

Desarrollo de Software

Practica 2.6 – Sistema de Facturación

Jesús Alberto Aréchiga Carrillo

22310439 4N

Profesor

José Luis García Cerpas

Mayo 2024

Guadalajara, Jalisco

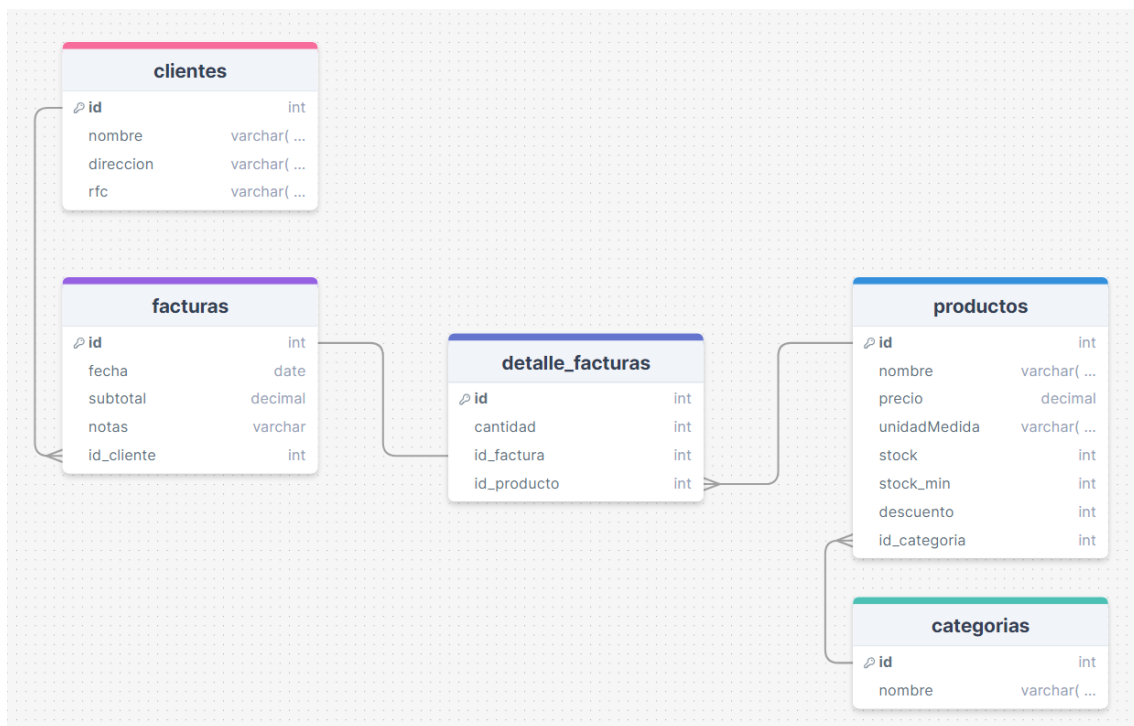
Introducción

Un sistema de facturación es un conjunto de procesos y herramientas utilizadas por las empresas para generar, enviar y administrar facturas por los bienes y servicios que ofrecen a sus clientes. Este sistema automatiza tareas como el cálculo de impuestos, el seguimiento de pagos y la generación de reportes financieros.

Es importante porque facilita la gestión financiera al registrar de manera precisa las transacciones comerciales, lo que ayuda a garantizar el cumplimiento de las obligaciones fiscales y a mantener un registro ordenado de los ingresos y gastos. Además, agiliza los procesos de facturación y cobro, lo que mejora la eficiencia operativa y la satisfacción del cliente.

Desarrollo

Primero hay que definir el diagrama relacional para poder hacer la base de datos física:



Ya teniendo el diagrama relacional, es posible hacer la base de datos física con sus respectivas relaciones.

Creación de la base de datos:

```
drop database if exists facturacion;  
  
create database facturacion;
```

```
use facturacion;

create table if not exists categorias (
id int primary key auto_increment,
nombre varchar(60)
);
create table if not exists productos (
id int not null auto_increment,
nombre varchar(60) null default null,
precio decimal(10,0) null default null,
unidadMedida varchar(60) null default null,
id_categoria int,
stock int,
stock_min int,
descuento int,
foreign key(id_categoria) references categorias(id),
primary key (id)
);

create table if not exists modo_pago(
id int primary key,
nombre varchar(60),
detalles varchar (60)
);

create table if not exists clientes(
id int not null default '0',
nombre varchar(60) null default null,
direccion varchar(60) null default null,
rfc varchar(15) null default null,
modo_pago int,
primary key (id),
foreign key (modo_pago) references modo_pago(id)
);

create table if not exists facturas(
id int not null auto_increment,
fecha date null default null,
id_cliente int null default null,
subtotal decimal(10,0) null default null,
notas varchar(60) default null,
foreign key (id_cliente) references clientes(id)
on delete cascade,
primary key (id)
);

create table if not exists detalle_fac(
id int not null auto_increment,
id_factura int null default null,
```

```

id_producto int null default null,
cantidad int null default null,
foreign key (id_factura) references facturas(id)
on delete cascade,
foreign key (id_producto) references productos(id)
on delete cascade,
primary key(id)
);

insert into categorias(id, nombre) values (1, 'Herramienta Manual'),
(2, 'Desarmadores');

insert into modo_pago(id, nombre, detalles) values (1, '4189334533345',
'Debito'),
(2, '123445677', 'Credito');

insert into clientes(id, nombre, direccion, rfc, modo_pago) values (1,
'Juan', 'conocido', 'aaax6666', 1),
(2, 'Pedro', 'conocido', 'aaax6666', 2),
(3, 'Oscar', 'conocido', 'aaax6666', 2);

insert into productos(id, nombre, precio, unidadMedida, id_categoria,
stock, stock_min, descuento) values
(1, 'Mazo', 90, 'pieza', 1, 100, 100, 60),
(2, 'Pinza', 75, 'pieza', 1, 50, 100, 60),
(3, 'Desarmador', 100, 'pieza', 2, 80, 100, 60),
(4, 'Martillo', 50, 'pieza', 1, 50, 20, 10),
(5, 'Taladro', 89, 'pieza', 1, 10, 10, 20);

insert into facturas values (1, '2024-05-02', 1, 500, 'Compra de 5
desarmadores'),
(2, '2024-05-02', 1, 165, 'Compra de una pinza y un mazo');

insert into detalles_fac values (1, 1, 1, 3),
(2, 1, 2, 1),
(3, 1, 3, 5);

```

Al tener este script, es posible empezar a hacer las operaciones deseadas:

Primero se crean facturas para después poder ingresar el detalle de la factura, agregando el producto.

	id	fecha	id_cliente	subtotal	notas			
<input type="checkbox"/>	Edit	Copy	Delete	1	2024-05-02	1	500	Compra de 5 desarmadores
<input type="checkbox"/>	Edit	Copy	Delete	2	2024-05-02	1	165	Compra de una pinza y un mazo

☐ Check all With selected: Edit Copy Delete Export

☐ Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

Console

Press Ctrl+Enter to execute query
 >insert into facturas values (1, '2024-05-02', 1, 500, 'Compra de 5 desarmadores'), (2, '2024-05-02', 1, 165, 'Compra de una pinza y un mazo');
 >select * from facturas;

	id	id_factura	id_producto	cantidad			
<input type="checkbox"/>	Edit	Copy	Delete	1	1	1	5

☐ Check all With selected: Edit Copy Delete Export

☐ Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Console

Press Ctrl+Enter to execute query
 >insert into detalle_fac values (1,1,1,5);
 >select * from detalle_fac;

De esta manera se pueden ir ingresando todos los datos necesarios para completar todas las facturas.

Para obtener la información, es necesario unir ciertos datos de diferentes tablas, por ejemplo, facturas, detalle de facturas, y productos.

nombre	cantidad	precio	total
Desarmador	5	100	500
Pinza	1	75	75
Mazo	1	90	90

☐ Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Console

Press Ctrl+Enter to execute query
 >select productos.nombre, detalle_fac.cantidad, productos.precio, (productos.precio*detalle_fac.cantidad) as total from productos, detalle_fac where detalle_fac.id_producto = productos.id and id_factura=1;

Para mostrar cual factura es la que se está viendo, se tiene que agregar la columna deseada, en este caso, sólo se agrega "id_factura" en el apartado de SELECT.

id_factura	nombre	cantidad	precio	total
1	Desarmador	5	100	500
1	Pinza	1	75	75
1	Mazo	1	90	90

☐ Show all Number of rows: 25 Filter rows: Search this table

Query results operations

Console

Press Ctrl+Enter to execute query
 >select id_factura, productos.nombre, cantidad, productos.precio, (productos.precio * detalle_fac.cantidad) as total from productos, detalle_fac where detalle_fac.id_producto = productos.id and id_factura=1;

Para mostrar el nombre del cliente y su dirección junto con los la ID y la fecha de la factura 1:

nombre	direccion	id	fecha
Juan	conocido	1	2024-05-02

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

☐ Print

☐ Copy to clipboard

☐ Export

☐ Display chart

☐ Create view

Console

Press Ctrl+Enter to execute query

>select clientes.nombre, clientes.direccion, facturas.id, facturas.fecha from facturas, clientes where clientes.id = facturas.id_cliente and facturas.id=1;

Para actualizar el total de la factura y mostrarla en la tabla de facturas:

	id	fecha	id_cliente	subtotal	notas
<input type="checkbox"/>	1	2024-05-02	1	665	Compra de 5 desarmadores
<input type="checkbox"/>	2	2024-05-02	1	165	Compra de una pinza y un mazo

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Sort by key: None

Query results operations

Console

Press Ctrl+Enter to execute query

>update facturas set subtotal = (select sum(productos.precio * detalle_fac.cantidad) from productos, detalle_fac where detalle_fac.id_producto=productos.id and facturas.id = 1) where facturas.id = 1;

>select * from facturas;

Para mostrar la fecha de vencimiento de cada factura a nombre del respectivo cliente:

id	fecha	vencimiento	nombre
1	2024-05-02	2024-06-01	Juan
2	2024-05-02	2024-06-01	Juan

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

Console

Press Ctrl+Enter to execute query

>select facturas.id, facturas.fecha, date_add(facturas.fecha, interval 30 day) as vencimiento, clientes.nombre from facturas, clientes where facturas.id_cliente=clientes.id;

Para mostrar los días que han pasado desde la creación de cierta factura, se utiliza datediff(exp1, exp2) donde exp1 y exp2 son las fechas y se hace la resta de exp1 a exp2. En este caso se utiliza now() como exp1:

id	fecha	dias	nombre
1	2024-05-02	0	Juan
2	2024-05-02	0	Juan

☐ Show all | Number of rows: 25 | Filter rows:

Query results operations

Console

Press Ctrl+Enter to execute query

>select facturas.id, facturas.fecha, datediff(now(), facturas.fecha) as dias, clientes.nombre from clientes, facturas where facturas.id_cliente=clientes.id;

Para mostrar la cantidad de productos por factura y su precio:

id_factura	cliente	producto	cantidad	precio
1	Juan	Desarmador	5	100
1	Juan	Pinza	1	75
1	Juan	Mazo	1	90

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Console

Press Ctrl+Enter to execute query

```
> select id_factura, clientes.nombre as cliente, productos.nombre as producto, cantidad, precio from clientes
  inner join facturas on facturas.id_cliente=clientes.id --esto solo para que muestre a clientes que tienen facturas
  inner join detalle_fac on detalle_fac.id_factura=facturas.id --para aquellas facturas que tengan productos agregados
  inner join productos on productos.id=detalle_fac.id_producto; --obtener el nombre del producto
```

Para mostrar el producto con el menor precio:

	id	nombre	precio
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Martillo	50

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

Console

Press Ctrl+Enter to execute query

```
> select id, nombre, precio from productos where precio = (SELECT min(precio) from productos);
```

Conclusión

Hay muchas formas muy variadas para utilizar las búsquedas y actualizaciones, incluso sería posible insertar datos nuevos haciendo operaciones con información de otras tablas.

En el caso del sistema de facturación, es más simple obtener las búsquedas que se necesitan componiendo una búsqueda con funciones, o también haciendo una búsqueda que utilice más de una tabla.