

Teori Basis Data

Pembuatan Tabel dan Constraint

Tessy Badriyah, PhD.

Pembuatan Tabel

Pembuatan Tabel dan Constraint

Tujuan Pembelajaran

- Memahami pembuatan tabel sebagai object utama dari database
- Dapat merubah definisi table
- Dapat melakukan perintah Drop, rename dan truncate pada table

Pembuatan table

- Statement CREATE TABLE

```
CREATE TABLE dept
      (deptno NUMBER(2) ,
       dname  VARCHAR2(14) ,
       loc    VARCHAR2(13)) ;
```

Table created.

- Memeriksa struktur tabel:

DESCRIBE dept

Name	Null?	Type
DEPTNO		NUMBER(2)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

Mengakses tabel dari user yang lain

- Tabel yang dimiliki oleh user yang lain, tempatnya tidak beradapada schema dari current user.
- Untuk mengakses table yang dimiliki oleh user lain, digunakan nama owner, tanda titik kemudian nama table

Pembuatan Tabel

- Contoh pembuatan tabel :

```
CREATE TABLE dept
      (deptno NUMBER(2) ,
       dname  VARCHAR2(14) ,
       loc    VARCHAR2(13)) ;
```

Table created.

Tipe table dalam Oracle

- Terdapat 2 (dua) macam table dalam Oracle Database :

- 1. Tabel User

Yaitu table yang dibuat dan dipelihara oleh user, berisi informasi user.

- 2. Data Dictionary

Yaitu table yang dibuat dan dipelihara oleh Oracle server, dan berisi informasi database.

Tipe Data

- Berikut tipe data yang dapat digunakan untuk mendefinisikan tipe data dari atribut/kolom pada tabel:

Data Type	Description
VARCHAR2 (<i>size</i>)	Variable-length character data
CHAR (<i>size</i>)	Fixed-length character data
NUMBER (<i>p</i> , <i>s</i>)	Variable-length numeric data
DATE	Date and time values
LONG	Variable-length character data up to 2 gigabytes
CLOB	Character data up to 4 gigabytes
RAW and LONG RAW	Raw binary data
BLOB	Binary data up to 4 gigabytes
BFILE	Binary data stored in an external file; up to 4 gigabytes
ROWID	A 64 base number system representing the unique address of a row in its table.

Pembuatan Tabel Menggunakan Sintak SubQuery

```
CREATE TABLE table  
    [(column, column...)]  
AS subquery;
```

```
CREATE TABLE dept80  
AS  
    SELECT employee_id, last_name,  
           salary*12 ANNSAL,  
           hire_date  
    FROM employees  
   WHERE department_id = 80;
```

Table created.

```
DESCRIBE dept80
```

Name	Null?	Type
EMPLOYEE_ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
ANNSAL		NUMBER
HIRE_DATE	NOT NULL	DATE

Perubahan table dengan ALTER TABLE

- Statement ALTER TABLE digunakan untuk :
 - Menambahkan kolom baru
 - Memodifikasi kolom yang sudah ada
 - Mendefinisikan nilai default untuk kolom baru
 - Menghapus kolom

```
ALTER TABLE table  
ADD          (column datatype [DEFAULT expr]  
              [, column datatype]...);
```

```
ALTER TABLE table  
MODIFY       (column datatype [DEFAULT expr]  
              [, column datatype]...);
```

```
ALTER TABLE table  
DROP        (column);
```

Menambahkan kolom baru

ALTER TABLE *table*
ADD *(column datatype [DEFAULT expr]*
 [, column datatype] ...);

Kolom baru

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE
149	Zlotkey	126000	29-JAN-00
174	Abel	132000	11-MAY-96
176	Taylor	103200	24-MAR-98

JOB_ID

DEPT80

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

```
ALTER TABLE dept80
ADD      (job_id VARCHAR2(9));
Table altered.
```

EMPLOYEE_ID	LAST_NAME	ANNSAL	HIRE_DATE	JOB_ID
149	Zlotkey	126000	29-JAN-00	
174	Abel	132000	11-MAY-96	
176	Taylor	103200	24-MAR-98	

Memodifikasi Kolom

```
ALTER TABLE      table  
MODIFY            (column datatype [DEFAULT expr]  
                  [, column datatype] ... );
```

```
ALTER TABLE dept80  
MODIFY      (last_name VARCHAR2(30));  
Table altered.
```

Menghapus Kolom

```
ALTER TABLE dept80  
DROP COLUMN job_id;  
Table altered.
```

Menghapus table

```
DROP TABLE dept80;  
Table dropped.
```

Merubah nama table

```
RENAME dept TO detail_dept;  
Table renamed.
```

Latihan

- Gunakan skema scott pada oracle database untuk menjawab soal2 berikut:

1. Buat table DEPARTMENT sehingga mempunyai struktur seperti ini:

Name	Null?	Type
ID		NUMBER(7)
NAME		VARCHAR2(25)

2. Dapatkan data dari table DEPT, masukkan kedalam table DEPARTMENT. Periksa hasilnya

```
SQL> SELECT * FROM DEPARTMENT;
```

ID	NAME
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

3. Buat table EMPLOYEE sehingga mempunyai struktur seperti ini :

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

4. Modifikasi struktur dari table EMPLOYEE, sehingga strukturnya menjadi :

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

5. Buat table EMPLOYEE2 yang memiliki struktur dari table EMP hanya pada field empno,ename, dan deptno. Beri nama kolom yang baru ini pada table EMPLOYEE2 : ID, LAST_NAME, DEPT_ID.

6. Drop table EMPLOYEE

7. Ubah nama table EMPLOYEE2 menjadi EMPLOYEE

Pembuatan Constraint

Tujuan Pembelajaran

- Memahami definisi Constraint
- Dapat Membuat Constraint
- Dapat Melakukan pemeliharaan Constraint

Apa itu Constraint?

- Constraint adalah batasan atau aturan yang ada pada table.
- Constraint mencegah penghapusan data dari suatu table yang mempunyai keterkaitan dengan table yang lain.
- Misal terdapat keterkaitan antara table department dengan table pegawai. Jika baris dengan nomer department '10' akan dihapus dari table department, sedangkan terdapat data pegawai yang bekerja pada department tersebut, maka dengan adanya constraint antara dua table, penghapusan tersebut tidak bisa dilakukan.

Tipe Constraint

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

Sintaks dan contoh Constraint

- Sintaks:

```
CREATE TABLE [schema.] table
    (column datatype [DEFAULT expr]
    [column_constraint],
    ...
    [table_constraint] [,...]);
```

- Contoh:

```
CREATE TABLE employees(
    employee_id  NUMBER(6),
    first_name   VARCHAR2(20),
    ...
    job_id       VARCHAR2(10) NOT NULL,
    CONSTRAINT emp_emp_id_pk
        PRIMARY KEY (EMPLOYEE_ID));
```

Constraint NOT NULL

EMPLOYEE_ID	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	90
101	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	90
102	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	90
103	Hunold	AHUNOLD	590.423.4567	03-JAN-90	IT_PROG	9000	60
104	Ernst	BERNST	590.423.4568	21-MAY-91	IT_PROG	6000	60
178	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	
200	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	10

...

20 rows selected.

NOT NULL constraint
(tidak ada baris yang
berisi nilai NULL
untuk kolom
last_name)

NOT NULL
constraint

Tidak adanya NOT
NULL constraint
(Terdapat baris yang
berisi nilai NULL untuk
kolom department_id)

- Contoh constraint NULL:

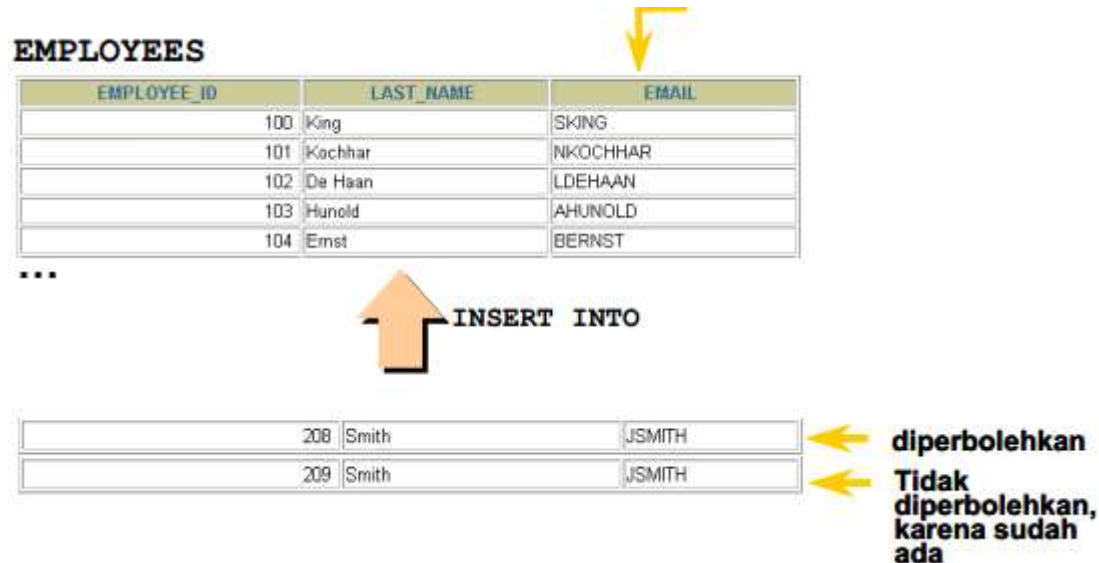
```
CREATE TABLE employees(
  employee_id  NUMBER(6),
  last_name    VARCHAR2(25) NOT NULL,
  salary       NUMBER(8,2),
  commission_pct NUMBER(2,2),
  hire_date    DATE
  CONSTRAINT emp_hire_date_nn
  NOT NULL,
  ...
```

diberi nama
sistem

Diberi
nama
user

Constraint UNIQUE

- Ilustrasi:



- Contoh:

```
CREATE TABLE employees(  
  employee_id      NUMBER(6),  
  last_name        VARCHAR2(25) NOT NULL,  
  email            VARCHAR2(25),  
  salary           NUMBER(8,2),  
  commission_pct   NUMBER(2,2),  
  hire_date        DATE NOT NULL,  
  ...  
  CONSTRAINT emp_email_uk UNIQUE(email));
```

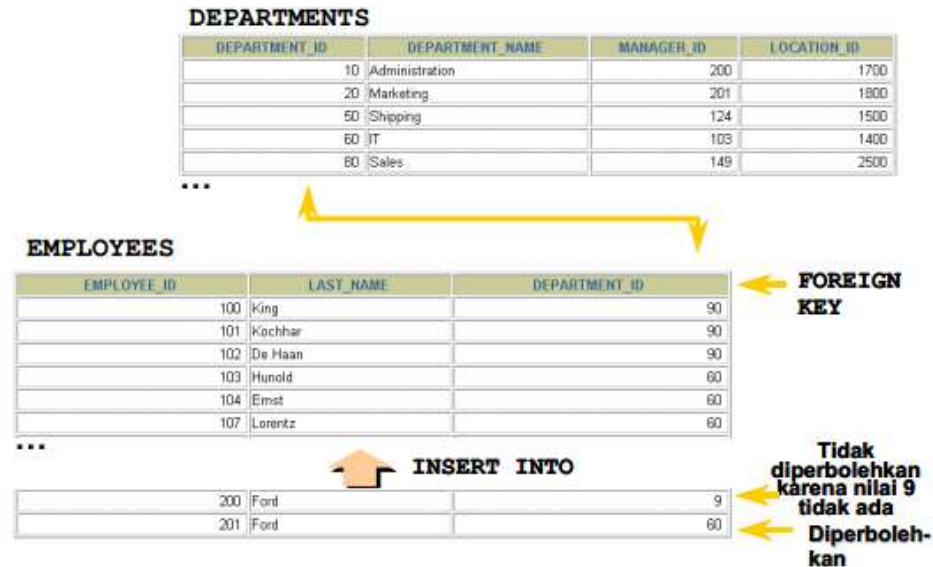

Constraint PRIMARY KEY

- Constraint PRIMARY KEY membentuk key yang unik untuk suatu table. Kolom yang didefinisikan sebagai PRIMARY KEY akan mengidentifikasi suatu baris data menjadi unik.
- Contoh penggunaan Constraint PRIMARY KEY :

```
CREATE TABLE departments(  
    department_id      NUMBER(4),  
    department_name     VARCHAR2(30)  
    CONSTRAINT dept_name_nn NOT NULL,  
    manager_id         NUMBER(6),  
    location_id         NUMBER(4),  
    CONSTRAINT dept_id_pk PRIMARY KEY(department_id));
```

Constraint FOREIGN KEY

- Ilustrasi:



- Contoh:

```
CREATE TABLE employees(  
    employee_id    NUMBER(6),  
    last_name      VARCHAR2(25) NOT NULL,  
    email          VARCHAR2(25),  
    salary         NUMBER(8,2),  
    commission_pct NUMBER(2,2),  
    hire_date      DATE NOT NULL,  
    ...  
    department_id  NUMBER(4),  
    CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
        REFERENCES departments(department_id),  
    CONSTRAINT emp_email_uk UNIQUE(email));
```

Keyword dari Constraint FOREIGN KEY

- FOREIGN KEY : Mendefinisikan kolom yang ada pada tabel anak
- REFERENCES : Mengidentifikasi tabel dan kolom dalam tabel induk
- ON DELETE CASCADE : Penghapusan baris dependent dalam tabel anak pada saat sebuah baris pada tabel induk dihapus
- ON DELETE SET NULL : Mengkonversi dependent foreign key ke nilai NULL.

Constraint CHECK

- Constraint CHECK digunakan untuk mendefinisikan suatu kondisi yang harus dipenuhi oleh tiap baris data dalam table.
- Contoh penggunaan :

```
..., salary  NUMBER(2)  
CONSTRAINT emp_salary_min  
CHECK (salary > 0),...
```

Penambahan constraint setelah tabel dibuat

- Misal ditambahkan constraint FOREIGN KEY pada tabel EMPLOYEES yang menunjukkan bahwa data manager harus valid atau berasal dari tabel EMPLOYEES.

```
ALTER TABLE    employees
ADD CONSTRAINT  emp_manager_fk
FOREIGN KEY (manager_id)
REFERENCES employees (employee_id);
Table altered.
```

Menghapus Constraint

- Cara penulisan :

ALTER TABLE table

DROP [CONSTRAINT constraint] type (column);

```
ALTER TABLE      employees
DROP CONSTRAINT    emp_manager_fk;
Table altered.
```

```
ALTER TABLE      departments
DROP PRIMARY KEY  CASCADE;
Table altered.
```

Disable dan Enable Constraint

```
ALTER TABLE      employees  
DISABLE CONSTRAINT emp_emp_id_pk CASCADE;  
Table altered.
```

```
ALTER TABLE      employees  
ENABLE CONSTRAINT  emp_emp_id_pk;  
Table altered.
```

Cascading Constraint

```
ALTER TABLE test1  
DROP (pk) CASCADE CONSTRAINTS;  
Table altered.
```

```
ALTER TABLE test1  
DROP (pk, fk, coll) CASCADE CONSTRAINTS;  
Table altered.
```


Menampilkan Constraint

```
SELECT    constraint_name, constraint_type,  
          search_condition  
FROM      user_constraints  
WHERE     table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	C	SEARCH_CONDITION
EMP_LAST_NAME_NN	C	"LAST_NAME" IS NOT NULL
EMP_EMAIL_NN	C	"EMAIL" IS NOT NULL
EMP_HIRE_DATE_NN	C	"HIRE_DATE" IS NOT NULL
EMP_JOB_NN	C	"JOB_ID" IS NOT NULL
EMP_SALARY_MIN	C	salary > 0
EMP_EMAIL_UK	U	

...

Menampilkan Kolom yang Berasosiasi dengan suatu Constraint

```
SELECT  constraint_name, column_name
FROM    user_cons_columns
WHERE   table_name = 'EMPLOYEES';
```

CONSTRAINT_NAME	COLUMN_NAME
EMP_DEPT_FK	DEPARTMENT_ID
EMP_EMAIL_NN	EMAIL
EMP_EMAIL_UK	EMAIL
EMP_EMP_ID_PK	EMPLOYEE_ID
EMP_HIRE_DATE_NN	HIRE_DATE
EMP_JOB_FK	JOB_ID
EMP_JOB_NN	JOB_ID

...