



ANIVERSARIO
Escuela
Superior de
Cómputo
1993-2018



Escuela Superior de Computo

Grupo: 2CV2

Bases de datos

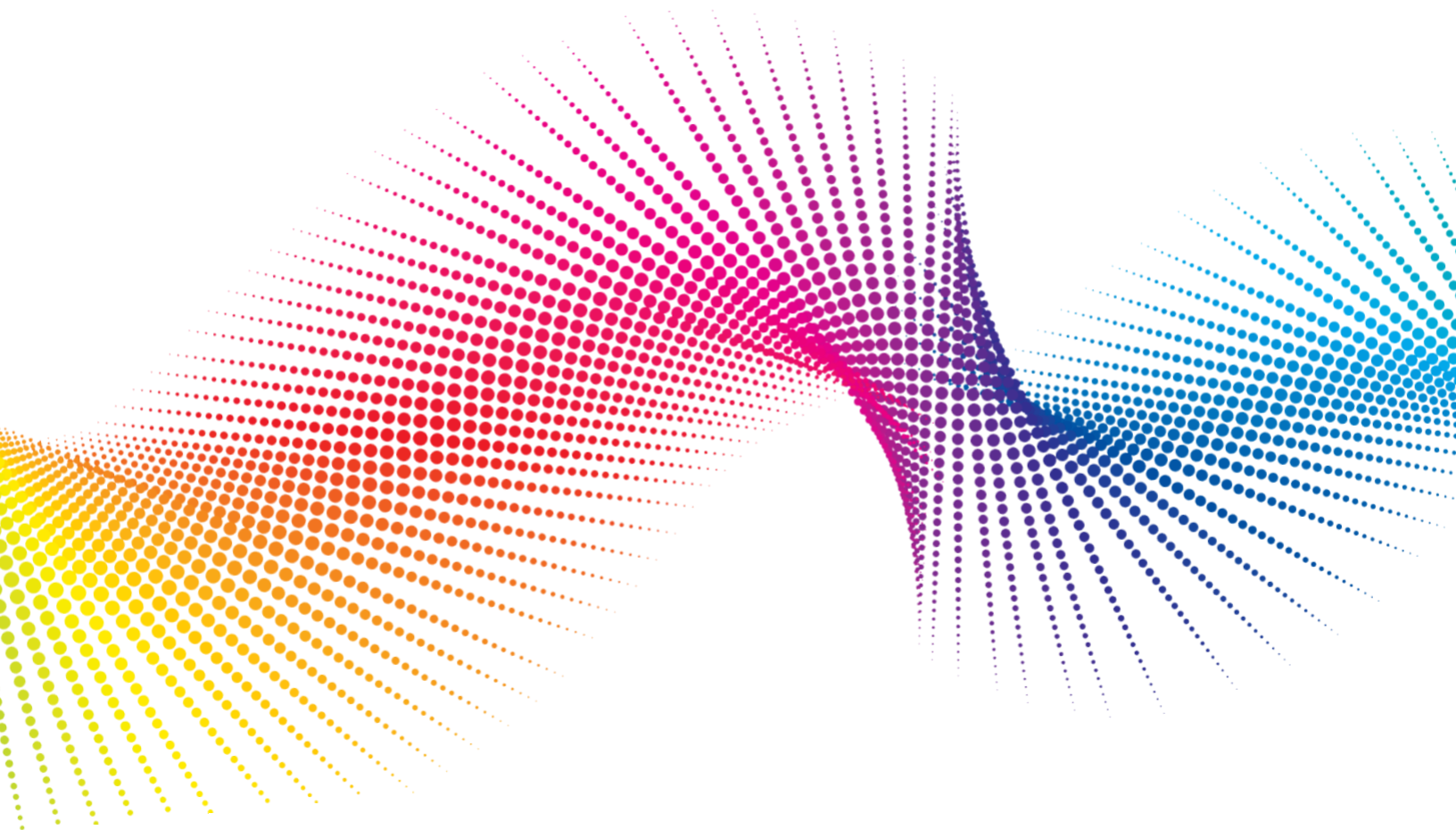
M. en C. Euler Hernández Contreras

3° Parcial

Viernes 25 de Mayo de 2018

Reporte de 9° practica **FINAL**

Alumno: Aaron Antonio Garcia Gonzalez



Índice	
Marco Teórico	3
Instrucciones:	5
Capturas de Pantalla	9
Conclusiones.....	14
Referencias	15

Marco Teórico

Hay situaciones en las que necesitamos hacer más de una consulta al mismo tiempo, y todas tienen que ser correctas para que los datos sean consistentes y tengan sentido, de otro modo tendríamos información "colgada" y desvinculada. En el mejor de los casos esta información provocaría un desperdicio de espacio en la base de datos, pero lo más probable es que además lleve a errores a la hora de mostrarse y de hacer cálculos con ella.

Parte de este problema se resuelve teniendo una buena estructura de base de datos, con claves foráneas relacionando los campos correspondientes.

Pero para mayor seguridad, es bueno usar lo que en jerga *database* se conoce como transacciones.

Las transacciones sirven para asegurar la consistencia de la información, asegurando que un conjunto de sentencias se ejecuten correctamente, o no se ejecuten.

Una transacción tiene dos finales posibles, COMMIT y ROLLBACK. Por defecto, MySQL trae activado el modo autocommit, es decir, realizada una transacción (por ejemplo un INSERT, UPDATE o DELETE) el mismo es confirmado apenas es ejecutado. Para desactivar el autocommit, se puede desactivar el autocomit ejecutando el comando:

```
SET AUTOCOMMIT=0;
```

Por defecto, MySQL se ejecuta en modo autocommit. Esto significa que tan pronto como se ejecuta una sentencia se actualiza (modifica) la tabla, MySQL almacenará la actualización en disco.

Si se están usando tablas de transacción segura (como InnoDB o BDB), se puede poner MySQL en modo no-autocommit con el comando siguiente:

```
SET AUTOCOMMIT=0
```

Después de desconectar el modo autocommit asignando cero a la variable *AUTOCOMMIT*, se debe usar COMMIT para almacenar los cambios en disco o ROLLBACK si se quieren ignorar los cambios hechos desde el principio de la transacción.

Si se quiere desactivar el modo autocommit para una serie de sentencias, se puede usar una sentencia START TRANSACTION:

```
START TRANSACTION;
```

```
SELECT @A:=SUM(salary) FROM table1 WHERE type=1;
```

```
UPDATE table2 SET summary=@A WHERE type=1;
```

```
COMMIT;
```

Se puede usar BEGIN y BEGIN WORK en lugar de START TRANSACTION para iniciar una transacción. START TRANSACTION fue añadido en MySQL 4.0.11; es la sintaxis SQL-99 y es el modo recomendado para empezar una transacción. BEGIN y BEGIN WORK están disponibles desde MySQL 3.23.17 y 3.23.19, respectivamente.

Si no se están usando tablas de transacción segura, cualquier cambio será almacenado inmediatamente, independientemente del estado del modo autocommit.

Si se usa una sentencia ROLLBACK después de actualizar una tabla no transaccional, se obtendrá un error (ER_WARNING_NOT_COMPLETE_ROLLBACK) como un aviso. Todas las tablas de transacción segura serán restauradas, pero cualquier tabla de transacción no segura no cambiará.

Si se usa START TRANSACTION o *AUTOCOMMIT=0*, se debe usar el diario binario MySQL para copias de seguridad en lugar del antiguo diario de actualización. Las transacciones se almacenan en el diario binario de una vez, después de COMMIT, para asegurar que las transacciones que se han rebobinado no se almacenen.

Se puede modificar el nivel de aislamiento para transacciones con *ISOLATION LEVEL*.

LOCK TABLES bloquea tablas para el hilo actual. UNLOCK TABLES libera cualquier bloqueo para el hilo actual. Todas las tablas bloqueadas por el hilo actual serán implícitamente desbloqueadas cuando el hilo realice otro LOCK TABLES, o cuando la conexión con el servidor se cierre.

Para usar LOCK TABLES en MySQL 4.0.2 se necesita el privilegio global LOCK TABLES y un privilegio SELECT en las tablas involucradas. En MySQL 3.23 se necesitan los privilegios SELECT, INSERT, DELETE y UPDATE para las tablas.

El principal motivo para usar LOCK TABLES es para la emulación de transacciones o para obtener mayor velocidad en la actualización de tablas. Esto se explica con más detalle más tarde.

Si el hilo obtiene un bloqueo READ en la tabla, ese hilo (y todos los demás) sólo pueden leer de la tabla. Si obtiene un bloqueo WRITE, entonces sólo el hilo que tiene el bloqueo puede leer de o escribir en la tabla. El resto de los hilos serán bloqueados.

La diferencia entre READ LOCAL y READ es que READ LOCAL permite ejecutar sentencias no conflictivas mientras el bloqueo esté activo. Esto no podrá ser usado si se van a manipular los ficheros de la base de datos fuera de MySQL mientras se mantiene el bloqueo.

Cuando se usa LOCK TABLES, se deben bloquear todas las tablas que se vayan a usar y se debe usar el mismo alias que se vaya a usar en las consultas. Si se está usando una tabla muchas veces en una consulta (con alias), se debe tener un bloqueo para cada alias.

Instrucciones:

Crear las instrucciones lectura, escritura

1. Mostrar el nombre de los socios que se apellidan hernandez

```
select idSocio, nombre
from socio
where nombre like "Hern_andez%";
```

2. Agregar un proveedor

```
insert into proveedor values
(3000,"Prove 1", "55-61-55-55-24");
select * from proveedor
where idproveedor between 200 and 3500;
```

- Instrucción a realizar
- Instrucción
- Notas u observaciones

Begin:

Le indicara al compilador que se comenzara a trabajar con el modo transaccional

Rollback:

Para usarlo, necesitamos primero el begin

3. Abrimos otra terminal, simulando otra transaccion(O2), y la cerramos inesperadamente, eso simulo una falla de sistema

```
Begin;
```

4. Abrimos otra terminal, simulando otra transaccion (N03), pero en este caso si la confrmaremos antes de cerrarla

```
Begin;
commit;
```

5. Abrimos otra terminal, simulando otra transaccion (N04)

Begin;

Lectura:

```
select idSocio, nombre
from socio
where nombre like "Hern_andez%";
```

Escritura:

```
insert into proveedor values
(3001,"Prove 2", "55-61-25-55-24");
```

Lectura

```
select idSocio, nombre
from socio
where nombre like "Hern_andez%";
```

```
Commit;
```

Falla el sistema de nuevo, entonces esta vez si guarda la insercion, ya que se confirmo antes del fallo

Bloqueos

`lock tables ____ [write/read]` (por cada relacion)
`unlock tables;` (Para desbloquear)

Multiples BLOQUEOS: Lectura (compartido- otras terminales pueden leer el contenido libremente, pero en la terminal donde esta la Transaccion1 no podra leer otras relaciones, las otras relaciones se encuentran aisladas para esa terminal).

Escritura:

1. Mostrar el nombre de los socios que se apellidan hernandez

```
select idSocio, nombre
from socio
where nombre like "Hernandez%";
```

2. Agregar un proveedor

```
insert into proveedor values
(3000,"Prove 1", "55-61-55-55-24");

select * from proveedor
where idproveedor between 200 and 3500;
```

3. Transaccion NO1

- 3.1. Bloquear socio

```
lock tables socio read;
```

- 3.2. Lectura socio

ES POSIBLE

Leer cualquier otra no es posible

- 3.3. Lectura club

- 3.4. Lectura proveedor ^

No nos permiteira

Abrir una nueva ventana y en teoria debe de dejar leer todas

4. Transaccion T2

- 4.1. Lectura socio

```
select * from socio;
```

- 4.2. Lectura club;

```
select * from club;
```

4.3. Lectura proveedor

```
select * from proveedor;
```

Abrimos otra terminal

Esta tambien puede hacer Todo

Regresamos a la primera terminal con

```
unlock tables;
```

Ahora ya se podra hacer todas lecturas que no pudimos hacer en la transaccion

5. Transaccion N02

Bloqueo escritura proveedor

5.1. Lectura proveedor

```
select * from proveedor  
where idproveedor between 200 and 3500;
```

5.2. Escritura en proveedor

```
insert into proveedor values  
(3024,"Prove 1", "55-61-55-55-24");  
insert into proveedor values  
(3030,"Prove 3", "55-61-55-55-24");
```

5.3. Lectura anterior

```
select * from proveedor  
where idproveedor between 200 and 3500;
```

5.4. Lectura Socio

```
select * from socio;
```

5.5. Lectura club

```
select * from club;
```

6. Transaccion NO3 (otra terminal)

6.1. Lectura socio

6.2. Lectura club

```
select * from socio;
```

6.3. Lectura club

```
select * from socio;
```

6.4. Lectura club

6.5. Lectura gerente
`select * from socio;`

6.6. Lectura proveedor

Nos dejara en espera, ya que esta bloqueada por la transaccion 1, aplicando la propiedad de aislamiento y se podra hacer hasta que se libere.

Vamos a otra terminal y hacemos las mismas lecturas que en la transaccion 3

Quitamos el bloqueo en la primera transaccion & al mismo tiempo se quitara la espera en las otras terminales

7. Transaccion NO4

7.1. Bloqueo de escritura proveedor

7.2. Escritura en proveedor

```
insert into proveedor values  
(3005,"Prove 4", "55-61-25-4X-24");  
select * from proveedor  
where idproveedor between 200 and 3500;
```

8. Transaccion NO5 (otra Transaccion)

8.1. Escritura

```
insert into proveedor values  
(3015,"Prove 6", "55-61-25-4Y-24");
```

Esta Transaccion se queda **en espera**

9. Transaccion NO6 (otra Transaccion)

9.1. Escritura

```
insert into proveedor values  
(3006,"Prove 6", "55-61-25-4Y-24");
```

Esta transaccion tambien se queda **en espera**

10. Transaccion NO07 (otra Transaccion)

10.1. Lectura

```
select * from proveedor  
where idproveedor between 200 and 350
```

Esta transaccion tambien se queda **en espera**

10.2. Desbloqueamos en la transaccion no1, de inmediato todas las transacciones que estaban en espera se liberan realizando la instrucción por la cual estaban retenidas.

Capturas de Pantalla

```
mysql> select idSocio, nombre
-> from socio
[   -> where nombre like "Hern_nde%";
+-----+-----+
| idSocio | nombre |
+-----+-----+
|      13 | HERNANDEZ ESTRADA ALAN YESSAHIR |
|      14 | HERNANDEZ ROSAS DANIEL ALEJANDRO |
|     113 | HERNANDEZ HERNANDEZ GUSTAVO |
|     114 | HERNANDEZ SOSA HAZEL ANTONIO |
|     141 | HERNANDEZ SANCHEZ ABIGAIN ELISA |
|     142 | HERNANDEZ SANTOS PEDRO EVERARDO |
|     170 | HERNANDEZ LOPEZ JOSE ROGELIO |
|     171 | HERNANDEZ MEJIA JESUS ALEJANDRO |
|     172 | HERNANDEZ MEJIA LUIS CARLOS |
+-----+-----+
9 rows in set (0.00 sec)
```

```
mysql> insert into proveedor values
-> (3034,"Prove 1", "55-61-55-55-24");
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> select * from proveedor
[   -> where idproveedor between 200 and 3500;
+-----+-----+-----+
| idProveedor | nombre | tel |
+-----+-----+-----+
|          3000 | Prove 1 | 55-61-55-55-24 |
|          3001 | Prove 2 | 55-61-25-55-24 |
|          3034 | Prove 1 | 55-61-55-55-24 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> use sams;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

```
[mysql> use sams;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> begin;
Query OK, 0 rows affected (0.00 sec)

mysql> insert into proveedor values
[   -> (3001,"Prove 2", "55-61-25-55-24");
ERROR 1062 (23000): Duplicate entry '3001' for key 'PRIMARY'
mysql> insert into proveedor values (3023,"Prove 2", "55-61-25-55-24");
Query OK, 1 row affected (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> lock tables socio read;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from proveedor;
ERROR 1100 (HY000): Table 'proveedor' was not locked with LOCK TABLES
mysql>
```

```
mysql> select * from socio;
```

idSocio	nombre	direccion	tel
1	ANGELES DE LA CRUZ ANDREE MICHEL IRVING	Av. Juarez. s/n	554-234-5555
2	ARMENDARIZ CRUZ OCTAVIO ARTURO	Av. Juarez. s/n	554-234-5555
3	CARVAJAL MURILLO ANA VICTORIA	Av. Juarez. s/n	554-234-5555
4	CASTAÑEDA GONZALEZ LEONARDO MAURICIO	Av. Juarez. s/n	554-234-5555
5	CASTILLO GONZALEZ IVAN	Av. Juarez. s/n	554-234-5555
6	CRUZ CHAVEZ EDGAR DANIEL	Av. Juarez. s/n	554-234-5555
7	FLORES GALICIA OMAR ISAIAS	Av. Juarez. s/n	554-234-5555
8	GARCIA CERVANTES ANEL	Av. Juarez. s/n	554-234-5555
9	GARCIA MORA OSVALDO	Av. Juarez. s/n	554-234-5555
10	GOMEZ TETLALMATZI JESSICA	Av. Juarez. s/n	554-234-5555
11	GONZALEZ ZAVALA HUGO CESAR	Av. Juarez. s/n	554-234-5555
12	GUZMAN RIOS RAFAEL ANGEL	Av. Juarez. s/n	554-234-5555
13	HERNANDEZ ESTRADA ALAN YESSAHIR	Av. Juarez. s/n	554-234-5555
14	HERNANDEZ ROSAS DANIEL ALEJANDRO	Av. Juarez. s/n	554-234-5555
15	LOPEZ OBREGON ERIKA LISSETTE	Av. Juarez. s/n	554-234-5555
16	MONROY MORAN SERGIO ISRAEL	Av. Juarez. s/n	554-234-5555
17	NAJER GONZALEZ LESSLYE ALY	Av. Juarez. s/n	554-234-5555
18	OVIDEO ESPINOZA JOSAFAT	Av. Juarez. s/n	554-234-5555
19	PEREZ MORALES MARCELA	Av. Juarez. s/n	554-234-5555
20	PIMENTEL MARTINEZ WALDO	Av. Juarez. s/n	554-234-5555
21	RAMIREZ LITENEZ ALAN CESAR	Av. Juarez. s/n	554-234-5555

```
mysql> select * from proveedor  
-> where idproveedor between 200 and 3500;
```

idProveedor	nombre	tel
3000	Prove 1	55-61-55-55-24
3001	Prove 2	55-61-25-55-24
3023	Prove 2	55-61-25-55-24
3034	Prove 1	55-61-55-55-24

4 rows in set (0.00 sec)

```
mysql> select * from proveedor  
-> where idproveedor between 200 and 3500;
```

idProveedor	nombre	tel
3000	Prove 1	55-61-55-55-24
3001	Prove 2	55-61-25-55-24
3023	Prove 2	55-61-25-55-24
3034	Prove 1	55-61-55-55-24

4 rows in set (0.00 sec)

```
mysql> insert into proveedor values  
-> (3024,"Prove 1", "55-61-55-55-24");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> insert into proveedor values  
-> (3030,"Prove 3", "55-61-55-55-24");  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from proveedor  
-> where idproveedor between 200 and 3500;
```

idProveedor	nombre	tel
3000	Prove 1	55-61-55-55-24
3001	Prove 2	55-61-25-55-24
3023	Prove 2	55-61-25-55-24
3024	Prove 1	55-61-55-55-24
3030	Prove 3	55-61-55-55-24
3034	Prove 1	55-61-55-55-24

6 rows in set (0.01 sec)

```
mysql> select * from club;
```

idClub	nombre	direccion
tel	idEdo	
1	Aguascalientes	Av. Independencia No. 2351, Col. La Concepción, C.P.20110
Tel.: (449)9 10 70 00	1	
2	Santa Mónica	Av. Siglo XXI No. 3980, Col. Rancho Santa Mónica, C.P.20289, Aguascalientes.
Tel.: (449)918-4009/918-4214	1	
3	Macro Plaza Tijuana	Blvd. Insurgentes No. 18015, Col. La Mesa c.P.22105
Tel.: (664)9695834	2	
4	Mexicali	Blvd. Lázaro Cárdenas 1801, Col. Ex ejido de Zacatecas C.P.21090
Tel.: (686)592-141-48	2	
5	Tijuana	Blvd. Sánchez Taboada 4005, Zona Río, c.P.22320
Tel.: (664)6-34-69-79	2	
6	Los Cabos	Carretera Transpeninsular Km.2, Los Cabos-La Paz Previo La Huerta, C.P.23463, Cabo San Lucas.
Tel.: (624) 144 71 24	3	
7	Coahuila de Zaragoza	Av. Agustín Olachea esq. Libramiento Sur s/n. Col. El Zacatal, La Paz.
Tel.: (981)816 6268	3	
8	Campeche	María Lavalle Manzana D Lote 1, Fraccionamiento 2, Col. Sector Fundadores de A-Kim-pech, c.P.24028
Tel.: (981)816 6268	4	
9	Ciudad del Carmen	Av.Periférico Norte No.52, Col. Asa Poniente, C.P.24118
Tel.: (983)286 1222	4	

192 rows in set (0.00 sec)

```
mysql> select * from proveedor;
```

```
[mysql> unlock tables;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql>
```

```
mysql> select * from proveedor;
```

idProveedor	nombre	tel
22	Grupo Bimbo S.A de C.V.	01(55)52-68-65-85
24	Sabritas	01(55)52-02-73-73
25	Colgate-Palmolive	01(800)001-1400
27	Sony Mexico	01(55)57-59-85-25
30	Samsung Electronics Mexico	01(55)57-59-85-25
32	LG Electronics México	01(55)57-58-67-56
33	Grupo La Moderna	01(772)279-79-01
3000	Prove 1	55-61-55-55-24
3001	Prove 2	55-61-25-55-24
3023	Prove 2	55-61-25-55-24
3024	Prove 1	55-61-55-55-24
3030	Prove 3	55-61-55-55-24
3034	Prove 1	55-61-55-55-24

13 rows in set (1 min 40.39 sec)

```

mysql> mysql> use sams;
ERROR 1049 (42000): Unknown database 'samuse'
mysql> use sams;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> insert into proveedor values
-> (3009,"Prove 4", "55-61-25-4X-24");
Query OK, 1 row affected (0.00 sec)

mysql> select * from proveedor
-> where idproveedor between 200 and 3500;
+-----+-----+-----+
| idProveedor | nombre | tel |
+-----+-----+-----+
| 3000 | Prove 1 | 55-61-55-55-24 |
| 3001 | Prove 2 | 55-61-25-55-24 |
| 3009 | Prove 4 | 55-61-25-4X-24 |
| 3023 | Prove 2 | 55-61-25-55-24 |
| 3024 | Prove 1 | 55-61-55-55-24 |
| 3030 | Prove 3 | 55-61-55-55-24 |
| 3034 | Prove 1 | 55-61-55-55-24 |
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql>

```

```

mysql> insert into proveedor values
-> (3015,"Prove 6", "55-61-25-4Y-24");

```

```

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.6.39 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use sams;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> insert into proveedor values
-> (3006,"Prove 6", "55-61-25-4Y-24");

```

```
Database changed
mysql> select * from proveedor
    -> where idproveedor between 200 and 3500;
```

```
mysql> insert into proveedor values
[    -> (3015,"Prove 6", "55-61-25-4Y-24");
Query OK, 1 row affected (7 min 59.32 sec)

mysql> █
```

```
Query OK, 1 row affected (4 min 9.26 sec)
```

```
mysql>
mysql> █
```

```
Database changed
mysql> select * from proveedor
    -> where idproveedor between 200 and 3500;
```

idProveedor	nombre	tel
3000	Prove 1	55-61-55-55-24
3001	Prove 2	55-61-25-55-24
3009	Prove 4	55-61-25-4X-24
3012	Prove 6	55-61-25-4Y-24
3015	Prove 6	55-61-25-4Y-24
3023	Prove 2	55-61-25-55-24
3024	Prove 1	55-61-55-55-24
3030	Prove 3	55-61-55-55-24
3034	Prove 1	55-61-55-55-24

```
9 rows in set (2 min 13.63 sec)
```

```
mysql>
mysql>
mysql> _
```

Conclusiones

Por el escaso tiempo de practicas, ya no pudimos ver mas a fondo el modo transaccional, pero si nos deja con la intriga para seguir estudiando este topico, y posteriormente en niveles mas avanzados de la carrera, poder tomar alguna optativa de esta area, me causa mucho ruido como trabajar en este modo en un sistema ya establecido, sera uno de mis proyectos para estas vacaciones, probablemente realice mi propio sistema, implementando lo que me quede con ganas de i plementar en el proyecto en equipos, espero poder seguir recibiendo apoyo de estos temas por parte tuya, te seguire molestando.

Referencias

R. Elmasri and S. Navathe, Fundamentos de sistemas de bases de datos. Pearson Education, 2007. [Online]. Available: <https://books.google.com.mx/books?id=NT3uPQAACAAJ>

C. Ricardo, Bases de datos. McGraw-Hill Interamericana, 2000. [Online]. Available: <https://books.google.com.mx/books?id=BmVHAQAACAAJ>

(2017) Mysql :: Mysql documentation. [Online]. Available: <https://dev.mysql.com/doc/>

<https://www.anerbarrena.com/tipos-dato-mysql-5024/>

<https://advenis.wordpress.com/2010/04/21/tipos-de-datos-en-mysql/>

<https://es.wikipedia.org/wiki/MySQL>

<https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-son-los-metadatos-y-cual-es-su-utilidad>