

Laporan Hasil Praktikum 11
“Membuat dan Mengelola Tabel dengan
Data Definition Language (DDL)”

Bahasa Query
Kelas DP

Dosen pengampu:
Christina Deny Rumiarti S.T., M.T.I.



Disusun oleh :
Johanes Yogtan Wicaksono Raharja
215314105

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA

2022

A. TUJUAN

Mahasiswa dapat:

1. Mendeskripsikan beberapa obyek basisdata
2. Membuat table
3. Mendeskripsikan tipe data yang dapat digunakan untuk menspesifikasikan definisi kolom
4. Mengubah definisi tabel
5. Melakukan drop, rename dan truncate tabel
6. Memahami bagaimana constraint dibuat pada saat membuat tabel.

B. ANALISIS

1. Buatlah sebuah tabel bernama DEPT1 dengan struktur tabel seperti gambar di bawah ini. Cek hasilnya dengan menampilkan struktur tabel DEPT1 yang baru saja Anda buat.

a) SS Query

```
--Nomor 1
CREATE TABLE DEPT1 (id NUMBER(7)
                     CONSTRAINT dep_id_pk PRIMARY KEY,
                     nama VARCHAR2(25));

DESC dept1;
SELECT * FROM dept1;
```

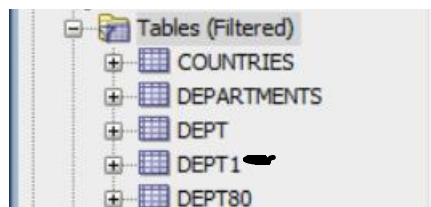
b) SS Output

Script

```
Table DEPT1 created.

Name Null?    Type
-----
ID   NOT NULL NUMBER(7)
NAMA                VARCHAR2(25)
```

Tabel



Result

ID	NAMA
----	------

c) Analisis

Perintah CREATE TABLE digunakan untuk membuat sebuah tabel DEPT1. Perintah (id NUMBER(7) CONSTRAINT dep_id_pk PRIMARY KEY, nama VARCHAR2(25)) merupakan kondisi dimana column dan datatype yang harus dimasukan ke dalam tabel. Kolom id menggunakan datatype NUMBER dengan panjang 7 dan merupakan primary key, dan kolom nama menggunakan datatype VARCHAR2 dan panjangnya 25. Perintah DESC dept1 merupakan perintah untuk mendeskripsikan struktur table. Perintah SELECT * FROM dept1 adalah perintah untuk menampilkan isi tabel.

2. Tampilkan Isi data DEPT1 dengan data yang diambil dari tabel DEPARTMENT. Pilih kolom yang sesuai dengan struktur tabel DEPT1

a) SS Query

```
--Nomor 2
INSERT INTO dept1
SELECT department_id, department_name
FROM departments;

SELECT * FROM dept1;
```

b) SS Output

Script

```
27 rows inserted.
```

Result

	ID	NAMA
1	10	Administration
2	20	Marketing
3	30	Purchasing
4	40	Human Resources
5	50	Shipping
6	60	IT
7	70	Public Relations
8	80	Sales
9	90	Executive
10	100	Finance

17	170	Manufacturing
18	180	Construction
19	190	Contracting
20	200	Operations
21	210	IT Support
22	220	NOC
23	230	IT Helpdesk
24	240	Government Sales
25	250	Retail Sales
26	260	Recruiting
27	270	Payroll

*Mengisi data yang sebelumnya dibuat di soal 1

c) Analisis

SELECT * FROM dept1; Perintah INSERT INTO digunakan untuk menyisipkan data record pada tabel. Dengan perintah SELECT yang digunakan untuk menampilkan data department_id dan department_name yang berasal dari tabel departements yang dideklarasikan oleh perintah FROM departments. Jadi perintah INSERT INTO akan memasukkan department_id dan department_name dari tabel departments ke dalam tabel dept1. SELECT * FROM dept1 adalah perintah untuk menampilkan isi tabel.

3. Buat tabel EMP1 dengan struktur seperti gambar berikut. Cek hasilnya dengan menampilkan struktur tabel EMP1 yang baru saja Anda buat.

a) SS Query

```
--Nomor 3
CREATE TABLE EMP1 (id NUMBER(7)
CONSTRAINT empl_id_pk PRIMARY KEY,
last_name VARCHAR2(25),
first_name VARCHAR2(26),
dept_id NUMBER(7)
CONSTRAINT empl_id_fk REFERENCES dept1(id));

DESC empl;
SELECT * FROM empl;
```

b) SS Output

Script

```
Table EMP1 created.

Name          Null?     Type
-----
ID            NOT NULL NUMBER(7)
LAST_NAME                VARCHAR2(25)
FIRST_NAME              VARCHAR2(26)
DEPT_ID                NUMBER(7)
```

Tabel



Result

ID	LAST_NAME	FIRST_NAME	DEPT_ID
----	-----------	------------	---------

c) Analisis

Perintah CREATE TABLE digunakan untuk membuat sebuah tabel EMP1. Perintah (id NUMBER(7) CONSTRAINT emp1_id_pk PRIMARY KEY, last_name VARCHAR2(25), first_name VARCHAR2(26), dept_id NUMBER(7) CONSTRAINT emp1_id_fk REFERENCES dept1(id)) merupakan kondisi dimana column dan datatype yang harus dimasukkan ke dalam tabel. Kolom id menggunakan datatype NUMBER dengan panjang 7 dan merupakan primary key, kolom last_name menggunakan datatype VARCHAR2 dengan panjang 25, kolom first_name menggunakan datatype VARCHAR2 dengan panjang 26, kolom dept_id dengan datatype NUMBER dengan panjang 7 dan dept_id akan menjadi foreign key dari tabel yang direferensikan yaitu tabel dept1. Perintah DESC emp1 merupakan perintah untuk mendeskripsikan struktur table. Perintah SELECT * FROM emp1 adalah perintah untuk menampilkan isi tabel

4. Modifikasi tabel EMP1 di atas, dengan menambah panjang FIRST_NAME menjadi 50. Cek hasil modifikasinya dengan menampilkan struktur tabel EMP1.

a) SS Query

```
--Nomor 4
ALTER TABLE emp1
MODIFY (first_name VARCHAR2(50));

DESC emp1;
```

b) SS Output

Script

Name	Null?	Type
ID	NOT NULL	NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(50)
DEPT_ID		NUMBER(7)

*Mengubah panjang first_name yang sebelumnya di soal 3 panjangnya 25

c) Analisis

Perintah ALTER TABLE digunakan untuk memodifikasi table emp1. Perintah MODIFY yang digunakan untuk mengubah data serta menjadi kondisi dari ALTER TABLE. Data yang dimodifikasi adalah first_name dengan datatype VARCHAR2(50). Pengecekan menggunakan perintah DESC emp1 untuk mendeskripsikan struktur tabel dan menampilkannya.

5. Buat tabel EMP2 berdasar struktur tabel EMPLOYEES. Pilih hanya kolom EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, dan DEPARTMENT_ID. Dalam tabel EMP2 beri nama masing-masing kolom tersebut sebagai ID, FIRST_NAME, LAST_NAME, SALARY, dan DEPT_ID.

a) SS Query

```
--Nomor 5
CREATE TABLE emp2
AS
SELECT employee_id "ID", first_name, last_name,
        salary, department_id "DEPT_ID"
FROM employees;

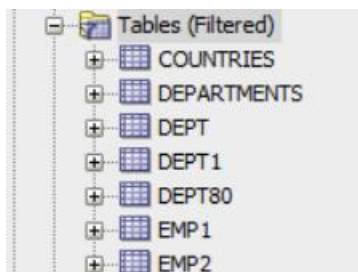
DESC emp2;
SELECT * FROM emp2;
```

b) SS Output
Script

```
Table EMP2 created.

Name          Null?     Type
-----
ID             NUMBER(6)
FIRST_NAME     VARCHAR2(20)
LAST_NAME     NOT NULL VARCHAR2(25)
SALARY         NUMBER(8,2)
DEPT_ID        NUMBER(4)
```

Tabel



Result

	ID	FIRST_NAME	LAST_NAME	SALARY	DEPT_ID
1	100	Steven	King	24000	90
2	101	Neena	Kochhar	17000	90
3	102	Lex	De Haan	17000	90
4	103	Alexander	Hunold	9000	60
5	104	Bruce	Ernst	6000	60
6	105	David	Austin	4800	60
7	106	Valli	Pataballa	4800	60
8	107	Diana	Lorentz	4200	60
9	108	Nancy	Greenberg	12008	100
10	109	Daniel	Faviet	9000	100
97	196	Alana	Walsh	3100	50
98	197	Kevin	Feeney	3000	50
99	198	Donald	OConnell	2600	50
100	199	Douglas	Grant	2600	50
101	200	Jennifer	Whalen	4400	10
102	201	Michael	Hartstein	13000	20
103	202	Pat	Fay	6000	20
104	203	Susan	Mavris	6500	40
105	204	Hermann	Baer	10000	70
106	205	Shelley	Higgins	12008	110
107	206	William	Gietz	8300	110

c) Analisis

Perintah CREATE TABLE digunakan untuk membuat sebuah tabel EMP2 dengan isi kolom tabel adalah employee_id dengan alias ID, first_name, last_name, salary, department_id dengan alias DEPT_ID dan semua data tersebut diambil dari tabel employees oleh perintah FROM employees. Perintah DESC emp2 merupakan perintah untuk mendeskripsikan struktur tabel . Perintah SELECT * FROM emp2 adalah perintah untuk menampilkan isi tabel.

6. Drop tabel EMP1.

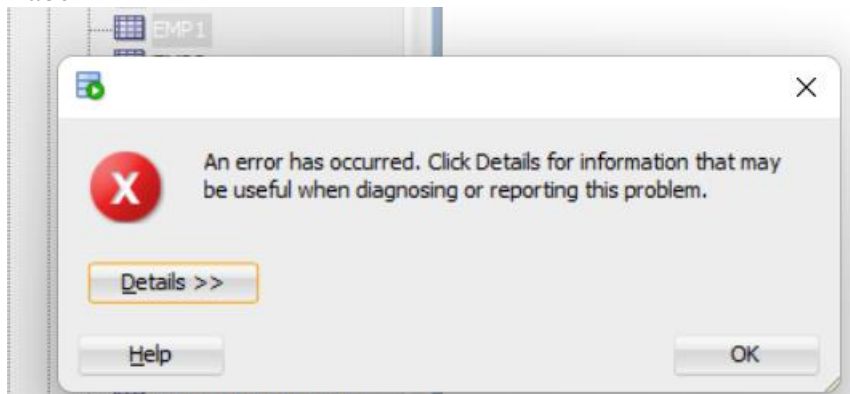
a) SS Query

```
--Nomor 6  
DROP TABLE emp1;
```

b) SS Output
Script

```
Table EMP1 dropped.
```

Tabel



*Dari SQLnya sudah eror, sudah berhasil dihapus namun tetap muncul di tabelnya

c) Analisis

Perintah DROP TABLE digunakan untuk menghapus tabel emp1. Pengecekan dengan cara deskripsi tabel dengan perintah DESC emp1. Bisa juga dengan refresh table, namun disini punya saya gak bisa.

7. Gantilah nama tabel EMP2 menjadi STAFF.

a) SS Query

```
--Nomor 7  
RENAME emp2 TO STAFF;
```

b) SS Output
Script

```
RENAME emp2 TO STAFF
```

Tabel

*Dilihat dari script sudah terubah penamaanya, namun SQLnya eror jadi tidak kelihatan perubahan tabelnya

c) Analisis

RENAME ... TO ... merupakan perintah yang digunakan untuk mengubah nama tabel emp2 menjadi STAFF

8. Drop kolom FIRST_NAME dari tabel STAFF. Cek struktur tabel STAFF.

a) SS Query

```
--Nomor 8
ALTER TABLE staff
DROP COLUMN first_name;

DESC staff;
```

b) SS Output

```
ALTER TABLE staff

DROP COLUMN first_name
Error report -
ORA-00904: "FIRST_NAME": invalid identifier
00904. 00000 - "%s: invalid identifier"
*Cause:
*Action:
Name          Null?    Type
-----
ID             NUMBER(6)
LAST_NAME NOT NULL VARCHAR2(25)
SALARY         NUMBER(8,2)
DEPT_ID        NUMBER(4)
```

*Dapat dilihat tidak ada first_name

c) Analisis

Perintah ALTER TABLE digunakan untuk memodifikasi staff. Dengan menggunakan perintah DROP COLUMN yang digunakan untuk menghapus data nama. Pengecekan menggunakan perintah DESC staff yang merupakan perintah untuk mendeskripsikan struktur tabel dan menampilkannya.

9. Cek apakah Anda masih memiliki tabel EMPLOYEES2 yang ada pada contoh 2. Jika belum, buatlah.

a) SS Query

```
--Nomor 9
DESC employees2;
```

b) SS Output
Script

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

c) Analisis

DESC merupakan perintah yang digunakan untuk mendekripsikan dan menampilkan struktur suatu tabel employees2. Perintah ini juga digunakan untuk mengecek apakah tabel yang kita deskripsikan tabel employees ada pada data tabel.

10. Buatlah tabel DEPARTMENTS2 dengan struktur sebagai berikut. Cek hasilnya dengan menampilkan struktur tabel DEPARTMENTS2 yang baru saja Anda buat.

a) SS Query

```
--Nomor 10
CREATE TABLE DEPARTMENTS2 (department_id NUMBER(10)
CONSTRAINT dept2_deptid_pk PRIMARY KEY
CONSTRAINT dept2_deptid_nn NOT NULL,
department_name VARCHAR2(30)
CONSTRAINT dept2_deptname_nn NOT NULL,
manager_id NUMBER(10)
CONSTRAINT dept2_manager_nn NOT NULL
CONSTRAINT dept2_manager_fk REFERENCES employees2
(employee_id), location_id NUMBER(10)
CONSTRAINT dept2_loc_nn NOT NULL);

DESC departments2;
SELECT * FROM departments2;
```

b) SS Output
Script

```
Table DEPARTMENTS2 created.
```

Name	Null?	Type
DEPARTMENT_ID	NOT NULL	NUMBER(10)
DEPARTMENT_NAME	NOT NULL	VARCHAR2(30)
MANAGER_ID	NOT NULL	NUMBER(10)
LOCATION_ID	NOT NULL	NUMBER(10)

Tabel

*Sudah terbentuk tabelnya, namun SQLnya eror jadi tidak kelihatan perubahan tabelnya

Result



c) Analisis

CREATE TABLE adalah perintah yang digunakan untuk membuat sebuah tabel yang dimana dalam kasus ini tabel yang dibuat adalah tabel departments2. Perintah department_id NUMBER(10) CONSTRAINT dept2_deptid_pk PRIMARY KEY CONSTRAINT dept2_deptid_nn NOT NULL, department_name VARCHAR2(30) CONSTRAINT dept2_deptname_nn NOT NULL, manager_id NUMBER(10) CONSTRAINT dept2_manager_nn NOT NULL CONSTRAINT dept2_manager_fk REFERENCES employees2 (employee_id), location_id NUMBER(10) CONSTRAINT dept2_loc_nn NOT NULL) merupakan kondisi dimana column dan datatype yang harus dimasukan ke dalam tabel. Kolom department_id menggunakan datatype NUMBER dengan panjang 10 dan merupakan primary key dan juga datanya tidak null atau kosong, kolom department_name menggunakan datatype VARCHAR2 dengan panjang 30 dan nilai department_name tidak null atau kosong, kolom manager_id menggunakan datatype NUMBER dengan panjang nilai 10 dengan nilai data tidak null atau kosong serta merupakan foreign key dari tabel yang direferensikan yaitu employees2, kolom location_id dengan datatype NUMBER dengan panjang nilai 10 dan nilai datanya tidak null atau kosong. Untuk mengecek bisa menggunakan perintah DESC departments2 yang merupakan perintah untuk mendeskripsikan struktur tabel serta menampilkannya. Perintah SELECT * FROM departments2 adalah perintah untuk menampilkan isi tabel.

11. Coba hapus tabel EMPLOYEES2, dan amati apa yang terjadi

a) SS Query

```
--Nomor 11
DROP TABLE employees2
```

b) SS Output
Script

```

Error starting at line : 75 in command -
DROP TABLE employees2
Error report -
ORA-02449: unique/primary keys in table referenced by foreign keys
02449. 00000 - "unique/primary keys in table referenced by foreign keys"
*Cause:      An attempt was made to drop a table with unique or
              primary keys referenced by foreign keys in another table.
*Action:     Before performing the above operations the table, drop the
              foreign key constraints in other tables. You can see what
              constraints are referencing a table by issuing the following
              command:
              SELECT * FROM USER_CONSTRAINTS WHERE TABLE_NAME = "tabnam";

```

c) Analisis

Perintah DROP TABLE digunakan untuk menghapus sebuah tabel yaitu tabel departments2. Tabel departments2 dapat dihapus karena tidak terdapat kunci unik atau primary key namun pada tabel ini hanya terdapat foreign key dari tabel employees2

12. Hapus tabel DEPARTMENTS2 dan amati apa yang terjadi.

a) SS Query

```

--Nomor 12
DROP TABLE departments2;

```

b) SS Output
Script

```

Table DEPARTMENTS2 dropped.

```

Tabel

*Sudah kehapus tabelnya, namun SQLnya eror jadi tidak kelihatan perubahan tabelnya

c) Analisis

Perintah DROP TABLE digunakan untuk menghapus tabel departments2. Tabel departments2 dapat dihapus karena tidak terdapat primary key namun pada tabel ini hanya terdapat foreign key dari tabel employees2

13. Ulangi langkah 11 dan amati apa yang terjadi.

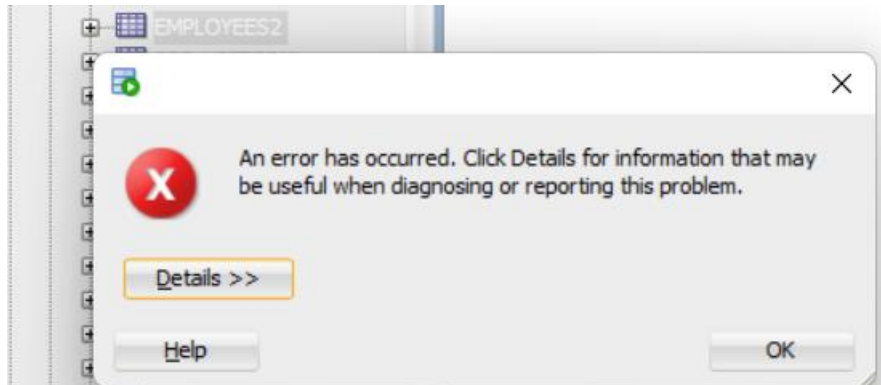
a) SS Query

```
--Nomor 13  
DROP TABLE employees2;
```

b) SS Output
Script

```
Table EMPLOYEES2 dropped.
```

Tabel



*Dari SQLnya sudah error, sudah berhasil dihapus namun tetap muncul di tabelnya

c) Analisis

Perintah DROP TABLE digunakan untuk menghapus sebuah tabel employees2. Berbeda dengan dinomor 11 pada penghapusan kali ini tabel employees2 berhasil dihapus, hal ini dikarenakan tabel yang merupakan kunci asing bagi tabel employees2 sudah tidak ada sehingga tabel ini tidak lagi terikat pada tabel manapun sehingga dapat dihapus

C. DAFTAR PUSTAKA

2021_MODUL 11 Bahasa Query – Membuat dan Mengelola Tabel dengan Data Definition Language (DDL)

