

---

# **BASES DE DATOS**

---

Practica 03

**DIAZ MEDINA JESÚS KAIMORTS**

**Viernes, 24 de Febrero del 2017.**

---

# INDICE

---

MARCO TEÓRICO.....	2
DESARROLLO DE LA PRÁCTICA.....	8
INSTRUCCIONES Y CODIFICACIÓN.....	8
ENTORNO DE DESARROLLO DE LA PRÁCTICA (SCREENSHOTS).....	13
DIAGRAMA ENTIDAD-RELACION.....	19
CONCLUSIONES.....	21
BIBLIOGRAFÍA .....	21

# BASES DE DATOS

## Practica 03

### MARCO TEÓRICO

- **MOTORES DE ALMACENAMIENTO**

Un motor de almacenamiento es una parte esencial de un SGDB puesto que se encarga de crear, recuperar, actualizar y borrar los datos de una base de datos.<sup>1</sup>

Para conocer que motores están disponibles, basta con escribir en una consola de MySQL la siguiente consulta:

```
show engines;
```

Y obtendremos algo como lo siguiente:

Engine	Support	Comment
MyISAM	DEFAULT	Default engine as of MySQL 3.23 with great performance
MEMORY	YES	Hash based, stored in memory, useful for temporary tables
InnoDB	YES	Supports transactions, row-level locking, and foreign keys
BerkeleyDB	NO	Supports transactions and page-level locking
BLACKHOLE	YES	/dev/null storage engine (anything you write to it disappears)
EXAMPLE	NO	Example storage engine
ARCHIVE	YES	Archive storage engine
CSV	YES	CSV storage engine
ndbcluster	DISABLED	Clustered, fault-tolerant, memory-based tables
FEDERATED	YES	Federated MySQL storage engine
MRG_MYISAM	YES	Collection of identical MyISAM tables
ISAM	NO	Obsolete storage engine

- **TIPOS DE MOTORES DE ALMACENAMIENTO**

Durante esta práctica solamente nos centraremos en dos motores de almacenamientos de los previamente mostrados, los cuales son: **MyISAM, MERGE y InnoDB.**

---

<sup>1</sup> (Mario López, 2008)

## MyISAM.

Se basa en el antiguo ISAM, al que añade muchas mejoras, es el motor que usa MySQL por defecto. Es una buena combinación entre funcionalidad y rendimiento, aunque carece de algunas características interesantes.

### Características

Límite de 2 <sup>32</sup> registros.	BLOB <sup>2</sup> y TEXT pueden ser índices.
Máximo de 64 índices por tabla.	Permite un gran tamaño en las tablas (hasta 256TB)
Máximo de 16 columnas por índice.	No soporta transacciones.
Los datos son independientes de la máquina y el sistema operativo.	Bloquea los datos a nivel de tabla.
Permite campos índices como NULL	No permite claves ajenas.

Este motor pone especial empeño en la rapidez de las operaciones de lectura (predominio de SELECT), es una de las razones por las que MySQL es tan popular en la web, ya que la mayoría de las operaciones que se realizan son de este tipo. Que no tenga que hacer comprobaciones de integridad referencial también influye en su velocidad.

### Ejemplo

```
CREATE TABLE pruebaMyISAM (  
    codigo varchar(5) default NOT NULL,  
    descripcion varchar(255) default NULL,  
    PRIMARY KEY (codigo)  
) ENGINE=MyISAM;
```

En general no hará falta indicar el uso de este motor pues es el que se usa por defecto.

## MERGE.

Permite combinar varias tablas de igual estructura en una única tabla, pudiendo así realizar consultas sobre una tabla que nos devuelve datos de varias.

Límite de 2 <sup>32</sup> registros.	No permite REPLACE
Las tablas “base” deben ser MyISAM	No soporta transacciones
Bloqueo a nivel de tabla	En su creación no comprueba que las tablas que usa existan y tengan una estructura idéntica.
No tiene índices, usa los de las tablas “base” (salvo FULLTEXT)	La lectura es más lenta al tener que ir consultando la clava en cada una de las tablas subyacentes.

Una de sus funcionalidades puede ser partir una tabla muy grande en otras más pequeñas y, al unir las con MERGE, permitirnos trabajar con ellas como si fuesen una sola.

<sup>2</sup> Binary Large Object (Generalmente son imágenes, archivos de sonido y otros objetos multimedia)

```
mysql> CREATE TABLE t1 (
->   a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
->   message CHAR(20));
mysql> CREATE TABLE t2 (
->   a INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
->   message CHAR(20));
mysql> INSERT INTO t1 (message) VALUES ('Testing'),('table'),('t1');
mysql> INSERT INTO t2 (message) VALUES ('Testing'),('table'),('t2');
mysql> CREATE TABLE total (
->   a INT NOT NULL AUTO_INCREMENT,
->   message CHAR(20), INDEX(a))
->   TYPE=MERGE UNION=(t1,t2) INSERT_METHOD=LAST;
```

## InnoDB.

- Está considerado como uno de los motores más avanzados para el almacenamiento de datos en MySQL. Provee un motor sólido con soporte completo de transacciones (es ACID compliant), permite el bloqueo de datos a nivel de registro permitiendo gran flexibilidad a la hora de utilizar las tablas, controla la integridad referencial, permite claves ajenas y tiene un sistema de recuperación de caídas. No obstante, la piedra de toque de InnoDB es su mecanismo de indexación y cache de los registros pues mantiene una caché de índices y datos en memoria y en disco proporcionando un muy alto rendimiento.

ACID Compliant	Permite claves ajenas y transacciones, soporte de integridad referencial.
<b>El tamaño máximo para una tabla es de 64TB.</b>	No permite índices de FULLTEXT
<b>No mantiene un contador interno de registros (selec count (*) from tabla lento al tener que recorrer todo el índice)</b>	Sistema de recuperación de caídas.
<b>Cambiar la ubicación de la base de datos/tabla es complicado.</b>	Los tamaños de sus logs deben ser inferior a 4GB.
<b>Una tabla no puede tener más de 1000 columnas.</b>	Bloqueo de datos a nivel de registro y no bloquea la lectura durante los selects (mejora la concurrencia)

Ejemplo:

```
mysql> CREATE TABLE CUSTOMER (A INT, B CHAR (20), INDEX (A))
-> ENGINE=InnoDB;
Query OK, 0 rows affected (0.00 sec)
mysql> BEGIN;
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO CUSTOMER VALUES (10, 'Heikki');
Query OK, 1 row affected (0.00 sec)
mysql> COMMIT;
Query OK, 0 rows affected (0.00 sec)
mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)
mysql> INSERT INTO CUSTOMER VALUES (15, 'John');
Query OK, 1 row affected (0.00 sec)
mysql> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)
mysql> SELECT * FROM CUSTOMER;
+-----+-----+
| A    | B      |
+-----+-----+
| 10   | Heikki |
+-----+-----+
1 row in set (0.00 sec)
```

## RESTRICCIONES DE INTEGRIDAD

Durante esta práctica solamente abarcaremos restricciones de integridad CASCADE, ya que es la restricción que hemos utilizado en las prácticas desarrolladas.

Las restricciones de integridad referencial en cascada permiten definir las acciones que MySQL lleva a cabo cuando un usuario intenta eliminar o actualizar una clave a la que apuntan las claves externas existentes.

Las cláusulas REFERENCES de las instrucciones CREATE TABLE y ALTER TABLE admiten las cláusulas ON DELETE y ON UPDATE. Las acciones en cascada también se pueden definir mediante el cuadro de diálogo Relaciones de clave externa.

- **[ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]**
- **[ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]**

NO ACTION es el valor predeterminado si no se especifica ON DELETE u ON UPDATE.

### ON DELETE NO ACTION

Especifica que si se intenta eliminar una fila con una clave a la que hacen referencia las claves externas de las filas existentes en otras tablas, se produce un error y se revierte la instrucción DELETE.

### ON UPDATE NO ACTION

Especifica que si se intenta actualizar un valor de clave en una fila a cuya clave hacen referencia las claves externas de filas existentes en otras tablas, se produce un error y se revierte la instrucción UPDATE.

### ON DELETE CASCADE

Especifica que si se intenta eliminar una fila con una clave a la que hacen referencia claves externas de filas existentes en otras tablas, todas las filas que contienen dichas claves externas también se eliminan.

### ON UPDATE CASCADE

Especifica que si se intenta actualizar un valor de clave de una fila a cuyo valor de clave hacen referencia claves externas de filas existentes en otras tablas, también se actualizan todos los valores que conforman la clave externa al nuevo valor especificado para la clave.<sup>3</sup>

---

<sup>3</sup> (Q., 2013)

## LENGUAJE DE MANIPULACIÓN DE DATOS (DML)

Es un idioma proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos. El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.<sup>4</sup>

Durante esta práctica mencionaremos los elementos que conforman al DML, sin embargo sólo nos enfocaremos en las sintaxis de las sentencias y cláusula que en la práctica se usarán.

Los elementos del lenguaje de manipulación de datos son:

- **SELECT** – Sirve para obtener datos de una base de datos.
- **INSERT** – Se aplica para insertar datos a una tabla.
- **DELETE** – Se usa para modificar datos existentes dentro de una tabla.
- **UPDATE** – Elimina todos los registros de la tabla; no borra los espacios asignados a los registros.

- **SELECT.**

Se guarda el resultado en una tabla llamada “result-set”.

- Sintaxis:

```
SELECT column_name, column_name  
FROM table_name;
```

```
SELECT * FROM table_name;
```

\* => Significa que queremos todas las columnas de la tabla.

- **WHERE**

La sentencia WHERE se usa para extraer sólo los registros que cumplan con una condición. Funciona como un filtro.

- Sintaxis:

```
SELECT column_name, column_name  
FROM table_name  
WHERE column_name operator value;
```

Para unir más de una condición (filtro) dentro de la consulta utilizamos “and”

- Sintaxis:

```
SELECT column_name, column_name  
FROM table_name  
WHERE column_name operator value  
and column_name operator value  
.....
```

---

<sup>4</sup> (cidecame.uaeh.edu.mx, s.f.)

- **CLAUSULA ORDER BY**

La cláusula ORDER BY se utiliza para ordenar los resultados a través de una o más columnas.

La cláusula ORDER BY ordena los registros de manera ascendente por defecto. Para hacerlo de manera descendente, se puede utilizar la cláusula DESC.<sup>5</sup>

- **Sintaxis:**

```
SELECT column_name, column_name  
FROM table_name  
ORDER BY column_name, column_name ASC|DESC;
```

---

<sup>5</sup> (Edu4Java, s.f.)



# DESARROLLO DE LA PRÁCTICA.

## INSTRUCCIONES Y CODIFICACIÓN

Cuando nosotros utilizamos `* = All` files ---> *Proyeccion*

```
SELECT * from __ nombre de relacion(es) __ (X)
where __ condiciones __ (reunion)
and __ condicion __ (seleccion)
and __ condicion __ (seleccion)
```

Producto cartesiano: Concatenar las tupas. Todos con todos.

**where:** clausula para establecer condiciones de reunión

usamos **and** para asociarlo

1. **create** database sams
2. **use** sams;
3. source **path:** \.... (arrastrar desde donde se ubica)

Resolver las siguientes consultas.

- 3.1 Mostrar toda la informacion de los proveedores
- 3.2 Mostrar el nombre y el telefono de los socios.
- 3.3 Que productos cuestan entre \$100 y \$600?
- 3.4Cuál es lla direccion de los siguientes clubes:  
Tepeyac  
Toreo  
Pachuca
- 3.5 Como se llaman los gerentes que se apellidan HERNANDEZ.
- 3.6 Cual es el tel del proveedor LG y de sabritas.
- 3.7 Que socios se apellidan Gonzalez, incluir el tel
- 3.8 Que servicios se tienen registrados?
- 3.9 Cuantos socios se apellidan MARTINEZ
- 3.10 Que socios se llaman JUAN??

Solucion:

1. **SELECT \* from** club;
2. **SELECT count(\*) from** club; -> Nos dice el numero de registros.
3. **SELECT idclub, nombre from** club;

```
select tel, nombre from club;
select idclub, idedo, from club;
select * from club
order by nombre DESC;
```

```
select nombre, idedo from club
order by idEdo;
```

```
select nombre, idedo from club
order by nombre;
```

- 
- 3.1 Mostrar toda la información de los proveedores

```
SELECT * from proveedor;
SELECT nombre from proveedor;
```

```
SELECT * from proveedor
order by nombre;
```

### 3.2 Mostrar el nombre y el telefono de los socios.

```
SELECT nombre, tel from socio;

SELECT nombre, tel from socio
order by nombre;
```

### 3.3 Que productos cuestan entre \$100 y \$600?

```
desc producto;

SELECT nombre, preciounitario
from producto
where preciounitario>=100
and preciounitario<=600

SELECT nombre, preciounitario
from producto
order by preciounitario;

SELECT nombre, preciounitario
from producto
where preciounitario>=1000
and preciounitario<=6000;
```

---

Otra manera

```
SELECT nombre, preciounitario
from producto
where preciounitario between 1000
and 6000;
```

---

```
SELECT nombre, preciounitario
from producto
where preciounitario between 1000
and 6000
order by preciounitario;
```

```
SELECT nombre, preciounitario
from producto
where preciounitario between 1000
and 6000
order by nombre;
```

### 3.4Cuál es lla direccion de los siguientes clubes:

Tepeyac  
Toreo  
Pachuca

```
SELECT nombre, direccion
from club
where nombre = "Tepeyac"
or nombre = "Toreo"
or nombre = "Pachuca";
```

```
SELECT nombre as CLUB, direccion //Renombrar por club (1
palabra)
from club
where nombre = "Tepeyac"
or nombre = "Toreo"
or nombre = "Pachuca";
```

```
SELECT nombre as "UN CLUB CHIDO", direccion //Renombrando con
mas palabras
from club
where nombre = "Tepeyac"
or nombre = "Toreo"
or nombre = "Pachuca";
```

### 3.5 Como se llaman los gerentes que se apellidan HERNANDEZ.

```
desc gerente;
SELECT nombre from gerente;
```

```
SELECT nombre from gerente      (wincard = comodin) -> apellido
paterno
where nombre like "HERNANDEZ%"; % hace referencia que no importa
lo que haya despues de la Z de HERNANDEZ
```

```
SELECT nombre from gerente      (wincard = comodin) -> Apellido
materno
where nombre like "% HERNANDEZ%"; % hace referencia que no
importa lo que haya despues de la Z de HERNANDEZ
```

```
SELECT nombre from gerente
where nombre like "% % ALE%";
```

or

```
SELECT nombre from gerente
where nombre like "HERNAN%"
or nombre like "% HERNAN%";
```

### 3.6 Cual es el tel del proveedor LG y de sabritas.

```
SELECT nombre, tel from proveedor
where nombre like "LG%"
or nombre like "Sabr%";
```

\*\*\*EL or en este caso funciona como and

3.7 Que socios se apellidan Gonzalez, incluir el tel

```
SELECT nombre, tel from socio
where nombre like "Gonzal%"
order by nombre;
```

3.8 Que servicios se tienen registrados?

```
select nombre from servicio;
```

3.9 Cuantos socios se apellidan MARTINEZ

```
SELECT count(*) from socio
where nombre like "Martinez%" or
nombre like "% Martinez%";
```

3.10 Que socios se llaman JUAN??

```
SELECT nombre from socio
where nombre like "Martinez%" or
nombre like "% Martinez%";
```

```
SELECT nombre from socio
where nombre like "% JUAN%";
```

#### 4. PARTE 2: CONSULTAS

- 4.1 Renombrar la relacion gerente y llamarle empleado.
- 4.2 Renombrar el campo nombre en estado y llamarle estado
- 4.3 Eliminar el campo tel de los socios
- 4.4 Quitar el enlace entre proveedor y producto
- 4.5 Redefinir la PK en la relacion empleado  
empleado(idGerente,nombre) PK Compuesta
- 4.6 Agregar un campo que permita almacenar el  
salario de un empleado
- 4.7 Agregar un campo que permita almacenar  
multiples correos para un socio.

SOLUCION:

- 4.1 Renombrar la relacion gerente y llamarle empleado.

```
ALTER TABLE gerente RENAME AS empleado;
```

- 4.2 Renombrar el campo nombre en estado y llamarle estado

```
ALTER TABLE estado CHANGE COLUMN nombre estado varchar(50);
```

- 4.3 Eliminar el campo tel de los socios

```
desc socio;
ALTER TABLE socio DROP COLUMN tel;
```

4.4 Quitar el enlace entre proveedor y producto (se elimina la llave foranea **no** el campo)

```
show create table producto;

ALTER TABLE producto
drop foreign key FK_producto_1;
```

4.5 Redefinir la PK en la relacion empleado  
empleado(idGerente,nombre) PK Compuesta

```
desc empleado;

ALTER TABLE empleado DROP PRIMARY KEY;
//Se hizo libremente ya que no hay relacion con otra entidad

ALTER TABLE empleado ADD PRIMARY KEY(idGerente,nombre);
```

4.6 Agregar un campo que permita almacenar el  
salario de un empleado

```
ALTER TABLE empleado ADD COLUMN salario DOUBLE;
```

4.7 Agregar un campo que permita almacenar  
multiples correos para un socio.

4.7.1

```
CREATE TABLE emailSocio(
idSocio int(10) unsigned not null,
email varchar(50) not null,
primary key(idSocio, email),
foreign key(idSocio) references Socio(idSocio)
on delete cascade on update cascade
);

desc socio;
show create table socio;
```

## ENTORNO DE DESARROLLO DE LA PRÁCTICA (SCREENSHOTS)

```
mysql> create database sams2;
Query OK, 1 row affected (0.00 sec)

mysql> use sams2;
Database changed
mysql> source C:\Users\Home\Documents\ESCOM\3er Semestre\02 Bases de Datos\P3\samsProfe.sql
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  |
+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> SELECT nombre from proveedor;
+-----+
| nombre                |
+-----+
| Grupo Bimbo S.A de C.V. |
| Sabritas               |
| Colgate-Palmolive      |
| Sony Mexico            |
| Samsung Electronics Mexico |
| LG Electronics México  |
| Grupo La Moderna       |
+-----+
7 rows in set (0.00 sec)

mysql>
mysql> SELECT * from proveedor
-> order by nombre;
+-----+-----+-----+
| idProveedor | nombre                | tel                |
+-----+-----+-----+
| 25          | Colgate-Palmolive      | 01(800)001-1400    |
| 22          | Grupo Bimbo S.A de C.V. | 01(55)52-68-65-85 |
| 33          | Grupo La Moderna       | 01(772)279-79-01  |
| 32          | LG Electronics México  | 01(55)57-58-67-56 |
| 24          | Sabritas               | 01(55)52-02-73-73 |
| 30          | Samsung Electronics Mexico | 01(55)57-59-85-25 |
| 27          | Sony Mexico            | 01(55)57-59-85-25 |
+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> SELECT nombre, tel from socio;
+-----+-----+
| nombre                | tel                |
+-----+-----+
| ANGELES DE LA CRUZ ANDREE MICHEL IRVING | 554-234-5555      |
| ARMENDARIZ CRUZ OCTAVIO ARTURO          | 554-234-5555      |
| CARVAJAL MURILLO ANA VICTORIA           | 554-234-5555      |
+-----+-----+
```

(La tabla mostrada por la instrucción **SELECT nombre,tel FROM socio;** , es muy larga, por tanto se recortó)

```
mysql> SELECT nombre, tel from socio
-> order by nombre;
```

nombre	tel
AGUILA PEREZ JUAN CARLOS	55-55-55-55-55
ALVARADO BAUTISTA ALBERTO ANGEL	55-55-55-55-92
ANGELES DE LA CRUZ ANDREE MICHEL IRVING	554-234-5555

(La tabla mostrada por la instrucción **SELECT nombre,tel FROM nombre;** , es muy larga, por tanto se recortó)

```
mysql> SELECT nombre, preciounitario
-> from producto
-> where preciounitario>=100
-> and preciounitario<=600;
Empty set (0.02 sec)

mysql>
mysql> SELECT nombre, preciounitario
-> from producto
-> order by preciounitario;
```

nombre	preciounitario
Barcel Hot Nuts	2.99
Pasta Estrella	3.79
Pasta Monito	3.79
Pasta Engrane	3.79

(La tabla mostrada por la instrucción **SELECT nombre,precioUnitario FROM producto ORDER BY precioUnitario;** , es muy larga, por tanto se recortó)

```
mysql> SELECT nombre, preciounitario
-> from producto
-> where preciounitario>=1000
-> and preciounitario<=6000;
```

nombre	preciounitario
Sony Bravia	4699
Sony PSP	3799
Horno Microondas AGE1072	1349
Horno Microondas AGE107W	1999
Refrigerador 4 puertas	4490
LG Celular GSM ME970	2322.42
LG Celular GSM ME230	4600
LG Celular GSM MG160	5129

8 rows in set (0.00 sec)

Realizan la misma  
operación

```
mysql> SELECT nombre, preciounitario
-> from producto
-> where preciounitario between 1000
-> and 6000;
```

nombre	preciounitario
Sony Bravia	4699
Sony PSP	3799
Horno Microondas AGE1072	1349
Horno Microondas AGE107W	1999
Refrigerador 4 puertas	4490
LG Celular GSM ME970	2322.42
LG Celular GSM ME230	4600
LG Celular GSM MG160	5129

8 rows in set (0.00 sec)

```
mysql> SELECT nombre, direccion
-> from club
-> where nombre = "Tepeyac"
-> or nombre = "Toreo"
-> or nombre = "Pachuca";
```

nombre	direccion
Tepeyac	Av. Ferrocarril Hidalgo No. 431, Col. Guadalupe Tepeyac, C.P.07840
Toreo	Blvd. Manuel Mvila Camacho No. 467, Col. Periodistas, C.P.11220
Pachuca	Av. Río de las Avenidas No. 701, Fracc. Las Terrazas, C.P.42098, Pachuca.

3 rows in set (0.01 sec)

```
mysql> SELECT nombre as CLUB, direccion
-> from club
-> where nombre = "Tepeyac"
-> or nombre = "Toreo"
-> or nombre = "Pachuca";
```

CLUB	direccion
Tepeyac	Av. Ferrocarril Hidalgo No. 431, Col. Guadalupe Tepeyac, C.P.07840
Toreo	Blvd. Manuel Mvila Camacho No. 467, Col. Periodistas, C.P.11220
Pachuca	Av. Río de las Avenidas No. 701, Fracc. Las Terrazas, C.P.42098, Pachuca.

3 rows in set (0.00 sec)



```
mysql> SELECT nombre as "UN CLUB CHIDO"  
-> from club  
-> where nombre = "Tepeyac"  
-> or nombre = "Toreo"  
-> or nombre = "Pachuca";
```

UN CLUB CHIDO
Tepeyac
Toreo
Pachuca

3 rows in set (0.00 sec)

```
mysql> SELECT nombre from gerente  
-> where nombre like "HERNANDEZ%";
```

nombre
HERNANDEZ AGUILAR CARLOS ALBERTO
HERNANDEZ ALAMILLA MIGUEL ANGEL
HERNANDEZ LOPEZ FILEMON
HERNANDEZ ORTIZ JESUS
HERNANDEZ GUERRERO JAVIER IRVING
HERNANDEZ OYARZABAL MAURICIO FRANCISCO

6 rows in set (0.01 sec)

```
mysql> SELECT nombre from gerente  
-> where nombre like "%HERNANDEZ%";
```

nombre
BALLESTEROS <u>HERNANDEZ</u> YOSELIN ANDREA
GALVAN HERNANDEZ MARIA ISABEL
GRACIAN HERNANDEZ SERGIO DEMIAN
HERRERA HERNANDEZ JENIFFER
LARA HERNANDEZ ADRIANA
VERTIZ HERNANDEZ CARLOS
LOPEZ HERNANDEZ YAZMIN SARAHI
GUEVARA HERNANDEZ LUIS AXELL
PERALTA HERNANDEZ IRVING DAVID
DIAZ HERNANDEZ FRANCISCO JAVIER
GOMEZ HERNANDEZ OMAR JESUS
GONZALEZ HERNANDEZ SAMUEL

12 rows in set (0.00 sec)

Apellido Paterno

% => Comodín

Apellido Materno

```
mysql> SELECT nombre, tel from proveedor
-> where nombre like "LG%"
-> or nombre like "Sabr%";
+-----+-----+
| nombre          | tel          |
+-----+-----+
| Sabritas        | 01(55)52-02-73-73 |
| LG Electronics México | 01(55)57-58-67-56 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT nombre, tel from socio
-> where nombre like "Gonzal%"
-> order by nombre;
+-----+-----+
| nombre          | tel          |
+-----+-----+
| GONZALEZ ANGELES FABIAN GERARDO | 55-55-55-55-101 |
| GONZALEZ DIAZ MARTIN AUGUSTO    | 554-234-5555    |
| GONZALEZ ESTRADA HILDA          | 55-55-55-55-68  |
| GONZALEZ GALLEGOS ROSA ISELA    | 55-55-55-55-69  |
| GONZALEZ HERNANDEZ URIEL        | 55-55-55-55-120 |
| GONZALEZ SANCHEZ MARS YUREN     | 554-234-5555    |
| GONZALEZ ZAVALA HUGO CESAR      | 554-234-5555    |
+-----+-----+
7 rows in set (0.00 sec)

mysql> select nombre from servicio;
+-----+
| nombre          |
+-----+
| Farmacia        |
| Foto Centro     |
| Joyería         |
| Óptica          |
| TMA Centro Llantero |
| Service Deli    |
| BWM Banco Walt-Mart |
| Punta del Cielo |
| Servicio a Domicilio |
| Apple shop      |
+-----+
10 rows in set (0.01 sec)

mysql> SELECT count(*) from socio
-> where nombre like "Martinez%" or
-> nombre like "% Martínez%";
+-----+
| count(*) |
+-----+
|          9 |
+-----+
1 row in set (0.10 sec)

mysql> SELECT nombre from socio
-> where nombre like "% JUAN%";
+-----+
| nombre          |
+-----+
| VILLA RUSSELL JUAN PABLO |
| MALDONADO HERNANDEZ JUAN RAFAEL |
| AGUILA PEREZ JUAN CARLOS |
| LINARES CERON JUAN ANTONIO |
| REYES HERNANDEZ JUAN ALFREDO |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT nombre from socio
-> where nombre like "Martinez%" or
-> nombre like "% Martinez%";
+-----+
| nombre |
+-----+
| PIMENTEL MARTINEZ WALDO |
| RAMIREZ MARTINEZ ERICK DANIEL |
| RODRIGUEZ MARTINEZ ARTURO |
| GARCIA MARTINEZ JESE ANDRES |
| MARTINEZ DORANTES ANTONIO ALEJANDRO |
| MARTINEZ GUTIERREZ JOSE ALFONSO |
| MENDOZA MARTINEZ ERICK DANIEL |
| MARTINEZ LUNA NICOLAS |
| MARTINEZ PEÑA JESUS ADRIAN |
+-----+
9 rows in set (0.00 sec)
```

## SEGUNDA PARTE

```
mysql> ALTER TABLE gerente RENAME AS empleado;
Query OK, 0 rows affected (0.25 sec)

mysql> ALTER TABLE estado CHANGE COLUMN nombre estado varchar(50);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc socio;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idSocio | int(10) unsigned | NO | PRI | NULL | |
| nombre | varchar(50) | NO | | NULL | |
| direccion | varchar(100) | NO | | NULL | |
| tel | varchar(15) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> ALTER TABLE socio DROP COLUMN tel;
Query OK, 0 rows affected (0.76 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE producto
-> drop foreign key FK_producto_1;
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente | int(10) unsigned | NO | PRI | NULL | |
| nombre | varchar(50) | NO | | NULL | |
| idClub | int(10) unsigned | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

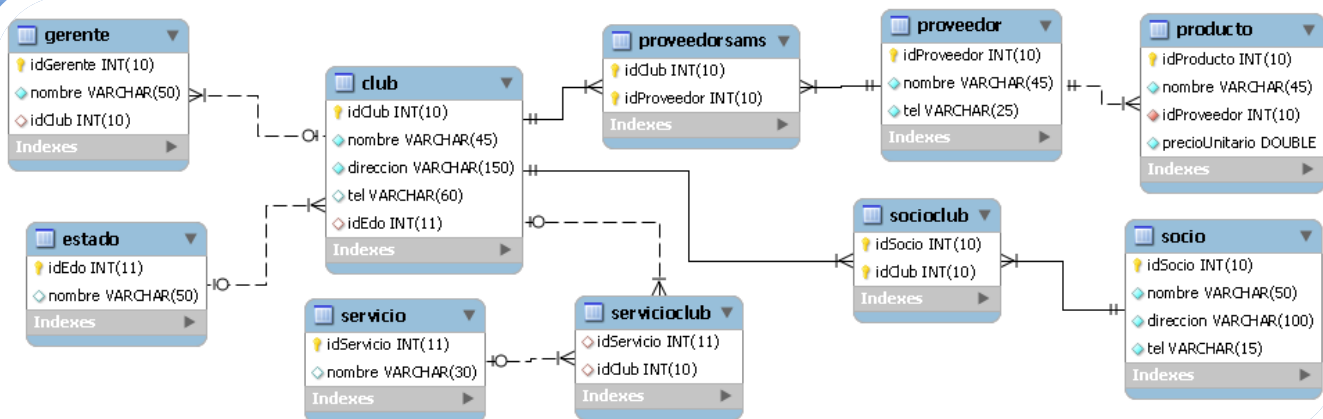
mysql>
mysql> ALTER TABLE empleado DROP PRIMARY KEY;
Query OK, 200 rows affected (0.80 sec)
Records: 200 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE empleado ADD PRIMARY KEY(idGerente,nombre);
Query OK, 0 rows affected (0.89 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

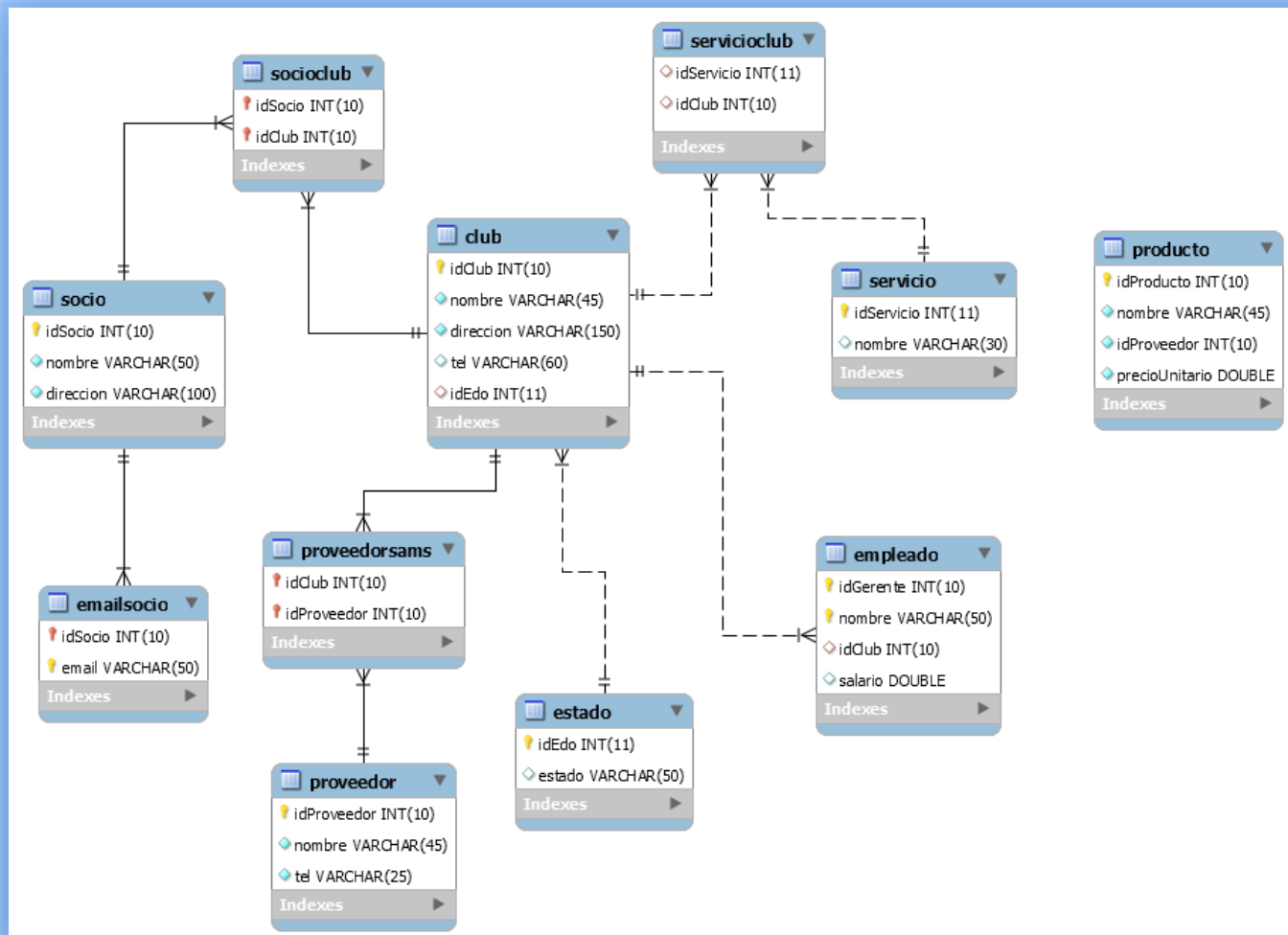
```
mysql> ALTER TABLE empleado ADD COLUMN salario DOUBLE;
Query OK, 0 rows affected (0.76 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> CREATE TABLE emailSocio(
-> idSocio int(10) unsigned not null,
-> email varchar(50) not null,
-> primary key(idSocio, email),
-> foreign key(idSocio) references Socio(idSocio)
-> on delete cascade on update cascade
-> );
Query OK, 0 rows affected (0.33 sec)
```

## DIAGRAMA ENTIDAD-RELACION



## DESPUÉS DE LA PRÁCTICA



# CONCLUSIONES

En el desarrollo de esta practica implementamos dos compiladores: DML (Lenguaje de manipulación de datos) y el DDL(Lenguaje de definición de datos), los cuales nos ayudaron en las operaciones, en su mayoría, de lectura y en la creación del repositorio y sus respectivas entidades. Además de proveer los niveles de aislamiento con relación a los objetos de la base de datos los cuales utilizaron índices de manera interna para leer el contenido (interno) en los archivos físicos. Asimismo, proyectamos e identificamos la integridad lógica de los datos al modificar el contexto de la BD (esquema conceptual) en el que trabajamos: SAMS, otorgando las restricciones pertinentes durante la implementación de atributos dentro de cada tabla con el fin de realizar las asociaciones necesarios para la integridad de la base de datos, esto gracias a un diagrama entidad-relación proporcionado por el profesor.

Cabe mencionar que en esta practica no se mantuvo la base de datos en un estado vacío (como las anteriores), sino que el docente facilitó un script con registros previamente establecidos. De esta manera, hemos avanzado con respecto a identificar el proceso de Operación y Mantenimiento del ciclo de vida de un sistema de base de datos.

# BIBLIOGRAFÍA

*cidecame.uaeh.edu.mx*. (s.f.). Obtenido de CIDECAAME:

[http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/53\\_\\_lenguaje\\_de\\_manipulacin\\_de\\_datos\\_dml.html](http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/53__lenguaje_de_manipulacin_de_datos_dml.html)

*Edu4Java*. (s.f.). Obtenido de SQL con MYSQL: <http://www.edu4java.com/es/sql/sql4.html>

Mario López, J. A. (15 de 01 de 2008). Obtenido de Motores de Almacenamiento en MySQL 5.0:

<https://iessanvicente.com/colaboraciones/motoresMySQL.pdf>

Q., E. (20 de 12 de 2013). *Microsoft*. Obtenido de Microsoft TechNet: [https://technet.microsoft.com/es-es/library/ms186973\(v=sql.105\).aspx](https://technet.microsoft.com/es-es/library/ms186973(v=sql.105).aspx)

Rodríguez., J. R. (2011-2012). *Introducción a las Bases de Datos*. .