



Instituto Politécnico Nacional  
Escuela Superior de Cómputo



# Bases de Datos

## Práctica no. 6: Procedures

Profesor: Euler Hernández Contreras

Alumno: Calva Hernández José Manuel

Grupo: 2CM12

## Índice

Índice.....	2
Marco Teórico .....	3
Instrucciones .....	5
Screenshots .....	9
Conclusiones .....	15
Referencias.....	15

## Marco Teórico

La mayoría de los sistemas de gestión de bases de datos incluyen una extensión del mismo SQL, llamada módulos de almacenamiento permanente (PSM), para permitir a los usuarios escribir procedimientos almacenados, llamados rutinas internas, dentro del espacio de proceso de la base de datos, en lugar de externamente. Es lo que históricamente se conoce como procedimientos almacenados de bases de datos, aunque pueden ser funciones o procedimientos. El término utilizado en el estándar SQL para los procedimientos almacenados es módulos almacenados persistentes, porque el DBMS almacena persistentemente estos programas, algo parecido a los datos persistentes almacenados por el DBMS.

Los procedimientos almacenados son útiles en las siguientes circunstancias:

- Si varias aplicaciones necesitan un mismo programa de bases de base de datos, este último se puede almacenar en el servidor e invocarlo desde esas aplicaciones. Esto reduce la duplicidad del esfuerzo y mejora la modularidad del software.
- La ejecución de un programa en el servidor puede reducir el coste derivado de la transferencia y la comunicación de datos entre el cliente y el servidor en ciertas situaciones.
- Estos procedimientos pueden mejorar la potencia de modelado de las vistas al permitir que los usuarios de bases de datos cuenten con tipos más complejos de datos derivados. Además, se pueden utilizar esos tipos para comprobar restricciones más complejas que quedan fuera de la especificación de aserciones y triggers.

En general, muchos DBMSs comerciales permiten escribir procedimientos y funciones almacenados en un lenguaje de programación de propósito general. De forma alternativa, un procedimiento almacenado puede estar compuesto por comandos SQL sencillos, como recuperaciones y actualizaciones.

En PL/SQL, un procedimiento es un bloque con nombre que tiene un conjunto opcional de parámetros. Cada parámetro contiene un nombre de parámetro, un uso (IN, OUT, IN OUT) y un tipo de dato. Un parámetro de entrada (IN) no debe cambiar dentro de un procedimiento. A un parámetro de salida (OUT) se le da un valor dentro de un procedimiento. Un parámetro de entrada-salida (IN OUT) debe tener un valor proporcionado fuera del procedimiento, pero puede cambiar dentro de éste. La especificación del tipo de datos no debe incluir ninguna limitación, como longitud. Por ejemplo, para un parámetro de hilera debe usar el tipo de dato VARCHAR2. No tiene que proporcionar la longitud en la especificación del tipo de dato para un parámetro.

Procedure Structure:

```
CREATE [OR REPLACE] PROCEDURE ProcedureName
    [ ( Parameter1, . . . , ParameterN ) ]
IS
    [ sequence of declarations ]
BEGIN
    sequence of statements
```

```
[ EXCEPTION
sequence of statements to respond to exceptions ]
END;
```

Como el DBMS almacena persistentemente los procedimientos y las funciones, debe ser posible llamarlos desde varias interfaces SQL y técnicas de programación. Se puede utilizar la sentencia CALL del estándar SQL para invocar un procedimiento almacenado (desde una interfaz interactiva o desde SQL incrustado o SQLJ). El formato de la sentencia es el siguiente:

CALL <nombre del procedimiento o función> «lista de argumentos» ;

Si esta sentencia es llamada desde JDBC, debe asignarse a un objeto de sentencia de tipo *CallableStatement*.

Lista de excepciones comunes predefinidas en PL/SQL

Excepción	Cuando surge
Cursor_Already_Open	Trata de abrir un cursor abierto anteriormente
Dup_Val_On_Index	Trata de guardar un valor duplicado en un índice único
Invalid_Cursor	Trata de realizar una operación inválida en un cursor, como cerrar un cursor que no se abrió anteriormente
No_Data_Found	La sentencia SELECT INTO no regresa ninguna fila
Rowtype_Mismatch	Trata de asignar valores con tipos de datos incompatibles entre un cursor y una variable
Timeout_On_Resource_exclusivo1	Ocurre un tiempo fuera como cuando esperamos un candado
Too_Many_Rows	La sentencia SELECT INTO regresa más de una fila

## Instrucciones

### 1.- Cargar script tt.sql

```
create database tt1;
use tt1;
source ...
```

### 2.- Creación de las siguientes vistas...

a) Nombre del tt y su número de aquellos tts donde ha sido sinodal fabiola ocampo.

```
create view v1 as
Select t.*
from tt t, sinodalia s, profesor p
where t.nott=s.nott
and (s.s1=p.idprof or s.s2=p.idprof or s.s3=p.idprof)
and p.nombre like "Fabiola%"
and p.apPaterno like "Ocampo%"
order by t.titulo, t.nott;
```

b) Mostrar las calificaciones de los tts remediales.

```
create view v2 as
Select califRevisor, califSinodales, nott
from presentacion
where tipo like "TT R%"
order by nott;
```

c) Mostrar la fecha de presentación y el tipo de tt que han sido dirigidos por el profesor Arturo Garfias.

```
create view v3 as
Select x.nott, x.fecha, x.tipo
from presentacion x, tt t, dirige d, profesor p
where x.nott=t.nott
and t.nott=d.nott
and d.idProf=p.idProf
and p.nombre like "Flavi%"
and p.apPaterno like "S_nch%"
order by 1;
```

d) Mostrar el nombre de los profesores que han hecho un doctorado.

```
create view v4 as
Select p.nombre, p.apPaterno, p.apMaterno
from profesor p, gradoProf gp, gradoEstudios ge
```

```
where p.idProf=gp.idProf
and gp.idGrado=ge.idGrado
and ge.descripcion like "doctor%"
order by 2,3;
```

e) Mostrar los tts que se han presentado en el año 2007.

```
create view v5 as
Select t.*, p.fecha
from tt t, presentacion p
where t.nott=p.nott
and fecha between "2007-01-01" and "2007-12-31"
order by nott;
```

f) Mostrar el nombre de los tts que ha sido sinodal la profesora Martha Rosa Cordero.

```
create view v6 as
Select t.*
from tt t, sinodalia s, profesor p
where t.nott=s.nott
and (s.s1=p.idprof or s.s2=p.idprof or s.s3=p.idprof)
and p.nombre like "Martha%"
and p.apPaterno like "Cordero%"
order by t.titulo, t.nott;
```

g) Cuántos tts se presentaron en el año 2008.

```
create view v7 as
Select count(*)
from presentacion
where fecha between "2008-01-01" and "2008-12-31";
```

h) Mostrar el dictamen de los tts que han sido revisados por la profesora Idalia Maldonado

```
create view v8 as
Select x.nott, x.dictamen
from presentacion x, tt t, sinodalia s, profesor p
where x.nott=t.nott
and t.nott=s.nott
and s.revisor=p.idprof
and p.nombre like "Idal%"
and p.appaterno like "Maldon%"
order by 2,1;
```

i) Mostrar el dictamen de los tts que han sido revisados por el profesor Ulises Vélez

```
create view v9 as
Select x.nott, x.dictamen
```

```

from presentacion x, tt t, sinodalia s, profesor p
where x.nott=t.nott
and t.nott=s.nott
and s.revisor=p.idprof
and p.nombre like "Ulise%"
and p.appaterno like "V_le%"
order by 1,2;

```

### 3.- Primera función (procedure)

a) Conocer el número de registros donde su profe de bd ha dirigido tts.

b) Conocer el nott y la fecha de presentación de la consulta anterior

```

delimiter &
create procedure p1()
begin
    Select count(*)
    from dirige d, profesor p
    where d.idProf=p.idProf
    and p.nombre like "Euler%"
    and p.appaterno like "Hern%";

    Select x.nott, x.fecha
    from presentacion x, tt t, dirige d, profesor p
    where x.nott=t.nott
    and t.nott=d.nott
    and d.idProf=p.idProf
    and p.nombre like "Euler%"
    and p.appaterno like "Hern%"
    order by 1, 2;
end &

```

### 4.- Segunda función (procedure)

a) Los tts que tienen en su título el nombre de redes de computadoras

b) Conocer el nombre de sus directores de la consulta anterior

```

delimiter &
create procedure p2()
begin
    Select *
    from tt
    where titulo like "%redes de computadoras%"
    order by titulo, nott;

    Select t.*, p.nombre, p.apPaterno, p.apMaterno

```

```
from tt t, dirige d, profesor p
where t.nott=d.nott
and d.idprof=p.idprof
and t.titulo like "%redes de computadoras%"
order by p.appaterno, p.apmaterno;
end &
```



## Screenshots

```
mysql> create database tt;
ERROR 1007 (HY000): Can't create database 'tt'; database exists
mysql> create database tt1;
Query OK, 1 row affected (0.00 sec)

mysql> use tt1;
Database changed
mysql> source C:\Users\manue\Documents\Escuela\3er Semestre\Bases de Datos\Scripts\tt.sql
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create view v1 as
-> Select t.*, p.nombre as NOMBRE
-> from tt t, sinodalia s, profesor p
-> where t.nottes.nott
-> and (s.s1=p.idprof or s.s2=p.idprof or s.s3=p.idprof)
-> and p.nombre like "Fabiola%"
-> and p.apPaterno like "Ocampo%"
-> order by t.titulo, t.nott;
Query OK, 0 rows affected (0.03 sec)

mysql> desc v1;
+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+
| noTT  | varchar(50)   | NO   |     | NULL    |       |
| titulo | varchar(300) | YES  |     | NULL    |       |
| NOMBRE | varchar(50)   | YES  |     | NULL    |       |
+-----+
3 rows in set (0.01 sec)

mysql> Select * from v1;
+-----+-----+-----+
| noTT | titulo                                                                 | NOMBRE |
+-----+-----+-----+
| 2010-0110 | Adaptaci3n de la arquitectura SOA al procedimiento 'Otorgamiento De Becas' | Fabiola |
| 2007-0102 | Aplicaci3n de simulaci3n a interpolaci3n en f3sica 'henoc' | Fabiola |
| 1996-0004 | Aracnotr3fn: Robot Experimental En Forma De Arana | Fabiola |
| 2008-0088 | Codificaci3n y decodificaci3n de informaci3n mediante reestructores exactos de estado implantados en microcontroladores | Fabiola |
+-----+-----+-----+
```

```
mysql> create view v2 as
-> Select califRevisor, califSinodales, nott
-> from presentacion
-> where tipo like "TT R%"
-> order by nott;
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> select * from v2;
```

califRevisor	califSinodales	nott
8	10	01-1-0031
9	10	01-1-0040
8	9	01-1-0041
10	10	01-1-0042
9	9	01-1-0043
8	10	01-1-0044
9	8	01-2-0047
10	10	01-2-0048
8	10	01-2-0049
10	10	04-1-0102
8	8	04-1-0104

```
mysql> create view v4 as
-> Select p.nombre, p.apPaterno, p.apMaterno
-> from profesor p, gradoProf gp, gradoEstudios ge
-> where p.idProf=gp.idProf
-> and gp.idGrado=ge.idGrado
-> and ge.descripcion like "doctor%"
-> order by 2,3;
```

Query OK, 0 rows affected (0.02 sec)

```
mysql> select * from v4;
```

nombre	apPaterno	apMaterno
Fernando	Aguilar	S nchez
Mario	Aldape	P,rez
Aurora	Aparicio	Castillo
Richard Alexander	Award	Reyes
Alejandro	Botello	Castillo
Juan Antonio	Castillo	Marrufo
Edgar Armando	Catal n	Salgado
Rolando F,lix	Cedillo	Caballero
Juan Carlos	Chimal	Eguja
Eduardo	Ch vez	Lima
Luis Carlos	Coronado	Garcja
Luis Carlos	Coronado	Garcja
David	Cruz	Rojas
Alma Delia	Cuevas	Rasgado

```
mysql> create view v5 as
-> Select t.*, p.fecha
-> from tt t, presentacion p
-> where t.nott=p.nott
-> and fecha between "2007-01-01" and "2007-12-31"
-> order by nott;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from v5;
```

noTT	titulo	fecha
07-1-0181	Prototipo para la Verificación de Usuarios y Estadísticas de Correo Institucional para la DCYC	2007-11-29
07-1-0183	Herramienta Web de Gestión Colaborativa de Reportes Técnicos	2007-11-29
07-1-0186	Visita Virtual al Proyecto Puerta Alameda	2007-11-29
07-1-0187	Sistema de Seguridad Para Redes Privadas	2007-11-30
07-1-0188	Sistema de Administración y Consulta para la Dirección General de Educación Tecnológica Agropecuaria (DGETA)	2007-11-30
07-1-0189	Sistema de Gestión Comercial	2007-11-30
2007-0001	Robots cooperativos	2007-05-16
2007-0002	Pronóstico poblacional y desarrollo urbano sustentable	2007-05-16
2007-0003	NEPTUNO 1: VEHICULO OPERADO A DISTANCIA PARA LA ADQUISICIÓN DE IMÁGENES Y SOFTWARE DE TRATAMIENTO DIGITAL	2007-05-16
2007-0004	Generación de imágenes tridimensionales holográficas mediante computadora	2007-05-17
2007-0005	Prototipo de un sistema de administración de voip en una red de área local	2007-05-17
2007-0006	Sistema de cifrado de video que usa codificación mpeg	2007-05-17

```
mysql> create view v6 as
-> Select t.*
-> from tt t, sinodalia s, profesor p
-> where t.nott=s.nott
-> and (s.s1=p.idprof or s.s2=p.idprof or s.s3=p.idprof)
-> and p.nombre like "Martha%"
-> and p.apPaterno like "Cordero%"
-> order by t.titulo, t.nott;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from v6;
```

noTT	titulo
2003-0611	Administrador De Un Servidor Linux Vía Web (adssel)
2004-0671	Analizador De Consultas Para Obtener La Estructura Del Cubo Que Contenga La Información Requerida
2000-0240	Animación Por Computadora Aplicada A Un Videojuego De Aventura En 3d
2008-0006	Byte-thoven SAEF ( sistema de apoyo a la enseñanza de piano)
2010-0146	CHIAPAS PALANQUE AL 3D BAJULAM BAK
2006-0142	comercio electrónico empleando palmas
2004-0705	Desarrollo De Herramientas Para El Análisis De Objetos Arqueológicos Y Análisis De Desplazamiento De Culturas Dhaa-adc
09-1-0237	Digycert: sistema generador de certificados digitales a través de un dispositivo móvil
2008-0163	Diseño y Desarrollo de un Módulo Transmisor-Receptor de RFID
11-1-0002	DISEÑO DE UNA ARQUITECTURA PARA UNA RED NEURONAL ARTIFICIAL PERCEPTRON SOBRE UN FPGA

```
mysql> create view v7 as
-> Select count(*)
-> from presentacion
-> where fecha between "2008-01-01" and "2008-12-31";
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from v7;
```

count(*)
33

```
1 row in set (0.00 sec)
```

```
mysql> create view v8 as
-> Select x.nott, x.dictamen
-> from presentacion x, tt t, sinodalia s, profesor p
-> where x.nott=t.nott
-> and t.nott=s.nott
-> and s.revisor=p.idprof
-> and p.nombre like "Idalia%"
-> and p.appaterno like "Maldona%"
-> order by 2,1;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from v8;
Empty set (0.00 sec)
```

```
mysql> create view v9 as
-> Select x.nott, x.dictamen
-> from presentacion x, tt t, sinodalia s, profesor p
-> where x.nott=t.nott
-> and t.nott=s.nott
-> and s.revisor=p.idprof
-> and p.nombre like "Ulise%"
-> and p.appaterno like "V_le%"
-> order by 1,2;
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select * from v9;
```

nott	dictamen
08-1-0217	aprobado
09-1-0244	aprobado
10-1-0287	aprobado
11-1-0010	aprobado
11-1-0027	aprobado
1997-0034	aprobado
1999-0138	aprobado
2000-0216	reprobado
2001-0304	aprobado
2002-0395	aprobado
2003-0498	aprobado
2003-0612	aprobado
2004-0710	aprobado
2005-0890	aprobado
2007-0004	aprobado
2007-0152	aprobado
2008-0070	aprobado
2008-0164	aprobado
2009-0035	aprobado
2010-0005	aprobado
2010-0053	aprobado
2010-0149	aprobado

```
22 rows in set (0.00 sec)
```

```

mysql> delimiter &
mysql> create procedure p1()
  -> begin
  -> Select count(*)
  -> from dirige d, profesor p
  -> where d.idProf=p.idProf
  -> and p.nombre like "Euler%"
  -> and p.appaterno like "Hern%";
  ->
  -> Select x.nott, x.fecha
  -> from presentacion x, tt t, dirige d, profesor p
  -> where x.nott=t.nott
  -> and t.nott=d.nott
  -> and d.idProf=p.idProf
  -> and p.nombre like "Euler%"
  -> and p.appaterno like "Hern%"
  -> order by 1, 2;
  -> end &
Query OK, 0 rows affected (0.05 sec)

mysql> delimiter ;
mysql> call p1;
+-----+
| count(*) |
+-----+
|      17 |
+-----+
1 row in set (0.02 sec)

+-----+-----+
| nott    | fecha    |
+-----+-----+
| 2007-0153 | 2007-11-28 |
| 2008-0160 | 2008-11-29 |
| 2010-0018 | 2010-12-04 |
+-----+-----+
3 rows in set (0.02 sec)

Query OK, 0 rows affected (0.03 sec)

```

```
mysql> delimiter &
mysql> create procedure p2()
-> begin
-> Select *
-> from tt
-> where titulo like "%redes de computadoras%"
-> order by titulo, nott;
->
-> Select t.*, p.nombre, p.apPaterno, p.apMaterno
-> from tt t, dirige d, profesor p
-> where t.nott=d.nott
-> and d.idprof=p.idprof
-> and t.titulo like "%redes de computadoras%"
-> order by p.apPaterno, p.apMaterno;
-> end &
```

Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter ;
mysql> call p2;
```

noTT	titulo
2010-0041	Agentes Monitores para Redes de Computadoras

1 row in set (0.00 sec)

noTT	titulo	nombre	apPaterno	apMaterno
2010-0041	Agentes Monitores para Redes de Computadoras	Carlos	Pineda	Guerrero
2010-0041	Agentes Monitores para Redes de Computadoras	Gilberto	S nchez	Quintanilla

2 rows in set (0.01 sec)

Query OK, 0 rows affected (0.05 sec)

## Conclusiones

La práctica resultó como un modo de conclusión y refuerzo para la parte de vistas, ya que volvimos a aplicar vistas y hacer consultas sobre ellas, nuevamente, sin mayor complicación.

Por otra parte, iniciamos a definir procedures para nuestra base de datos, estos nos permiten realizar múltiples consultas al mismo tiempo, aunque de la teoría aprendí que es posible realizar muchas más tareas complejas sobre la base de datos, sin embargo, dado que esto fue únicamente una introducción, la mayor complicación fue aplicar correctamente la sintaxis del procedure para que nos marque un error al momento de ingresarlo en la terminal de MySQL. Esto se debe a que la sintaxis de las consultas ya la tenemos bien aprendida, por lo que lo único nuevo fue como incluir varias de ellas dentro del procedure.

## Referencias

- Ramez, E., & Navathe, S. (2000). *Sistemas de Bases de Datos: Conceptos Fundamentales* (1st ed.). México: Pearson Educación.
- Ricardo, C., Campos Olguín, V., & Enríquez Brito, J. (2010). *Bases de datos*. México: McGraw-Hill Interamericana.
- Mannino, M. (2007). *Administración de Bases de Datos. Diseño y desarrollo de aplicaciones* (3rd ed.). México: McGraw-Hill Interamericana.