Host : b51yewg1zyoujlyul3zg-mysql.services.clever-cloud.com

User: uix7vmqdztdwbv2y

Password: WS8wEtDLRM41z5V9p4Hx

Name database: b51yewg1zyoujlyul3zg

Orden de tablas.

1. Clientes

create table clientes(

nombre varchar(60) not null ,

apellidoP varchar(60) not null ,

apellidoM varchar(60) not null ,

dirCalle varchar(150) not null ,

dirNum int not null ,

dirColonia varchar(150) not null ,

dirCP int not null ,

dirCiudad varchar(150) not null ,

correoElectronico varchar(220),

telefono bigint not null ,

idCliente int not null auto\_increment primary key

);

1. Almacenes

create table almacenes(

idAlmacen int not null auto\_increment primary key ,

dirCalle varchar(150) not null ,

dirNum int not null ,

dirColonia varchar(150) not null ,

dirCP int not null ,

dirCiudad varchar(150) not null ,

correoAlmacen varchar(220) not null ,

telefono bigint not null

);

1. Autores

create table autores(

idAutor int not null auto\_increment,

nombre varchar(60) not null,

apellidoP varchar(60) not null ,

apellidoM varchar(60) not null ,

dirCalle varchar(150) not null ,

dirNum int not null ,

dirColonia varchar(150) not null ,

dirCP int not null ,

dirCiudad varchar(150) not null ,

correoAutor varchar(220) not null,

telefono bigint not null ,

primary key (idAutor)

);

1. Libro

create table libro (

id int not null auto\_increment,

titulo varchar(100) not null ,

isbn int not null unique ,

fechaPublicacion date not null ,

descripcion enum('Libro') not null ,

autor int not null ,

existente int not null ,

volumen int not null,

primary key (id),

foreign key (autor) references autores(idAutor)

);

1. Revista

create table revista (

id int not null auto\_increment,

titulo varchar(100) not null ,

isbn int not null unique ,

fechaPublicacion date not null ,

descripcion enum('Revista') not null ,

numeroEdicion int not null ,

existente int not null ,

volumen int not null,

primary key (id)

);

1. Ebook

create table Ebook (

id int not null auto\_increment,

titulo varchar(100) not null ,

isbn int not null unique ,

fechaPublicacion date not null ,

descripcion enum('Ebook') not null ,

dispositivo enum('Movil' , 'Tablet','Pc') ,

existente int not null ,

primary key (id),

so enum('Android' , 'IOS')

);

1. Libros (datos libros y almacén)

create table libros (

idAlmacen int not null ,

idLibro int ,

idRevista int ,

idEbook int,

foreign key (idAlmacen) references almacenes(idAlmacen),

foreign key (idLibro) references libro(id),

foreign key (idRevista) references revista(id),

foreign key (idEbook) references Ebook(id)

);

1. Carrito

create table carrito(

idCarrito int not null auto\_increment primary key ,

fechaCompra date not null ,

libro int not null ,

ejemplares int not null ,

cliente int not null ,

foreign key (cliente) references clientes(idCliente),

foreign key (libro) references libros(idLibro),

foreign key (libro) references libros(idRevista),

foreign key (libro) references libros(idEbook)

);

Consultas.

(Insertar un cliente)

insert into clientes (nombre, apellidoP, apellidoM,

dirCalle, dirNum, dirColonia, dirCP, dirCiudad,

correoElectronico, telefono) values (

'Luis Mario','Cervantes','Suarez',

'Francisco Orozco',20,'Revolucion',91100,'Xalapa',

'l.mario.cs31@gmail.com',2281155853

);

select idCliente,nombre,apellidoM from clientes;

Salida:



(Registrar un almacen)

insert into almacenes (dirCalle, dirNum, dirColonia, dirCP, dirCiudad,

correoAlmacen, telefono) values (

'Avila Camacho',23,'Centro',91200,'Xalapa',

'almacenAC@uv.libros.mx',2281155853

);

select idAlmacen,dirCalle,dirCiudad from almacenes;

Salida:



(Agregar un Ebook)

insert into Ebook (titulo, isbn,

fechaPublicacion, descripcion,

dispositivo, existente, so) values (

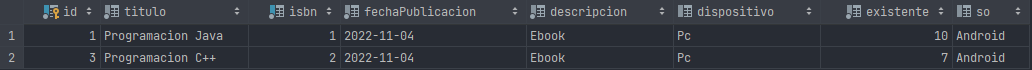
'Programacion Java',00001,

'2022-11-4','Ebook','Pc',10,'Android'

);

select \* from Ebook;

salida:



(Guardar libros en su almacen)

insert into libros (idAlmacen, idEbook) values (

1,3

);

*select* idAlmacen,idEbook *from* libros *where* idAlmacen=1; *#mostrar los id de los Ebook que hay en el almacen 1*

*select* dirCalle,idEbook,titulo *from* libros,Ebook,almacenes *where* id=idEbook *AND* libros.idAlmacen=almacenes.idAlmacen; *#muestra los libros Ebook en los almacenes*

*select* libros.idAlmacen,dirCalle,idEbook,titulo *from* libros,Ebook,almacenes *where* libros.idAlmacen=almacenes.idAlmacen *AND* Ebook.id=libros.idEbook *and* libros.idAlmacen=1; *#Muestra los libros de cierto almacen*

Variante ↑

Instrucciones.

Mostrar el id del almacén y su dirección de calle, también el nombre el ebook junto con su id y la cantidad que hay disponible. (TODOS LOS ALMACENES)

Paso 1. Extracción de datos.

Queremos el id del almacén en el que guardamos el libro, entonces necesitamos la tabla “Libros”

Queremos la dir.Calle del almacén, el dato se encuentra en la tabla “Almacenes”

El nombre del Ebook esta en su misma tabla, también su id y la cantidad disponible, así que usaremos la tabla de “ebook”

Paso 2. Identificar los atributos que usaremos

TABLA LIBROS: libros.idAlmacen

TABLA ALMACENES: almacenes.dirCalle

TABLA EBOOK: ebook.titulo, ebook.id, ebook.existente

Paso 3. Estructurar la consulta

Si queremos ver “el id del almacén y su dirección de calle, también el nombre el ebook junto con su id y la cantidad que hay disponible”

Entonces lo metemos en el select. ↓

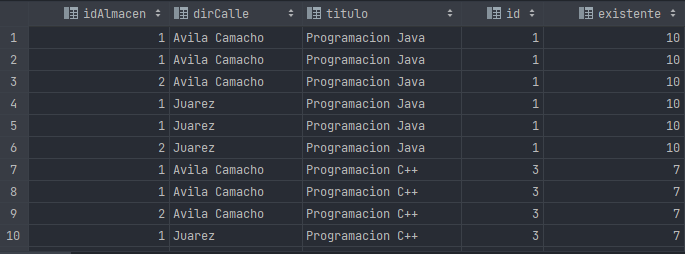
Select idAlmacen, dirCalle, titulo, id, cantidad from…

A continuación, ponemos el nombre de las tablas donde están nuestros datos. (paso 1)

Select libros.idAlmacen, dirCalle, titulo, id, existente from libros,almacenes,Ebook;

(nota: en libros.idAlmacen, se declara asi porque el atributo se repite en libros y almacenes y con la instrucción resaltada indicamos de que tabla queremos el dato)

Hasta ahora la consulta nos da la siguiente salida.



La consulta se realiza correctamente, pero los resultados no son los que queremos ver, asi que para arreglar esto nos pasamos a la sentencia where.

Para esto podemos probar con un

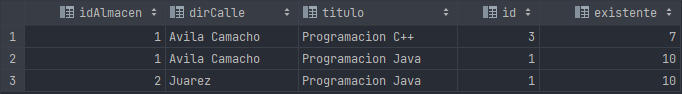
where libros.idAlmacen = almacenes.idAlmacen y Ebook.id=idEbook

“Cuando libros.idAlmacen sea igual a almacenes.idAlmacen y Ebook.id sea igual a idEbook(atributo de la tabla libros) “

La consulta es:

Select libros.idAlmacen, dirCalle, titulo, id, existente from libros,almacenes,Ebook where libros.idAlmacen = almacenes.idAlmacen and Ebook.id=idEbook;

Salida:



Ahora que sabes el proceso para obtener los libros que hay en los almacenes, hacer el siguiente ejercicio

Instrucciones.

Mostrar el id del almacén y su dirección de calle, también el nombre el ebook junto con su id y la cantidad que hay disponible. (del almacén “Juarez”)

Como salida debes obtener:



Solución: <https://docs.google.com/document/d/1hk4u6hpJSPCsYQiP6suy-SJI6Z7siT6fi9W7cmOvrVc/edit?usp=sharing>