

**TECNOLÓGICO DE ESTUDIOS
SUPERIORES DE VALLE DE BRAVO**

INGENIERÍA EN SISTEMAS COMPUTACIONALES

MATERIA:

Fundamentos de Base de Datos

Manual de prácticas

PROFESOR:

M. EN I.S.C. Mariana Carolyn Cruz Mendoza



Contenido

Unidad 1 INTRODUCCIÓN A LAS BASES DE DATOS	7
Práctica Abogados	9
Práctica Zoológico.....	10
Práctica Náutica.....	12
Unidad 2 DISEÑO DE BASES DE DATOS CON EL MODELO ER.....	14
Base de datos “Estudiantes”	14
Práctica Multiconsultas Estudiantes	21
Base de datos “Abarrotes”	22
Práctica Multiconsultas Abarrotes.....	30
Unidad 3 MODELO RELACIONAL.....	32
Base de datos “Universidad”	32
Práctica Multiconsulta Universidad.....	41
Base de datos “Hospital”	43
Práctica Hospitales.....	50
Unidad 4 NORMALIZACIÓN DE BASES DE DATOS	50
Práctica Normalización.	50
Práctica Cardinalidad	55
Consultas.....	58
Unidad 5 ALGEBRA RELACIONAL.....	62
Práctica Algebra relacional.....	62
Unidad 6 INTRODUCCIÓN A SQL.....	65
Practica Consultas complejas.....	71
Conclusión.	78
Referencias web	78

Tabla de ilustraciones.

Ilustración 1 Figura 1.(TuInstitutoOnline.com: M.Donosó, G.García, P.Gargallo, A.Martínez, 2021).....	8
Ilustración 2 Diagrama	10
Ilustración 3 Diagrama	12

Ilustración 4 Diagrama a mano.....	13
Ilustración 5 Diagrama digital	13
Ilustración 6 Diseño externo	15
Ilustración 7 Modelo entidad relación	15
Ilustración 8 Crear base de datos.....	16
Ilustración 9 Creación de tablas	16
Ilustración 10 Visualización de la creación de la BD	17
Ilustración 11 Visualización de la tabla creada.....	17
Ilustración 12 Inserción de registros en las tablas.....	18
Ilustración 13 inserción en siguiente registro	19
Ilustración 14 Inserción de registros en la tabla estados.....	19
Ilustración 15	20
Ilustración 16 Consulta de tabla municipios y estudiantes.	20
Ilustración 17 Consulta estudiantes ISC.....	21
Ilustración 18 Consulta estado	22
Ilustración 19 Maquetado	23
Ilustración 20 Modelo entidad relación	23
Ilustración 21 Revisar las BD	24
Ilustración 22 Usar BD	24
Ilustración 23 Mensaje de creación de tabla productos.....	25
Ilustración 24 Consulta de tablas de la base de datos.	26
Ilustración 25 Mensaje de creación primary key id_producto.	26
Ilustración 26 Mensaje de creación foreign key de id_categoria	26
Ilustración 27 Mensaje de creación foreign key de id_zona.	27
Ilustración 28 Descripción de la tabla productos.	27
Ilustración 29 Consulta de registros de la tabla de categorías	28
Ilustración 30 Consulta de registros de la tabla de zonas	28
Ilustración 31 Consulta de registro de la tabla productos	29
Ilustración 32 Consulta de la tabla de productos.....	30
Ilustración 33 Consulta de nombre y precio de productos.....	30
Ilustración 34 Consulta de nombre de productos del pasillo.	31

Ilustración 35 Maquetado	34
Ilustración 36 Modelo entidad relación	34
Ilustración 37 Creación de la base de datos	35
Ilustración 38 Usar base de datos	35
Ilustración 39 Visualización de la tabla creada.....	35
Ilustración 40 Inserción de registros en las tablas.....	36
Ilustración 41 Consulta de la tabla personas.....	37
Ilustración 42 Inserción de registros en la tabla publicacioninformes.....	37
Ilustración 43 Tablas creadas en la BD.....	38
Ilustración 44 Consulta de tablas.....	39
Ilustración 45 Consulta de tablas.....	39
Ilustración 46 Consulta 1	40
Ilustración 47 Consulta 2.....	41
Ilustración 48 Consulta 3.....	42
Ilustración 49 Consulta 4.....	42
Ilustración 50 Maquetado	44
Ilustración 51 Modelo entidad relación	44
Ilustración 52 Revisar las BD	45
Ilustración 53 Usar BD	45
Ilustración 54 Mensaje de creación de tabla personas.....	46
Ilustración 55 Consulta de tablas de la base de datos.....	46
Ilustración 56 Mensaje de creación primary key id_producto.....	47
Ilustración 57 Mensaje de creación foreign key.....	47
Ilustración 58 Insertar datos.....	48
Ilustración 59 Consulta de registros de la tabla de categorías	48
Ilustración 60 Descripción de la tabla productos.....	49
Ilustración 61 llaves foranes (relación).....	49
Ilustración 62 Consulta de nombre completo de médicos.....	50
Ilustración 63 Modelo entidad/relación.....	51
Ilustración 64 Base de Datos.....	52
Ilustración 65 Consulta 1	53

Ilustración 66 Consulta 2	53
Ilustración 67 Consulta 3	54
Ilustración 68 Consulta 4	54
Ilustración 69 Consulta 5	54
Ilustración 70 Diagrama E-R	55
Ilustración 71 Diagrama Relacional	55
Ilustración 72 Creación DB	56
Ilustración 73 Llaves Foráneas	56
Ilustración 74 Inserción de datos	56
Ilustración 75 Consulta 1	57
Ilustración 76 Diagrama Relacional	57
Ilustración 77 Llave foránea	58
Ilustración 78 Inserción de datos	58
Ilustración 79 Consulta 1 y 2	58
Ilustración 80 Diagrama E-R	58
Ilustración 81 Diagrama Relacional	59
Ilustración 82 Creación de tablas	59
Ilustración 83 Llaves foráneas	59
Ilustración 84 Inserción de datos	60
Ilustración 85 Consulta 1	60
Ilustración 86 Diagrama E-R	60
Ilustración 87 Diagrama Relacional	61
Ilustración 88 Creación de tablas	61
Ilustración 89 Llaves Foráneas	61
Ilustración 90 Inserción de datos	62
Ilustración 91 Consulta 1	62
Ilustración 92 Consulta 1	62
Ilustración 93 Consulta 2	63
Ilustración 94 Consulta 3	64
Ilustración 95 Consulta 4	64
Ilustración 96 Consulta 5	65

Ilustración 97 Consulta	66
Ilustración 98 Consulta	67
Ilustración 99 Consulta	67
Ilustración 100 Consulta	68
Ilustración 101 Consulta	68
Ilustración 102 Consulta	68
Ilustración 103 Consulta	69
Ilustración 104 Consulta	69
Ilustración 105 Consulta	70
Ilustración 106 Consulta	70
Ilustración 107 Consulta	71
Ilustración 108 Consulta	72
Ilustración 109 Consulta	73
Ilustración 110 Consulta	74
Ilustración 111 Consulta	75
Ilustración 112 Consulta	77

Unidad 1 INTRODUCCIÓN A LAS BASES DE DATOS

Introducción

Se analizará el tipo de relaciones básicas dentro de una la identificación de datos, prácticamente las bases de datos se han vuelto de mayor importancia en todo ámbito laboral y personal, ya que estas facilitan la búsqueda, la gestión y el almacenamiento de diversos datos, prácticamente las bases de datos no tienen una aplicación en la cual se especifica, ya que se pueden aplicar en toda área y dependiendo de las necesidades que se tienen que cubrir con ella.

Antes de armar un modelo de base de datos se tiene que hacer un estudio previo y conocer sus tipos, modelos, su simbología y el modelo en general para poder hacer uso correcto de ella y aplicarla de manera correcta.

Como definición de base de datos se entenderá que se trata de un conjunto de datos interrelacionados y almacenados sin redundancias innecesarias, los cuales sirven a las aplicaciones sin estar relacionados de una manera directa entre ellos. Una base de datos puede ser utilizada por varias aplicaciones y usuarios. (*Introducción a Las Bases de Datos, 2007*)

Modelo entidad relación.

El modelo entidad relación es una herramienta que permite representar de manera simplificada los componentes que participan en un proceso de negocio y el modo en el que estos se relacionan entre sí. (Business, 2018)

Elementos principales.

- Entidades: El modelo contará con una entidad por cada uno de los componentes del proceso de negocio. Así, en un negocio de venta de suscripciones a revistas, podemos tener entidades "Cliente", "Dirección", "Factura", "Producto", o "Incidencias", entre otras. (Business, 2018)
- Atributos: Los atributos, componente fundamental de cada modelo entidad-relación, nos permiten describir las propiedades que tiene cada entidad. "Nombre", "Primer Apellido", "Segundo Apellido", "Fecha de

nacimiento”, “Género” o “Segmento de valor” serán atributos de la entidad “Cliente”.

- Relaciones: Con las relaciones se establecen vínculos entre parejas de entidades. Cada “Cliente” tendrá una “Dirección” de envío en la que recibirá la suscripción, podrá estar suscrito a uno o varios “Productos”, y recibirá una “Factura” con la periodicidad acordada. (Business, 2018)

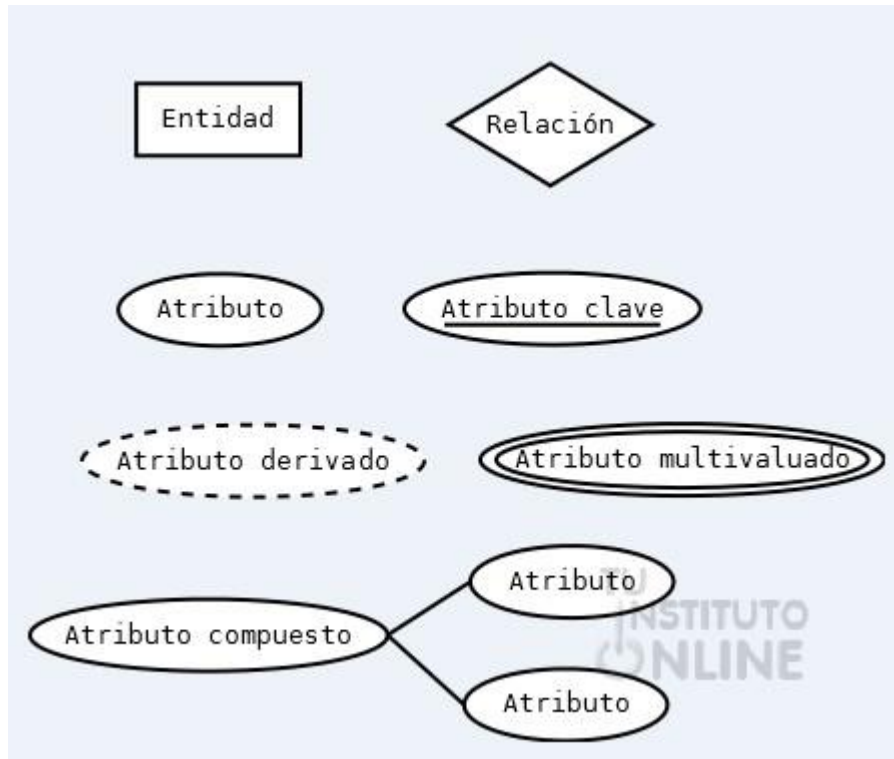


Ilustración 1 Figura 1.(TuInstitutoOnline.com: M.Donosó, G.García, P.Gargallo, A.Martínez, 2021)

Además, es común que, en el modelo entidad-relación, los conectores que indican que dos entidades A y B están relacionadas entre sí tengan una apariencia gráfica diferente dependiendo del tipo de relación que exista entre ellas. (Business, 2018)

Tipos de relaciones.

- Relación uno a uno: Un “individuo” de la entidad A solamente puede estar relacionado con un “individuo” de la entidad B, y ese “individuo” de la entidad B no puede estar relacionado con otros “individuos” de la entidad A. (Business, 2018)

- Relación uno a muchos: Un “individuo” de la entidad A puede estar relacionado con uno o varios “individuos” de la entidad B, y esos “individuos” de la entidad B no pueden estar relacionados con otros “individuos” de la entidad A.
- Relación muchos a muchos.: Cada “individuo” de la entidad A puede estar relacionado con uno o varios “individuos” de la entidad B, y cada “individuo” de la entidad B puede estar relacionado con varios “individuos” de la entidad A. (Business, 2018)

Práctica Abogados

Problemática.

Se requiere diseñar una base de datos relacional para los asuntos que lleva un gabinete de abogados, cada asunto tiene un número de expediente que lo identifica, y corresponde a un solo cliente. Del asunto se debe almacenar el periodo (fecha de inicio y fecha de archivo o finalización), su estado (en trámite, archivado, etc.), así como los datos personales del cliente al que le pertenece (DNI, nombre, dirección, etc.). Algunos asuntos son llevados por uno o varios procuradores de los que nos interesa también los datos personales.

Desarrollo.

1.- Como cualquier problema, tenemos que empezar a leer e identificar las entidades (u objetos) y sus atributos, con las cuales vamos a realizar un diagrama. Asuntos

- Número de expediente
- Periodo (fecha de inicio, fecha de finalización)
- Estado (tramite, archivado) Clientes
- DNI
- Nombre
- Dirección

Procuradores

- DNI
- Nombre
- Dirección

2.- Una vez con la información procedemos a realizar un diagrama especificando las entidades y los atributos los cuales los relacionas por medio de una flecha y una relación declarada.

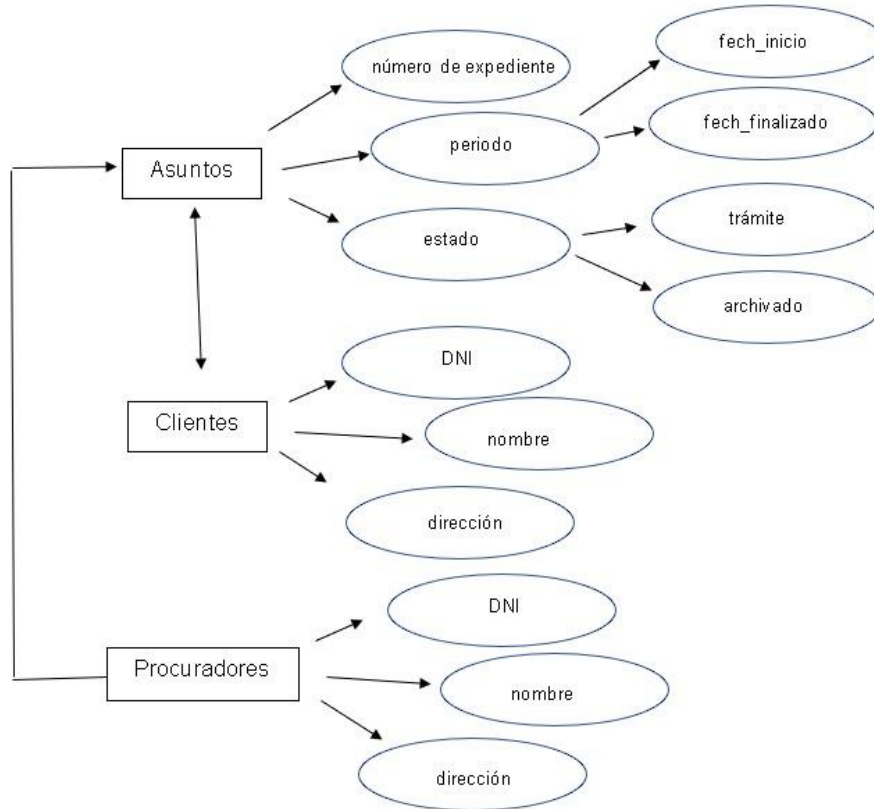


Ilustración 2 Diagrama

Práctica Zoológico

Problemática.

Se quiere diseñar una base de datos relacional que almacene información relativa a los zoos existentes en el mundo, así como las especies animales que estos albergan. De cada zoo se conoce el nombre, ciudad y país donde se encuentra, tamaño (en m²) y presupuesto anual. De cada especie animal se almacena el

nombre vulgar y el nombre científico, familia a la que pertenece y si se encuentra en peligro de extinción. Además, se debe guardar información sobre cada animal que los zoos poseen, como su número de identificación, especie, sexo, año de nacimiento, país de origen y continente.

Desarrollo

1. Identificamos las entidades y respectivamente sus atributos donde se ingresan los datos.

Zoo

- Tamaño (m2)
- País
- Ciudad
- Nombre
- Presupuesto anual Registro animal
- Sexo
- Número de identificación
- Especie
- Continente
- Año de nacimiento
- País de origen

Especie animal

- Nombre científico
- Nombre vulgar
- Familia
- Peligro de extinción

2. Una vez identificado todas las entidades junto a sus atributos procedemos a realizar el diagrama "entidad relación", pero este es diferente al primero, ya que en este tenemos que especificar el tipo de relación (cardinalidad) la cual puedes determinar.

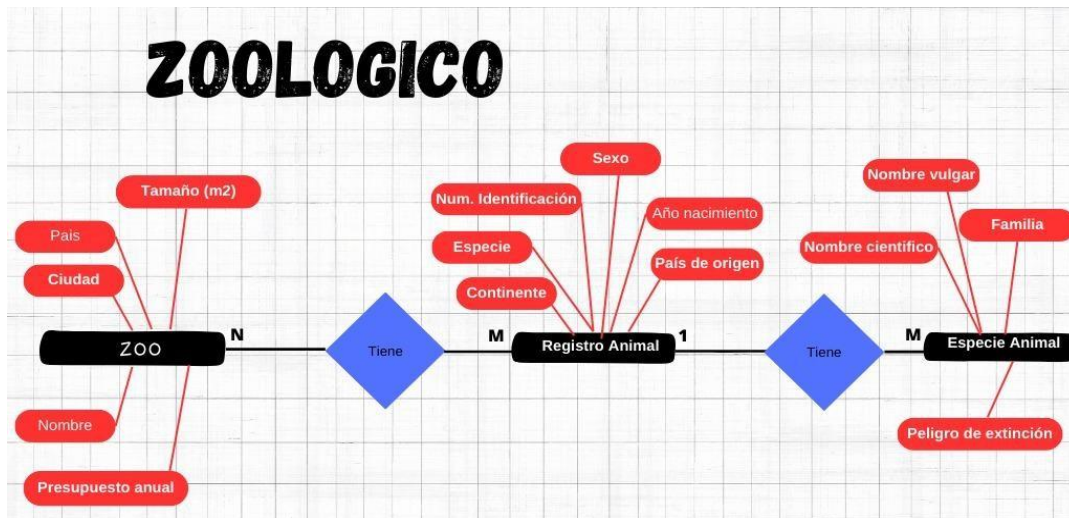


Ilustración 3 Diagrama

Práctica Náutica

Problemática.

Se quiere diseñar una base de datos relacional para gestionar los datos de los socios de un club náutico. De cada socio se guardan los datos personales y los datos del o de los barcos que posee: número de matrícula, nombre, numero del amarre y cuota que paga por el mismo. Además, se quiere mantener información sobre las salidas realizadas por cada barco, como la fecha y hora de salida, el destino y los datos personales del patrón, que no tiene por qué ser el propietario del barco, ni es necesario que sea socio del club.

Desarrollo.

1. Identificar las entidades y sus respectivos atributos, los cuales si ponemos atención en el texto es fácil de identificar. Datos del barco

- Numero de matricula
- Nombre
- Número del amarre
- Cuota que paga Salidas del barco

- Fecha
- Hora de salida
- Destino
- Datos personales del patrón

Socio

- Datos personales.

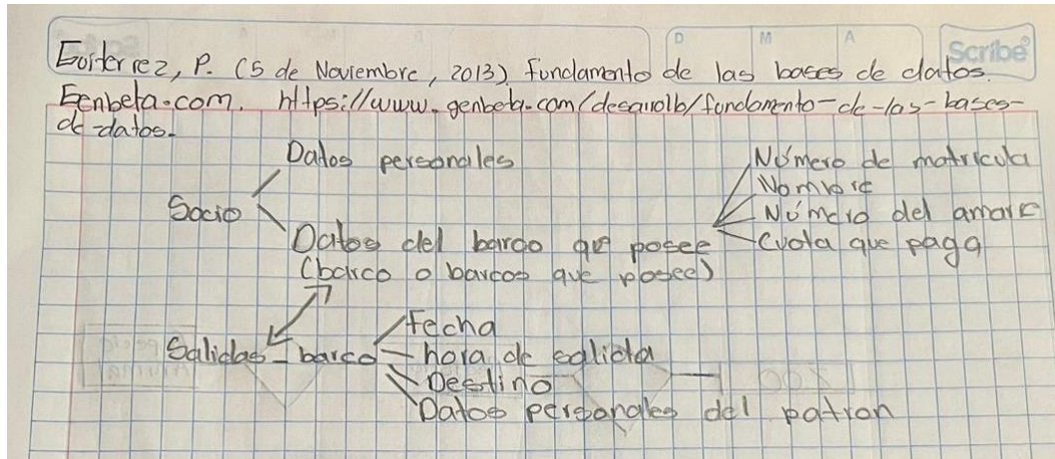


Ilustración 4 Diagrama a mano

2.-Realizamos el diagrama colocando una relación entre las diferentes entidades la cual trae sus atributos, todo esto junto a su cardinalidad o mejor conocida como tipo de relación.

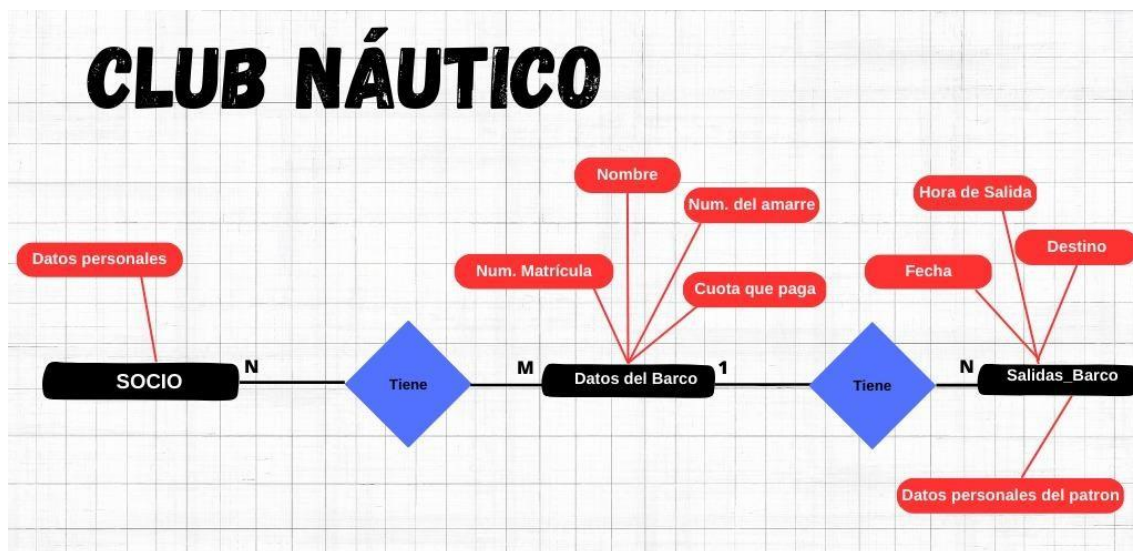


Ilustración 5 Diagrama digital

Unidad 2 DISEÑO DE BASES DE DATOS CON EL MODELO ER

Base de datos “Estudiantes”

Problemática de la base de datos.

Los estudiantes se encuentran matriculados en una carrera. Se necesitan conocer los datos personales de cada estudiante. (No. De cuenta, CURP, RFC, Nombre, ap, am, dirección completa).

¿Qué es el diseño de las bases de datos?

Un buen diseño de base de datos es, por tanto, aquel que: Divide la información en tablas basadas en temas para reducir los datos redundantes. Proporciona a Access la información necesaria para unir la información en las tablas según sea necesario. (*Conceptos Básicos Del Diseño de Una Base de Datos - Soporte Técnico de Microsoft, 2021*)

Un buen diseño de base de datos es, por tanto, aquel que:

- Divide la información en tablas basadas en temas para reducir los datos redundantes.
- Proporciona a Access la información necesaria para unir la información en las tablas según sea necesario. (*Conceptos Básicos Del Diseño de Una Base de Datos - Soporte Técnico de Microsoft, 2021*)
- Ayuda a respaldar y garantizar la precisión y la integridad de la información.
- Se ajusta a sus necesidades de informes y procesamiento de datos. (*Conceptos Básicos Del Diseño de Una Base de Datos - Soporte Técnico de Microsoft, 2021*)

Diseño externo (maquetado).

No. cuenta

Nombre Ap. Paterno

Ap. Materno Estado Municipio

Referencia

CURP RFC Carrera

Ilustración 6 Diseño externo

Diagrama entidad relación (BD estudiantes).

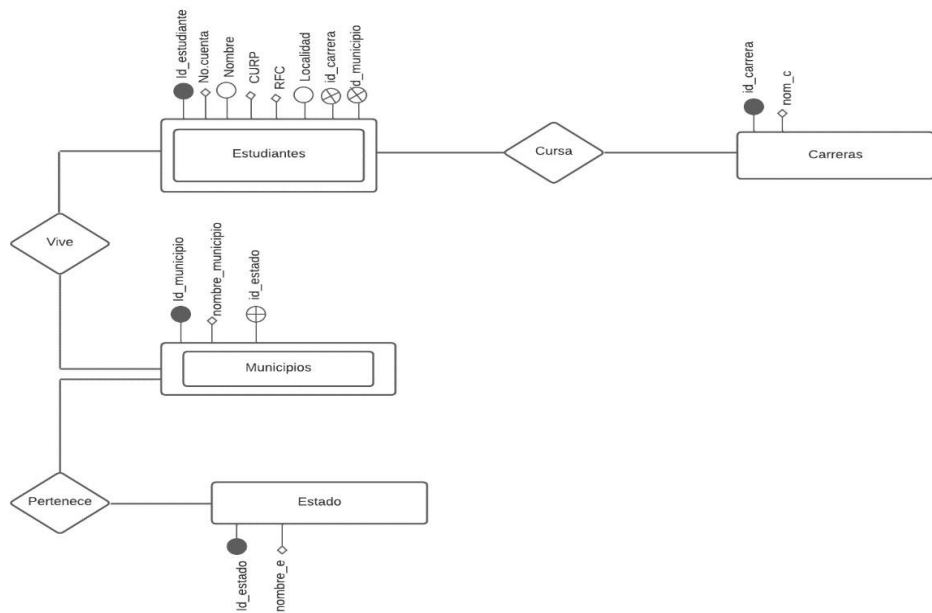


Ilustración 7 Modelo entidad relación

Pasos para realizar la práctica.

```
mysql> create database e
Query OK, 1 row affected

mysql> show databases;
+-----+
| Database |
+-----+
| abarrotes |
| clase1 |
| estudiantes |
| estudiantes1 |
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
```

Ilustración 8 Crear base de datos

- Lo primero a realizar es ingresar al SGBD, crear base de datos para poder trabajar con ella.
- El comando para crear una base de datos es “*create database*” y el nombre que de la base de datos (*create database “nombre”*) y para observar si se creó correctamente utilizamos (*show databases*).

```
mysql> use estudiantes;
Database changed
mysql> describe estudiantes;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_estudiante | int | NO | PRI | NULL | |
| no_cuenta | int | YES | UNI | NULL | |
| nombre | varchar(25) | YES | | NULL | |
| apellido_paterno | varchar(25) | YES | | NULL | |
| apellido_materno | varchar(25) | YES | | NULL | |
| curp | varchar(50) | YES | UNI | NULL | |
| rfc | varchar(50) | YES | UNI | NULL | |
| cp | int | YES | | NULL | |
| localidad | varchar(50) | YES | | NULL | |
| id_carrera | int | YES | MUL | NULL | |
| id_municipio | int | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_estudiantes |
+-----+
| carreras |
| estado |
| estudiantes |
| municipios |
+-----+
```

Ilustración 9 Creación de tablas

- Para utilizarla base de datos utilizar el comando “use y el nombre de la base de datos”.
- El comando “*describe*” es para describir los datos de cada tabla.
- Show tables sirve para visualizar las tablas dentro de la base de datos.


```
mysql> show create table estudiantes;
+-----+
| Table          | Create Table
+-----+
| estudiantes | CREATE TABLE `estudiantes` (
  `id_estudiante` int NOT NULL,
  `no_cuenta` int DEFAULT NULL,
  `nombre` varchar(25) DEFAULT NULL,
  `apellido_paterno` varchar(25) DEFAULT NULL,
  `apellido_materno` varchar(25) DEFAULT NULL,
  `curp` varchar(50) DEFAULT NULL,
  `rfc` varchar(50) DEFAULT NULL,
  `cp` int DEFAULT NULL,
  `localidad` varchar(50) DEFAULT NULL,
  `id_carrera` int DEFAULT NULL,
)
```

Ilustración 10 Visualización de la creación de la BD

- Al usar el comando “*show create table*” y el nombre de la tabla, sirve para visualizar en la consola la creación de dicha tabla.

```
mysql> create table estudiantes(id_estudiante int primary key,
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_estudiantes1 |
+-----+
| estudiantes            |
+-----+
1 row in set (0.00 sec)

mysql> describe estudiantes;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_estudiante | int       | NO   | PRI | NULL    |       |
| numero_cuenta | int       | YES  | UNI | NULL    |       |
| nombre        | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Ilustración 11 Visualización de la tabla creada.

- Al crearse la base de datos, se tienen que crear las tablas correspondientes, en las cuales se ingresaran los datos más adelante.
- Para crear una tabla es con el comando “*create table*” más el nombre y entre paréntesis todos los atributos de la tabla, para llegar a este punto se requiere realizar un diagrama entidad relación para identificar primary keys, uniques, etc.
- Query: *create table estudiantes (id_estudiante int primary key, numero_cuenta int unique, nombre varchar (50));*

```
mysql> select * from carreras;
+----+-----+
| id_carrera | nom_c |
+----+-----+
| 1 | ISC |
+----+-----+
1 row in set (0.00 sec)

mysql> insert into carreras value (2,"Ingenieria Forestal");
Query OK, 1 row affected (0.01 sec)

mysql> insert into carreras value (3,"Ingenieria Industrial");
Query OK, 1 row affected (0.01 sec)

mysql> select * from carreras;
+----+-----+
| id_carrera | nom_c |
+----+-----+
| 2 | Ingenieria Forestal |
| 3 | Ingenieria Industrial |
| 1 | ISC |
+----+-----+
3 rows in set (0.00 sec)

mysql> insert into carreras value (4,"Ingenieria Mecatronica");
Query OK, 1 row affected (0.01 sec)

mysql> insert into carreras value (5,"Ingenieria Civil");
Query OK, 1 row affected (0.01 sec)

mysql> select * from carreras;
+----+-----+
| id_carrera | nom_c |
+----+-----+
| 5 | Ingenieria Civil |
| 2 | Ingenieria Forestal |
| 3 | Ingenieria Industrial |
| 4 | Ingenieria Mecatronica |
| 1 | ISC |
+----+-----+
5 rows in set (0.00 sec)
```

Ilustración 12 Inserción de registros en las tablas

- Para visualizar los datos ingresados en las tablas usamos el siguiente comando *“select * from “* con el nombre de la tabla para saber su contenido.
- Para insertar datos a una tabla es *insert into “nombre de la tabla” value (id,”cadena”)*.
- Query (para ver el contenido de la tabla estudiantes): *select * from estudiantes;*
- Query (para insertar datos a la tabla carrera): *insert into value {2, “ingeniería forestal”}*.

```
mysql> show tables;
+-----+
| Tables_in_estudiantes |
+-----+
| carreras |
| estado |
| estudiantes |
| municipios |
+-----+
4 rows in set (0.00 sec)

mysql> insert into estado value (1,"Estado de Mexico");
Query OK, 1 row affected (0.01 sec)

mysql> insert into estado value (2,"Queretaro");
Query OK, 1 row affected (0.01 sec)

mysql> insert into estado value (3,"Michoacan");
Query OK, 1 row affected (0.01 sec)

mysql> insert into estado value (4,"Hidalgo");
Query OK, 1 row affected (0.01 sec)

mysql> select * from estado;
+----+-----+
| id_estado | nombre_e |
+----+-----+
| 1 | Estado de Mexico |
| 4 | Hidalgo |
| 3 | Michoacan |
| 2 | Queretaro |
+----+-----+
4 rows in set (0.00 sec)
```

Ilustración 13 inserción en siguiente registro

Para insertar valores a toda tabla utilizamos el mismo comando, cambiando el nombre de la tabla, y entre paréntesis los datos correspondientes.

- Query: *insert into estado values (1, "Estado de mexico");*

```
mysql> describe municipios;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_municipio | int       | NO   | PRI | NULL    |       |
| nom_municipio | varchar(50) | YES |     | NULL    |       |
| id_estado    | int       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (1,"Valle de bravo",1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (2,"Villa Victoria",1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (3,"Bernal",2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (4,"Pinal de amales",2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (5,"Zitacuaro",3);
Query OK, 1 row affected (0.01 sec)
```

Ilustración 14 Inserción de registros en la tabla estados.

- Ingresamos los datos a la tabla de municipios, como clave para no confundir, se puede describir la tabla municipios para ver los datos que pide.
- Query: *insert into municipios (id_municipio, nom_municipio,id_estado) value (1, "Valle de bravo",1);*

```

MySQL 8.0 Command Line Cli x + v

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (1,'Valle de bravo',1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into municipios (id_municipio,nom_municipio,id_estado) values (2,'Villa Victoria',1);
Query OK, 1 row affected (0.01 sec)

mysql> select * from municipios;
+-----+-----+-----+
| id_municipio | nom_municipio | id_estado |
+-----+-----+-----+
| 1 | Valle de bravo | 1 |
| 2 | Villa Victoria | 1 |
| 3 | Bernal | 2 |
| 4 | Pinal de amales | 2 |
| 5 | Zitacuaro | 3 |
| 6 | Morelia | 3 |
| 7 | Pachuca | 4 |
| 8 | Actopan | 4 |
+-----+-----+-----+

```

Ilustración 15

- Una vez ingresado los datos necesarios, podemos visualizarlos.
- Query: *select * from municipios;*

```

MySQL 8.0 Command Line Cli x + v

mysql> insert into estudiantes (id_estudiante,no_cuenta,nombre,apellido_paterno,apellido_materno,curp,rfc,cp,localidad,id_carrera,id_municipio) values (5,202107027,'Juan Pablo','Perez','Santo Tomas de los Platanos',1,1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes (id_estudiante,no_cuenta,nombre,apellido_paterno,apellido_materno,curp,rfc,cp,localidad,id_carrera,id_municipio) values (6,202107007,'Mariana','Campos','Villa de allende ',1,1);
Query OK, 1 row affected (0.01 sec)

mysql> select * from estudiantes;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_estudiante | no_cuenta | nombre | apellido_paterno | apellido_materno | curp | rfc | cp | localidad | id_carrera | id_municipio |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 202107005 | Jonathan | Hernandez | Guzman | HEGJ030820HMKR2NA3 | HEGJ030820 | 50960 | Santiago del Monte | 1 | 2 |
| 2 | 202107021 | Diego | Mercado | Tavira | PIED030806HMKR2NA3 | PIED030806 | 51200 | Valle de bravo | 1 | 1 |
| 3 | 202107059 | Armando | Guadarrama | Osorio | GLOL030100HMKR2NA3 | GLOL030100 | 51200 | Las Joyas | 1 | 1 |
| 4 | 202107013 | Jose de Jesus | Solorzano | Rangel | SORJ030202HMKR2NA3 | SORJ030202 | 51230 | Las violetas | 1 | 1 |
| 5 | 202107027 | Juan Pablo | Perez | Ortega | PEDJ031008HMKR2NA3 | PEDJ031008 | 51100 | Santo Tomas de los Platanos | 1 | 1 |
| 6 | 202107007 | Mariana | Campos | Ceballos | CACH030920HMKR2NA3 | CACH030920 | 51000 | Villa de allende | 1 | 1 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> insert into estudiantes (id_estudiante,no_cuenta,nombre,apellido_paterno,apellido_materno,curp,rfc,cp,localidad,id_carrera,id_municipio) values (7,202007004,'Ariel','Quintero','Tlaxiaco de Bravo ',1,1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes (id_estudiante,no_cuenta,nombre,apellido_paterno,apellido_materno,curp,rfc,cp,localidad,id_carrera,id_municipio) values (8,202107026,'Eslly Jared','Chala','Villa Victoria',1,2);
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes (id_estudiante,no_cuenta,nombre,apellido_paterno,apellido_materno,curp,rfc,cp,localidad,id_carrera,id_municipio) values (9,202107057,'Angel','Carro','Apulinar',1,1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes (id_estudiante,no_cuenta,nombre,apellido_paterno,apellido_materno,curp,rfc,cp,localidad,id_carrera,id_municipio) values (10,202107063,'Hector Gamaliel','Volquez','Villa Victoria',1,2);
Query OK, 1 row affected (0.01 sec)

mysql> select * from estudiantes;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_estudiante | no_cuenta | nombre | apellido_paterno | apellido_materno | curp | rfc | cp | localidad | id_carrera | id_municipio |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | 202107005 | Jonathan | Hernandez | Guzman | HEGJ030820HMKR2NA3 | HEGJ030820 | 50960 | Santiago del Monte | 1 | 2 |
| 2 | 202107021 | Diego | Mercado | Tavira | PIED030806HMKR2NA3 | PIED030806 | 51200 | Valle de bravo | 1 | 1 |
| 3 | 202107059 | Armando | Guadarrama | Osorio | GLOL030100HMKR2NA3 | GLOL030100 | 51200 | Las Joyas | 1 | 1 |
| 4 | 202107013 | Jose de Jesus | Solorzano | Rangel | SORJ030202HMKR2NA3 | SORJ030202 | 51230 | Las violetas | 1 | 1 |
| 5 | 202107027 | Juan Pablo | Perez | Ortega | PEDJ031008HMKR2NA3 | PEDJ031008 | 51100 | Santo Tomas de los Platanos | 1 | 1 |
| 6 | 202107007 | Mariana | Campos | Ceballos | CACH030920HMKR2NA3 | CACH030920 | 51000 | Villa de allende | 1 | 1 |
| 7 | 202007004 | Ariel | Quintero | Bautista | QUBA090923HMKR2NA3 | QUBA090923 | 51200 | Valle de Bravo | 1 | 1 |
| 8 | 202107026 | Eslly Jared | Chala | Carbajal | CACE030421HMKR2NA3 | CACE030421 | 50960 | Villa Victoria | 1 | 2 |
| 9 | 202107057 | Angel | Carro | Apulinar | CAAR030330HMKR2NA3 | CAAR030330 | 51260 | Amatlan | 1 | 1 |
| 10 | 202107063 | Hector Gamaliel | Volquez | Garcia | VEGH030405HMKR2NA3 | VEGH030405 | 50900 | Villa Victoria | 1 | 2 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> show databases;

```

Ilustración 16 Consulta de tabla municipios y estudiantes.

- Ingresar los datos a la tabla estudiantes
- Query: *insert into estudiantes (id_estudiante, no_cuenta, nombre, apellido_paterno, apellido_materno, curp, rfc, cp, localidad, id_carrera, id_municipio) values (...);*

*id_municipio) values (10,202107063, "Hector Gamaliel", "Velazquez",
"Garcia",
"vegho30405hmc/rca2", "vegho30405",50900, "Villa Victoria",1,2);*

Práctica Multiconsultas Estudiantes

Con la BD estudiantes, mostrar el nombre de los estudiantes de ISC.

Análisis.

¿Qué? Nombre, apellido paterno, apellido materno.

¿Dónde? Estudiantes, carreras.

Condición: nom_c="ISC" Comando por ejecutar.

```
select E.nombre, E.apellido_paterno, E.apellido_materno from Estudiantes E,  
Carreras C where c.nom_c= "ISC";
```

Recuerda que se puede utilizar alias para referir a algunas entidades, en este caso utilizar E para dirigir a estudiantes y C para carreras.

La descripción del comando dice que mostrara el nombre, apellido paterno y apellido materno de las personas que estudian ingeniería en sistemas.

```
mysql> select E.nombre, E.apellido_paterno, E.apellido_materno  
+-----+-----+-----+  
| nombre      | apellido_paterno | apellido_materno |  
+-----+-----+-----+  
| Jonathan    | Hernandez        | Guzman          |  
| Diego       | Mercado         | Tavira          |  
| Armando     | Guadarrama      | Osorio          |  
| Jose de Jesus | Solorzano       | Rangel          |  
| Juan Pablo  | Perez           | Ortega          |  
| Mariana     | Campos          | Ceballos        |  
| Ariel       | Quintero        | Bautista        |  
| Esly Jared  | Chala           | Carbajal        |  
| Angel       | Carro           | Apolinar        |  
| Hector Gamaliel | Velazquez       | Garcia          |  
+-----+-----+-----+  
10 rows in set (0.01 sec)
```

Ilustración 17 Consulta estudiantes ISC

Práctica 2 (consulta)

Utilizando la BD estudiantes, mostrar el nombre del estado en el que vive Diego Mercado Tavira.

Análisis.

¿Qué? Nombre_e

¿Dónde? Estudiantes, municipio, estado.

Condición: Nombre="Diego",
apellido_paterno="Mercado",apellido_materno="Tavira".

Comando para ejecutar.

```
Select nombre_e from estado, estudiantes,municipios where
estudiantes.nombre="Diego" and estudiantes.apellido_paterno="Mercado" and
estudiantes.apellido_materno="Tavira" and estudiantes.id_municipio=
municipios.id_municipio and municipios.idestado=estado.id_estado;
```

```
mysql> select nombre_e
nicipios.id_municipio a
+-----+
| nombre_e |
+-----+
| Estado de Mexico |
+-----+
1 row in set (0.00 sec)

mysql> |
```

Ilustración 18 Consulta estado

Descripción del Query.

Esto indica que se mostrara el estado donde el nombre sea Diego, el apellido paterno Mercado y el apellido materno Tavira.

Base de datos "Abarrotes"

Problemática de la base de datos.

Se desea diseñar una BD para registro de abarrotes con las siguientes características.

Nombre del producto, clave, precio, categoría (semillas, carnes frías, vinos y licores...) así como la zona en donde se encuentran ubicados (pasillo A, B...).

Diseño externo (maquetado).

Registro abarrotes.

Nombre del producto	<input type="text"/>
Clave	<input type="text"/>
Precio	<input type="text"/>
Categoría	<input type="text" value="▼"/>
Zona	<input type="text" value="▼"/>

Ilustración 19 Maquetado

Diagrama entidad relación (BD abarrotes).

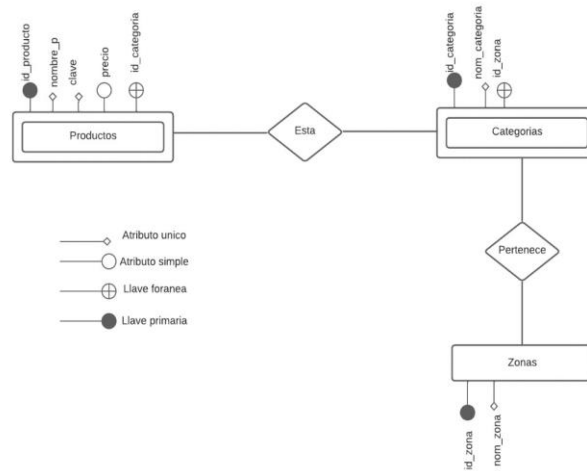
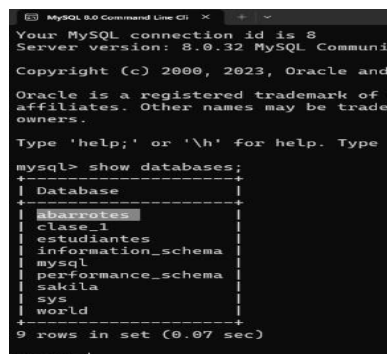


Ilustración 20 Modelo entidad relación

Pasos para realizar la práctica.

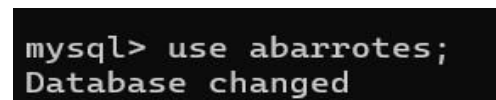
1._El primer paso es abrir el Mysql command line, una vez ingresada la contraseña del usuario, digitar el comando *Create database abarrotes;* este comando permite crear la base de datos, para consultar su creación, digitar el comando *show databases;* debe aparecer un listado de todas las bases de datos que se han creado y entre ellas debe existir una de nombre “abarrotes” como se observa en la siguiente imagen (lizhernandez, 2013).



```
mysql> show databases;
+-----+
| Database |
+-----+
| abarrotes |
| class1   |
| estudiantes |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
9 rows in set (0.07 sec)
```

Ilustración 21 Revisar las BD

2._Después de corroborar la creación de la base de datos ahora se necesita usar la base de datos con el comando *use abarrotes;* este comando permite ingresar a la base de datos creada, debe de aparecer un mensaje de que se cargó la base de datos (lizhernandez, 2013).



```
mysql> use abarrotes;
Database changed
```

Ilustración 22 Usar BD

3._El siguiente paso es crear las tablas del modelo entidad relación indicada, en este caso comenzar con las entidades fuertes que es “productos” el comando para realizarla es *create table productos (id_producto int, nombre_producto char(20), clave int, precio float, id_categoria int, id_zona int);*

Este comando se divide en varias secciones, la primera sección inicia en `create table productos`, esta parte significa que se va a crear una tabla llamada productos.

La segunda sección es la declaración de entidades o campos en este caso colocar de color rojo los campos a excepción de los dos últimos ya que ellos son campos foráneos que veremos más adelante y azul el tipo de dato que permite dicho campo y de verde la longitud de caracteres permitidos al momento de hacer registros

```
(id_producto int, nombre_producto char(20), clave int, precio float, id_categoria int, id_zona int);
```

En la tercera sección se puede notar que hay dos entidades de color morado (*id_categoria* y *id_zona*), estas se encuentran diferente ya que estos campos serán campos foráneos invocados de otras entidades, esto para que la tabla productos tenga una relación con las demás tablas.

Al dar enter aparece el siguiente mensaje justamente debajo de tu comando, en caso de que no aparezca dicho mensaje será necesario revisar su escritura o sintaxis (lizhernandez, 2013).



```
Query OK, 0 rows affected (0.05 sec)
```

Ilustración 23 Mensaje de creación de tabla productos

4._ Después de crear la entidad productos, prosigue a crearse la entidad categorías, al igual que la entidad anterior, colocar el siguiente comando `Create table categorías` (*id_categoria* int, *nombre_categoria* char(30)); (Gittins, L. 2013);

5._Por ultimo crear la última entidad que es zonas, comenzamos el comando con `create table zonas` (*id_zona* int, *nombre_zona* char(30)); (Gittins, L. 2013);

6._Una vez terminado realizar la consulta de las entidades con el comando

```
show tables; (Gittins, L. 2013);
```

```
mysql> show tables;
+-----+
| Tables_in_abarrotes |
+-----+
| categorias           |
| productos           |
| zonas                |
+-----+
```

Ilustración 24 Consulta de tablas de la base de datos.

7._El siguiente paso consiste en colocar las primary key y foreign key en las entidades correspondientes, iniciar declarando la primary key en el campo de `id_producto` de la entidad `productos` (para modificar una tabla es necesario el comando `alter table`, el comando para declarar el primary key es *`alter table productos add constraint primary key(id_producto)`*; aparecer el siguiente mensaje de lo contrario será necesario revisar sintaxis (SiteGround, W. H. (s/f)).

```
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 25 Mensaje de creación primary key `id_producto`.

8._Ahora que se creó la primary key definida, retomar los dos campos que se colocó de color morado en el paso 3 para poder declarar la llave foránea (foreign key), digitar el comando *`alter table productos add constraint foreign key (id_categoria) references categorías (id_categoria)`*; y aparece el siguiente mensaje de lo contrario será necesario revisar sintaxis (SiteGround, W. H. (s/f)).

```
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 26 Mensaje de creación foreign key de `id_categoria`

9._Ahora colocar la llave foránea al último campo de la tabla de productos que es el id_zonas, digitar el siguiente comando, *alter table productos add constraint foreign key (id_zona) references zonas (id_zona);* y aparece el siguiente mensaje de lo contrario será necesario revisar sintaxis (SiteGround, W. H. (s/f)).

```
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 27 Mensaje de creación foreign key de id_zona.

10._Realizar descripción para verificar que las llaves estén declaradas en los campos correspondientes, usar el comando describe productos; (SiteGround, W. H (s/f)).

```
mysql> describe productos;
+-----+-----+-----+-----+-----+-----+
| Field          | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_producto    | int       | NO   | PRI | NULL     |       |
| nombre_producto | char(30)  | YES  |     | NULL     |       |
| clave          | char(30)  | YES  |     | NULL     |       |
| precio         | float     | YES  |     | NULL     |       |
| id_categoria   | int       | YES  | MUL | NULL     |       |
| id_zona        | int       | YES  | MUL | NULL     |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Ilustración 28 Descripción de la tabla productos.

11._ Ahora que las entidades tienen las características completas ahora comenzar a realizar la inserción de registros, colocar los registros de la tabla categorías, colocar el comando *insert into categorias values(1, "asimentos_animal");* como se observa los datos del registro tenían que ir en el orden en el que se colocó los campos de la tabla ya que de lo contrario los datos no serán admitidos en otro campo y el registro marcará error, hacer lo mismo pero cambiando los datos de id_categoria y nombre de categoría, pero para todos. Utilizar el mismo comando, al

terminar colocar el comando `select * from categorías;` el cual permite ver que registros y que datos tiene la tabla categorías (Quintana, A. (s/f)).

```
mysql> select * from categorías;
+-----+-----+
| id_categoria | nombre_categoria |
+-----+-----+
|          1 | Alimentos_animal |
|          2 | Bebidas           |
|          3 | Golosinas         |
|          4 | chatarra          |
+-----+-----+
4 rows in set (0.00 sec)
```

Ilustración 29 Consulta de registros de la tabla de categorías

12._Con las bases del paso 11, realizar la inserción de 4 registros en nuestra tabla de zonas, colocar el comando, `insert into zonas values(1, "pasillo A");` al igual que los registros anteriores replicar el comando solo cambiando el id_zona y el nombre de la zona, al terminar consultar la tabla con el comando `select * from zonas;` (Quintana, A. (s/f)).

```
mysql> select * from zonas;
+-----+-----+
| id_zona | nombre_zona |
+-----+-----+
|        1 | Pasillo A   |
|        2 | Pasillo B   |
|        3 | Pasillo C   |
|        4 | Pasillo D   |
+-----+-----+
4 rows in set (0.00 sec)
```

Ilustración 30 Consulta de registros de la tabla de zonas

13._Ahora que se tiene las tablas de categorías y zonas llenas ahora sí, realizar la inserción de registros en la tabla ya que si estas no estuvieran llenas la clave id_categoria y el id_zona que se coloca en las tablas no podrían ser convocados en la tabla de productos con la foreign key es por es que se debe iniciar con las entidades que no tienen foreign key (Quintana, A. (s/f)).

14._Antes de iniciar los registros debemos recordar que los datos deben colocar en el orden exacto de los campos. Iniciar los registros con el *comando insert into productos values(1, "huevos",1234,100,1,1);*, con el anterior comando se le dio un entender a la computadora que el registro con el id=1 son huevos, con la clave 1234 con un precio de 100 y que se encontrara en la categoría con id=1 y en la zona con el id=1.

Consultar antes de realizar los demás registros utilizando el comando *select * from productos;* (Quintana, A. (s/f)).

```
mysql> select * from productos;
+-----+-----+-----+-----+-----+-----+
| id_producto | nombre_producto | clave | precio | id_categoria | id_zona |
+-----+-----+-----+-----+-----+-----+
|          1 | huevos          | 12345 | 100    | 1           | 1       |
+-----+-----+-----+-----+-----+-----+
```

Ilustración 31 Consulta de registro de la tabla productos

15._Al confirmar el registro, utilizando el mismo comando realizar el registro de 4 productos más, cambiando la información del registro, en este caso asignando cada producto a una zona y categoría diferente esto para que la llave foránea trabaje relacionando las entidades aún más.

Después de registrar volver a consultar la tabla con el comando *select * from productos;* (Comandos SQL Básicos. (s/f)).

```
mysql> select * from productos;
+-----+-----+-----+-----+-----+-----+
| id_producto | nombre_producto | clave | precio | id_categoria | id_zona |
+-----+-----+-----+-----+-----+-----+
| 1 | huevos | 12345 | 100 | 1 | 1 |
| 2 | Soda | 54321 | 20 | 2 | 2 |
| 3 | Carlos v | 56789 | 5 | 3 | 3 |
| 4 | papas | 78945 | 20 | 4 | 4 |
| 5 | chicharrones | 124585 | 15 | 4 | 4 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Ilustración 32 Consulta de la tabla de productos.

Práctica Multiconsultas Abarrotes

Utilizando la BD abarrotes, mostrar el precio y el nombre de los productos de chatarra.

Análisis.

¿Qué? Precio, nombre.

¿Dónde? Productos, categorías.

Condición: categorías.nom_categoria="Chatarra".

Comando por ejecutar.

Select precio,nombre_p from productos, categorías where categorías.nom_categoria="chatarra" and productos.id_categoria=categorías.id_categoria;

```
mysql> select productos.nombre_producto, precio
+-----+-----+
| nombre_producto | precio |
+-----+-----+
| papas | 20 |
| chicharrones | 15 |
+-----+-----+
2 rows in set (0.00 sec)
```

Ilustración 33 Consulta de nombre y precio de productos.

Descripción del Query.

Este comando va a mostrar el nombre y el precio del producto donde la única condición es que la categoría sea chatarra en este caso en nuestro registro el 4 pertenece a esa zona.

Práctica 2 (consulta).

Con la BD abarrotes, mostrar el nombre de los productos ubicados en el pasillo A.

Análisis.

¿Qué? Nombre_p

¿Dónde? Productos, categorías y zonas.

Condición. Zonas.nom_zona="Pasillo A".

Comando por ejecutar.

```
Select productos.nombre_producto from productos, zonas where  
productos.id_zona= "1" and productos.id_zona=zonas.id_zona;
```

```
mysql> select productos.nombre_producto  
+-----+  
| nombre_producto |  
+-----+  
| huevos          |  
+-----+  
1 row in set (0.00 sec)
```

Ilustración 34 Consulta de nombre de productos del pasillo.

Descripción del Query.

Indica que al seleccionar los nombres de los productos de la tabla productos, zonas, donde el nombre de la zona sea "pasillo A" que este corresponde al número 1 en la tabla de zonas, y que el id_cat de la tabla productos sea igual al id_cat de la tabla categorías y el id_zona de la tabla categorías sea igual al id_zona de la tabla zonas.

Descripción de los querrys utilizados.

Comandos utilizados	Descripción.	Ejemplo
- Create database	Sirve para crear nuestra base de datos en el SGBD.	<i>Create database estudiantes;</i>
- Show databases	Ayuda a visualizar las bases de datos que tenemos en nuestro SGBD.	<i>Show databases;</i>
- Use "nombre de la base de datos"	Prácticamente este es para utilizar la base de datos que se quiere modificar.	<i>Use estudiantes;</i>
- Show tables	Con este comando se puede visualizar las tablas que están dentro de la base de datos.	<i>Show tables;</i>
- Describe "nombre de la tabla"	Sirve para describir todos los datos que tiene en las tablas, (se puede especificar cual tabla deseas describir)	<i>Describe carreras;</i>
- Create table	Ayuda a crear una tabla dentro de la base de datos, dentro de este comando especificar los atributos que se desea asignar.	<i>Create table carreras (id_carrera int primary key, nombre_carrera varchar (40));</i>
- Select * from	Ayuda a visualizar si los datos se están ingresando correctamente.	<i>Select * from carreras;</i>

Unidad 3 MODELO RELACIONAL

Base de datos "Universidad".

Problemática de la base de datos.

En una universidad se encuentran artículos los cuales se publican en revistas, publicaciones revistas, publicaciones de congresos, congresos, diversos tipos de

congresos, frecuencias en diferentes ciudades y países también están los investigadores que tienen distintos temas.

¿Qué es el diseño de las bases de datos?

Un buen diseño de base de datos es, por tanto, aquel que: Divide la información en tablas basadas en temas para reducir los datos redundantes. Proporciona a Access la información necesaria para unir la información en las tablas según sea necesario. (*Conceptos Básicos Del Diseño de Una Base de Datos - Soporte Técnico de Microsoft, 2021*)

Un buen diseño de base de datos es, por tanto, aquel que:

- Divide la información en tablas basadas en temas para reducir los datos redundantes.
- Proporciona a Access la información necesaria para unir la información en las tablas según sea necesario. (*Conceptos Básicos Del Diseño de Una Base de Datos - Soporte Técnico de Microsoft, 2021*)
- Ayuda a respaldar y garantizar la precisión y la integridad de la información.
- Se ajusta a sus necesidades de informes y procesamiento de datos. (*Conceptos Básicos Del Diseño de Una Base de Datos - Soporte Técnico de Microsoft, 2021*)

Diseño externo (maquetado).

UNIVERSIDAD			
Personas	<input type="text"/>	Centro	<input type="text" value="Seleccione"/>
Investigadores	<input type="text" value="Seleccione"/>	Temas	<input type="text" value="Seleccione"/>
Articulos	<input type="text" value="Seleccione"/>	Copias	<input type="text" value="Seleccione"/>
Ubicaciones	<input type="text" value="Seleccione"/>	Informes	<input type="text" value="Seleccione"/>
Congresos	<input type="text" value="Seleccione"/>	Tipo Congreso	<input type="text" value="Seleccione"/>
Publicacion Congresos	<input type="text" value="Seleccione"/>	Publicacion Informes	<input type="text" value="Seleccione"/>
Revistas	<input type="text" value="Seleccione"/>	Publicacion Revistas	<input type="text" value="Seleccione"/>
Frecuencia	<input type="text" value="Seleccione"/>	Ciudad	<input type="text" value="Seleccione"/>
Editores	<input type="text" value="Seleccione"/>	Pais	<input type="text" value="Seleccione"/>

Ilustración 35 Maquetado

Diagrama entidad relación (BD estudiantes).

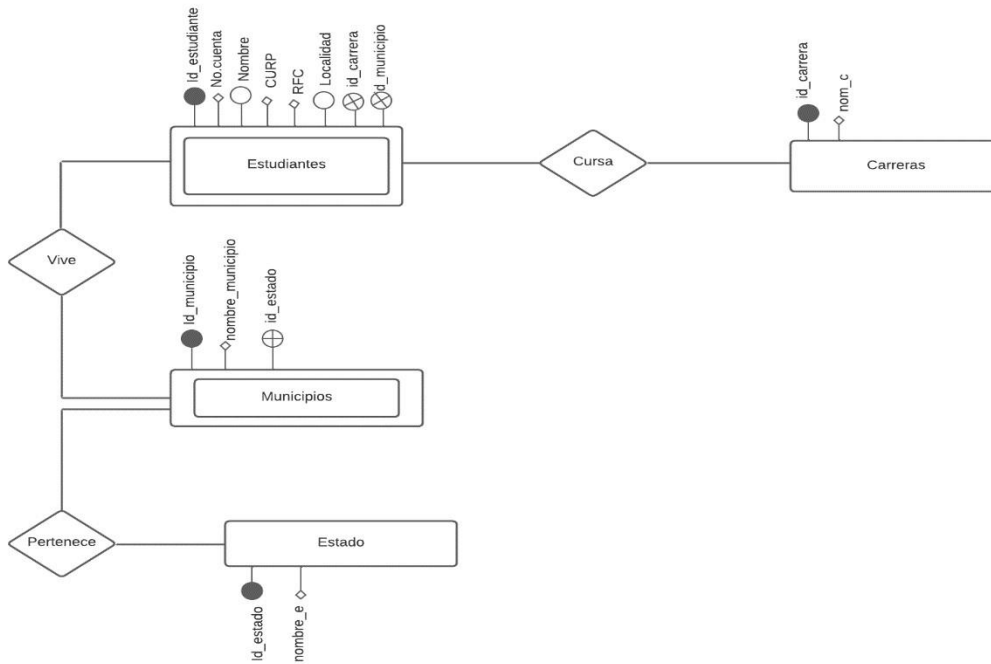


Ilustración 36 Modelo entidad relación

Pasos para realizar la práctica.

```
mysql> create database universidad1;
Query OK, 1 row affected (0.10 sec)

mysql>
```

Ilustración 37 Creación de la base de datos

- Lo primero que se debe de hacer al ingresar a nuestro SGBD, es crear la base de datos para poder trabajar con ella.
- El comando que se utiliza es *create database* y el nombre que se le asigne y para visualizar si se creó correctamente utilizar *show databases*.

```
mysql> describe personas;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_persona | int           | NO   | PRI | NULL    |       |
| nom_p      | varchar(80)  | YES  |     | NULL    |       |
| ap_p      | varchar(50)  | YES  |     | NULL    |       |
| ap_m      | varchar(50)  | YES  |     | NULL    |       |
| correo     | varchar(40)  | YES  |     | NULL    |       |
| id_cendro  | int           | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Ilustración 38 Usar base de datos

- Al utilizar *show create table* y el nombre de la tabla, visualizar en la consola la creación de dicha tabla.

```
mysql> create table personas(id_persona int primary key not null,nom_p varchar (80) ,ap_p varchar (50),ap_m varchar (50)
,correo varchar (40),id_cendro int not null );
Query OK, 0 rows affected (0.00 sec)
```

Ilustración 39 Visualización de la tabla creada.

- Una vez creada la base de datos, crear las correspondientes tablas, en la cual ingresar datos más adelante.

- Para crear una tabla es *create table* más el nombre y entre paréntesis todos los atributos de la tabla, para llegar a este punto realizar un diagrama entidad relación para identificar primary keys, uniques, etc.
- *create table personas (id_persona int primary key not null, nom_p varchar (80), ap_p varchar (50), ap_m varchar (50), correo varchar (40), id_centro int not null);*

```
mysql> describe table personas;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | select_type | table | partitions | type | possible_keys | key | key_len | ref | rows | filtered | Extra |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | SIMPLE | personas | NULL | ALL | NULL | NULL | NULL | NULL | 1 | 100.00 | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set, 1 warning (0.03 sec)

mysql> insert into personas values(1,"Diego","Mercado","Tavira","l2021070@Vbravo.tecnm.mx",1);
Query OK, 1 row affected (0.11 sec)

mysql> insert into personas values(2,"Miguel","Sanchez","Hernandez","l202107004@Vbravo.tecnm.mx",2);
Query OK, 1 row affected (0.09 sec)

mysql> insert into personas values(3,"Patricio","Torres","De Paz","l202107017@Vbravo.tecnm.mx",3);
Query OK, 1 row affected (0.07 sec)

mysql> insert into personas values(4,"Simon","Carlos","Cervantes","l202107055@Vbravo.tecnm.mx",4);
Query OK, 1 row affected (0.03 sec)

mysql> select * from personas;
+-----+-----+-----+-----+-----+-----+
| id_persona | nom_p | ap_p | ap_m | correo | id_centro |
+-----+-----+-----+-----+-----+-----+
| 1 | Diego | Mercado | Tavira | l2021070@Vbravo.tecnm.mx | 1 |
| 2 | Miguel | Sanchez | Hernandez | l202107004@Vbravo.tecnm.mx | 2 |
| 3 | Patricio | Torres | De Paz | l202107017@Vbravo.tecnm.mx | 3 |
| 4 | Simon | Carlos | Cervantes | l202107055@Vbravo.tecnm.mx | 4 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Ilustración 40 Inserción de registros en las tablas

- Para visualizar los datos ingresados en las tablas escribir el siguiente comando *select * from* con el nombre de la tabla la cual deseas saber su contenido.
- Para insertar datos a una tabla es *insert into "nombre de la tabla" value (id, "cadena");*
- Query (para ver el contenido de la tabla estudiantes): *select * from estudiantes;*

```
mysql> select * from personas;
+-----+-----+-----+-----+-----+-----+
| id_persona | nom_p | ap_p | ap_m | correo | id_cendro |
+-----+-----+-----+-----+-----+-----+
| 1 | Diego | Mercado | Tavira | L2021070@Vbravo.tecnm.mx | 1 |
| 2 | Miguel | Sanchez | Hernandez | L202107004@Vbravo.tecnm.mx | 2 |
| 3 | Patricio | Torres | De Paz | L202107017@Vbravo.tecnm.mx | 3 |
| 4 | Simon | Carlos | Cervantes | L202107055@Vbravo.tecnm.mx | 4 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Ilustración 41 Consulta de la tabla personas.

- Para insertar valores a toda tabla es el mismo comando, cambiando el nombre de la tabla, y entre paréntesis los datos correspondientes.
- Query: *insert into personas*

values(4, "Simon", "Carlos", "Cervantes", "L202107055@Vbravo.tecnm.mx", 4);

```
mysql> insert into publicacioninformes values (1,1,6,2020,"Enero"),(2,2,7,2021,"Agosto");
Query OK, 5 rows affected (0.06 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> select * from publicacioninformes;
+-----+-----+-----+-----+-----+
| id_publicacion_inf | id_informe | id_articulo | anio | mes |
+-----+-----+-----+-----+-----+
| 1 | 1 | 6 | 2020 | Enero |
| 2 | 2 | 7 | 2021 | Marzo |
| 3 | 3 | 8 | 2022 | Febrero |
| 4 | 4 | 9 | 2018 | Diciembre |
| 5 | 5 | 10 | 2015 | Agosto |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Ilustración 42 Inserción de registros en la tabla publicacioninformes.

- Ingresar los datos a la tabla de municipios, como clave para no confundir, se puede describir la tabla municipios para ver los datos que pide.
- Query: *insert into publicación informes values*

(1,1,6,2020,"Enero"),(2,2,7,2021,"Marzo"),(3,3,8,2022,"Febrero"),(4,4,9,2018,"Diciembre"),(5,5,10,2015,"Agosto");

```
mysql> show tables;
+-----+
| Tables_in_universidad |
+-----+
| articulos              |
| asignaautores          |
| asignatemas            |
| asignatemasr           |
| centros                |
| ciudades                |
| congresos              |
| copias                  |
| editores                |
| frecuencias             |
| informes                |
| investigadores         |
| paises                  |
| personas                |
| pubcongresos           |
| publicacionesrevistas  |
| publicacioninformes   |
| revistas                |
| temas                  |
| tipocongresos          |
| ubicaciones            |
+-----+
21 rows in set (0.10 sec)

mysql>
```

Ilustración 43 Tablas creadas en la BD

- Una vez ingresado sabiendo como crear las tablas e insertar datos ya no es necesario poner todas las tablas de la base de datos así que ya acabando solo utilizar el comando *show tables*; para ver todas las tablas creadas a continuación son 21 tablas.

```
mysql> select * from articulos;
+-----+-----+-----+
| id_articulo | titulo                                | palabasc |
+-----+-----+-----+
| 1           | El Medio Ambiente                    | Contaminacion |
| 2           | El Nuevo Futuro                      | Tecnologia    |
| 3           | Influencia                           | Psicologia    |
| 4           | Beneficios del Ejercicio              | Salud         |
| 5           | La Comunicacion Efectiva              | Sociabilidad  |
| 6           | El Futuro del Trabajo                  | Trabajo en Casa |
| 7           | Educacion Emocional                   | Emociones     |
| 8           | La Etica en la AI                     | Etica         |
| 9           | La Neurociencia del Aprendizaje       | Cerebro       |
| 10          | El Arte de la Programacion             | Programacion  |
+-----+-----+-----+
10 rows in set (0.07 sec)

mysql> select * from temas;
+-----+-----+
| id_temas | nom_tema |
+-----+-----+
| 4        | Arduino |
| 1        | Ciencias Naturales |
| 2        | Matematicas |
| 5        | Programacion |
| 3        | Programacion en Python |
+-----+-----+
5 rows in set (0.04 sec)
```

Ilustración 44 Consulta de tablas.

```
mysql> select * from investigadores;
+-----+-----+
| id_investigadores | id_persona |
+-----+-----+
|          1 | 1 |
|          2 | 2 |
|          3 | 3 |
|          4 | 4 |
|          5 | 5 |
+-----+-----+
5 rows in set (0.04 sec)

mysql> select * from informes;
+-----+-----+-----+
| id_informe | num_centro | centro |
+-----+-----+-----+
|          1 |          1 | A1 |
|          2 |          2 | A2 |
|          3 |          3 | B1 |
|          4 |          4 | B2 |
|          5 |          5 | C1 |
+-----+-----+-----+
5 rows in set (0.02 sec)

mysql> select * from editores;
+-----+-----+
| id_editor | id_persona |
+-----+-----+
|          1 |          6 |
|          2 |          7 |
|          3 |          8 |
|          4 |          9 |
|          5 |         10 |
+-----+-----+
5 rows in set (0.02 sec)
```

Ilustración 45 Consulta de tablas.

- Ver las tablas creadas ya con la inserción de datos.
- El comando que se utiliza para esto es *select * from* y el nombre de la tabla.
- A continuación, mostrar algunas de las tablas ya con información.

Práctica 1 (Consulta)

Con la BD universidad, mostrar el nombre apellido paterno y materno de los investigadores.

Análisis.

¿Qué? Nombre, apellido paterno, apellido materno.

¿Dónde? Personas, investigadores.

Comando por ejecutar.

```
select p.nom_p,p.ap_p,p.ap_m from personas p, investigadores i where  
p.id_persona=i.id_persona;
```

Recuerda que se puede utilizar alias para referir a alguna entidad, en este caso utilizar “p” para referir a personas y “i” para investigadores.

La descripción del comando dice que mostrara el nombre, apellido paterno y apellido materno de las personas que son investigadores.

```
mysql> select p.nom_p,p.ap_p,p.ap_m fr  
+-----+-----+-----+  
| nom_p | ap_p | ap_m |  
+-----+-----+-----+  
| Diego | Mercado | Tavira |  
| Miguel | Sanchez | Hernandez |  
| Patricio | Torres | De Paz |  
| Simon | Carlos | Cervantes |  
| Axel | Loza | Gonzalez |  
+-----+-----+-----+  
5 rows in set (0.17 sec)  
mysql>
```

Ilustración 46 Consulta 1

Práctica 2 (consulta)

Utilizando la BD Universidad, mostrar el nombre, apellido paterno y materno de los editores

Análisis.

¿Qué? Nombre, apellido materno, apellido paterno

¿Dónde? De editores.

Comando para ejecutar.

```
select p.nom_p,p.ap_p,p.ap_m from personas p, editores e where  
p.id_persona=e.id_persona;
```



```
mysql> select p.nom_p,p.ap_p,p.ap_m from per
+-----+-----+-----+
| nom_p      | ap_p      | ap_m      |
+-----+-----+-----+
| Jonathan   | Hernandez  | Gonzmán   |
| Angel      | Carro     | Apolinar  |
| Juan Pablo | Perez     | Ortega    |
| Miguel     | Rodriguez  | Hermida   |
| Jose de Jesus | Solorzano | Rangel    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Ilustración 47 Consulta 2

Descripción del Query.

Este nos indica que mostrara el nombre, apellido paterno y materno de los editores.

Práctica Multiconsulta Universidad

Con la BD universidad, mostrar el nombre de los artículos de publicación congresos que está en artículos.

Análisis.

¿Qué? Nombre del artículo.

¿Dónde? Publicación congresos.

Comando por ejecutar.

```
select a.titulo from articulos a, pubcongresos p where a.id_articulo=p.id_congreso;
```

Recuerda utilizar alias para referir a alguna entidad, en este caso utilizamos “a” para referir a pub congresos y “p” para artículos.

La descripción del comando indica que muestra el nombre de los artículos de publicación congreso

```
mysql> select a.titulo from artic
+-----+
| titulo |
+-----+
| El Medio Ambiente |
| El Nuevo Futuro |
| Influencia |
| Beneficios del Ejercicio |
| La Comunicacion Efectiva |
+-----+
5 rows in set (0.05 sec)

mysql>
```

Ilustración 48 Consulta 3

Práctica 4 (Consulta)

Con la BD universidad, mostrar el nombre de la persona y nombre de tema de investigadores.

Análisis.

¿Qué? Nombre de la persona

¿Dónde? Investigadores.

Comando por ejecutar.

```
select p.nom_p, t.nom_tema from asignatemas a, temas t, investigadores i, personas p
where i.id_investigadores=a.id_investigador and a.id_tema=t.id_temas and
p.id_persona=i.id_persona;
```

```
mysql> select p.nom_p, t.nom_tema from asignatemas a, temas t, investigadores i, personas p
where a.id_temas=t.id_temas and p.id_persona=i.id_persona;
+-----+-----+
| nom_p | nom_tema |
+-----+-----+
| Diego | Ciencias Naturales |
| Miguel | Matematicas |
| Patricio | Programacion en Python |
| Simon | Arduino |
| Axel | Programacion |
+-----+-----+
5 rows in set (0.06 sec)

mysql>
```

Ilustración 49 Consulta 4

Base de datos “Hospital”

Problemática de la base de datos.

Se desea diseñar una base de datos para guardar la información sobre médicos, empleados y pacientes de un centro de salud. De los médicos se desea saber su nombre, dirección, teléfono, población, provincia, código postal, NIF, número de la seguridad social, número de colegiado y si es médico titular, médico interino o médico sustituto. Cada médico tiene un horario en el que pasa consulta, pudiendo ser diferente cada día de la semana. Los datos de los médicos sustitutos no desaparecen cuando finalizan una sustitución, se les da una fecha de baja. Así, cada sustituto puede tener varias fechas de alta y fechas de baja, dependiendo de las sustituciones que haya realizado. Si la última fecha de alta es posterior a la última fecha de baja, el médico está realizando una sustitución en la actualidad en el centro de salud. El resto de los empleados son los ATS, ATS de zona, auxiliares de enfermería, celadores y administrativos. De todos ellos se desea conocer su nombre, dirección, teléfono, población, provincia, código postal, NIF y número de la seguridad social. De todos, médicos y empleados, se mantiene también información sobre los períodos de vacaciones que tienen planificados y de los que ya han disfrutado. Por último, de los pacientes se conoce su nombre, dirección, teléfono, código postal, NIF, número de la seguridad social y médico que les corresponde.

Diseño externo (maquetado).

Medicos

Nombre Apellido P Apellido M

Direccion Telefono Provincia

Codigo Postal NIF NSS

Titular
 Ingerterono
 Sustituto

Consulta Inicio Hora Minutos AM
 Consulta fin Hora Minutos PM

Num. de colegiado Fecha de alta

Fecha de baja Fecha vacacional

Ilustración 50 Maquetado

Diagrama entidad relación (BD abarrotos).

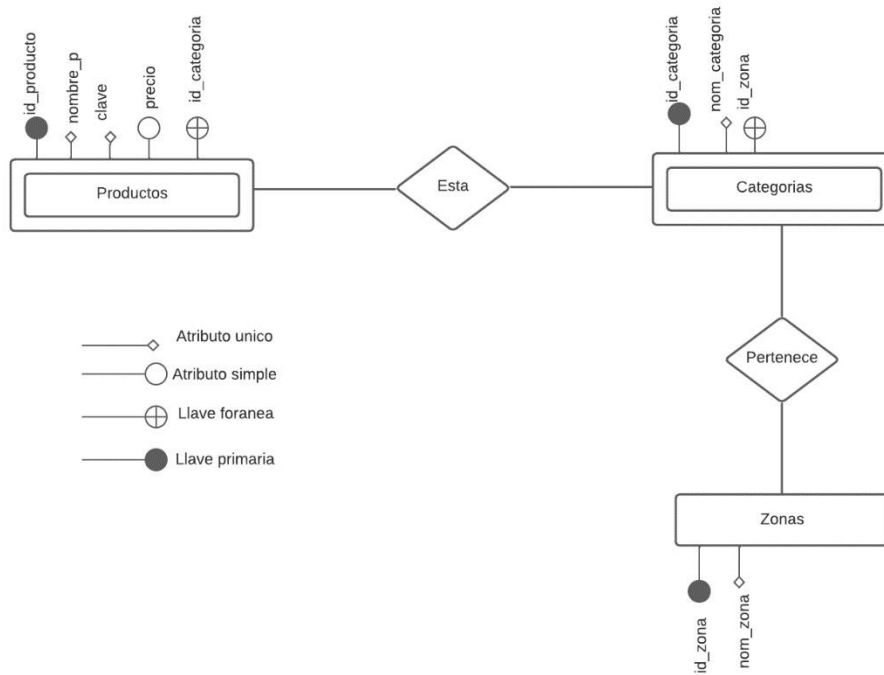


Ilustración 51 Modelo entidad relación

Pasos para realizar la práctica.

1._El primer paso es abrir el Mysql command line, ingresar la contraseña del usuario, digitar el comando *Create database Hospital;* este comando nos permite crear la base de datos, consultar su creación digitar el comando *show databases;* aparece un listado de todas las bases de datos que se han creado y entre ellas.

```
mysql> show databases;
+-----+
| Database |
+-----+
| escuela |
| estudiantes |
| hospital |
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| tienda |
| universidad |
| universidad1 |
| world |
+-----+
12 rows in set (0.18 sec)

mysql>
```

Ilustración 52 Revisar las BD

2._Una vez que corroboro la creación de la base de datos ahora, usar la base de datos con el comando *use hospital;* este comando permite ingresar a la base de datos creada, deberá de aparecer un mensaje de que se cargó la base de datos (lizhernandez, 2013).

```
mysql> use hospital;
Database changed
mysql>
```

Ilustración 53 Usar BD

3._En el siguiente paso es crear las tablas del modelo entidad relación indica, en este caso comenzar con nuestra primer tabla, el comando para realizarla es *create table personas (id_persona int primary key not null, nom_p varchar(100), ap_p varchar(100), am_p varchar(100), tel_p float, nif_p int unique, direccion varchar(100) unique, nss_p float, c_p int unique);*

Este comando se divide en varias secciones, la primera sección inicia en *create table personas*; esta parte significa crear una tabla llamada personas.

Al dar enter deberá aparecer el siguiente mensaje justamente debajo de tu comando, en caso de que no aparezca dicho mensaje será necesario revisar escritura o sintaxis (lizhernandez, 2013).

```
mysql> create table personas(id_persona int primary key not null, nom_p varchar(100), ap_p varchar(100), am_p varchar(100), tel_p float, nif_p int unique, direccion varchar(100) unique, nss_p float, c_p int unique);
Query OK, 0 rows affected (0.24 sec)

mysql>
```

Ilustración 54 Mensaje de creación de tabla personas

4._ Después de crear la tabla personas, crear las demás tablas que hay en el diagrama de entidad relación, al igual que la primera tabla solo debe ser una guía y saber si va un primary key, unique, not null entre otros.(Gittins, L. 2013).

6._Ya que las entidades (tablas) fueron creadas realizar la consulta de las entidades con el comando *show tables*; (Gittins, L. 2013).

```
mysql> show tables;
+-----+
| Tables_in_hospital |
+-----+
| asigna_fechas      |
| asignapv          |
| dias              |
| empleados         |
| fecha_altas       |
| fecha_bajas       |
| horarios          |
| medicos           |
| pacientes         |
| periodo_vacacional_disfrutado |
| periodo_vacacional_planificado |
| personal          |
| personas          |
| poblaciones       |
| provincias        |
| sustitutos        |
| tipo_e            |
| tipo_m            |
+-----+
18 rows in set (0.08 sec)

mysql>
```

Ilustración 55 Consulta de tablas de la base de datos.

7._El siguiente paso consiste en colocar las primary key y foreign key en las entidades correspondientes, iniciar declarando la primary key en el campo de id_persona de la entidad productos (para modificar una tabla es necesario el comando alter table, el comando para declarar el primary key *es alter table productos add constraint primary key(id_producto);* y deberá aparecer el siguiente mensaje de lo contrario será necesario revisar sintaxis (SiteGround, W. H. (s/f)).

```
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 56 Mensaje de creación primary key id_producto.

8._Ahora la primary key definida, retomar los dos campos que se colocó de color morado en el paso 3 para poder declarar la llave foránea (foreign key), digitar el comando *alter table productos add constraint foreign key* para poder tener llave foránea. SiteGround, W. H. (s/f)).

```
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 57 Mensaje de creación foreign key.

9._ Teniendo las tablas creadas introducir la información con un

insert into personas values(1,"Diego","Mercado","Javira", 7224668115, 000000511, "El manguito 66",0285728105,21200); así con todas nuestras tablas insertamos datos y vemos nuestras tablas (SiteGround, W. H. (s/f)).

```
mysql> insert into personas values(1,"Diego","Mercado","Tavira",7224668115,00000511,"El manguito 66"
Query OK, 1 row affected (0.14 sec)

mysql> show table personas;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version near 'personas' at line 1
mysql> select *from personas;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_persona | nom_p | ap_p | am_p | tel_p | nif_p | direccion | nss_p | c_p |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Diego | Mercado | Tavira | 7224670000 | 511 | El manguito 66 | 285728000 | 21200 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> |
```

Ilustración 58 Insertar datos.

10._Ya que se tiene la relación con las tablas colocar *select *from* y el nombre de cada tabla, para ver las tablas creadas ya con los datos insertados. (SiteGround, W. H.

```
mysql> select * from dias;
+-----+-----+
| id_asignadia | dia |
+-----+-----+
| 7 | Domingo |
| 4 | Jueves |
| 1 | Lunes |
| 2 | Martes |
| 3 | Miercoles |
| 6 | Sabado |
| 5 | Viernes |
+-----+-----+
7 rows in set (0.06 sec)

mysql> select * from empleados;
+-----+-----+-----+
| id_empleado | id_personal | id_tipo_e |
+-----+-----+-----+
| 1 | 4 | 1 |
| 2 | 5 | 5 |
| 3 | 1 | 1 |
| 4 | 2 | 3 |
| 5 | 3 | 4 |
+-----+-----+-----+
5 rows in set (0.07 sec)

mysql> select * from tipo_e;
+-----+-----+
| id_tipo_e | tipo_e |
+-----+-----+
| 5 | Administrativos |
| 1 | ATS |
| 2 | ATS de zona |
| 3 | Auxiliares de enfermeria |
| 4 | Celadores |
+-----+-----+

mysql> select * from horarios;
+-----+-----+-----+-----+-----+
| id_horario | id_asignadia | hora_inicio | hora_fin | id_medico |
+-----+-----+-----+-----+-----+
| 1 | 1 | 10:00 am | 6:30 pm | 1 |
| 2 | 2 | 7:00 am | 4:00 pm | 2 |
| 3 | 3 | 12:15 am | 8:30 pm | 3 |
| 4 | 4 | 09:10 am | 6:00 pm | 4 |
| 5 | 5 | 11:00 am | 7:00 pm | 5 |
+-----+-----+-----+-----+-----+
5 rows in set (0.12 sec)

mysql> select * from provincias;
+-----+-----+
| id_provincia | provincia |
+-----+-----+
| 1 | Guanajuato |
| 2 | Hidalgo |
| 3 | Jalisco |
| 4 | Oaxaca |
| 5 | Yucatan |
+-----+-----+
5 rows in set (0.09 sec)

mysql>
```

Ilustración 59 Consulta de registros de la tabla de categorías

11._Para verificar usar el comando *describe* y el nombre de la tabla que se quiere específicamente para verificar si se colocó bien los datos (*Comandos SQL Básicos*. (s/f)).


```
mysql> describe personas;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_persona | int | NO | PRI | NULL | |
| nom_p | varchar(100) | YES | | NULL | |
| ap_p | varchar(100) | YES | | NULL | |
| am_p | varchar(100) | YES | | NULL | |
| tel_p | float | YES | | NULL | |
| nif_p | int | YES | UNI | NULL | |
| direccion | varchar(100) | YES | UNI | NULL | |
| nss_p | float | YES | | NULL | |
| c_p | int | YES | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.07 sec)

mysql>
```

Ilustración 60 Descripción de la tabla productos.

12._Utilizar phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando un navegador web

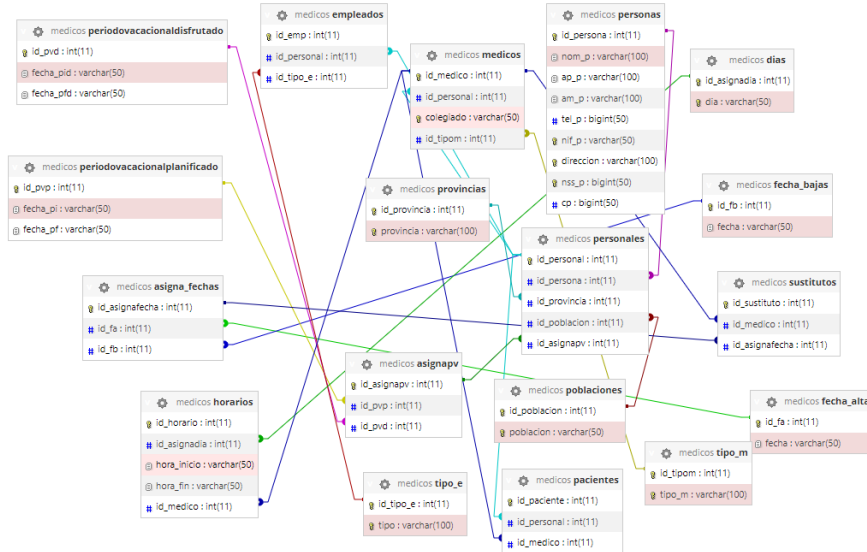


Ilustración 61 llaves foranes (relación)

Inserta los datos de la corrida de escritorio

La corrida

La consulta 1 ya está acomodada ya solo acomoda las demás que están en el documento que le entregamos a Jhona

Práctica Hospitales

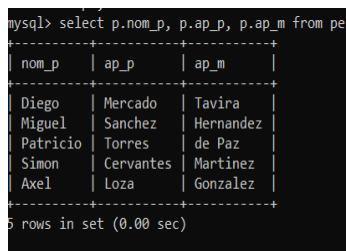
Utilizando la BD Hospital, Mostrar el nombre completo de los médicos titulares

¿Qué? Nombre, apellido paterno, apellido materno.

¿Dónde? medicos.

Comando por ejecutar.

```
select p.nom_p, p.ap_p, p.ap_m from personas p, personasles pe, medicos m where  
m.id_persona=pe.id_persona and pe.id_persona = p.id_persona;
```



```
mysql> select p.nom_p, p.ap_p, p.ap_m from per  
+-----+-----+-----+  
| nom_p | ap_p | ap_m |  
+-----+-----+-----+  
| Diego | Mercado | Tavira |  
| Miguel | Sanchez | Hernandez |  
| Patricio | Torres | de Paz |  
| Simon | Cervantes | Martinez |  
| Axel | Loza | Gonzalez |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

Ilustración 62 Consulta de nombre completo de médicos.

Descripción del Query.

Este comando muestra el nombre, apellido paterno y apellido materno de los médicos.

Unidad 4 NORMALIZACIÓN DE BASES DE DATOS

Normalización de Bases de Datos

Práctica Normalización.

Problemática.

- Realización de consultas SQL
- Modelo entidad/relación

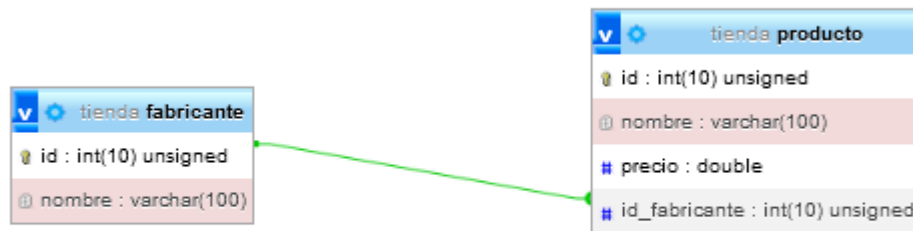


Ilustración 63 Modelo entidad/relación

Base de datos para mysql

En este apartado solo copiamos nuestra base de datos y la pasamos a phpMyAdmin para sacar el modelo entidad/relación

```

DROP DATABASE IF EXISTS tienda;
CREATE DATABASE tienda CHARACTER SET utf8mb4;
USE tienda;

CREATE TABLE fabricante (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL
);

CREATE TABLE producto (
  id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(100) NOT NULL,
  precio DOUBLE NOT NULL,
  id_fabricante INT UNSIGNED NOT NULL,
  FOREIGN KEY (id_fabricante) REFERENCES fabricante(id)
);

INSERT INTO fabricante VALUES(1, 'Asus');
INSERT INTO fabricante VALUES(2, 'Lenovo');
INSERT INTO fabricante VALUES(3, 'Hewlett-Packard');
INSERT INTO fabricante VALUES(4, 'Samsung');
INSERT INTO fabricante VALUES(5, 'Seagate');
INSERT INTO fabricante VALUES(6, 'Crucial');
INSERT INTO fabricante VALUES(7, 'Gigabyte');
INSERT INTO fabricante VALUES(8, 'Huawei');
INSERT INTO fabricante VALUES(9, 'Xiaomi');

INSERT INTO producto VALUES(1, 'Disco duro SATA3 1TB', 86.99, 5);
INSERT INTO producto VALUES(2, 'Memoria RAM DDR4 8GB', 120, 6);
INSERT INTO producto VALUES(3, 'Disco SSD 1 TB', 150.99, 4);
INSERT INTO producto VALUES(4, 'GeForce GTX 1050Ti', 185, 7);
INSERT INTO producto VALUES(5, 'GeForce GTX 1080 Xtreme', 755, 6);
INSERT INTO producto VALUES(6, 'Monitor 24 LED Full HD', 202, 1);
INSERT INTO producto VALUES(7, 'Monitor 27 LED Full HD', 245.99, 1);
INSERT INTO producto VALUES(8, 'Portátil Yoga 520', 559, 2);
INSERT INTO producto VALUES(9, 'Portátil Ideapd 320', 444, 2);
INSERT INTO producto VALUES(10, 'Impresora HP Deskjet 3720', 59.99, 3);
INSERT INTO producto VALUES(11, 'Impresora HP Laserjet Pro M26nw', 180, 3);

```

Ilustración 64 Base de Datos

Consultas

- 1.- Sacar el nombre y precio de los productos: *select upper(nombre), precio from producto;*

```
mysql> use tienda;
Database changed
mysql> select upper(nombre), precio from producto;
```

upper(nombre)	precio
DISCO DURO SATA3 1TB	86.99
MEMORIA RAM DDR4 8GB	120
DISCO SSD 1 TB	150.99
GEFORCE GTX 1050TI	185
GEFORCE GTX 1080 XTREME	755
MONITOR 24 LED FULL HD	202
MONITOR 27 LED FULL HD	245.99
PORTÁTIL YOGA 520	559
PORTÁTIL IDEAPD 320	444
IMPRESORA HP DESKJET 3720	59.99
IMPRESORA HP LASERJET PRO M26NW	180

```
11 rows in set (0.00 sec)
```

Ilustración 65 Consulta 1

2.- Sacar los precios más bajos de los productos: *select lower(nombre) as nom_min, precio from producto;*

```
mysql> select lower(nombre) as nom_min, precio from producto;
```

nom_min	precio
disco duro sata3 1tb	86.99
memoria ram ddr4 8gb	120
disco ssd 1 tb	150.99
geforce gtx 1050ti	185
geforce gtx 1080 xtreme	755
monitor 24 led full hd	202
monitor 27 led full hd	245.99
portátil yoga 520	559
portátil ideapd 320	444
impresora hp deskjet 3720	59.99
impresora hp laserjet pro m26nw	180

```
11 rows in set (0.00 sec)
```

Ilustración 66 Consulta 2

3.- Sacar el las iniciales de los productos: *select nombre, upper(substr(nombre,1,2)) as iniciales from fabricante;*

```
mysql> select nombre ,upper(substr
+-----+-----+
| nombre      | iniciales |
+-----+-----+
| Asus        | AS        |
| Lenovo      | LE        |
| Hewlett-Packard | HE        |
| Samsung     | SA        |
| Seagate     | SE        |
| Crucial     | CR        |
| Gigabyte    | GI        |
| Huawei     | HU        |
| Xiaomi     | XI        |
+-----+-----+
9 rows in set (0.00 sec)
```

Ilustración 67 Consulta 3

4.- Sacar el nombre de los que tienen el precio redondeado: *select nombre, round(precio) as Precio_redondeado from producto;*

```
mysql> select nombre, round(precio) as Precio_redondeado from producto;
+-----+-----+
| nombre      | Precio_redondeado |
+-----+-----+
| Disco duro SATA3 1TB | 87 |
| Memoria RAM DDR4 8GB | 120 |
| Disco SSD 1 TB | 150 |
| GeForce GTX 1050Ti | 185 |
| GeForce GTX 1080 Xtreme | 755 |
| Monitor 24 LED Full HD | 202 |
| Monitor 27 LED Full HD | 245 |
| Portátil Yoga 520 | 559 |
| Portátil Ideapd 320 | 444 |
| Impresora HP Deskjet 3720 | 59 |
| Impresora HP Laserjet Pro M26nw | 180 |
+-----+-----+
11 rows in set (0.00 sec)
```

Ilustración 68 Consulta 4

5.- Sacar el precio entero de los productos: *select nombre, truncate(precio,0) as precio_entero from producto;*

```
mysql> select nombre, truncate(precio,0) as precio_en
+-----+-----+
| nombre      | precio_entero |
+-----+-----+
| Disco duro SATA3 1TB | 86 |
| Memoria RAM DDR4 8GB | 120 |
| Disco SSD 1 TB | 150 |
| GeForce GTX 1050Ti | 185 |
| GeForce GTX 1080 Xtreme | 755 |
| Monitor 24 LED Full HD | 202 |
| Monitor 27 LED Full HD | 245 |
| Portátil Yoga 520 | 559 |
| Portátil Ideapd 320 | 444 |
| Impresora HP Deskjet 3720 | 59 |
| Impresora HP Laserjet Pro M26nw | 180 |
+-----+-----+
11 rows in set (0.00 sec)
```

Ilustración 69 Consulta 5

Práctica Cardinalidad

Problemática

- Un estudiante puede acceder a una beca
- Diagrama Entidad-relación.

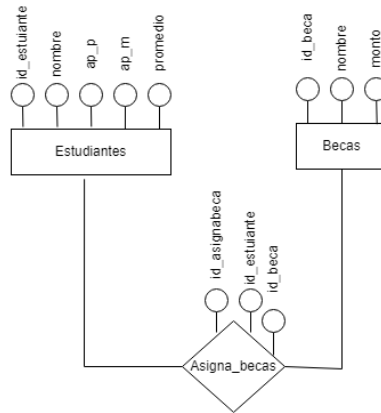


Ilustración 70 Diagrama E-R

Diagrama relacional.

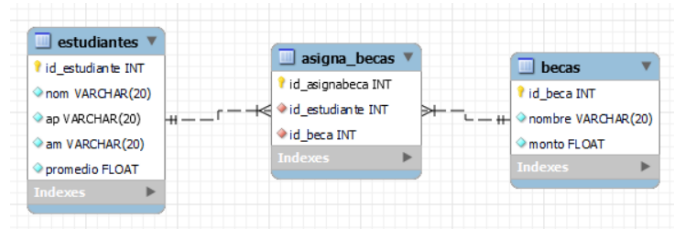


Ilustración 71 Diagrama Relacional

Creación de la base de datos

1. Creamos la base de datos llamada Práctica1

2. Creamos las tablas en este caso ocuparemos solo 3 que serán:

- Estudiantes
- Asignabecas
- becas

```

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

mysql> use ejerciciol
Database changed
mysql> create table estudiantes(id_estudiante int primary key, nom varchar(20) not null, ap varchar(20) not null, am varchar(20) not null, promedio float not null);
Query OK, 0 rows affected (0.03 sec)

mysql> create table becas(id_beca int primary key, nombre varchar(20) not null, monto float not null);
Query OK, 0 rows affected (0.01 sec)

mysql> create table asigna_becas(id_asignabeca int primary key, id_estudiante int unique not null, id_beca int not null);
Query OK, 0 rows affected (0.02 sec)

```

Ilustración 72 Creación DB

3. Agregar llaves foráneas.

```

mysql> alter table asigna_becas add constraint foreign key(id_estudiante) refer
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table asigna_becas add constraint foreign key(id_beca) references
ERROR 3734 (HY000): Failed to add the foreign key constraint. Missing column 'i
becas(id_beca);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe asigna_becas;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_asignabeca | int | NO | PRI | NULL | |
| id_estudiante | int | NO | UNI | NULL | |
+-----+-----+-----+-----+-----+-----+

```

Ilustración 73 Llaves Foráneas

4. Insertar datos en las tablas correspondientes.

```

mysql> insert into estudiantes values(1,"Mariaa");
ERROR 1136 (21501): Column count doesn't match value count at row 1
mysql> insert into estudiantes values(1,"Mariana","Cruz","Mendoza",92);
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes values(2,"David","Loza","Perez",70);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(3,"Pedro","Carranza","Mercado",90);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(4,"Juan","Menchoca","Arias",75);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(5,"Daniela","Lujano","Loza",65);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(6,"Javier","C.", "Rodriguez",72);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(7,"Martha","Almazán","Tavira",92);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(8,"Juan","Pérez","Armenta",93);
Query OK, 1 row affected (0.00 sec)

mysql> insert into estudiantes values(9,"Daniel","Juarez","Alvarado",76);
Query OK, 1 row affected (0.00 sec)

mysql> insert into becas values(1,"Pronabe",6000);
Query OK, 1 row affected (0.00 sec)

```

Ilustración 74 Inserción de datos

Consultas

1. Nombres completos de todos los estudiantes que tienen beca, consultar becas no utilizadas.

```
mysql> select e.nom,e.ap,e.am from estudiantes e, asigna_becas a where e.id_estudiante=a.id_estudiante;
+-----+-----+-----+
| nom   | ap    | am    |
+-----+-----+-----+
| Mariana | Cruz  | Mendoza |
| Pedro  | Carranza | Mercado |
| Martha | Almazán | Tavira |
| Juan   | Pérez | Armenta |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select nom from becas where id_becas not in(select id_beca from asigna_becas);
ERROR 1054 (42S22): Unknown column 'nom' in 'field list'
mysql> select nombre from becas where id_becas not in(select id_beca from asigna_becas);
ERROR 1054 (42S22): Unknown column 'id_becas' in 'IN/ALL/ANY subquery'
mysql> describe becas;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_beca | int           | NO   | PRI | NULL    |       |
| nombre  | varchar(20)   | NO   |     | NULL    |       |
| monto   | float         | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select nombre from becas where id_beca not in(select id_beca from asigna_becas);
+-----+
| nombre |
+-----+
| Mama soltera |
+-----+
1 row in set (0.00 sec)
```

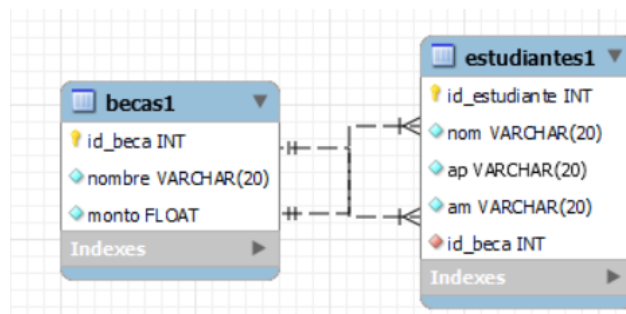
Ilustraci3n 75 Consulta 1

Prctica 3

Problemtica

1. Realizar el mismo ejemplo, pero sin la tabla asigna_becas, agregar (id_beca en la entidad estudiantes).
2. Est base de datos no est bien estructurada.

Diagrama relacional.



Ilustraci3n 76 Diagrama Relacional

1. Poner llave foránea en la tabla estudiantes1

```
mysql> alter table estudiantes1 add constraint foreign key(id_beca) references becas1(id_beca);
Query OK, 0 rows affected (0.08 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 77 Llave foránea

2. Inertar los datos

```
mysql> insert into becas1 values(1,"Pronabe",6000),(2,"Transporte",2500),(3,"Utiles",3000),(4,"Mama soltera",2500);
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into estudiantes1 values(1,"Mariana","Cruz","Mendoza",1),(2,"David","Loza","Perez",2),(3,"Pedro","Carranza","Mercado",3),(4,"Juan","Menchaca","Arias",2),(5,"Daniela","Laguna","Loza",4),(6,"Javier","Carro","Rodriguez",1),(7,"Martha","Aleman","Tavira",2),(8,"Juan","Perez","Armenta",2),(9,"Daniel","Juarez","Alvarado",2);
Query OK, 9 rows affected (0.00 sec)
Records: 9 Duplicates: 0 Warnings: 0
```

Ilustración 78 Inserción de datos

Consultas

1. Nombres completos de todos los estudiantes que tienen beca, consultar becas no utilizadas.

```
mysql> select nom,ap,am from estud
+-----+-----+-----+
| nom  | ap  | am  |
+-----+-----+-----+
| Mariana | Cruz | Mendoza |
| David  | Loza | Perez  |
| Pedro  | Carranza | Mercado |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select nom,ap,am from estud
+-----+-----+-----+
| nom  | ap  | am  |
+-----+-----+-----+
| Mariana | Cruz | Mendoza |
| David  | Loza | Perez  |
| Pedro  | Carranza | Mercado |
| Juan   | Menchaca | Arias  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

Ilustración 79 Consulta 1 y 2

Práctica 4

Problemática

- Los docentes imparten varias materias
- Diagrama entidad-relación.

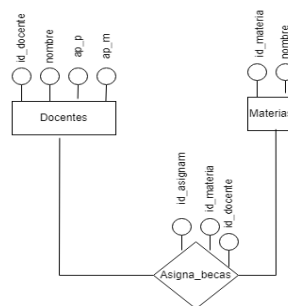


Ilustración 80 Diagrama E-R

Diagrama relacional.

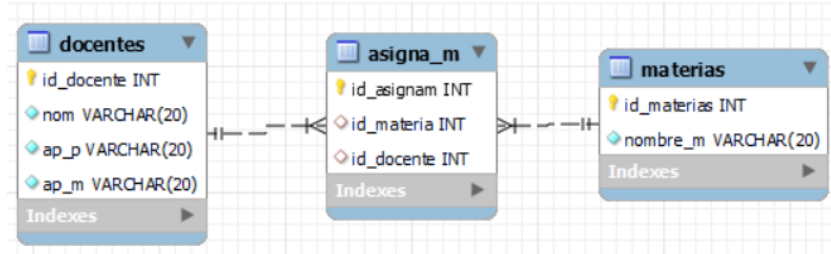


Ilustración 81 Diagrama Relacional

Creación de la base de datos

1. Creamos la base de datos llamada docentes

2. Crear las tablas en este caso ocuparemos solo 3 que serán:

- Docentes
- Imparte_materias
- Materias

```
mysql> create table docentes(id_docente int primary key, nom varchar(20)not null, ap varchar(20) not null, am varchar(20)not null);
Query OK, 0 rows affected (0.02 sec)

mysql> create table materias(id_materias int primary key, nombre_m varchar(20)not null unique);
Query OK, 0 rows affected (0.02 sec)

mysql> create table asigna_m(id_asignam int primary key, id_materia int, id_docente int);
Query OK, 0 rows affected (0.01 sec)
```

Ilustración 82 Creación de tablas

3. Se tienen que agregar las llaves foráneas con la relación que se tiene.

```
mysql> alter table asigna_m add constraint foreign key(id_materia) references materias(id_materias);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table asigna_m add constraint foreign key(id_docente) references docentes(id_docente);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Ilustración 83 Llaves foráneas

4. Se hace la inserción de datos en las tablas

```

mysql> insert into docentes values(1,"Juan","Pérez","Ortega");
Query OK, 1 row affected (0.01 sec)

mysql> insert into docentes values(2,"Jhoana","Marcos","Lopez"),(3,"Diego","Cazares","Lua"),(4,"Paola","Avila","Ponce"),(5,"Citlali","Aybar","Hernandez");
Query OK, 4 rows affected (0.01 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> insert into materias values(1,"Matematicas"),(2,"Fisica"),(3,"Biología"),(4,"Artes"),(5,"Geografía");
Query OK, 5 rows affected (0.00 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> insert into asigna_m values(1,1,1);
Query OK, 1 row affected (0.01 sec)

mysql> insert into asigna_m values(2,2,2),(3,3,2),(4,4,2),(5,5,1),(6,2,2),(7,3,3),(8,4,4),(9,5,5),(10,1,5);
Query OK, 9 rows affected (0.01 sec)
Records: 9 Duplicates: 0 Warnings: 0

```

Ilustración 84 Inserción de datos

Consulta

1. los nombres de las materias impartidas por el docente Juan.

```

mysql> select m.nombre_m from
+-----+
| nombre_m |
+-----+
| Matematicas |
| Geografía |
+-----+
2 rows in set (0.01 sec)

```

Ilustración 85 Consulta 1

Práctica 5

Problemática

- Los estudiantes cursan materias
- Diagrama entidad-relación.

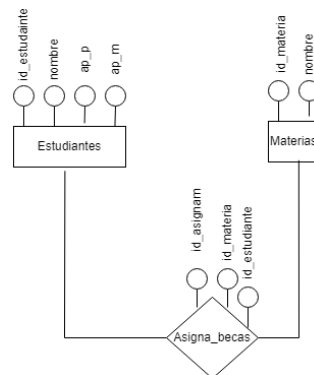


Ilustración 86 Diagrama E-R

Diagrama relacional.



Ilustración 87 Diagrama Relacional

Creación de la base de datos

1. Crear la base de datos llamada estudiantesmareias.

2. Crear las tablas en este caso usar solo 3 que serán:

- Estudiantes
- Asigna_m
- Materias

```
mysql> create table estudiantes(id_estudiante int primary key, nom varchar(20)not null, ap varchar(20)not null, am varchar(20)not null);
Query OK, 0 rows affected (0.01 sec)

mysql> create table materias(id_materia int primary key,nombre_m varchar(20)unique not null);
Query OK, 0 rows affected (0.02 sec)

mysql> create table asigna_m(id_asignam int primary key, id_estudiante int, id_materia int);
Query OK, 0 rows affected (0.02 sec)
```

Ilustración 88 Creación de tablas

3. Se tienen que agregar las llaves foráneas con la relación que se tiene.

```
mysql> alter table asigna_m add constraint foreign key(id_estudiante) references estudiantes(id_estudiante);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table asigna_m add constraint foreign key(id_materia) references materias(id_materia);
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

Ilustración 89 Llaves Foráneas

4. Se hace la inserción de datos en las tablas.

```

mysql> insert into estudiantes values(1,'Jose','Marcos','Vilchis');
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes values(2,'Lilian','Cruz','Campos');
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes values(3,'Jenny','Mendoza','Blanca');
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes values(4,'Carol','Cruz','Lopez');
Query OK, 1 row affected (0.01 sec)

mysql> insert into estudiantes values(5,'Diego','Hernandez','Casas');
Query OK, 1 row affected (0.01 sec)

```

Ilustración 90 Inserción de datos

Consulta

1. Nombres de las materias que los estudiantes cursan

```

mysql> select m.nombre_m from est
+-----+
| nombre_m |
+-----+
| Matematicas |
| Fisica |
| Artes |
+-----+
3 rows in set (0.01 sec)

```

Ilustración 91 Consulta 1

Unidad 5 ALGEBRA RELACIONAL

Práctica Algebra relacional

1.-Mostrar el título de las películas para mayores de 18 años

Procedural

$\pi_{Peliculas.titulo}(Peliculas \times Clasificaciones)_{Clasificaciones.desc_clas="Mayores de 18"}$

Sentencia SQL:

*select peliculas.titulo from peliculas, clasificaciones where
peliculas.id_clas=clasificaciones.id_clas and clasificaciones.desc_clas = "Apta para
mayores de 18 años. ";*



Ilustración 92 Consulta 1

Consulta 2

2.-Mostrar los días en que se proyecta "pinocho"

Procedural

$\pi_{Dias.desc_dia}(Peliculas \ X \ AsignaHorarios \ X \ Horarios \ X \ dias)_{Peliculas.titulo="Pinocho"}$

Sentencia SQL

```
select dias_ss.desc_dia from dias_ss,horarios,asigna_horarios,peliculas where  
horarios.id_dia=dias_ss.id_dia and horarios.id_horario=asigna_horarios.id_horario  
and asigna_horarios.id_pelicula=peliculas.id_pelicula and  
peliculas.titulo="Pinocho";
```



Ilustración 93 Consulta 2

Consulta 3

3.- Mostrar los datos de las salas que tienen descuento de estudiante

Procedural

$\sigma_{descuento.desc_desc="Dia del estudiante"}(salas \ X \ cines \ X \ asigna_descuentos \ X \ descuentos)$

Sentencia SQL

```
select salas.* from salas,asigna_descuentos,cines,descuentos where  
descuentos.desc_desc="Dia del estudiante" and  
asigna_descuentos.id_cine=cines.id_cine and salas.id_cine=cines.id_cine and  
descuentos.id_descuento=asigna_descuentos.id_descuento;
```

	id_sala	deso_sala	precio	id_cine
	2	General	50	3

Ilustración 94 Consulta 3

Consulta 4

4.-Mostrar los datos de las protagonistas de titanic

Procedural:

$\sigma_{\text{peliculas.titulo}=\text{"Titanic"}}(\text{Protagonistas X AsignaProtagonistas X Peliculas})$

Sentencia SQL:

select protagonistas. from protagonistas, asigna_protagonistas, peliculas where protagonistas.id_protagonista=asigna_protagonistas.id_protagonista and peliculas.titulo="Titanic" and asigna_protagonistas.id_pelicula=peliculas.id_pelicula;*

	id_protagonista	nom_p	ap_p	am_p
	1	Javier	Gonzalez	Garcia
	2	Juan	Jaramillo	Benitez
	3	Manuel	Perez	Hernandez

Ilustración 95 Consulta 4

Consulta 5

5.-Mostrar el número de cines que tiene Guadalajara

Procedural:

$\pi_{\text{Count}(*)}(\text{Ciudades X Cines})_{\text{Ciudades.nom}=\text{"Guadalajara"}}$

Sentencia SQL:


```

select      cines.numero      from      ciudades,cines      where
ciudades.id_ciudad=cines.id_ciudad      and
ciudades.nom_ciudad="Guadalajara";

```

Opciones extra

	numero
<input type="checkbox"/> Editar Copiar Borrar	341
<input type="checkbox"/> Editar Copiar Borrar	258
<input type="checkbox"/> Editar Copiar Borrar	302
<input type="checkbox"/> Editar Copiar Borrar	441
<input type="checkbox"/> Editar Copiar Borrar	91
<input type="checkbox"/> Editar Copiar Borrar	215
<input type="checkbox"/> Editar Copiar Borrar	469
<input type="checkbox"/> Editar Copiar Borrar	477
<input type="checkbox"/> Editar Copiar Borrar	63
<input type="checkbox"/> Editar Copiar Borrar	232
<input type="checkbox"/> Editar Copiar Borrar	211
<input type="checkbox"/> Editar Copiar Borrar	170

Ilustración 96 Consulta 5

Unidad 6 INTRODUCCIÓN A SQL

Prácticas 1

Lenguaje natural

- Mostrar el nombre de la sala, la capacidad y boletos disponibles que tienen

Sentencia SQL

- ```

SELECT c.nom_ciudad AS Ciudad, s.desc_sala AS Tipo_sala, p.titulo
AS Pelicula, CONCAT(d.desc_dia, ' ', h.hi, '-',h.ht) AS Horario
FROM peliculas p, ciudades c INNER JOIN cines n ON
n.id_ciudad=c.id_ciudad INNER JOIN salas s ON
s.id_cine=n.id_cine INNER JOIN asigna_peliculas a ON
a.id_sala=s.id_sala INNER JOIN asigna_horarios i ON
a.id_asignah=i.id_asignah INNER JOIN horarios h ON
i.id_horario=h.id_horario INNER JOIN dias_ss d ON

```

*h.id\_dia=d.id\_dia WHERE i.id\_pelicula=p.id\_pelicula ORDER BY(s.desc\_sala);*

| Ciudad      | Tipo_sala | Pelicula                    | Horario               |
|-------------|-----------|-----------------------------|-----------------------|
| Monterrey   | 3D        | La pasión de Cristo         | Martes 13:00-15:00    |
| Monterrey   | 3D        | Pinocho                     | Lunes 16:00-18:00     |
| Monterrey   | 3D        | La La Land                  | Viernes 13:00-15:00   |
| Monterrey   | 3D        | El rey Arturo               | Domingo 20:00-22:00   |
| Monterrey   | 3D        | Forrest Gump                | Domingo 16:00-18:00   |
| Monterrey   | 3D        | El golpe                    | Domingo 13:00-15:00   |
| Monterrey   | 3D        | La isla siniestra           | Sábado 13:00-15:00    |
| Monterrey   | General   | La vida es bella            | Miercoles 20:00-22:00 |
| Monterrey   | General   | El silencio de los corderos | Martes 20:00-22:00    |
| Monterrey   | General   | La pasión de Cristo         | Martes 13:00-15:00    |
| Monterrey   | General   | Last X                      | Lunes 16:00-18:00     |
| Monterrey   | General   | El golpe                    | Domingo 16:00-18:00   |
| Monterrey   | General   | Titanic                     | Sábado 16:00-18:00    |
| Guadalajara | Sala 4DX  | El resplandor               | Sábado 13:00-15:00    |
| Guadalajara | Sala 4DX  | El Rey León                 | Jueves 16:00-18:00    |

Ilustración 97 Consulta

## Lenguaje natural

- Mostrar las películas cuyo horario después de las 6:00pm

## Sentencia SQL

- *select titulo from peliculas, horarios, asignahorarios where asignahorarios.id\_pelicula=peliculas.id\_pelicula and asignahorarios.id\_horario=horarios.id\_horario and horarios.hi>"18:00";*

✓ Mostrando filas 0 - 13 (total de 14, La consulta tardó 0,0328 segundos.)

```
select titulo from peliculas, horarios, asignahorarios where asignahorarios.id_pelicula=peliculas.id_pelicula and
asignahorarios.id_horario=horarios.id_horario and horarios.hi>"18:00";
```

| titulo                                      |
|---------------------------------------------|
| Fast X                                      |
| La vida es bella                            |
| El exorcista                                |
| La La Land                                  |
| El Gran Hotel Budapest                      |
| Titanic                                     |
| El Señor de los Anillos: El retorno del Rey |
| El silencio de los corderos                 |
| El sexto sentido                            |
| Psicosis                                    |
| Avatar                                      |
| Forrest Gump                                |
| El rey Arturo                               |
| El Padrino                                  |

Ilustración 98 Consulta

## Prácticas 2

### Sentencia SQL

- *select sum (venta\_boletos.cantidad) from venta\_boletos,tickets where date (Tickets.fecha)="19/06/2023" and venta\_boletos.id\_ticket=Tickets.id\_ticket;*

| SUM(venta_boletos.cantidad) |
|-----------------------------|
| 6                           |

Ilustración 99 Consulta

### Sentencia SQL

- *select sum (venta\_boletos.cantidad) as total from venta\_boletos inner join tickets on venta\_boletos.id\_ticket=tickets.id\_ticket where date (tickets.fecha)="2023-19-06";*

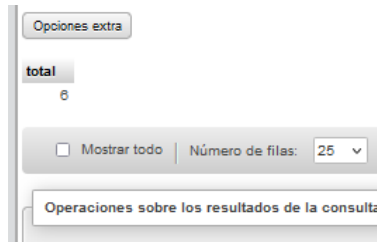


Ilustración 100 Consulta

## Sentencia SQL

- $$\text{select sum (cantidad) FROM venta_boletos WHERE id_ticket in (select id_ticket FROM tickets WHERE date (fecha)='19-06-2023');$$

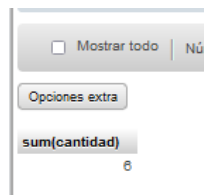


Ilustración 101 Consulta

## Sentencia SQL

- $$\text{select asiento - (select sum(cantidad) from venta_boletos were id_ticket in (select id_ticket from tickets where date(fecha)='19-06-2023')) from salas where id_sala=1;}$$

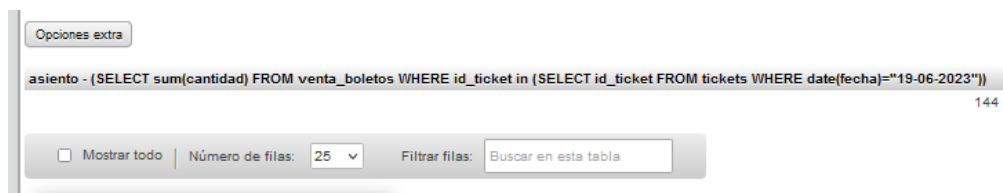


Ilustración 102 Consulta

## Prácticas 3

### 5 consultas

1.-Mostrar la calle del cine y salas que se encuentran en CDMX

## Sentencia SQL

- *select c.calle, s.desc\_sala from salas s, cines c, ciudades ci where s.id\_cine = c.id\_cine and ci.id\_ciudad = c.id\_ciudad and ci.nom\_ciudad like "%CDMX%";*

| Opciones extra   |                           |
|------------------|---------------------------|
| calle            | desc_sala                 |
| Calle Cuauht?moc | Sala Premium Large Format |

Ilustración 103 Consulta

2.-Mostrar que días pasa pinocho

### Sentencia SQL

- *select p.titulo, d.desc\_dia from peliculas p, asignahorarios ah, horarios h, dias d where p.id\_pelicula = ah.id\_pelicula and ah.id\_horario = h.id\_horario and d.id\_dia = h.id\_dia and p.titulo like "%pino%";*

| Opciones extra |          |
|----------------|----------|
| titulo         | desc_dia |
| Pinocho        | Lunes    |

Ilustración 104 Consulta

3.-Mostrar la la descripción día y el número de ticket

### Sentencia SQL

- *select p.titulo, d.desc\_dia, count(\*) as tickets\_disponibles from peliculas p, asigna\_horarios ah, horarios h, dias d where p.id\_pelicula = ah.id\_pelicula and ah.id\_horario = h.id\_horario and d.id\_dia = h.id\_dia and p.titulo like '%avatar%' group by p.titulo, d.desc\_dia;*

| TITULO | DESC_DIA | TICKETS_DISPONIBLES |
|--------|----------|---------------------|
| Avatar | Viernes  | 1                   |

Ilustración 105 Consulta

4.- Mostrar el nombre completo del director el título de la película con más ventas.

#### Sentencia SQL

- *select di.nom\_d, p.titulo, sum(vb.cantidad) cantidad from directores di inner join peliculas pe on di.id\_director=pe.id\_director inner join asigna\_horarios ah on pe.id\_pelicula=ah.id\_pelicula inner join asigna\_peliculas ap on ah.id\_asigna\_horario=ap.id\_asigna\_horario inner join venta\_boletos vb on ap.id\_asigna\_pelicula=vb.id\_asigna\_pelicula order by sum(vb.cantidad) limit 1;*

| nom_d  | titulo        | cantidad |
|--------|---------------|----------|
| Manuel | La cenicienta | 6        |

Ilustración 106 Consulta

5.- Muestra la descripción de la sala en la que se proyectará la película de La cenicienta.

### Sentencia SQL

- *select s.desc\_sala from salas s join asignapeliculas ap on s.id\_sala = ap.id\_sala join asignahorarios ah on ap.id\_asignapelicula = ah.id\_asignahorario join peliculas p on ap.id\_asignahorario = p.id\_pelicula where p.titulo = 'La cenicienta';*

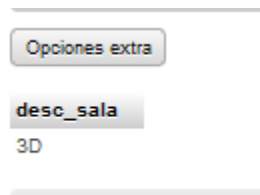


Ilustración 107 Consulta

### Practica Consultas complejas

1.-Mostrar la calle del cine y salas que se encuentran en CDMX

#### Inner join:

```
select c.calle, s.desc_sala from salas s inner join cines c on s.id_cine = c.id_cine inner join ciudades ci on ci.id_ciudad = c.id_ciudad where ci.nom_ciudad like '%cdmx%';
```

#### Subconsulta:

```
select c.calle, s.desc_sala from salas s join cines c on s.id_cine = c.id_cine where c.id_ciudad in (select ci.id_ciudad from ciudades ci where ci.nom_ciudad like '%cdmx%');
```

#### Multiconsulta:

```
select c.calle, s.desc_sala from (select c.id_cine, c.calle from cines c join ciudades ci on
ci.id_ciudad = c.id_ciudad where ci.nom_ciudad like '%cdmx%') as c join salas s on
s.id_cine = c.id_cine;
```

Opciones extra

| calle            | desc_sala                 |
|------------------|---------------------------|
| Calle Cuauht?moc | Sala Premium Large Format |

Ilustración 108 Consulta

2.-Mostrar que días proyectan

pinocho **Inner Join:**

```
select p.titulo, d.desc_dia from peliculas p inner join asignahorarios ah on p.id_pelicula
= ah.id_pelicula inner join horarios h on ah.id_horario = h.id_horario inner join dias d
on d.id_dia = h.id_dia where p.titulo like '%pino%';
```

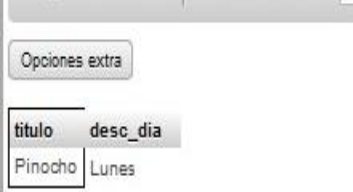
**Subconsulta:**

```
select p.titulo, d.desc_dia from peliculas p join asignahorarios ah on p.id_pelicula =
ah.id_pelicula join horarios h on ah.id_horario = h.id_horario join dias d on d.id_dia =
h.id_dia where p.titulo like '%pino%';
```

**Multiconsulta:**



```
select p.titulo, d.desc_dia from peliculas p join asignahorarios ah on p.id_pelicula =
ah.id_pelicula join horarios h on ah.id_horario = h.id_horario join dias d on d.id_dia =
h.id_dia where p.titulo like '%pino%';
```



| titulo  | desc_dia |
|---------|----------|
| Pinocho | Lunes    |

Ilustración 109 Consulta

3.-Mostrar la descripción día y el numo del ticket

**Inner join:**

```
select p.titulo, d.desc_dia, count(*) as tickets_disponibles from peliculas p inner join
asigna_horarios ah on p.id_pelicula = ah.id_pelicula inner join horarios h on
ah.id_horario = h.id_horario inner join dias d on d.id_dia = h.id_dia where p.titulo like
'%avatar%' group by p.titulo, d.desc_dia;
```

**Subconsulta:**

```
select p.titulo, d.desc_dia, count(*) as tickets_disponibles from peliculas p,
asigna_horarios ah, horarios h, dias d where p.id_pelicula = ah.id_pelicula and
ah.id_horario = h.id_horario and d.id_dia = h.id_dia and p.titulo like '%avatar%' group
by p.titulo, d.desc_dia;
```

**Multiconsulta:**

```

select p.titulo, d.desc_dia, count(*) as tickets_disponibles from peliculas p join
asigna_horarios ah on p.id_pelicula = ah.id_pelicula

join horarios h on ah.id_horario = h.id_horario, dias d where d.id_dia = h.id_dia

and p.titulo like '%avatar%' group by p.titulo, d.desc_dia;

```

Opciones extra

| TITULO | DESC_DIA | TICKETS_DISPONIBLES |
|--------|----------|---------------------|
| Avatar | Viernes  | 1                   |

Ilustración 110 Consulta

4.- Mostrar el nombre completo del director el título de la película con más ventas.

**Inner join:**

```

select di.nom_d, p.titulo, sum(vb.cantidad) cantidad

from directores di inner join peliculas pe on di.id_director = pe.id_director

inner join asignahorarios ah on pe.id_pelicula = ah.id_pelicula

inner join asignapeliculas ap on ah.id_asignahorario = ap.id_asignahorario

inner join venta_boletos vb on ap.id_asignapelicula = vb.id_asignapelicula

order by sum(vb.cantidad) limit 1;

```

**Subconsulta:**

```

select di.nom_d, postitulo, cantidad
from (select di.nom_d, postitulo, sum(vb.cantidad) cantidad from directores di
inner join peliculas pe on di.id_director = pe.id_director
inner join asignahorarios ah on pe.id_pelicula = ah.id_pelicula
inner join asignapeliculas ap on ah.id_asignahorario = ap.id_asignahorario
inner join venta_boletos vb on ap.id_asignapelicula = vb.id_asignapelicula
group by di.nom_d, postitulo order by sum(vb.cantidad) limit 1) as subconsulta;

```

### Multiconsulta:

```

select di.nom_d, postitulo, sum(vb.cantidad) cantidad
from directores di, peliculas pe, asignahorarios ah, asignapeliculas ap, venta_boletos vb
where di.id_director = pe.id_director and pe.id_pelicula = ah.id_pelicula
and ah.id_asignahorario = ap.id_asignahorario and ap.id_asignapelicula =
vb.id_asignapelicula group by di.nom_d, postitulo order by sum(vb.cantidad) limit 1;

```

| nom_d  | titulo        | cantidad |
|--------|---------------|----------|
| Manuel | La cenicienta | 6        |

Ilustración 111 Consulta

5.- Muestra la descripción de la sala en la que se proyectará la película de La cenicienta.

### Inner join:

```
select s.desc_sala from salas s join asignapeliculas ap on s.id_sala = ap.id_sala

join asignahorarios ah on ap.id_asignapelicula = ah.id_asignahorario

join peliculas p on ap.id_asignahorario = p.id_pelicula

where p.titulo = 'La cenicienta';
```

### Subconsulta:

```
select s.desc_sala from salas s where s.id_sala in (select ap.id_sala from
asignapeliculas ap join asignahorarios ah on ap.id_asignapelicula =
ah.id_asignahorario join peliculas p on ap.id_asignahorario = p.id_pelicula

where p.titulo = 'La cenicienta');
```

### Multiconsulta:

```
select s.desc_sala

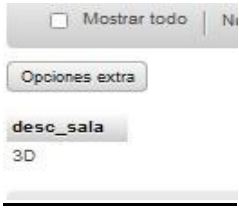
from (select s.id_sala

from salas s join asignapeliculas ap on s.id_sala = ap.id_sala

join asignahorarios ah on ap.id_asignapelicula = ah.id_asignahorario

join peliculas p on ap.id_asignahorario = p.id_pelicula

where p.titulo = 'La cenicienta') as sala_id join salas s on s.id_sala = sala_id.id_sala;
```



*Ilustración 112 Consulta*

## Conclusión.

Como se pudo observar detrás de cada proceso de información hay un diagrama el cual te ayuda a organizar la implementación en una base de datos. Una base de datos esta empleada hoy en día en todas las áreas, desde una abarrotería, hasta una empresa de alta gama, tiene un amplio campo donde se puede aplicar por lo tanto es importante conocer, desarrollar e implementar las habilidades correctas.

Mediante esta documentación pudimos indagar más a fondo sobre los conceptos básicos en este modelo, aprender a utilizarlo y a desarrollarlo de una buena manera. La materia "Fundamentos de Bases de Datos" es fundamental para cualquier persona interesada en comprender y trabajar con bases de datos. Al concluir esta asignatura, se pueden extraer varias conclusiones importantes:

Importancia de las bases de datos: Las bases de datos son fundamentales en casi todas las aplicaciones y sistemas que almacenan y gestionan información. Comprender los fundamentos de las bases de datos es esencial para diseñar, administrar y utilizar eficientemente estos sistemas.

## Referencias web

A, Q. ((s/f)). *Modificar tablas en MySQL*. Obtenido de Styde.net:  
<https://styde.net/modificar-tablas-en-mysql-mariadb/>

Castillo. (26 de noviembre de 2020). *Prácticas de base de datos*. Obtenido de Slideshare.net: <https://es.slideshare.net/hectoralfonsovelazquezalfonso/22-Prácticas-base-de-datos>

Donoso, M. (08 de febrero de 2021). *Bases de Datos*. Obtenido de TuInstitutoOnline.com:  
<https://www.tuinstitutoonline.com/cursos/baseavanzado1>

Gallardo, P. (10 de enero de 2018). *Modelo entidad relación*. Obtenido de Esic.edu:  
<https://www.esic.edu/rethink/tecnologia/modelo-entidad-relaciondescripcion-aplicaciones>

García., G. (07 de mayo de 2007). *Introducción a las bases de datos*. Obtenido de Desarrolloweb.com: <https://desarrolloweb.com/articulos/introduccion-basedatos.html>

Hernandez. (13 de marzo de 2021). *Arquitectura de las bases de datos*. Obtenido de Desarrolloweb.com: <https://desarrolloweb.com/articulos/arquitectura-base-de-datos.html>

Hernandez, L. (04 de abril de 2013). *Comandos en MySQL*. Obtenido de [www.uv.mx](http://www.uv.mx):  
<https://www.uv.mx/personal/lizhernandez/files/2013/04/Comandos-mysql.pdf>

L, G. (13 de septiembre de 2013). *Como insertar registros en MySQL*. Obtenido de [techlandia.com](http://techlandia.com):  
[https://techlandia.com/insertar-registros-mysqlcomo\\_400299/](https://techlandia.com/insertar-registros-mysqlcomo_400299/)

Microsoft. (25 de octubre de 2022). *Conceptos básicos sobre bases de datos*. Obtenido de [microsoft.com](http://microsoft.com): <https://support.microsoft.co/es-es/office/conceptos-básicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204#:~:text=Una%20base%20de%20datos%20es,programa%20de%20procesamiento%20de%20texto>

P, G. (05 de noviembre de 2013). *Fundamento de las bases de datos*. Obtenido de [Gembeta.com](http://Gembeta.com): <https://www.genbeta.com/desarrollo/fundamento-de-las-bases-de-datosmodelo-entidad-relacion>

Plus, H. (23 de abril de 2021). *Clasificación de las bases de datos*. Obtenido de [www.hostingplus.mx](http://www.hostingplus.mx): <https://www.hostingplus.mx/blog/clasificacion-de-las-bases-de-datos-conoce-todos-los-tipos/>

Ramirez, O. (06 de marzo de 2022). *Modelos de Datos*. Obtenido de [Google.com](http://Google.com):  
<https://sites.google.com/site/orlandoramirezfbd4sc4a/unidad-i/1-4-modelos>