

PRÁCTICA #3

PL/SQL



► Ejercicios-Sesión#2

1. PL-SQL – SESIÓN 2



Ejercicio 3. Crear un procedimiento que tenga como parámetros de entrada el nombre de una compañía y una fecha. Dicho procedimiento debe realizar las siguientes operaciones:

1. Comprobar que existen en la BD llamadas realizadas en la fecha que se pasa como parámetro. En caso contrario lanzar una ⚠ excepción y mostrar el mensaje “No hay llamadas de teléfonos de la compañía <compañía> en la fecha <fecha> en la BD”.
2. Para cada teléfono de la compañía que pasa por parámetro, el procedimiento debe mostrar la siguiente información: número de teléfono, número total de llamadas realizadas en la fecha indicada, número de llamadas de duración mayor de 100 segundos realizadas en la fecha, porcentaje que suponen estas últimas respecto al total de las realizadas.
3. Un resumen del número de llamadas realizadas por todos los teléfonos de la compañía indicada en la fecha pasada por parámetro.

1. PL-SQL – SESIÓN 2

Ejemplo de ejecución:

```
SQL> exec llamadas_cia('Aotra', '01/10/06');
```

Llamadas realizadas por los teléfonos de la Compañía "Aotra"

Tlf. Origen	Num_LL	Num_100	%
678111222	0	0	0%
678234234	2	2	100%

Numero Llamadas Realizadas: 2

```
SQL> exec llamadas_cia('Kietostar', '16/10/06');
```

Llamadas realizadas por los teléfonos de la Compañía "Kietostar"

Tlf. Origen	Num_LL	Num_100	%
654123321	2	1	50%
654789789	0	0	0%
654234234	0	0	0%
654012012	0	0	0%
654345345	1	0	0%

Numero Llamadas Realizadas: 3



Análisis de la salida generada por el procedimiento

VARIABLES:

num_LL_Total
(COUNT*)

num_LL_100
¿(duración>100)?

porcentaje

$$\frac{\text{num_LL_100}}{\text{num_LL_Total}} * 100$$

```
SQL> exec llamadas_cia('Aotra', '01/10/06');
```

Llamadas realizadas por los teléfonos de la Compañía "Aotra"

Tlf. Origen	Num_LL	Num_100	%
678111222	0	0	0%
678234234	2	2	100%

Numero Llamadas Realizadas: 2

Datos solicitados relativos a las llamadas realizadas por cada uno de los teléfonos de los abonados de 'Aotra' el '01/10/06'

Lista de teléfonos de la compañía 'Aotra'

CURSOR C_TELEFONOS_CIA

num_Total_Cia
VARIABLE TOTAL

CURSOR C_LLAMADAS_TF ('678111222')

El cursor c_llamadas_tf tomará como parámetro el teléfono que se esté tratando en el cursor c_telefonos_cia, que será del que se obtendrán los datos solicitados relativos a sus llamadas



→ CURSOR C_TELEFONOS_cia

1

Sin Título6.sql

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```
1 SELECT TEL.numero
2 FROM COMPAÑIA C INNER JOIN TELEFONO TEL
3 ON C.cif = TEL.compañia
4 WHERE C.nombre = 'Aotrá';
```

Resultado de la Consulta

Todas las Filas Recuperadas: 2 en 0,935 segundos

FOR1

NUMERO
678111222
678234234

tf_origen

Lista de teléfonos de la compañía 'Aotrá'

num_LL_Total = 2
num_LL_100 = 2
porcentaje = 100%
num_Total_Cia = 2

1

Sin Título6.sql

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```
1 SELECT tf_origen, tf_destino, duracion
2 FROM LLAMADA LLAM
3 WHERE TO_CHAR(fecha_hora, 'dd/mm/yy') = '01/10/06'
4 AND tf_origen = '678111222';
```

Resultado de la Consulta

Todas las Filas Recuperadas: 2 en 0,137 segundos

CURSOR C_LLAMADAS_TF ('678111222')

TF_ORIGEN TF_DESTINO DURACION

num_LL_Total = 0
num_LL_100 = 0
porcentaje = 0%
num_Total_Cia = 0

Llamadas de teléfono de abonados de 'Aotrá' realizadas el '01/10/06'

2

Sin Título6.sql

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```
1 FROM LLAMADA LLAM
2 WHERE TO_CHAR(fecha_hora, 'dd/mm/yy') = '01/10/06'
3 AND tf_origen = '678234234';
```

Resultado de la Consulta

Todas las Filas Recuperadas: 2 en 0,137 segundos

CURSOR C_LLAMADAS_TF ('678234234')

FOR2

TF_ORIGEN	TF_DESTINO	DURACION
678234234	654123321	2045
678234234	654789789	703

>100
>100



Análisis del funcionamiento del procedimiento

1 → **CURS**OR C_LLAMADAS_TF ('678234234')

2

Hoja de Trabajo de SQL Historial

Hoja de Trabajo Generador de Consultas

```

2 FROM LLAMADA LLAM
3 WHERE TO_CHAR(fecha_hora,'dd/mm/yy') = '01/10/06'
4 AND tf_origen = '678234234';
    
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,137 segundos

	TF_ORIGEN	TF_DESTINO	DURACION
1	678234234	654123321	2045
2	678234234	654789789	703

FOR2



TF. Origen	NumLL	Num100	%
678111222	0	0	0%
678234234	2	2	100%

CURSOR C_TELEFONOS_CIA
↓
FOR1

...
#Total_Cia = 0

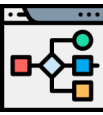
Para cada tupla del Cursor c_telefonos_cia hacer
#LL_Total=0; #LL_100=0; **CORRELACIÓN** ⚠

Para cada tupla del Cursor
c_llamadas_tf('678234234') hacer **FOR**2
si (c2tupla.duracion>100) entonces
#LL_100++; **CURS**OR C_LLAMADAS_TF ('678234234')
f_si
#LL_Total++; #Total_Cia++;



f_para
si (#LL_100>0) entonces
porcentaje=(#LL_100/#LL_Total)*100;
si no
porcentaje = 0;
f_si;
mostrar_linea_detalle_por_pantalla()
f_para
mostrar_linea_Total_Cia_por_pantalla()
...





Implementación del procedimiento



Definimos la estructura general de nuestro procedimiento:

CREATE or REPLACE

PROCEDURE llamadas_cia (**cia** COMPAÑIA.nombre%type, **fecha** DATE)

←  **PARÁMETROS**  →

/* Declaraciones locales*/

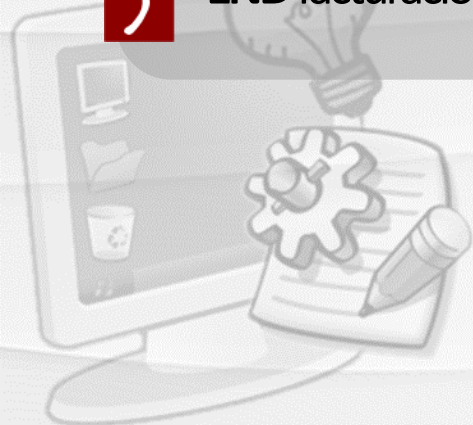
1 BEGIN

/* Sentencias */

2 EXCEPTION

/* Tratamiento de Excepciones*/

3 END facturación;



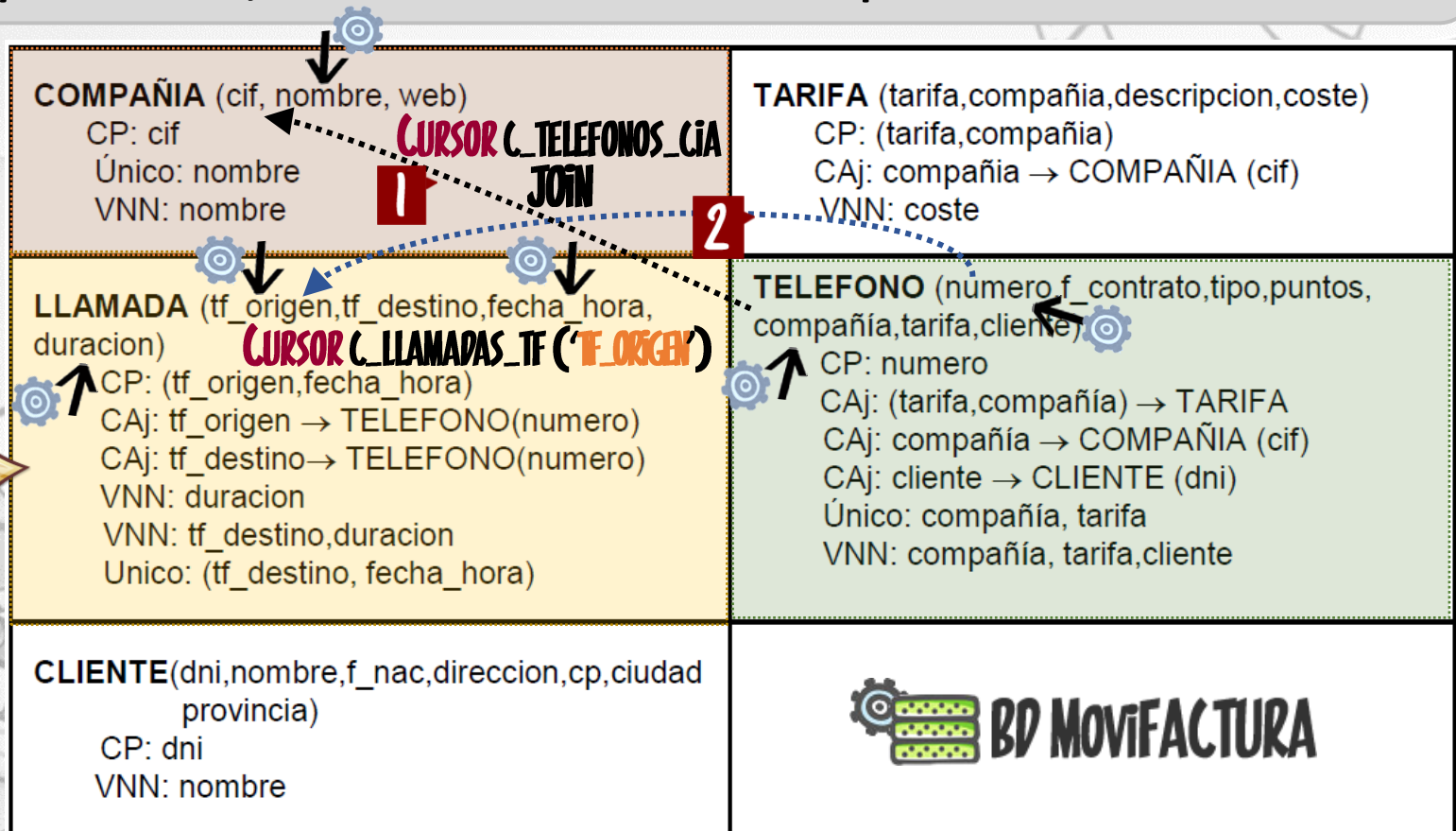


2



Observando la estructura de la BD y los datos que hemos de manejar en la consulta, se deduce que, en primer lugar, será necesario realizar un INNER JOIN de las tablas **TELÉFONO** y **COMPANIA** para obtener el conjunto de *teléfonos* de la compañía especificada como *parámetro en la llamada al procedimiento*.

A continuación, para cada uno de los teléfonos obtenidos, y considerando sólo las llamadas realizadas en la fecha especificada como *parámetro en la llamada al procedimiento*, calcularemos los distintos valores que se nos solicita.





Como se nos pide mostrar información sobre las LLAMADAS realizadas por los teléfonos de la compañía que se le pase como parámetro al procedimiento, lo primero que haremos será definir, en la *zona de declaraciones locales*, un cursor, `c_teléfonos_cia`, cuya consulta asociada deberá devolver el conjunto de teléfonos pertenecientes a la compañía cuyo **nombre** se le pase como parámetro al procedimiento:



CREATE OR REPLACE

PROCEDURE llamadas_cia (**cia** COMPAÑIA.nombre%type, **fecha** DATE)

CURSOR c_teléfonos_cia IS

SELECT número

FROM COMPAÑIA C INNER JOIN TELEFONO TEL

ON C.cif = TEL.compañia

WHERE C.nombre = **cia**;

⚠ CORRELACIÓN

A continuación, para cada uno de los teléfonos obtenidos mediante el cursor anterior, tendremos que obtener sus LLAMADAS realizadas en la **fecha** especificada como parámetro del procedimiento. Para ello, definiremos, en la *zona de declaraciones locales*, un cursor parametrizado, `c_teléfonos_cia`, que tomará como parámetro el *número de teléfono* que estemos tratando en el cursor `llamadas_cia`, el cual se corresponderá con el *teléfono origen* de la llamadas:

2



CURSOR c_llamadas_tf (**tff** LLAMADA.tf_origen%type) IS

SELECT tf_origen, tf_destino, duración

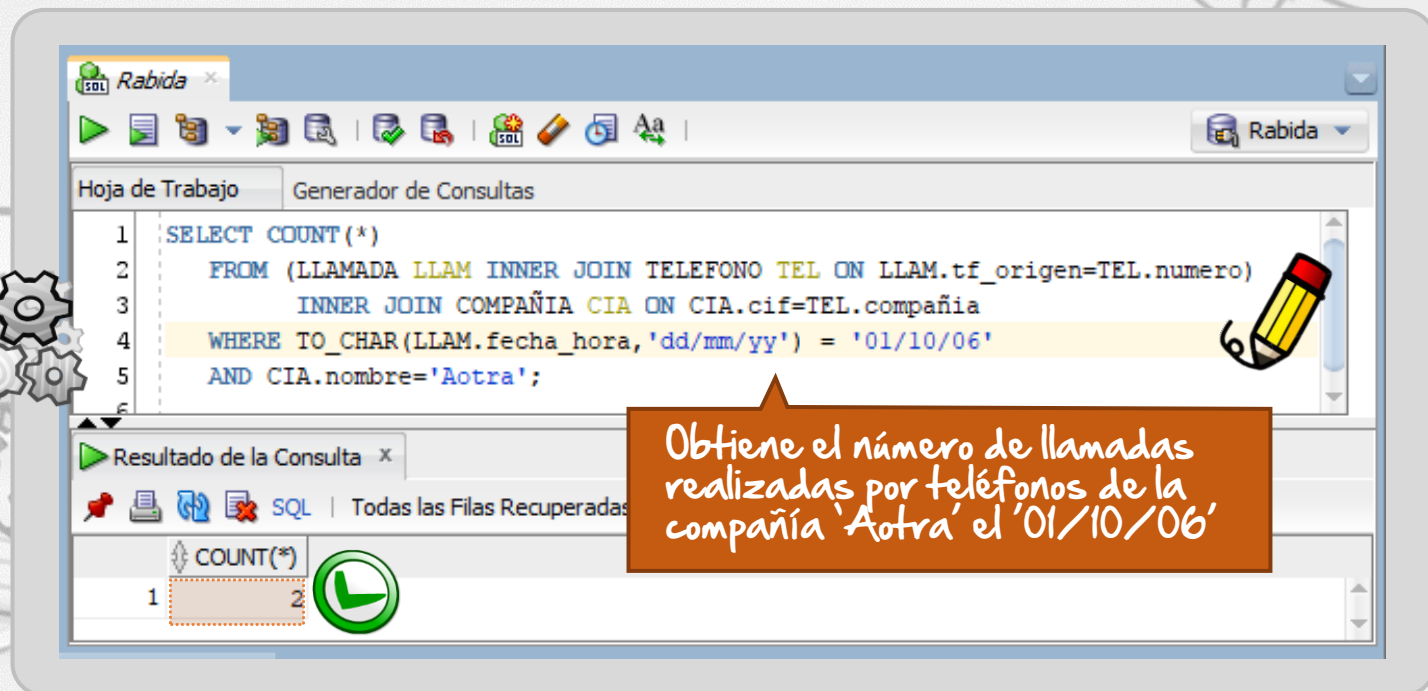
FROM LLAMADA LLAM

WHERE TO_CHAR(fecha_hora,'dd/mm/yy') = **fecha** and tf_origen = **tff**;



Se propone controlar, mediante el uso de excepciones, el hecho de que no existan llamadas realizadas en la fecha especificada en la llamada al procedimiento. Para ello, definiremos la excepción `no_existe_llamadas_fecha` en la zona de declaraciones y definiremos un cursor implícito que recupere el número de registros existentes de llamadas realizadas en la fecha indicada por teléfonos de la compañía especificada, y almacene el valor obtenido en la variable `filas_llamadas_fecha`, que también declararemos en la zona de declaraciones, de manera que, si dicho valor es igual a 0, se deberá activar la excepción `no_existe_llamadas_fecha`. Su tratamiento consistirá, simplemente, en mostrar, por pantalla un mensaje informativo.

► La definición y resultado de la consulta asociada al cursor para los parámetros del procedimiento de nuestro ejemplo sería el siguiente:



The screenshot shows the SQL Developer interface with a query window titled 'Rabida'. The query is as follows:

```
1 SELECT COUNT(*)
2 FROM (LLAMADA LLAM INNER JOIN TELEFONO TEL ON LLAM.tf_origen=TEL.numero)
3      INNER JOIN COMPAÑIA CIA ON CIA.cif=TEL.compañia
4 WHERE TO_CHAR(LLAM.fecha_hora, 'dd/mm/yy') = '01/10/06'
5 AND CIA.nombre='Aotra';
```

Below the query, the 'Resultado de la Consulta' window shows the result of the query. The result is a single row with the value 2 in the column 'COUNT(*)'.

COUNT(*)
2

An orange callout box points to the result, stating: 'Obtiene el número de llamadas realizadas por teléfonos de la compañía 'Aotra' el '01/10/06''.



▶ La definición y tratamiento de la excepción sería el siguiente:

1

```
filas_llamadas_fecha INTEGER;  
no_existe_llamadas_fecha EXCEPTION;
```



Se controla la excepción al principio del Bloque de Sentencias

2**BEGIN**

```
SELECT COUNT(*) INTO filas_llamadas_fecha  
FROM (LLAMADA LLAM INNER JOIN TELEFONO TEL  
      ON LLAM.tf_origen=TEL.numero)  
      INNER JOIN COMPAÑIA CIA ON CIA.cif=TEL.compañia  
WHERE TO_CHAR(LLAM.fecha_hora,'dd/mm/yy') = fecha  
      AND CIA.nombre= cia;
```



```
IF filas_llamadas_fecha = 0 THEN  
  RAISE no_existe_llamadas_fecha;  
END IF;
```



3**EXCEPTION**

```
WHEN no_existe_llamadas_fecha THEN  
  dbms_output.put_line('No hay llamadas del ' || fecha || ' en la BD!!:O');  
WHEN OTHERS THEN  
  dbms_output.put_line('Ha ocurrido un error!! :(');
```

```
END llamadas_cia;
```





- ↳ El funcionamiento del cuerpo del procedimiento sería el siguiente:
- ▶ En primer lugar, *definiremos un bucle FOR con el que recorreremos cada uno de los teléfonos obtenidos mediante la consulta asociada al cursor c_telefonos_cia.*
 - ⇒ Como, para cada *teléfono*, tenemos que calcular el número de llamadas realizadas, el número de llamadas con una duración superior a 100, así como el total de llamadas realizadas por todos los teléfonos de la compañía en la fecha especificada, necesitaremos definir tres variables *acumuladoras*: num_LL_Total, num_LL_100 y num_Total_Cia, con las que iremos contabilizando, cada uno de las llamadas según convenga. Estas variables se definirán en la zona de declaraciones y se inicializarán a 0 al comienzo de este bucle externo.
 - ⇒ Asimismo, será necesario declarar una variable porcentaje_100, en la que registraremos el porcentaje que supone el número de llamadas realizadas con una duración superior a 100 respecto al total de llamadas.



1

```
num_LL_Total  INTEGER;
num_LL_100    INTEGER;
num_Total_Cia INTEGER;
porcentaje    NUMBER(5,2);
```



2

```
...
BEGIN
num_Total_Cia := 0;
FOR v_telefono IN c_telefonos_cia LOOP
    num_LL_Total := 0; num_LL_100 := 0;
END LOOP;
...
↓
```




- Dentro del bucle anterior será necesario definir otro bucle para recorrer las tuplas del cursor `c_llamadas_tf`(*teléfono*), mediante el que podremos acceder a las llamadas realizadas por el teléfono indicado en la llamada al cursor, que se correspondería con el teléfono que estemos tratando en el cursor `c_telefonos_cia`, en la *fecha* especificada en el *parámetro en la llamada al procedimiento*.
- ⇒ En este bucle será donde contabilizaremos las llamadas realizadas por cada uno de los teléfonos de la compañía que se especifique en la llamada al procedimiento, cuántas de ellas han tenido una duración superior a 100, contabilizaremos el número total de llamadas y, al finalizar su ejecución, calcularemos el *porcentaje* que supone el número de llamadas realizadas con una duración superior a 100 respecto al total de llamadas.

2



```
num_Total_Cia := 0;
FOR v_telefono IN c_telefonos_cia LOOP
    num_LL_Total := 0; num_LL_100 := 0;
    FOR v_llamada IN c_llamadas_tf(v_telefono.numero) LOOP
        IF (v_llamada.duracion > 100) THEN
            num_LL_100 := num_LL_100 + 1;
        END IF;
        num_LL_Total := num_LL_Total + 1;
        num_Total_Cia := num_Total_Cia + 1;
    END LOOP;
    IF (num_LL_100 <> 0) THEN porcentaje := (num_LL_100 / num_LL_Total) * 100;
    ELSE porcentaje := 0;
    END IF;
```





2 

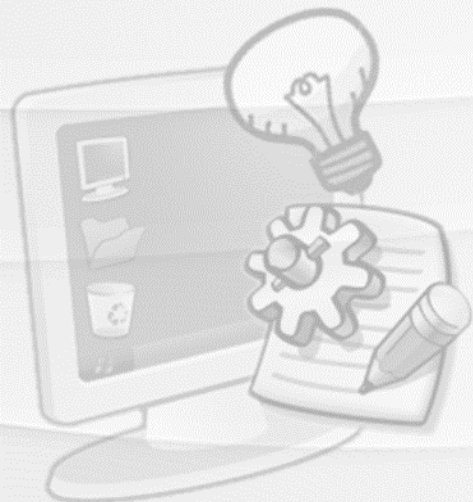


```
dbms_output.put_line( rpad(v_telefono.numero,13) || rpad(num_LL_Total,8) ||  
                      rpad(num_LL_100,13) || rpad(porcentaje || '%',7) );
```

```
END LOOP;
```

```
dbms_output.put_line('Numero Total de LLamadas:' || num_Total_Cia );
```

```
...
```





Definición final del procedimiento:



CREATE OR REPLACE

PROCEDURE **llamadas_cia** (**cia** COMPAÑIA.nombre%type,
fecha DATE) IS

Parámetros del procedimiento

/* Declaraciones locales */

num_LL_Total **INTEGER**;
num_LL_100 **INTEGER**;
num_Total_Cia **INTEGER**;
porcentaje **NUMBER**(5,2);
filas_llamadas_fecha **INTEGER**;
no_existe_llamadas_fecha **EXCEPTION**;

Variable que recoge el valor del cursor implícito

CURSOR c_teléfonos_cia **IS**

SELECT numero
FROM COMPAÑIA C **INNER JOIN** TELEFONO TEL
ON C.cif = TEL.compañia
WHERE C.nombre = **cia**;

Cursor cuya consulta asociada obtiene todos los teléfonos pertenecientes a la compañía cuyo nombre es recibido como parámetro de entrada del procedimiento

CURSOR c_llamadas_tf (**tf** LLAMADA.tf_origen%type) **IS**

SELECT tf_origen, tf_destino, duracion
FROM LLAMADA LLAM
WHERE TO_CHAR(fecha_hora, 'dd/mm/yy') = **fecha** **AND** tf_origen = **tf**;

Cursor cuya consulta recupera las llamadas realizadas por el teléfono que recibe como parámetro en la fecha de la llamada al procedimiento





2

BEGIN

/* Bloque de Sentencias */

```
SELECT COUNT(*) INTO filas_llamadas_fecha
FROM (LLAMADA LLAM INNER JOIN TELEFONO TEL
      ON LLAM.tf_origen=TEL.numero)
     INNER JOIN COMPAÑIA CIA ON CIA.cif=TEL.compañia
WHERE TO_CHAR(LLAM.fecha_hora,'dd/mm/yy') = fecha
      AND CIA.nombre= cia;
```

```
IF filas_llamadas_fecha = 0 THEN
  RAISE no_existe_llamadas_fecha;
END IF;
```

```
dbms_output.put_line('Tlf. Origen NUM_LL NUM_LL_100 PORCENT%');
dbms_output.put_line('-----');
```

```
num_Total_Cia := 0;
```

```
FOR v_telefono IN c_telefonos_cia LOOP
  num_LL_Total := 0; num_LL_100 := 0;
```

```
  FOR v_llamada IN c_llamadas_tf(v_telefono.numero) LOOP
```

```
    IF (v_llamada.duracion > 100) THEN
      num_LL_100 := num_LL_100 + 1;
```

```
    END IF;
```

```
    num_LL_Total := num_LL_Total + 1;
```

```
    num_Total_Cia := num_Total_Cia + 1;
```

```
  END LOOP;
```

❗ Consulta que contabiliza el número de llamadas realizadas en la fecha especificada en la llamada. Si el valor obtenido es cero, dispararía la excepción



❗ Bucles anidados mediante los que vamos obteniendo los datos estadísticos que ha de mostrar el procedimiento para los teléfonos de la compañía especificada

**2**

```
IF (num_LL_100 <> 0) THEN
    porcentaje := (num_LL_100 / num_LL_Total) * 100;
ELSE porcentaje := 0;
END IF;

dbms_output.put_line(rpad(v_telefono.numero, 13) || rpad(num_LL_Total, 8) ||
    rpad(num_LL_100, 13) || rpad(porcentaje || '%', 7));

END LOOP;

dbms_output.put_line('Numero Total de LLamadas : ' || num_Total_Cia );
```

3

/* Bloque de Tratamiento de Excepciones */

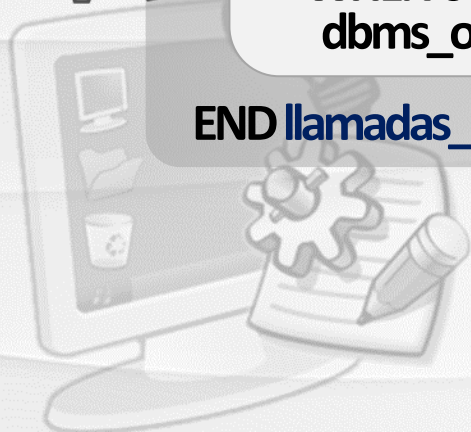
EXCEPTION

WHEN no_existe_llamadas_fecha **THEN**
dbms_output.put_line('No hay llamadas del ' || **fecha** || ' en la BD!!:O');

WHEN others **THEN**
dbms_output.put_line('Ha ocurrido un error!!:(');

END llamadas_cia;

i Capturamos las excepciones que se produzcan y mostramos un mensaje informativo





EJECUTAR PROCEDIMIENTO DESDE SQL DEVELOPER:

Oracle SQL Developer : C:\Users\Marc\AppData\Roaming\SQL Developer\Sin Título5.sql

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Ejecutamos el script

Procedimiento PL-SQL registrado

Hacemos la llamada al procedimiento con la compañía y fecha deseada

```
SET SERVEROUTPUT ON;
execute llamadas_cia('Kietostar', '16/10/06');
```

Salida de Script x

Tarea terminada en 0,227 segundos

Tlf. Origen	NUM_LL	NUM_LL_100	PORCENT%
654123321	2	1	50%
654789789	0	0	0%
654234234	0	0	0%
654012012	0	0	0%
654345345	1	0	0%

Numero Total de LLamadas:3

Procedimiento PL/SQL terminado correctamente.

Se ha guardado: C:\Users\Marc\llamadas_cia.sql | Línea 2 Columna 47 | Insertar | Modificado | Windows: CF



1. PL-SQL – SESIÓN 2

➡ **Ejercicio 4.** Crear un disparador que examine la inserción de nuevas llamadas y controle que, en la fecha y hora en la que se produce la nueva llamada, los teléfonos origen y destino no están atendiendo ni realizando ninguna llamada iniciada en esa misma fecha y hora.

Ejemplos de ejecución:




```
SQL> insert into llamada  
      values ('654345345', '678234234', '21/10/06 21:05:13', 123);
```

ERROR en línea 1:

ORA-20001: El teléfono origen 654345345 está realizando o recibiendo una llamada a las 21/10/06 21:05:13,000000

ORA-06512: en "MF.NUEVA_LLAMADA", línea 17

ORA-04088: error durante la ejecución del disparador 'MF.NUEVA_LLAMADA'



```
SQL> insert into llamada  
      values ('654345345', '678234234', '12/09/06 11:52:19', 123);
```

ERROR en línea 1:

ORA-20002: El teléfono destino 678234234 está realizando o recibiendo una llamada a las 12/09/06 11:52:19,000000

ORA-06512: en "MF.NUEVA_LLAMADA", línea 29

ORA-04088: error durante la ejecución del disparador 'MF.NUEVA_LLAMADA'



insert into mf.llamada
values ('654345345', '678234234', '21/10/06 21:05:13', 123);

Consultas previas

a

b

Cálculo del número de llamadas realizadas, en la fecha especificada, en las que participan el teléfono origen y destino que se desea registrar

TF_ORIGEN	TF_DESTINO	FECHA_HORA	DURACION
1 654123321	654345345	26/09/06 23:05:55,000000000	921
2 654123321	654345345	16/10/06 18:45:35,000000000	785
3 654123321	654345345	16/10/06 19:07:10,000000000	38
4 654123321	654345345	18/10/06 09:35:50,000000000	1053
5 654123321	654345345	21/10/06 21:05:13,000000000	257
6 666010101	666456456	27/10/06 11:41:33,000000000	435
7 666010101	654234234	01/11/06 10:31:00,000000000	124
8 666010101	666789789	05/09/06 17:09:01,000000000	210
9 654789789	654012012	01/10/06 13:55:30,000000000	75
10 654789789	666456456	05/10/06 17:05:03,000000000	430
11 654789789	678111222	08/10/06 10:55:15,000000000	51
12 678111222	654789789	07/10/06 19:25:51,000000000	51
13 678111222	654234234	03/11/06 17:45:31,000000000	63
14 678111222	678234234	12/09/06 11:52:19,000000000	63
15 678111222	678234234	23/09/06 19:17:44,000000000	101
16 666789789	666010101	11/09/06 10:19:03,000000000	45
17 666789789	666456456	15/09/06 14:12:44,000000000	409
18 666789789	666456456	15/09/06 18:13:24,000000000	105
19 654012012	678234234	13/09/06 10:32:07,000000000	312
20 654012012	678111222	01/09/06 21:08:59,000000000	93
21 654012012	666010101	21/09/06 11:12:13,000000000	501
22 678234234	666010101	30/09/06 08:49:55,000000000	413
23 678234234	654123321	01/10/06 13:12:11,000000000	2045
24 678234234	654789789	01/10/06 19:01:07,000000000	703
25 678234234	666789789	12/10/06 10:21:02,000000000	827
26 654345345	654012012	01/11/06 12:01:02,000000000	311
27 654345345	654012012	03/11/06 10:43:57,000000000	207
28 654345345	654234234	03/11/06 17:31:09,000000000	421
29 654345345	654123321	16/10/06 19:07:10,000000000	38
30 654345345	654123321	21/10/06 21:05:13,000000000	257

SQL Query (a):

```
select count(*) as NUM_LLAM_SIMULT_ORIG
from MF.LLAMADA
where (tf_origen = '654345345' and fecha_hora = '21/10/06 21:05:13') or
(tf_destino = '654345345' and fecha_hora = '21/10/06 21:05:13');
```

SQL Query (b):

```
select count(*) as NUM_LLAM_SIMULT_ORIG
from MF.LLAMADA
where (tf_origen = '678234234' and fecha_hora = '21/10/06 21:05:13') or
(tf_destino = '678234234' and fecha_hora = '21/10/06 21:05:13');
```




Implementación del trigger



Definimos la estructura general de nuestro disparador:

Disparador de Tabla- el trigger se activará una vez para cada fila afectada por la orden.

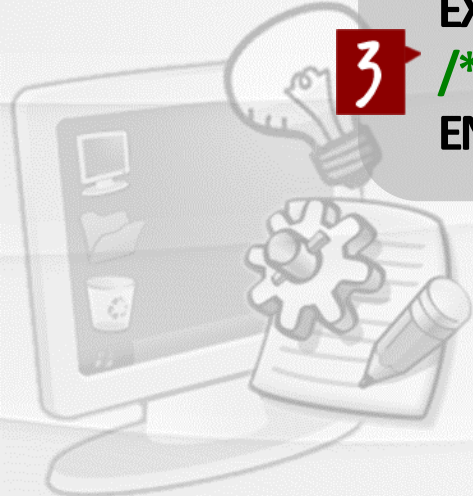
```
CREATE or REPLACE  
TRIGGER llamada_simultanea BEFORE INSERT ON LLAMADA  
FOR EACH ROW
```

Diagram annotations:
- A blue arrow points down to **BEFORE** with the label *Temporización*.
- A red label **EVENTO DE DISPARO** is above **INSERT**.
- A red label **TABLA** is above **LLAMADA**.
- An upward arrow points from *CABECERA DEL TRIGGER* to the **LLAMADA** box.

DECLARE

- 1 /* Declaraciones locales */
BEGIN
- 2 /* Sentencias Bloque PL/SQL */
EXCEPTION
- 3 /* Tratamiento de Excepciones */
END;

CABECERA DEL TRIGGER





2

- ▷ Observando la estructura de la BD y los datos que hemos de manejar en nuestro trigger, se deduce que éste se deberá disparar cuando, al tratar de insertar un nuevo registro en la tabla LLAMADA, relativo a una llamada realizada por un determinado *teléfono origen* a un determinado *teléfono destino* en una *fecha* determinada, Si:
 - ↳ existe ya registrada en la base de datos una llamada en la que dicho *teléfono origen* forma parte de otra llamada realizada en dicha *fecha*, en la que participa como *teléfono origen* o *teléfono destino* de la llamada,
 - ↳ existe ya registrada en la base de datos una llamada en la que dicho *teléfono destino* forma parte de otra llamada registrada en dicha *fecha*, en la que participa como *teléfono origen* o *destino* de la llamada.
- ▷ Para determinar si se da alguna de dichas situaciones, utilizaremos un CURSOR IMPLÍCITO para cada caso, mediante los que recuperaremos , por un lado, el número de registros de la tabla LLAMADA en los que el *teléfono origen de la llamada que desea añadirse* forma parte de otra llamada existente realizada en dicha fecha y, por otro, el número de registros en los que el *teléfono destino* forma parte de otra llamada realizada en dicha fecha previamente registrada.
- ▷ En cada caso, tendremos que analizar si el valor obtenido por sendos cursores es mayor que 0, lo cual indicará que ya existe en la tabla un registro correspondiente a una llamada realizada en la misma fecha que la que deseamos registrar, en el que participa alguno de los teléfonos como teléfono origen o destino. Si se produce esta situación tendremos que disparar, para cada caso, un error definido por el usuario (RAISE_APPLICATION_ERROR) que mostrará un mensaje informativo de tal circunstancia.



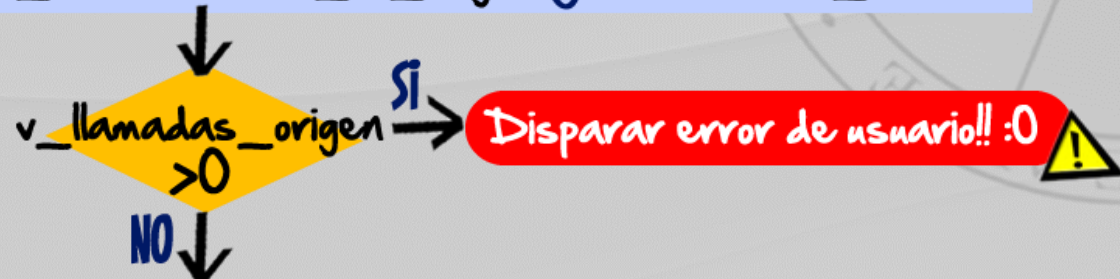
2

nuevo_tf_origen	nuevo_tf_destino	nueva_fecha	
TF_ORIGEN	TF_DESTINO	FECHA_HORA	DURACIÓN
654345345	678234234	21/10/06 21:05:13	123

MF.LLAMADA Añadir la tupla a la tabla

a

$v_llamadas_origen = num_llamadas$ en las que
 $(tf_origen = nuevo_tf_origen \text{ y } fecha = nueva_fecha) = 0$
 $(tf_destino = nuevo_tf_origen \text{ y } fecha = nueva_fecha)$



b

$v_llamadas_destino = num_llamadas$ en las que
 $(tf_origen = nuevo_tf_destino \text{ y } fecha = nueva_fecha) = 0$
 $(tf_destino = nuevo_tf_destino \text{ y } fecha = nueva_fecha)$





Definición final del disparador:

CABECERA DEL TRIGGER



```
CREATE or REPLACE
TRIGGER llamada_simultanea BEFORE INSERT ON LLAMADA
FOR EACH ROW
```

DECLARE

1 **/* Declaraciones locales */**

```
v_llamadas_orig INTEGER;
v_llamadas_dest INTEGER;
```

Variables en las que almacenaremos los valores obtenidos por los COUNTs de los cursores implícitos

BEGIN

2 **/* Sentencias Bloque PL/SQL */**

```
SELECT COUNT(*) INTO v_llamadas_orig FROM LLAMADA
WHERE (tf_origen = :new.tf_origen AND fecha_hora = :new.fecha_hora) OR
(tf_destino = :new.tf_origen AND fecha_hora = :new.fecha_hora);
```

PSEUDOREGISTROS :NEW y :OLD

:NEW contiene los nuevos valores a insertar

1 Consulta que contabiliza el número de llamadas realizadas en la fecha especificada por el teléfono origen de la llamada a insertar

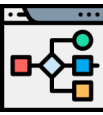
```
IF (v_llamadas_orig > 0) THEN
```

```
RAISE_APPLICATION_ERROR(-20001, 'El teléfono origen ' || :new.tf_origen || ' está
realizando o recibiendo una llamada a las ' || :new.fecha_hora);
```

```
END IF;
```

1 Si el teléfono origen ya participa en una llamada registrada en la BD a la misma hora, dispararíamos un error definido por el usuario indicando tal circunstancia





❗ Consulta que contabiliza el número de llamadas realizadas en la fecha especificada por el teléfono destino de la llamada a insertar

2

/* Sentencias Bloque PL/SQL */

```
SELECT COUNT(*) INTO v_llamadas_dest FROM LLAMADA  
WHERE (tf_origen = :new.tf_destino AND fecha_hora = :new.fecha_hora) OR  
      (tf_destino = :new.tf_destino AND fecha_hora = :new.fecha_hora);
```

```
IF (v_llamadas_dest > 0) THEN
```

```
  RAISE_APPLICATION_ERROR(-20001, 'El teléfono destino ' || :new.tf_destino || ' está  
  realizando o recibiendo una llamada a las ' || :new.fecha_hora);
```

```
END IF;
```

```
END;
```

❗ Si el teléfono destino ya participa en una llamada registrada en la BD a la misma hora, dispararíamos un error definido por el usuario indicando tal circunstancia



⚠ El disparador debe ser “BEFORE INSERT/FOR EACH ROW”, ya que *éste es el único caso en el que NO se produciría un fallo de TABLA MUTANTE que se da cuando se accede a la tabla de disparo en el cuerpo del disparador.*

⚠ Con BEFORE INSERT or UPDATE, se producirá un *error de tabla mutante* complicado de encontrar ya que no se genera exactamente el mensaje descriptivo que en principio cabría esperar.



EJECUTAR UN INSERT EN LA TABLA LLAMADA:



Oracle SQL Developer : C:\Users\Marc\AppData\Roaming\SQL Developer\Sin Título5.sql

Archivo Editar Herramientas Ventana Ayuda

Ejecutamos el script

Conexiones

- Procedimientos
- Funciones
- Operadores
- Colas
- Tablas de Colas
- Disparadores
- LLAMADA_SIMULTANEA**
- Tos

Disparador registrado

Hacemos la inserción en la tabla LLAMADA que dispara la ejecución de nuestro trigger

```

1 SET SERVEROUTPUT ON;
2 insert into llamada values ('654345345', '678234234', '21/10/06 21:05:13', 123);
    
```

Salida de Script x

Tarea terminada en 0,642 segundos

Error que empieza en la línea: 2 del comando :

```

insert into llamada values ('654345345', '678234234', '21/10/06 21:05:13', 123)
    
```

Informe de error -

```

ORA-20001: El teléfono origen 654345345 está realizando o recibiendo una llamada a las 21/10/06 21:05:13,000000
ORA-06512: en "PROF2.LLAMADA_SIMULTANEA", línea 11
ORA-04088: error durante la ejecución del disparador 'PROF2.LLAMADA_SIMULTANEA'
    
```

Se genera un error desde el disparador al tratar de insertar el registro de llamada, ya que el teléfono origen ya participa en una llamada registrada en la BD a la misma hora,