

## 2. PENANGANAN CONSTRAINT

### Objektif :

Setelah menyelesaikan bahasan ini maka disarankan dapat melakukan hal berikut :

1. Mendefinisikan constraint
2. Membuat dan memelihara constraint



## 2.1. Constraint

Oracle server menggunakan constraint/batasan untuk mencegah pemasukkan data yang tidak benar ke dalam table. jadi dapat menggunakan batasan untuk :

- Menjamin pelaksanaan aturan pada level table saat sebuah baris dimasukkan, diupdate, atau dihapus dari table tersebut.
- Mencegah penghapusan sebuah table jika terdapat ketergantungan dari table lain.
- Menyediakan aturan untuk tool Oracle, seperti Oracle Developer.

### Data Integrity Constraint

Constraint	Deskripsi
NOT NULL	Menyebutkan bahwa kolom tersebut tidak boleh bernilai null
UNIQUE	Menyebutkan bahwa sebuah kolom atau kombinasi kolom harus memiliki nilai yang unik untuk setiap barisnya.
PRIMARY KEY	Yang secara unik mengidentifikasi setiap baris dalam table
FOREIGN KEY	Membentuk dan menerapkan aturan relasi foreign key antara kolom dengan kolom table referensi
CHECK	Menyebutkan kondisi yang harus diikuti

### Panduan Untuk Constraint

Semua constraint disimpan pada data dictionary. Constraint akan mudah diingat jika diberikan nama sesuai dengan isinya. Nama constraint harus mengikuti aturan standar penamaan obyek. Jika tidak diberikan nama, Oracle akan membuat nama dengan format SYS\_Cn, dengan n adalah bilangan integer agar nama constraint menjadi unik.

Constraint dapat didefinisikan saat membuat table atau setelah table terbentuk. Constraint yang telah didefinisikan dapat dilihat pada table data dictionary USER\_CONSTRAINT.

## 2.2. Mendefinisikan Constraint

**Sintaks :** CREATE TABLE [schema.]table  
(column datatype [DEFAULT expr]  
[column\_constraint],  
.....  
[table\_constraint] [ , ..... ] ); *dengan*

:

schema sama dengan nama pemilik  
table nama table  
DEFAULT expr nilai default jika tidak terdapat nilai pada statement insert  
column nama kolom datatype tipe data dan panjangnya  
kolom column\_constraint integrity constraint sebagai bagian dari definisi  
kolom table\_constraint integrity constraint sebagai bagian dari definisi table

Constraint dapat didefinisikan pada salah satu dari kedua level berikut ini :

Level Constraint	Deskripsi
Kolom	Mereferensikan sebuah kolom dan didefinisikan dalam spesifikasi kolom yang bersangkutan. Dapat berupa integrity constraint apa saja.
Table	Mereferensikan satu atau lebih kolom dan didefinisikan terpisah dari definisi kolom pada table. Dapat berupa integrity constraint apa saja kecuali NOT NULL

**Level Kolom** column [CONSTRAINT constraint\_name] constraint\_type,

**Level Table** column,

.....

[CONSTRAINT constraint\_name] constraint\_type  
(column, .....),

### 2.3. Constraint Not Null

Constraint NOT NULL memastikan bahwa nilai null tidak diperbolehkan di kolom.

Constraint NOT NULL hanya dapat dispesifikasikan pada level kolom, bukan table. contoh

:

```
SQL> CREATE TABLE emp (  
2      empno      NUMBER(4),  
3      ename      VARCHAR2(10) NOT NULL,  
4      job        VARCHAR2(9),  
5      mgr        NUMBER(4),  
6      hiredate   DATE,  
7      sal        NUMBER(7,2),  
8      comm       NUMBER(7,2),  
9      deptno     NUMBER(7,2) NOT NULL);
```

*Lihat Video*

Karena constraint tersebut tidak diberi nama, Oracle Server akan membuat nama untuk constraint tersebut. Berikan nama constraint pada saat menspesifikasikannya

```
..... deptno NUMBER(7,2)  
        CONSTRAINT emp_deptno_nn NOT NULL .....
```

### 2.4. Constraint Unique Key

Constraint integritas UNIQUE KEY mengharuskan setiap nilai pada sebuah kolom atau sekelompok kolom (key) bernilai unik, tidak ada 2 baris pada table memiliki duplikat di kolom atau kelompok kolom yang ditentukan. Kolom (atau kelompok kolom) yang termasuk dalam pendefinisian constraint UNIQUE KEY disebut sebagai unique key. Jika key unik terdiri lebih dari 1 kolom, maka kelompok kolom tersebut disebut sebagai composite unique key.

Constraint UNIQUE KEY mengijinkan input null kecuali jika didefinisikan constraint NOT NULL pada kolom yang sama. Sebuah null pada kolom (atau semua kolom composite unique key ) tidak bertentangan dengan constraint UNIQUE Key.

Constraint UNIQUE key dapat didefinisikan pada level kolom atau table. Sebuah composite unique key dibuat menggunakan pendefinisian level table.

Contoh :

```
SQL> CREATE TABLE dept (  
2      deptno      NUMBER(2),  
3      dname       VARCHAR2(14),  
4      loc         VARCHAR2(13),  
5      CONSTRAINT  dept_dname_uk  UNIQUE(dname));
```

*Lihat Video*

## 2.5. Constraint Primary Key

Constraint primary key membuat primary key untuk table. Hanya satu primary key yang dibuat untuk setiap table. Constraint primary key adalah sebuah kolom atau kombinasi kolom yang mengidentifikasi setiap baris dalam table secara unik. Constraint ini mengharuskan nilai yang unik pada kolom atau kombinasi kolom dan memastikan tidak ada kolom yang merupakan bagian dari primary key berisi nilai null.

Constraint primary key dapat didefinisikan pada level kolom atau table. Composite primary key dibuat menggunakan definisi level table.

Contoh :

```
SQL> CREATE TABLE dept (  
2      deptno      NUMBER(2),  
3      dname       VARCHAR2(14),  
4      loc         VARCHAR2(13),  
5      CONSTRAINT  dept_dname_uk  UNIQUE(dname),  
6      CONSTRAINT  dept_deptno_pk PRIMARY KEY (deptno));
```

*Lihat Video*

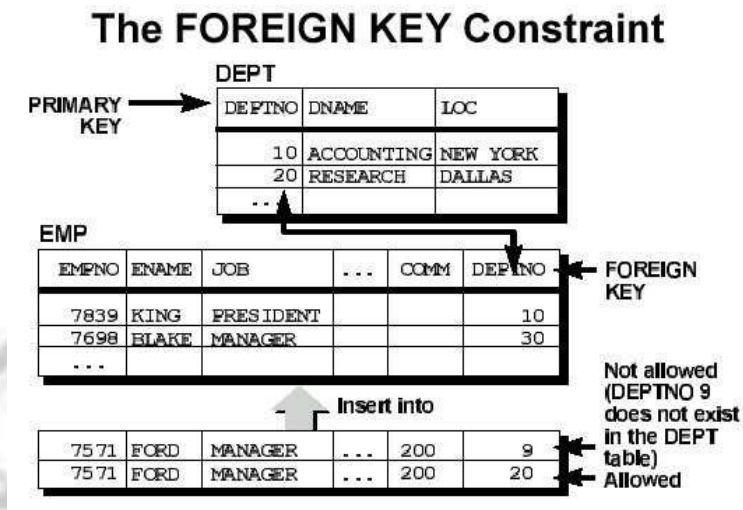
## 2.6 . Constraint Foreign Key

Foreign key atau constraint integritas referensial, menugaskan sebuah kolom atau kombinasi kolom sebagai foreign key dan membentuk sebuah relasi dengan primary key atau unique key pada table yang sama atau pada table yang berbeda.

Nilai foreign key harus sesuai dengan nilai yang ada di parent table atau bernilai null. Foreign key berdasarkan nilai data dan seutuhnya merupakan logical pointer bukan physical.



Constraint foreign key dapat didefinisikan pada level constraint kolom atau table. Composite foreign key harus dibuat dengan menggunakan pendefinisian level table.



Gambar 2.1. batasan Foreign Key

Contoh :

```
SQL> CREATE TABLE emp (
2      empno      NUMBER(4),
3      ename      VARCHAR2(10) NOT NULL,
4      job        VARCHAR2(9),
5      mgr        NUMBER(4),
6      hiredate   DATE,
7      sal        NUMBER(7,2),
8      comm       NUMBER(7,2),
9      deptno     NUMBER(7,2) NOT NULL,
10     CONSTRAINT emp_deptno_fk FOREIGN KEY (deptno)
11     REFERENCES dept (deptno));
```

*Lihat Video*

Foreign key juga dapat didefinisikan pada level kolom, yang berarti bahwa foreign key hanya terdiri dari 1 kolom. Sintaksnya berbeda pada tidak adanya keywords foreign key.

Contoh :

```
SQL> CREATE TABLE emp
      (... ,
      Deptno NUMBER(2) CONSTRAINT emp_deptno_fk REFERENCES
      dept (deptno) , ...
      );
```

Foreign key didefinisikan pada child table dan table yang berisi kolom referensi adalah parent table. Foreign key didefinisikan menggunakan kombinasi keywords berikut ini :

- FOREIGN KEY digunakan untuk mendefinisikan kolom pada child table pada constraint level table.
- REFERENCES mengidentifikasikan table dan kolom di parent table.
- ON DELETE CASCADE mengindikasikan bahwa saat baris di parent table dihapus, baris yang bergantung / berhubungan dengannya di child table juga akan dihapus. Tanpa option ON DELETE CASCADE, baris di parent table tidak dapat dihapus jika memiliki hubungan referensi dengan baris di child table.

## 2.7. Constraint Check

Constraint check mendefinisikan kondisi yang harus dipenuhi setiap baris. Kondisi yang dimaksud dapat menggunakan struktur yang sama dengan query, dengan beberapa pengecualian berikut :

- Bereferensi ke pseudocolumn CURRVAL, NEXTVAL, LEVEL, dan ROWNUM
- Memanggil fungsi SYSDATE, UID, USER, dan USERENV
- Query yang mengambil nilai dari baris lain.

Suatu kolom tunggal dapat memiliki beberapa constraint CHECK. Tidak ada batas jumlah constraint CHECK yang dapat didefinisikan terhadap sebuah kolom. Constraint CHECK dapat didefinisikan pada level kolom atau table.

Contoh :

```
SQL> CREATE TABLE dept (  
2      deptno      NUMBER(2),  
3      dname       VARCHAR2(14),  
4      loc         VARCHAR2(13),  
5      Constraint dept_loc_ck CHECK(loc in('New York','Chicago','Boston','Dallas')));
```

*Lihat Video*

## 2.8. Menambahkan Constraint

Penambahan constraint pada table yang ada menggunakan statement ALTER TABLE dengan klausa ADD.

**Sintaks :**

```
ALTER TABLE table
ADD [ CONSTRAINT constraint ] type
(column);
```

**dengan :**

type	tipe constraint
column	nama kolom yang dikenai constraint

### Tuntunan Menambahkan Constraint

- Menambah, menghapus, mengaktifkan maupun menonaktifkan constraint, namun tidak dapat memodifikasi strukturnya.
- Dapat menambahkan constraint NOT NULL pada kolom yang telah ada menggunakan klausa MODIFY di statement ALTER TABLE

**Catatan :** Pendefinisian kolom NOT NULL jika dan hanya jika table kosong, karena data tidak dapat dimasukkan bersamaan dengan penambahan kolom.

Contoh :

```
SQL> ALTER TABLE emp
2     ADD CONSTRAINT emp_mgr_fk
3     FOREIGN KEY (mgr) REFERENCES emp (empno);
```

*Lihat Video*



## 2.9. Menghapus Constraint

Untuk menghapus constraint, dapat dilakukan identifikasi nama constraint dari view data dictionary USER\_CONSTRAINT dan USER\_CONS\_COLUMNS. Kemudian gunakan statement ALTER TABLE dengan klausa DROP. Penggunaan option CASCADE pada klausa DROP menyebabkan semua constraint yang berhubungan juga ikut terhapus.

*Sintaks :*

```
ALTER TABLE table  
DROP PRIMARY KEY | UNIQUE (column) |  
CONSTRAINT constraint [CASCADE];
```

### Tuntunan Menghapus Constraint

- Gunakan klausa DISABLE pada statement CREATE TABLE maupun ALTER TABLE.
- Klausa CASCADE menonaktifkan integrity constraint yang berhubungan / bergantung.

## 2.10. Mengaktifkan Constraint

Aktifkan constraint tanpa harus membuat ulang dengan menggunakan statement ALTER TABLE dengan klausa ENABLE.

*Sintaks :*

```
ALTER TABLE table  
ENABLE CONSTRAINT constraint;
```

### Tuntunan Mengaktifkan Constraint

- Jika constraint diaktifkan, constraint tersebut diaplikasikan pada semua data di table. Semua data di table tersebut harus sesuai dengan constraint.
- Jika constraint UNIQUE key atau PRIMARY KEY diaktifkan, maka indeks UNIQUE atau PRIMARY KEY dibuat secara otomatis.
- Gunakan klausa ENABLE pada statement CREATE TABLE atau ALTER TABLE.

## 2.11. Cascading Constraint

Statement berikut mengilustrasikan penggunaan klausa CASCADE CONSTRAINT.

Asumsikan table test1 dibuat dengan sintaks sebagai berikut :

```
SQL> CREATE TABLE test1 (  
2   pk NUMBER PRIMARY KEY,  
3   fk NUMBER,  
4   col1 NUMBER,  
5   col2 NUMBER,  
6   CONSTRAINT fk_constraint FOREIGN KEY (fk) REFERENCES test1,  
7   CONSTRAINT ck1 CHECK (pk > 0 and col1 > 0 ),  
8   CONSTRAINT ck2 CHECK (col2 > 0));
```

Akan muncul pesan kesalahan untuk statement berikut ini :

```
SQL > ALTER TABLE test1 DROP (pk); ----- pk adalah parent key
```

```
SQL > ALTER TABLE test1 DROP (col1); ----- col1 merupakan referensi dari  
constraint multi kolom ck1.
```

Dengan menuliskan statement berikut ini akan menghapus kolom pk, constraint primary key, constraint foreign key, fk\_constraint dan constraint check ck1.

```
SQL > ALTER TABLE test1 DROP (pk) CASCADE CONSTRAINTS;
```

Jika semua kolom yang direferensikan constraint sudah didefinisikan pada kolom yang dihapus sudah dihapus, maka CASCADE CONSTRAINT tidak diperlukan. Sebagai contoh, asumsikan bahwa tidak ada referential constraint dari table lain yang berhubungan dengan kolom pk, maka statement berikut dapat dijalankan tanpa klausa CASCADE CONSTRAINT.

```
SQL > ALTER TABLE test1 DROP (pk, fk, col1);
```

## 2.12. Melihat Constraint

Untuk melihat semua constraint pada table, lakukan query terhadap table USER\_CONSTRAINTS.

Contoh :

```
SQL> SELECT constraint_name, constraint_type,  
2      search_condition  
3      FROM    user_constraints  
4      WHERE   table_name = 'EMP';
```

*Lihat Video*

<u>CONSTRAINT_NAME</u>	<u>C</u>	<u>SEARCH_CONDITION</u>
SYS_C00674	C	EMPNO IS NOT NULL
SYS_C00675	C	DEPTNO IS NOT NULL
EMP_EMPNO_PK	P	

### **Catatan :**

Pada tipe constraint C adalah CHECK, P adalah PRIMARY KEY, R adalah REFERENTIAL INTEGRITY, dan U adalah UNIQUE key. Perlu diperhatikan bahwa constraint NOT NULL sebenarnya adalah constraint CHECK.

Dapat dilihat nama kolom yang terlibat dalam constraint dengan melakukan query terhadap view data dictionary USER\_CONS\_COLUMNS. View ini sangat bermanfaat untuk constraint yang diberikan nama oleh system.