

GUIA DE LABORATORIO NRO1

Características del SQL *PLUS

- SQL *PLUS, es una herramienta indispensable para un administrador de base de datos.
- Es un entorno en modo texto y no contiene un motor PL/SQL local, es decir, las instrucciones o comandos se envían directamente a la base de datos.
- SQL *PLUS quizás sea la más sencilla de las herramientas de desarrollo de PL/SQL. Permite al usuario introducir instrucciones SQL y bloques PL/SQL de forma interactiva mediante una línea de comandos.
- Generalmente, SQL *PLUS se distribuye junto con el servidor de Oracle y se encuentra disponible como parte de la instalación estándar de Oracle.
- Dado que SQL *PLUS es un tema tan amplio, que no se puede estudiar en una sola sesión de clases, se van a estudiar principalmente las funciones que interesan a los DBA (database administrator).

Entorno del SQL *PLUS

El SQL *PLUS tiene un entorno orientado al carácter.



```
F:\oracle\ora90\bin\SQLPLUS.exe

SQL*Plus: Release 9.0.1.0.1 - Production on Tue Mar 21 19:26:16 2003

(c) Copyright 2001 Oracle Corporation. All rights reserved.

Introduzca el nombre de usuario: scott
Introduzca la contraseña:

Conectado a:
Oracle9i Enterprise Edition Release 9.0.1.1.1 - Production
With the Partitioning option
JServer Release 9.0.1.1.1 - Production

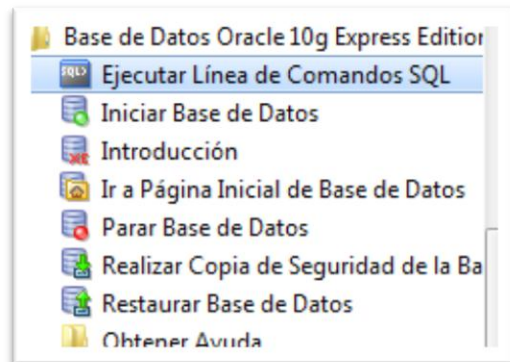
SQL> _
```

Conexión con la Base de Datos

Antes de realizar cualquier instrucción en la base de datos, es necesario establecer una conexión con el servidor de datos. Esta operación se puede realizar de una de las siguientes formas:

Mediante el paso de un identificador de usuario y una contraseña y/o una cadena de conexión en la línea de comandos utilizada para iniciar SQL *PLUS para el entorno orientado al carácter.

Una primera forma sería después de ejecutar el SQL *PLUS desde el explorador de Windows:



SQL*Plus: Release 11.2.0.1.0 Production on Lun Ago 1 19:13:09 2011
Copyright (c) 1982, 2010, Oracle. All rights reserved.

Introduzca el nombre de usuario: SCOTT

Introduzca la contraseña: TIGER

Conectado a:

Oracle11g Enterprise Edition Release 9.0.1.1.1 - Production with the partitioning
option Jserver Release 9.0.1.1.1 - Production

O también especificando la cadena de conexión:

SQL*Plus: Release 11.2.0.1.0 Production on Lun Ago 1 19:13:09 2011
Copyright (c) 1982, 2010, Oracle. All rights reserved.

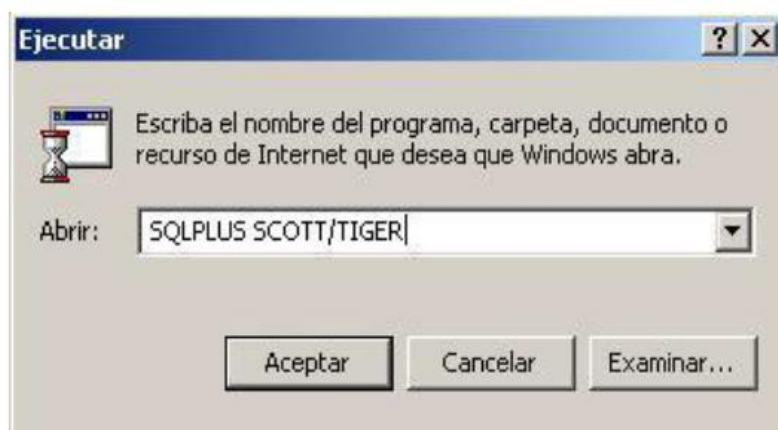
Introduzca el nombre de usuario: SCOTT/TIGER@CIBERTEC

Conectado a:

Oracle11g Enterprise Edition Release 9.0.1.1.1 - Production with the
partitioning option Jserver Release 9.0.1.1.1 - Production

SQL>

Una segunda forma sería iniciando el SQL *PLUS desde el menú ejecutar del botón de inicio:



Mediante el uso del comando CONNECT, una vez dentro de SQL*PLUS (en ambos entornos).

```
SQL> CONNECT
Introduzca el nombre de usuario: SCOTT
Introduzca la contraseña: ****
Conectado.
SQL>
```

Comandos de Configuración de Sesión

Existe un grupo de comandos que nos permiten configurar el SQL *PLUS mientras estamos conectados a la base de datos. Algunos de los comandos se muestran en la siguiente tabla:

COMANDO	ABREVIATURA	DESCRIPCION
SET COMPATIBILITY	SET COM V7 / V8 native	Configura la compatibilidad con la versión de oracle espec.
SET NUMFORMAT	SET NUM \$ 999.999.99	Configura el formato numérico de salida de las consultas
SET PAGESIZE	SET PAGES 250	Configura el número de líneas por paginas
SET LINESIZE	SET LINE 250	Configura el número total de caracteres por línea
SET SERVEROUTPUT {ON / OFF} [SIZE n] [FORMAT (wrapped / truncate/word_wrapped)]	SET SERVEROUT ON	Activa o desactiva la salida en pantalla provocadas por los procedimientos almacenados o bloques PL/SQL en SQL*PLUS que utilizan el paquete DBMS_OUTPUT.PUT_LINE
SET EDITFILE	SET EDITF "file.sql"	Permite cambiar el nombre del archivo que abre y crea el editor de comandos del sistema operativo

Por ejemplo, el comando LINESIZE nos permite determinar cuántos caracteres se mostrarán por cada línea de texto que devuelven las consultas a la base de datos. La cantidad por defecto es 80 caracteres. Para poner en práctica el efecto del comando, hagamos la siguiente consulta:

```
SQL> SELECT * FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30

...ésta será la salida por pantalla. Ahora, si aumentamos la cantidad de caracteres por línea de texto, el resultado de la consulta se mostraría más ordenado.

```
SQL> SET LINESIZE 200
SQL> SELECT * FROM emp;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17/12/80	800		20
7499	ALLEN	SALESMAN	7698	20/02/81	1600	300	30
7521	WARD	SALESMAN	7698	22/02/81	1250	500	30
7566	JONES	MANAGER	7839	02/04/81	2975		20

14 filas seleccionadas.

Por ejemplo, el comando PAGESIZE configura el número de línea por página. Para poner en práctica el efecto del comando, hagamos la siguiente consulta:

```
SQL> SET PAGESIZE 250;
SQL> SELECT * FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

14 rows selected.

```
SQL>
```

Ejecutando archivos de comandos desde SQL *PLUS

Para ejecutar archivos que contengan comandos válidos, se utiliza el comando START o el carácter @. Ejemplo:

```
SQL> START @unidad:\ruta\mi_archivo;
```

o también:

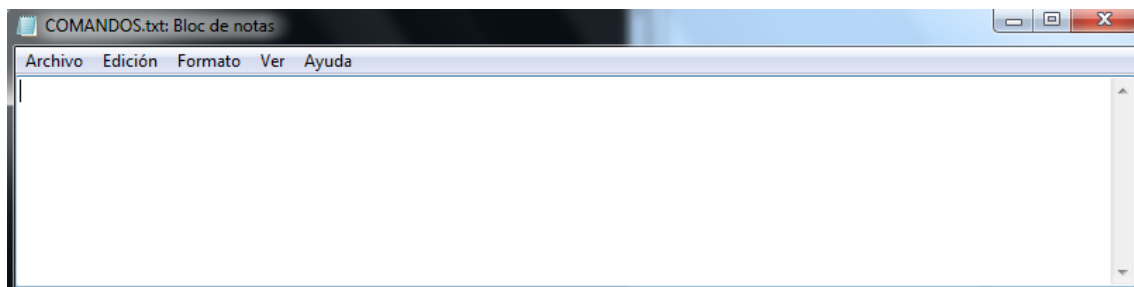
```
SQL> @unidad:\ruta\mi_archivo;
```

La extensión del archivo por defecto es .sql. Podemos utilizar archivos con extensión diferente al por defecto (cualquier archivo que pueda ser leído con un editor de texto) pero debemos especificar la extensión en el comando de ejecución. Por ejemplo, para ejecutar un archivo con extensión .txt: este sería el comando:

```
SQL> @unidad:\ruta\mi_archivo.txt;
```

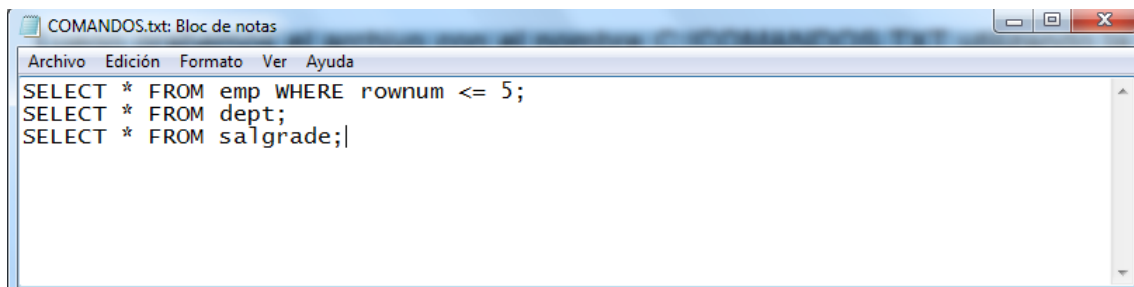
Hagamos el siguiente caso práctico. Vamos crear un archivo llamado comandos.txt utilizando el editor de Windows.

```
SQL> EDIT COMANDOS.TXT
```



Editamos el buffer (comandos.sql) ingresando los siguientes comandos SQL:

```
SELECT * FROM emp WHERE rownum <= 5;  
SELECT * FROM dept;  
SELECT * FROM salgrade;
```



Luego grabemos el archivo con el nombre COMANDOS.TXT utilizando la ruta:
C:/DATA/

Ahora ejecutamos el archivo con el comando START o con el carácter @ de la siguiente forma:

SQL> START C:/DATA/comandos.txt o también SQL> @ C:/DATA/comandos.txt

```
SQL> START C:/DATA/COMANDOS.TXT;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Si ha creado carpetas cuyo nombre tengan espacio en blanco o el nombre del archivo también tenga espacios en blanco, debe encerrar en apóstrofes el archivo a ejecutar.

Ejemplo:

SQL> @'c:\curso lp6\mis comandos.txt';

Uso comando SPOOL en SQL *PLUS

Este comando es de gran utilidad para un operador del SQL *PLUS. Pues este comando permite registrar, en un archivo especificado, los resultados de nuestras consultas y operaciones realizadas a la base de datos. No solamente registra los resultados sino que también registra los comandos utilizados. La sintaxis es la siguiente:

SQL> SPOOL [nombre_archivo[.extension]]

Ejemplo: los siguientes comandos serán registrados en el archivo sesion02.txt cuando se cierre el spool a través del comando SPOOL OFF.

```
SQL> SPOOL C:/DATA/SESION02.TXT;
SQL> SET LINESIZE 150;
SQL> SELECT * FROM EMP;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

```

14 rows selected.

SQL> SELECT * FROM BONUS;

no rows selected

SQL> SPOOL OFF;
SQL>
```

Cuando se crea el archivo sesion02.txt, este archivo se mantendrá abierto y vacío, hasta que no se especifique que se cierre el spool (SPOOL OFF). Si usted no especifica la extensión al archivo a través del comando SPOOL automáticamente el archivo se le asignará la extensión LST.

Si editamos el archivo sesion02.txt creado con el comando spool encontraremos lo siguiente:

```
SQL> EDIT C:/DATA/sesion02.txt
```

Si usted vuelve a crear el archivo sesion02.txt con el comando SPOOL, para nuestro caso el archivo ya existe, lógicamente el archivo se volverá a crear perdiéndose el contenido del mismo. Tenga cuidado al especificar el nombre del archivo. Asegúrese de que el archivo a crear no exista.