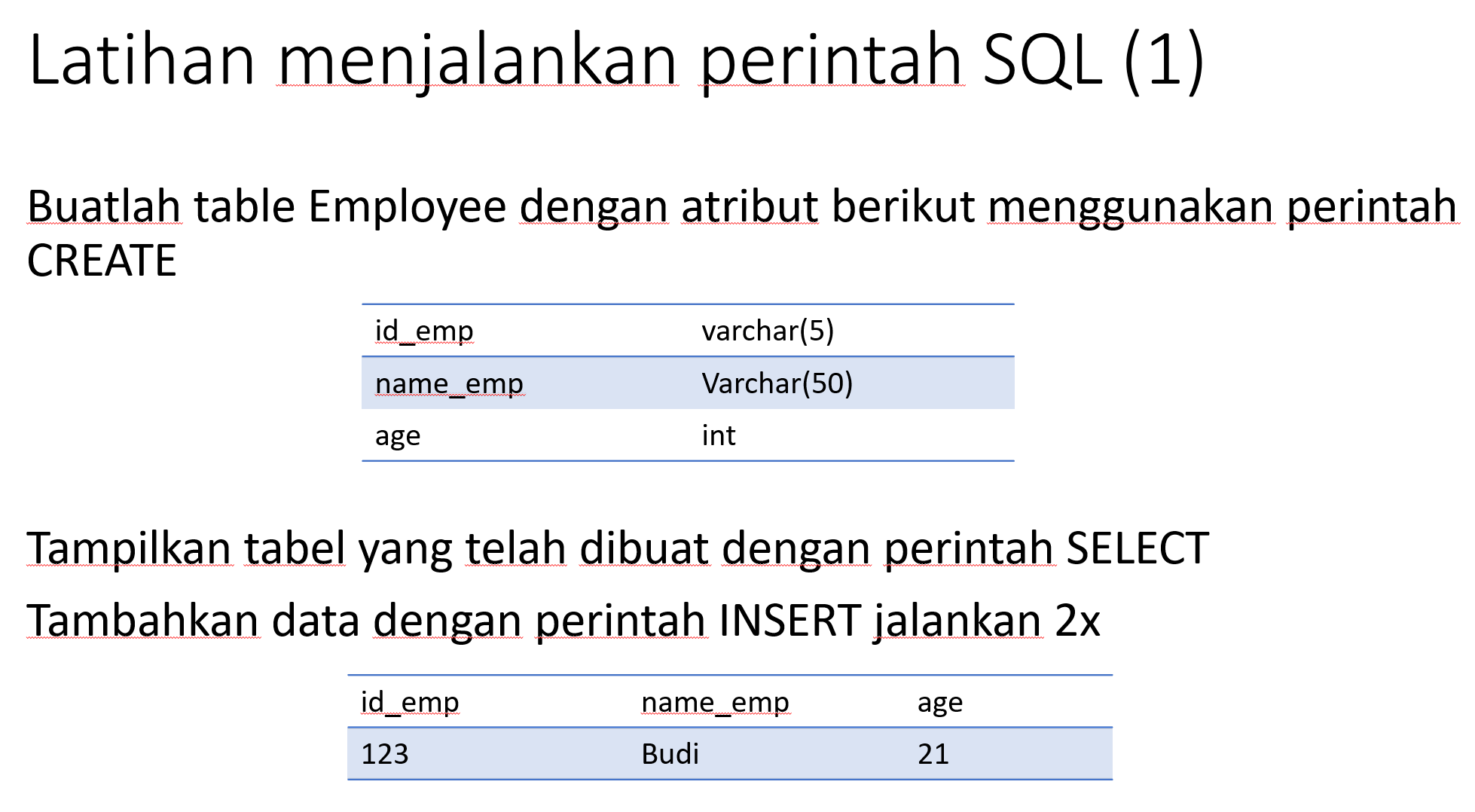
Dandy Tri Widianto – 2023071011

**Tugas 2: Latihan Perintah SQL**

****

**Buat database dan tabel:**

CREATE DATABASE employeeDB;

CREATE TABLE employee (

rowid INT,

id\_emp VARCHAR(5),

name\_emp VARCHAR(50),

age INT

);

**Tampilkan Tabel:**

SELECT \* FROM employeedb.employee;

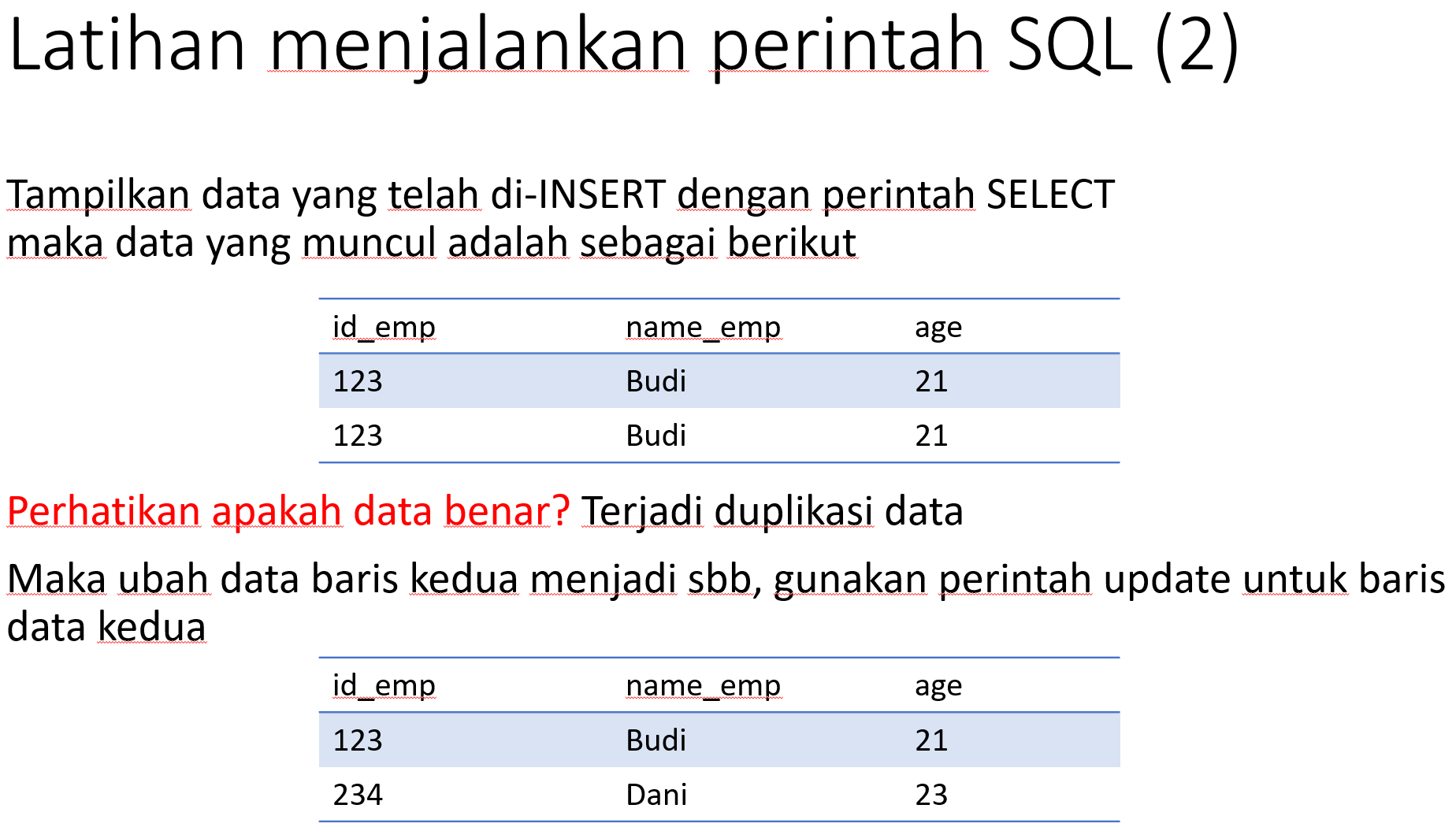
* Hasil:



**Tambah Data:**

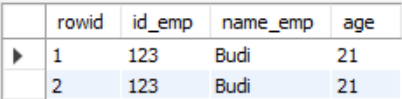
INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (1, '123', 'Budi', 21);



**Tampilkan tabel:**

SELECT \* FROM employeedb.employee;



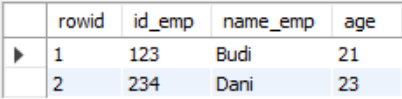
**Terjadi duplikasi data, maka lakukan UPDATE:**

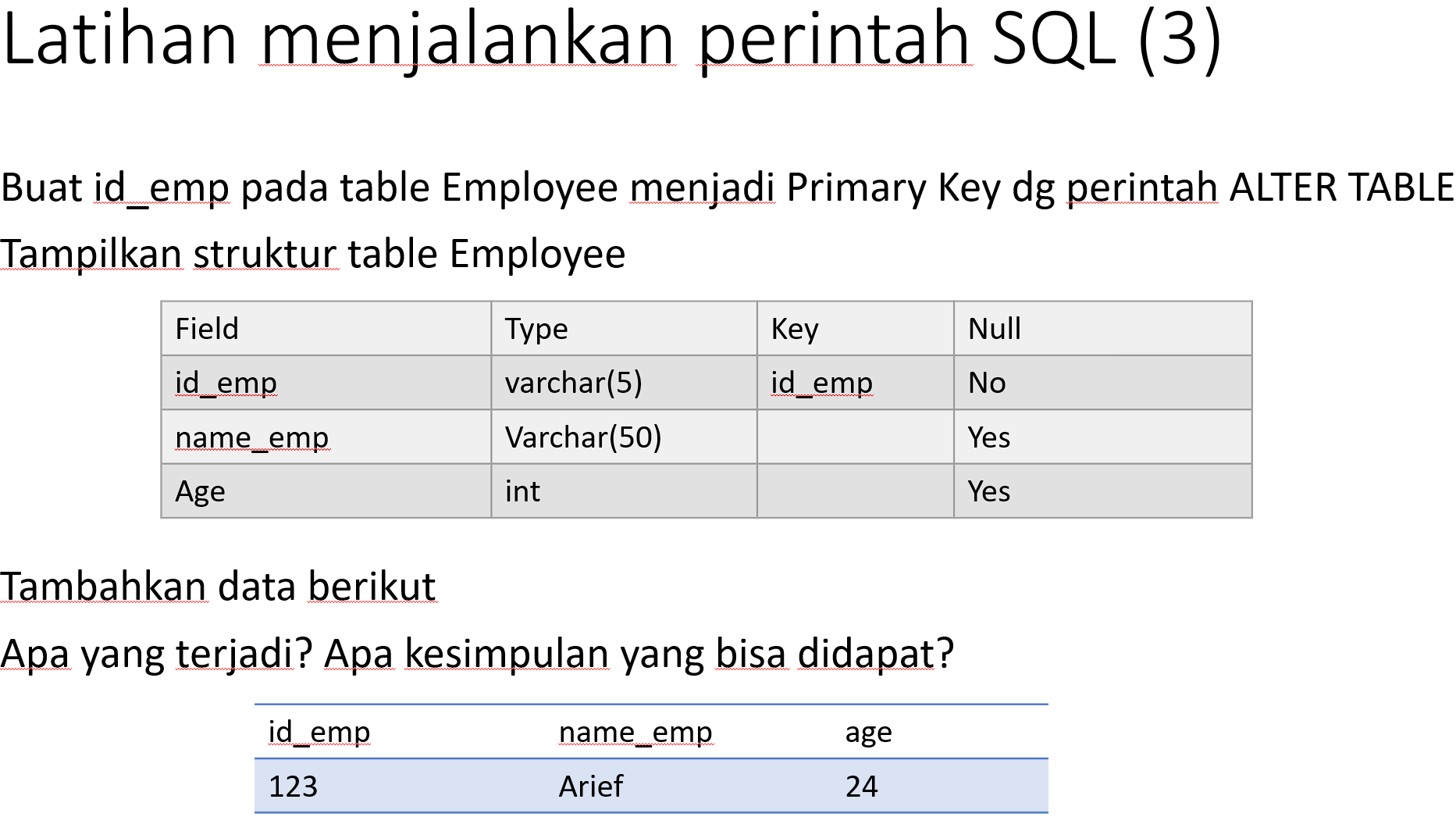
UPDATE employee

SET rowid = 2, id\_emp = 234, name\_emp = 'Dani', age= 23

WHERE rowid = 2;

* Hasil:



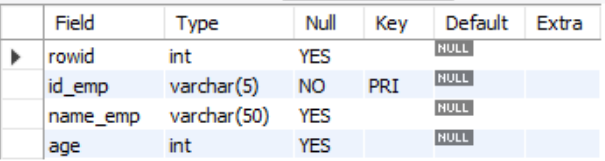


**id\_emp menjadi Primary Key:**

ALTER TABLE employee

ADD PRIMARY KEY (id\_emp);

**Tampilkan struktur tabel:**

****

**Tambah data:**

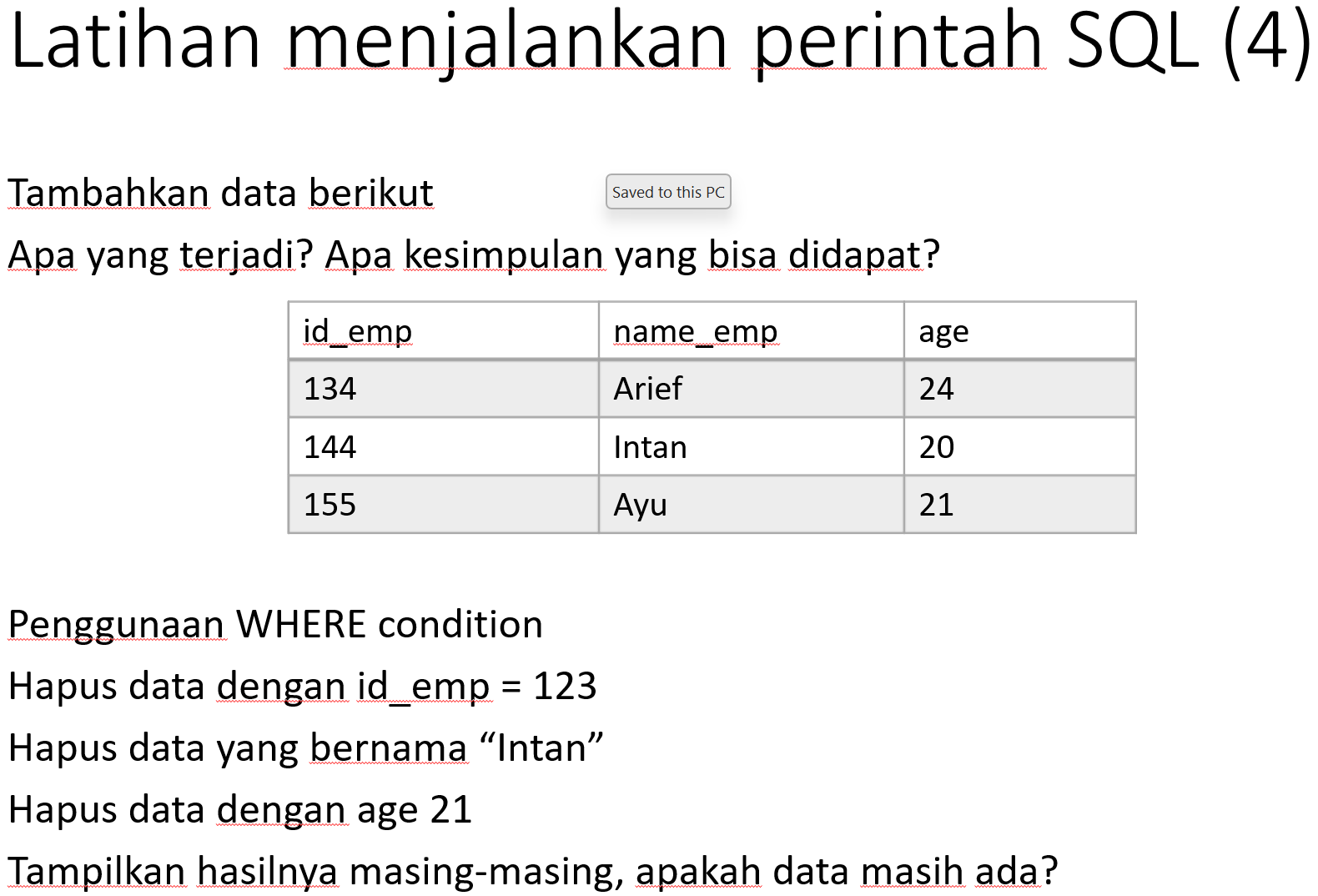
INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (3, '123', 'Arief', 24);

* Hasil:

****

Penambahan data VALUES (3, '123', 'Arief', 24); gagal.



**Tambah data:**

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (3, '134', 'Arief', 24);

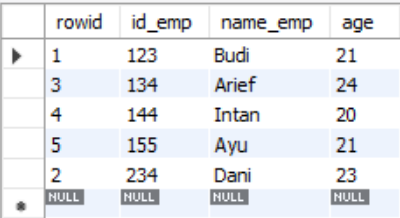
INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (4, '134', 'Intan', 20);

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

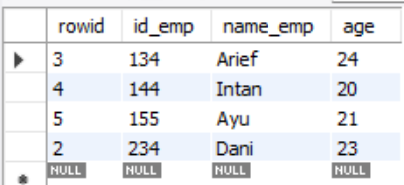
VALUES (5, '134', 'Ayu', 21);

* Hasil:

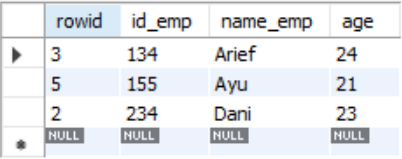
****

**Penggunaan WHERE condition:**

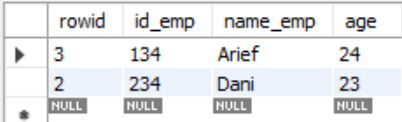
DELETE FROM employee WHERE id\_emp = '123';

****

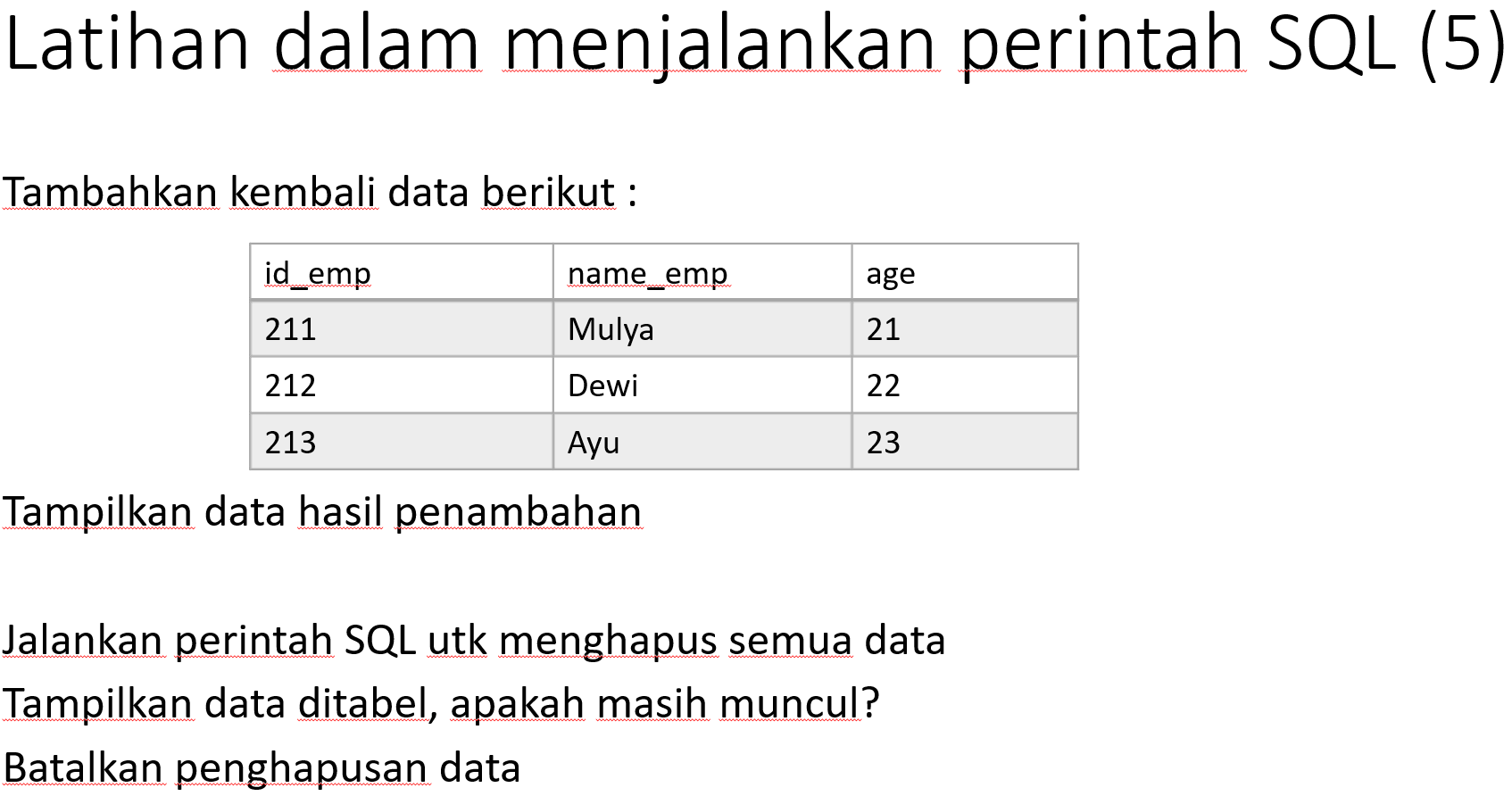
DELETE FROM employee WHERE name\_emp = 'Intan';



DELETE FROM employee WHERE age = 21;



Berdasarkan hasil, data setelah dilakukan penghapusan menjadi menghilang.



**Tambah data:**

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (3, '211', 'Mulya', 21);

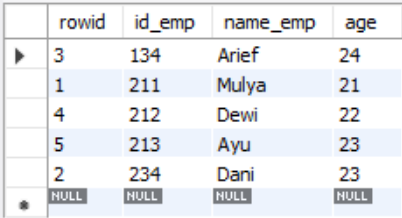
INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (4, '212', 'Dewi', 22);

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (5, '213', 'Ayu', 23);

* Hasil:

****

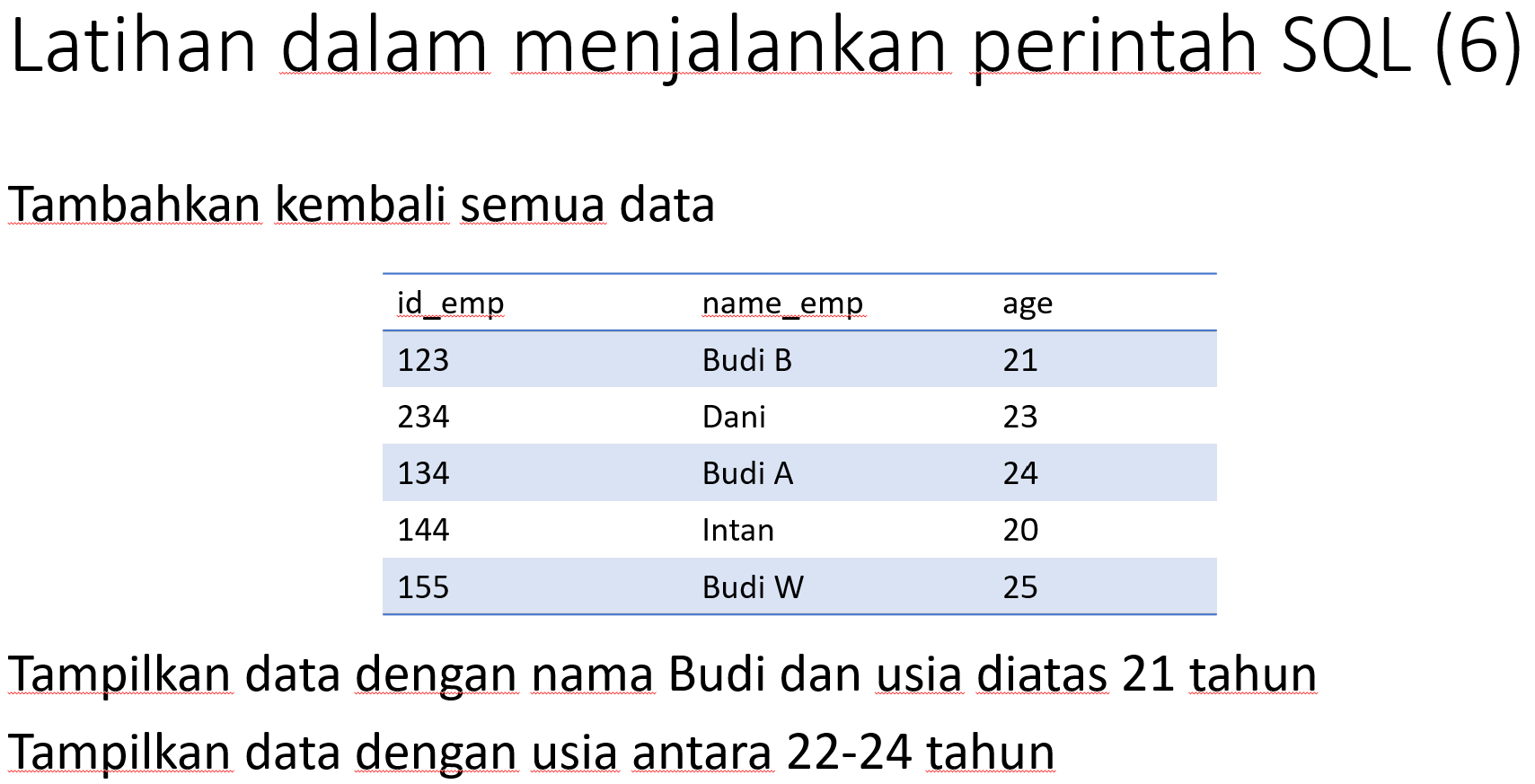
**Menghapus semua data:**

DELETE FROM employee;

* Hasil:



Data terhapus semua. Tidak ada cara mengembalikan data, karena di mySQL secara default, data tersebut hilang secara permanen dan tidak dapat langsung dipulihkan.



**Tambahkan kembali semua data:**

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (1, '123', 'Budi B', 21);

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (2, '234', 'Dani', 23);

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (3, '134', 'Budi A', 24);

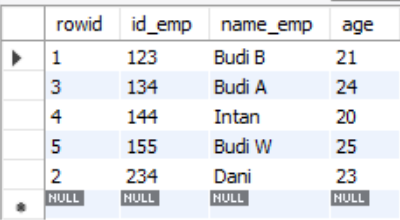
INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (4, '144', 'Intan', 20);

INSERT INTO employee (rowid, id\_emp, name\_emp, age)

VALUES (5, '155', 'Budi W', 25);

* Hasil:

****

**Tampilkan data dengan nama Budi dan Usia > 21:**

SELECT \* FROM employee

WHERE name\_emp = 'Budi' AND age > 21;

* Hasil:



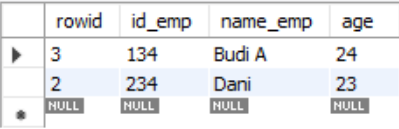
Hasilnya adalah tidak ada data yang muncul, karena dalam database nama Budi tidak spesifik, terdapat 3 nama Budi yang berbeda seperti pada tabel di atas.

**Tampilkan data dengan usia antara 22-24 tahun:**

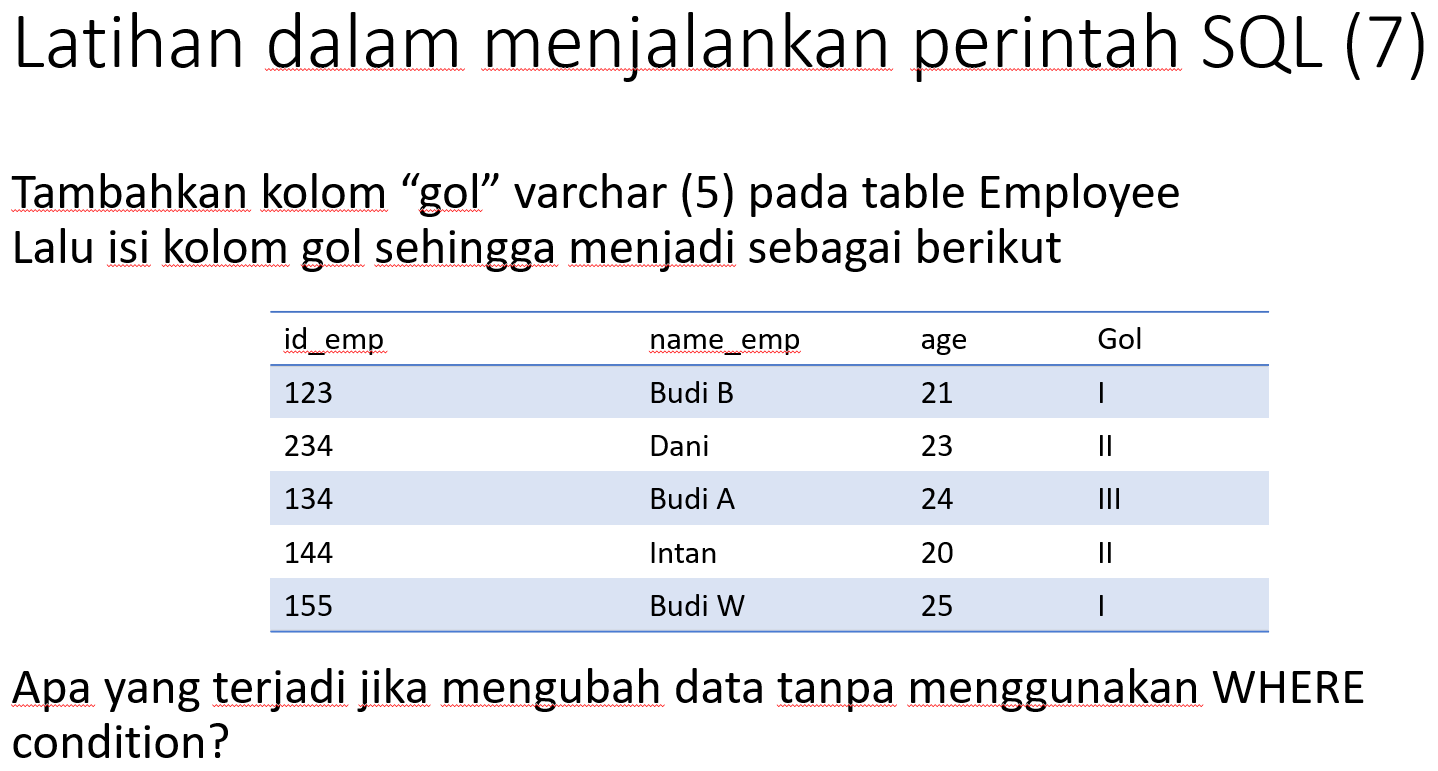
SELECT \* FROM employee

WHERE age BETWEEN 22 AND 24;

* Hasil:



Hanya terdapat dua data usia antara 22-24 tahun.

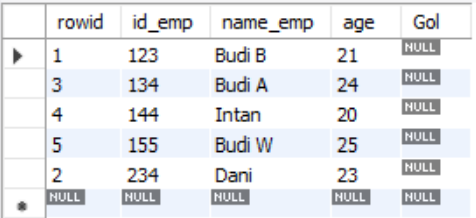


**Tambah kolom “Gol” VARCHAR(5) pada tabel:**

ALTER TABLE employee

ADD gol VARCHAR(5);

* Hasil:



**Tambah data di kolom Gol:**

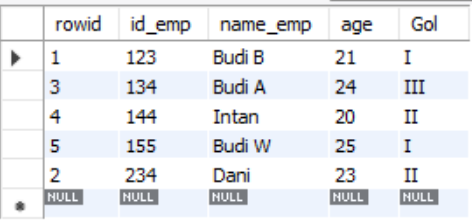
UPDATE employee

SET gol = 'I'

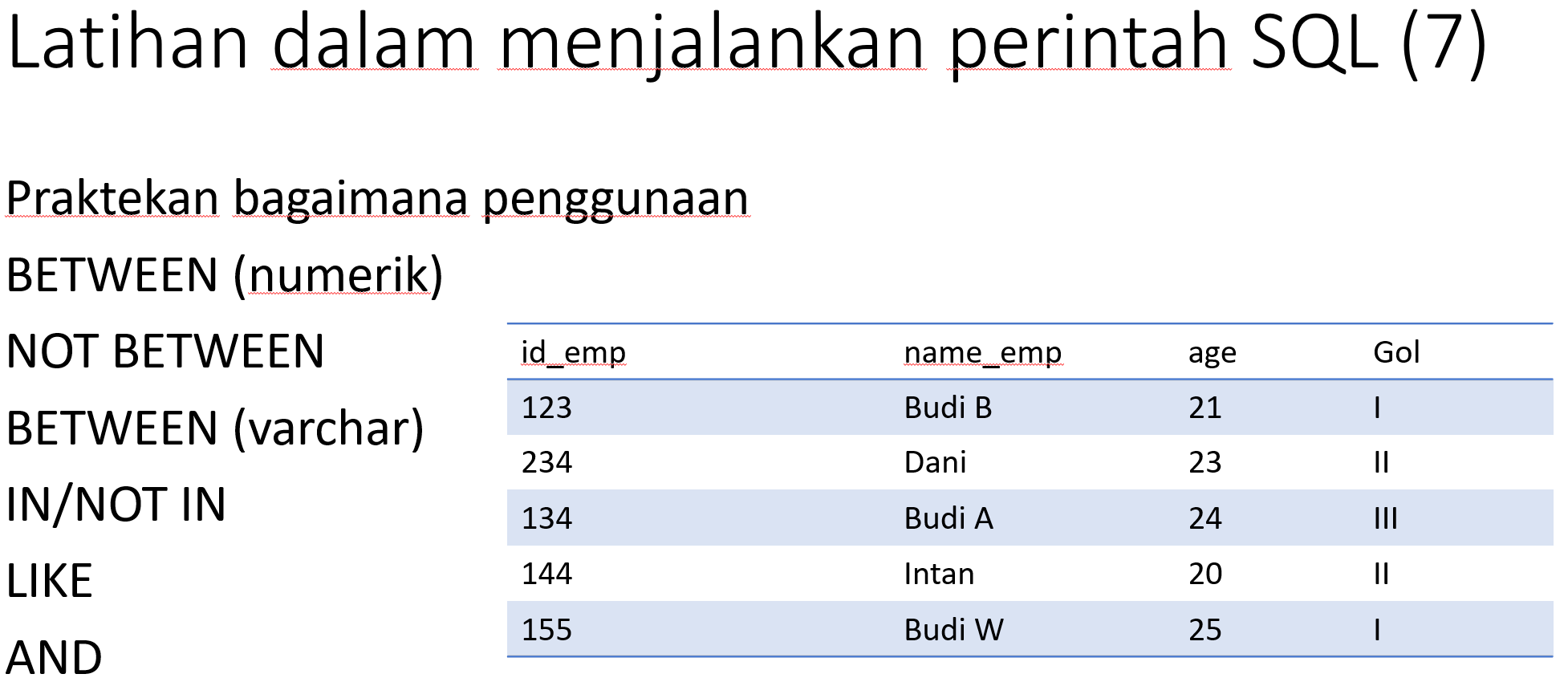
WHERE rowid = 1;

Ulangi syntax di atas sampai data gol terisi semua sesuai perintah.

* Hasil:



Jika kita tidak menggunakan WHERE condition, maka semua baris data dalam tabel tersebut akan terpengaruh oleh perintah yang diberikan.

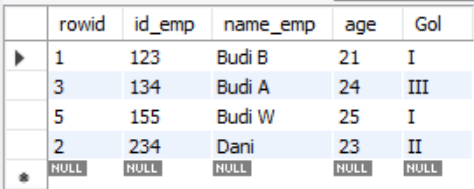


**BETWEEN (numerik):**

SELECT \* FROM employee

WHERE age BETWEEN 21 AND 25;

* Hasil:

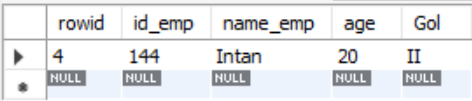


**NOT BETWEEN:**

SELECT \* FROM employee

WHERE age NOT BETWEEN 21 AND 25;

* Hasil:

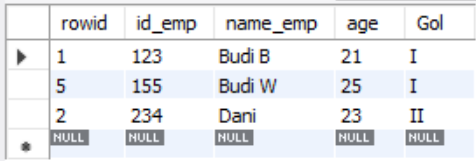


**BETWEEN (varchar):**

SELECT \* FROM employee

WHERE name\_emp BETWEEN 'Budi B' AND 'Dani';

* Hasil:



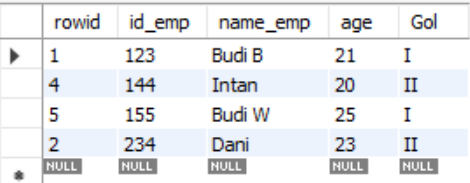
**IN/NOT IN:**

IN:

SELECT \* FROM employee

WHERE gol IN ('I', 'II');

* Hasil:

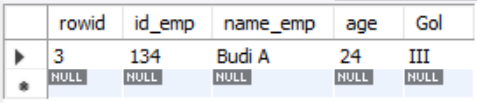


NOT IN:

SELECT \* FROM employee

WHERE gol NOT IN ('I', 'II');

* Hasil:



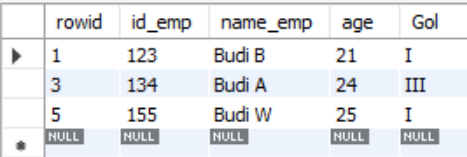
**LIKE:**

SELECT \* FROM employee

WHERE name\_emp LIKE '%Budi%';

// Jadi yang namanya ada ‘Budi’-nya akan muncul dalam tampilan tabel.

* Hasil:



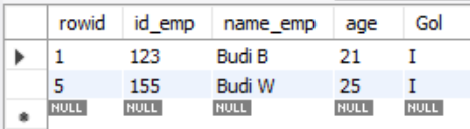
**AND:**

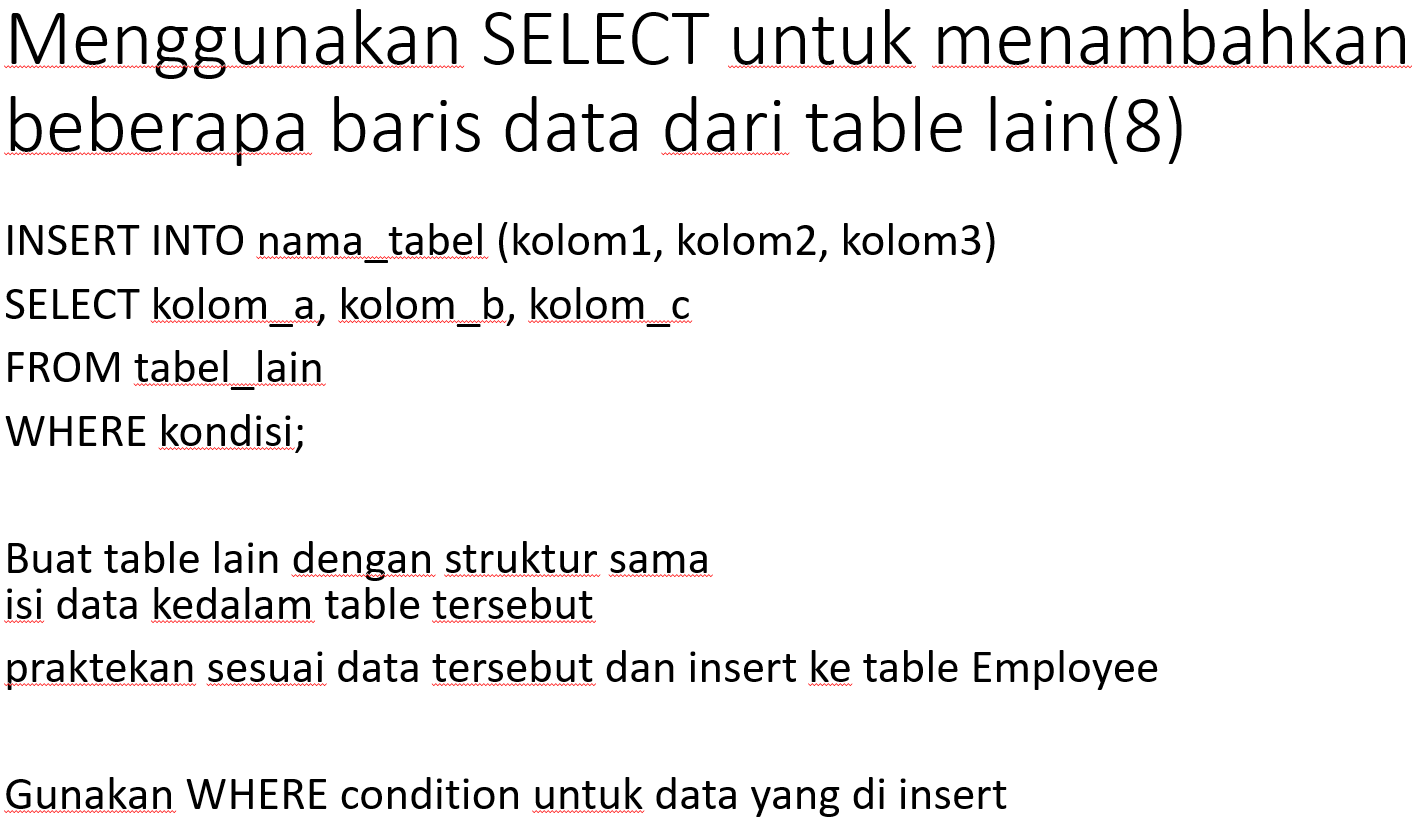
SELECT \* FROM employee

WHERE age BETWEEN 21 AND 25

AND gol = 'I';

* Hasil:





**Buat tabel baru dengan struktur yang sama:**

CREATE TABLE new\_employee AS

SELECT \* FROM employee WHERE 1=0;

* Hasil:

  
Setelah diisi data:

UPDATE new\_employee

SET gol = 'III'

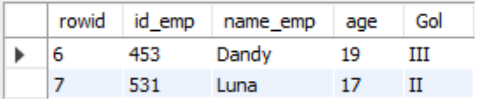
WHERE rowid = 6;

UPDATE new\_employee

SET gol = 'II'

WHERE rowid = 7;

* Hasil:



**Memindahkan data dari tabel new\_employee ke tabel employee:**

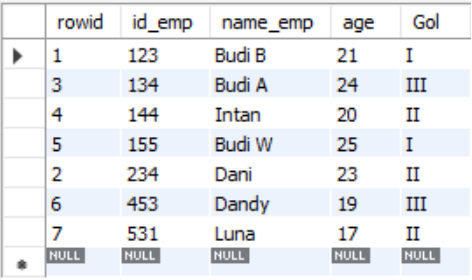
INSERT INTO employee (rowid, id\_emp, name\_emp, age, gol)

SELECT rowid, id\_emp, name\_emp, age, gol

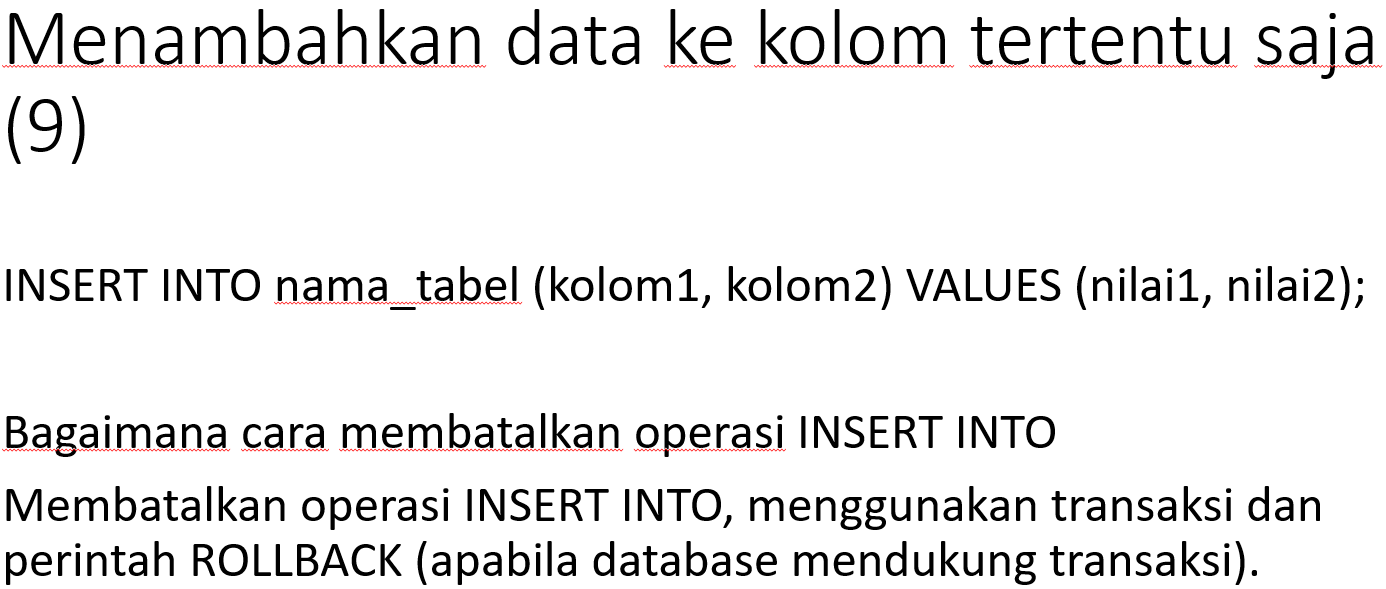
FROM new\_employee

WHERE rowid IN ('6', '7');

* Hasil:



Data dari tabel new\_employee telah dipindahkan ke tabel employee.

****

**Membatalkan perintah INSERT INTO:**

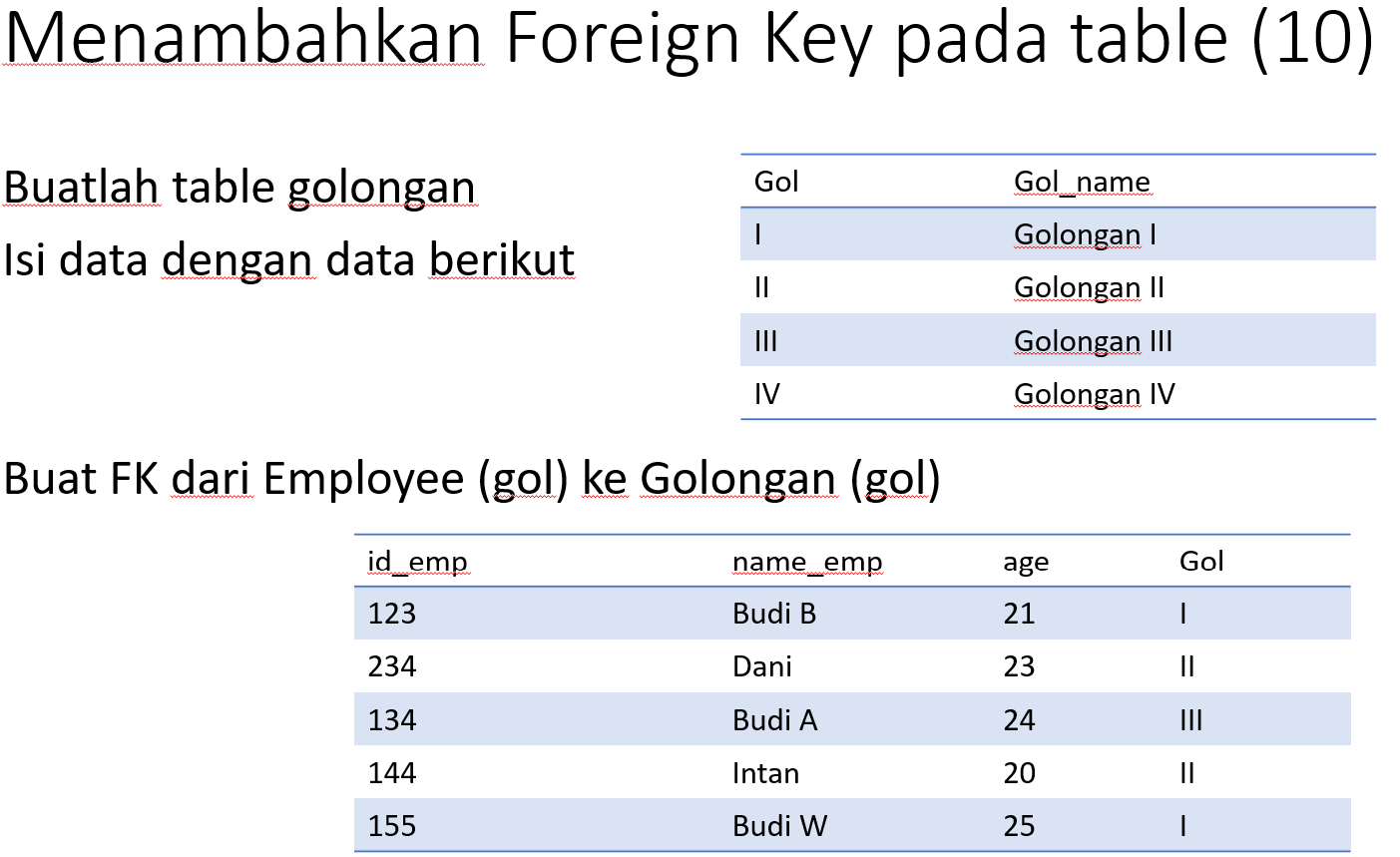
START TRANSACTION; -- Memulai transaksi

INSERT INTO employee (rowid, id\_emp, name\_emp, age, gol)

VALUES (10, 215, Pendi, 28, II);

-- Jika ada kesalahan atau kita ingin membatalkan, kita jalankan:

ROLLBACK; -- Membatalkan transaksi



**Buat tabel golongan:**

CREATE TABLE golongan (

gol VARCHAR(5),

gol\_name VARCHAR(20)

);

* Hasil:



**Tambahkan data:**

INSERT INTO golongan (gol, gol\_name)

VALUES ('I', 'Golongan I');

INSERT INTO golongan (gol, gol\_name)

VALUES ('II', 'Golongan II');

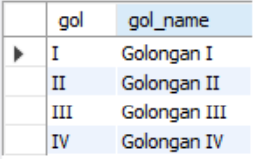
INSERT INTO golongan (gol, gol\_name)

VALUES ('III', 'Golongan III');

INSERT INTO golongan (gol, gol\_name)

VALUES ('IV', 'Golongan IV');

* Hasil:



**Buat Foreign Key dari tabel employee (gol) ke tabel golongan (gol):**

CREATE INDEX gol ON golongan(gol);

// Membuat indeks pada tabel dengan mengizinkan nilai duplikat

**Lalu dalam tabel employee lakukan perintah:**

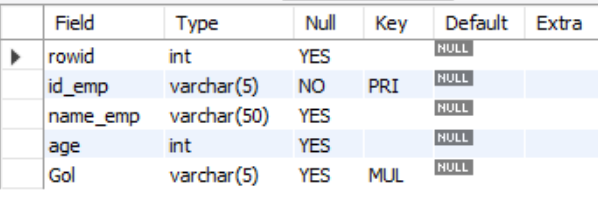
ALTER TABLE employee

ADD FOREIGN KEY (gol)

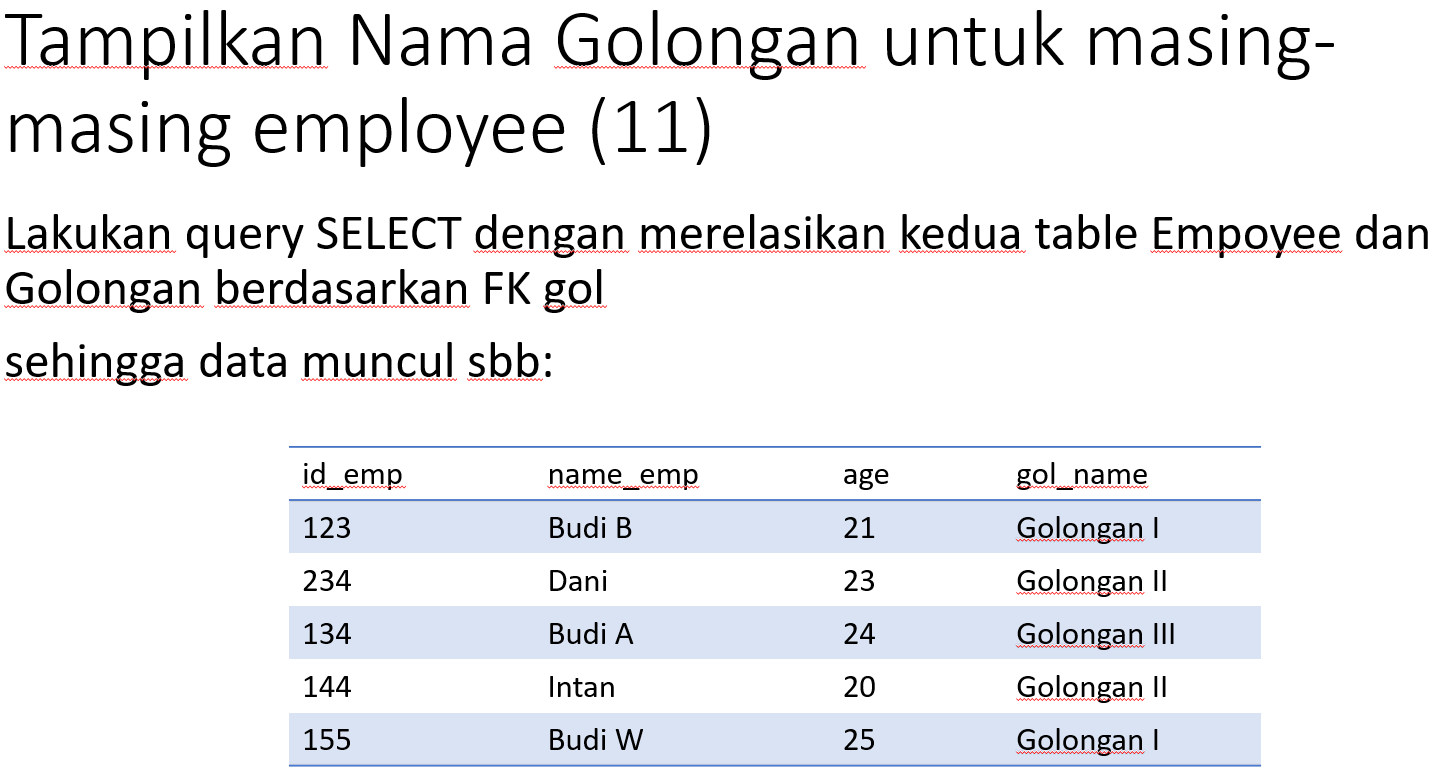
REFERENCES golongan(gol);

* Hasil:

DESC employee;

****

Sudah berhasil merelasikan antara tabel employee dengan tabel golongan dengan cara menambahkan foreign key antara tabel employee dan golongan melalui kolom ‘gol’.



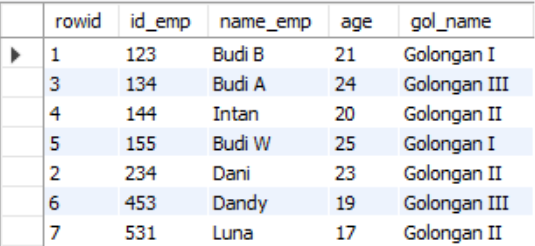
**Merelasikan kedua tabel:**

SELECT e.rowid, e.id\_emp, e.name\_emp, e.age, g.gol\_name

FROM employee e

INNER JOIN golongan g ON e.gol = g.gol;

* Hasil:



Berikut hasil merelasikan antara tabel employee dengan tabel golongan.