



1. IMPLEMENTACIÓN DEL ESQUEMA DE FRAGMENTACIÓN - ADMINISTRACIÓN DE PROYECTOS DE UNA EMPRESA DE TI

1. IMPLEMENTACIÓN DEL ESQUEMA DE FRAGMENTACIÓN - ADMINISTRACIÓN DE PROYECTOS DE UNA EMPRESA DE TI.....	1
1.1. Objetivo.....	2
1.2. Conceptos básicos para trabajar con SQL*Plus, lineamientos de programación.....	2
1.3. Creación de usuario.....	2
1.3.1. Ejemplo, script de creación de usuarios.....	2
1.3.2. Ejemplo de ejecución.....	3
1.4. Implementación de fragmentos - Código DDL.....	4
1.4.1. Creación de los objetos en cada PDB.....	4
1.5. Exploración del diccionario de datos.....	4
1.5.1. Ejemplo, script de creación de usuarios.....	5
Ejemplo.....	5
1.5.2. Referencias de integridad.....	5
Ejemplo.....	6
1.6. Carga inicial de datos.....	7
Ejemplo.....	8
1.6.1. Conteo de registros.....	9
1.6.1.1. Ejemplo: Conteo de registros.....	9
Ejemplo.....	9
1.7. Validación de resultados.....	10
1.8. Contenido de la entrega.....	10

1.1. Objetivo

Implementar el esquema de fragmentación realizado en el ejercicio práctico anterior empleando Pluggable Databases (PDBs) .

1.2. Conceptos básicos para trabajar con SQL*Plus, lineamientos de programación

Revisar los conceptos y los lineamientos para trabajar con los ambientes de bases de datos explicados en el documento [comun/configuracion-inicial-ambiente.pdf](#) , asegurarse de haber realizado las siguientes configuraciones:

- Historial de comandos con `rlwrap`
- Personalización del prompt
- Familiarizarse con el concepto y uso del spool.



1.3. Creación de usuario

- Crear un script llamado `s-01-<iniciales>-creacion-usuarios.sql`
- `<iniciales>` corresponde al valor de las iniciales del estudiante (las mismas empleadas en el ejercicio práctico anterior).
- El script deberá conectarse a cada PDB y crear un usuario llamado `consultora_bdd`
- El usuario deberá contar con las siguientes configuraciones:
 - Cuota ilimitada en el tablespace `users`.
 - Privilegios para crear sesión, tablas, procedimientos y secuencias (no asignar más privilegios de los necesarios).
 - Password: se recomienda el mismo que el nombre de usuario por simplicidad y propósitos del curso..

Para mayores detalles respecto a conceptos y ejemplos para realizar la creación de usuarios, revisar el documento [BD/practicas/practica02/lenguaje-control-acceso.pdf](#)

1.3.1. Ejemplo, script de creación de usuarios

```
--@Autor:          Jorge A. Rodríguez C
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Creación de usuarios para la máquina pc-jrc.

prompt Conectándose a <iniciales>bdd_s1 como usuario SYS
connect sys@<iniciales>bdd_s1 as sysdba
prompt creando usuario consultora_bdd
--completar
prompt conectándose a <iniciales>bdd_s2 como usuario SYS
```

```
--completar
prompt creando usuario consultora_bdd
--completar

prompt Listo
exit
```

1.3.2. Ejemplo de ejecución

```
[jorge@jrc-ora-pc scripts]$ sqlplus /nolog

idle> @s-01-<iniciales>-creacion-usuarios.sql

Conectandose a <iniciales>bdd_s1 como usuario SYS
Enter password:
Connected.
creando usuario consultora_bdd
User created.
otorgando permisos mínimos necesarios
Grant succeeded.

conectandose a <iniciales>bdd_s2 como usuario SYS
Enter password:
Connected.
User created.
otorgando permisos mínimos necesarios
Grant succeeded.
```

- En la línea 1, observar que no se emplea el usuario **oracle** para ejecutar el script. Se recomienda ejecutarlo con el usuario ordinario del sistema operativo y cambiarse al directorio donde se encuentran los scripts del ejercicio práctico. No copiar los archivos dentro de los directorios cuyo dueño es el usuario **oracle**. Por ejemplo, en alguna carpeta a partir de **/u01** o **/home/oracle**.
- En la línea 1, observar que se emplea **/nolog** para entrar a SQL*Plus sin autenticar. Esto debido a que el script se encarga de conectarse a la PDB correspondiente.
- Finalmente, empleando el comando **start** se ejecuta el script. En Oracle se puede emplear también **'@'** para invocar la ejecución de un script.
- Para comprobar que la creación de usuarios fue correcta, se puede ejecutar la siguiente instrucción:

```
sqlplus consultora_bdd/consultora_bdd@<iniciales>bdd_s1
```

- Forma segura ocultando el password

```
sqlplus consultora_bdd@<iniciales>bdd_s1
```

- Al entrar en sesión se muestra el prompt personalizado con el usuario y la PDB actual, por ejemplo:

```
consultora_bdd@jrcbdd_s1>
```

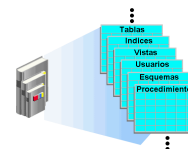


1.4. Implementación de fragmentos - Código DDL

- Crear un script SQL con la siguiente nomenclatura:
`s-02-<iniciales>-<nombre-pdb>-ddl.sql`
- El script contendrá la definición de los fragmentos que serán creados en cada PDB, es decir se tendrán 2 archivos, uno por PDB
- Asignar un nombre adecuado y entendible a cada restricción de referencia. El contar con un nombre claro permitirá identificar fácilmente posibles errores de integridad referencial. Por convención el nombre de la restricción de referencia es:
`<nombre_tabla_hija>_<nombre_campo_fk>`
- Para los nombres de las restricciones de llave primaria emplear la convención
`<nombre_tabla_pk>`
- Para las FKs emplear la notación `<nombre-tabla-padre>_<nombre_atributo>_fk`

1.4.1. Creación de los objetos en cada PDB.

- Crear un archivo llamado `s-03-<iniciales>-main-ddl.sql`
- El archivo deberá conectarse a cada PDB y ejecutar los scripts de creación de objetos.



1.5. Exploración del diccionario de datos

Ejecutar las siguientes sentencias que muestran las características de los objetos creados en cada PDB. Las sentencias se deben ejecutar en ambas PDBs.

1.5.1. Ejemplo, script de creación de usuarios.

- Generar un archivo llamado **s-04-<iniciales>-consulta-fragmentos.sql**
- El script deberá conectarse a cada PDB y deberá mostrar la lista de fragmentos (tablas) que fueron creadas. **C1. Incluir en el reporte** el contenido del script y el resultado obtenido. Ordenar la lista por el nombre del fragmento. Tip: emplear la vista del diccionario de datos **user_tables**.

Ejemplo

```
--@Autor:          Jorge Rodriguez
--@Fecha creación: dd/mm/yyyy
--@Descripción:     Consulta de fragmentos creados en jrc-pc
Prompt Conectando a S1 - <iniciales>bdd_s1
connect consultora_bdd/consultora_bdd@<iniciales>bdd_s1
Prompt mostrando lista de fragmentos
col table_name format a30
select table_name from user_tables

Prompt Conectando a S2 - <iniciales>bdd_s2
connect consultora_bdd/consultora_bdd@<iniciales>bdd_s2
Prompt mostrando lista de fragmentos

--completar

Prompt Listo!
exit
```

1.5.2. Referencias de integridad

- Generar un archivo llamado **s-05-<iniciales>-consulta-restricciones.sql**
- El script deberá conectarse a cada PDB y deberá mostrar las relaciones de referencia que existen entre los fragmentos de cada PDB. Los campos a mostrar son: Nombre de la tabla hija, nombre de la restricción de referencia, nombre de la tabla padre y tipo de restricción.
- Ordenar la lista por el nombre de la tabla padre.
- Tips:
 - En Oracle la vista **user_constraints** contiene los datos referentes a las restricciones de una tabla.
 - El atributo **constraint_type** define el tipo de restricción, que en este caso sólo interesan las restricciones de referencia: 'R'
 - El campo **r_constraint_name** contiene el nombre de la restricción de la llave primaria de la tabla padre. Este campo puede ser empleado como condición de join para obtener el nombre de la tabla padre.



Ejemplo

```
--@Autor:           Jorge Rodriguez
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Consulta de restricciones de referencia

Prompt mostrando lista de restricciones de referencia
col tabla_padre format A30
col tabla_hija format A30
col nombre_restriccion format A30
set linesize 200

select th.table_name tabla_hija, th.constraint_name nombre_restriccion,
       tp.table_name tabla_padre, th.constraint_type tipo_constraint
from user_constraints th, user_constraints tp
where th.constraint_type='R'
and th.r_constraint_name=tp.constraint_name
order by tabla_padre;
```

- Observar el comando **col** empleado para dar formato a las columnas y poder visualizar los resultados correctamente.
- Notar que el código del script es exactamente el mismo a ejecutar en los 2 nodos.
- Para evitar ejecutar manualmente el script en ambos nodos o para evitar duplicar el código en cada nodo, se recomienda crear un script adicional **s-05-<iniciales>-consulta-restricciones-main.sql** que se conecte a cada sitio e invoque al script anterior:

```
--@Autor:           Jorge Rodriguez
--@Fecha creación:  dd/mm/yyyy
--@Descripción:     Consulta de restricciones de referencia en jrc-pc

Prompt Conectando a S1 - <iniciales>bdd_s1
connect consultora_bdd/consultora_bdd@<iniciales>bdd_s1
--ejecuta la misma consulta en ambas pdbs
@s-05-<iniciales>-consulta-restricciones.sql

Prompt Conectando a S2 - <iniciales>bdd_s2
connect consultora_bdd/consultora_bdd@<iniciales>bdd_s1
--ejecuta la misma consulta en ambas pdbs
@s-05-<iniciales>-consulta-restricciones.sql

Prompt Listo!
exit
```

C2. Incluir en el reporte la salida de la consulta.

1.6. Carga inicial de datos.

Debido a que hasta este punto no se cuenta con ningún nivel de transparencia implementado, la carga y consulta de datos se debe realizar de forma manual en cada PDB. La implementación de los distintos tipos y niveles de transparencia se implementarán en ejercicios prácticos posteriores.

- Crear un script `s-06-<iniciales>-carga.sql` El script deberá contener las sentencias insert necesarias a ejecutar en cada PDB para cargar la siguiente información.
- El script deberá conectarse a cada PDB y realizar las inserciones de forma manual considerando el esquema de fragmentación.

PAIS

PAIS_ID	CLAVE	NOMBRE	REGION
1	MX	MEXICO	A
2	JAP	JAPON	B

OFICINA

OFICINA_ID	NOMBRE	PAIS_ID	OFCINA_ANEXA_ID
1	OFICINA 1 DE CDMX	1	
2	OFICINA 1 DE TOKIO	2	1

EMPLEADO

EMPLEADO_ID	NOMBRE	AP_PAT	AP_MAT	RFC	FOTO	NUM_CUENTA	EMAIL	JEFE_ID
1	JUAN	LOPEZ	LARA	LOLA890802KML	Empty_blob()	09934902	juanq@m.com	
2	CARLOS	BAEZ	AGUIRRE	BAAGCA982613	Empty_blob()	04449321	carlos@m.com	1
3	EVA	ZAVALA	CORTES	ZACOEVO40423	Empty_blob()	67382342	eva@m.com	1

PROYECTO

PROYECTO_ID	NOMBRE	CLAVE	FECHA_INICIO	FECHA_FIN	RESPONSABLE_ID	OFICINA_ID
1	PROYECTO A	P00A	16/02/2009	31/12/2012	1	1
2	PROYECTO B	P00B	14/01/2008	31/12/2013	2	2
3	PROYECTO C	P00C	06/05/2008		3	2

PROYECTO_PDF

NUM_PDF	PROYECTO_ID	ARCHIVO_PDF	TAMANIO
1	1	Empty_blob()	2
1	2	Empty_blob()	4

PAGO_EMPLEADO

PAGO_EMPLEADO_ID	IMPORTE	FECHA_PAGO	PROYECTO_ID	EMPLEADO_ID
1	1500.45	01/02/2017	1	1
2	5490.45	01/08/2017	3	2
3	760.67	31/12/2016	2	3

Ejemplo

El siguiente pseudocódigo muestra la estructura recomendada de este script.

```
--@Autor:          Jorge Rodriguez
--@Fecha creación: dd/mm/yyyy
--@Descripción:     Archivo de carga inicial en jrc-pc

Prompt Conectando a S1 - <iniciales>dd_s1
connect consultora_bdd/consultora_bdd@<iniciales>bdd_s1

--si ocurre un error, La ejecución se detiene.
whenever sqlerror exit rollback;

Prompt limpiando.
--delete from ...

Prompt Cargando datos
--insert into ..

--hacer commit al terminar
commit;

Prompt Conectando a S2 - <iniciales>bdd_s2
connect consultora_bdd/consultora_bdd@<iniciales>bdd_s2

--si ocurre un error, La ejecución se detiene.
whenever sqlerror exit rollback;

Prompt limpiando.
--delete from ...

Prompt Cargando datos
--insert into ..
--hacer commit al terminar
commit;
Prompt Listo!
exit
```

- Observar que se agregan sentencias **delete**. Esto permite ejecutar el script N veces sin provocar errores de duplicidad ya que primero se realiza una limpieza de las tablas y posteriormente se realiza la carga inicial. Esto funciona correctamente ya que las tablas se encuentran vacías.

1.6.1. Conteo de registros

- Generar una sentencia SQL que muestre el conteo de todos los registros en cada PDB. Generar un script llamado **s-07-<iniciales>-consultas.sql**
- Esta consulta permitirá validar de forma manual el cumplimiento de las 3 reglas de fragmentación.
- El script deberá conectarse a ambas PDBs y generar una consulta similar a la siguiente.

1.6.1.1. Ejemplo: Conteo de registros

```
idle> start s-07-jrc-consulta-datos.sql
conectando a sitio s1
Connected.
Realizando conteo de registros
```

PAIS_1	OFICINA_1	EMPLEADO_1	PROYECTO_1	PROYECTO_PDF_1	PAGO_EMPLEADO_2
x	x	x	x	x	x

```
conectando a sitio s2
Connected.
Realizando conteo de registros
```

```
. . .
. . .
. . .
```

- Las consultas deben emplear los mismos alias y orden de consultas mostrados.
- Se deberá obtener un solo registro por cada PDB.
- Tip: emplear subqueries en la cláusula **select**.
- **C3. Incluir en el reporte** el código y el resultado de ejecución.



Ejemplo

```
select
  (select count(*) from f_jrc_pais_1 ) as pais_1,
  (select count(*) from f_jrc_banco_1) as banco_1,
  . . .
from dual;
```

1.7. Validación de resultados.

- De la carpeta del ejercicio práctico, obtener todos los scripts.
 - Ejecutar el archivo `s-08-validacion-main.sql` El script solicitará ciertos datos para poder realizar la validación. En caso de existir errores, revisar los mensajes y corregir.
- C4. Incluir en el reporte** el resultado.

1.8. Contenido de la entrega

Elementos comunes a todos los ejercicios especificados en la rúbrica general en el documento `presentacion-curso.pdf`

Contenidos de este documento:

- C1. Código y salida del script `s-04-<iniciales>-consulta-fragmentos.sql`
- C2. Salida del script `s-05-<iniciales>-consulta-restricciones-main.sql` y `s-05-<iniciales>-consulta-restricciones.sql`
- C3. Código y salida del script `s-07-<iniciales>-consultas.sql`
- C4. Salida de ejecución del script de validación `s-08-validacion-main.sql`