Labwork 5 db

--1.1  
create table employees (  
 employee\_id INTEGER,  
 first\_name TEXT,  
 last\_name TEXT,  
 age INTEGER CHECK (age BETWEEN 18 and 65),  
 salary NUMERIC CHECK (salary > 0)  
);  
--1.2  
create table products\_catalog(  
 product\_id INTEGER,  
 product\_name TEXT,  
 regular\_price NUMERIC,  
 discount\_price NUMERIC,  
 CONSTRAINT valid\_discount check (  
 regular\_price > 0 AND  
 discount\_price > 0 AND  
 discount\_price < regular\_price  
 )  
);  
  
--1.3  
create table bookings (  
 booking\_id integer,  
 check\_in\_date DATE,  
 check\_out\_date date,  
 num\_guests INTEGER CHECK(num\_guests BETWEEN 1 and 10),  
 CHECK (check\_out\_date > check\_in\_date)  
);  
  
--1.4  
insert into bookings values (1,'2025-05-01','2025-05-05',3)  
-- который выше подходит под чек а вот ниже нет пот дата нарушает чек  
insert into bookings values(1,'2025-06-09','2021-03-19',8)  
-- выдает ошибку  
  
--PART 2  
  
create table customers (  
 customer\_id INTEGER not null,  
 email TEXT not null,  
 phone text,  
 registration\_date date not null  
  
);  
  
--task 2.2  
create table inventory(  
 item\_id INTEGER not null,  
 item\_name text not null,  
 quantity integer not null check (quantity >= 0),  
 unit\_price numeric not null check (unit\_price > 0),  
 last\_updated TIMESTAMP not null  
  
);  
  
--part 3 3.1  
create table users (  
 user\_id integer,  
 username text unique ,  
 email text unique ,  
 created\_at timestamp  
  
);  
  
--part 3.2  
create table course\_enrollments(  
 enrollment\_id integer,  
 student\_id integer,  
 course\_code text,  
 semester text,  
 constraint unique\_enrollment unique (student\_id,course\_code,semester)  
  
);  
  
--part 4  
CREATE TABLE departments (  
 dept\_id INTEGER PRIMARY KEY,  
 dept\_name TEXT NOT NULL,  
 location TEXT  
);  
  
--4.2  
CREATE TABLE student\_courses (  
 student\_id INTEGER,  
 course\_id INTEGER,  
 enrollment\_date DATE,  
 grade TEXT,  
 PRIMARY KEY (student\_id, course\_id)  
);  
  
--part 5 5.1  
CREATE TABLE employees\_dept (  
 emp\_id INTEGER PRIMARY KEY,  
 emp\_name TEXT NOT NULL,  
 dept\_id INTEGER REFERENCES departments(dept\_id),  
 hire\_date DATE  
);  
  
-- 5.2  
CREATE TABLE authors (  
 author\_id INTEGER PRIMARY KEY,  
 author\_name TEXT NOT NULL,  
 country TEXT  
);  
  
CREATE TABLE publishers (  
 publisher\_id INTEGER PRIMARY KEY,  
 publisher\_name TEXT NOT NULL,  
 city TEXT  
);  
  
CREATE TABLE books (  
 book\_id INTEGER PRIMARY KEY,  
 title TEXT NOT NULL,  
 author\_id INTEGER REFERENCES authors(author\_id),  
 publisher\_id INTEGER REFERENCES publishers(publisher\_id),  
 publication\_year INTEGER,  
 isbn TEXT UNIQUE  
);  
  
--5.3  
CREATE TABLE categories (  
 category\_id INTEGER PRIMARY KEY,  
 category\_name TEXT NOT NULL  
);  
  
CREATE TABLE products\_fk (  
 product\_id INTEGER PRIMARY KEY,  
 product\_name TEXT NOT NULL,  
 category\_id INTEGER REFERENCES categories(category\_id) ON DELETE RESTRICT  
);  
  
CREATE TABLE orders (  
 order\_id INTEGER PRIMARY KEY,  
 order\_date DATE NOT NULL  
);  
  
CREATE TABLE order\_items (  
 item\_id INTEGER PRIMARY KEY,  
 order\_id INTEGER REFERENCES orders(order\_id) ON DELETE CASCADE,  
 product\_id INTEGER REFERENCES products\_fk(product\_id),  
 quantity INTEGER CHECK (quantity > 0)  
);  
  
--6  
CREATE TABLE customers (  
 customer\_id SERIAL PRIMARY KEY,  
 name TEXT NOT NULL,  
 email TEXT UNIQUE NOT NULL,  
 phone TEXT,  
 registration\_date DATE NOT NULL  
);  
  
CREATE TABLE products (  
 product\_id SERIAL PRIMARY KEY,  
 name TEXT NOT NULL,  
 description TEXT,  
 price NUMERIC CHECK (price >= 0),  
 stock\_quantity INTEGER CHECK (stock\_quantity >= 0)  
);  
  
CREATE TABLE orders (  
 order\_id SERIAL PRIMARY KEY,  
 customer\_id INTEGER REFERENCES customers(customer\_id) ON DELETE CASCADE,  
 order\_date DATE NOT NULL,  
 total\_amount NUMERIC CHECK (total\_amount >= 0),  
 status TEXT CHECK (status IN ('pending','processing','shipped','delivered','cancelled'))  
);  
  
CREATE TABLE order\_details (  
 order\_detail\_id SERIAL PRIMARY KEY,  
 order\_id INTEGER REFERENCES orders(order\_id) ON DELETE CASCADE,  
 product\_id INTEGER REFERENCES products(product\_id),  
 quantity INTEGER CHECK (quantity > 0),  
 unit\_price NUMERIC CHECK (unit\_price >= 0)  
);