

Pertemuan 14

NOT NULL and UNIQUE Constraints

Tujuan Intruksional	:
Pokok Bahasan ini mempelajari tentang <i>Query NOT NULL and UNIQUE Constraints</i> pada <i>Database Programming</i> .	
Kompetensi Yang Diharapkan	:
Mahasiswa diharapkan memahami tentang penulisan <i>syntax SQL NOT NULL and UNIQUE Constraints</i> pada <i>Database Programming</i> .	
Waktu Pertemuan	: 100 Menit

13.1 Constraints; NOT NULL dan UNIQUE Constraints

Constraint adalah batasan atau aturan yang ada pada table.

Constraint berfungsi untuk mencegah penghapusan data dari suatu table yang mempunyai keterkaitan dengan table yang lain, dapat juga mencegah pemasukan data yang tidak benar kedalam tabel.

Syntax dasar SQL Costraints

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

Constraint Pada Level Column/Field

```
CREATE TABLE mahasiswa  
(npmmahasiswa NUMBER(10) CONSTRAINT mahasiswa_npmmahasiswa_num_pk  
PRIMARY KEY,  
first_name VARCHAR2(10) NOT NULL,  
last_name VARCHAR2(10)  
email VARCHAR2(50));
```

Ket:

Nama constraint: mahasiswa_namamahasiswa_num_pk.

namamahasiswa sebagai Primary Key.

Constraint pada tingkat Table

- Constraint tingkat tabel dicantumkan secara terpisah dari definisi kolom dalam pernyataan CREATE TABLE.
- Constraint NOT NULL ditentukan hanya pada tingkat kolom, bukan tingkat tabel.
- Constraint UNIQUE, PRIMARY KEY, FOREIGN KEY, dan CHECK dapat didefinisikan pada tingkat kolom atau tabel

Contoh:

```
CREATE TABLE mahasiswa (  
  npmmahasiswa NUMBER(6) NOT NULL,  
  first_name VARCHAR2(20),  
  last_name VARCHAR2(20),  
  phone VARCHAR2(20),  
  email VARCHAR2(10) NOT NULL,  
  CONSTRAINT mahasiswa_phone_email_uk UNIQUE (email,phone));
```

```
CREATE TABLE mahasiswa (  
  npmmahasiswa    NUMBER(10),  
  first_name      VARCHAR2(20),  
  last_name       VARCHAR2(20),  
  phone           VARCHAR2(20) CONSTRAINT phone_email_uk UNIQUE(email,phone),  
  email           VARCHAR2(10) CONSTRAINT NOT NULL,  
  CONSTRAINT emailmahasiswa_email NOT NULL,  
  CONSTRAINT mahasiswa_npmmahasiswa_pk PRIMARY KEY (npmmahasiswa);
```

COMPOSITE UNIQUE KEY VIOLATION
Composite keys harus didefinisikan pada tingkat

NAME VIOLATION
Bila menggunakan istilah CONSTRAINT,
itu harus diikuti dengan nama constraint.

NOT NULL VIOLATION NOT NULL
constraints hanya dapat didefinisikan pada tingkat

13.2 Lima Jenis *Constraint* dalam database Oracle.

a) **NOT NULL constraints**

Sebuah kolom yang mempunyai constraint NOT NULL artinya bahwa kolom itu tidak boleh null. Contohnya adalah kolom jenis kelamin itu tidak boleh null, karena setiap manusia pasti mempunyai jenis kelamin baik pria maupun wanita.

b) **UNIQUE constraints**

- Constraint Unique Key ini sebenarnya mirip dengan constraint Primary Key, yaitu sama-sama bernilai unik. Bedanya adalah kalau Unique Key ini boleh bernilai NULL sedangkan Primary Key tidak boleh bernilai NULL.
- Perbedaan lainnya adalah Unique Key dapat diaplikasikan lebih dari satu pada suatu table, sedangkan Primary Key hanya boleh satu.
- Misal: Bagi Bisnis dalam memastikan bahwa tidak ada dua orang yang memiliki alamat email yang sama. (Maka kolom email menggunakan UNIQUE)

c) **PRIMARY KEY constraints**

- Primary Key atau Kunci Utama adalah sebuah kolom atau gabungan dari beberapa kolom yang mendefinisikan sebuah record yang unik. Dalam sebuah table hanya boleh memiliki sebuah (satu) primary key, tidak boleh lebih.
- Kolom yang mempunyai Primary Key tidak boleh NULL.
- Contoh: sebuah Pegawai tidak boleh memiliki NIP (Nomor Induk Pegawai) yang lebih dari satu (tidak unik).

d) **FOREIGN KEY Constraints**

Foreign Key atau Kunci sebuah kolom yang nilainya mempunyai referensi ke table lainnya. Biasanya table yang menjadi referensi adalah tabel yang mempunyai field Primary Key. Contoh Melihat sebuah Foreign Key:

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110

- Tabel yang berisi foreign key disebut tabel "child" dan tabel yang berisi kunci yang direferensikan disebut tabel "parent".
- Primary key dari tabel DEPARTEMEN, department_id, juga muncul di tabel KARYAWAN sebagai kolom foreign key.
- Foreign Key harus sesuai dengan nilai yang ada di tabel induk atau menjadi NULL.
- Nilai primary key dapat ada tanpa nilai foreign key yang sesuai; Namun, Foreign Key harus memiliki Primary key yang sesuai.

Contoh Syntax SQL Foreign Key pada tingkat kolom:

```
CREATE TABLE employees_pg
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0) CONSTRAINT c_emps_dept_id_fk
REFERENCES departments(department_id),
email VARCHAR2(25));
```

Contoh Syntax SQL Foreign Key pada tingkat table:

```
CREATE TABLE employees_pg
(employee_id NUMBER(6,0) CONSTRAINT copy_emp_pk PRIMARY KEY,
first_name VARCHAR2(20),
last_name VARCHAR2(25),
department_id NUMBER(4,0),
```

```
email VARCHAR2(25),  
CONSTRAINT c_emps_dept_id_fk FOREIGN KEY (department_id)  
REFERENCES departments(department_id));
```

e) CHECK constraints

Check Constraint akan mengecek data atau nilai yang masuk, apabila sesuai kondisi yang telah ditentukan maka bernilai true atau boleh masuk, sedangkan kalau bernilai false maka data tidak boleh masuk.

Contoh Syntax CHECK

```
CREATE TABLE copy_job_history  
(employee_id NUMBER(6,0),  
start_date DATE,  
end_date DATE,  
job_id VARCHAR2(10),  
department_id NUMBER(4,0),  
CONSTRAINT cjhist_emp_id_st_date_pk  
PRIMARY KEY(employee_id, start_date),  
CONSTRAINT cjhist_end_ck CHECK (end_date > start_date));
```

Syntax pada kolom:

```
salary NUMBER(8,2) CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

Syntax pada Table:

```
CONSTRAINT employees_min_sal_ck CHECK (salary > 0)
```

13.3 Mengelola *Constraints*

Menambahkan Check Constraint dengan Alter Table

Syntax SQL

```
ALTER TABLE table_name  
ADD [CONSTRAINT constraint_name] type of constraint (column_name);
```

Jika kendala adalah kendala FOREIGN KEY, kata kunci REFERENSI harus disertakan dalam pernyataan.

Contoh:

```
ALTER TABLE employees_pg  
ADD CONSTRAINT emp_dept_fk FOREIGN KEY (department_id)  
REFERENCES departments (department_id)  
ON DELETE CASCADE;
```

DEPARTMENTS - Parent

DEPARTMENT_ID	DEPT_NAME	MANAGER_ID	LOCATION_ID
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	-	1700

EMPLOYEE - Child

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
205	Shelley	Higgins	110
206	William	Gietz	110