INFORME EJECUTIVO

CONTRATO DE PRESTACION DE SERVICIOS FGN-NC-0107-2020 CELEBRADO ENTRE FISCALÍA GENERAL DE LA NACIÓN Y LA EMPRESA EUPHORIANET S.A.S, CUYO OBJETO ES PRESTAR EL SERVICIO ESPECIALIZADO PARA EL DESARROLLO DE REQUERIMIENTOS EN LOS SISTEMAS DE INFORMACIÓN Y MEJORAMIENTO DE LOS PROCESOS ACTUALMENTE CONSTRUIDOS DE LA FISCALÍA GENERAL DE LA NACIÓN.

14/10/2020

1. Contenido

2. ALCANCE **3**

3. ACTIVIDAD DE DESARROLLO **4**

3.1 Análisis inicial y hallazgos *4*

3.2 Proceso sugerido para el particionamiento *7*

3.3 Ventajas y Desventajas del procedimiento *13*

3.4 Otras opciones *14*

2. ALCANCE

Analizar y encontrar planes de trabajo optimos para mover la tabla: ORDENESPJ_DIA_RESUMEN de su tablespace actual a un nuevo tablespace de datos.

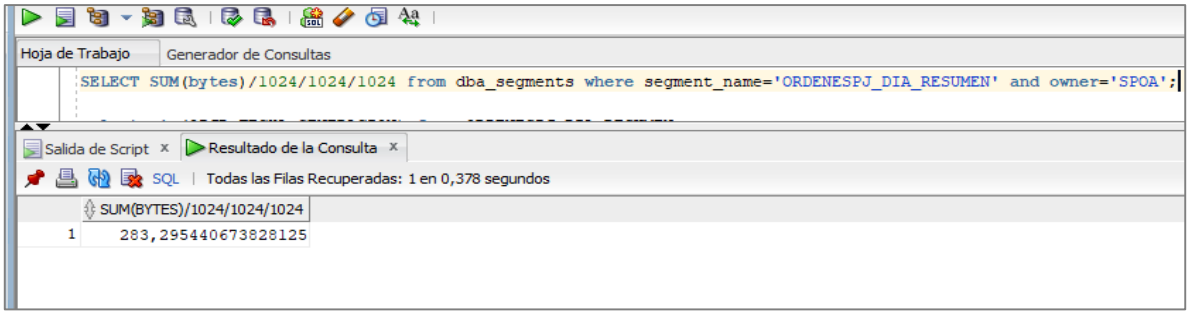
El objetivo de la tarea es reducir la fragmentación de la tabla, y MINIMIZAR la probabilidad de la aparición del error: ORA-01653.

3. ACTIVIDAD DE DESARROLLO

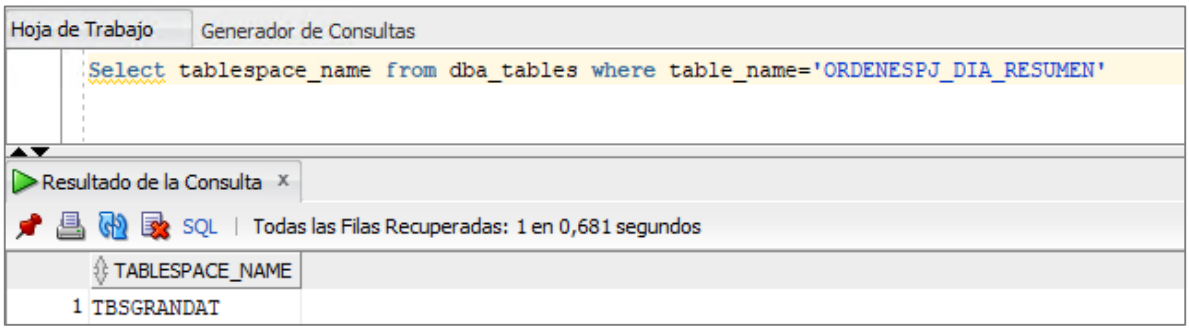
3.1 Análisis inicial y hallazgos

El análisis de la tabla: SPOA.ORDENESPJ_DIA_RESUMEN entrega los siguientes datos:

La tabla tiene un tamaño de 283 GB y no se encuentra particionada



La tabla se encuentra ubicada en el tablespace:TBSGRANDAT tal como se puede apreciar en la siguiente imagen:



La tabla tiene la llave primaria del tipo UK (Unique Key) de nombre: OPJ_UK compuesta por las columnas: OPJD_DIA + OPJD_ORPJ_ID + OPJD_ACIN_ID, como se puede apreciar en la siguiente imagen:

Página de bienvenida | conexion_soporte_dba | ORDENESPJ_DIA_RESUMEN

Columnas | Datos | Model | Restricciones | Permisos | Estadísticas | Disparadores | Flashback | Dependencias | Detalles | Particiones | Índices | SQL

Acciones...

	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_OWNER	R_TABLE_NAME	R_CONSTRAINT_NAME	DELETE_RULE	STATU
1	OPJD_UK	Unique	(null)	(null)	(null)	(null)	(null)	ENABLE
2	SYS_C00557998	Check	"OPJD_SISTEMA" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLE
3	SYS_C00557999	Check	"OPJD_DIA" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLE
4	SYS_C00558000	Check	"OPJD_CASO_ID" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLE
5	SYS_C00558001	Check	"OPJD_CASO_NOTICIA" IS NOT NULL	(null)	(null)	(null)	(null)	ENABLE

Columnas

Refrescar: 0

	COLUMN_NAME	COLUMN_POSITION
1	OPJD_DIA	1
2	OPJD_ORPJ_ID	2
3	OPJD_ACIN_ID	3

A nivel de tablespaces se identifica un alto índice de fragmentación, tal como se puede apreciar a continuación:

	TABLESPACE_NAME	FREE_CHUNKS	LARGEST_CHUNK	FRAGMENTATION_INDEX
1	TBSGRANDAT	68755	724,48	0,3
2	TBSGRANAUD	1350955	10158,08	0,32
3	TBS_AUD_PARTITION	8655	4063,23	0,43
4	TBSMEDIAUD	247798	9373,44	0,56
5	TBSPEQUAUD	92709	2932	0,78
6	UNDOTBS1	3903	4063,23	0,82
7	TBSCASOAUD	11783	10158,08	0,84
8	TBS_UNIFIED_AUDIT_01	369	2097,15	1,4
9	TBSPEQUIND	68076	10158,08	2,36
10	UNDOTBS2	4426	4063,23	2,6
11	TBSMEDIDAT	3553	4063,23	3,07
12	TBSPEQUDAT	3168	5897,12	3,15
13	SYSAUX	3609	4063,23	3,2
14	TBSGRANAUDIND	1023	6625,12	4,84
15	TBSCASODAT	626	10158,08	6,63
16	USERS	72	4063,23	11,11
17	TBSUSER	193	10158,08	16,16
18	TBSGRANIND	13	8192	17,79

Encabezando el TOP de los tablespace con mayor fragmentación se encuentra el tablespace TBSGRANDAT, en el cual reside la tabla ORDENESPJ_DIA_RESUMEN (Se considera una alta fragmentación cuando el índice de fragmentación está por debajo de 20)

La tabla en mención adicional a la llave UNIQUE_KEY tiene tres (3) índices adicionales que en total suman cerca de 45 GB, índices que tampoco se encuentran particionados.

Hoja de Trabajo | Generador de Consultas

```
select index_name ,(Select Round(SUM(BYTES)/1024/1024/1024,0)
                    From dba_segments where segment_name=tabidx.index_name) "Size_OBJ_GB"
from dba_indexes tabidx
where tabidx.table_name='ORDENESPJ_DIA_RESUMEN'
```

Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 4 en 0,443 segundos

	INDEX_NAME	Size_OBJ_GB
1	OPJD_UK	12
2	OPJD_DIA_SECC_I	12
3	OPJD_DIA_ANIO_I	12
4	OPJD_MES_GENERACION_I	9

Los parámetros PCTFREE e INITRANS para la tabla y sus índices se encuentra definidos en 10 y 2 (Los valore por default), lo que puede llegar a originar problemas de contención en eventos de alta concurrencia sobre los bloques de datos de la tabla y/o índices.

Bajo las actuales condiciones se sugiere el siguiente plan de acción:

- Crear un nuevo TableSpace para allí alojar las tablas particionadas de gran tamaño con nombre: TBS_PART_DAT01
- Crear un nuevo TableSpace para allí alojar los índices particionados asociados a las tablas de gran tamaño con nombre: TBS_PART_IDX01
- Particionar la tabla: ORDENESPJ_DIA_RESUMEN y todos sus índices asociados a través del paquete de base de datos DBMS_REDEFINITION con el fin de poder mover en "tiempo real" la tabla en mención al nuevo tablespace TBS_PART_DAT01
- Cambiar el Valor del Parametro INITRANS y PCTFREE para la tabla

3.2 Proceso sugerido para el particionamiento

A continuación se describe el proceso que se debiere seguir para hacer el movimiento de la tabla ORDENESPJ_DIA_RESUMEN al nuevo tablespace que incluye al mismo tiempo el particionamiento de la tabla. Es de anotar que dicho proceso se debe probar en un ambiente de pruebas con el fin de establecer los tiempos de procesamiento y los ajustes que hubiere lugar antes de aplicar en producción

```
-----------------------------------------------------------------
Paso 1: Validar que la Tabla se Pueda Redefinir
-----------------------------------------------------------------

EXEC Dbms_Redefinition.Can_Redef_Table('SPOA','ORDENESPJ_DIA_RESUMEN');
```

```
SQL> EXEC Dbms_Redefinition.Can_Redef_Table('SPOA','ORDENESPJ_DIA_RESUMEN');
BEGIN Dbms_Redefinition.Can_Redef_Table('SPOA','ORDENESPJ_DIA_RESUMEN'); END;

*
ERROR at line 1:
ORA-12089: cannot online redefine table "SPOA"."ORDENESPJ_DIA_RESUMEN" with no
primary key
ORA-06512: at "SYS.DBMS_REDEFINITION", line 244
ORA-06512: at "SYS.DBMS_REDEFINITION", line 5516
ORA-06512: at line 1
```

Como se observa la tabla no está preparada para poder usar DBMS_REDEFINITION. Se exige como condición que la tabla tenga una llave primaria. En caso tal se debe cambiar el tipo UNIQUE_KEY POR PRIMARY KEY.

En un escenario hipotético se deberían ejecutar las siguientes sentencias SQL:

```
SQL> CREATE INDEX SPOA.OPJD_ID_FECHA ON
SPOA.ORDENESPJ_DIA_RESUMEN(OPJD_ACIN_ID,OPJD_ORPJ_ID,OPJD_ACIN_FECHA)
PCTFREE 20 INITRANS 20 TABLESPACE TBS_PART_IDX01;

SQL>ALTER TABLE SPOA.ORDENESPJ_DIA_RESUMEN ADD CONSTRAINT PK_OPJD
PRIMARY KEY (OPJD_ACIN_ID,OPJD_ORPJ_ID,OPJD_ACIN_FECHA) USING INDEX
SPOA.OPJD_ID_FECHA;
```

Realizado el paso anterior se pude validar nuevamente que la tabla se pueda redefinir, tal como se ve en la siguiente imagen:

```
SQL> EXEC Dbms_Redefinition.Can_Redef_Table('SPOA','ORDENESPJ_DIA_RESUMEN');

PL/SQL procedure successfully completed.

SQL>
```

Paso 2: Crear Tabla Interina Particion Diaria

```
CREATE TABLE SPOA.ORDENESPJ_DIA_RES_INT
(
  OPJD_SISTEMA                VARCHAR2(10 BYTE) ,
  OPJD_DIA                    NUMBER(8)         ,
  OPJD_CASO_ID                NUMBER(10)        ,
  OPJD_CASO_NOTICIA           VARCHAR2(23 BYTE) ,
  OPJD_CASO_PRIORIZADO        VARCHAR2(1 BYTE),
  OPJD_CASO_ESNO_ID           NUMBER(6),
  OPJD_CASO_ESNO_NOMBRE       VARCHAR2(25 BYTE),
  OPJD_DIFI_ID_ASIGNADA       NUMBER(6),
  OPJD_DIFI_NOMBRE_ASIGNADA   VARCHAR2(300 BYTE),
  OPJD_SECC_ID_ASIGNADA       NUMBER(6),
  OPJD_SECC_NOMBRE_ASIGNADA   VARCHAR2(300 BYTE),
  OPJD_UNID_ID_ASIGNADA       NUMBER(12),
  OPJD_UNID_NOMBRE_ASIGNADA   VARCHAR2(300 BYTE),
  OPJD_DESP_ID_ASIGNADO       NUMBER(12),
  OPJD_DESP_NOMBRE_ASIGNADO   VARCHAR2(300 BYTE),
  OPJD_DIFI_ID_IMPARTE        NUMBER(6),
  OPJD_DIFI_NOMBRE_IMPARTE    VARCHAR2(300 BYTE),
  OPJD_SECC_ID_IMPARTE        NUMBER(6),
  OPJD_SECC_NOMBRE_IMPARTE    VARCHAR2(300 BYTE),
  OPJD_UNID_ID_IMPARTE        NUMBER(12),
  OPJD_UNID_NOMBRE_IMPARTE    VARCHAR2(300 BYTE),
  OPJD_DESP_ID_IMPARTE        NUMBER(12),
  OPJD_DESP_NOMBRE_IMPARTE    VARCHAR2(300 BYTE),
  OPJD_DEPA_ID_IMPARTE        NUMBER(4),
  OPJD_DEPA_NOMBRE_IMPARTE    VARCHAR2(120 BYTE),
  OPJD_MUNI_ID_IMPARTE        NUMBER(4),
  OPJD_MUNI_NOMBRE_IMPARTE    VARCHAR2(120 BYTE),
  OPJD_ACIN_ID_ULTIMA         NUMBER(10),
  OPJD_ACTU_ID_ULTIMA         NUMBER(6),
  OPJD_ACTU_NOMBRE_ULTIMA     VARCHAR2(300 BYTE),
  OPJD_ACIN_FECHA_ULTIMA      DATE,
  OPJD_ACIN_FECHA_REG_ULTIMA  DATE,
```

```
  OPJD_ACIN_ID                NUMBER(10),
  OPJD_ACTU_ID                NUMBER(6),
  OPJD_ACTU_NOMBRE            VARCHAR2(300 BYTE),
  OPJD_ACIN_TERMINO           NUMBER(4),
  OPJD_ACIN_FECHA             DATE,
  OPJD_ACIN_FECHA_REGISTRO    DATE,
  OPJD_ACIN_FECHA_LIMITE      DATE,
  OPJD_ACIN_DIAS_PRORROGA     VARCHAR2(200 BYTE),
  OPJD_ACIN_DIAS_RESTANTES    NUMBER(8),
  OPJD_ACIN_COLOR             VARCHAR2(12 BYTE),
  OPJD_ORPJ_ID                NUMBER(12),
  OPJD_FECHA_GENERACION       DATE,
  OPJD_ANIO_GENERACION        NUMBER(4),
  OPJD_MES_GENERACION         NUMBER(6),
  OPJD_FUCA_ID                NUMBER(12),
  OPJD_FUNC_ID                NUMBER(12),
  OPJD_FUNC_USUARIO           VARCHAR2(30 BYTE),
  OPJD_FUNC_ESTADO            VARCHAR2(1 BYTE),
  OPJD_FUNC_ESTADO_NOMBRE     VARCHAR2(20 BYTE),
  OPJD_FUNC_PERS_ID           NUMBER(14),
  OPJD_FUNC_NUMDOC            VARCHAR2(20 BYTE),
  OPJD_FUNC_NOMBRES           VARCHAR2(300 BYTE),
  OPJD_FUNC_DIFI_ID           NUMBER(6),
  OPJD_FUNC_DIFI_NOMBRE       VARCHAR2(300 BYTE),
  OPJD_FUNC_SECC_ID           NUMBER(6),
  OPJD_FUNC_SECC_NOMBRE       VARCHAR2(300 BYTE),
  OPJD_GRPJ_DIFI_ID           NUMBER(6),
  OPJD_GRPJ_DIFI_NOMBRE       VARCHAR2(300 BYTE),
  OPJD_GRPJ_SECC_ID           NUMBER(6),
  OPJD_GRPJ_SECC_NOMBRE       VARCHAR2(300 BYTE),
  OPJD_GRPJ_SECCION_ID        NUMBER(6),
  OPJD_GRPJ_SECCION_NOMBRE    VARCHAR2(300 BYTE),
  OPJD_GRPJ_ID                NUMBER(6),
  OPJD_GRPJ_NOMBRE            VARCHAR2(300 BYTE),
  OPJD_ORTR_ID                NUMBER(18),
  OPJD_ORTR_NUMERO_OT         NUMBER(12),
  OPJD_ORTR_ESOR_ID           CHAR(1 BYTE),
  OPJD_ORTR_ESOR_NOMBRE       VARCHAR2(30 BYTE),
  OPJD_ORTR_FECHA_ASIGNACION  DATE
) TABLESPACE TBS_PART_DAT01 PCTFREE    20 INITRANS   10
PARTITION BY RANGE (OPJD_ACIN_FECHA) INTERVAL(numtodsinterval(1, 'DAY'))
(
 PARTITION PART_01 VALUES LESS THAN (TO_DATE('1-1-2000', 'DD-MM-YYYY'))
TABLESPACE TBS_PART_DAT01 ,
   PARTITION PART_02 VALUES LESS THAN (TO_DATE('1-2-2000', 'DD-MM-YYYY'))
TABLESPACE TBS_PART_DAT01

)
```

Paso 3: Iniciar Proceso de Redefinicion

```
EXEC  DBMS_REDEFINITION.START_REDEF_TABLE('SPOA',  'ORDENESPJ_DIA_RESUMEN'
,'ORDENESPJ_DIA_RES_INT');
```

```
SQL> EXEC DBMS_REDEFINITION.START_REDEF_TABLE('SPOA', 'ORDENESPJ_DIA_RESUMEN' ,'ORDENESPJ_DIA_RES_INT');

PL/SQL procedure successfully completed.

SQL>
```

Paso 4: Crear Dependencias con la Tabla Interina

```
DECLARE
  retval NUMBER(5);

BEGIN
   dbms_redefinition.copy_table_dependents('SPOA',
'ORDENESPJ_DIA_RESUMEN', 'ORDENESPJ_DIA_RES_INT',
                copy_indexes=>0,
                copy_triggers=>TRUE,
                copy_constraints=>TRUE,
                copy_privileges=>TRUE,
                ignore_errors=>FALSE,
                num_errors=>retval);
   dbms_output.put_line('Errores Encontrados Definicion de Dependencia
'||retval);
END;
```

```
SQL> DECLARE
  retval NUMBER(5);

BEGIN
   dbms_redefinition.copy_table_dependents('SPOA', 'ORDENESPJ_DIA_RESUMEN', 'ORDENESPJ_DIA_RES_INT',
                copy_indexes=>0,
                copy_triggers=>TRUE,
  2    3    4    5    6    7    8                copy_constraints=>TRUE,
                copy_privileges=>TRUE,
                ignore_errors=>FALSE,
                num_errors=>retval);
   dbms_output.put_line('Errores Encontrados Definicion de Dependencia '||retval);
END;
  9   10   11   12   13   14
 15  /

PL/SQL procedure successfully completed.
```

Paso 5: Sincronizar Tablas

```
EXEC DBMS_REDEFINITION.SYNC_INTERIM_TABLE('SPOA',
'ORDENESPJ_DIA_RESUMEN', 'ORDENESPJ_DIA_RES_INT');
```

Paso 6: Finalizar el Proceso de Redefinicion

```
EXEC DBMS_REDEFINITION.FINISH_REDEF_TABLE('SPOA','ORDENESPJ_DIA_RESUMEN',
'ORDENESPJ_DIA_RES_INT')
```

```
SQL> EXEC DBMS_REDEFINITION.SYNC_INTERIM_TABLE('SPOA', 'ORDENESPJ_DIA_RESUMEN', 'ORDENESPJ_DIA_RES_INT');

PL/SQL procedure successfully completed.

SQL> EXEC DBMS_REDEFINITION.FINISH_REDEF_TABLE('SPOA','ORDENESPJ_DIA_RESUMEN', 'ORDENESPJ_DIA_RES_INT');

PL/SQL procedure successfully completed.

SQL>
```

**Paso 7: VERIFICAR QUE LAS PARTICIONES DE LA TABLA FUERON CREADAS
Y ALEATORIAMENTE ELEGIR
PARTICIONES QUE VALIDEN QUE EN ESTAS EXISTEN REGISTROS**

```
SQL> set linesize 120
SQL> col PARTITION_NAME format a20
SQL> col HIGH_VALUE format 99999999999


SELECT  PARTITION_NAME, HIGH_VALUE,NUM_ROWS FROM DBA_TAB_PARTITIONS WHERE
TABLE_NAME='ORDENESPJ_DIA_RESUMEN' AND TABLE_OWNER='SPOA';

SQL> SELECT COUNT(*) FROM SPOA.ORDENESPJ_DIA_RESUMEN PARTITION(PART_X1);
SQL> SELECT COUNT(*) FROM SPOA.ORDENESPJ_DIA_RESUMEN PARTITION(PART_X2);
SQL> SELECT COUNT(*) FROM SPOA.ORDENESPJ_DIA_RESUMEN PARTITION(PART_Xn);
```

```
-----------------------------------------------------------------------
Paso 8: Re-crear Indices Particionados
-----------------------------------------------------------------------
CREATE INDEX SPOA.OPJD_DIA_ANIO_I ON  "SPOA"."ORDENESPJ_DIA_RESUMEN"
("OPJD_DIA", "OPJD_ANIO_GENERACION")                    TABLESPACE
TBS_PART_IDX01   PCTFREE 20 INITRANS  10 LOCAL PARALLEL 2;

CREATE INDEX SPOA.OPJD_DIA_SECC_I ON "SPOA"."ORDENESPJ_DIA_RESUMEN" ("OPJD_DIA",
NVL("OPJD_GRPJ_SECC_ID","OPJD_FUNC_SECC_ID")) TABLESPACE TBS_PART_IDX01
PCTFREE 20 INITRANS  10 LOCAL PARALLEL 2;

CREATE INDEX SPOA.OPJD_MES_GENERACION_I  ON "SPOA"."ORDENESPJ_DIA_RESUMEN"
("OPJD_MES_GENERACION")                          TABLESPACE TBS_PART_IDX01
PCTFREE 20 INITRANS  10 LOCAL PARALLEL 2;



ALTER TABLE SPOA.ORDENESPJ_DIA_RESUMEN DISABLE CONSTRAINT  OPJD_UK

CREATE UNIQUE INDEX SPOA.OPJD_UK ON SPOA.ORDENESPJ_DIA_RESUMEN (OPJD_DIA,
OPJD_ORPJ_ID, OPJD_ACIN_ID)

TABLESPACE TBS_PART_IDX01 PCTFREE 20 INITRANS  10 ;


-----------------------------------------------------------------------
Paso 9: Activar Enable Row Movement
-----------------------------------------------------------------------

ALTER TABLE SPOA.ORDENESPJ_DIA_RESUMEN ENABLE ROW MOVEMENT

-- MOVER REG. ENTRE LAS PARTICIONES

-----------------------------------------------------------------------
Paso 10: RECREAR ESTADISTICAS DE LA TABLA
-----------------------------------------------------------------------
exec dbms_stats.gather_table_stats(ownname=>'SPOA',-
     tabname=>'ORDENESPJ_DIA_RESUMEN',-
     estimate_percent => dbms_stats.auto_sample_size,-
     cascade=>true,-
     method_opt=>'for all columns size auto',-

    degree=>3);
```

Paso 10: ELIMINAR TABLA INTERINA USADA PARA EL PROCESO

```
SQL> DROP TABLE SPOA.ORDENESPJ_DIA_RES_INT;

Table dropped.
```

3.3 Ventajas y Desventajas del procedimiento

Ventajas

Las ventajas que conlleva el mover la tabla SPOA.ORDENESPJ_DIA_RESUMEN a un nuevo tablespace son:

- Reducción de la fragmentación de la tabla y los índices asociados a la tabla:
- Minimizar las posibles apariciones del error: ORA-1653.
- Aplicar las buenas prácticas de Oracle que sugiere que tablas de más de 4 GB de tamaño son candidatas a ser particionadas.

Desventajas

- Se requiere de al menos 350 GB de espacio disponible para poder ejecutar las operaciones.
- No se garantiza que la ejecución de la tarea resuelva por completo la aparición del error ORA-1653 para las operaciones que se realicen sobre tabla en mención
- Puede generar contención y/o lentitud sobre los procesos que actualmente ejecutan los usuarios sobre la tabla: SPOA.ORDENESPJ_DIA_RESUMEN cuando se esté realizando la tarea del movimiento de la tabla al nuevo tablespace.
- Se desconoce el tiempo real del proceso, hasta tanto no se hagan las pruebas correspondientes en un ambiente similar que permita determinar los tiempos aproximados para realizar el movimiento de la tabla a un nuevo tablespace

- Requiere que se defina una llave Primaria para la tabla: SPOA.ORDENESPJ_DIA_RESUMEN y en lo posible que dicha llave pueda ser única en cada partición con el fin de optimizar el acceso a la tabla y no tener que usar índices globales.
- En caso de proceder con el particionamiento se debe evaluar si el índice del tipo UNIQUE_KEY que actualmente esta puede ser remplazado por la llave primaria que se haya definido.

3.4 Otras opciones

- Crear una tabla con la estructura de la tabla particionada incluyendo los índices particionados y apuntando a un nuevo TableSpace
- Realizar un volcado de la tabla actual sobre la nueva estructura a través de un EXPORT-IMPORT y al final proceder con un RENAME del objeto.

La gran desventaja de este método radica en el tiempo de procesamiento que puede durar la tarea, tiempo que se considerar como tiempo de indisponibilidad total sobre la tabla mientras se realiza el backup y la restauración.