Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bases de Datos

Práctica no. 8:

Bloqueos

Profesor: Euler Hernández Contreras

Alumno: Calva Hernández José Manuel

Grupo: 2CM12

## Índice

[Índice 2](#_Toc500523685)

[Marco Teórico 3](#_Toc500523686)

[Instrucciones 5](#_Toc500523687)

[Screenshots 10](#_Toc500523688)

[Conclusiones 20](#_Toc500523689)

[Referencias 20](#_Toc500523690)

## Marco Teórico

Las técnicas de control de la concurrencia se utilizan para garantizar la ausencia de interferencias o la propiedad de aislamiento de las transacciones que se ejecutan simultáneamente. La mayoría de estas técnicas garantizan la serialización de las planificaciones, utilizando protocolos (conjuntos de reglas) que garantizan esa serialización. Un importante conjunto de protocolos emplea la técnica del bloqueo de los elementos de datos para evitar que varias transacciones accedan concurrentemente a los elementos

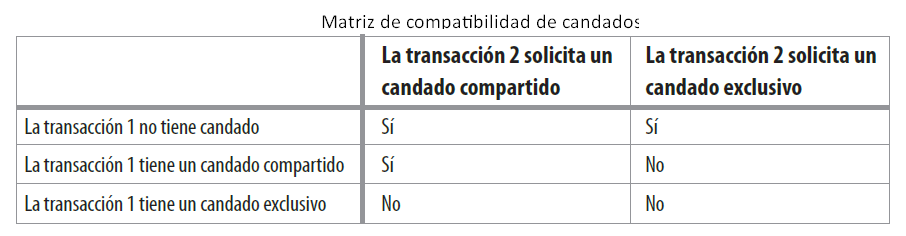
Un bloqueo, también llamado candado, es una variable asociada a un elemento de datos que describe el estado de ese elemento respecto a las posibles operaciones que se le puedan aplicar. Generalmente, hay un bloqueo por cada elemento de datos de la base de datos. Los bloqueos se utilizan como un medio para sincronizar el acceso de las transacciones concurrentes a los elementos de la base de datos.

**Tipos de bloqueos**

Bloqueo binario: Un bloqueo binario puede tener dos estados o valores: bloqueado y desbloqueado (o 1 y O, para simplificar). Cada elemento X de la base de datos tiene un bloqueo distinto. Si el valor del bloqueo sobre X es 1, el elemento X no podrá ser accedido por una operación de base de datos que solicite el elemento. Si el valor del bloqueo sobre X es O, es posible acceder al elemento cuando es solicitado. Nos referiremos al valor (o estado) actual del bloqueo asociado con el elemento X como bloquear(X).

Bloqueos compartidos/exclusivos (o lectura/escritura): Debemos permitir que varias transacciones tengan acceso al mismo elemento X si todas ellas acceden a X sólo para leer. Sin embargo, si una transacción va a escribir un elemento X, debe tener acceso exclusivo a X. Con este fin, se utiliza un tipo de bloqueo diferente denominado bloqueo de modo múltiple. En este esquema (denominado bloqueos compartido/exclusivo o de lectura/escritura) hay tres operaciones de bloqueo: bloquearJectura(X), bloquear\_escritura(X) y desbloquear(X). Un bloqueo asociado con un elemento X, BLOQUEAR(X), tiene ahora tres posibles estados: bloqueado para lectura, bloqueado para escritura o desbloqueado. Un elemento bloqueado para lectura también se denomina de lectura compartida porque otras transacciones pueden leer el elemento, mientras que un elemento bloqueado para escritura se denomina de escritura exclusiva porque una sola transacción posee en exclusiva el bloqueo de un elemento.

La siguiente matriz explica el funcionamiento de los bloqueos compartidos/exclusivos.



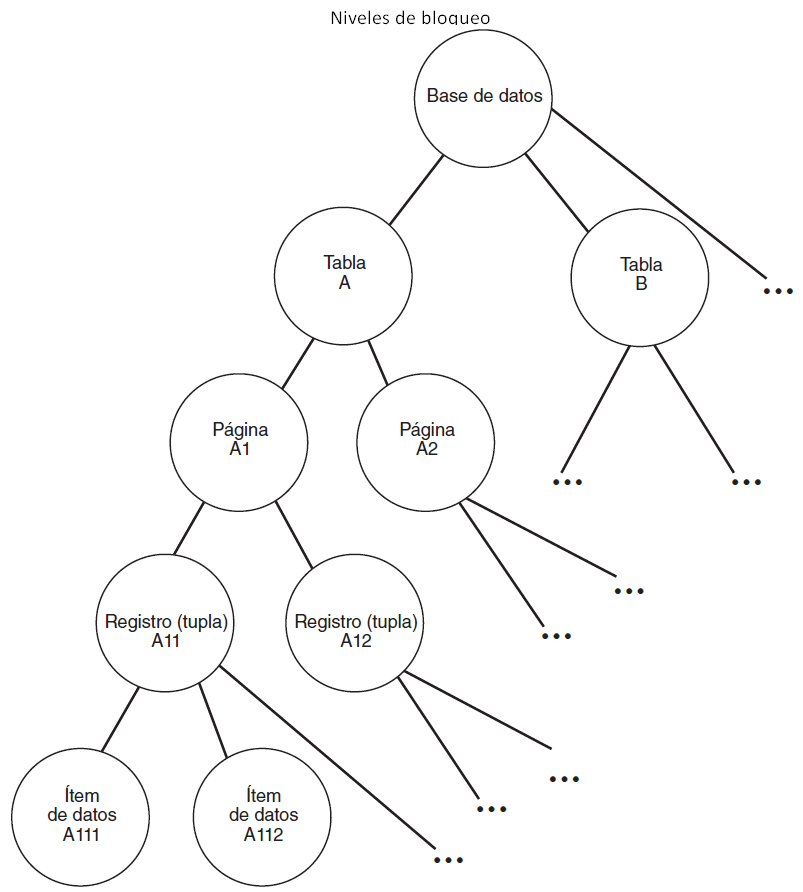
**Bloqueo de dos fases**

Un esquema de bloqueo que garantiza la seriabilidad es el protocolo de bloqueo de dos fases. De acuerdo con las reglas de este protocolo, cada transacción se desarrolla en dos fases: primero una fase de crecimiento en la que adquiere todos los bloqueos necesarios para la transacción, y luego una fase de encogimiento en la que libera sus bloqueos. No hay ningún requerimiento de que todos los bloqueos se obtengan de manera simultánea. Normalmente, la transacción adquiere algunos bloqueos, hace cierto procesamiento y adquiere bloqueos adicionales que necesite. Sin embargo, nunca libera ningún bloqueo hasta que ha alcanzado una etapa en la que no se necesitarán nuevos bloqueos. Las reglas son:

* Una transacción debe adquirir un bloqueo sobre un ítem antes de operar en éste. Para el acceso de sólo lectura basta un bloqueo compartido. Para tener acceso a la escritura se requiere un bloqueo exclusivo.
* Una vez que la transacción libera uno de los bloqueos, ya no adquiere nuevos bloqueos.

**Niveles de bloqueo**

Los candados se pueden aplicar a nivel de ítem de datos, de un registro, de un archivo, de un grupo de archivos o de toda la base de datos. Pocos sistemas implementan candados a nivel de ítem de datos debido a lo indirecto del procedimiento. Algunos bloquean páginas al mismo tiempo. La finura o granularidad de los bloqueos se expresa por medio de una estructura jerárquica en que los nodos representan objetos de datos de diferentes tamaños, como se ilustra en la siguiente figura. Aquí, el nodo raíz representa toda la base de datos, los nodos del nivel 1 indican tablas, los del nivel 2 son páginas de tablas, los del nivel 3 son registros, y los restantes representan ítem de datos. Siempre que un nodo esté bloqueado, todos sus descendientes también lo estarán.



## Instrucciones

Cargar script sams

create database sams;

use sams;

source ...

Trabajar con una operación de escritura y otra de lectura

insert into gerente (idgerente, nombre) values (302, "HERNANDEZ CONTRERAS EULERCITO");

select \* from gerente

where idgerente between 298 and 310;

Bloqueo de lectura:

t1.

lock tables gerente read;

l(gerente)

l(estado)

l(socio)

l(producto)

t2.

l(gerente)

l(estado)

l(socio)

l(producto)

Trabajar sobre proveedor

insert into proveedor values (34, "Mi negocio chido!!!", "55-55-55-55-55");

Bloqueo de escritura:

t1.

lock tables gerente write;

e(proveedor)

l(proveedor)

l(estado)

l(gerente)

t2.

l(estado)

l(gerente)

l(proveedor) ... // stand by

t1.

unlock tables;

t2.

... l(proveedor)

t1.

bloquear lectura(gerente)

bloquear escritura(proveedor)

lock tables gerente read, proveedor write;

l(gerente)

l(proveedor)

l(estado)

l(socio)

insert into proveedor values (35, "Proveedor de marihuana!!", "55-55-55-55-56");

t1.

bloquear escritura(proveedor)

bloquear lectura(socio)

lock tables proveedor write, socio read;

l(proveedor)

l(socio)

t2.

bloquear escritura(socio)

lock tables socio write;

e(socio)

l(socio)

t3.

bloquear lectura(proveedor)

lock tables proveedor read;

l(proveedor)

t1.

bloquear escritura(socio)

lock tables socio write;

------------------------------------

lock tables \_\_\_\_\_\_\_\_\_ [read/write];

unlock tables;

--------------------------------------

Crear los siguientes stored procedures...

1. Mostrar los gerentes existentes en las sucursales del Estado de México

delimiter $

create procedure sp1()

begin

select g.nombre as gerente, c.nombre as club

from gerente g, club c, estado e

where g.idclub = c.idclub

and c.idedo = e.idedo

and e.nombre like "%M\_xico%"

order by 2, 1;

end $

delimiter ;

2. Los servicios que tiene el club Morelia

delimiter $

create procedure sp2()

begin

select s.nombre

from club c, servicioclub sc, servicio s

where c.idclub = sc.idclub

and sc.idservicio = s.idservicio

and c.nombre like "%Morel%"

order by 1;

end $

delimiter ;

3. Nombre de las sucursales en que están registrados los socios que se apellidan García

delimiter $

create procedure sp3()

begin

select c.nombre

from club c, socioclub sc, socio s

where c.idclub = sc.idclub

and sc.idsocio = s.idsocio

and s.nombre like "%Garc\_a%"

order by 1;

end $

delimiter ;

4. Mostrar el nombre y precio unitario de los productos suministrados por Sabritas

delimiter $

create procedure sp4()

begin

select po.nombre, po.preciounitario

from proveedor p, producto po

where po.idproveedor = p.idproveedor

and p.nombre like "%Sabritas%"

order by 1, 2;

end $

delimiter ;

5. Mostrar el nombre del club y sus gerentes de aquellos clubes existentes en los siguientes cp (90610,64780,54080)

delimiter $

create procedure sp5()

begin

select c.nombre, g.nombre

from club c, gerente g

where c.idclub = g.idclub

and (c.direccion like "%90610%"

or c.direccion like "%64780%"

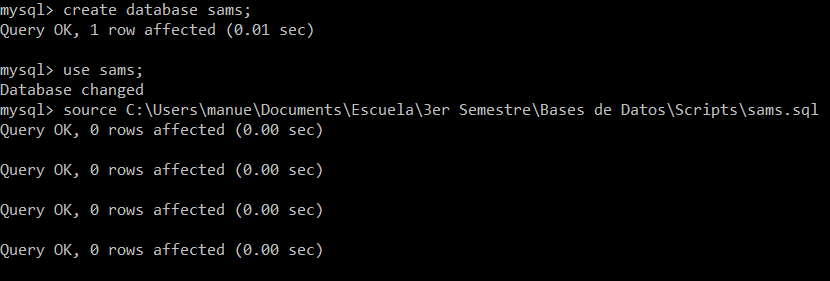
or c.direccion like "%54080%")

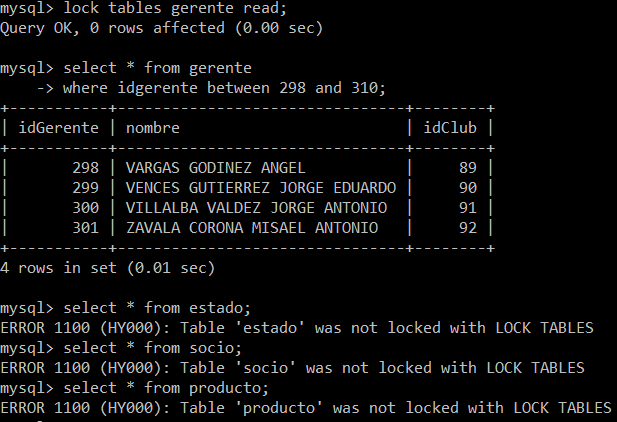
order by 1, 2;

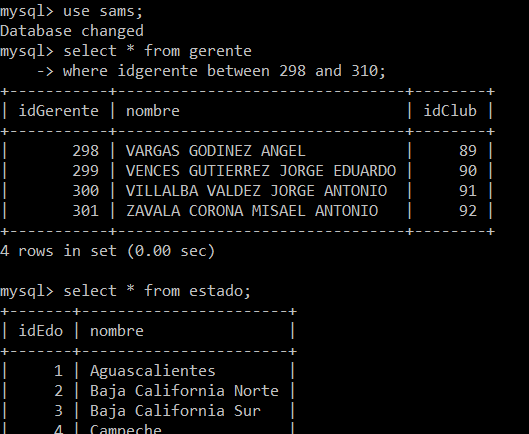
end $

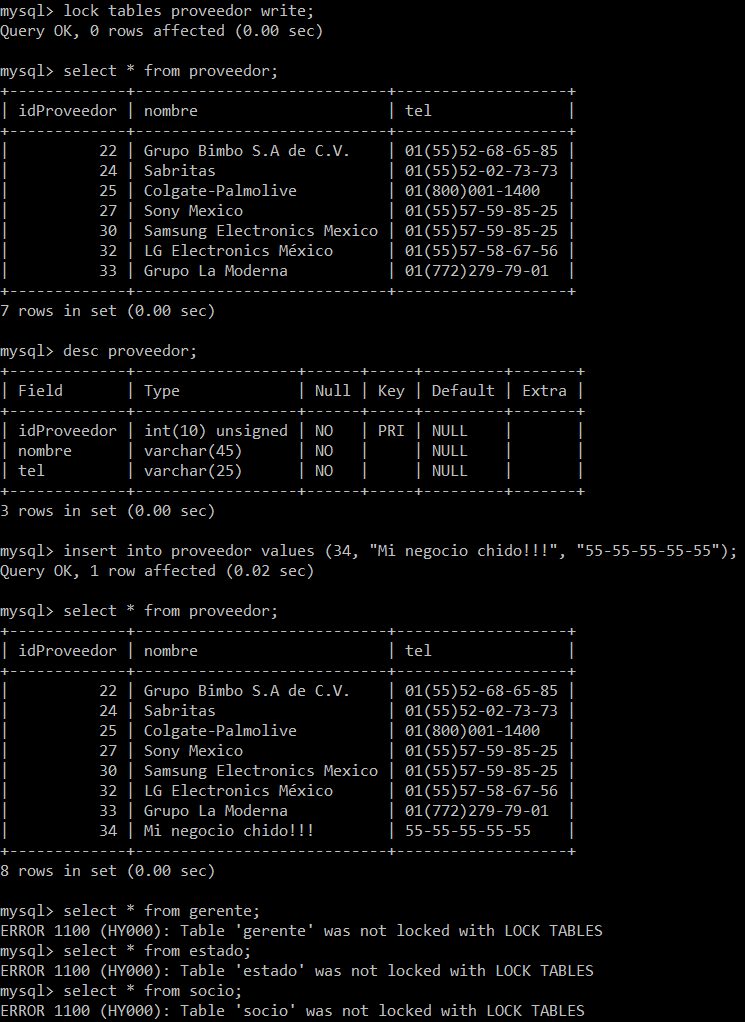
delimiter ;

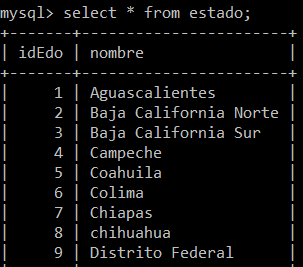
## Screenshots





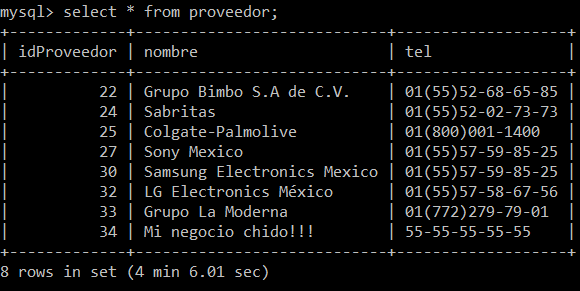


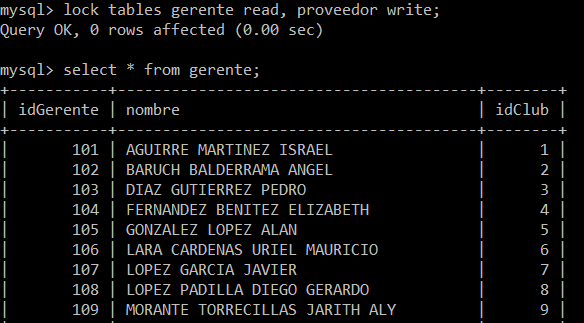


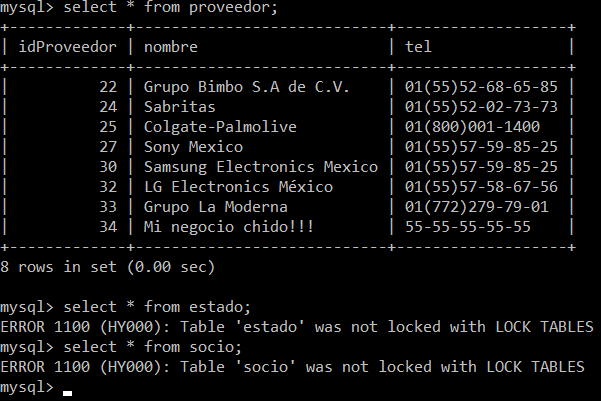


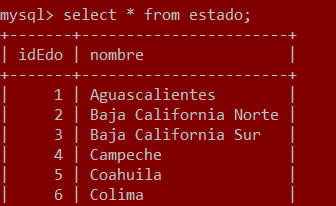


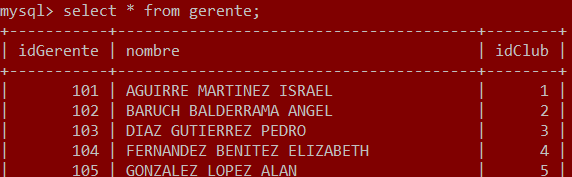


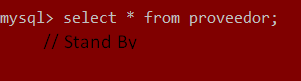


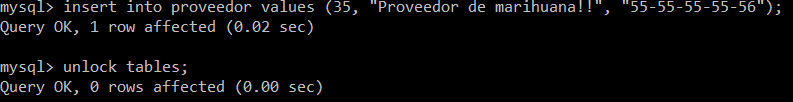


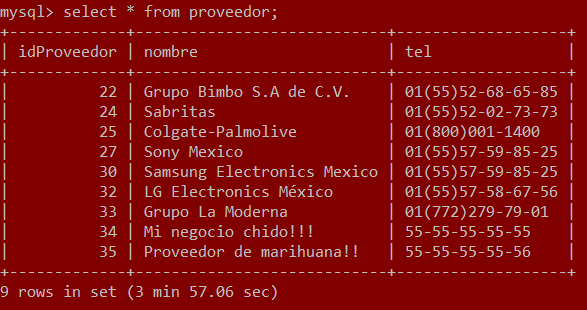


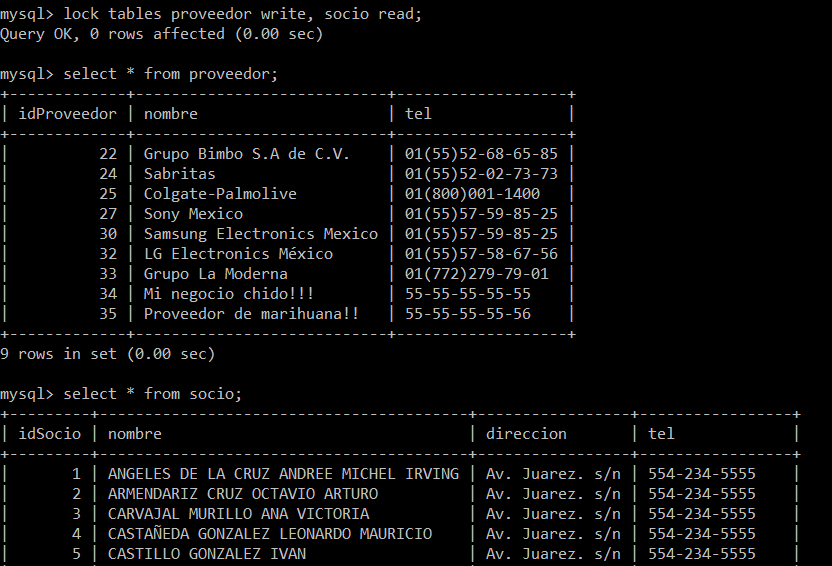












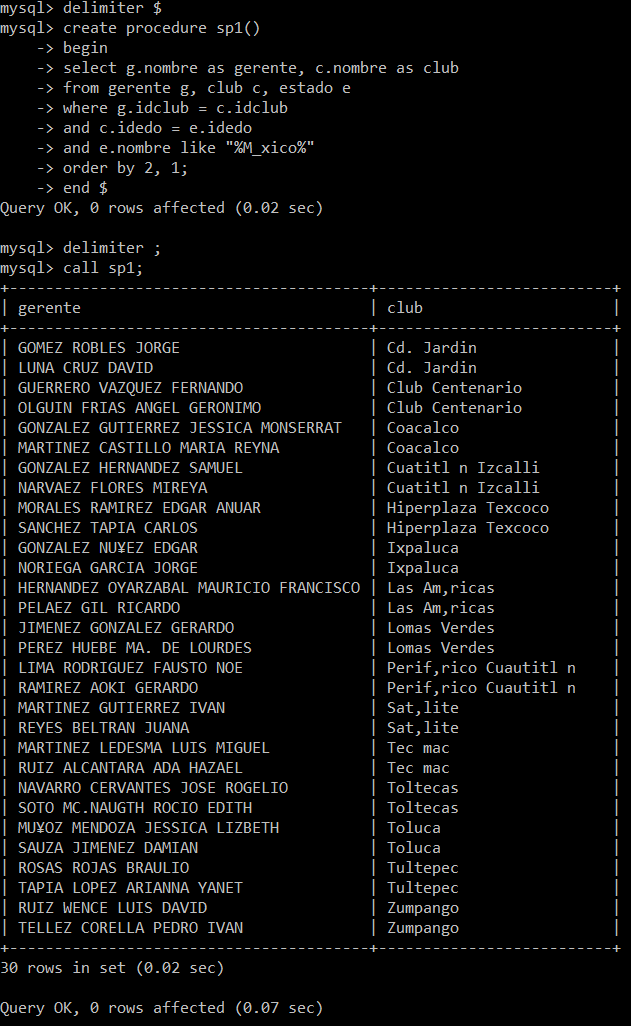




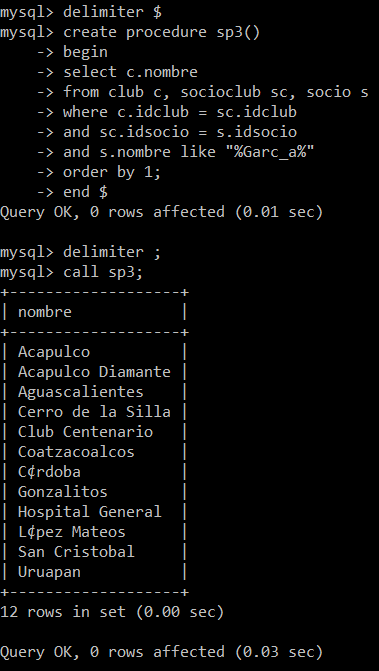


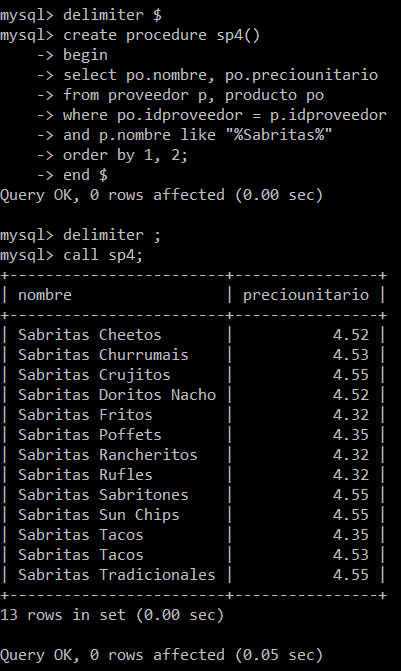














## Conclusiones

La práctica resultó interesante debido a que pudimos comprobar cómo funcionan los bloqueos en mysql, de los cuáles ya habíamos hablado de manera teórica en clase pero que pudimos comprobar funcionalmente en el laboratorio. Lo primero que hay que hacer mención, es que el manejo de bloqueos simples resulta bastante sencillo, sin embargo, el manejo de bloqueos múltiples comienza a crecer en dificultad a medida que vamos incluyendo más tablas y operaciones, ya que hay que tomar en cuenta la prioridad tanto del tipo de bloqueo como de otros bloqueos entrantes provenientes de otras terminales como pudimos constatar en la práctica.

Por otra parte, continuamos con procedures, una función que ya habíamos visto de manera superficial pero que ahora practicamos nuevamente para comprender mejor su funcionamiento, sin embargo, hace falta ahondar más en esto, debido a que la complejidad de los procedures va más allá de una simple selección.

## 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.