***URMANOVA SAMIRA***

***ID:24B032087***

-- Часть 1: CHECK  
  
-- Задание 1.1  
CREATE TABLE employees  
(  
 employee\_id integer,  
 first\_name text,  
 last\_name text,  
 age integer CHECK (age >= 18 AND age <= 65), -- возраст от 18 до 65  
 salary numeric CHECK (salary > 0) -- зарплата положительная  
);  
  
-- Нормальные данные  
INSERT INTO employees VALUES (1, 'Kim', 'Namjoon', 25, 50000);  
INSERT INTO employees VALUES (2, 'Kim', 'Seokjin', 30, 60000);  
  
-- Эти не пройдут (ошибки)  
-- INSERT INTO employees VALUES (3, 'Test', 'Age', 17, 50000); -- маловато лет  
-- INSERT INTO employees VALUES (4, 'Test', 'Salary', 25, -100); -- зарплата отрицательная  
  
-- Задание 1.2  
CREATE TABLE products\_catalog (  
 product\_id integer,  
 product\_name text,  
 regular\_price numeric,  
 discount\_price numeric,  
 CONSTRAINT valid\_discount CHECK ( -- названное ограничение  
 regular\_price > 0  
 AND discount\_price > 0  
 AND discount\_price < regular\_price -- скидка меньше обычной цены  
 )  
);  
  
INSERT INTO products\_catalog VALUES (1, 'BTS Album', 1000, 800);  
INSERT INTO products\_catalog VALUES (2, 'BT21 Plush', 500, 450);  
  
-- Ошибочные данные  
-- INSERT INTO products\_catalog VALUES (3, 'Error', 0, 800); -- цена ноль  
-- INSERT INTO products\_catalog VALUES (4, 'Error', 1000, 1200); -- скидка больше цены  
  
-- Задание 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), -- гостей от 1 до 10  
 CHECK (check\_out\_date > check\_in\_date) -- выезд после заезда  
);  
  
INSERT INTO bookings VALUES (1, '2023-10-01', '2023-10-05', 2);  
INSERT INTO bookings VALUES (2, '2023-11-01', '2023-11-03', 4);  
  
-- Не пройдет  
-- INSERT INTO bookings VALUES (3, '2023-10-01', '2023-09-28', 2); -- даты наоборот  
-- INSERT INTO bookings VALUES (4, '2023-10-01', '2023-10-05', 0); -- ноль гостей  
  
-- Часть 2: NOT NULL  
  
-- Задание 2.1  
CREATE TABLE customers  
(  
 customer\_id integer NOT NULL, -- обязательно к заполнению  
 email text NOT NULL, -- обязательно  
 phone text, -- можно пропустить  
 registration\_date date NOT NULL -- обязательно  
);  
  
INSERT INTO customers VALUES (1, 'rm@bts.com', NULL, '2023-01-15');  
INSERT INTO customers VALUES (2, 'jin@bts.com', '+123456789', '2023-02-20');  
  
-- Ошибки NOT NULL  
-- INSERT INTO customers VALUES (NULL, 'test@bts.com', NULL, '2023-03-01'); -- нет айди  
-- INSERT INTO customers VALUES (3, NULL, NULL, '2023-03-01'); -- нет почты  
  
-- Задание 2.2  
CREATE TABLE inventory  
(  
 item\_id integer NOT NULL,  
 item\_name text NOT NULL,  
 quantity integer NOT NULL CHECK (quantity >= 0), -- не отрицательный остаток  
 unit\_price numeric NOT NULL CHECK (unit\_price > 0), -- цена положительная  
 last\_updated timestamp NOT NULL  
);  
  
INSERT INTO inventory VALUES (1, 'Light Stick', 50, 1500, '2023-10-01 12:00:00');  
INSERT INTO inventory VALUES (2, 'Photocard Set', 30, 3500, '2023-10-01 12:05:00');  
  
-- Часть 3: UNIQUE  
  
-- Задание 3.1  
CREATE TABLE users  
(  
 user\_id integer,  
 username text UNIQUE, -- уникальный логин  
 email text UNIQUE, -- уникальная почта  
 created\_at timestamp  
);  
  
INSERT INTO users VALUES (1, 'rm\_leader', 'namjoon@bts.com', *NOW*());  
INSERT INTO users VALUES (2, 'jin\_worldwide', 'seokjin@bts.com', *NOW*());  
  
-- Дубликаты не пройдут  
-- INSERT INTO users VALUES (3, 'rm\_leader', 'other@bts.com', NOW()); -- такой username уже есть  
-- INSERT INTO users VALUES (4, 'new\_user', 'namjoon@bts.com', NOW()); -- такая почта уже есть  
  
-- Задание 3.2  
CREATE TABLE course\_enrollments  
(  
 enrollment\_id integer,  
 student\_id integer,  
 course\_code text,  
 semester text,  
 UNIQUE (student\_id, course\_code, semester) -- уникальная комбинация  
);  
  
INSERT INTO course\_enrollments VALUES (1, 101, 'MUSIC101', 'Fall2023');  
INSERT INTO course\_enrollments VALUES (2, 102, 'DANCE101', 'Fall2023');  
  
-- Нельзя записаться дважды на один курс  
-- INSERT INTO course\_enrollments VALUES (3, 101, 'MUSIC101', 'Fall2023');  
  
-- Часть 4: PRIMARY KEY  
  
-- Задание 4.1  
CREATE TABLE departments  
(  
 dept\_id integer PRIMARY KEY, -- главный ключ  
 dept\_name text NOT NULL,  
 location text  
);  
  
INSERT INTO departments VALUES (1, 'Music', 'Seoul');  
INSERT INTO departments VALUES (2, 'Dance', 'Busan');  
INSERT INTO departments VALUES (3, 'Production', 'Daegu');  
  
-- Ошибки PK  
-- INSERT INTO departments VALUES (1, 'Duplicate', 'City'); -- повтор айди  
-- INSERT INTO departments VALUES (NULL, 'No ID', 'City'); -- айди пустой  
  
-- Задание 4.2  
CREATE TABLE student\_courses  
(  
 student\_id integer,  
 course\_id integer,  
 enrollment\_date date,  
 grade text,  
 PRIMARY KEY (student\_id, course\_id) -- составной ключ  
);  
  
INSERT INTO student\_courses VALUES (101, 1, '2023-09-01', 'A');  
INSERT INTO student\_courses VALUES (102, 1, '2023-09-01', 'B');  
  
-- Часть 5: FOREIGN KEY  
  
-- Задание 5.1  
CREATE TABLE employees\_dept  
(  
 emp\_id integer PRIMARY KEY,  
 emp\_name text NOT NULL,  
 dept\_id integer REFERENCES departments (dept\_id), -- ссылка на отделы  
 hire\_date date  
);  
  
INSERT INTO employees\_dept VALUES (1, 'Min Yoongi', 1, '2020-01-15');  
INSERT INTO employees\_dept VALUES (2, 'Jung Hoseok', 2, '2019-03-20');  
  
-- Несуществующий отдел  
-- INSERT INTO employees\_dept VALUES (3, 'Error', 999, '2023-01-01');  
  
-- Задание 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), -- автор существует  
 publisher\_id integer REFERENCES publishers (publisher\_id), -- издатель существует  
 publication\_year integer,  
 isbn text UNIQUE -- уникальный ISBN  
);  
  
INSERT INTO authors VALUES (1, 'Kim Namjoon', 'South Korea');  
INSERT INTO authors VALUES (2, 'Min Yoongi', 'South Korea');  
  
INSERT INTO publishers VALUES (1, 'Big Hit', 'Seoul');  
INSERT INTO publishers VALUES (2, 'HYBE', 'Seoul');  
  
INSERT INTO books VALUES (1, 'BTS Biography', 1, 1, 2020, '978-5-699-12014-7');  
INSERT INTO books VALUES (2, 'Music Production Guide', 2, 2, 2021, '978-5-17-090987-3');  
  
-- Задание 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 ON DELETE RESTRICT -- нельзя удалить если есть продукты  
);  
  
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 ON DELETE CASCADE, -- удаляем заказ - удаляем items  
 product\_id integer REFERENCES products\_fk,  
 quantity integer CHECK (quantity > 0)  
);  
  
INSERT INTO categories VALUES (1, 'Albums');  
INSERT INTO products\_fk VALUES (1, 'Proof Album', 1);  
INSERT INTO orders VALUES (1, '2023-10-01');  
INSERT INTO order\_items VALUES (1, 1, 1, 2);  
  
-- Тесты ON DELETE  
-- DELETE FROM categories WHERE category\_id = 1; -- ошибка (RESTRICT)  
DELETE FROM orders WHERE order\_id = 1; -- order\_items удалились сами (CASCADE)  
  
-- Часть 6: Интернет-магазин  
  
CREATE TABLE ecom\_customers  
(  
 customer\_id integer PRIMARY KEY,  
 name text NOT NULL,  
 email text UNIQUE NOT NULL, -- уникальная почта  
 phone text,  
 registration\_date date NOT NULL  
);  
  
CREATE TABLE ecom\_products  
(  
 product\_id integer PRIMARY KEY,  
 name text NOT NULL,  
 description text,  
 price numeric CHECK (price >= 0), -- цена не отрицательная  
 stock\_quantity integer CHECK (stock\_quantity >= 0) -- остаток не отрицательный  
);  
  
CREATE TABLE ecom\_orders  
(  
 order\_id integer PRIMARY KEY,  
 customer\_id integer REFERENCES ecom\_customers ON DELETE CASCADE, -- удаляем пользователя - удаляем заказы  
 order\_date date NOT NULL,  
 total\_amount numeric CHECK (total\_amount >= 0),  
 status text CHECK (status IN ('pending', 'processing', 'shipped', 'delivered', 'cancelled')) -- статус из списка  
);  
  
CREATE TABLE ecom\_order\_details  
(  
 order\_detail\_id integer PRIMARY KEY,  
 order\_id integer REFERENCES ecom\_orders ON DELETE CASCADE, -- удаляем заказ - удаляем детали  
 product\_id integer REFERENCES ecom\_products ON DELETE RESTRICT, -- нельзя удалить товар если он в заказе  
 quantity integer CHECK (quantity > 0), -- положительное количество  
 unit\_price numeric CHECK (unit\_price >= 0) -- не отрицательная цена  
);  
  
-- Тестовые данные  
INSERT INTO ecom\_customers VALUES  
(1, 'Kim Namjoon', 'rm@bts.com', '+79111111111', '2023-01-10'),  
(2, 'Kim Seokjin', 'jin@bts.com', '+79222222222', '2023-01-12'),  
(3, 'Min Yoongi', 'suga@bts.com', '+79333333333', '2023-01-15'),  
(4, 'Jung Hoseok', 'jhope@bts.com', '+79444444444', '2023-01-20'),  
(5, 'Park Jimin', 'jimin@bts.com', '+79555555555', '2023-01-25');  
  
INSERT INTO ecom\_products VALUES  
(1, 'Proof Album', 'BTS Official Album', 30000, 50),  
(2, 'Light Stick', 'Official ARMY Bomb', 80000, 30),  
(3, 'BT21 Plush', 'RJ Plush Toy', 5000, 100),  
(4, 'Photocard Set', 'BTS Member Photocards', 15000, 40),  
(5, 'Concert DVD', 'Yet To Come in BUSAN', 40000, 20);  
  
INSERT INTO ecom\_orders VALUES  
(1, 1, '2023-10-01', 30000, 'delivered'),  
(2, 2, '2023-10-02', 85000, 'processing'),  
(3, 3, '2023-10-03', 5000, 'shipped'),  
(4, 4, '2023-10-04', 55000, 'pending'),  
(5, 5, '2023-10-05', 15000, 'delivered');  
  
INSERT INTO ecom\_order\_details VALUES  
(1, 1, 1, 1, 30000),  
(2, 2, 2, 1, 80000),  
(3, 2, 3, 1, 5000),  
(4, 3, 3, 1, 5000),  
(5, 4, 4, 1, 15000),  
(6, 4, 5, 1, 40000),  
(7, 5, 4, 1, 15000);  
  
-- Проверка ограничений магазина  
INSERT INTO ecom\_orders VALUES (6, 1, '2023-10-06', 20000, 'pending');  
  
-- Эти не должны работать:  
-- INSERT INTO ecom\_customers VALUES (6, 'Test', 'rm@bts.com', NULL, NOW()); -- почта уже есть  
-- INSERT INTO ecom\_products VALUES (6, 'Test', 'Test', -100, 10); -- отрицательная цена  
-- INSERT INTO ecom\_orders VALUES (7, 999, NOW(), 100, 'pending'); -- нет такого пользователя  
-- INSERT INTO ecom\_order\_details VALUES (8, 1, 1, -1, 100); -- отрицательное количество  
  
-- Проверка каскадного удаления  
DELETE FROM ecom\_orders WHERE order\_id = 1; -- удалится заказ и его детали