



Escuela Superior de Computo

Grupo: 2CV2

Bases de datos

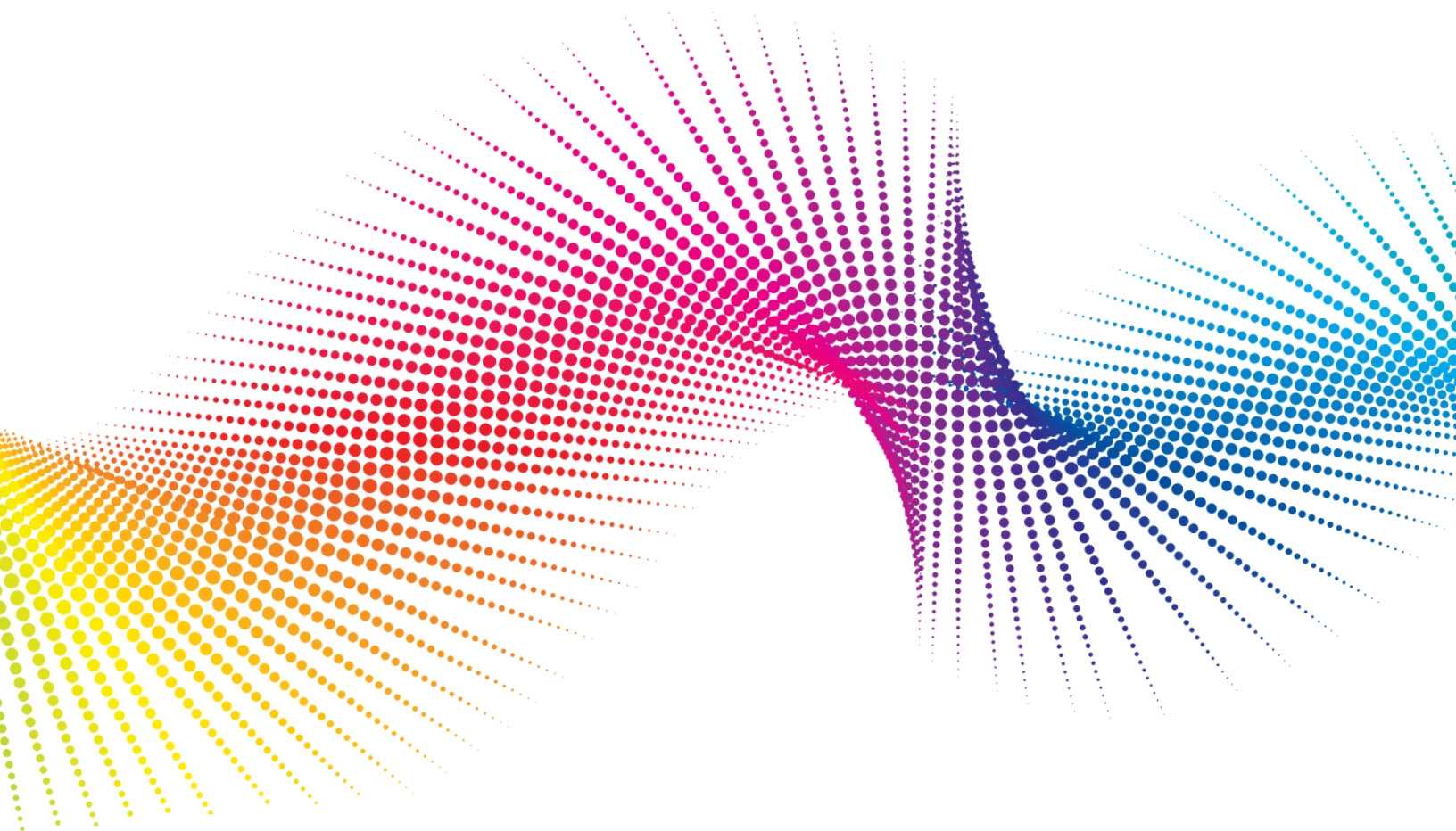
M. en C. Euler Hernández Contreras

1° Parcial

Viernes 16 de febrero de 2018

Reporte de 1° practica

Alumno: Aaron Antonio Garcia Gonzalez



Índice

Marco Teórico	3
Definición de base de datos	3
Sistema Gestor de Bases de Datos	3
SQL	4
Conceptos:.....	5
Tipos de datos usados:	5
Instrucciones:	6
Capturas de pantalla:	8
Conclusiones	17
Referencias.....	17

Marco Teórico

Definición de base de datos

Una base de datos es un conjunto de datos que se encuentran organizados, estructurados y almacenados para su acceso y manipulación.

Características

Entre las principales características de los sistemas de base de datos podemos mencionar:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar.

Sistema Gestor de Bases de Datos

Sistema de administración de bases de datos: consiste en un conjunto de programas utilizados para definir, administrar y procesar una base de datos y sus aplicaciones. A los sistemas de administración de bases de datos también se les llama Sistemas de Gestión de Bases de Datos (SGBD). Un sistema de administración de bases de datos es una herramienta de propósito general que permite crear bases de datos de cualquier tamaño y complejidad y con propósitos específicos distintos.

El objetivo principal de un sistema de administración de bases de datos es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente. Los SGBD se diseñan para gestionar grandes cantidades de información. La gestión de los datos implica tanto la definición de estructuras para almacenar la información como la provisión de mecanismos para la manipulación de la información. Además, los sistemas de bases de datos deben proporcionar la fiabilidad de la información almacenada, a pesar de las caídas del sistema o los intentos de acceso sin autorización. Si los datos van a ser compartidos entre varios usuarios, el sistema debe evitar posibles datos contradictorios.

Un sistema de base de datos tiene los siguientes componentes:

1. Aplicaciones:
Es la interfaz entre la base de datos y el usuario; estas pueden ser desarrolladas por un lenguaje de alto nivel (aplicaciones desarrolladas en VBasic, Java, C++, C#) y/o aplicaciones desarrolladas en un lenguaje semiestructurado (javascript, php, asp).
2. Conectores (*drivers*):
Son los componentes que permiten el enlace entre el SGBD y las interfaces desarrolladas en un lenguaje de programación; éstas contienen las clases y/o funciones necesarias para llevar a cabo la comunicación entre las aplicaciones con el Sistema Gestor de Base de Datos.
3. Sistema Gestor de Base de Datos (SGBD):
Es la aplicación que permite a los usuarios definir, crear y mantener la base de datos y proporciona acceso controlado a la misma.
4. Repositorio de datos:
Que esta conformado por los objetos de la base de datos:
 - a) Datos:

un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla. En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas o campos. Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.

Un registro es un conjunto de campos que contienen los datos que pertenecen a una misma repetición de entidad. Se le asigna automáticamente un número consecutivo (número de registro) que en ocasiones es usado como índice, aunque lo normal y práctico es asignarle a cada registro un campo clave para su búsqueda.

b) Metadatos:

Datos que describen otros datos o "datos sobre datos". De forma general, en efecto, el concepto de metadatos se refiere a aquellos datos que hablan de los datos, es decir, describen el contenido de los archivos o la información de los mismos.

c) Índices:

El índice de una base de datos es una estructura de datos que mejora la velocidad de las operaciones, por medio de identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros de una tabla en una base de datos. Los índices pueden ser creados usando una o más columnas, proporcionando la base tanto para búsquedas rápidas al azar como de un ordenado acceso a registros eficiente.

d) Bitácoras:

Permite guardar las transacciones realizadas sobre una base de datos en específico, de tal manera que estas transacciones puedan ser auditadas y analizadas posteriormente. Pueden obtenerse datos específicos de la transacción como son la Operación que se realizó, el Usuario de BD y Windows, Fecha, Maquina y Programa.

e) Funciones:

Archivos que facilitan operaciones dentro de la base de datos, pueden ser definidos por el usuario.

SQL

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje específico del dominio que da acceso a un sistema de gestión de bases de datos relacionales que permite especificar diversos tipos de operaciones en ellos. Una de sus características es el manejo del álgebra y el cálculo relacional que permiten efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como hacer cambios en ellas.

Características generales de SQL

SQL es un lenguaje de acceso a bases de datos que explota la flexibilidad y potencia de los sistemas relacionales y permite así gran variedad de operaciones.

Es un lenguaje declarativo de "alto nivel" o "de no procedimiento" que, gracias a su fuerte base teórica y su orientación al manejo de conjuntos de registros y no a registros individuales permite una alta productividad en codificación y la orientación a objetos. De esta forma, una sola sentencia puede equivaler a uno o más programas que se utilizarían en un lenguaje de bajo nivel orientado a registros.

SQL también tiene las siguientes características:

- Lenguaje de definición de datos: El LDD de SQL proporciona comandos para la definición de esquemas de relación, borrado de relaciones y modificaciones de los esquemas de relación.
- Lenguaje interactivo de manipulación de datos: El LMD de SQL incluye lenguajes de consultas basado tanto en álgebra relacional como en cálculo relacional de tuplas.
- Integridad: El LDD de SQL incluye comandos para especificar las restricciones de integridad que deben cumplir los datos almacenados en la base de datos.
- Definición de vistas: El LDD incluye comandos para definir las vistas.
- Control de transacciones: SQL tiene comandos para especificar el comienzo y el final de una transacción.
- SQL incorporado y dinámico: Esto quiere decir que se pueden incorporar instrucciones de SQL en lenguajes de programación como: C++, C, Java, PHP, Cobol, Pascal y Fortran.
- Autorización: El LDD incluye comandos para especificar los derechos de acceso a las relaciones y a las vistas.

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base datos de código abierto más popular del mundo,¹² y una de las más populares en general junto a Oracle y Microsoft SQL Server, sobre todo para entornos de desarrollo web.

Conceptos:

Tabla: es el nombre que recibe cada una de las relaciones que se establecen entre los datos almacenados; cada nueva relación da lugar a una tabla. Están formadas por filas, también llamadas tuplas, donde se describen los elementos que configuran la tabla (es decir, los elementos de la relación establecida por la tabla), columnas o campos, con los atributos y valores correspondientes, y el dominio (tipo de dato), concepto que agrupa a todos los valores que pueden figurar en cada columna.

Llaves: elementos que impiden la duplicidad de registros, una de las grandes desventajas que presentan otros modelos de organización y gestión de bases de datos.

Llaves primarias: son los atributos según el tipo de relación que se ha definido en la tabla. Pueden añadirse otros atributos específicos y propios.

Llaves foráneas: son las llaves que se definen para cada una de las llaves primarias establecidas para los elementos o entidades de una relación.

Tipos de datos usados:

Varchar: Tipo de datos que admite caracteres alfanuméricos. Su uso es similar a Char(x). A la hora de definir un campo de datos Varchar deberemos especificar el número máximo de caracteres que podrá aceptar en la entrada de datos, donde x es un número entre 1-255. A diferencia de Char, este tipo de datos es variable en su longitud, admitiendo entradas inferiores a la establecida.

Int: Este es un tipo de datos numéricos de tipo entero. Este tipo de datos guarda valores enteros (no decimales) entre -2.147.483.648 y 2.147.483.647.

Double: Número de coma flotante de precisión doble. Es un tipo de datos igual al anterior cuya única diferencia es el rango numérico que abraza, siendo este el comprendido entre 1,7976931348623157E+308 hasta -2,2250738585072014E-308, 0, y 2,2250738585072014E-308 to 1,7976931348623157E+308.

Date: Formato de Fecha. Su representación es en formato de fecha numérica del tipo 'YYYY-MM-DD' (Año con cuatro dígitos, Mes con dos dígitos, día con dos dígitos). Su rango es '1000-01-01' (1 de enero del año 1000, en el cual yo era aún muy pequeño) hasta '9999-12-31' (31 de diciembre del 9999).

Instrucciones:

1. Crear una base de datos practica1
2. Usar la base de datos practica1
3. Crear la relación estado con los atributos:
 - idEdo int not null primary key
 - nombre varchar(30)
4. Crear la relacion proveedor con los atributos:
 - dProv int not null primary key
 - nombre varchar(40)
 - dir varchar(300)
 - webpage varchar(80)
5. Crear la relación miembro llamada producto con los atributos:
 - idProv int not null primary key
 - nombre varchar(50)
 - contNeto varchar(20)
 - pu double
 - idproveedor int
 - foreign key(idproveedor) references proveedor(idProv)
on delete cascade on update cascade
6. Crear la relación miembro llamada sucursal con los atributos:
 - idSuc int not null primary key
 - nombre varchar(30)
 - dir varchar(300)
 - tel int
 - idEdo int
 - foreign key(idEdo) references estado(idEdo)
on delete cascade on update cascade
7. Crear la relacion miembro llamada sp con los atributos:
 - idProv int not null
 - idSuc int not null
 - primary key(idProv,idSuc)
 - fecha date
 - foreign key(idprov) references producto(idProv)
on delete cascade on update cascade
 - foreign key(idSuc) references sucursal(idSuc)
on delete cascade on update cascade
8. Mostrar todas las relaciones
9. Verificamos que existan las llaves foráneas
10. Cambiar el nombre de la relación nombre y ponerle "Club"

11. Cambiar el nombre del campo nombre en la relación proveedor y llamarle “proveedor”
12. Cambiar el tipo de dato en teléfono del club y ponerlo de tipo Varchar
13. Cambiar el tamaño de la dir en proveedor a 350
14. Agregar una columna en la relación proveedor para almacenar su correo electrónico
15. Eliminar el campo el contenido neto del producto
16. Almacenar más de un teléfono por sucursal
 - Eliminar la columna tel de la relación club
17. Crear la nueva relación telsuc con los atributos:
 - idSuc int not null
 - tel varchar(20) not null
 - primary key(idSuc,tel)
 - foreign key(idSuc) references club(idSuc)
on delete cascade on update cascade
18. Eliminar la llave foránea de producto incluyendo el identificador
19. Eliminar la llave primaria de la relación proveedor
20. Crear la llave primaria de proveedor compuesta por idProv y proveedor
21. Agregar una columna proveedor Varchar(40) en la relación producto
22. Agregar la llave foránea en la relación producto
 - alter table producto
 - add foreign key (idproveedor, proveedor)
 - references proveedor(idProv, proveedor)
 - on delete cascade on update cascade

Capturas de pantalla:

```
mysql> create database practica1;
Query OK, 1 row affected (0.00 sec)

mysql> use practica1;
Database changed
mysql> create table estado(
  -> idEdo int not null primary key,
  -> nombre varchar(30)
  -> );
Query OK, 0 rows affected (0.10 sec)

mysql> create table proveedor(
  -> idProv int not null primary key,
  -> nombre varchar(40),
  -> dir varchar(300),
  -> webpage varchar(80)
  -> );
Query OK, 0 rows affected (0.09 sec)

mysql> create table producto(
  -> idProv int not null primary key,
  -> nombre varchar(50),
  -> contNeto varchar(20),
  -> pu double,
  -> idproveedor int,
  -> foreign key(idproveedor) references
  -> proveedor(idProv)
  -> on delete cascade on update cascade
  -> );
Query OK, 0 rows affected (0.11 sec)

mysql> create table sucursal(
  -> idSuc int not null primary key,
  -> nombre varchar(30),
  -> dir varchar(300),
  -> tel int,
  -> idEdo int,
  -> foreign key(idEdo) references
  -> estado(idEdo)
  -> on delete cascade on update cascade
  -> );
Query OK, 0 rows affected (0.13 sec)
```



```
mysql> create table sp(
  -> idProv int not null,
  -> idSuc int not null,
  -> primary key(idProv,idSuc),
  -> fecha date,
  -> foreign key(idprov) references
  -> producto(idProv)
  -> on delete cascade on update cascade,
  -> foreign key(idSuc) references
  -> sucursal(idSuc)
  -> on delete cascade on update cascade
  -> );
```

Query OK, 0 rows affected (0.13 sec)

```
mysql> show tables;
```

```
+-----+
| Tables_in_practica1 |
+-----+
| estado               |
| producto             |
| proveedor            |
| sp                   |
| sucursal             |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> show create table estado;
```

```
+-----+-----+
| Table | Create Table
+-----+-----+
| estado | CREATE TABLE `estado` (
  `idEdo` int(11) NOT NULL,
  `nombre` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`idEdo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> show create table producto;
```

```
+-----+-----+
| Table | Create Table
+-----+-----+
```

```
+-----+
| producto | CREATE TABLE `producto` (  
  `idProv` int(11) NOT NULL,  
  `nombre` varchar(50) DEFAULT NULL,  
  `contNeto` varchar(20) DEFAULT NULL,  
  `pu` double DEFAULT NULL,  
  `idproveedor` int(11) DEFAULT NULL,  
  PRIMARY KEY (`idProv`),  
  KEY `idproveedor` (`idproveedor`),  
  CONSTRAINT `producto_ibfk_1` FOREIGN KEY (`idproveedor`) REFERENCES `proveedor` (`idProv`) ON D  
ELETE CASCADE ON UPDATE CASCADE  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |  
+-----+  
  
+-----+  
1 row in set (0.00 sec)  
  
mysql> show create table proveedor;  
+-----+  
+-----+  
| Table      | Create Table  
  
          |  
+-----+  
+-----+  
+-----+  
| proveedor | CREATE TABLE `proveedor` (  
  `idProv` int(11) NOT NULL,  
  `nombre` varchar(40) DEFAULT NULL,  
  `dir` varchar(300) DEFAULT NULL,  
  `webpage` varchar(80) DEFAULT NULL,  
  PRIMARY KEY (`idProv`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |  
+-----+  
+-----+  
+-----+  
1 row in set (0.00 sec)  
  
mysql> show create table sucursal;  
+-----+  
+-----+  
+-----+  
| Table      | Create Table
```

```

-----+
| sucursal | CREATE TABLE `sucursal` (
  `idSuc` int(11) NOT NULL,
  `nombre` varchar(30) DEFAULT NULL,
  `dir` varchar(300) DEFAULT NULL,
  `tel` int(11) DEFAULT NULL,
  `idEdo` int(11) DEFAULT NULL,
  PRIMARY KEY (`idSuc`),
  KEY `idEdo` (`idEdo`),
  CONSTRAINT `sucursal_ibfk_1` FOREIGN KEY (`idEdo`) REFERENCES `estado` (`idEdo`) ON DELETE CASC
ADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+

```

```

-----+
1 row in set (0.00 sec)

```

```
mysql> show create table sp;
```

```

+-----+
| Table | Create Table

```

```

-----+
| sp      | CREATE TABLE `sp` (
  `idProv` int(11) NOT NULL,
  `idSuc` int(11) NOT NULL,
  `fecha` date DEFAULT NULL,
  PRIMARY KEY (`idProv`,`idSuc`),
  KEY `idSuc` (`idSuc`),
  CONSTRAINT `sp_ibfk_1` FOREIGN KEY (`idProv`) REFERENCES `producto` (`idProv`) ON DELETE CASCAD
E ON UPDATE CASCADE,
  CONSTRAINT `sp_ibfk_2` FOREIGN KEY (`idSuc`) REFERENCES `sucursal` (`idSuc`) ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+

```

```
mysql> alter table sucursal RENAME AS club;
Query OK, 0 rows affected (0.06 sec)

mysql> show tables;
+-----+
| Tables_in_practica1 |
+-----+
| club                 |
| estado               |
| producto             |
| proveedor            |
| sp                   |
+-----+
5 rows in set (0.00 sec)

mysql> alter table proveedor CHANGE COLUMN nombre proveedor varchar(40);
Query OK, 0 rows affected (0.16 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc proveedor;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv     | int(11)       | NO   | PRI | NULL    |       |
| proveedor  | varchar(40)   | YES  |     | NULL    |       |
| dir        | varchar(300)  | YES  |     | NULL    |       |
| webpage    | varchar(80)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.04 sec)

mysql> alter table club MODIFY COLUMN tel varchar(20);
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc club;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idSuc      | int(11)       | NO   | PRI | NULL    |       |
| nombre     | varchar(30)   | YES  |     | NULL    |       |
| dir        | varchar(300)  | YES  |     | NULL    |       |
| tel        | varchar(20)   | YES  |     | NULL    |       |
| idEdo      | int(11)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql> alter table proveedor MODIFY COLUMN dir varchar(350);
Query OK, 0 rows affected (0.13 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc proveedor;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv | int(11) | NO | PRI | NULL | |
| proveedor | varchar(40) | YES | | NULL | |
| dir | varchar(350) | YES | | NULL | |
| webpage | varchar(80) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

```
mysql> alter table proveedor ADD COLUMN email varchar(60);
Query OK, 0 rows affected (0.20 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc proveedor;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv | int(11) | NO | PRI | NULL | |
| proveedor | varchar(40) | YES | | NULL | |
| dir | varchar(350) | YES | | NULL | |
| webpage | varchar(80) | YES | | NULL | |
| email | varchar(60) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

```
mysql> alter table producto DROP COLUMN contNeto;
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc producto;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv | int(11) | NO | PRI | NULL | |
| nombre | varchar(50) | YES | | NULL | |
| pu | double | YES | | NULL | |
| idproveedor | int(11) | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

```
mysql> alter table club drop COLUMN tel;
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create table telsuc(
->     idSuc int not null,
->     tel varchar(20) not null,
->     primary key(idSuc,tel),
->     foreign key(idSuc) references
->     club(idSuc)
->     on delete cascade on update cascade
-> );
Query OK, 0 rows affected (0.13 sec)

mysql>
mysql> desc telsuc;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idSuc | int(11)       | NO   | PRI | NULL    |       |
| tel   | varchar(20)   | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.02 sec)

mysql> alter table producto drop foreign key producto_ibfk_1;
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc producto;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv         | int(11)       | NO   | PRI | NULL    |       |
| nombre        | varchar(50)   | YES  |     | NULL    |       |
| pu             | double        | YES  |     | NULL    |       |
| idproveedor    | int(11)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.02 sec)
```

```
mysql> show create table producto;
```

```
+-----+-----+
| Table | Create Table
+-----+-----+
| producto | CREATE TABLE `producto` (
  `idProv` int(11) NOT NULL,
  `nombre` varchar(50) DEFAULT NULL,
  `pu` double DEFAULT NULL,
  `idproveedor` int(11) DEFAULT NULL,
  PRIMARY KEY (`idProv`),
  KEY `idproveedor` (`idproveedor`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> alter table proveedor DROP PRIMARY KEY;
```

```
Query OK, 0 rows affected (0.17 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc proveedor;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv | int(11) | NO | | NULL | |
| proveedor | varchar(40) | YES | | NULL | |
| dir | varchar(350) | YES | | NULL | |
| webpage | varchar(80) | YES | | NULL | |
| email | varchar(60) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

```
mysql> alter table proveedor ADD primary key(idProv, proveedor);
```

```
Query OK, 0 rows affected (0.15 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc proveedor;
```

```
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProv | int(11) | NO | PRI | NULL | |
| proveedor | varchar(40) | NO | PRI | NULL | |
| dir | varchar(350) | YES | | NULL | |
| webpage | varchar(80) | YES | | NULL | |
| email | varchar(60) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

```
mysql> alter table producto ADD COLUMN proveedor varchar(40);
Query OK, 0 rows affected (0.15 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc producto;
```

Field	Type	Null	Key	Default	Extra
idProv	int(11)	NO	PRI	NULL	
nombre	varchar(50)	YES		NULL	
pu	double	YES		NULL	
idproveedor	int(11)	YES	MUL	NULL	
proveedor	varchar(40)	YES		NULL	

5 rows in set (0.02 sec)

```
mysql> alter table producto
-> add foreign key (idproveedor, proveedor)
-> references proveedor(idProv, proveedor)
-> on delete cascade on update cascade;
```

```
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show create table producto;
```

Table	Create Table
producto	<pre>CREATE TABLE `producto` (`idProv` int(11) NOT NULL, `nombre` varchar(50) DEFAULT NULL, `pu` double DEFAULT NULL, `idproveedor` int(11) DEFAULT NULL, `proveedor` varchar(40) DEFAULT NULL, PRIMARY KEY (`idProv`), KEY `idproveedor` (`idproveedor`,`proveedor`), CONSTRAINT `producto_ibfk_1` FOREIGN KEY (`idproveedor`, `proveedor`) REFERENCES `proveedor` (`idProv`, `proveedor`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1</pre>

1 row in set (0.00 sec)

Conclusiones

Para ser mi primer acercamiento en bases de datos, pude diferenciar entre tener datos en bruto y datos de manera estructurada, sé qué fue lo más básico, pero estoy ansioso por seguir adentrándome en este mundo.

Puedo ver que SQL es un lenguaje de bases de datos, en el cual con lo poco que he aprendido permite realizar muchas cosas, es un lenguaje muy natural, prácticamente como lo piensas lo escribes, claro en inglés.

Me impresiono la instrucción “on delete cascade on update cascade” es una instrucción increíblemente útil, espero ponerla en practica y ver en primera fila como afecta a las tuplas.

Referencias

R. Elmasri and S. Navathe, Fundamentos de sistemas de bases de datos. Pearson Educación, 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>