Ejercicio 2 - tema 6
Estructuras lógicas de almacenamiento - Segmentos

Vicente Romero Andrade

I. Objetivo

EL objetivo es comprender el mecanismo de creación de los distintos tipos de segmentos a partir de la creación de una tabla. Explorar las vistas del diccionario de datos para verificar la creación de segmentos asociados a los objetos de un usuario.

II. Desarrollo

A. C1. s-01-config-compartido.sql y respuesta del inciso C

```
whenever sqlerror exit rollback
set serveroutput on
connect sys/system2 as sysdba
-- A
alter system set dispatchers='(dispatchers=2)(
  PROTOCOL=tcp)' scope=memory;
alter system set shared_servers=4 scope=memory;
show parameter;
--B
alter system register;
--C
select program,pid,pname
  from v$process
    where pname like'S0%' or pname like 'D0%'
    order by program;

whenever sqlerror continue
```

Código 1. s-01-config-compartido.sql

captura_1.png

Figure 1. respuesta inciso C

B. C2. s-02-conexiones.sql y tnsnames.ora

```
whenever sqlerror exit rollback
set serveroutput on
connect sys@vrabda2_dedicated/system2 as sysdba

declare
  v_count number;
  v_username varchar2(30) := 'VRA0501';
  v_table varchar2(30) := 'T01_SESSION_DATA';
begin
  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username
  ||'.'||v_table;
  end if;
  execute immediate 'create table '|| v_username
  ||'.'||v_table||'(
    id number,
    sid number,
    logon_time date,
    username varchar2(20),
    status varchar2(8),
    server varchar2(20),
    osuser varchar2(30),
    process varchar2(12),
    port number
)';
end;
/
-- A
insert into vra0501.t01_session_data(
id,sid,logon_time,username,status,server,
osuser,process,port)
select 1,sid,logon_time,username,status,server,
    osuser,process,port from v$session where
  username = 'SYS';
commit;
--B
connect sys@vrabda2_shared/system2 as sysdba

insert into vra0501.t01_session_data(
id,sid,logon_time,username,status,server,
osuser, process,port)
select 2,sid,logon_time,username,status,server,
    osuser, process,port from v$session where
  username = 'SYS';
commit;

--C
connect VRA0501@vrabda2_dedicated/VRA0501

insert into vra0501.t01_session_data(
id,sid,logon_time,username,status,server,
osuser, process,port)
select 3,sid,logon_time,username,status,server,
    osuser, process,port from v$session where
  username = 'VRA0501';
commit;

--D
connect VRA0501@vrabda2_shared/VRA0501

insert into vra0501.t01_session_data(
id,sid,logon_time,username,status,server,
osuser, process,port)
select 4,sid,logon_time,username,status,server,
    osuser, process,port from v$session where
  username = 'VRA0501';
commit;

whenever sqlerror continue
```

Código 2. s-02-conexiones.sql

```
# tnsnames.ora Network Configuration File: /u01/
  app/oracle/product/19.0.0/dbhome_1/network/admin
  /tnsnames.ora
# Generated by Oracle configuration tools.

VRABDA1 =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = pc-vra.fi.
  unam)(PORT = 1521))
```

```
      (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = vrabda1.fi.unam)
      )
    )

  LISTENER_VRABDA1 =
    (ADDRESS = (PROTOCOL = TCP)(HOST = pc-vra.fi.unam)(PORT = 1521))

  VRABDA2 =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = pc-vra.fi.unam)(PORT = 1521))
      (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = vrabda2)
      )
    )

  VRABDA2_DEDICATED =
          (DESCRIPTION =
                  (ADDRESS_LIST =
                          (ADDRESS = (PROTOCOL = TCP)(HOST = pc-vra.fi.unam)(PORT = 1521))
                  )
          (CONNECT_DATA =
                  (SERVICE_NAME = vrabda2)
                  (SERVER=DEDICATED)
          )
  )

  VRABDA2_SHARED =
          (DESCRIPTION =
                  (ADDRESS_LIST =
                          (ADDRESS = (PROTOCOL = TCP)(HOST = pc-vra.fi.unam)(PORT = 1521))
                  )
          (CONNECT_DATA =
                  (SERVICE_NAME = vrabda2)
                  (SERVER=SHARED)
          )
  )
```

Código 3. tnsname.ora

C. C3. s-03-consultas.sql

```
whenever sqlerror exit rollback
set serveroutput on
connect sys/system2 as sysdba

declare
  v_count number;
  v_username varchar2(30) := 'VRA0501';
  v_table1 varchar2(30) := 'T02_DISPATCHER_CONFIG';
  v_table2 varchar2(30) := 'T03_DISPATCHER';
  v_table3 varchar2(30) := 'T04_SHARED_SERVER';
  v_table4 varchar2(30) := 'T05_QUEUE';
  v_table5 varchar2(30) := 'T06_VIRTUAL_CIRCUIT';
begin
  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table1
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table1;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table2
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table2;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table3
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table3;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table4
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table4;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table5
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table5;
  end if;
end;
/
create table vra0501.t02_dispatcher_config as(
  select 1 as id,dispatchers,connections,sessions,
  service from v$dispatcher_config
);

create table vra0501.t03_dispatcher as(
  select 1 as id,name,network,status,messages,
  trunc(bytes/(1024*1024),2) messages_mb,
  (select count(*) from v$circuit) circuits_created,
  trunc(idle/(60*60),2) idle_min
  from v$dispatcher
);


create table vra0501.t04_shared_server as(
  select 1 as id,name,status,messages,
  trunc(bytes/(1024*1024),2) messages_mb,
  requests,trunc(idle/(60*60),2) idle_min,
  trunc(busy/(60*60),2) busy_min
  from v$shared_server
);

create table vra0501.t05_queue as(
  select 1 as id,queued,wait,totalq from v$queue
);

create table vra0501.t06_virtual_circuit as(
  select 1 as id,c.circuit,dp.name,c.server,c.status
    ,c.queue
  from v$dispatcher dp join v$circuit c on(
    dp.paddr=c.dispatcher)
);

whenever sqlerror continue
```

Código 4. s-03-consultas.sql

captura_2.png

Figure 2. T02_DISPATCHER_CONFIG

captura_3.png

Figure 3. T03_DISPATCHER

captura_4.png

Figure 4. T04_SHARED_SERVER

captura_5.png

Figure 5. T05_QUEUE

captura_6.png

Figure 6. T06_VIRTUAL_CIRCUIT

D. C4. Respuesta inciso A y s-04-procesos.sql

```
whenever sqlerror exit rollback
set serveroutput on
connect sys/system2 as sysdba

declare
  v_count number;
  v_username varchar2(30) := 'VRA0501';
  v_table1 varchar2(30) := 'T07_SESSION_INFO_CONTEXT';
  v_table2 varchar2(30) := 'T08_SESSION_INFO_VIEW';
  v_table3 varchar2(30) := 'T09_PROCESS_INFO';
  v_table4 varchar2(30) := 'T10_BACKGROUND_PROCESS';
  v_table5 varchar2(30) := 'T11_FOREGROUND_PROCESS';
begin
  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table1
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table1;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table2
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table2;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table3
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table3;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table4
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table4;
  end if;

  --Verificar si la table existe
  select count(*) into v_count
  from all_tables
  where table_name = v_table5
  and owner = v_username;
  --Si existe la tabla, entonces se borra
  if v_count > 0 then
    execute immediate 'drop table '|| v_username ||'.'||v_table5;
  end if;
end;
/
create table vra0501.t07_session_info_context(
    host varchar2(20),
    os_user varchar2(30),
    user_id number,
    session_id number
);

insert into vra0501.t07_session_info_context(host,
    os_user, user_id,session_id)
    select sys_context('USERENV','HOST') as host,
    sys_context('USERENV','OS_USER') as os_user,
    sys_context('USERENV', 'SESSION_USERID') as
    user_id,
    sys_context('USERENV','SID') as session_id
    from dual;

create table vra0501.t08_session_info_view as(
  select s.sid as session_id,s.paddr as
    process_address,
  s.username bd_username,s.status as session_status,
  s.port as client_port,s.process as
    os_client_process_id,
  s.program as client_program from v$process p
  join gv$session s on p.addr=s.paddr
  join (select sys_context('USERENV','SID') as
    session_id
  from dual) c on c.session_id=s.sid where s.
    username = 'SYS'
);

create table vra0501.t09_process_info as(
    select sosid,pname,background, tracefile from
    v$process
  join gv$session s on addr=s.paddr join (
    select sys_context('USERENV','SID') as
    session_id
    from dual) c on s.sid = c.session_id
);

create table vra0501.t10_background_process as(
    select addr, sosid, pname,username as
    os_username, background
    from v$process where background='1'
);

create table vra0501.t11_foreground_process as(
    select p.addr, p.sosid, p.pname, p.username as
    bd_username, s.osuser as os_username, p.
    background
    from v$process p left outer join v$session s
```

```
    on(p.addr=s.paddr) where p.background is null
);

whenever sqlerror continue
```

Código 5. s-04-procesos.sql

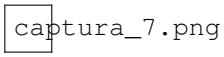

Figure 7. T07_SESSION_INFO_CONTEXT

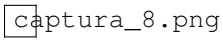

Figure 8. T08_SESSION_INFO_VIEW

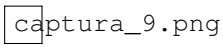

Figure 9. T09_PROCESS_INFO

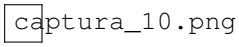

Figure 10. T10_BACKGROUND_PROCESS

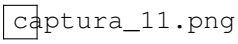

Figure 11. T11_FOREGROUND_PROCESS

```
ps -ef | grep oracle
```

Código 6. instrucciones SO

III. Conclusiones