

Tugas 2: Latihan Perintah SQL

Latihan menjalankan perintah SQL (1)

Buatlah table Employee dengan atribut berikut menggunakan perintah CREATE

id_emp	varchar(5)
name_emp	Varchar(50)
age	int

Tampilkan tabel yang telah dibuat dengan perintah SELECT

Tambahkan data dengan perintah INSERT jalankan 2x

id_emp	name_emp	age
123	Budi	21

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:

	rowid	id_emp	name_emp	age
--	-------	--------	----------	-----

Tambah Data:

```
INSERT INTO employee (rowid, id_emp, name_emp, age)
```

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

Latihan menjalankan perintah SQL (2)

Tampilkan data yang telah di-INSERT dengan perintah SELECT
maka data yang muncul adalah sebagai berikut

id_emp	name_emp	age
123	Budi	21
123	Budi	21

Perhatikan apakah data benar? Terjadi duplikasi data

Maka ubah data baris kedua menjadi sbb, gunakan perintah update untuk baris data kedua

id_emp	name_emp	age
123	Budi	21
234	Dani	23

Tampilkan tabel:

SELECT * FROM employeedb.employee;

	rowid	id_emp	name_emp	age
▶	1	123	Budi	21
	2	123	Budi	21

Terjadi duplikasi data, maka lakukan UPDATE:

UPDATE employee

SET rowid = 2, id_emp = 234, name_emp = 'Dani', age= 23

WHERE rowid = 2;

- Hasil:

	rowid	id_emp	name_emp	age
▶	1	123	Budi	21
	2	234	Dani	23

Latihan menjalankan perintah SQL (3)

Buat id_emp pada table Employee menjadi Primary Key dg perintah ALTER TABLE
Tampilkan struktur table Employee

Field	Type	Key	Null
<u>id_emp</u>	varchar(5)	<u>id_emp</u>	No
<u>name_emp</u>	Varchar(50)		Yes
Age	int		Yes

Tambahkan data berikut

Apa yang terjadi? Apa kesimpulan yang bisa didapat?

id_emp	name_emp	age
123	Arief	24

id_emp menjadi Primary Key:

ALTER TABLE employee

ADD PRIMARY KEY (id_emp);

Tampilkan struktur tabel:

	Field	Type	Null	Key	Default	Extra
►	rowid	int	YES		NULL	
	id_emp	varchar(5)	NO	PRI	NULL	
	name_emp	varchar(50)	YES		NULL	
	age	int	YES		NULL	

Tambah data:

INSERT INTO employee (rowid, id_emp, name_emp, age)

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

- Hasil:

❌ 22 20:20:40 INSERT INTO employee (rowid, id_emp, name_emp, age) VALUES (3, '123', ... Error Code: 1062. Duplicate entry '123' for key 'employee.PRIMARY' 0.000 sec

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

Latihan menjalankan perintah SQL (4)

Tambahkan data berikut

Saved to this PC

Apa yang terjadi? Apa kesimpulan yang bisa didapat?

id_emp	name_emp	age
134	Arief	24
144	Intan	20
155	Ayu	21

Penggunaan WHERE condition

Hapus data dengan id_emp = 123

Hapus data yang bernama "Intan"

Hapus data dengan age 21

Tampilkan hasilnya masing-masing, apakah data masih ada?

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:

	rowid	id_emp	name_emp	age
▶	1	123	Budi	21
	3	134	Arief	24
	4	144	Intan	20
	5	155	Ayu	21
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

Penggunaan WHERE condition:

DELETE FROM employee WHERE id_emp = '123';

	rowid	id_emp	name_emp	age
▶	3	134	Arief	24
	4	144	Intan	20
	5	155	Ayu	21
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

DELETE FROM employee WHERE name_emp = 'Intan';

	rowid	id_emp	name_emp	age
▶	3	134	Arief	24
	5	155	Ayu	21
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

DELETE FROM employee WHERE age = 21;

	rowid	id_emp	name_emp	age
▶	3	134	Arief	24
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

Berdasarkan hasil, data setelah dilakukan penghapusan menjadi menghilang.

Latihan dalam menjalankan perintah SQL (5)

Tambahkan kembali data berikut :

id_emp	name_emp	age
211	Mulya	21
212	Dewi	22
213	Ayu	23

Tampilkan data hasil penambahan

Jalankan perintah SQL utk menghapus semua data

Tampilkan data ditabel, apakah masih muncul?

Batalkan penghapusan data

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:

	rowid	id_emp	name_emp	age
▶	3	134	Arief	24
	1	211	Mulya	21
	4	212	Dewi	22
	5	213	Ayu	23
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

Menghapus semua data:

```
DELETE FROM employee;
```

- Hasil:

	rowid	id_emp	name_emp	age
▶*	NULL	NULL	NULL	NULL

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

Latihan dalam menjalankan perintah SQL (6)

Tambahkan kembali semua data

<u>id_emp</u>	<u>name_emp</u>	age
123	Budi B	21
234	Dani	23
134	Budi A	24
144	Intan	20
155	Budi W	25

Tampilkan data dengan nama Budi dan usia diatas 21 tahun

Tampilkan data dengan usia antara 22-24 tahun

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:

	rowid	id_emp	name_emp	age
▶	1	123	Budi B	21
	3	134	Budi A	24
	4	144	Intan	20
	5	155	Budi W	25
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

Tampilkan data dengan nama Budi dan Usia > 21:

```
SELECT * FROM employee
```

```
WHERE name_emp = 'Budi' AND age > 21;
```

• Hasil:

	rowid	id_emp	name_emp	age
*	NULL	NULL	NULL	NULL

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:

	rowid	id_emp	name_emp	age
▶	3	134	Budi A	24
	2	234	Dani	23
*	NULL	NULL	NULL	NULL

Hanya terdapat dua data usia antara 22-24 tahun.

Latihan dalam menjalankan perintah SQL (7)

Tambahkan kolom “gol” varchar (5) pada table Employee
Lalu isi kolom gol sehingga menjadi sebagai berikut

<u>id_emp</u>	<u>name_emp</u>	age	Gol
123	Budi B	21	I
234	Dani	23	II
134	Budi A	24	III
144	Intan	20	II
155	Budi W	25	I

Apa yang terjadi jika mengubah data tanpa menggunakan WHERE condition?

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

ALTER TABLE employee

ADD Gol VARCHAR(5);

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	NULL
	3	134	Budi A	24	NULL
	4	144	Intan	20	NULL
	5	155	Budi W	25	NULL
	2	234	Dani	23	NULL
*	NULL	NULL	NULL	NULL	NULL

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:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	I
	3	134	Budi A	24	III
	4	144	Intan	20	II
	5	155	Budi W	25	I
	2	234	Dani	23	II
✱	NULL	NULL	NULL	NULL	NULL

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

Latihan dalam menjalankan perintah SQL (7)

Praktekan bagaimana penggunaan

BETWEEN (numerik)

NOT BETWEEN

BETWEEN (varchar)

IN/NOT IN

LIKE

AND

id_emp	name_emp	age	Gol
123	Budi B	21	I
234	Dani	23	II
134	Budi A	24	III
144	Intan	20	II
155	Budi W	25	I

BETWEEN (numerik):

SELECT * FROM employee

WHERE age BETWEEN 21 AND 25;

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	I
	3	134	Budi A	24	III
	5	155	Budi W	25	I
	2	234	Dani	23	II
✱	NULL	NULL	NULL	NULL	NULL

NOT BETWEEN:

SELECT * FROM employee

WHERE age NOT BETWEEN 21 AND 25;

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	4	144	Intan	20	II
*	NULL	NULL	NULL	NULL	NULL

BETWEEN (varchar):

SELECT * FROM employee

WHERE name_emp BETWEEN 'Budi B' AND 'Dani';

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	I
	5	155	Budi W	25	I
	2	234	Dani	23	II
*	NULL	NULL	NULL	NULL	NULL

IN/NOT IN:

IN:

SELECT * FROM employee

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

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	I
	4	144	Intan	20	II
	5	155	Budi W	25	I
	2	234	Dani	23	II
*	NULL	NULL	NULL	NULL	NULL

NOT IN:

```
SELECT * FROM employee
```

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

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	3	134	Budi A	24	III
•	NULL	NULL	NULL	NULL	NULL

LIKE:

```
SELECT * FROM employee
```

```
WHERE name_emp LIKE '%Budi%';
```

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

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	I
	3	134	Budi A	24	III
	5	155	Budi W	25	I
•	NULL	NULL	NULL	NULL	NULL

AND:

```
SELECT * FROM employee
```

```
WHERE age BETWEEN 21 AND 25
```

```
AND gol = 'I';
```

- Hasil:

	rowid	id_emp	name_emp	age	Gol
▶	1	123	Budi B	21	I
	5	155	Budi W	25	I
•	NULL	NULL	NULL	NULL	NULL

Menggunakan SELECT untuk menambahkan beberapa baris data dari table lain(8)

```
INSERT INTO nama_tabel (kolom1, kolom2, kolom3)
SELECT kolom_a, kolom_b, kolom_c
FROM tabel_lain
WHERE kondisi;
```

Buat table lain dengan struktur sama
isi data kedalam table tersebut
praktekan sesuai data tersebut dan insert ke table Employee

Gunakan WHERE condition untuk data yang di insert

Menambahkan data ke kolom tertentu saja (9)

```
INSERT INTO nama_tabel (kolom1, kolom2) VALUES (nilai1, nilai2);
```

Bagaimana cara membatalkan operasi INSERT INTO

Membatalkan operasi INSERT INTO, menggunakan transaksi dan perintah ROLLBACK (apabila database mendukung transaksi).

Menambahkan Foreign Key pada table (10)

Buatlah table golongan

Isi data dengan data berikut

<u>Gol</u>	<u>Gol_name</u>
I	<u>Golongan I</u>
II	<u>Golongan II</u>
III	<u>Golongan III</u>
IV	<u>Golongan IV</u>

Buat FK dari Employee (gol) ke Golongan (gol)

<u>id_emp</u>	<u>name_emp</u>	<u>age</u>	<u>Gol</u>
123	Budi B	21	I
234	Dani	23	II
134	Budi A	24	III
144	Intan	20	II
155	Budi W	25	I

Tampilkan Nama Golongan untuk masing-masing employee (11)

Lakukan query SELECT dengan merelasikan kedua table Employee dan Golongan berdasarkan FK gol

sehingga data muncul sbb:

<u>id_emp</u>	<u>name_emp</u>	<u>age</u>	<u>gol_name</u>
123	Budi B	21	<u>Golongan I</u>
234	Dani	23	<u>Golongan II</u>
134	Budi A	24	<u>Golongan III</u>
144	Intan	20	<u>Golongan II</u>
155	Budi W	25	<u>Golongan I</u>