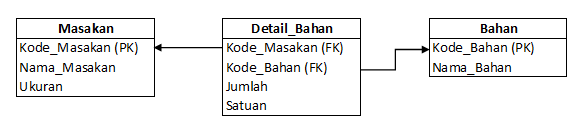
|  |  |
| --- | --- |
| No | : P7 |
| Tugas | : Kuis |
| Dosen pengampu | : Galih Hermawan, S.Kom., M.T. |
| NIM | : 10120095 |
| Nama | : Adrian Naufal Wijaya |
| Kelas | : Basdat2 UL |

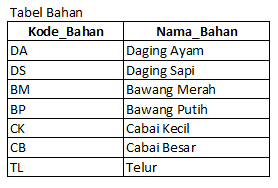
**KUIS SQL**

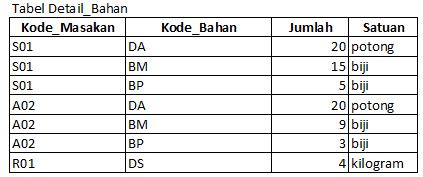
* Terdapat sebuah desain basis data dengan skema relasi sebagai berikut.



* Berikut ini contoh data pada masing-masing tabel.





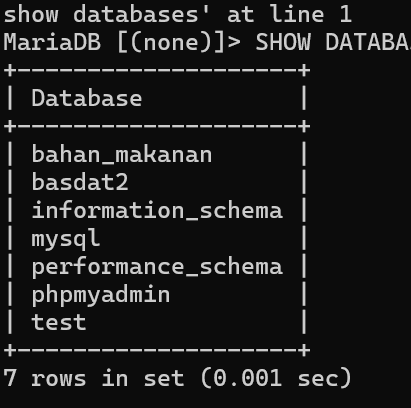


**Buat kueri SQL sesuai dengan persoalan berikut serta sertakan *screen capture* dari eksekusi kueri dan luarannya.**

1. Buat database Bahan\_Masakan dan pilih database tersebut.

Jawaban Kueri

CREATE DATABASE Bahan\_Masakan;

Screenshoot program

1. Buat tabel Masakan, Bahan, dan Detail\_Bahan. Sesuaikan kebutuhan *primary key* (PK), *foreign key* (PK) - gunakan ***cascade***, dan spesifikasi lainnya.

Jawaban Kueri

CREATE DATABASE IF NOT EXISTS bahan\_makanan;

USE bahan\_makanan;

CREATE TABLE Masakan (

Kode\_Masakan VARCHAR(10) PRIMARY KEY,

Nama\_Masakan VARCHAR(100) NOT NULL,

Ukuran VARCHAR(20)

) ENGINE=InnoDB;

CREATE TABLE Bahan (

Kode\_Bahan VARCHAR(10) PRIMARY KEY,

Nama\_Bahan VARCHAR(100) NOT NULL

) ENGINE=InnoDB;

CREATE TABLE Detail\_Bahan (

Kode\_Masakan VARCHAR(10),

Kode\_Bahan VARCHAR(10),

Jumlah INT NOT NULL,

Satuan VARCHAR(20) NOT NULL,

PRIMARY KEY (Kode\_Masakan, Kode\_Bahan),

FOREIGN KEY (Kode\_Masakan) REFERENCES Masakan(Kode\_Masakan)

ON DELETE CASCADE

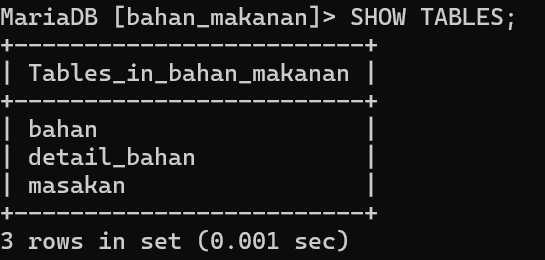
ON UPDATE CASCADE,

FOREIGN KEY (Kode\_Bahan) REFERENCES Bahan(Kode\_Bahan)

ON DELETE CASCADE

ON UPDATE CASCADE

) ENGINE=InnoDB;

Hasil Screenshoot

1. Isi data ke masing-masing tabel sesuai contoh pada tabel di soal.

Jawaban Kueri

INSERT INTO Masakan (Kode\_Masakan, Nama\_Masakan, Ukuran) VALUES

('S01', 'Soto Ayam', '15 liter'),

('S02', 'Soto Daging', '15 liter'),

('A01', 'Ayam Goreng', '30 Potong'),

('A02', 'Ayam Bakar', '20 Potong'),

('R01', 'Rawon Daging Sapi', '10 liter');

INSERT INTO Bahan (Kode\_Bahan, Nama\_Bahan) VALUES

('DA', 'Daging Ayam'),

('DS', 'Daging Sapi'),

('BM', 'Bawang Merah'),

('BP', 'Bawang Putih'),

('CK', 'Cabai Kecil'),

('CB', 'Cabai Besar'),

('TL', 'Telur');

INSERT INTO Detail\_Bahan (Kode\_Masakan, Kode\_Bahan, Jumlah, Satuan) VALUES

('S01', 'DA', 20, 'potong'),

('S01', 'BM', 15, 'biji'),

('S02', 'BP', 5, 'biji'),

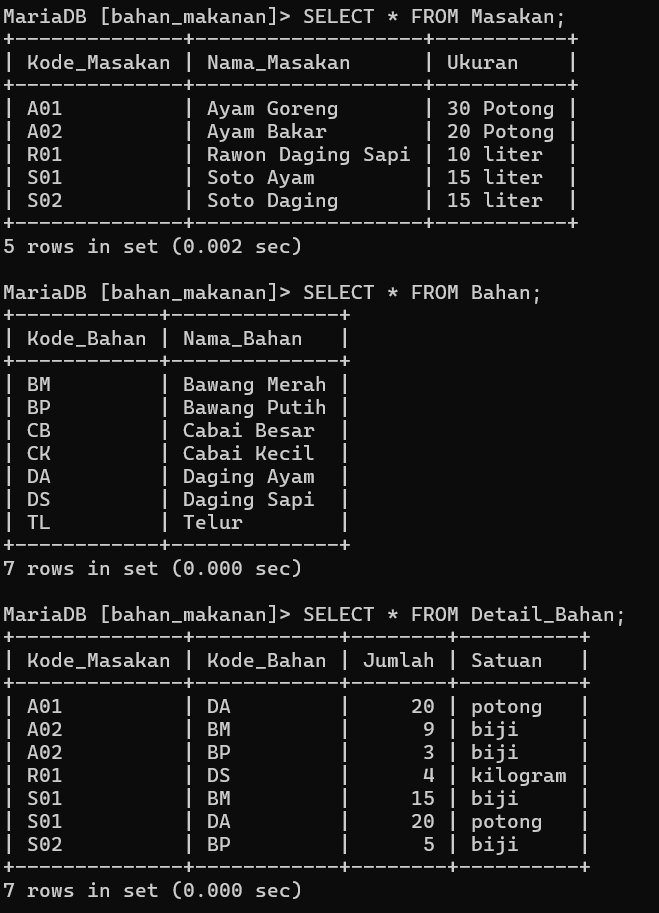
('A01', 'DA', 20, 'potong'),

('A02', 'BM', 9, 'biji'),

('A02', 'BP', 3, 'biji'),

('R01', 'DS', 4, 'kilogram');

Hasil Screenshoot



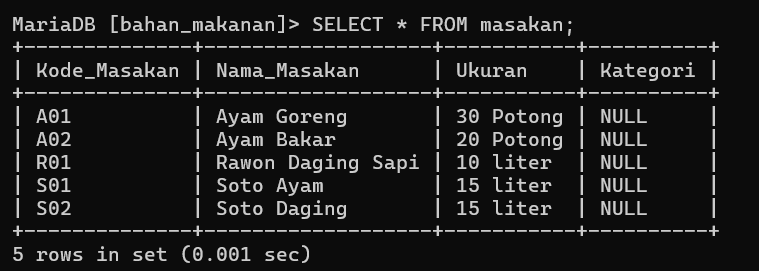
1. Tambahkan atribut baru pada tabel Masakan dengan nama Kategori, tipe varchar (30).

Jawaban Kueri

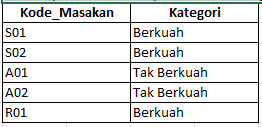
ALTER TABLE masakan

ADD Kategori VARCHAR (30);

Hasil Screenschoot



1. Isi atau perbaharui nilai pada atribut Kategori (yang masih kosong) dengan isi sebagai berikut. [ tips: gunakan perintah *update* untuk setiap masakan ]



Jawaban kueri

UPDATE masakan SET Kategori = 'Berkuah' WHERE Kode\_Masakan = 'S01';

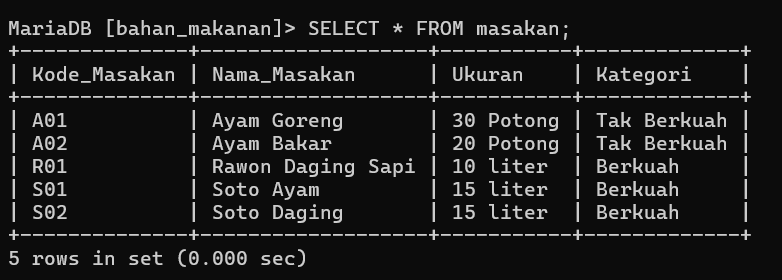
UPDATE masakan SET Kategori = 'Berkuah' WHERE Kode\_Masakan = 'S02';

UPDATE masakan SET Kategori = 'Tak Berkuah' WHERE Kode\_Masakan = 'A01';

UPDATE masakan SET Kategori = 'Tak Berkuah' WHERE Kode\_Masakan = 'A02';

UPDATE masakan SET Kategori = 'Berkuah' WHERE Kode\_Masakan = 'R01';

Hasil Screnshoot



1. Tampilkan data Masakan yang mengandung nama Daging, di mana ukurannya 10 liter.

[ tips: pencarian string 🡪 nama\_atribut LIKE “% kata %” ]

Luaran yang diharapkan:

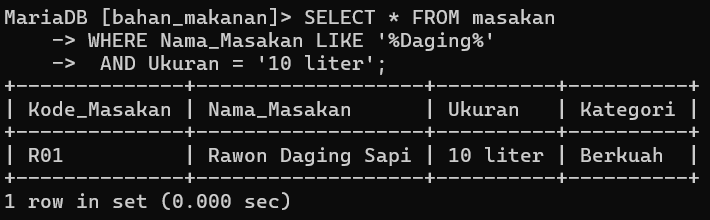
|  |  |  |
| --- | --- | --- |
| **Kode\_Masakan** | **Nama\_Masakan** | **Ukuran** |
| R01 | Rawon Daging Sapi | 10 liter |

Jawaban Kueri

SELECT \* FROM masakan

WHERE Nama\_Masakan LIKE '%Daging%'

AND Ukuran = '10 liter';

Hasil Sreenshot

1. Tampilkan jumlah bahan paling sedikit yang dibutuhkan untuk memasak pada tabel Detail\_Bahan. [ tips: fungsi agregasi ]

Luaran yang diharapkan:

|  |
| --- |
| **Jumlah** |
| 3 |

Jawaban kueri

SELECT \* FROM detail\_bahan

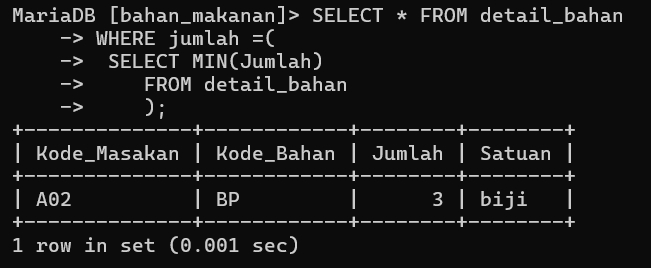
WHERE jumlah =(

SELECT MIN(Jumlah)

FROM detail\_bahan

);

Hasil Screenshoot



1. Tampilkan data lengkap pada Detail\_Bahan yang membutuhkan jumlah bahan paling banyak. [ tips: gunakan self-JOIN pada Detail\_Bahan untuk mendapatkan baris dengan jumlah tertinggi ]

Luaran yang diharapkan:

|  |  |  |  |
| --- | --- | --- | --- |
| **Kode\_Masakan** | **Kode\_Bahan** | **Jumlah** | **Satuan** |
| S01 | DA | 20 | potong |
| A02 | DA | 20 | potong |

Jawaban Kueri

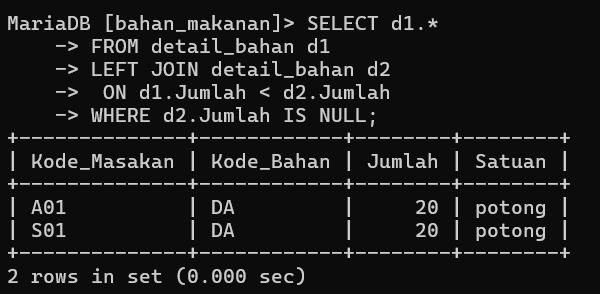
SELECT d1.\*

FROM detail\_bahan d1

LEFT JOIN detail\_bahan d2

ON d1.Jumlah < d2.Jumlah

WHERE d2.Jumlah IS NULL;

Hasil Screenshoot

1. Tampilkan jumlah masakan di masing-masing kategorinya. [ tips: pengelompokan ]

Luaran yang diharapkan:

|  |  |
| --- | --- |
| **Kategori** | **Jumlah** |
| Berkuah | 3 |
| Tak Berkuah | 2 |

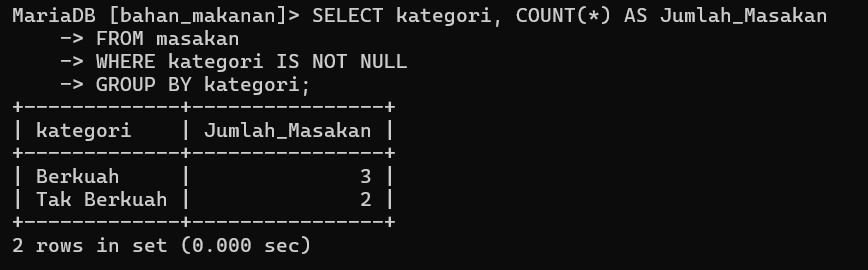
Jawaban Kueri

SELECT kategori, COUNT(\*) AS Jumlah\_Masakan

FROM masakan

WHERE kategori IS NOT NULL

GROUP BY kategori;

Hasil Screenshoot

1. Tampilkan data bahan yang belum pernah digunakan dalam masakan. [ tips: left outer join ]

Luaran yang diharapkan:

|  |  |
| --- | --- |
| **Kode\_Bahan** | **Nama\_Bahan** |
| CK | Cabai Kecil |
| CB | Cabai Besar |
| TL | Telur |

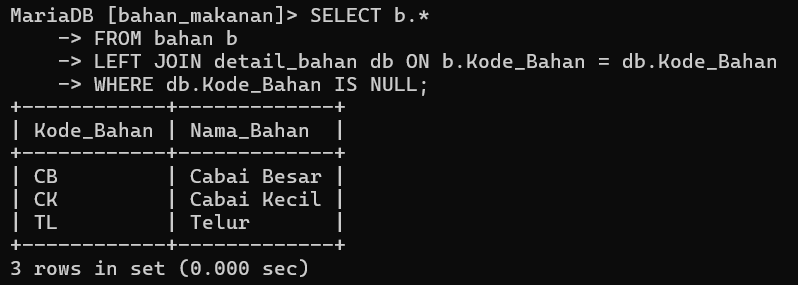
Jawaban kueri

SELECT b.\*

FROM bahan b

LEFT JOIN detail\_bahan db ON b.Kode\_Bahan = db.Kode\_Bahan

WHERE db.Kode\_Bahan IS NULL;

Hasil Screenshoot

1. Tampilkan kode bahan apa yang paling jarang (minimal 1) dalam sebuah masakan. [ tips: pengelompokan dan pengurutan ]

Luaran yang diharapkan:

|  |  |
| --- | --- |
| **Kode\_Bahan** | **Jumlah\_Pakai** |
| DS | 1 |

Jawaban kueri

SELECT Kode\_Bahan, SUM(Jumlah) AS Total\_Jumlah

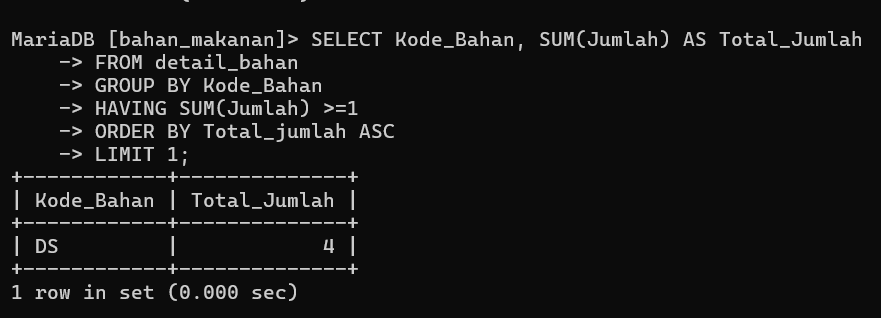
FROM detail\_bahan

GROUP BY Kode\_Bahan

HAVING SUM(Jumlah) >=1

ORDER BY Total\_jumlah ASC

LIMIT 1;

Hasil Screenshoot

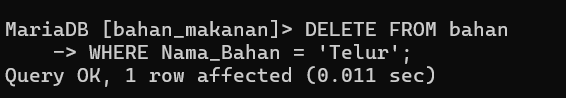
1. Hapus data Bahan yang bernama Telur.

Jawaban Kueri

DELETE FROM bahan

WHERE Nama\_Bahan = 'Telur';

Hasil Screenshoot



1. Hapus data Detail\_Bahan yang mengandung bahan daging sapi. [ tips: hapus dan join melibatkan dua tabel ]

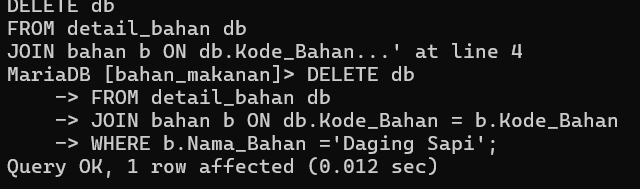
Jawaban kueri

DELETE db

FROM detail\_bahan db

JOIN bahan b ON db.Kode\_Bahan = b.Kode\_Bahan

WHERE b.Nama\_Bahan ='Daging Sapi';

Hasil screenshoot

1. Ubahlah data jumlah bahan pada masakan yang mengandung bahan BM dengan jumlah naik sebesar 30%.

Jawaban kueri

UPDATE Detail\_Bahan

SET Jumlah = ROUND(Jumlah \* 1.3)

WHERE Kode\_Bahan = 'BM';

Hasil Screenshoot

