Task 1

use testdb;

DROP TABLE IF EXISTS users;

CREATE TABLE IF NOT EXISTS users(

id INT AUTO\_INCREMENT,

name VARCHAR(255),

email VARCHAR(255),

PRIMARY KEY(id)

);

DROP TABLE IF EXISTS profiless;

CREATE TABLE IF NOT EXISTS profiless(

id INT AUTO\_INCREMENT,

user\_id INT,

birthday VARCHAR(255),

phone VARCHAR(255),

PRIMARY KEY(id),

FOREIGN KEY(user\_id) REFERENCES users(id) ON DELETE CASCADE

);

INSERT INTO users(name, email) VALUES

('username 1', 'email@email.com'),

('username 2', 'email@email.com'),

('username 3', 'email@email.com'),

('username 4', 'email@email.com');

INSERT INTO profiless(user\_id, birthday, phone) VALUES

(1, '21.21.21', '+380380380'),

(2, '22.22.22', '+380380380'),

(3, '23.23.23', '+380380380'),

(4, '24.24.24', '+380380380');

SELECT u.id, u.name, u.email, p.birthday, p.phone

FROM users as u

JOIN profiless as p

ON p.user\_id = u.id

;

UPDATE profiless

SET phone = '+0000000000'

WHERE id = 4;

DELETE FROM users

WHERE id = 1;

Task 2

DROP TABLE IF EXISTS articles;

CREATE TABLE IF NOT EXISTS articles(

id INT AUTO\_INCREMENT,

title VARCHAR(255),

content TEXT,

PRIMARY KEY(id)

);

DROP TABLE IF EXISTS comments;

CREATE TABLE IF NOT EXISTS comments(

id INT AUTO\_INCREMENT,

article\_id INT,

content TEXT,

author VARCHAR(255),

PRIMARY KEY(id),

FOREIGN KEY(article\_id) REFERENCES articles(id) ON DELETE CASCADE

);

INSERT INTO articles(title, content) VALUES

('title 1', 'content 1'),

('title 2', 'content 2'),

('title 3', 'content 3'),

('title 4', 'content 4');

INSERT INTO comments(article\_id, content, author) VALUES

(1, 'content 1', 'author 1'),

(1, 'content 2', 'author 2'),

(1, 'content 3', 'author 3'),

(2, 'content 1', 'author 1'),

(2, 'content 2', 'author 2'),

(3, 'content 1', 'author 3'),

(3, 'content 2', 'author 3'),

(4, 'content 4', 'author 4'),

(4, 'content 4', 'author 4')

;

SELECT a.id, a.title, a.content, count(c.id) as comments

FROM articles AS a

JOIN comments c

ON a.id = c.article\_id

GROUP BY a.id;

SELECT a.title as article, a.content as article\_content, c.content as comment, c.author as comment\_author

FROM articles AS a

JOIN comments c on a.id = c.article\_id

WHERE a.id = 4

;

DELETE FROM comments

WHERE article\_id = 4

;

Task 3

DROP TABLE IF EXISTS users\_movies;

DROP TABLE IF EXISTS users;

DROP TABLE IF EXISTS movies;

CREATE TABLE IF NOT EXISTS users(

id INT AUTO\_INCREMENT,

name VARCHAR(255),

PRIMARY KEY(id)

);

CREATE TABLE IF NOT EXISTS movies(

id INT AUTO\_INCREMENT,

name VARCHAR(255),

PRIMARY KEY(id)

);

CREATE TABLE IF NOT EXISTS users\_movies(

id INT AUTO\_INCREMENT,

user\_id INT,

movie\_id INT,

PRIMARY KEY(id),

FOREIGN KEY(user\_id) REFERENCES users(id) ON DELETE CASCADE,

FOREIGN KEY(movie\_id) REFERENCES movies(id) ON DELETE CASCADE

);

INSERT INTO users(name) VALUES

('username 1'),

('username 2'),

('username 3'),

('username 4');

INSERT INTO movies(name) VALUES

('movie 1'),

('movie 2'),

('movie 3'),

('movie 4');

INSERT INTO users\_movies(user\_id, movie\_id) VALUES

(1, 1),

(1, 2),

(1, 3),

(1, 4),

(2, 1),

(2, 2),

(2, 3),

(3, 1),

(3, 2),

(3, 4),

(4, 1),

(4, 3),

(4, 4);

SELECT u.name as users\_who\_watched\_movie\_1

FROM users\_movies

JOIN movies m on m.id = users\_movies.movie\_id

JOIN users u on u.id = users\_movies.user\_id

WHERE m.name = 'movie 1';

SELECT m.name as movies\_wached\_by\_user\_with\_id\_2

FROM users\_movies

JOIN users u on u.id = users\_movies.user\_id

JOIN movies m on m.id = users\_movies.movie\_id

WHERE u.id = 2;

INSERT INTO users\_movies(user\_id, movie\_id) VALUE

(4, 2);

Task 4

DROP TABLE IF EXISTS orders;

CREATE TABLE IF NOT EXISTS orders(

id INT AUTO\_INCREMENT,

date VARCHAR(255),

PRIMARY KEY(id)

);

DROP TABLE IF EXISTS products;

CREATE TABLE IF NOT EXISTS products(

id INT AUTO\_INCREMENT,

name VARCHAR(255),

price INT,

PRIMARY KEY(id)

);

DROP TABLE IF EXISTS orders\_products;

CREATE TABLE IF NOT EXISTS orders\_products(

id INT AUTO\_INCREMENT,

order\_id INT,

product\_id INT,

PRIMARY KEY(id),

FOREIGN KEY(order\_id) REFERENCES orders(id) ON DELETE CASCADE,

FOREIGN KEY(product\_id) REFERENCES products(id) ON DELETE CASCADE

);

INSERT INTO orders(date) VALUES

('2022-01-01'),

('2022-01-02'),

('2022-01-03'),

('2022-01-04');

INSERT INTO products(name, price) VALUES

('product 1', 1),

('product 2', 2),

('product 3', 3),

('product 4', 4);

INSERT INTO orders\_products(order\_id, product\_id) VALUES

(1, 1),

(1, 2),

(1, 3),

(1, 4),

(2, 1),

(2, 2),

(2, 3),

(3, 1),

(3, 2),

(3, 4),

(4, 1),

(4, 3),

(4, 4);

SELECT p.name as product\_name, o.date as ordered\_date

FROM orders\_products

JOIN orders o on o.id = orders\_products.order\_id

JOIN products p on p.id = orders\_products.product\_id

;

SELECT p.name

FROM orders\_products

JOIN products p on p.id = orders\_products.product\_id

JOIN orders o on o.id = orders\_products.order\_id

WHERE o.id > 2

GROUP BY p.name;

DELETE FROM orders\_products

WHERE order\_id = 3;