

Capítulo 1

SQL*Plus

SQL*Plus es una herramienta que nos ofrece Oracle para interactuar con sus bases de datos. Existen varias versiones de este producto, por lo que podemos encontrar la herramienta `sqlplus`, que es interactiva pero sólo en modo texto, o `SQL*Plus WorkSheet` que ofrece una interfaz más amigable en un entorno gráfico. Esta última se puede arrancar en el entorno Linux de docencia (máquina `oracle`) mediante el comando `oemapp worksheet`.

La mayoría de la personalización del entorno, para todas las versiones, se hace a través de parámetros que pueden ser visualizados y modificados. Para ver los parámetros se usa la orden `SHOW` (`SHOW ALL` muestra todos los parámetros) y para modificarlos se utiliza `SET <parametro> <valor>`.

En este capítulo estudiaremos las operaciones más comunes, así como algunas de las opciones de personalización del entorno, indicando los comandos necesarios o las opciones de menú o ventanas de diálogo disponibles para cada caso.

1.1. Conexión y desconexión a una base de datos

La conexión a una base de datos requiere varios parámetros:

login: Es el nombre de usuario que se conecta.

password: La clave (secreta) del usuario.

Servicio: El nombre del servicio o base de datos al que se desea conectar. Si no se indica ninguna, se conectará a la base de datos por defecto del sistema.

Conectar como: Indica qué papel va a realizar el usuario cuando se conecte. Puede ser *normal*, *SYSOPER* o *SYSDBA*. El modo normal se usa para trabajar contra la base de datos como un usuario normal, mientras que los otros dos se usan para tareas de administración de la base de datos. que los otros dos lo hacen con

En `SQL*Plus WorkSheet` se realiza la conexión pulsando el botón con el dibujo de un enchufe, o bien en la opción del menú, y abre el cuadro de diálogo de la Figura 1.1 en el que se pueden introducir los cuatro parámetros.

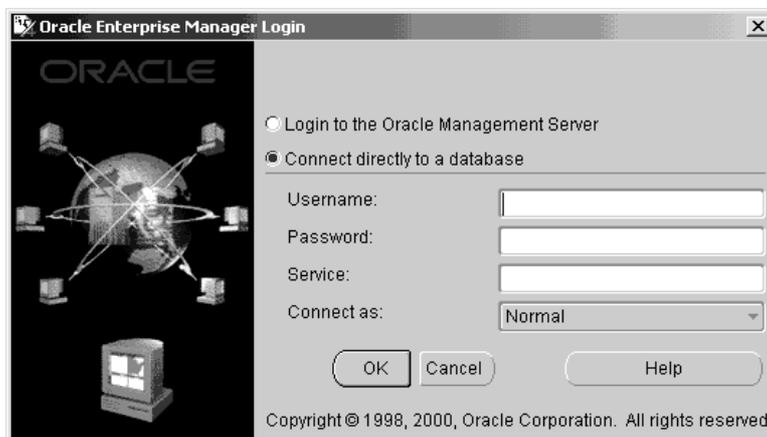


Figura 1.1: Ventana de entrada a SQL*Plus WorkSheet

En `sqlplus` se indicará con el comando `CONNECT`, que tiene la siguiente sintaxis:

```
CONNECT [usuario/clave[@servicio] [AS {SYSOPER|SYSDBA}]]
```

En este caso, sólo la palabra `CONNECT` es imprescindible. Si no se indica nada más, el sistema pide el login y el password, y se conecta como un usuario normal a la base de datos global (base de datos o servicio por defecto) de Oracle. El comando `CONNECT` también se puede hacer en `SQL*Plus WorkSheet`, pero no de forma interactiva. Es decir, se admitirían las tres últimas sentencias del siguiente ejemplo, pero no la primera ya que requeriría que el usuario teclease el login y el password, y en `SQL*Plus WorkSheet` este tipo de interactividad no es posible, ya que el usuario teclea los comandos en una ventana y la respuesta se da en otra.

Ejemplos de conexiones:

```
CONNECT          -- Pide login/password y se conecta a la BD por defecto,
                -- como usuario normal

CONNECT scott/tiger -- Se conecta como usuario scott con permisos normales
                -- a la BD por defecto

CONNECT SYSTEM/manager@EMPRESA -- Se conecta como usuario SYS con permisos
                -- normales la BD por EMPRESA

CONNECT SYSTEM/manager@EMPRESA AS SYSDBA -- Se conecta como usuario SYS con
                -- permisos de DBA a la BD por EMPRESA
```

Usando `sqlplus` los parámetros de conexión se pueden pasar directamente como argumentos del programa, en la línea de comandos. Así, ejecutando `sqlplus SYSTEM/manager@EMPRESA` se ejecutaría el programa seguido de la tercera sentencia de conexión del ejemplo anterior.

Para desconectarse del sistema (sin salir del programa), usaremos el comando `DISCONNECT`. Para abandonar el programa existe el comando `QUIT`, y para `SQL*Plus WorkSheet` hay la opción `Exit` en el menú `File`.

En la máquina `oracle` de docencia se ha implementado el control de acceso a la base de datos utilizando los usuarios del sistema operativo. Por ello, para entrar en `sqlplus` se hará

```
$ sqlplus /
```

Si se elige `SQL*Plus WorkSheet`, se deben dejar en blanco todos los campos, lo que es equivalente al comando `CONNECT /`.

1.2. Guardar una sesión con SQL*Plus

Si deseamos guardar tanto lo que nosotros tecleamos como la respuesta de `SQL*Plus` en una sesión de usuario (o parte de ella), utilizaremos el comando `SPOOL`:

- `SPOOL <nombre_fichero>`: Empieza a grabar, en el fichero indicado (si no se pone extensión, se añade `.lst` automáticamente), tanto lo que el usuario teclea como lo que el sistema responde. El nombre del fichero, `<nombre_fichero>` puede incluir un path absoluto (`c:\sesion.lst` o `/home/scott/sesion.lst`), un path relativo, o simplemente el nombre del fichero (`sesion.lst`). En este caso, el fichero se guarda en el directorio en donde se arranca por defecto `SQL*Plus WorkSheet` (en Windows NT) o en el directorio desde el que se llama al programa si se arranca desde una línea de comandos.

Si el fichero `<nombre_fichero>` existe, se sobrescribe.

- `SPOOL OFF`: Detiene la grabación y cierra el fichero en el que se estaba grabando.
- `SPOOL OUT`: Detiene la grabación, cierra el fichero y lo manda a la impresora por defecto del sistema.

Además, en la herramienta `SQL*Plus WorkSheet`, en el menú `Edit` existe la opción `Save Output As...` que guarda todo lo que el sistema ha respondido a nuestros comandos (es decir, el contenido de la ventana inferior), y `Save Input As...` que guarda el contenido de la ventana superior, es decir, la consulta que se está editando actualmente (o la que se acaba de ejecutar si no se ha borrado). Existe también la opción de borrar el contenido de esas ventanas.

1.3. Configurar el tamaño de la pantalla/página

SQL*Plus nos permite visualizar los datos de una consulta en pantalla, pero también nos permite componer informes relativamente complejos que podemos ver y/o imprimir. Así, el tamaño de la pantalla o página va a determinar el aspecto final del mismo: el número de columnas (ancho de página o tamaño de línea) va a determinar el aspecto de las tablas (si la línea es más larga que el tamaño indicado, la tabla será muy difícil de leer), y el número de filas (tamaño de página) va a determinar donde van las cabeceras y pies de página, además del hecho de que si una tabla (consulta) ocupa más de una página, la cabecera con los nombres de los atributos aparecerá al principio de cada página.

El tamaño de la pantalla/página se controla con dos parámetros: `LINESIZE` y `PAGESIZE` que controlan, respectivamente, el tamaño de las líneas (número de columnas) y el tamaño de la página (número de líneas). El tamaño actual en un momento dado se puede ver con la orden `SHOW`:

```
SHOW LINESIZE
SHOW PAGESIZE
```

Para modificar estos valores, se utilizará `SET`:

```
SET LINESIZE 132
SET PAGESIZE 80
```

Adicionalmente, existe el parámetro `PAUSE` (para ver su estado: `SHOW PAUSE`) que, si se activa, hará una pausa antes de mostrar cada “pantallazo” de información. Este parámetro es realmente útil sólo en la herramienta `sqlplus` en modo texto, ya que `SQL*Plus WorkSheet` dispone de barras de scroll que nos permite ver las páginas anteriores. En `sqlplus` se debe pulsar `Enter` para mostrar la siguiente página (nota: antes de la primera página también, ya que la pausa la hace antes de mostrar nada). Para activar la pausa se usará

```
SET PAUSE ON
```

y para desactivarla,

```
SET PAUSE OFF
```

1.4. Edición y ejecución de comandos

En `SQL*Plus` hay dos tipos de comandos: los propios de `SQL*Plus`, y sentencias `SQL`. Para ejecutar cualquiera de ellas, hay diferencias si el entorno es `sqlplus SQL*Plus WorkSheet`. Veamos como ejecutar ambas clases, primero de uno en uno y luego un conjunto de comandos a la vez (normalmente a partir de un fichero).

1.4.1. Ejecución de comandos individuales

Si se desea ejecutar directamente un comando propio de `SQL*Plus` (por ejemplo, `SET PAUSE ON`) se puede teclear directamente. Utilizando `sqlplus`, el comando debe ir en una sola línea, y se ejecuta al pulsar `ENTER`. En `SQL*Plus WorkSheet` puede ocupar más de una línea y se ejecuta pulsando `Control+ENTER`, pulsando el segundo botón (con el dibujo de un rayo), o en la opción correspondiente del menú.

Las sentencias `SQL` pueden ocupar más de una línea en cualquiera de los entornos. Sin embargo, no puede haber líneas en blanco dentro de una sentencia. Para finalizarla hay varias formas:

1. Una barra (`/`), que debe estar obligatoriamente en una nueva línea después de la sentencia, y ser el único carácter de esa línea.

```
SELECT *
  FROM JUGADOR
/
```

2. Un punto y coma (`;`) al final de la misma, sin importar que esté en una línea aparte o al final de la última línea. Suele ser la forma preferida:

```
SELECT *
  FROM JUGADOR;
```

3. Acabar la sentencia sin ningún terminador. En `sqlplus` se deberá pulsar `Enter` con una línea en blanco para que devuelva el prompt `\tt SQL>` y podamos introducir más comandos.

```
SELECT *
FROM JUGADOR
```

Las alternativas 1 y 2 terminan la sentencia. En `sqlplus`, al pulsar `Enter`, esta se ejecuta inmediatamente. En `SQL*Plus WorkSheet` hay que ejecutarla pulsando `Control+ENTER`, o bien usando el botón o la opción de menú. En cambio, la alternativa 3 sólo almacena la sentencia en el buffer de SQL. Para ejecutarla, se tecleará `RUN` (o `R`, o `/`) seguido en `ENTER` en `sqlplus`, o ejecutando de la forma usual en `SQL*Plus WorkSheet`.

Nota: En `sqlplus`, mientras escribimos la sentencia, irán apareciendo en la parte izquierda números correlativos, que simplemente indican el número de línea de la consulta. Si pulsamos `ENTER` dos veces y siguen apareciendo números, quiere decir que la sentencia no se ha dado por acabada. Puede ser que esto suceda incluso tecleando el punto y coma o la barra, si nos hemos equivocado en alguna cosa en la sentencia (este comportamiento es intencional para algunos casos, como para escribir bloques de código PL/SQL). Para terminar la sentencia, teclearemos un punto (.) sólo en una línea y pulsaremos `ENTER`. Esto debería finalizar la sentencia.

Finalmente debemos indicar que en el buffer de SQL se almacena la última sentencia SQL (no los comandos de `SQL*Plus`). Para ver su contenido usaremos el comando `L[IST]`, y para ejecutarla, `/` o `R[UN]`, aunque esto a veces no funcione de forma muy correcta en el entorno `SQL*Plus WorkSheet`.

La propia edición de la consulta desde el entorno `sqlplus` no es nada amigable, ya que es un editor de líneas demasiado básico. Por ello, se ofrece la posibilidad de teclear las consultas desde un editor externo, que se invoca con la orden

```
ED[IT]
```

para la consulta actual, o

```
ED[IT] <nombre_fichero>
```

para crear o modificar la consulta en un nuevo fichero. Desde el editor de textos se teclea la consulta y se graba, y al salir la consulta se encuentra en el buffer de SQL, para poder ser ejecutada o guardada (o modificada de nuevo).

El editor por defecto en entornos Windows suele ser el Block de notas (`NOTEPAD.EXE`), y en entornos Linux/Unix, el que se indica en la variable de entorno `EDITOR`, o `vi` si esa variable no está definida. Para cambiar el editor que se usa desde `sqlplus` utilizaremos la orden

```
DEFINE_EDITOR=<nombre_del_editor_preferido>
```

Para cargar en el buffer de SQL una sentencia escrita en un fichero de texto, usaremos el comando `GET`:

```
GET <nombre_fichero>
```

Se supone que el fichero tiene extensión `.sql` si no se indica otra distinta.

```
GET consulta.sql          -- Carga consulta.sql
GET consulta              -- Carga consulta.sql
GET consulta.txt          -- Carga consulta.txt
```

A continuación de cargar la sentencia, podremos ver su contenido (`LIST`) o ejecutarla (`RUN`).

Para guardar una consulta (sólo el código SQL) usaremos la orden `SAVE`, que también supone que tendrá una extensión `.sql` si no se indica explícitamente otra.

```
SAVE <fichero> [{CREATE|REPLACE|APPEND}]
```

La opción `CREATE` es la opción por defecto, y crea el fichero si no existe. Si existe, dará un error. Para sobrescribirlo deberemos usar la opción `REPLACE`. La opción `APPEND` añade la consulta al final del fichero indicado.

En el entorno `SQL*Plus WorkSheet` estas acciones están también disponibles en las opciones de los menús.

1.4.2. Ejecución de secuencias de comandos

Una secuencia de comandos estará formada por varios comandos, bien `SQL*Plus` o bien `SQL`, siempre acabados por un punto y coma (en este caso, también los comandos de `SQL*Plus` deben acabar en `;`).

No es posible conseguir directamente una secuencia de comandos en `sqlplus`, pero sí en `SQL*Plus WorkSheet`. Para ejecutarla se sigue el procedimiento normal.

En ambos entornos podremos ejecutar directamente un fichero o *script* que contiene una secuencia de comandos. Para ello, usaremos cualquiera de los siguientes comandos (son equivalentes):

```
@fichero
STA[RT] fichero
```

Suponiendo que nuestras consultas están almacenadas en un fichero llamado `fichero.sql`, como en casos anteriores, la extensión `.sql` se sobreentiende si no se indica otra. Además, en `SQL*Plus WorkSheet` existe una opción del menú que realiza esta acción.

Cuando se ejecutan las sentencias de un archivo de esta forma, sólo salen los resultados, pero no las sentencias. Si deseamos que también salgan las sentencias, lo indicaremos con el comando `SET ECHO ON` (para desactivarlo, usaremos `SET ECHO OFF`). Así, la siguiente secuencia de comandos ejecuta una colección de sentencias almacenadas en el fichero `consultas.sql` y guarda el resultado, formado por cada sentencia seguida del resultado que produce, en el fichero `resultado.lst`:

```
SET ECHO ON
SPOOL resultado
START consultas
SPOOL OFF
```

1.5. Formatear columnas

Cuando se ejecuta una sentencia `SELECT`, la cabecera de cada columna del resultado tendrá el nombre que se indica en la sentencia, y usará el formato del tipo de datos correspondiente. Podemos cambiar estas y otras con el comando `COLUMN`. Debemos hacer notar que este comando es de `SQL*Plus` y no de `SQL`, y toma efecto desde que se ejecutan hasta que se anulan o el usuario se desconecta. No se indica aquí la sintaxis completa del comando `COLUMN`, sino solamente algunas opciones para modificar los parámetros indicados. Para estos, la sintaxis es la siguiente:

```
COLUMN <nombre_columna> [HEADING <cabecera>] [FORMAT <formato>]
```

donde `<nombre_columna>` indica una de las columnas (expresiones) que se seleccionarán en una `SELECT` posterior. La `<cabecera>` es una constante de texto e irá entre comillas simples. Si queremos que la cabecera ocupe más de una línea, se usa la barra vertical (`|`) como indicador de cambio de línea. Entre los `<formato>`s existentes, podemos destacar los siguientes:

- Para valores numéricos: tantos 9's como la anchura máxima deseada. Podemos incluir entre los nueve como separador decimal (mediante `D` o `.`), el separador de grupos (`G`), y/o hacer que el símbolo de dólar (`$`) o el símbolo local para la moneda (`L`) preceda al número.
- Para valores alfanuméricos: `An`, donde `n` es un número que especifica la anchura máxima del campo. Se puede indicar adicionalmente el parámetro `WORD_WRAPPED` para indicar que si es posible salte de línea por los espacios en blanco.

Ejemplos:

```
COLUMN SALARIO FORMAT L9G999D99
COLUMN NOMBRE_JUGADOR HEADING 'Nombre del Jugador' FORMAT A15 WORD_WRAPPED
```

1.6. Propiedades de los objetos de la base de datos: el catálogo

No hay una forma directa de describir todos los objetos de la base de datos (tablas, vistas, índices, triggers, etc.) desde `SQL*Plus`. Sin embargo, podemos acceder a la mayoría de las propiedades de estos objetos mediante consultas al catálogo de la base de datos, y utilizando alguna orden adicional.

El **catálogo** o diccionario de datos de una base de datos es un conjunto de tablas que contienen metadatos, es decir, datos sobre los datos: nombres de tablas, vistas, índices, procedimientos, etc. En Oracle, la tabla `DICTIONARY` o su sinónimo `DICT` nos da el nombre y una descripción de las tablas del catálogo. Usando las tablas del catálogo, así como alguna orden adicional de `SQL*Plus`, podemos obtener información sobre nuestras tablas o vistas:

- La vista del sistema `ALL_TABLES` contiene información sobre todas las tablas de la base de datos, y la vista `USER_TABLES`, sobre las tablas de las que el usuario es dueño. Así podemos descubrir, entre otras cosas, los nombres de las tablas y sus dueños (creadores). Para las vistas existen las equivalentes `ALL_VIEWS` y `USER_VIEWS`. Usando `USER_TAB_COLUMNS` o su sinónimo `COLS` podemos saber algo más de las columnas definidas en nuestras tablas, como por ejemplo el valor por defecto de una columna, si lo tiene.
- `DESC[RIBE] <tabla>`: nos muestra sus atributos, indicando el nombre, el tipo y si admite o no valores nulos. Este comando también se puede utilizar con las vistas.
- La vista `ALL_CONSTRAINTS` contiene todas las restricciones (claves primarias, foráneas, etc.) de la base de datos, y `USER_CONSTRAINTS` las creadas por el usuario actual.