Consultas SQL - SQLite3

Repositorio Bases de datos para aplicaciones: practicas con SQL

Tablas

Creación de la tabla Clientes:

```
CREATE TABLE clientes (
    id_cliente integer PRIMARY KEY AUTOINCREMENT,
    nombre varchar(50),
    email varchar(50)
);

INSERT INTO clientes(nombre,email)
VALUES
('Dejah','dejah@email.com'),
('Jonh','jonh@email.com');
```

Consulta

```
SELECT * FROM clientes;
```

Resultado

```
id_cliente  nombre      email
----------  ----------  ---------------
1           Dejah       dejah@email.com
2           Jonh        jonh@email.com
```

Creación de la tabla Productos:

```
CREATE TABLE productos(
    id_producto integer PRIMARY KEY AUTOINCREMENT,
    producto varchar(50),
    precio_unitario float
);
```

```
INSERT INTO productos(producto,precio_unitario)
VALUES
('Lápiz',5),
('Libreta',20);
```

Consulta

```
SELECT * FROM productos;
```

Resultadp

```
id_producto  producto    precio_unitario
-----------  ----------  ---------------
1            Lápiz       5.0
2            Libreta     20.0
```

Ventas

```
CREATE TABLE ventas(
    id_venta integer PRIMARY KEY AUTOINCREMENT,
    fecha date,
    id_cliente integer REFERENCES clientes(id_cliente)
);

INSERT INTO ventas(fecha,id_cliente)
VALUES
('2020/01/01',1),
('2020/01/02',1),
('2020/01/03',2);
```

SELECT * FROM ventas;

```
id_venta    fecha       id_cliente
----------  ----------  ----------
1           2020/01/01  1
2           2020/01/02  1
3           2020/01/03  2
```

Detalle ventas

```
CREATE TABLE detalle_ventas(
    id_detalle_venta integer PRIMARY KEY AUTOINCREMENT,
```

```
    id_venta integer REFERENCES ventas(id_venta),
    id_producto integer REFERENCES productos(id_producto),
    cantidad_producto integer,
    precio_unitario float,
    total_x_producto float
);


INSERT INTO
detalle_ventas(id_venta,id_producto,cantidad_producto,precio_unitario,total_x_prod
ucto)
VALUES
(1,1,2,5,10),
(1,2,10,20,200),
(2,2,1,20,20),
(3,1,10,5,50),
(3,2,10,20,200);
```

SELECT * FROM detalle_ventas;

```
id_detalle_venta  id_venta    id_producto  cantidad_producto  precio_unitario
total_x_producto
----------------  ----------  -----------  -----------------  ---------------  ---
-------------
1                 1           1            2                  5.0
10.0
2                 1           2            10                 20.0
200.0
3                 2           2            1                  20.0
20.0
4                 3           1            10                 5.0
50.0
5                 3           2            10                 20.0
200.0
```

Proveedores

```
CREATE TABLE proveedores (
    id_proveedor integer PRIMARY KEY AUTOINCREMENT,
    proveedor varchar(50),
    nombre_contacto varchar(100),
    email_contacto varchar(50)
);

INSERT INTO proveedores(proveedor,nombre_contacto,email_contacto)
VALUES
('ACME','Bruce Wayne','bruce@acme.com'),
('Cloud9','Diana Prince','diana@cloud9.com');
```

SELECT * FROM proveedores;

```
id_proveedor  proveedor   nombre_contacto  email_contacto
------------  ----------  ---------------  --------------
1             ACME        Bruce Wayne      bruce@acme.com
2             Cloud9      Diana Prince     diana@scribe.c
```

Compras

```
CREATE TABLE compras(
    id_compra integer PRIMARY KEY AUTOINCREMENT,
    fecha date,
    id_proveedor integer REFERENCES proveedores(id_proveedor)
);

INSERT INTO compras(fecha,id_proveedor)
VALUES
('2020/01/01',1),
('2020/02/02',2),
('2020/03/03',2);
```

SELECT * FROM compras;

```
id_compra   fecha       id_proveedor
----------  ----------  ------------
1           2020/01/01  1
2           2020/02/02  2
3           2020/03/03  2
```

Detalle de Compras

```
CREATE TABLE detalle_compras(
    id_detalle_compra integer PRIMARY KEY AUTOINCREMENT,
    id_compra integer REFERENCES compras(id_compra),
    id_producto integer REFERENCES productos(id_producto),
    cantidad_producto integer,
    precio_unitario float,
    total_x_producto float
);


INSERT INTO
detalle_compras(id_compra,id_producto,cantidad_producto,precio_unitario,total_x_pr
oducto)
VALUES
```

```
(1,1,100,5,500),
(1,2,150,20,3000),
(2,1,200,5,1000),
(2,2,250,20,5000),
(3,1,300,5,600);
```

SELECT * FROM detalle_compras;

```
id_detalle_compra  id_compra   id_producto  cantidad_producto  precio_unitario
total_x_producto
-----------------  ----------  -----------  -----------------  ---------------  --
--------------
1                  1           1            100                5.0
500.0
2                  1           2            150                20.0
3000.0
3                  2           1            200                5.0
1000.0
4                  2           2            250                20.0
5000.0
5                  3           1            300                5.0
600.0
```

---

1. Consultas Ventas

Consulta 1:

Mostrar id_cliente, nombre, email, fecha, id_venta, id_producto, producto, cantidad_producto,precio_unitario,total_producto para cada detalle_venta

Script SQL

```
select clientes.id_cliente, clientes.nombre, clientes.email, ventas.fecha,
ventas.id_venta, productos.id_producto,
productos.producto,detalle_ventas.id_detalle_venta,
detalle_ventas.cantidad_producto,detalle_ventas.precio_unitario,detalle_ventas.tot
al_x_producto

from clientes,ventas,detalle_ventas, productos

WHERE clientes.id_cliente = ventas.id_cliente AND
ventas.id_venta = detalle_ventas.id_venta AND
detalle_ventas.id_producto = productos.id_producto;
```

Resultado

```
id_cliente  nombre      email            fecha       id_detalle_venta  id_venta
id_producto  producto    cantidad_producto  precio_unitario  total_x_producto
----------  ----------  ---------------  ----------  ----------------  ----------
----------  ----------  -----------------  --------------  ----------------
1           Dejah       dejah@email.com  2020/01/01  1                 1
1            Lápiz       2                  5.0             10.0
1           Dejah       dejah@email.com  2020/01/01  2                 1
2            Libreta     10                 20.0            200.0
1           Dejah       dejah@email.com  2020/01/02  3                 2
2            Libreta     1                  20.0            20.0
2           Jonh        jonh@email.com   2020/01/03  4                 3
1            Lápiz       10                 5.0             50.0
2           Jonh        jonh@email.com   2020/01/03  5                 3
2            Libreta     10                 20.0            200.0
```

2. **Consulta 2:** Mostrar el total_venta por cada venta

```
id_venta    total_venta
----------  -----------
1           210.0
2           20.0
3           250.0
```

3. **Consulta 3:** Mostrar el nombre del cliente y total_venta por cada venta

```
nombre      id_venta    total_venta
----------  ----------  -----------
Dejah       1           210.0
Dejah       2           20.0
Jonh        3           250.0
```

4. **Consulta 4:** Mostrar el nombre del cliente y el total que pagado

```
nombre      total_venta
----------  -----------
Dejah       230.0
Jonh        250.0
```

5. **Consulta 5:** Mostrar la cantidad total de productos vendida por cada producto

```
producto    cantidad_producto
----------  -----------------
```

```
Libreta     21
Lápiz       12
```

6. **Consulta 6:** Mostrar el total vendido por dia

```
fecha       total_venta
----------  -----------
2020/01/01  210.0
2020/01/02  20.0
2020/01/03  250.0
```

7. **Consulta 7:** Mostrar el dia que menos se ha vendido

```
fecha       total_venta
----------  -----------
2020/01/02  20.0
```

Consultas Compras

---

8. **Consulta 8:** Mostrar id_proveedor, proveedor, nombre_contacto, email_contacto, fecha, id_compra, id_producto, producto, cantidad_producto,precio_unitario,total_producto para cada **detalle_compra**

```
id_proveedor  proveedor   nombre_contacto  email_contacto  fecha       id_compra
id_producto  producto       cantidad_producto  precio_unitario  total_x_producto
------------  ----------  ---------------  --------------  ----------  ----------
-----------  -------------  -----------------  ---------------  ----------------
1             ACME        Bruce Wayne      bruce@acme.com  2020/01/01  1
1            Lápiz acme 2H  100                5.0              500.0
1             ACME        Bruce Wayne      bruce@acme.com  2020/01/01  1
2            Libreta scrib  150                20.0             3000.0
2             Cloud9      Diana Prince     diana@scribe.c  2020/02/02  2
1            Lápiz acme 2H  200                5.0              1000.0
2             Cloud9      Diana Prince     diana@scribe.c  2020/02/02  2
2            Libreta scrib  250                20.0             5000.0
2             Cloud9      Diana Prince     diana@scribe.c  2020/03/03  3
1            Lápiz acme 2H  300                5.0              600.0
```

9. **Consulta 9:** Mostrar el total_compra por cada **compra**

```
id_compra   total_compra
----------  ------------
1           3500.0
2           6000.0
3           600.0
```

10. **Consulta 10:** Mostrar el proveedor, nombre_contacto, email_contacto y total_compra por cada compra

```
proveedor   nombre_contacto  email_contacto  id_compra   total_compra
----------  ---------------  --------------  ----------  ------------
ACME        Bruce Wayne      bruce@acme.com  1           3500.0
Cloud9      Diana Prince     diana@scribe.c  2           6000.0
Cloud9      Diana Prince     diana@scribe.c  3           600.0
```

11. **Consulta 11:** Mostrar el proveedor y el total que se le ha comprado

```
proveedor   total_compra
----------  ------------
ACME        3500.0
Cloud9      6600.0
```

12. **Consulta 12:** Mostrar la cantidad total de productos comprados por cada producto

```
producto                   cantidad_producto
-------------------------  -----------------
Libreta scribe profesional  400
Lápiz acme 2H               600
```

13. **Consulta 13:** Mostrar el total comprado por día

```
fecha       total_compra
----------  ------------
2020/01/01  3500.0
2020/02/02  6000.0
2020/03/03  600.0
```

14. **Consulta 14:** Mostrar el dia que más se ha comprado

```
fecha       total_compra
----------  ------------
2020/02/02  6000.0
```

Consultas Triggers

---

15. **Consulta 15:** Modificar mediante sql la estructura de la tabla **productos** e insertar el campo existencias de tipo entero y con un valor default de 100.

Tabla productos antes de modificar su estructura

```
id_producto  producto       precio_unitario
-----------  -------------  ---------------
1            Lápiz acme 2H  5.0
2            Libreta scrib  20.0
```

Tabla productos después de insertar existencias

```
id_producto  producto       precio_unitario  existencias
-----------  -------------  ---------------  -----------
1            Lápiz acme 2H  5.0              100
2            Libreta scrib  20.0             100
```

16. **Consulta 16:** Crear un trigger que después de insertar un producto en detalle_ventas, dejando preci_unitario y total_x_producto con un valor de 0, actualice el precio_unitario del producto insertado trayendolo directmente de la tabla productos.

Ejemplo consulta SQL:

```
INSERT INTO
detalle_ventas(id_venta,id_producto,cantidad_producto,precio_unitario,total_x_prod
ucto)
VALUES (1,1,2,0,0);
```

Resultado de la consulta:

```
id_detalle_venta  id_venta    id_producto  cantidad_producto  precio_unitario
total_x_producto
----------------  ----------  -----------  -----------------  ---------------  ---
-------------
1                 1           1            2                  5.0
10.0
2                 1           2            10                 20.0
200.0
3                 2           2            1                  20.0
20.0
```

```
4                3           1           10                  5.0
50.0
5                3           2           10                  20.0
200.0
6                1           1           2                   5.0             0.0
```

17. **Consulta 17** Crear un trigger que después de insertar un producto en detalle_ventas, actualice las existencias de productos:

Existencias de productos:

```
id_producto  producto       precio_unitario  existencias
-----------  -------------  ---------------  -----------
1            Lápiz acme 2H  5.0              100
2            Libreta scrib  20.0             100
```

Nuevo detalle_venta

```
INSERT INTO
detalle_ventas(id_venta,id_producto,cantidad_producto,precio_unitario,total_x_prod
ucto)
VALUES (1,1,2,0,0);
```

Existencias actualizadas

```
id_producto  producto       precio_unitario  existencias
-----------  -------------  ---------------  -----------
1            Lápiz acme 2H  5.0              98
2            Libreta scrib  20.0             100
```

18. **Consulta 18** Crear un trigger que después de actualizar el detalle_ventas, actualice las total_x_prodcutos con la operación total_x_producto = cantidad_producto * precio_unitario:

Detalle ventas antes del TRIGGER

```
id_detalle_venta  id_venta    id_producto  cantidad_producto  precio_unitario
total_x_producto
----------------  ----------  -----------  -----------------  ---------------  ---
-------------
1                 1           1            2                  5.0
10.0
2                 1           2            10                 20.0
200.0
3                 2           2            1                  20.0
20.0
```

```
4                3           1           10                5.0
50.0
5                3           2           10                20.0
200.0
6                1           1           2                 5.0             0.0
7                1           1           2                 5.0             0.0
```

Insertar nuevo detalle_ventas

```
INSERT INTO
detalle_ventas(id_venta,id_producto,cantidad_producto,precio_unitario,total_x_prod
ucto)
VALUES (1,1,10,0,0);
```

Tabla detalle_ventas actualizada con el trigger.

```
id_detalle_venta  id_venta    id_producto  cantidad_producto  precio_unitario
total_x_producto
----------------  ----------  -----------  -----------------  ---------------  ---
-------------
1                 1           1            2                  5.0
10.0
2                 1           2            10                 20.0
200.0
3                 2           2            1                  20.0
20.0
4                 3           1            10                 5.0
50.0
5                 3           2            10                 20.0
200.0
6                 1           1            2                  5.0
10.0
7                 1           1            2                  5.0
10.0
8                 1           1            10                 5.0
50.0
```

Transactions

COMMIT

```
CREATE TABLE clientes (
    id_cliente integer PRIMARY KEY AUTOINCREMENT,
```

```
    nombre varchar(50),
    email varchar(50)
);

INSERT INTO clientes(nombre,email)
VALUES
('Dejah','dejah@email.com'),
('Jonh','jonh@email.com');
```

SELECT * FROM clientes;

```
1|Dejah|dejah@email.com
2|Jonh|jonh@email.com
```

Transaction COMMIT

```
BEGIN TRANSACTION;

INSERT INTO clientes(nombre,email)
VALUES
('Jane','jane@email.com');

COMMIT;
```

SELECT * FROM clientes;

```
1|Dejah|dejah@email.com
2|Jonh|jonh@email.com
3|Jane|jane@email.com
```

Nota: COMMIT aplica la transaccion

ROLLBACK

```
CREATE TABLE clientes (
    id_cliente integer PRIMARY KEY AUTOINCREMENT,
    nombre varchar(50),
    email varchar(50)
);

INSERT INTO clientes(nombre,email)
VALUES
```

```
('Dejah','dejah@email.com'),
('Jonh','jonh@email.com');
```

SELECT * FROM clientes;

```
1|Dejah|dejah@email.com
2|Jonh|jonh@email.com
```

Transaction ROLLBACK

```
BEGIN TRANSACTION;

INSERT INTO clientes(nombre,email)
VALUES
('Jane','jonh@email.com');

ROLLBACK;
```

SELECT * FROM clientes;

```
1|Dejah|dejah@email.com
2|Jonh|jonh@email.com
```

Nota: ROLLBACK anula la transacion

PERMISSIONS

```
create database demo_users;

use demo_users;


create table clientes(
id int primary key auto_increment,
name varchar(50) not null
);

insert into clientes(name)
values
('Jane'),
('John');

select * from clientes;
```

Crear usuario

```
CREATE USER 'invitado'@'localhost' IDENTIFIED BY '123456';
```

Asignar permisos

```
GRANT ALL PRIVILEGES ON demo_users.* TO 'administrador'@'localhost';
```

Aplicar cambios

```
FLUSH PRIVILEGES;
```

Tipos de permisos

- **ALL** PRIVILEGES- asigna todos los permisos
- **CREATE** Permite crear nuevas bases o tablas.
- **DROP** Permite eliminar bases o tablas.
- **DELETE** Permite borrar bases o tablas.
- **INSERT** Permite insertar nuevos registros en las tablas.
- **SELECT** Permite seleccionar registros de las tablas.
- **UPDATE** Permite actualizar datos de los registros de las tablas.
- **GRANT OPTION** Permite dar o quitar permisos a los usuarios.

Quitar permisos

```
REVOKE Select ON demo_users.* TO 'invitado'@'localhost';
```

Mostrar lista de permisos

```
SHOW GRANTS FOR 'invitado'@'localhost';
```

Eliminar usuario

```
DROP USER 'invitado'@'localhost';
```