--

-- File generated with SQLiteStudio v3.3.2 on Fri Feb 10 14:30:03 2023

--

-- Text encoding used: UTF-8

--

-- Table: company\_data

DROP TABLE IF EXISTS company\_data;

CREATE TABLE company\_data (

Company\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

Name TEXT,

Invoice\_text TEXT,

Branch\_name TEXT,

Tax\_text TEXT

);

-- Table: day

DROP TABLE IF EXISTS day;

CREATE TABLE day (

Day\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Date DATE NOT NULL,

Income DOUBLE DEFAULT NULL

);

-- Table: customers

DROP TABLE IF EXISTS customers;

CREATE TABLE customers (

Customer\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Day DATE NOT NULL,

Name VARCHAR (50) NOT NULL,

Phone TEXT (20) NOT NULL,

Discount DOUBLE NOT NULL DEFAULT 0

);

-- Table: discount\_type

DROP TABLE IF EXISTS discount\_type;

CREATE TABLE discount\_type (

Discount\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Name VARCHAR (50) NOT NULL

);

-- Table: discounts

DROP TABLE IF EXISTS discounts;

CREATE TABLE discounts (

Discount\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Description VARCHAR (255) DEFAULT NULL,

Start\_date DATE DEFAULT NULL,

End\_date DATE DEFAULT NULL,

Percentage INT (11) DEFAULT NULL,

Value DOUBLE DEFAULT NULL

);

-- Table: products

DROP TABLE IF EXISTS products;

CREATE TABLE products (

Product\_id INTEGER NOT NULL PRIMARY KEY,

Name VARCHAR (50) DEFAULT NULL,

Price DOUBLE DEFAULT NULL,

Discount DOUBLE NOT NULL,

Stock INT (11) DEFAULT NULL

);

-- Table: promo

DROP TABLE IF EXISTS promo;

CREATE TABLE promo (

Promo\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

Promo TEXT,

Description TEXT,

GreaterThan INTEGER,

Percentage INTEGER,

Value INTEGER,

Start\_date TEXT,

End\_date TEXT

);

-- Table: taxreports

DROP TABLE IF EXISTS taxreports;

CREATE TABLE taxreports (

Report\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Start\_invoice INT (11) NOT NULL,

End\_invoice INT (11) NOT NULL,

Start\_date DATE NOT NULL,

End\_date DATE NOT NULL,

Discount INT (11) NOT NULL

);

-- Table: users

DROP TABLE IF EXISTS users;

CREATE TABLE users (

User\_id INT (11) NOT NULL primary key auto\_increment,

Name VARCHAR (50) NOT NULL,

Password VARCHAR (50) NOT NULL,

Group\_id INT (11) NOT NULL,

connected INTEGER DEFAULT 0

);

-- Table: promo\_customer

DROP TABLE IF EXISTS promo\_customer;

CREATE TABLE promo\_customer (

Promo\_customer\_id INTEGER PRIMARY KEY AUTO\_INCREMENT,

Promo\_id INTEGER,

Customer\_id INTEGER,

constraint FK\_pc\_promo foreign key (Promo\_id) references promo (Promo\_id),

constraint FK\_pc\_customer foreign key (Customer\_id) references customers (Customer\_id)

);

-- Table: receive

DROP TABLE IF EXISTS receive;

CREATE TABLE receive (

Receive\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Product\_id INT (11) DEFAULT NULL,

Day DATE DEFAULT NULL,

Amount INT (11) DEFAULT NULL,

constraint FK\_receive\_product foreign key (Product\_id) references products(Product\_id)

);

-- Table: invoice

DROP TABLE IF EXISTS invoice;

CREATE TABLE invoice (

Invoice\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Day DATE DEFAULT NULL,

Subtotal DOUBLE DEFAULT NULL,

Discount DOUBLE DEFAULT 0,

Total DOUBLE NOT NULL DEFAULT 0,

Discount\_type INTEGER,

Customer\_id INT (11) DEFAULT NULL,

Seller\_id INT (11) DEFAULT NULL,

Promo\_id INTEGER,

constraint FK\_invoice\_customer foreign key (Customer\_id) references customers(Customer\_id),

constraint FK\_invoice\_seller foreign key (Seller\_id) references users(User\_id),

constraint FK\_invoice\_promo foreign key (Promo\_id) references promo(Promo\_id)

);

-- Table: sales

DROP TABLE IF EXISTS sales;

CREATE TABLE sales (

Sale\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Invoice\_id INT (11) DEFAULT NULL,

Product\_id INT (11) DEFAULT NULL,

Name VARCHAR (50) NOT NULL,

Price DOUBLE NOT NULL,

Discount DOUBLE NOT NULL,

Totalprice DOUBLE DEFAULT NULL,

Number INT (11) DEFAULT NULL,

constraint FK\_sales\_invoice foreign key (Invoice\_id) references invoice(Invoice\_id),

constraint FK\_sales\_products foreign key (Product\_id) references products(Product\_id)

);

-- Table: back

DROP TABLE IF EXISTS back;

CREATE TABLE back (

Back\_id INTEGER NOT NULL PRIMARY KEY AUTO\_INCREMENT,

Invoice\_id INT (11) DEFAULT NULL,

Product\_id INT (11) DEFAULT NULL,

Name VARCHAR (50) DEFAULT NULL,

Price DOUBLE DEFAULT NULL,

Discount DOUBLE DEFAULT NULL,

Totalprice DOUBLE DEFAULT NULL,

Selling\_date DATE DEFAULT NULL,

Retrieving\_date DATE DEFAULT NULL,

constraint FK\_back\_invoice foreign key (Invoice\_id) references invoice(invoice\_id),

constraint FK\_back\_product foreign key (Product\_id) references products(Product\_id)

);