

ESCOM - IPN
BASES DE DATOS 2CM12

Introducción a las Bases de Datos

ALUMNO:

Rosas Hernandez Oscar
Andres

PROFESOR:

Euler Hernandez Contreras

Reporte 1 y Reporte 2

1. Parte Teórica

¿Cómo se almacenan grandes colecciones de datos en constante cambio en disco? ¿Cómo podemos hacer que diferentes agentes recuperen, editen y agreguen datos al mismo tiempo?

En lugar de implementar estas funcionalidades nosotros mismos, usaremos una Database Management System (DBMS). Una pieza especial de software para administrar bases de datos. El DBMS organiza y almacena los datos. Mediace los accesos y los cambios en la base de datos.

1.1. Introducción

Las bases de datos relacionales facilitan evitar la duplicidad de información e inconsistencias de datos. La mayoría de los sistemas de bases de datos utilizados hoy en día son relacionales.

En el modelo relacional, los datos se dividen en diferentes **tables**. Una tabla funciona como una matriz o una hoja de cálculo.

Normalmente, las columnas imponen un tipo de datos que pueden contener. Las columnas también pueden especificar otras restricciones: si es obligatorio que las filas tengan un valor en esa columna, si el valor de la columna debe ser único en todas las filas y cosas así.

Las columnas se denominan más comúnmente atributos. Si una columna sólo permite números enteros, decimos que es un campo entero. Diferentes tablas utilizan diferentes tipos de campos.

La organización de una tabla de base de datos viene dada por sus campos y las restricciones que imponen.

Todas las entradas de datos son filas y el sistema de base de datos no aceptará una fila en una tabla si viola el esquema de la tabla. Esa es una gran limitación del modelo relacional. Cuando las características de los datos varían demasiado, la adaptación de los datos a un esquema fijo puede ser problemática. Pero si está trabajando con datos de estructura homogénea, un esquema fijo le ayudará a garantizar que los datos son válidos.

1.1.1. Relaciones

Para evitar la redundancia de datos la clave esta en crear diversas tablas e ir relacionandolas.

Para lograr esto tenemos que crear relaciones, y es imposible hablar de relaciones sin hablar de llaves:

- **Primary Key:** La idea de una llave primaria es crear un atributo que tiene, tiene pero tiene que ser único y es recomendable que casi no cambie.
- **Foreign Key:** La idea de una llave secundaria es crear un atributo que tiene que alguna otra tabla sea la llave primaria. Eso es todo.

1.2. SQL como Lenguaje

SQL **no** es tan potente como una máquina universal de Turing.

Es decir, hay algunos cálculos que son posibles utilizando un lenguaje de programación de propósito general, pero no son posibles con SQL. SQL también no admite acciones como la entrada de usuarios, la salida a las pantallas ó la comunicación a través de la red. Dichas computaciones y acciones deben escribirse en un lenguaje principal (C, C++ ó Java) con consultas SQL incorporadas que acceden a los datos de la base de datos

Los programas de aplicación son programas que se utilizan para interactuar con la base de datos de esta manera.

2. Parte Practica: Practica 1

Veamos por pasos que es lo que hicimos:

■ Crear la Base de Datos:

```
CREATE DATABASE P1;
USE DATABASE P1;
```

■ Construimos las Relaciones Propietarias:

```
CREATE TABLE TrabajoTerminal (
  NumeroDeTrabajoTerminal INT NOT NULL PRIMARY KEY,
  TituloDelTrabajo VARCHAR(80)
);

CREATE TABLE Departamento (
  IDDepartamento INT NOT NULL PRIMARY KEY,
  Nombre VARCHAR(50)
);

CREATE TABLE Presentacion (
  IDPresentacion INT NOT NULL PRIMARY KEY,
  Fecha date,
  CalificacionSeguimiento INT,
  CalificacionSinodales INT,
  Tipo VARCHAR(8)
);

CREATE TABLE Profesor (
  IDProfesor INT NOT NULL PRIMARY KEY,
  IDDepartamento INT,
  Nombre VARCHAR(10),
  ApellidoPaterno VARCHAR(10),
  ApellidoMaterno VARCHAR(10),
  Academia VARCHAR(20),
  Salario REAL,

  FOREIGN KEY(IDDepartamento)
    REFERENCES Departamento(IDDepartamento)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

CREATE TABLE DirigeTrabajoTerminal (
  NumeroTrabajo INT NOT NULL,
  IDProfesor INT NOT NULL,

  PRIMARY KEY(NumeroTrabajo, IDProfesor),

  FOREIGN KEY(NumeroTrabajo)
    REFERENCES TrabajoTerminal(NumeroDeTrabajoTerminal)
    ON DELETE CASCADE
    ON UPDATE CASCADE,

  FOREIGN KEY(IDProfesor)
    REFERENCES Profesor(IDProfesor)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

■ Alteramos las Relaciones Es decir, vamos a hacer lo siguiente:

- Renombrar las Relación Profesor a Catedratico
- Agregar un campo a la relación Presentacion
- Cambiar el Nombre de Departamento
- Agregar telefono del Catedratico
- Cambiar el tipo de dato del Tel. del Catedratico

```
ALTER TABLE Profesor RENAME AS Catedratico;  
  
ALTER TABLE Presentacion ADD COLUMN  
    Dictamen VARCHAR(10);  
  
ALTER TABLE Departamento CHANGE COLUMN  
    Nombre NombreDelDepartamento VARCHAR(50);  
  
ALTER TABLE Catedratico ADD COLUMN  
    Telefono INT;  
  
ALTER TABLE Catedratico MODIFY COLUMN  
    Telefono VARCHAR(15);
```

■ Agregar Foreign Key:

```
ALTER TABLE Presentacion ADD COLUMN NumeroTrabajo INT;  
  
ALTER TABLE Presentacion ADD  
    FOREIGN KEY(NumeroTrabajo)  
        REFERENCES TrabajoTerminal(NumeroDeTrabajoTerminal)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE;
```

■ Cambiar la definición de la PK en Presentación:

```
ALTER TABLE Presentacion DROP PRIMARY KEY;  
  
ALTER TABLE Presentacion ADD  
    PRIMARY KEY(IDPresentacion , Fecha);
```

2.1. Evidencias

```
mysql> CREATE DATABASE P1;  
Query OK, 1 row affected (0.00 sec)  
  
mysql>
```

Figura 1: Empecemos por crear la BD

```
mysql> CREATE TABLE Departamento (  
-> IDDepartamento INT NOT NULL PRIMARY KEY,  
-> Nombre VARCHAR(50)  
-> );  
Query OK, 0 rows affected (0.08 sec)  
  
mysql>  
mysql> CREATE TABLE Presentacion (  
-> idPresentacion INT NOT NULL PRIMARY KEY,  
-> Fecha date,  
-> CalificacionSeguimiento INT,  
-> CalificacionSinodales INT,  
-> Tipo VARCHAR(8)  
-> );  
Query OK, 0 rows affected (0.11 sec)
```

Figura 2: Creando tables

```
mysql> CREATE TABLE TrabajoTerminal (  
-> NumeroDeTrabajoTerminal INT NOT NULL PRIMARY KEY,  
-> TituloDelTrabajo VARCHAR(80)  
-> );  
Query OK, 0 rows affected (0.07 sec)
```

Figura 3: Creando tables

```
mysql> CREATE TABLE Profesor (  
-> IDProfesor INT NOT NULL PRIMARY KEY,  
-> IDDepartamento INT,  
-> Nombre VARCHAR(10),  
-> ApellidoPaterno VARCHAR(10),  
-> ApellidoMaterno VARCHAR(10),  
-> ACADEMIA VARCHAR(20),  
-> Salario REAL,  
->  
-> FOREIGN KEY(IDDepartamento)  
-> REFERENCES Departamento(IDDepartamento)  
-> ON DELETE CASCADE  
-> ON UPDATE CASCADE  
-> );  
Query OK, 0 rows affected (0.08 sec)  
  
mysql> █
```

Figura 4: Creando tables

```
mysql>  
mysql> CREATE TABLE DirigeTrabajoTerminal (  
-> NumeroTrabajo INT NOT NULL,  
-> IDProfesor INT NOT NULL,  
->  
-> PRIMARY KEY(NumeroTrabajo, IDProfesor),  
->  
-> FOREIGN KEY(NumeroTrabajo)  
-> REFERENCES TrabajoTerminal(NumeroDeTrabajoTerminal)  
-> ON DELETE CASCADE  
-> ON UPDATE CASCADE,  
->  
-> FOREIGN KEY(IDProfesor)  
-> REFERENCES Profesor(IDProfesor)  
-> ON DELETE CASCADE  
-> ON UPDATE CASCADE  
-> );  
Query OK, 0 rows affected (0.07 sec)  
  
mysql>
```

Figura 5: Creando tables

```
mysql> ALTER TABLE Profesor RENAME AS Catedratico;
Query OK, 0 rows affected (0.04 sec)

mysql> █
```

Figura 6: Modificandolas

```
mysql> ALTER TABLE Profesor RENAME AS Catedratico;
Query OK, 0 rows affected (0.04 sec)

mysql> ALTER TABLE Presentacion ADD COLUMN Dictamen VARCHAR(10);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figura 7: Modificandolas

```
+-----+
| Tables_in_p1 |
+-----+
| Catedratico   |
| Departamento |
| DirigeTrabajoTerminal |
| Presentacion  |
| TrabajoTerminal |
+-----+
5 rows in set (0.00 sec)
```

Figura 8: Veamos las tablas que tenemos

```
mysql> ALTER TABLE Presentacion ADD
[   -> FOREIGN KEY (NumeroTrabajo)
[   -> REFERENCES TrabajoTerminal(NumeroDeTrabajoTerminal)
[   -> ON DELETE CASCADE
[   -> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> █
```

Figura 9: Modificandolas


```
mysql> ALTER TABLE Presentacion DROP PRIMARY KEY;
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE Presentacion ADD
->
Display all 802 possibilities? (y or n)
-> PRIMARY KEY(IDPresentacion, Fecha);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figura 10: Modificamos las Key

3. Parte Practica: Practica 2

Veamos por pasos que es lo que hicimos:

- Crear las relaciones del nuevo ejercicio:

```
CREATE TABLE Empleado (
  IDEmpleado INT NOT NULL PRIMARY KEY,
  Nombre VARCHAR(20),
  Direccion VARCHAR(300),
  Telefono int,
  Genero VARCHAR(10)
);

CREATE TABLE Cinemex (
  IDCine INT NOT NULL PRIMARY KEY,
  Nombre VARCHAR(30),
  Direccion VARCHAR(300),
  Telefono int,
  email VARCHAR(30)
);

CREATE TABLE EmpleadoCinemex (
  IDEmpleado INT NOT NULL,
  IDCine INT NOT NULL,

  PRIMARY KEY (IDEmpleado, IDCine),

  FOREIGN KEY (IDCine)
    REFERENCES Cinemex(IDCine)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  FOREIGN KEY (IDEmpleado)
    REFERENCES Empleado(IDEmpleado)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

- Modificamos un poco las relaciones:

```
ALTER TABLE Empleado ADD COLUMN Email VARCHAR(30);
ALTER TABLE Empleado ADD COLUMN Salario REAL;
```

- Añadamos la relación gerente:

```
CREATE TABLE Gerente (
  IDGerente INT NOT NULL,
  Nombre VARCHAR(30),
  Turno VARCHAR(20),
  Salario REAL,
  Telefono INT,
  IDCine INT,
  FOREIGN KEY (IDCine)
    REFERENCES Cinemex(IDCine)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

- Modifiquemos un poco las relaciones:

```
ALTER TABLE Gerente MODIFY COLUMN Telefono VARCHAR(20);
ALTER TABLE Empleado RENAME AS Asociado;

ALTER TABLE Asociado MODIFY COLUMN Direccion VARCHAR(350);
```

- Vamos a Cambiar una Llave Primaria Es decir, vamos a hacer lo siguiente:

- Eliminar las LLaves Foraneas de EmpleadoCinemex y Gerente

```
SHOW CREATE TABLE EmpleadoCinemex;
SHOW CREATE TABLE Gerente;
```

- Buscamos el Contraint

```
ALTER TABLE EmpleadoCinemex DROP FOREIGN KEY empleadocinemex_ibfk_1;
SHOW CREATE TABLE EmpleadoCinemex;
```

```
ALTER TABLE Gerente DROP FOREIGN KEY gerente_ibfk_1;
SHOW CREATE TABLE Gerente;
```

- Eliminar la llave primaria y Coloca una nueva llave primaria

```
ALTER TABLE Cinemex DROP PRIMARY KEY;
```

```
ALTER TABLE Cinemex ADD PRIMARY KEY(IDCine, Nombre);
DESC Cinemex;
```

- Añadir campos necesarios y unirlos

```
ALTER TABLE Gerente ADD COLUMN NombreCinemex VARCHAR(30);
ALTER TABLE EmpleadoCinemex ADD COLUMN NombreCinemex VARCHAR(30);
```

```
ALTER TABLE Gerente
ADD FOREIGN KEY(IDCine, NombreCinemex)
REFERENCES Cinemex(IDCine, Nombre)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

```
ALTER TABLE EmpleadoCinemex
ADD FOREIGN KEY(IDCine, NombreCinemex)
REFERENCES Cinemex(IDCine, Nombre)
ON DELETE CASCADE
ON UPDATE CASCADE;
```

- Añadamos otra relación:

```
CREATE TABLE Cartelera (
  IDCartelera INT NOT NULL PRIMARY KEY,
  FechaInicio DATE,
  FechaFin DATE,
  Clasificacion VARCHAR(5),
  IDCine INT,
  NombreCinemex VARCHAR(30),
  FOREIGN KEY(IDCine, NombreCinemex)
  REFERENCES Cinemex(IDCine, Nombre)
  ON DELETE CASCADE
  ON UPDATE CASCADE
);
```

■ Añadamos una Base externa:

```
CREATE DATABASE Sams;  
USE Sams;  
  
SOURCE /Users/mac/Downloads/TareasPendientes/BasesDeDatos/sams.sql
```

■ Veamos la información de dicha base:

```
SELECT * FROM club;  
  
SELECT idclub, nombre FROM club;  
  
SELECT idclub, nombre FROM club  
ORDER BY nombre DESC;  
  
SELECT COUNT(*) AS proveedor FROM proveedor;  
SELECT COUNT(*) AS "numero_de_proveedor" FROM proveedor;  
  
SELECT nombre AS socio FROM socio  
ORDER BY nombre;  
  
SELECT nombre FROM servicio;  
  
SELECT nombre, precioUnitario FROM producto  
WHERE precioUnitario BETWEEN 100 AND 3000  
ORDER BY precioUnitario;  
  
SELECT * FROM Gerente  
ORDER BY Nombre;
```

3.1. Evidencias

```
mysql> CREATE DATABASE P2; USE P2;
Query OK, 1 row affected (0.00 sec)
```

Figura 11: Creamos la base de datos

```
mysql> CREATE TABLE Cinemex (
  ->   IDCine INT NOT NULL PRIMARY KEY,
  ->   Nombre
Display all 764 possibilities? (y or n)
  ->   Nombre VARCHAR(30),
  ->   Direccion VARCHAR(300),
  ->   Telefono int,
  ->   email
Display all 764 possibilities? (y or n)
  ->   email VARCHAR(30)
  -> );
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE TABLE Empleado (
  ->   IDEmpleado INT NOT NULL PRIMARY KEY,
  ->   Nombre
Display all 764 possibilities? (y or n)
  ->   Nombre VARCHAR(20),
  ->   Direccion VARCHAR(300),
  ->   Telefono int,
  ->   Genero
Display all 764 possibilities? (y or n)
  ->   Genero VARCHAR(10)
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE EmpleadoCinemex (
  ->   IDEmpleado INT NOT NULL,
  ->   IDCine INT NOT NULL,
  ->
  ->   PRIMARY KEY (IDEmpleado, IDCine),
  ->
  ->   FOREIGN KEY (IDCine)
  ->   REFERENCES Cinemex(IDCine)
  ->   ON DELETE CASCADE
  ->   ON UPDATE CASCADE,
  ->   FOREIGN KEY (IDEmpleado)
  ->   REFERENCES Empleado(IDEmpleado)
  ->   ON DELETE CASCADE
  ->   ON UPDATE CASCADE
  -> );
Query OK, 0 rows affected (0.06 sec)
```

Figura 12: Creando tables

```
mysql> ALTER TABLE Empleado ADD COLUMN Email VARCHAR(30);
Query OK, 0 rows affected (0.30 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE Empleado ADD COLUMN Salario REAL;
Query OK, 0 rows affected (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figura 13: Modifiquemos un poco las tablas

```
mysql> CREATE TABLE Gerente (
    ->     IDGenerente INT NOT NULL,
    ->     Nombre VARCHAR(30),
    ->     Turno VARCHAR(20),
    ->     Salario REAL,
    ->     Telefeno INT,
    ->     IDCine INT,
    ->     FOREIGN KEY (IDCine)
    ->         REFERENCES Cinemex(IDCine)
    ->         ON DELETE CASCADE
    ->         ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.07 sec)
```

Figura 14: Creando tables

```
mysql> ALTER TABLE Gerente MODIFY COLUMN Telefeno VARCHAR(20);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE Empleado RENAME AS Asociado;
Query OK, 0 rows affected (0.03 sec)

mysql>
mysql> ALTER TABLE Asociado MODIFY COLUMN Direccion VARCHAR(350);
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figura 15: Creando tables

```
mysql> ALTER TABLE EmpleadoCinemex DROP FOREIGN KEY empleadocinemex_ibfk_1;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW CREATE TABLE EmpleadoCinemex;
+-----+
| Table | Create Table Statement |
+-----+
| EmpleadoCinemex | CREATE TABLE `EmpleadoCinemex` (
  `IDCine` int(11) NOT NULL,
  `Nombre` varchar(30) NOT NULL,
  `Direccion` varchar(300) NOT NULL,
  `Telefono` int(11) NOT NULL,
  `email` varchar(30) NOT NULL,
  PRIMARY KEY (`IDCine`, `Nombre`),
  FOREIGN KEY (`IDCine`) REFERENCES `Cinemex` (`IDCine`),
  FOREIGN KEY (`Nombre`) REFERENCES `Cinemex` (`Nombre`)
) ENGINE=InnoDB
```

Figura 16: Eliminar keys

```
mysql> ALTER TABLE Cinemex DROP PRIMARY KEY;
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> ALTER TABLE Cinemex ADD PRIMARY KEY(IDCine, Nombre);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESC Cinemex;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| IDCine | int(11) | NO | PRI | NULL |
| Nombre | varchar(30) | NO | PRI | NULL |
| Direccion | varchar(300) | YES | | NULL |
| Telefono | int(11) | YES | | NULL |
| email | varchar(30) | YES | | NULL |
+-----+
5 rows in set (0.09 sec)
```

Figura 17: Añadimos una nueva llave primaria

```
mysql> ALTER TABLE Gerente ADD COLUMN NombreCinemex VARCHAR(30);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE EmpleadoCinemex ADD COLUMN NombreCinemex VARCHAR(30);
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE Gerente
-> ADD FOREIGN KEY(IDCine, NombreCinemex)
-> REFERENCES Cinemex(IDCine, Nombre)
-> ON DELETE CASCADE
-> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.11 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE EmpleadoCinemex
-> ADD FOREIGN KEY(IDCine, NombreCinemex)
-> REFERENCES Cinemex(IDCine, Nombre)
-> ON DELETE CASCADE
-> ON UPDATE CASCADE;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figura 18: Las volvemos a unir

```
mysql> CREATE DATABASE Sams;
Query OK, 1 row affected (0.00 sec)

mysql> USE DATABASE Sams;
ERROR 1049 (42000): Unknown database 'database'
mysql> USE Sams;
Database changed
mysql> SOURCE /Users/mac/Downloads/TareasPendientes/BasesDeDatos/sams.sql
```

Figura 19: Creamos la nueva base y llenamos datos


```
mysql> SELECT * FROM club;
```

idClub	nombre	direccion	tel	idEstado
1	Aguascalientes	Av. Independencia No. 2351, Col. La Concepción, C.P.20118	Tel.: (449) 10 70 00	1
2	Santa Monica	Av. Siglo XXI No. 3906, Col. Rancho Santa Monica, C.P.20209, Aguascalientes.	Tel.: (449) 918-4000/918-4214	1
3	Macro Plaza Tijuana	Blvd. Insurgentes No. 10815, Col. La Mesa C.P.22185	Tel.: (664) 9695834	2
4	Mexicali	Blvd. Lázaro Cárdenas 1801, Col. Ex ejido de Zacatecas C.P.21998	Tel.: (666) 552-141-48	2
5	Tijuana	Blvd. 5 miles Toluca 4805, Zona Río, C.P.22250	Tel.: (664) 16-34 40-70	2
6	Los Cabos	Carretera Transpeninsular Km.2, Los Cabos-La Paz Previa La Huerta, C.P.23463, Cabo San Lucas.	Tel.: (624) 144 71 24	3
7	Cola de Ballena	Av. Agustín Blázquez esp. Libramiento Sur s/n, Col. El Zapatal, La Paz.	Tel.: (961) 616 6366	3
8	Camacho	Maria Lavalle Hermana D Lote 1, Fraccionamiento 2, Col. Sector Andadores de A-Kia-poch, C.P.24828	Tel.: (961) 616 6366	4
9	Ciudad del Carmen	Av.Periférico Norte No.52, Col. Asa Poniente, C.P.24118	Tel.: (963) 286 1222	4
10	Saltillo	Av. Mariano Ortiz Garza No. 841, col. saltillo 400, C.P.25206	Tel.: (844) 436-18-60	5
11	Torresón	Blvd. Diagonal Reforma No. 3801 Dte. col. San Marcos, C.P.27848	Tel.: (877) 7-32-11-01	5
12	Colima	Felipe Sevilla del Río No.49, Col. Vista Hermosa, C.P.28816	Tel.: (312) 323-51-72	6
13	Coahuila	Blvd. de las Federaciones Re 128 + 200, col. Orlizano, C.P.	Tel.: (963) 632 94 94	7
14	San Cristóbal	Eje vial No.1 esp. Niño de Atocha, col. El Relicario, C.P.29206, San Cristóbal de las Casas	MBL	7
15	Tapachula	Carretera a Puerto Naderá Km 3.5, col. Valle Dorado II, C.P.30708	Tel.: (962) 628-62880 al 80	7
16	Tuxtla Gutiérrez	Blvd. Belisario Dominguez No.1550, Col. Jardines de Tuxtla, C.P.29020	Tel.: (961) 615 7720	7
17	Ciudad Juárez	Av.Ejército Nacional No. 7445, Col. Partido Iglesias, C.P.32399	Tel.: (656) 629-60-67	8
18	Chihuahua	Periférico de la Juventud No. 2280, Col. Haciendas del Valle, C.P.31238	Tel.: (614) 429 13 00	8
19	Monterrey	Av. Numero No. 700, Col. Real Universidad, C.P. 31225	MBL	9
20	Acapulco	Calzada Acapulco No. 430, Col. Ex-Hacienda Copala, C.P.14300	Tel.: (55) 56-79-32-51/56-79-31-07	9
21	Plaza Oriente	Av. Canal de Teconile No. 1520, Col. Alfonso Ortiz Tirado, C.P. 89020	Tel.: (55) 5000 5211	9
22	Panamá	Av. Ejército Nacional No. 559, col. Granada, C.P.31326	Tel.: (55) 25-46-63-08	9
23	San Nicolás	Av. San Nicolás No. 838, Col. la Rivera Novato, C.P. 81100	Tel.: (55) 25-25-25-25	9

```
mysql> SELECT idCLUB, nombre FROM club;
```

idCLUB	nombre
1	Aguascalientes
2	Santa Mónica
3	Macro Plaza Tijuana
4	Mexicali
5	Tijuana
6	Los Cabos
7	Cola de Ballena

```
mysql> SELECT idclub, nombre FROM club
-> ORDER BY nombre DESC;
```

idclub	nombre
59	Zumpango
108	Zacatecas
105	Xalapa
76	Via Capu
94	Villahermosa
29	Villa Jardin
104	Veracruz
61	Uruapan
70	Universidad Mty
27	Universidad DE

Figura 20: Veamos los datos

```
mysql> SELECT COUNT(*) AS proveedor FROM proveedor;
+-----+
| proveedor |
+-----+
|          7 |
+-----+
1 row in set (0.04 sec)

mysql> SELECT COUNT(*) AS "numero de proveedor" FROM proveedor;
+-----+
| numero de proveedor |
+-----+
|          7 |
+-----+
1 row in set (0.00 sec)
```

```
mysql>
mysql> SELECT nombre FROM servicio;
+-----+
| nombre |
+-----+
| Farmacia |
| Foto Centro |
| Joyeria |
| Optica |
| TMA Centro Llantero |
| Service Deli |
| BWM Banco Walt-Mart |
| Punta del Cielo |
| Servicio a Domicilio |
| Apple shop |
+-----+
10 rows in set (0.00 sec)
```

```
mysql> SELECT nombre AS socio FROM socio
-> ORDER BY nombre;
+-----+
| socio |
+-----+
| AGUILA PEREZ JUAN CARLOS |
| ALVARADO BAUTISTA ALBERTO ANGEL |
| ANGELES DE LA CRUZ ANDREE MICHEL IRVING |
| ANGELES RODRIGUEZ EDUARDO NERI |
| ARMENDARIZ CRUZ OCTAVIO ARTURO |
| ARVIZU MENDOZA AGUSTIN URIEL |
| AVILA COLIN NURIA DEL CARMEN |
| AYALA HERNANDEZ EDUARDO |
| BAUTISTA EUGENIO CESAR |
| BAUTISTA SAN MARTIN SAUL ALEJANDRO |
| BECERRA ABARCA ALBERTO ISAIAS |
| BLANCO ALVAREZ BRENDA |
| BOCANEGRA SOSA JOSE DE JESUS |
| BONILLA SANCHEZ MAURICIO EDUARDO |
| BRAVO ALVAREZ EDGAR |
```

Figura 21: Veamos los datos

```
mysql> SELECT nombre, precioUnitario FROM producto
-> WHERE precioUnitario BETWEEN 100 AND 3000
-> ORDER BY precioUnitario;
```

nombre	precioUnitario
Horno Microondas AGE1072	1349
Horno Microondas AGE107W	1999
LG Celular GSM ME970	2322.42

3 rows in set (0.00 sec)

```
mysql> SELECT * FROM Gerente
-> ORDER BY Nombre;
```

idGerente	nombre	idClub
242	ACEVES RESENDIZ JOSE EDUARDO	33
238	ACOSTA CAUDILLO LIDIA	29
243	ACOSTA GOMEZ ALEJANDRO	34
167	AGUILERA LUNA ORESTES JOB	66
272	AGUIRRE ENCISO GANDHI	63
101	AGUIRRE MARTINEZ ISRAEL	1
202	ALVARADO GARCIA ALBERTO YUSSEL	101
203	ANDRADE OLIVARES FRANK	102
273	ANGELES GONZALEZ MARIANA	64
274	ARROYO MENDOZA DANIEL	65
115	BALLESTEROS HERNANDEZ YOSELIN ANDREA	15
116	BARRIENTOS AGUILAR MARCO VINICIO	16
102	BARUCH BALDERRAMA ANGEL	2
275	BASURTO ESQUIVEL YAEL	66
276	BASURTO SEGOVIA GABRIELA	67
244	BAUTISTA MORA LUIS ALBERTO	35
204	BAUTISTA SANCHEZ ALBERTO DAVID	103
277	BERNAL SALGADO ALBERTO	68

Figura 22: Veamos los datos

4. Conclusiones

Gracias a esta practica pudimos comprender mucho mejor como es que funcionan las bases de datos y lo facil que puede llegar a ser manipularlas para usarlas en sistemas computacionales.

Aprendimos sobre el modelo relacional y como podemos separar nuestra informacion gracias a las tablas. Vimos como es que podemos volver a unir gracias a las relaciones, el uso de llaves primarias y secundarias.

Referencias

- [1] Databases, Liberty Hall Chichester 1999 Bob Hudson
- [2] Computer Science Distilled,