TEST DISKOMINFO 2024

Nama: Pranto Suwarno

posisi: Backend Developer  
  
SOAL A. Database

1.query create users table:

CREATE DATABASE sekolahku;

USE sekolahku;

CREATE TABLE users (

id INT(11) NOT NULL AUTO\_INCREMENT,

username VARCHAR(50) NOT NULL,

email VARCHAR(50) NOT NULL,

password VARCHAR(50) NOT NULL,

created\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (id)

);

INSERT INTO users (username, email, password, created\_at, updated\_at) VALUES

('Andi', 'andi@andi.com', 'password1', '2024-08-08 04:12:05', '2024-08-08 04:12:05'),

('Budi', 'budi@budi.com', 'password2', '2024-08-08 04:12:05', '2024-08-08 04:12:05'),

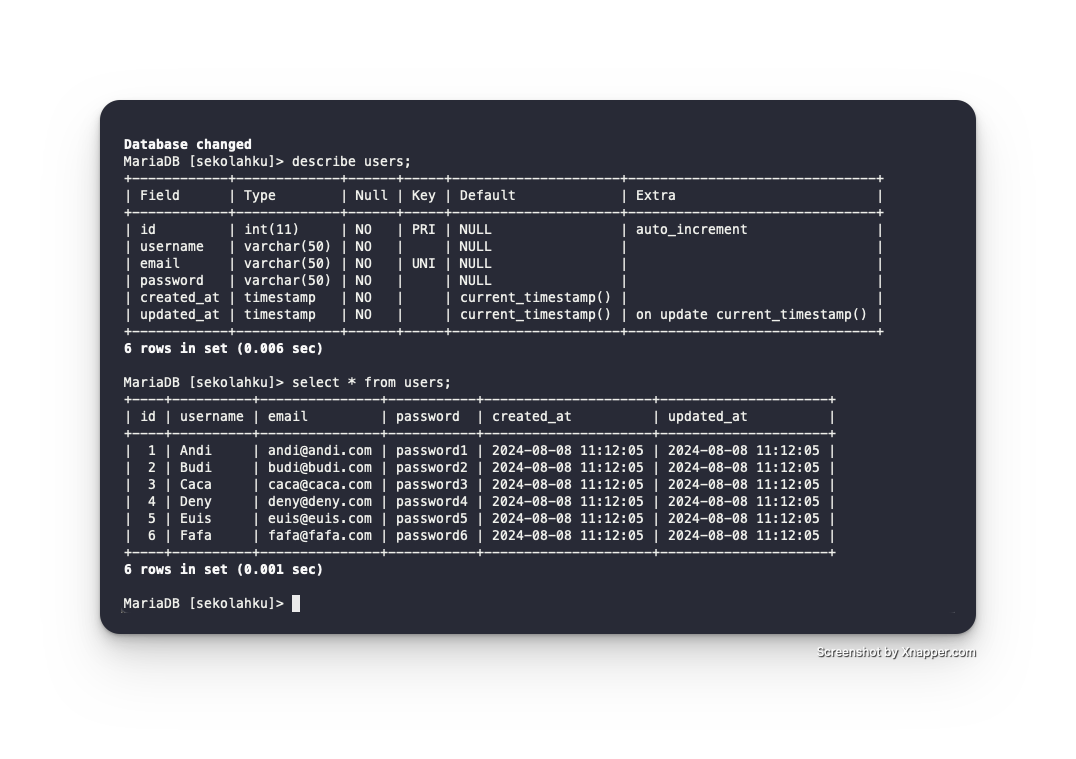
('Caca', 'caca@caca.com', 'password3', '2024-08-08 04:12:05', '2024-08-08 04:12:05'),

('Deny', 'deny@deny.com', 'password4', '2024-08-08 04:12:05', '2024-08-08 04:12:05'),

('Euis', 'euis@euis.com', 'password5', '2024-08-08 04:12:05', '2024-08-08 04:12:05'),

('Fafa', 'fafa@fafa.com', 'password6', '2024-08-08 04:12:05', '2024-08-08 04:12:05');

output:



2. query create courses table

CREATE TABLE courses (

id INT(11) NOT NULL AUTO\_INCREMENT,

course VARCHAR(50) NOT NULL,

mentor VARCHAR(50) NOT NULL,

title VARCHAR(50) NOT NULL,

PRIMARY KEY (id)

);

INSERT INTO courses (course, mentor, title) VALUES

('C++', 'ari', 'Dr.'),

('C#', 'ari', 'Dr.'),

('C#', 'ari', 'Dr.'),

('CSS', 'cania', 'S.Kom'),

('HTML', 'cania', 'S.Kom'),

('javascript', 'cania', 'S.kom'),

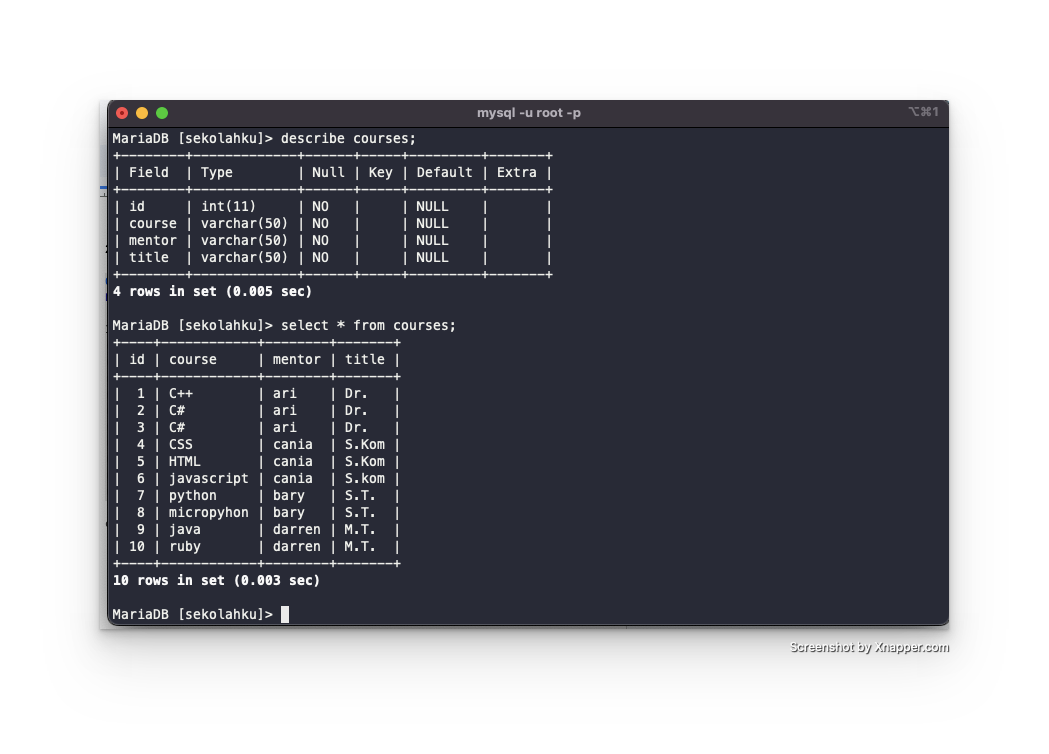
('python', 'bary', 'S.T.'),

('micropyhon', 'bary', 'S.T.'),

('java', 'darren', 'M.T.'),

('ruby', 'darren', 'M.T.');

output:



3. query create table useCourse relation with table users and courses

CREATE TABLE userCourse (

user\_id INT(11) NOT NULL,

course\_id INT(11) NOT NULL,

PRIMARY KEY (user\_id, course\_id),

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

FOREIGN KEY (course\_id) REFERENCES courses(id) ON DELETE CASCADE ON UPDATE CASCADE

);

INSERT INTO userCourse (user\_id, course\_id) VALUES

(1, 1),

(1, 2),

(1, 3),

(2, 4),

(2, 5),

(2, 6),

(3, 7),

(3, 8),

(3, 9),

(4, 1),

(4, 2),

(4, 3),

(5, 4),

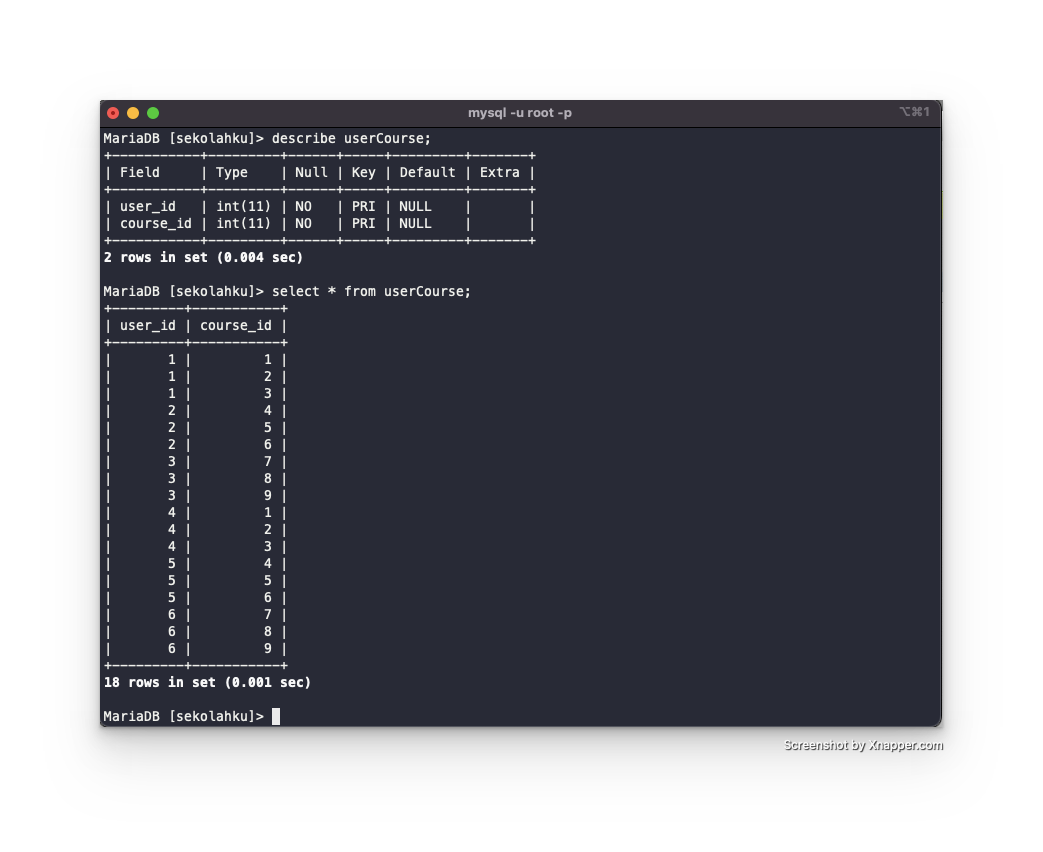
(5, 5),

(5, 6),

(6, 7),

(6, 8),

(6,9);

output:  


4. query

SELECT

u.username AS peserta\_didik,

c.course AS mata\_kuliah,

c.mentor AS nama\_mentor,

c.title AS gelar\_mentor

FROM

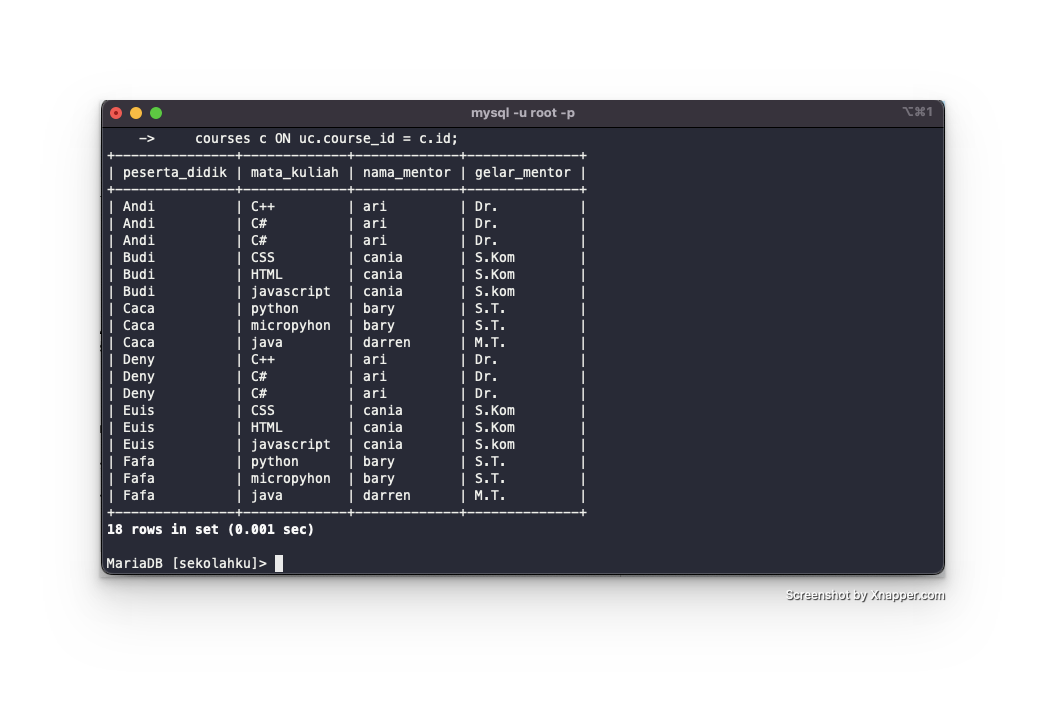
userCourse uc

JOIN

users u ON uc.user\_id = u.id

JOIN

courses c ON uc.course\_id = c.id;

output:  


5. query

SELECT

u.username AS peserta\_didik,

c.course AS mata\_kuliah,

c.mentor AS nama\_mentor,

c.title AS gelar\_mentor

FROM

userCourse uc

JOIN

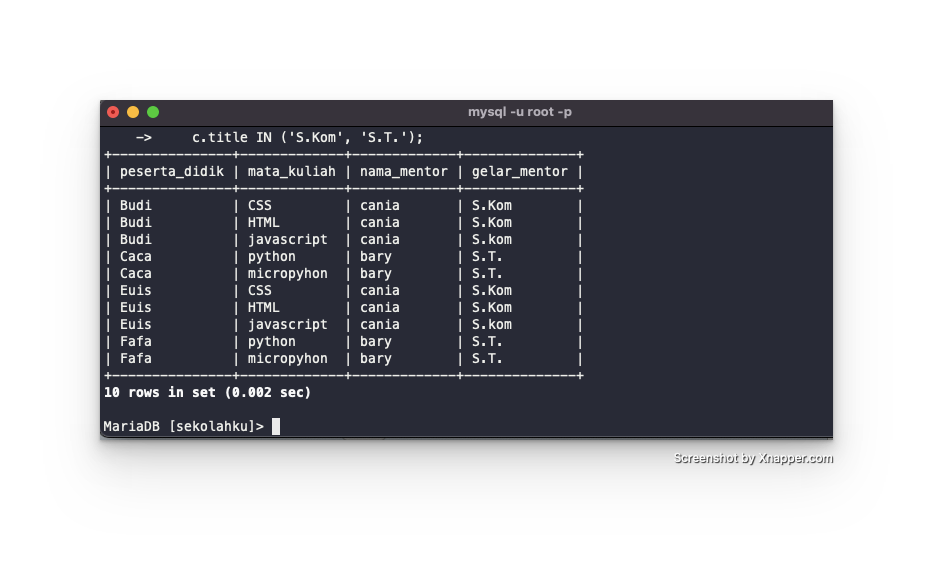
users u ON uc.user\_id = u.id

JOIN

courses c ON uc.course\_id = c.id

WHERE

c.title IN ('S.Kom', 'S.T.');

output:   


6. query

SELECT

u.username AS peserta\_didik,

c.course AS mata\_kuliah,

c.mentor AS nama\_mentor,

c.title AS gelar\_mentor

FROM

userCourse uc

JOIN

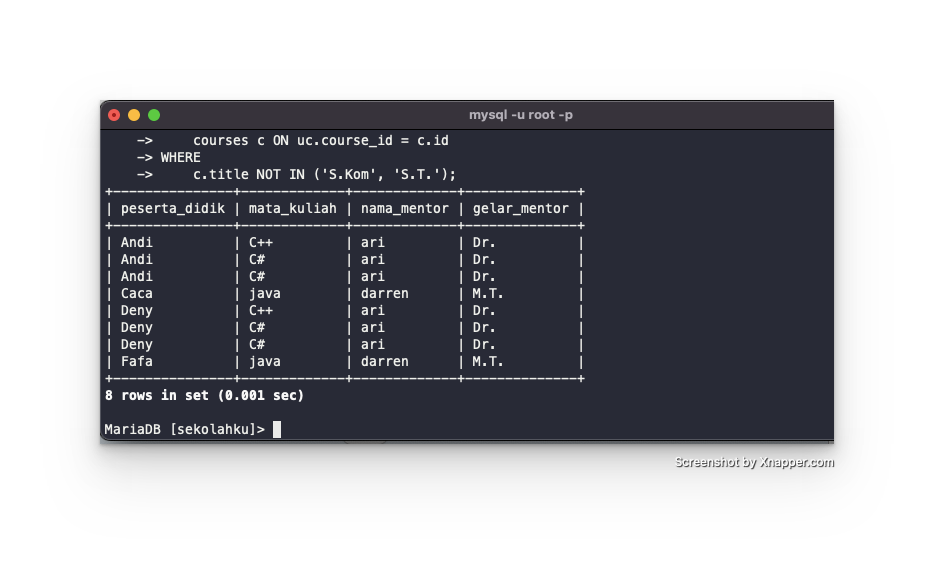
users u ON uc.user\_id = u.id

JOIN

courses c ON uc.course\_id = c.id

WHERE

c.title NOT IN ('S.Kom', 'S.T.');

output:  


7. query

SELECT

c.course AS mata\_kuliah,

c.mentor AS nama\_mentor,

c.title AS gelar\_mentor,

COUNT(uc.user\_id) AS jumlah\_peserta\_didik

FROM

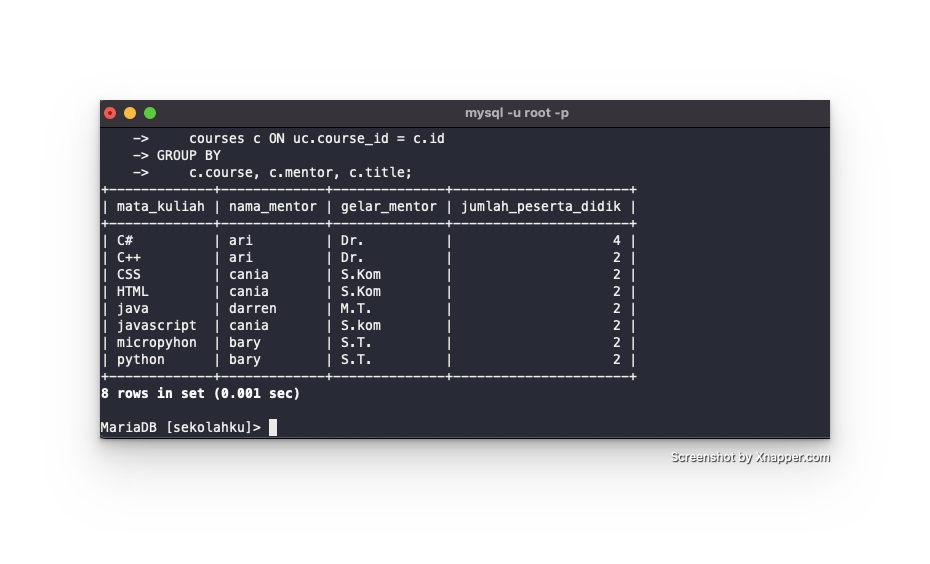
userCourse uc

JOIN

courses c ON uc.course\_id = c.id

GROUP BY

c.course, c.mentor, c.title;

output:  


8. query

SELECT

c.mentor AS mentor,

COUNT(uc.user\_id) AS jumlah\_peserta,

COUNT(uc.user\_id) \* 2000000 AS total\_fee

FROM

userCourse uc

JOIN

courses c ON uc.course\_id = c.id

GROUP BY

c.mentor;

output:  
