CREATE TABLE currencies (  
 currency\_id SERIAL PRIMARY KEY,  
 name char UNIQUE NOT NULL  
);  
  
CREATE TABLE publishers (  
 publisher\_id serial PRIMARY KEY,  
 name varchar(100) UNIQUE NOT NULL  
);  
  
CREATE TABLE statuses (  
 status\_id serial primary key,  
 name varchar(50) UNIQUE NOT NULL  
);  
  
-- drop table books cascade;  
CREATE TABLE books (  
 book\_id serial primary key,  
 author varchar(100) NOT NULL,  
 title varchar(100) NOT NULL,  
 description varchar(500),  
 price numeric NOT NULL,  
 currency\_id int NOT NULL default 1,  
 remains int NOT NULL CHECK ( remains >= 0 ),  
 publisher\_id int,  
  
 FOREIGN KEY (currency\_id) REFERENCES currencies (currency\_id) on delete set default,  
 FOREIGN KEY (publisher\_id) REFERENCES publishers (publisher\_id) on delete set null  
);  
  
CREATE TABLE consumers (  
 consumer\_id serial PRIMARY KEY,  
 name varchar(50) NOT NULL,  
 surname varchar(100) NOT NULL,  
 email varchar(200) UNIQUE NOT NULL,  
 password varchar(200) NOT NULL,  
 telephone varchar(8) UNIQUE NOT NULL,  
 address varchar(500)  
);  
  
-- drop table orders cascade ;  
CREATE TABLE orders (  
 order\_id serial primary key,  
 total\_price int NOT NULL CHECK ( total\_price >= 0 ),  
 creation\_date DATE NOT NULL DEFAULT *CURRENT\_DATE*,  
 end\_date DATE NOT NULL DEFAULT *CURRENT\_DATE*,  
 status\_id int,  
 consumer\_id int,  
  
 FOREIGN KEY (status\_id) references statuses (status\_id) on delete set null,  
 FOREIGN KEY (consumer\_id) references consumers (consumer\_id) on delete cascade  
);  
  
-- drop table ordersBooks;  
CREATE TABLE ordersBooks (  
 order\_book\_id serial primary key,  
 order\_id int,  
 book\_id int,  
 quantity int NOT NULL CHECK ( quantity > 0 ),  
  
 foreign key (order\_id) references orders (order\_id) on delete cascade,  
 foreign key (book\_id) references books (book\_id) on delete cascade  
);  
  
INSERT INTO currencies  
VALUES (1, 'P'),  
 (2, '$');  
  
INSERT INTO publishers (name)  
VALUES ('Махаон'), ('Росмэн'), ('АСТ');  
  
INSERT INTO books (author, title, description, price, currency\_id, remains, publisher\_id)  
VALUES ('Джоан Роулинг', 'Гарри Поттер и Филосовский камень',  
 'Книга, покорившая мир, эталон литературы для читателей всех возрастов, синоним успеха. ' ||  
 'Книга, сделавшая Джоан Роулинг самым читаемым писателем современности. Книга, ставшая культовой ' ||  
 'уже для нескольких поколений. «Гарри Поттер и Философский камень» - история начинается.',  
 700, 1, 20, 2),  
 ('Льюис Кэрролл', 'Алиса в стране чудес', 'Сказочная история об одном волшебном сне, в котором девчка Алиса,' ||  
 ' провалившись в кроличью нору, попадает в страну чудес, в мир невероятных приключений и удивительных встреч с ' ||  
 'самыми необычными существами. Классическая повесть-сказка в этом издании адаптирована с учётом возрастных ' ||  
 'особенностей детей, уже освоивших самостоятельное чтение и обладающих достаточным словарным запасом.',  
 353, 1, 40, 1);  
  
-- select \* from books;  
  
-- DROP TABLE baskets;  
CREATE TABLE baskets (  
 basket\_id serial primary key,  
 consumer\_id int not null,  
  
 foreign key (consumer\_id) references consumers (consumer\_id)  
);  
  
-- DROP TABLE basketBooks;  
CREATE TABLE basketBooks (  
 basketBooks\_id serial primary key,  
 basket\_id int NOT NULL,  
 book\_id int NOT NULL,  
 quantity int NOT NULL CHECK ( quantity > 0 ),  
  
 foreign key (basket\_id) references baskets (basket\_id) on DELETE cascade,  
 foreign key (book\_id) references books (book\_id) on delete cascade  
)

