

# Serranito 8

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?.

```
conner@SGBD:~$ ps -ef | grep cj
oracle      3259      1  0 11:24 ?        00:00:00 xe_cjq0_XE
conner      3327  3271  0 11:24 pts/1    00:00:00 grep --color=auto cj
conner@SGBD:~$
```

2. Ver el contenido del paquete dbms\_scheduler.

Contiene lo siguiente;

1. CREATE\_JOB
2. DROP\_JOB
3. STOP\_JOB
4. RUN\_JOB
5. COPY\_JOB
6. ENABLE
7. DISABLE
8. SET\_ATTRIBUTE

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
```

Campos de DBA\_SCHEDULER\_JOBS:

- OWNER: Propietario del trabajo programado.
- JOB\_NAME: Nombre del trabajo programado.
- JOB\_TYPE: Tipo de trabajo programado (por ejemplo, programado, externo, clásico).
- JOB\_ACTION: Acción que se ejecutará cuando se ejecute el trabajo programado (por ejemplo, una secuencia de comandos PL/SQL, un procedimiento almacenado o una llamada a un programa externo).
- NUMBER\_OF\_ARGUMENTS: Número de argumentos que se le pasarán al trabajo programado.
- START\_DATE: Fecha de inicio programada para el trabajo.
- REPEAT\_INTERVAL: Intervalo de repetición para el trabajo programado.
- END\_DATE: Fecha de finalización programada para el trabajo.
- ENABLED: Indica si el trabajo programado está habilitado o deshabilitado.
- STATE: Estado actual del trabajo programado (por ejemplo, EJECUTANDO, DETENIDO).

```
SQL> desc dba_scheduler_job_log;
```

Nombre	?Nulo?	Tipo
LOG_ID	NOT NULL	NUMBER
LOG_DATE		TIMESTAMP(6) WITH TIME ZONE
OWNER		VARCHAR2(128)
JOB_NAME		VARCHAR2(261)
JOB_SUBNAME		VARCHAR2(261)
JOB_CLASS		VARCHAR2(128)
OPERATION		VARCHAR2(30)
STATUS		VARCHAR2(30)
USER_NAME		VARCHAR2(128)
CLIENT_ID		VARCHAR2(64)
GLOBAL_UID		VARCHAR2(32)
CREDENTIAL_OWNER		VARCHAR2(261)
CREDENTIAL_NAME		VARCHAR2(261)
DESTINATION_OWNER		VARCHAR2(261)
DESTINATION		VARCHAR2(261)
ADDITIONAL_INFO		CLOB

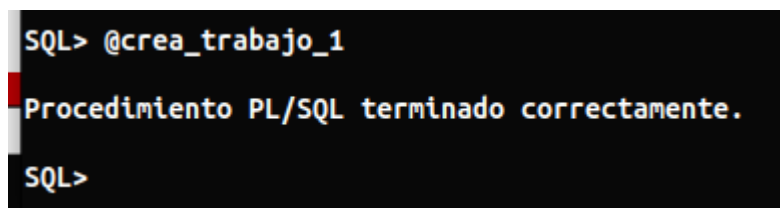
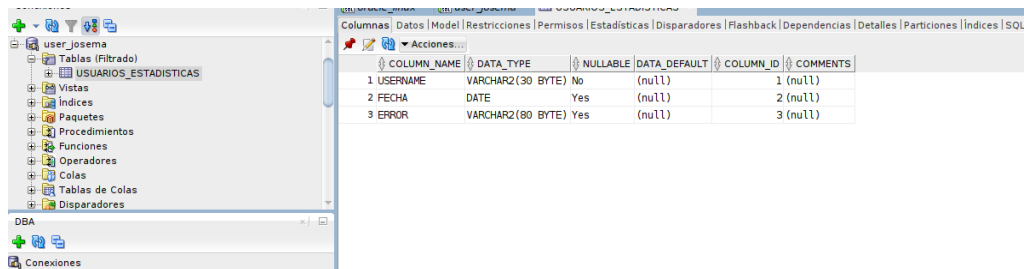
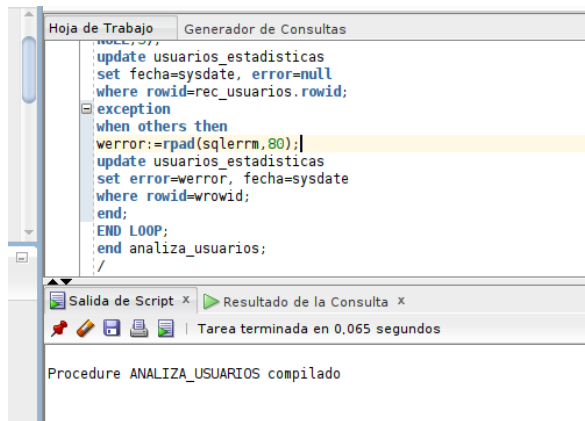
Campos de DBA\_SCHEDULER\_JOB\_LOG:

- LOG\_ID: Identificador único del registro del trabajo programado.
- JOB\_NAME: Nombre del trabajo programado.
- OWNER: Propietario del trabajo programado.
- STATUS: Estado actual del trabajo programado (por ejemplo, EXITOSO, ERROR).
- LOG\_DATE: Fecha y hora de la ejecución del trabajo programado.
- ADDITIONAL\_INFO: Información adicional sobre el trabajo programado.

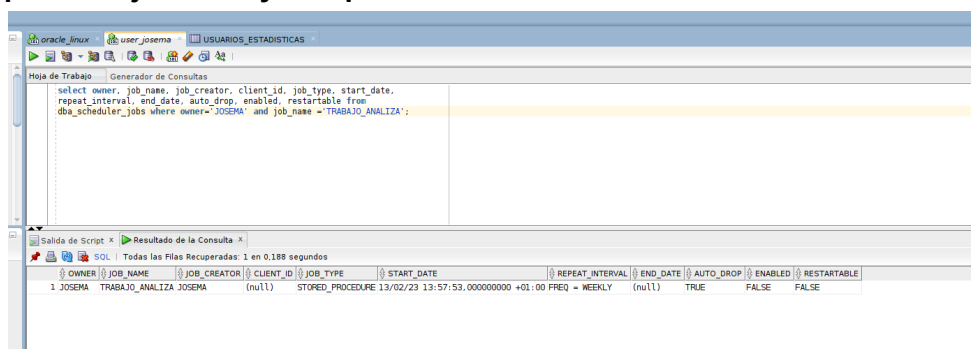
**4. Como usuario SYSTEM crear un trabajo que llamará al procedimiento descrito seguidamente, también propiedad del SYSTEM, que permite analizar el esquema de un cierto numero de usuarios y generar las estadísticas internas usadas por el analizador sintáctico de Oracle (previamente deben crearse la tabla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

Creo la tabla:

```
SQL> @usuario
Tabla creada.
```



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='JOSEMA' and job_name='TRABAJO_ANALIZA'; 2 3
```

OWNER	CLIENT_ID	JOB_NAME	JOB_TYPE	START_DATE	JOB_CREATOR
JOSEMA		TRABAJO_ANALIZA	STORED_PROCEDURE	13/02/23 13:57:53.000000000 +01:00	JOSEMA

```
SQL>
```

```
/bin/bash
/bin/bash 177x31
activa_trabajo1.sql *
```

```
GNU nano 6.2
BEGIN
DBMS_SCHEDULER.ENABLE('TRABAJO_ANALIZA');
END;
/
```

```
SQL> @activa_trabajo1

Procedimiento PL/SQL terminado correctamente.

SQL> |
```

Hoja de Trabajo | Generador de Consultas

```
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='JOSEMA' and job_name='TRABAJO_ANALIZA';
```

Salida de Script x | Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,005 segundos

	OWNER	JOB_NAME	JOB_CREATOR	CLIENT_ID	JOB_TYPE	START_DATE	REPEAT_INTERVAL	END_DATE	AUTO_DROP	ENABLED	RESTARTABLE
1	JOSEMA	TRABAJO_ANALIZA	JOSEMA	(null)	STORED_PROCEDURE	13/02/23 13:57:53.000000000 +01:00	FREQ = WEEKLY	(null)	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?.

El valor es 0

```
SQL> select job_name, max_runs, max_failures, failure_count from
dba_scheduler_jobs where owner='JOSEMA' and job_name='TRABAJO_ANALIZA'; 2
```

JOB_NAME	MAX_RUNS	MAX_FAILURES	FAILURE_COUNT
TRABAJO_ANALIZA			0

```
SQL> |
```

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).

```
GNU nano 6.2 fuerza_trabajo1.sql *
BEGIN
DBMS_SCHEDULER.RUN_JOB('TRABAJO_ANALIZA',FALSE);
END;
/
```

```
SQL> @fuerza_trabajo1

Procedimiento PL/SQL terminado correctamente.

SQL> |
```

```
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='JOSEMA'
and job_name ='TRABAJO_ANALIZA'; 2 3
```

RUN_COUNT	FECHA	DURACION
1	13-02-2023 02:22	+0000000000 00:00:00.142128

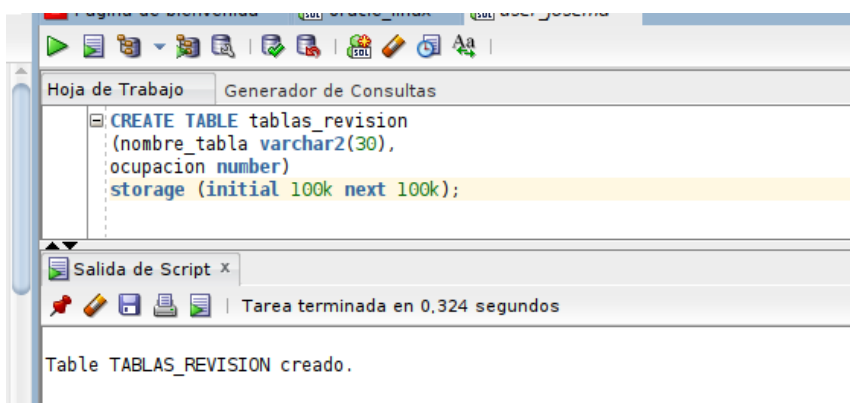
```
SQL> |
```

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.

```
SQL> CREATE USER prueba01 identified by oracle;

Usuario creado.

SQL>
```



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

The screenshot shows the SQL Developer interface. The top pane displays a PL/SQL procedure named `chequear_tablas`. The procedure iterates through tables in `user_tables`, calculates their extension percentage, and inserts data into `tablas_revision` if the percentage exceeds 80%. The bottom pane shows the execution output, confirming that the `TABLAS_REVISION` table was created and the procedure was compiled successfully.

```

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;
/

```

Salida de Script x

Tarea terminada en 0,414 segundos

Table TABLAS\_REVISION creado.

Procedure CHEQUEAR\_TABLAS compilado

The terminal window shows a SQL\*Plus session. It displays the release information (21.0.0.0.0 - Production on Mar Feb 14 00:50:21 2023) and the connection to an Oracle Database 21c Express Edition. The user creates a table named `tablas_revision` with two columns: `nombre_tabla` (varchar2(30)) and `ocupacion` (number). The storage parameters are set to `initial 100k next 100k`. The session confirms the table was created successfully.

```

# sqlplus prueba01/oracle@192.168.0.35

SQL*Plus: Release 21.0.0.0.0 - Production on Mar Feb 14 00:50:21 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

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

Tabla creada.

SQL>

```

The screenshot shows a nano text editor window with the file `chequear_tablas.sql`. It contains the same PL/SQL procedure code as shown in the SQL Developer screenshot.

```

GNU nano 6.2 chequear_tablas.sql *
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> @chequear_tablas
Procedimiento creado.
SQL> |
```

```
GNU nano 6.2 crea_trabajo2.sql *
/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;
/
```

```
SQL> @crea_trabajo2
SP2-0734: inicio "/home/CURS..." de comando desconocido - resto de la
línea ignorado.
BEGIN
*
ERROR en línea 1:
ORA-27477: "PRUEBA01"."TRABAJO_CHEQUEAR_TABLAS" ya existe
ORA-06512: en "SYS.DBMS_ISCHED", línea 174
ORA-06512: en "SYS.DBMS_SCHEDULER", línea 286
ORA-06512: en línea 2
SQL> |
```

- ¿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”.

Porque el usuario prueba01 no tiene permiso para crear un job, por lo que tendremos que asignarle los permisos.

```
# josema on ~
# sqlplus sys/or4cl3@192.168.0.35 as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Mar Feb 14 00:55:30 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> grant create job to prueba01;
Concesión terminada correctamente.

SQL> |
```

```
SQL> @crea_trabajo2
SP2-0734: inicio "/home/CURS..." de comando desconocido - resto de la
línea ignorado.

Procedimiento PL/SQL terminado correctamente.

SQL> exit
```

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

Resultado:

```
-----
JOB_TYPE
-----
START_DATE
-----
REPEAT_INTERVAL
-----
--
END_DATE
-----
AUTO_ ENABL RESTA
-----
18/08/21 03:00:00,760629 ETC/UTC
-----
JOB_NAME
-----
```

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?.

```
/bin/bash 69x27
GNU nano 6.2 fuerza_trabajo2.sql *
BEGIN
DBMS_SCHEDULER.RUN_JOB('TRABAJO_CHEQUEAR_TABLAS',FALSE);
END;
/
```

```
# sqlplus prueba01/oracle@192.168.0.35

SQL*Plus: Release 21.0.0.0.0 - Production on Mar Feb 14 01:03:13 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Hora de Última Conexión Correcta: Mar Feb 14 2023 00:56:12 +01:00

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

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

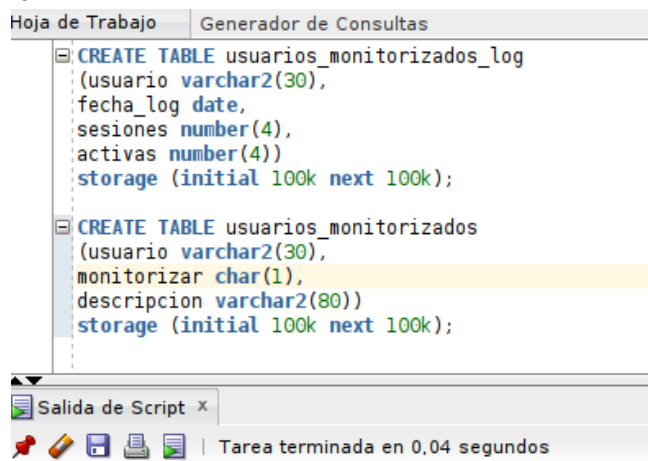
```
SQL> @fuerza_trabajo2

Procedimiento PL/SQL terminado correctamente.

SQL> |
```



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.



Hoja de Trabajo | Generador de Consultas

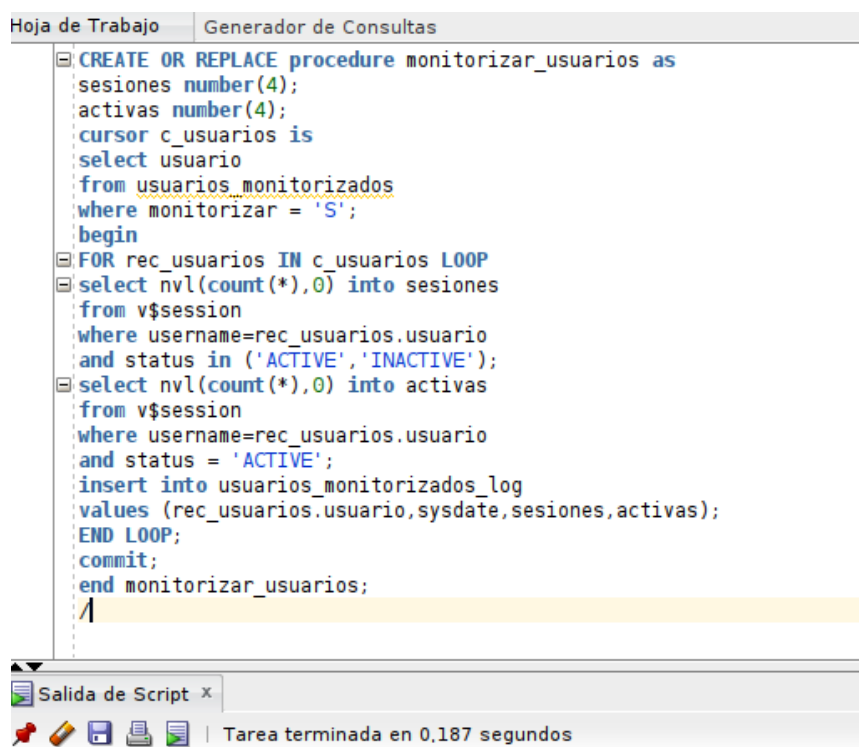
```
CREATE TABLE usuarios_monitorizados_log
(usuario varchar2(30),
fecha_log date,
sesiones number(4),
activas number(4))
storage (initial 100k next 100k);

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

Salida de Script x

Tarea terminada en 0,04 segundos

Table USUARIOS\_MONITORIZADOS creado.



Hoja de Trabajo | Generador de Consultas

```
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;
/
```

Salida de Script x

Tarea terminada en 0,187 segundos

Procedure MONITORIZAR\_USUARIOS compilado

```
# sqlplus sys/or4cl3@192.168.0.35 as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Mar Feb 14 01:09:50 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> grant select on V_$SESSION to system;

Concesión terminada correctamente.

SQL> |
```

```
/bin/bash 69x13
GNU nano 6.2 monitorizar_usuarios.sql *
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
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecut
```

```
SQL> @monitorizar_usuarios

Procedimiento creado.

SQL> |
```

```
/bin/bash 69x13
GNU nano 6.2 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,
```

```
SQL> @crea_trabajo3

Procedimiento PL/SQL terminado correctamente.

SQL> |
```

## 11. Monitorizar el número de conexiones del usuario “prueba01”.

```
SQL> desc usuarios_monitorizados
```

Nombre	¿Nulo?	Tipo
USUARIO		VARCHAR2(30)
MONITORIZAR		CHAR(1)
DESCRIPCION		VARCHAR2(80)

```
SQL>
```

```
SQL> insert into usuarios_monitorizados values ('PRUEBA01','S',null);
```

```
1 fila creada.
```

```
SQL> commit;
```

```
Confirmación terminada.
```

```
SQL> |
```

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

USUARIO	FECHA	SESIONES	ACTIVAS
PRUEBA01	14-02-2023 01:13:20	0	0

```
SQL> |
```

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

```
0 TRUE  
TRABAJO MONITORIZAR USUARIOS                                STORED_PROCEDURE 14/02/23 01:12:20,000000 +01:00 14/0  
2/23 01:16:20,547720 +01:00  
FREQ = MINUTELY; INTERVAL=1  
0 TRUE  
XMILOB NFS CLEANUP JOB                                STORED_PROCEDURE
```

## 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; 2 3
```

JOB_NAME	NEXT_RUN_DATE	JOB_TYPE	START_DATE
REPEAT_INTERVAL			
FAILURE_COUNT ENABL			
TRABAJO_CHEQUEAR_TABLAS	14/02/23 01:53:05,575924 +01:00	STORED_PROCEDURE	14/02/23 00:53:05,00
0000 +01:00			
FREQ = HOURLY; INTERVAL=1			
0 TRUE			

```
SQL> |
```

14. Conectarse como usuario “prueba01” e intentar eliminar de la cola de trabajos el trabajo “TRABAJO\_MONITORIZAR\_USUARIOS” perteneciente al usuario SYSTEM, ¿qué sucede?.

```
GNU nano 6.2 /bin/bash 64x13
borra_analiza_usuarios.sql *
BEGIN
DBMS_SCHEDULER.DROP_JOB('TRABAJO_MONITORIZAR_USUARIOS');
END;
/
```

```
SQL> @borra_analiza_usuarios
BEGIN
*
ERROR en línea 1:
ORA-27475: job desconocido "PRUEBA01"."TRABAJO_MONITORIZAR_USUARIOS"
ORA-06512: en "SYS.DBMS_ISCHED", línea 277
ORA-06512: en "SYS.DBMS_SCHEDULER", línea 751
ORA-06512: en línea 2
SQL>
```

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> |
```

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.

```
GNU nano 6.2 /bin/bash 128x15
modificar_atributos.sql *
BEGIN
DBMS_SCHEDULER.SET_ATTRIBUTE('TRABAJO_MONITORIZAR_USUARIOS', 'REPEAT_INTERVAL', 'FREQ = MINUTELY; INTERVAL = 2');
END;
/
```

```
SQL> show user
USER es "SYS"
SQL> |
```

```
SQL> @modificar_atributos

Procedimiento PL/SQL terminado correctamente.
SQL> |
```

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

```
JOB_NAME
-----
JOB_TYPE
-----
START_DATE
-----
NEXT_RUN_DATE
-----
REPEAT_INTERVAL
-----
FAILURE_COUNT ENABL
-----
14/02/23 01:28:20,000000 +01:00
-----
JOB_NAME
-----
JOB_TYPE
-----
START_DATE
-----
NEXT_RUN_DATE
-----
REPEAT_INTERVAL
-----
FAILURE_COUNT ENABL
-----
FREQ = MINUTELY; INTERVAL = 2
```

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 "SYS"
SQL> BEGIN
2 DBMS_SCHEDULER.DROP_JOB('TRABAJO_CHEQUEAR_TABLAS',TRUE);
3 END;
4 /
BEGIN
*
ERROR en línea 1:
ORA-27475: job desconocido "SYS"."TRABAJO_CHEQUEAR_TABLAS"
ORA-06512: en "SYS.DBMS_ISCHED", línea 277
ORA-06512: en "SYS.DBMS_SCHEDULER", línea 751
ORA-06512: en línea 2

SQL> |
```

18. Conectarse como usuario PRUEBA01 y eliminar el trabajo que realiza el chequeo de ocupación de tablas ("TRABAJO\_CHEQUEAR\_TABLAS").

```
# sqlplus prueba01/oracle@192.168.0.35

SQL*Plus: Release 21.0.0.0.0 - Production on Mar Feb 14 01:28:46 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2022, Oracle. All rights reserved.

Hora de Última Conexión Correcta: Mar Feb 14 2023 01:17:58 +01:00

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

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> |
```

```
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'; 2 3  
  
ninguna fila seleccionada  
  
SQL> |
```