PRACTICAS.

TRABAJOS.

1. **Identificar los procesos de sistema operativo que corresponden a la base de datos, ¿existe algún proceso coordinador de trabajos -cjqNNN- corriendo en la instancia?.**

```
/home/CURSO/curso01 (CURSO01)> ps -ef|grep cj
curso01  25608 25524  0 13:11 pts/2    00:00:00 grep cj
```

2. **Ver el contenido del paquete dbms_scheduler.**

Revisar documentación en linea: "Oracle Database PL/SQL Packages and Types Reference".

3. **Ver la descripción de las vistas dba_scheduler_jobs y dba_scheduler_job_log. Buscar la descripción de cada uno de los campos en la documentación en línea.**

```
SQL> desc dba_scheduler_jobs
 Nombre                                    ?Nulo?   Tipo
 ----------------------------------------- -------- ----------------------------
 OWNER                                     NOT NULL VARCHAR2(30)
 JOB_NAME                                  NOT NULL VARCHAR2(30)
 JOB_SUBNAME                                        VARCHAR2(30)
 JOB_CREATOR                                        VARCHAR2(30)
 CLIENT_ID                                          VARCHAR2(64)
 GLOBAL_UID                                         VARCHAR2(32)
 PROGRAM_OWNER                                      VARCHAR2(4000)
 PROGRAM_NAME                                       VARCHAR2(4000)
 JOB_TYPE                                           VARCHAR2(16)
 JOB_ACTION                                         VARCHAR2(4000)
 NUMBER_OF_ARGUMENTS                                NUMBER
 SCHEDULE_OWNER                                     VARCHAR2(4000)
 SCHEDULE_NAME                                      VARCHAR2(4000)
 SCHEDULE_TYPE                                      VARCHAR2(12)
 START_DATE                                         TIMESTAMP(6) WITH TIME ZONE
 REPEAT_INTERVAL                                    VARCHAR2(4000)
 EVENT_QUEUE_OWNER                                  VARCHAR2(30)
```

```
EVENT_QUEUE_NAME                        VARCHAR2(30)
EVENT_QUEUE_AGENT                       VARCHAR2(30)
EVENT_CONDITION                         VARCHAR2(4000)
EVENT_RULE                              VARCHAR2(65)
END_DATE                                TIMESTAMP(6) WITH TIME ZONE
JOB_CLASS                               VARCHAR2(30)
ENABLED                                 VARCHAR2(5)
AUTO_DROP                               VARCHAR2(5)
RESTARTABLE                             VARCHAR2(5)
STATE                                   VARCHAR2(15)
JOB_PRIORITY                            NUMBER
RUN_COUNT                               NUMBER
MAX_RUNS                                NUMBER
FAILURE_COUNT                           NUMBER
MAX_FAILURES                            NUMBER
RETRY_COUNT                             NUMBER
LAST_START_DATE                         TIMESTAMP(6) WITH TIME ZONE
LAST_RUN_DURATION                       INTERVAL DAY(9) TO SECOND(6)
NEXT_RUN_DATE                           TIMESTAMP(6) WITH TIME ZONE
SCHEDULE_LIMIT                          INTERVAL DAY(3) TO SECOND(0)
MAX_RUN_DURATION                        INTERVAL DAY(3) TO SECOND(0)
LOGGING_LEVEL                           VARCHAR2(4)
STOP_ON_WINDOW_CLOSE                    VARCHAR2(5)
INSTANCE_STICKINESS                     VARCHAR2(5)
RAISE_EVENTS                            VARCHAR2(4000)
SYSTEM                                  VARCHAR2(5)
JOB_WEIGHT                              NUMBER
NLS_ENV                                 VARCHAR2(4000)
SOURCE                                  VARCHAR2(128)
DESTINATION                             VARCHAR2(128)
COMMENTS                                VARCHAR2(240)
FLAGS                                   NUMBER

SQL> desc dba_scheduler_job_log
 Nombre                                  ?Nulo?   Tipo
 --------------------------------------- -------- ----------------------------
 LOG_ID                                  NOT NULL NUMBER
 LOG_DATE                                         TIMESTAMP(6) WITH TIME ZONE
 OWNER                                            VARCHAR2(30)
 JOB_NAME                                         VARCHAR2(65)
 JOB_SUBNAME                                      VARCHAR2(65)
 JOB_CLASS                                        VARCHAR2(30)
 OPERATION                                        VARCHAR2(30)
 STATUS                                           VARCHAR2(30)
 USER_NAME                                        VARCHAR2(30)
 CLIENT_ID                                        VARCHAR2(64)
```

GLOBAL_UID VARCHAR2(32)
ADDITIONAL_INFO CLOB

4. **Como usuario SYSTEM crear un trabajo que llamará al procedimiento descrito seguidamente, también propiedad del SYSTEM, que permite analizar el esquema de un cierto numero de usuarios y generar las estadísticas internas usadas por el analizador sintáctico de Oracle (previamente deben crearse la tabla "usuarios_estadisticas" y el procedimiento "analiza_usuarios").**
   Se ejecutara en el momento de su creación y con periodicidad semanal.

```
create table usuarios_estadisticas (username varchar2(30) not null,
      fecha date,
      error varchar2(80))
      tablespace users
      storage (initial 16K next 16K maxextents 10);


CREATE OR REPLACE procedure analiza_usuarios as
  cursor usuarios is
     select username,rowid from system.usuarios_estadisticas;
     werror varchar2(80);
     wrowid urowid;
  begin
     FOR rec_usuarios IN usuarios LOOP
     wrowid:=rec_usuarios.rowid;
     begin
        DBMS_UTILITY.ANALYZE_SCHEMA(rec_usuarios.username,'ESTIMATE',
        NULL,3);
        update system.usuarios_estadisticas
           set fecha=sysdate, error=null
           where rowid=rec_usuarios.rowid;
     exception
        when others then
           werror:=rpad(sqlerrm,80);
           update system.usuarios_estadisticas
              set error=werror, fecha=sysdate
              where rowid=wrowid;
     end;
     END LOOP;
  end analiza_usuarios;
/
```

Crear un "script" llamado crea_trabajo1.sql, por ejemplo, con el contenido siguiente:

```
/home/CURSO/curso01 (CURSO01)> vi crea_trabajo_1.sql

BEGIN
      DBMS_SCHEDULER.CREATE_JOB(
           job_name => 'TRABAJO_ANALIZA',
           job_type => 'STORED_PROCEDURE',
           job_action => 'ANALIZA_USUARIOS',
           start_date => SYSDATE,
           repeat_interval => 'FREQ = WEEKLY');
END;
/
```

Ejecutarlo desde sqlplus:

```
SQL> @crea_trabajo1.sql

Procedimiento PL/SQL terminado correctamente.
```

5. **Comprobar en las vistas la información sobre el trabajo "TRABAJO_ANALIZA". ¿Está el trabajo habilitado o no?, en caso de estar deshabilitado ... habilitar el trabajo para su ejecución y comprobar de nuevo sus características.**

```
SQL> select owner, job_name, job_creator, client_id, job_type, start_date,
repeat_interval, end_date, auto_drop, enabled, restartable from
dba_scheduler_jobs  where owner='SYSTEM' and job_name ='TRABAJO_ANALIZA';

OWNER                          JOB_NAME                       JOB_CREATOR
------------------------------ ------------------------------ ------------------------
CLIENT_ID                                                     JOB_TYPE
----------------------------------------------------------------------------------------
START_DATE
----------------------------------------------------------------------------------------
REPEAT_INTERVAL
----------------------------------------------------------------------------------------
END_DATE                                                      AUTO_ ENABL RESTA
----------------------------------------------------------------------------------------
SYSTEM                         TRABAJO_ANALIZA                SYSTEM
                                                              STORED_PROCEDURE
08/12/06 14:41:19,000000 +01:00
FREQ = WEEKLY
```

*TRUE **FALSE** FALSE*

Como se observa en la consulta, el trabajo está deshabilitado (columna ENABLED igual a FALSE). También se observa que el trabajo será borrado cuando se complete (AUTO_DROP igual a TRUE).

Se habilita el trabajo para su ejecución:

```
/home/CURSO/curso01 (CURSO01)> vi activa_trabajo1.sql
BEGIN
DBMS_SCHEDULER.ENABLE('TRABAJO_ANALIZA');
END;
/
~
"activa_trabajo1.sql" 4L, 55C escritos
```

Se lanza el procedimiento DBMS_SCHEDULER.ENABLE y se comprueba las características del trabajo:

```
SQL> @activa_trabajo1.sql
Procedimiento PL/SQL terminado correctamente.
```

```
SQL> select owner, job_name, job_creator, client_id, job_type, start_date, repeat_interval, end_date, auto_drop, enabled, restartable from dba_scheduler_jobs  where owner='SYSTEM' and job_name ='TRABAJO_ANALIZA';

OWNER                          JOB_NAME                       JOB_CREATOR
------------------------------------------------------------------------------
CLIENT_ID                                                     JOB_TYPE
------------------------------------------------------------------------------
START_DATE
---------------------------------------------------------------------------
REPEAT_INTERVAL
-------------------------------------------------------------------------------------------
END_DATE                                                        AUTO_ ENABL RESTA
-------------------------------------------------------------------------------------------
SYSTEM                         TRABAJO_ANALIZA                SYSTEM
                                                              STORED_PROCEDURE
08/12/06 14:41:19,000000 +01:00
FREQ = WEEKLY
                                                              TRUE  TRUE  FALSE
```

6. ¿Cuál es el valor para el máximo número de ejecuciones y máximo número de fallos para este trabajo?. ¿Y su número de fallos?.

```
SQL> select job_name, max_runs, max_failures, failure_count from
dba_scheduler_jobs where owner='SYSTEM' and job_name ='TRABAJO_ANALIZA';

JOB_NAME                       MAX_RUNS MAX_FAILURES FAILURE_COUNT
------------------------------------------------------------------------------
TRABAJO_ANALIZA                                                0
```

7. Forzar la ejecución del trabajo "TRABAJO_ANALIZA". Comprobar antes y después de forzar la ejecución el valor para el número de ejecuciones, última fecha de ejecución y duración de la última ejecución ("RUN_COUNT", "LAST_START_DATE" y "LAST_RUN_DURATION" respectivamente).

```
/home/CURSO/curso01 (CURSO01)> vi fuerza_trabajo1.sql
BEGIN
DBMS_SCHEDULER.RUN_JOB('TRABAJO_ANALIZA',FALSE);
END;
/
~
"fuerza_trabajo1.sql" 4L, 62C escritos

SQL> select run_count, to_char(last_start_date,'dd-mm-yyyy hh:mi') FECHA,
last_run_duration DURACION from dba_scheduler_jobs where owner='SYSTEM'
and job_name ='TRABAJO_ANALIZA';

 RUN_COUNT FECHA           DURACION
----------------------------------------------------------
         0

SQL> @fuerza_trabajo1.sql
Procedimiento PL/SQL terminado correctamente.

SQL> select run_count, to_char(last_start_date,'dd-mm-yyyy hh:mi') FECHA,
last_run_duration DURACION from dba_scheduler_jobs where owner='SYSTEM' and
job_name ='TRABAJO_ANALIZA';

 RUN_COUNT FECHA                     DURACION
-------------------------------------------------------------------------------
         1 08-12-2006 04:12          +000000000 00:00:00.168563
```

8. Como usuario "prueba01" crear un trabajo que llamara al procedimiento descrito seguidamente, también propiedad de "prueba01", que permite averiguar cuáles de las tablas propiedad del usuario tienen ocupado más del 80% de las extensiones que le son permitidas (debe tener cuota sobre el espacio de almacenamiento donde se creará la tabla y permiso para crear procedimientos).

Insertará una fila en la tabla "tablas_revision" por cada una de las tablas que cumplan la condición. Se ejecutará cada hora.

Previamente debe crearse la siguiente tabla:

```
CREATE TABLE tablas_revision
        (nombre_tabla  varchar2(30),
        ocupacion  number)
        storage (initial 100k next 100k);
```

¿En que espacio de almacenamiento se ha creado la tabla?, ¿con que parámetros de almacenamiento?.

```
CREATE OR REPLACE procedure chequear_tablas as
extensiones integer;
maximo_extensiones integer;
ocupacion integer;
porcentaje_extensiones integer := 80;
cursor c_tablas is select table_name from user_tables;

begin
  FOR rec_tables IN c_tablas LOOP
    select count(*) into extensiones
    from user_extents
    where segment_name=rec_tables.table_name;

    select max_extents into maximo_extensiones
    from user_tables
    where table_name=rec_tables.table_name;

    ocupacion:=(round(extensiones*100/maximo_extensiones));

    IF (ocupacion > porcentaje_extensiones) THEN
            insert into tablas_revision values (rec_tables.table_name,
ocupacion);
      commit;
```

```
    END IF;

  END LOOP;

  end chequear_tablas;
  /
```

```
SQL> connect prueba01
Introduzca su clave:
Conectado.

SQL> CREATE TABLE tablas_revision
  2  (nombre_tabla  varchar2(30),
  3  ocupacion  number)
  4  storage (initial 100k next 100k);

Tabla creada.
```

Para ver las caracteristicas de almacenamiento de la tabla consultar la vista "user_tables".

Crear un "script" llamado chequear_tablas.sql, por ejemplo, que contenga las sentencias de creacion del procedimiento:

```
SQL> @chequear_tablas.sql

Procedimiento creado.
```

Crear un "script" llamado crea_trabajo2.sql, por ejemplo, con el contenido siguiente:

```
/home/CURSO/curso01 (CURSO01)> vi crea_trabajo2.sql
BEGIN
DBMS_SCHEDULER.CREATE_JOB(
job_name => 'TRABAJO_CHEQUEAR_TABLAS',
job_type => 'STORED_PROCEDURE',
job_action => 'CHEQUEAR_TABLAS',
start_date => SYSDATE,
repeat_interval => 'FREQ = HOURLY; INTERVAL=1',
ENABLED => TRUE,
AUTO_DROP => FALSE,
```

```
COMMENTS => 'Trabajo para chequear tablas cada hora');
END;
/
~
"crea_trabajo2.sql" 12L, 311C escritos
```

```
SQL> @crea_trabajo2.sql
BEGIN
*
ERROR en linea 1:
ORA-27486: privilegios insuficientes
ORA-06512: en "SYS.DBMS_ISCHED", linea 99
ORA-06512: en "SYS.DBMS_SCHEDULER", linea 262
ORA-06512: en linea 2
```

¿Por qué se ha producido el error anterior al intentar crear el trabajo?: el usuario "prueba01" carece de permisos para poder crear trabajos. Es necesario otorgar el permiso "create job".

```
SQL> connect system
Introduzca la contrase?a:
Conectado.

SQL> grant create job to prueba01;
Concesion terminada correctamente.
```

Se vuelve a realizar la conexión como "prueba01" y se ejecuta "crea_trabajo2.sql".

```
SQL> connect prueba01
Introduzca la contrase?a:
Conectado.

SQL>  @crea_trabajo2.sql
Procedimiento PL/SQL terminado correctamente.
```

Se consultan las características del trabajo.

SQL> select job_name, job_creator, client_id, job_type, start_date, repeat_interval, end_date, auto_drop, enabled, restartable from user_scheduler_jobs

```
JOB_NAME                  JOB_CREATOR               CLIENT_ID
JOB_TYPE
------------------------------------------------------------------------------
START_DATE
---------------------------------------------------------------------------
REPEAT_INTERVAL
-------------------------------------------------------------------------------------------
END_DATE                                                  AUTO_ ENABL RESTA
-------------------------------------------------------------------------------------------
TRABAJO_CHEQUEAR_TABLAS   PRUEBA01
STORED_PROCEDURE
08/12/06 17:01:50,000000 +01:00
FREQ = HOURLY; INTERVAL=1
                                                          FALSE TRUE  FALSE
```

9. Obtener, a nivel de sistema operativo, un listado de los procesos asociados a la base de datos.
Forzar la ejecución del trabajo "TRABAJO_CHEQUEAR_TABLAS", mientras está ejecutándose volver a obtener, a nivel de sistema operativo, un listado de los procesos asociados a la base de datos. ¿Qué procesos se observan en ambos casos?.

Se crea un "script" llamado "fuerza_trabajo2.sql" con el contenido siguiente:

```
/home/CURSO/curso01 (CURSO01)> vi  fuerza_trabajo2.sql
BEGIN
DBMS_SCHEDULER.RUN_JOB('TRABAJO_CHEQUEAR_TABLAS',FALSE);
END;
/
~
"fuerza_trabajo2.sql" 4L, 69C escritos
/home/CURSO/curso01 (CURSO01)>
```

Se obtiene el listado de los procesos asociados a la bd.

```
/home/CURSO/curso01 (CURSO01)> ps -ef|grep CURSO01
oracle   18916     1  0 Dec03 ?        00:00:01 ora_pmon_CURSO01
...
```

oracle 27563 1 0 17:01 ? 00:00:00 ora_cjq0_CURSO01
curso01 27616 27594 0 17:17 pts/1 00:00:00 grep CURSO01

Como usuario "prueba01" se lanza el "script" "fuerza_trabajo2.sql":

SQL> show user
USER es "PRUEBA01"

SQL> @fuerza_trabajo2.sql
Procedimiento PL/SQL terminado correctamente.

Se obtiene nuevamente el listado de los procesos asociados a la bd.

/home/CURSO/curso01 (CURSO01)> ps -ef|grep CURSO01
oracle 18916 1 0 Dec03 ? 00:00:01 ora_pmon_CURSO01
...
oracle 27563 1 0 17:01 ? 00:00:00 ora_cjq0_CURSO01
oracle 27618 1 2 17:17 ? 00:00:00 ora_j000_CURSO01
curso01 27620 27594 0 17:17 pts/1 00:00:00 grep CURSO01

Si posteriormente se vuelve a ejecutar esta consulta, se observará que ha desaparecido el proceso esclavo "ora_j000_CURSO01".

10. Como usuario SYSTEM crear un trabajo que llame al procedimiento "monitorizar_usuarios" descrito seguidamente, también propiedad del SYSTEM, que permite monitorizar el numero de sesiones activas de usuario en la instancia. Se ejecutará cada minuto.

```
CREATE OR REPLACE procedure monitorizar_usuarios as
sesiones number(4);
activas number(4);
cursor c_usuarios is
select usuario
from usuarios_monitorizados
where monitorizar = 'S';

begin
FOR rec_usuarios IN c_usuarios LOOP

select nvl(count(*),0) into sesiones
```

```
from v$session
where username=rec_usuarios.usuario
and status in ('ACTIVE','INACTIVE');

select nvl(count(*),0) into activas
from v$session
where username=rec_usuarios.usuario
and status = 'ACTIVE';

insert into usuarios_monitorizados_log
values (rec_usuarios.usuario,sysdate,sesiones,activas);

END LOOP;
commit;
end monitorizar_usuarios;
/
```

Previamente deben crearse las siguientes tablas:

```
CREATE TABLE usuarios_monitorizados_log
        (usuario varchar2(30),
        fecha_log date,
        sesiones number(4),
        activas number(4))
        storage (initial 100k next 100k);

CREATE TABLE usuarios_monitorizados
        (usuario varchar2(30),
        monitorizar char(1),
        descripcion varchar2(80))
        storage (initial 100k next 100k);
```

```
SQL> CREATE TABLE usuarios_monitorizados_log
 (usuario varchar2(30),
fecha_log date,
sesiones number(4),
activas number(4))
storage (initial 100k next 100k);

Tabla creada.

SQL> CREATE TABLE usuarios_monitorizados
 (usuario varchar2(30),
monitorizar char(1),
descripcion varchar2(80))
```

storage (initial 100k next 100k);

Tabla creada.

Conectarse como usuario SYS y dar permisos sobre vista SESSION:

SQL> connect / as sysdba;
Conectado.

SQL> grant select on V_$SESSION to system;
Concesion terminada correctamente.

Crear un "script" llamado monitorizar_usuarios.sql, por ejemplo, que contenga las sentencias de creacion del procedimiento:

SQL> @monitorizar_usuarios.sql
Procedimiento creado.

Crear un "script" llamado crea_trabajo3.sql, por ejemplo, con el contenido siguiente:

/home/CURSO/curso01 (CURSO01)> vi crea_trabajo3.sql

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB(
job_name => 'TRABAJO_MONITORIZAR_USUARIOS',
job_type => 'STORED_PROCEDURE',
job_action => 'MONITORIZAR_USUARIOS',
start_date => SYSDATE,
repeat_interval => 'FREQ = MINUTELY; INTERVAL=1',
ENABLED => TRUE,
AUTO_DROP => FALSE,
COMMENTS => 'Trabajo para monitorizar usuarios');
END;
/
~
"crea_trabajo3.sql" 12L, 314C escritos
```

SQL> @crea_trabajo3.sql
Procedimiento PL/SQL terminado correctamente.

11. Monitorizar el número de conexiones del usuario "prueba01".

Se crea una entrada en la tabla "usuarios_monitorizados" indicando que se quiere monitorizar al usuario "prueba01".

```
SQL> desc usuarios_monitorizados
 Nombre                                    ?Nulo?   Tipo
 ----------------------------------------------------------------------
 USUARIO                                            VARCHAR2(30)
 MONITORIZAR                                        CHAR(1)
 DESCRIPCION                                        VARCHAR2(80)

SQL> insert into usuarios_monitorizados values ('PRUEBA01','S',null);
1 fila creada.

SQL> commit;
Confirmacion terminada.
```

Se abren distintas sesiones del usuario "prueba01", cuyo número quedará reflejado en "usuarios_monitorizados_log".

```
SQL> select USUARIO, to_char(FECHA_LOG,'dd-mm-yyyy hh:mi:ss') FECHA,
SESIONES, ACTIVAS from usuarios_monitorizados_log order by 2;

USUARIO              FECHA                  SESIONES      ACTIVAS
-------------------------------------------------------------------------------------
PRUEBA01             09-12-2006 10:12:07        1             0
PRUEBA01             09-12-2006 10:12:07        2             0
PRUEBA01             09-12-2006 10:12:07        2             0
PRUEBA01             09-12-2006 11:12:07        2             0
```

12.Consultar todos los trabajos definidos para el usuario SYSTEM obteniendo el nombre de trabajo, tipo de trabajo, fecha de la próxima ejecución, intervalo, numero de fallos producidos y su estado.

```
SQL> select job_name, job_type, start_date,  next_run_date, repeat_interval,
failure_count, enabled from dba_scheduler_jobs  where owner='SYSTEM' order by
job_name;

JOB_NAME                              JOB_TYPE             START_DATE
NEXT_RUN_DATE
```

```
REPEAT_INTERVAL
FAILURE_COUNT ENABL
------------------------------------------------------------------------------------------------
TRABAJO_ANALIZA                 STORED_PROCEDURE 08/12/06 14:41:19,000000
+01:00                          15/12/06 14:41:19,000000 +01:00
FREQ = WEEKLY
         0 TRUE

TRABAJO_MONITORIZAR_USUARIOS    STORED_PROCEDURE    09/12/06
10:36:07,000000 +01:00                          09/12/06 12:56:07,000000 +01:00
FREQ = MINUTELY; INTERVAL=1
         0 TRUE
```

13. Consultar todos los trabajos definidos para el usuario PRUEBA01 obteniendo el nombre de trabajo, tipo de trabajo, fecha de la próxima ejecución, intervalo, numero de fallos producidos y su estado.

```
SQL> select job_name, job_type, start_date,  next_run_date, repeat_interval,
failure_count, enabled from dba_scheduler_jobs  where owner='PRUEBA01' order
by job_name;

JOB_NAME                                JOB_TYPE                START_DATE
NEXT_RUN_DATE
REPEAT_INTERVAL
FAILURE_COUNT ENABL
------------------------------------------------------------------------------------------------
TRABAJO_CHEQUEAR_TABLAS                 STORED_PROCEDURE    08/12/06
17:01:50,000000 +01:00                          09/12/06 13:01:50,000000 +01:00
FREQ = HOURLY; INTERVAL=1
         0 TRUE
```

14. Conectarse como usuario "prueba01" e intentar eliminar de la cola de trabajos el trabajo "TRABAJO_MONITORIZAR_USUARIOS" perteneciente al usuario SYSTEM, ¿qué sucede?.

Se crea un "script" de nombre, por ejemplo, "borra_analiza_usuarios.sql" con el siguiente contenido:

```
/home/CURSO/curso01 (CURSO01)> vi borra_analiza_usuarios.sql
BEGIN
DBMS_SCHEDULER.DROP_JOB('TRABAJO_MONITORIZAR_USUARIOS');
```

```
END;
/
~
"borra_analiza_usuarios.sql" 4L, 70C escritos

SQL> connect prueba01
Introduzca la contrase?a:
Conectado.

SQL> @borra_analiza_usuarios.sql
BEGIN
*
ERROR en linea 1:
ORA-27475: "PRUEBA01.TRABAJO_MONITORIZAR_USUARIOS" debe ser job
ORA-06512: en "SYS.DBMS_ISCHED", linea 178
ORA-06512: en "SYS.DBMS_SCHEDULER", linea 544
ORA-06512: en linea 2
```

No existe ningún trabajo perteneciente al usuario prueba01 llamado de esta forma.

15. Deshabilitar el trabajo que realiza el chequeo de tablas, *"TRABAJO_CHEQUEAR_TABLAS" perteneciente al usuario PRUEBA01*. Consultar las características del trabajo.

```
SQL> show user
USER es "PRUEBA01"

SQL>  begin
  2   DBMS_SCHEDULER.DISABLE('TRABAJO_CHEQUEAR_TABLAS');
  3   END;
  4  /

Procedimiento PL/SQL terminado correctamente.

SQL> select job_name, job_type, enabled from user_scheduler_jobs  where
job_name='TRABAJO_CHEQUEAR_TABLAS'

JOB_NAME                       JOB_TYPE             ENABL
------------------------------------------------------------------------------
TRABAJO_CHEQUEAR_TABLAS        STORED_PROCEDURE  FALSE
```

16. Conectarse como usuario SYSTEM y modificar el trabajo que llama al procedimiento de monitorización de usuarios para que se realice cada dos minutos. Consultar las nuevas características del trabajo.

```
/home/CURSO/curso01 (CURSO01)> vi modificar_atributos.sql
BEGIN
DBMS_SCHEDULER.SET_ATTRIBUTE('TRABAJO_MONITORIZAR_USUARIOS','REPEAT_IN
TERVAL','FREQ = MINUTELY; INTERVAL = 2');
END;
/
~
"modificar_atributos.sql" [Nuevo] 4L, 125C escritos

SQL> show user
USER es "SYSTEM"

SQL> @modificar_atributos.sql
Procedimiento PL/SQL terminado correctamente.


SQL> select job_name, job_type, start_date,  next_run_date, repeat_interval,
failure_count, enabled from dba_scheduler_jobs  where
job_name='TRABAJO_MONITORIZAR_USUARIOS';

JOB_NAME                       JOB_TYPE  START_DATE
NEXT_RUN_DATE
REPEAT_INTERVAL
FAILURE_COUNT ENABL
------------------------------------------------------------------------------------------
TRABAJO_MONITORIZAR_USUARIOS   STORED_PROCEDURE 09/12/06
10:36:07,000000 +01:00                    09/12/06 13:36:07,000000 +01:00
FREQ = MINUTELY; INTERVAL = 2
          0 TRUE
```

17. Conectarse como usuario SYSTEM y eliminar el trabajo que realiza el chequeo de ocupación de tablas (“TRABAJO_CHEQUEAR_TABLAS”).

```
SQL> SHOW USER
USER es "SYSTEM"

SQL> BEGIN
  2  DBMS_SCHEDULER.DROP_JOB('TRABAJO_CHEQUEAR_TABLAS',TRUE);
  3  END;
```

```
  4  /
BEGIN
*
ERROR en linea 1:
ORA-27475: "SYSTEM.TRABAJO_CHEQUEAR_TABLAS" debe ser job
ORA-06512: en "SYS.DBMS_ISCHED", linea 178
ORA-06512: en "SYS.DBMS_SCHEDULER", linea 544
ORA-06512: en linea 2
```

No existe ningún trabajo perteneciente al usuario SYSTEM llamado de esta forma y ni siquiera el administrador puede borrar trabajos de otro usuario.

18. Conectarse como usuario PRUEBA01 y eliminar el trabajo que realiza el chequeo de ocupación de tablas ("TRABAJO_CHEQUEAR_TABLAS").

```
SQL> show user
USER es "PRUEBA01"

SQL>  BEGIN
  2   DBMS_SCHEDULER.DROP_JOB('TRABAJO_CHEQUEAR_TABLAS',TRUE);
  3   END;
  4  /

Procedimiento PL/SQL terminado correctamente.
```

SQL> select job_name, job_type, start_date, next_run_date, repeat_interval, failure_count, enabled from ***user_scheduler_jobs*** *where job_name='TRABAJO_CHEQUEAR_TABLAS';*

```
ninguna fila seleccionada
```