER, RENAME, REVOKE, SELECT, SYNONYM, SYSDATE, TABLE, UPDATE, USER, VALUES, VIEW, WHERE...

4. SQL*PLUS

Es la herramienta usada para meter comandos SQL y mostrar los resultados. Los comandos SQL normalmente ocuparán varias líneas, SQL*Plus muestra automáticamente los números de línea. Para ejecutar un comando, deberemos finalizarlo con un punto y coma (;). Si se nos olvida ponerlo, podremos ponerlo en la línea vacía.

```
SQL> select *  
2 from employees;
```

O

```
SQL> select *  
2 from employees  
3 ;
```

¿Qué nos muestra este comando? (hacer)

4.1. SQL Buffer

SQL * Plus almacena los comandos SQL más recientes en una zona conocida como el buffer SQL. El buffer de SQL es un importante concepto de SQL * Plus. Podemos visualizar el contenido del buffer de SQL mediante un comando: L (probarlo).

Podemos editar el contenido del búfer de SQL con un editor externo que nos dará más facilidad de uso para modificar comandos SQL.

```
SQL> define _editor=Notepad
```

Una vez hecho esto, podemos invocarlo únicamente escribiendo:

```
SQL> edit
```

Lo modificamos en el notepad, al salir lo guardamos y luego sólo tenemos que pulsar:

```
SQL> r
```

4.2. Ajustando la configuración de SQL*Plus

Podemos modificar el entorno de numerosas formas.

Algunos ejemplos:

```
SQL> set pagesize 22  
SQL> set pause "Hit [Enter]..."  
SQL> set pause on
```

4.3. Salvando comandos

El comando **SAVE** crea un script que contiene lo que haya en el buffer.
Si hacemos:

```
SQL> select * from departments;  
SQL> save BLA.sql
```

Se crea un script **BLA.sql** que puedo ejecutar en cualquier momento:

```
SQL> @BLA.sql
```

4.4. Describiendo objetos de la Base de Datos

Cuando ejecutamos comandos SQL es conveniente ver/saber la estructura de una tabla, por ejemplo, para ver los nombres y los tipos de las columnas. Para ello utilizaremos el comando **descr**

```
SQL> descr employees;
```

4.5. Algunos comandos importantes SQL

Comando	Descripción
SAVE	Salva el contenido del buffer en un script.
START o @	Ejecuta un script
SPOOL	Copia la salida por pantalla en un fichero
SET	Cambia configuración del sql*plus
COL ... FOR	Cambia el formato de salida de las columnas: COLUMN description FORMAT a26 COL bonus FOR 9999.99
STORE SET	Guarda los cambios de configuración en un fichero
DESCRIBE	Describe la estructura de una tabla
/ o R	Repite última sentencia sql que haya en el buffer
CLEAR BUFFER	Limpia buffer

5. DATA DEFINITION

5.1. Esquema y usuarios

Antes de comenzar a crear tablas con SQL necesitamos entender cómo se guardan internamente los datos en la base de datos. Para entrar en la base de datos primero necesitamos especificar un usuario y una password. Este proceso te identifica como usuario de la base de datos.

La base de datos Oracle tiene, en general, una relación uno-uno entre un usuario de la base de datos y un esquema con el mismo nombre.

Las diferencias entre un usuario de base de datos y un esquema de base de datos:

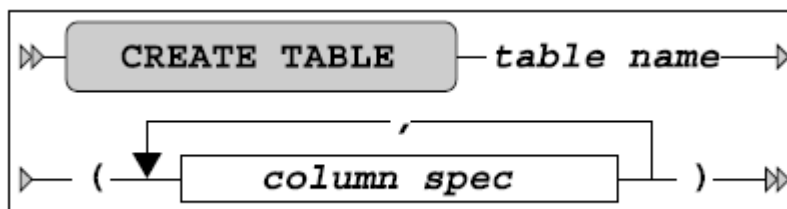
- Un usuario de base de datos tiene una contraseña y los privilegios en ciertas bases de datos.
- Un esquema de base de datos es una colección lógica de objetos de base de datos (como tablas, índices, vistas, etc) que suele ser propiedad del usuario con el mismo nombre.

Normalmente cuando nos conectamos a una BD Oracle, nos conectamos automáticamente al esquema de BD correspondiente al mismo nombre.

Sin embargo, también es posible que algunos usuarios de bases de datos no tengan su propio esquema, es decir, no tengan ningún objeto de base de datos propio, y no tengan los privilegios para crearlos. Estos usuarios son, por ejemplo, usuarios autorizados para recuperar o manipular datos en un esquema de base de datos diferente.

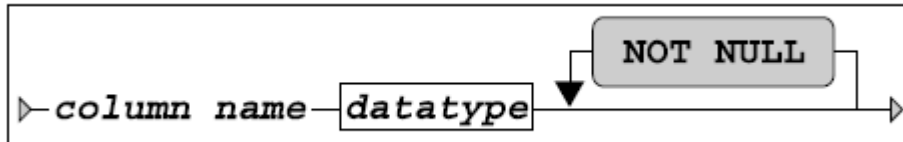
5.2. CREACIÓN DE TABLAS

El comando para crear tablas es: **CREATE TABLE**. Se debe especificar el nombre (nuevo) seguido de todas las columnas separadas por comas entre paréntesis.



Esto no muestra la sintaxis completa del comando `CREATE TABLE`. Es sólo una pequeña muestra básica de dicho comando.

Las especificaciones de las columnas tendrán varios componentes, por ejemplo:



Cada especificación de la columna empieza con un nombre de columna, seguida por el tipo de datos. Si se agrega la expresión opcional NOT NULL a una definición de columna, cada fila de la tabla que creemos debe tener un valor para esta columna. En otras palabras, se define la columna como un atributo obligatorio.

La adición NOT NULL es un ejemplo de una restricción. Podemos especificar muchas restricciones adicionales en el comando CREATE TABLE. Otros tipos de restricciones son por ejemplo: UNIQUE, CHECK, PRIMARY KEY y FOREIGN KEY.

5.3. TIPOS DE DATOS

Oracle soporta algunos tipos de datos estándar (ANSI/ISO SQL estándar). En general, hay 3 categorías de datos: numéricos, texto y fecha. Los más importantes en Oracle son NUMBER, VARCHAR o VARCHAR2 Y DATE. Ejemplos:

NUMBER(4)	Entero con un máximo de 4 dígitos.
NUMBER(6,2)	Numérico con un máximo de 6 dígitos y dos decimales.
VARCHAR2 (25)	Alfanumérica de máximo 25 caracteres
CHAR (4)	Alfanumérico de 4 caracteres
CLOB	Alfanumérico, el máximo de un VARCHAR2
DATE	Fecha (dd-mm-aa)

5.4. COMANDOS PARA LA CREACIÓN DE LAS TABLAS CON LAS QUE TRABAJAREMOS

Las definiciones de restricción no las tendremos por ahora en cuenta. Después iremos completando el comando de creación de tablas con dichas restricciones.

```
create table EMPLOYEES
( empno number(4) not null
, ename varchar2(8) not null
, init varchar2(5) not null
, job varchar2(8)
, mgr number(4)
, bdate date not null
, msal number(6,2) not null
, comm number(6,2)
, deptno number(2) );
```

```
create table DEPARTMENTS
( deptno number(2) not null
, dname varchar2(10) not null
, location varchar2(8) not null
, mgr number(4) );
```

```
create table SALGRADES
( grade number(2) not null
, lowerlimit number(6,2) not null
, upperlimit number(6,2) not null
, bonus number(6,2) not null );
```

```
create table COURSES
( code varchar2(6) not null
, description varchar2(30) not null
, category char(3) not null
, duration number(2) not null );
```

```
create table OFFERINGS
( course varchar2(6) not null
, begindate date not null
, trainer number(4)
, location varchar2(8) );
```

```
create table REGISTRATIONS  
( attendee number(4) not null  
  , course varchar2(6) not null  
  , begindate date not null  
  , evaluation number(1) );
```

```
create table HISTORY  
( empno number(4) not null  
  , beginyear number(4) not null  
  , begindate date not null  
  , enddate date  
  , deptno number(2) not null  
  , msal number(6,2) not null  
  , comments varchar2(60) );
```

5.5. EL DICCIONARIO DE DATOS

Si estamos interesados en conocer datos sobre nuestro modelo tales como las tablas que tenemos, las columnas, cuáles de ellas están indexadas, qué privilegios tenemos, etc. tenemos que consultar el diccionario de datos. Ya hemos visto anteriormente el comando DESC.

Algunos usos de vistas del DD:

VISTA	DESCRIPCIÓN
DICTIONARY	Descripción del diccionario de datos
DICT_COLUMNS	Descripción de las columnas de DD
ALL_USERS	Información sobre los usuarios de la BD
ALL_INDEXES ¹	Todos los índices
ALL_OBJECTS ¹	Todos los objetos
ALL_SYNONYMS ¹	Todos los sinónimos
ALL_TABLES ¹	Todas las tablas
ALL_VIEWS ¹	Todas las vistas
USER_INDEXES ²	Los índices del esquema de un usuario
USER_SEQUENCES ²	Las secuencias del usuario
USER_OBJECTS ²	Los objetos del usuario
USER_SYNONYMS ²	Los sinónimos del usuario
USER_TABLES ²	Las tablas
USER_TAB_COLUMNS ²	Las columnas
USER_VIEWS ²	Las vistas
CAT	Tablas del esquema del usuarios
COLS	Columnas de las tablas del usuario
DUAL	Tabla comodín cuando queremos consultar o hacer algún cálculo que no está en ninguna tabla: por ejemplo, una suma de fechas o de números.

Sinónimos: cuando yo tengo una tabla y necesito dar permisos, por ejemplo, de selección a otro usuario sobre esa tabla, para que no tenga que poner: marta.tabla, me creo un sinónimo:

CREATE SYNONYM clientes FOR marta.clientes;

Secuencias: tiene por objeto conseguir generar columnas numéricas autoincrementables. Por ejemplo:

```
CREATE SEQUENCE idPedSec START WITH 1 INCREMENT BY 1;
```

Nos crearíamos un trigger "before insert for each row" sobre la tabla en cuestión y dentro de ese trigger capturamos el siguiente valor de la secuencia y se lo asignamos al campo que queremos autoincrementar en la tabla.