

# Práctica 1. Introducción a los SGBD y a SQL

## 0. Introducción

Esta primera práctica tiene como objetivo la revisión de algunos conceptos básicos relacionados con las bases de datos y los sistemas gestores de bases de datos, sus componentes y los tipos de usuarios que los utilizan. También se incluye una pequeña introducción a SQL (Structured Query Language), que es el lenguaje de consultas estándar que usaremos durante la mayoría de las sesiones de laboratorio. Además se explica el entorno de trabajo de las prácticas y las herramientas con que cuenta el SGBD Oracle para interactuar con la base de datos.

## 1. Sistemas Gestores de Bases de Datos

Empezaremos recordando las diferencias, ya estudiadas previamente, entre una base de datos y un sistema gestor de bases de datos:

- Una base de datos es un conjunto de datos relacionados entre si.
- Un sistema gestor de bases de datos (en adelante SGBD<sup>1</sup>), es un conjunto de programas que permite a los usuarios crear y mantener una o varias bases de datos.

Las bases de datos y sus herramientas forman parte actualmente del núcleo informático de cualquier empresa u organización.

Los principales SGBD libres son:

- PostgreSQL (<http://www.postgresql.org>)
- Firebird (<http://www.firebirdsql.org>)
- SQLite (<http://www.sqlite.org>)
- MySQL (<http://www.mysql.com>)

Por otra parte entre los SGBD comerciales podemos destacar:

- IBM DB2 (<http://www-01.ibm.com/software/data/db2/>)
- IBM IMS. Base de Datos Jerárquica (<http://www-01.ibm.com/software/data/ims/>)
- IBM Informix (<http://www-01.ibm.com/software/data/informix/>)
- Microsoft Access (<http://office.microsoft.com/es-es/access>)
- Microsoft SQL Server (<http://www.microsoft.com/sqlserver/en/us/default.aspx>)
- Oracle (<http://www.oracle.com/>)
- Sybase (<http://www.sybase.com/>)

Cualquiera de estos sistemas tienden a ofrecer, quizá con diferentes filosofías, las mismas facilidades de gestión y soporte para el lenguaje SQL. Por ello, aunque a lo largo del curso nos centraremos principalmente en el uso del SGBD Oracle 12c, los conceptos subyacentes son aplicables para entender los principales SGBD actuales.

Oracle fue uno de los primeros SGBD basados en el modelo relacional, que utilizó el lenguaje SQL para definición y acceso a los datos. La primera versión data del año 1.978, y en sucesivas versiones ha presentado diferentes mejoras. Su utilización está bastante extendida en grandes sistemas.

Entre los SGBD libres, actualmente el más extendido es MySQL desarrollado por MySQL AB (actualmente es una empresa subsidiaria de Oracle) que lo oferta bajo licenciamiento dual. Es un SGBD relacional, multihilo y multiusuario.

<sup>1</sup> DBMS son las siglas de Database Management System.

Las principales características que presentan los SGBD actuales orientados a un volumen de información medio/alto son:

- **Ejecución de operaciones eficientes.-** Esto se consigue mediante el optimizador de consultas, el uso de diferentes mecanismos de indexación y algoritmos de ejecución y la adecuada gestión del buffer.
- **Multiusuario.-** Múltiples usuarios pueden acceder a la vez a la misma o distinta información de la BD.
- **Concurrencia.-** En un ambiente multiusuario concurrente se garantiza la consistencia de los datos mediante transacciones, bloqueo sobre los datos (locking) y otros mecanismos como, por ejemplo, el uso de segmentos de rollbacks.
- **Seguridad.-** Sistema de acceso mediante palabra clave (password), privilegios (system/object/granting privileges) y papeles (grant/revoke), limitaciones de recursos (cuotas y límites de recursos para cada usuario), auditorías del sistema (statement/privilege/object auditing). Y en Oracle, por ejemplo, la disponibilidad de Trusted Oracle.
- Además, permite restricciones de integridad declarativas y el uso de **procedimientos y disparadores** (triggers) SQL almacenados.
- **Copias de seguridad y recuperación de la base de datos.-** Para recuperar errores de usuario, de sentencias, de instancia, de discos. Permiten realizar copias de seguridad en frío y en caliente.
- **Portabilidad.-** Cada SGBD tiene versiones disponibles en los principales sistemas operativos (Linux, Windows, Solaris, AIX ... etc.), permitiendo el traspaso inmediato de las BD entre cada uno de ellos. Además disponen de herramientas de migración de BD entre diferentes SGBD.
- **Procesamiento distribuido.-** Se utiliza una arquitectura cliente-servidor adaptable, que permite usar varios procesadores.
- **Bases de datos distribuidas.-** Múltiples servidores de bases de datos conectados a través de una red aparecen al usuario como una única base de datos lógica. Para mantener la consistencia de actualizaciones se utilizan técnicas específicas, como la del two-phase-commit.

## 2. Componentes de la BD

### 2.1 Esquemas y objetos del esquema

Un esquema es una colección de objetos. Los objetos del esquema son las estructuras lógicas que son propiedad de un usuario de la base de datos. Conviene indicar que estamos hablando de estructuras lógicas y no de su almacenamiento físico. No todos los SGBD organizan el almacenamiento físico de los objetos de la misma manera.

Veamos a continuación los diferentes tipos de objetos que pueden integrarse en un esquema.

#### 2.1.1 Tablas

Una tabla es la unidad básica de almacenamiento lógico de datos en una base de datos relacional. Esta formada por varias columnas, que reciben el nombre de atributos. Del mismo modo puede contener varias filas, a las que se denomina también tuplas. Cada columna tiene un tipo de dato y un conjunto de restricciones asociado, lo que permite al SGBD saber cómo debe manipular los datos. Ejemplo de la tabla de ALUMNOS:

NUM_EXP	TITULACION	APELLIDO1	APELLIDO2	NOMBRE	} ATRIBUTOS
1	101	PEREZ	PEREZ	JUAN	
2	101	GARCIA	SANCHEZ	PAULA	} TUPLAS
3	102	LARRA	ALVAREZ	PEDRO	
4	102	VAZQUEZ	PEDROSA	IRENE	
5	103	TENA	TENA	ALBERTO	

### 2.1.2 Vistas

Una vista es una presentación de los datos ajustada a las características del usuario. Una vista puede también interpretarse como una consulta almacenada. Las vistas realmente no contienen datos almacenados<sup>2</sup>, sino que derivan sus datos de las tablas sobre las que se basan, referidas como tablas base de la vista. Las tablas base pueden ser a su vez tablas o bien vistas.

Exactamente igual que las tablas, las vistas pueden ser consultadas, actualizadas, recoger nuevas tuplas a insertar o eliminar tuplas a borrar, aunque eso sí, con ciertas restricciones. Todas las operaciones realizadas sobre una vista afectan a las tablas base de la misma.

Usualmente, las vistas se utilizan con los siguientes propósitos:

- Restringir el acceso a partes de una tabla, proporcionando así un mecanismo de seguridad sobre una tabla.
- Ocultar la complejidad de los datos, por ejemplo, para construir un *join* entre dos o más tablas.
- Simplificar tareas de búsqueda de datos a los usuarios.
- Presentar los datos en una perspectiva diferente a como se almacenan en la tabla base, por ejemplo, para renombrar columnas sin modificar la tabla.
- Almacenar consultas complejas, a las que se asigna un nombre.

Ejemplo: Vista de alumnos de la titulación 101.

NUM_EXP	TITULACION	APELLIDO1	APELLIDO2	NOMBRE
1	101	PEREZ	PEREZ	JUAN
2	101	GARCIA	SANCHEZ	PAULA
3	102	LARRA	ALVAREZ	PEDRO
4	102	VAZQUEZ	PEDROSA	IRENE
5	103	TENA	TENA	ALBERTO

### 2.1.3 Secuencias

Una secuencia permite generar una lista ordenada de números únicos para columnas numéricas de una tabla. Las secuencias simplifican las tareas de programación generando de forma automática valores únicos en serie.

Las secuencias son independientes de las tablas, de modo que una misma secuencia puede usarse en más de una tabla. Después de su creación, una secuencia puede ser utilizada por más de un usuario para generar números de secuencia reales.

Este objeto no existe como tal en algunos SGBD, como en MySQL, pero a cambio se dispone de la cláusula `AUTO_INCREMENT` que puede usarse dentro de la definición de una columna en una tabla.

<sup>2</sup> Con la excepción de lo que se denominan vistas materializadas.

### 2.1.4 Unidades de programa

Son procedimientos o funciones, es decir conjuntos de sentencias SQL agrupadas como una unidad ejecutable para llevar a cabo una tarea específica y que se encuentran almacenados dentro de la propia BD.

### 2.1.5 Disparadores

Un disparador es una acción que se ejecuta automáticamente cuando se produce un determinado evento en la BD. Las acciones pueden consistir en la ejecución de una unidad de programa.

### 2.1.6 Sinónimos

Un sinónimo es un alias para una tabla, vista, secuencia, o unidad de programa. Un sinónimo no es realmente un objeto en sí mismo, sino que se utiliza para referenciar directamente a un objeto. Los sinónimos pueden ser públicos o privados. Un usuario individual puede crear sinónimos privados, que están disponibles a dicho usuario. Los administradores de la base de datos construyen sinónimos públicos que ponen el objeto referenciado disponible para uso general por cualquier usuario de la base de datos. No todos los SGBD permiten crear este tipo de objeto.

### 2.1.7 Índices

Se trata de estructuras opcionales asociadas con tablas que pueden ser creadas para mejorar el rendimiento en el acceso a los datos de dichas tablas. Los índices se construyen para acelerar el acceso a los datos. Cuando se procesa una consulta, los SGBD puede usar uno o más índices disponibles con objeto de localizar eficientemente las tuplas requeridas. Un índice puede crearse sobre una o más columnas de una tabla. Una vez creado, son automáticamente mantenidos por el SGBD.

Los índices son lógicamente y físicamente independientes de los datos. Pueden ser creados y eliminados en cualquier momento sin que afecten en sí a los datos de la tabla o de otros índices. Pero pueden afectar muy negativamente a consultas que los usasen previamente.

## 2.2 El diccionario de datos

Cada base de datos contiene un conjunto de información denominada metadatos (datos acerca de los datos) . Estos metadatos reciben el nombre de diccionario de datos o catálogo del sistema. Se trata de un conjunto de tablas y vistas que son utilizadas por los usuarios como referencia de **sólo lectura** de la base de datos. Por ejemplo, el diccionario almacena información sobre:

- Objetos de la base de datos (tablas, vistas, índices, restricciones, etc.)
- Usuarios válidos de una base de datos
- Información sobre las restricciones de integridad definidas para las tablas
- Cantidad de espacio asignado a los objetos de un esquema y uso del mismo.

Con objeto de reflejar de forma segura el estado de la base de datos durante todo el tiempo de vida de la misma, el diccionario se actualiza automáticamente por el SGBD en respuesta a acciones específicas. Los datos del diccionario son críticos para la operación de la base de datos.

La estructura e implementación del diccionario de datos está en función del SGBD.

En Oracle cada BD tiene su propio diccionario de datos al que se accede mediante la sentencia SQL de consulta SELECT sobre una serie de vistas. Por ejemplo:

```
SELECT object_name, object_type
FROM USER_OBJECTS;
```

En MySQL la información del diccionario de datos se encuentra centralizada dentro de la BD INFORMATION\_SCHEMA y el acceso a la información de la misma, al igual que en Oracle, debe hacerse mediante sentencias SQL de consulta sobre las vistas que contiene.

### 3. SQL

El lenguaje SQL (Structured Query Language) es el lenguaje por excelencia tanto para definición y gestión de las estructuras de la BD como para acceso y manipulación de la propia información que contiene.

Los inicios de este lenguaje se remontan al lenguaje SEQUEL implementado por primera vez en un SGBD comercial en 1979 por Oracle. En 1986 fue estandarizado por ANSI-ISO dando lugar a la versión SQL1 o SQL86. Las versiones más soportadas por los SGBD actuales son la SQL92 o SQL2 y la SQL:2003. Tanto MySQL como Oracle cumplen el estándar SQL92 y en mayor o menor medida el SQL:2003.

Las sentencias SQL pueden dividirse en varias categorías:

- Sentencias del Lenguaje de Manipulación de datos (DML),
- Sentencias del Lenguaje de Definición de datos (DDL)
- Sentencias de control de transacciones (TCL)
- Sentencias de control de datos (DCL)
- Sentencias de control de sesión. (Oracle)
- Sentencias de control de sistema (Oracle)
- Sentencias de SQL embebido

En esta asignatura veremos las sentencias incluidas en las cuatro primeras categorías que son comunes en la mayoría de los SGBD, pero solo profundizaremos en el uso de las sentencias del DML y DDL.

#### 3.1 Sentencias del Lenguaje de Manipulación de Datos (DML)

Permiten cambiar o consultar los datos contenidos en una tabla de la BD, pero no permiten cambiar la estructura de una tabla u otro objeto.

Las principales sentencias son: SELECT (consulta), INSERT (alta), UPDATE (actualización) y DELETE (borrado).

**Ejemplo 1:** Consulta de los alumnos de la titulación 102

```
SELECT num_exp, apellido1, apellido2, nombre
FROM alumnos
WHERE titulacion=102;
```

**Ejemplo 2:** Actualización de los alumnos de la titulación 102 a la 105.

```
UPDATE alumnos
SET titulacion = 105
WHERE titulacion = 102;
```

**Ejemplo 3:** Borrado de todos los alumnos de la titulación 103.

```
DELETE
FROM alumnos
WHERE titulacion = 103
```

**Ejemplo 4:** Alta de un alumno en la titulación 102.

```
INSERT INTO alumnos
        (num_exp,apellido1,apellido2,nombre,titulacion)
VALUES (snum_exp.NEXTVAL,'RUIZ','VALERO','BENITO',102);
```

### 3.2 Sentencias del Lenguaje de Definición de Datos (DDL)

Permiten crear, borrar o modificar la estructura de un objeto del esquema, es decir su definición.

Las principales sentencias son: CREATE (creación), DROP (borrado) y ALTER (modificación).

**Ejemplo:** Creación de la la tabla de provincias:

```
CREATE TABLE provincias
(cod_provincia          NUMBER(2),
 nombre_provincia       VARCHAR2(30) NOT NULL,
 PRIMARY KEY (cod_provincia));
```

### 3.3 Sentencias de control de transacciones (TCL)

Garantizan la consistencia de los datos, organizando las órdenes SQL en transacciones lógicas que se completan o fallan como una unidad operacional.

Las sentencias SQL de control de transacciones son: COMMIT, ROLLBACK y SAVEPOINT.

### 3.4 Sentencias de control de datos (DCL)

En este grupo se incluyen las sentencias que cambian los permisos que poseen los usuarios en general o con relación a los objetos del esquema.

Son las sentencias GRANT (otorgar privilegios) y REVOKE (denegar privilegios).

## 4. Tipos de usuarios

Dado que los SGBD se encuentra soportados por un sistema operativo (SO), para poder acceder como usuario a una base de datos, es preciso tener capacidad de acceso al sistema. Sin embargo, no todos los usuarios del sistema son necesariamente usuarios de la base de datos. Para tener capacidad de conectar con una base de datos, normalmente es preciso poseer determinados permisos a nivel del SO.

Así, en principio, podemos identificar dos perfiles de usuarios: usuarios del sistema y usuarios de la base de datos.

### 4.1 Usuarios del sistema

Usuario del sistema es todo aquel que posee una cuenta de acceso dentro de un servidor específico. El encargado de abrir estas cuentas es el administrador del sistema. Cada cuenta se

identifica por un *login* y, generalmente requiere la introducción de una palabra clave sólo conocida por el usuario propietario de la cuenta.

Una vez conectado al sistema, las diferentes operaciones que se puedan realizar están sujetas a los derechos que hayan sido concedidos a dicho usuario por el administrador del sistema.

## 4.2 Usuarios de la base de datos

Ser usuario de sistema no otorga automáticamente derechos de acceso a la base de datos. Para ello es preciso que el administrador de la base de datos abra cuentas de acceso específico a un usuario del sistema.

Por ejemplo, en una base de datos ORACLE se distinguen tres categorías básicas de usuarios: *connect*, *resource* y *dba*. Veamos brevemente cada una de ellas.

<b>CONNECT</b>	Este tipo de usuario sólo puede utilizar los objetos para los que se le ha concedido explícitamente derechos de acceso. Es el mínimo nivel de acceso a una base de datos.
<b>RESOURCE</b>	Este tipo de usuario puede utilizar los objetos para los cuales poseen permiso de acceso y, además, crear sus propios objetos.
<b>DBA</b>	Es el usuario encargado de administrar la base de datos

En MySQL no existen categorías básicas de usuarios. Por defecto existe un usuario *root* que tiene todos los privilegios (DBA); al resto de usuarios se le otorgarán un mayor o menor número de privilegios en función de las tareas que tengan que desarrollar sobre la BD.

## 5. El administrador de la base de datos (DBA o ABD)

Ya que una base de datos puede ser bastante grande y tener muchos usuarios, una persona o grupo de personas deben ser responsables de la gestión del sistema. El administrador de la base de datos es este responsable.

Cada base de datos requiere por lo menos una persona para realizar las tareas administrativas. Si la base de datos es grande, estas tareas administrativas pueden ser compartidas entre varios administradores.

Son responsabilidades del administrador:

- Instalar y actualizar el SGBD y sus herramientas de aplicación.
- Crear cada una de las bases de datos necesarias.
- Crear y modificar el esquema conceptual de la base de datos (junto con los analistas de las aplicaciones).
- Dar de alta a los usuarios y mantener la seguridad del sistema.
- Arrancar y detener la base de datos.
- Planificar y organizar el almacenamiento físico de la base de datos.
- Planificar las políticas de copias de seguridad y respaldo de la base de datos.
- Restaurar la base de datos tras una parada anormal de la base de datos.
- Monitorizar la actividad de la base de datos.
- Controlar y monitorizar el acceso de los usuarios a la base de datos.
- Mejorar el rendimiento con el ajuste de parámetros y con la explotación de las estadísticas suministradas por el sistema.

## 6. El entorno de trabajo para las prácticas

### 6.1 La sala de prácticas

Las prácticas se realizarán habitualmente en las salas Norba 1 y/o Norba 2. Estas salas cuentan cada una con 15 equipos que se encuentran conectados a la intranet de la UEX.

Cada equipo de la sala cuenta con dos sistemas operativos instalados:

- Linux
- Windows

Para las prácticas tendremos que seleccionar inicialmente el arranque de Windows. Una vez arrancado deberemos seleccionar el usuario windows **usuario** (sin contraseña).

Los ficheros necesarios para la realización de las prácticas deberán ser creados siempre en el mismo directorio (usualmente en Mis Documentos\practbd). Una vez finalizada la práctica deberán ser borrados todos los ficheros.

No obstante, es altamente recomendable que cada estudiante trabaje con su propio portátil, para lo que necesitará instalar al menos la herramienta SQL Developer, que se comenta en los siguientes apartados.

### 6.2 El servidor de bases de datos y las herramientas de acceso disponibles

Para las prácticas, se dispone de un servidor Linux/Debian que tiene instalado el SGBD Oracle 12c, donde tendremos almacenados todos los objetos necesarios.

En los equipos no está instalada de manera local ninguna base de datos, sino que realizaremos operaciones con la base de datos Oracle disponible en el servidor. Para ello podemos usar varios interfaces diferentes basados en la utilidad denominada SQL\*Plus.

Los datos necesarios para el acceso remoto a cada BD tanto en la sala de prácticas como fuera de ella serán proporcionados de manera individual a cada alumno en las clases prácticas o a través del campus virtual.

## 7. La herramienta SQL\*PLUS de Oracle

### 7.1 ¿Cómo empezar a trabajar con SQL\*PLUS?

Para trabajar con las utilidades SQL\*Plus de ORACLE, deberemos proporcionar siempre, al menos, la siguiente información:

- **usuario**. Usuario de la base de datos
- **password**. Contraseña asociada al usuario de la base de datos

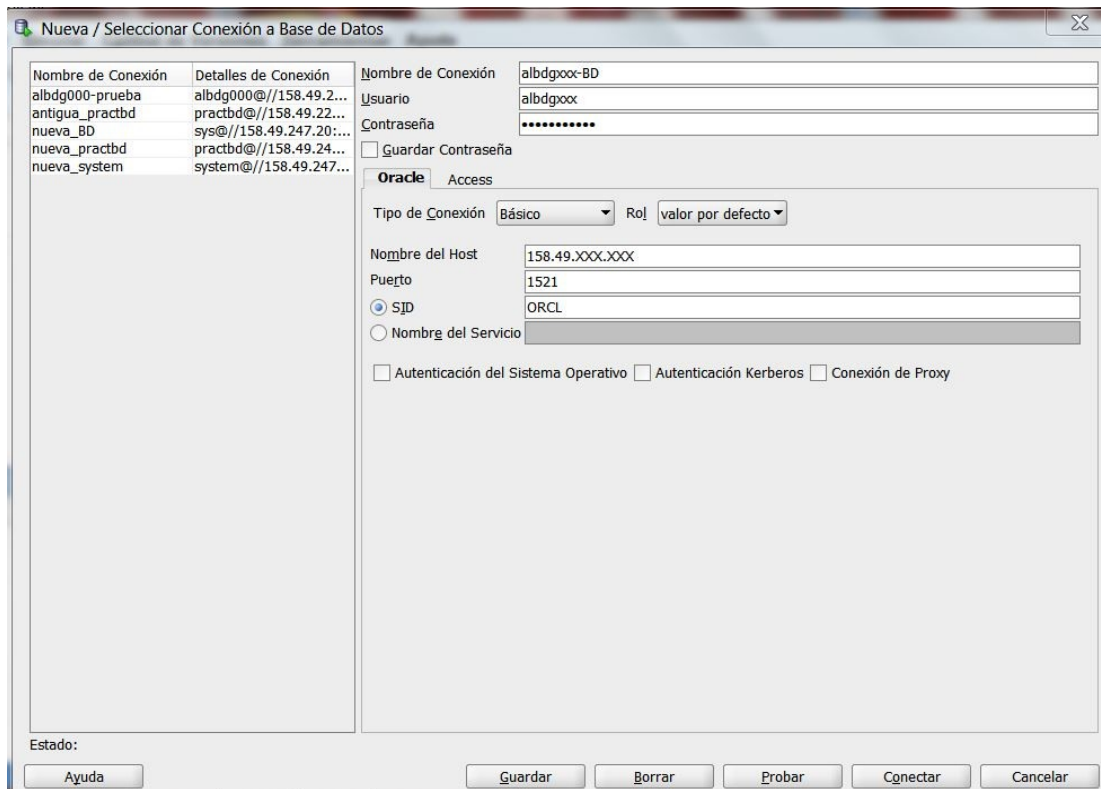
Dependiendo de la interfaz usada habrá que proporcionar, además, datos adicionales.

Oracle nos brinda varias posibilidades a la hora de conectarnos a una base de datos para poder ejecutar sentencias SQL:

- **SQL Developer**. Es una versión gráfica gratuita de SQL\*Plus que nos permite realizar las tareas básicas sobre una base de datos, tales como definir, modificar y borrar objetos de la base de datos o manipular el contenido de los mismos, ejecutar sentencias SQL y scripts, editar y depurar código PL/SQL, etc. Nosotros sólo usaremos una pequeña parte de la misma: la que nos permite ejecutar sentencias SQL y scripts. Para usarla es preciso descargar la versión



correspondiente de la web de Oracle<sup>3</sup>. Una vez instalada la herramienta y en ejecución, para conectarnos al servidor de Oracle se deberá abrir una nueva conexión, indicando los parámetros que se indiquen en la sesión.



- **SQL\*Plus en modo línea de comandos.** Permite la ejecución de ficheros con sentencias SQL desde la línea de comandos. También permite la ejecución interactiva de sentencias SQL. Si no se tiene instalado el SGBD de manera local, será preciso instalar Instant Client<sup>4</sup> (el paquete básico y el de SQL\*Plus). Para su ejecución desde una sesión MS-DOS abierta (línea de comandos):

```
C:\>sqlplus
```

```
SQL*Plus: Release 9.2.0.1.0 - Production on Lun Oct 13 11:21:25 2003
```

```
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
```

```
Introduzca el nombre de usuario: albd001@bd
```

```
Introduzca la contraseña:
```

```
Conectado a:
```

```
Oracle9i Enterprise Edition Release 9.2.0.1.0 - Production  
With the Partitioning, OLAP and Oracle Data Mining options  
JServer Release 9.2.0.1.0 - Production
```

<sup>3</sup> La versión 3.2.2 puede descargarse de:

<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/sqldev-downloads-v322-2080107.html>

<sup>4</sup> <http://www.oracle.com/technetwork/database/features/instant-client/index-100365.html>

```
SQL> exit
Desconectado de Oracle9i Enterprise Edition Release 9.2.0.1.0 -
Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.1.0 - Production

C:\>
```

## 7.2 Finalización de la sesión de trabajo

Debemos teclear desde el prompt de SQL\*PLUS la instrucción **exit**. En SQL Developer debemos desconectar la conexión.

## 7.3 Introducción de comandos

No se distinguen mayúsculas de minúsculas excepto en los datos. Existen dos tipos de comandos:

- **Comandos SQL:** Todos tienen que acabar con punto y coma. Siguen la definición del lenguaje SQL, que suele ofrecer pocas variaciones para cualquier base de datos relacional.
- **Comandos SQL\*PLUS:** Son exclusivos de Oracle, no tienen relación directa con las operaciones sobre los datos, sino que se utilizan para optimizar la interfaz de acceso a los datos. No es necesario que terminen en punto y coma. No todos los comandos pueden ejecutarse en SQL Developer.

**Ejemplo:** Introducir las siguientes instrucciones, donde la primera es una instrucción de SQL que nos devolverá el contenido de la tabla de proveedores; y la segunda es una sentencia de SQL\*Plus que permite mostrar todas las variables de entorno definidas.

```
select *
from practbd.proveedores;
show all
```

## 7.4 Ejecución de un fichero SQL

Podemos ejecutar un fichero desde SQL\*Plus mediante el comando **start**.

Ejemplo: `SQL> start prueba` o `SQL> sta prueba` o `SQL> @prueba`

Para cortar la ejecución del fichero sin esperar a su conclusión basta pulsar <Ctrl+C>.

Además, también puede llamarse a SQL\*Plus desde el intérprete de comandos de DOS, e indicarle que tome como entrada los comandos que figuran en el fichero correspondiente que se le pasa como parámetro. Del mismo modo puede hacerse incluyendo la llamada en un fichero ejecutable del sistema operativo (.bat). La sintaxis genérica que se utiliza es la siguiente:

```
sqlplus usuario/contraseña@cadena_de_conexión5 @fichero_a_ejecutar
```

Ejemplo: `sqlplus albdgxxx/xxxxxxxxxx@bd @prueba`

## 7.5 Inserción de comentarios

Existen varias formas para insertar comentarios en ficheros SQL:

- Mediante la sentencia **REM** o doble guión (--):

```
REM Esto es una consulta
-- Segunda línea
```

<sup>5</sup> Cadena de conexión es lo mismo que servicio, en nuestro caso será BD.

```
Select * from practbd.envios;
```

- Mediante `/* ..... */`, permite poner comentarios en medio de sentencias SQL. Por ejemplo, es válida la siguiente sentencia:

```
Select * /* toda la informacion */  
From practbd.envios /* nombre de la tabla */;
```

### 7.6 Volcado del resultado de una consulta

Mediante el comando **spool**. Este comando una vez que está activado, envía al fichero que se haya designado toda la información que se esté mostrando por pantalla hasta que se desactive el spool:

SQL> spool d:\practbd\salida (Crea un fichero de spool llamado salida.lst y manda allí toda lo que aparezca en pantalla a partir de este momento)

SQL > start prueba (Las operaciones que se vayan efectuando envían también su salida al fichero de spool)

SQL> spool (El comando spool sin parámetros nos dice si existe un fichero activo y su nombre)

SQL > spool off (Desactiva el spool y cierra el fichero)

Por defecto se añade la extensión **".lst"** a los ficheros generados con el comando SPOOL, pero puede indicarse explícitamente otra extensión, como **".txt"**