

Key dan Peranannya di Database Relasional

Djoni Darmawikarta

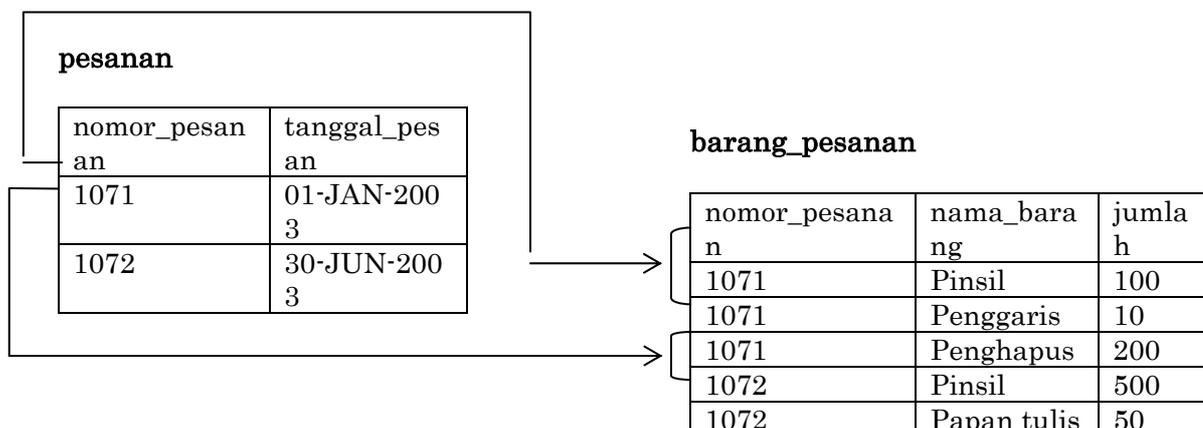
djoni_darmawikarta@yahoo.ca

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Key didalam database relasional dibutuhkan untuk menghubungkan satu tabel data dengan yang lain. Sebagai contoh (lihat diagram berikut) *nomor_pesanan* sebagai key di tabel *pesanan* digunakan didalam tabel *barang_pesanan* untuk mengidentifikasi (menghubungkan) pesanan yang berisi (terdiri dari) barang-barang tersebut.



Untuk mencapai tujuan ini (mengidentifikasi/menghubungkan), key harus unik didalam tabel. Dengan kata lain, nilai data yang ditentukan sebagai key tidak boleh dipakai di order yang lain. Jadi, nomor pesanan 1071 hanya boleh ada disatu order. Secara umum, dapat dikatakan, bahwa suatu nilai key hanya boleh dipakai disatu baris data (row).

Ada sejumlah sifat-sifat penting yang harus dipahami tentang key ini, agar rancangan dan pemakaiannya optimum.

Selanjutnya akan dibahas: Tiga macam key (Primary, Unique, dan Foreign), sifat-sifat dan penerapannya di Oracle. Agar mudah disimak dan praktis, contoh-contoh akan menyertai pembahasan.

Tiga Macam Key

Seperti sudah disinggung diatas, key, baik Primary maupun Unique, haruslah unik. Selain unik, nilai Primary key tidak boleh null (kosong, tidak diketahui, tidak dapat ditentukan); Unique key boleh null. Perlu dipahami, bahwa suatu nilai null tidak pernah sama dengan nilai null yang lain, bahkan tidak pernah sama dengan dirinya sendiri. Jadi, disuatu tabel sejumlah data (row) Unique key-nya boleh lebih dari satu yang, atau semuanya, memiliki nilai null.

Misalkan kita membuat tabel pesanan dengan SQL statement sebagai contoh Unique key, sebagai berikut:

```
CREATE TABLE pesanan (  
  nomor_pesanan VARCHAR2(6)  
  CONSTRAINT nomor_pesanan_uk UNIQUE  
  , tanggal_pesan DATE NOT NULL);
```

Kemudian kita masukkan dua buah data (baris) sebagai berikut:

```
INSERT INTO pesanan VALUES (1071, '01-JAN-2003');  
INSERT INTO pesanan (tanggal_pesan) VALUES ('30-JUN-2003');
```

Maka kedua data masuk kedalam tabel.

<i>nomor_pesanan</i>	<i>tanggal_pesanan</i>
1071	01-JAN-2003
	30-JUN-2003

Tetapi andaikan nomor_pesanan adalah Primary key, maka INSERT yang kedua akan gagal. Oracle akan mengeluarkan pesan kesalahan (error message):

```
ORA-01400: cannot insert NULL into ("PESANAN"."NOMOR_PESANAN")
```

Key boleh terdiri lebih dari satu kolom data (didalam istilah database relasional, bagian data terkecil disebut kolom (column), seperti *nomor_pesanan* dalam contoh kita). Misalkan perusahaan punya beberapa cabang dan nomor pesanan disatu kantor tidak ada hubungan dengan nomor

pesanan kantor lain, dengan kata lain nomor pesanan yang sama digunakan disemua kantor, maka untuk membedakan nomor nomor pesanan diseluruh cabang ditambahkan identitas cabang (misalnya kode_cabang) sebagai bagian dari key. Key yang terdiri lebih dari satu kolom seperti ini disebut composite key (dalam contoh kasus kita composite key adalah nomor_pesanan dan kode_cabang). Secara umum dapat dikatakan, bahwa untuk composite key, keunikannya ditinjau dari gabungan semua kolom data yang membentuknya.

Untuk Unique key, bila salah satu kolom atau lebih didalam composite key bernilai null, nilai kolom lainnya yang ada didalam composite key tidak boleh sama; tetapi diperbolehkan bila semua kolomnya bernilai null.

Memilih dan menggunakan kolom sebagai key perlu memperhatikan aturan berikut:

- Disatu tabel hanya boleh ada satu Primary key.
- Disatu tabel boleh ada lebih dari satu Unique key.
- Untuk composite key, kolom yang sudah dipakai oleh Primary key atau Unique key boleh digunakan lagi untuk Unique key yang lain, asal tidak seluruh kolom.

Didalam hubungan kedua tabel dalam contoh diatas, tabel *pesanan* sering disebut sebagai tabel orang-tua (parent), sedang *barang_pesanan* disebut tabel anak (child)

nomor_pesanan di tabel *barang_pesanan*, yang nilainya mengikuti nilai dari tabel *pesanan*, disebut Foreign key – dapat dikatakan bahwa: Foreign key adalah penghubung kedua table. Foreign key boleh berasal dari Primary dan/atau Unique key tabel parent, dan boleh composite. Istilah lain yang kadang dipakai adalah “migrated key” (maksudnya sama dengan Foreign key) Database Oracle tidak membolehkan data di tabel child dimana foreign key-nya belum ada di tabel parent. Misalnya, kita tidak bisa memasukkan data barang pesanan untuk pesanan bernomor 1073 sebelum pesanan nomor 1073 ada didalam tabel *pesanan*.

Kita buat tabel *barang_pesanan* dengan Foreign key yang menghubungkannya (me-referensi) dengan *nomor_pesanan* (key di tabel *pesanan*) sebagai berikut:

```
CREATE TABLE barang_pesanan (  
  nomor_pesanan VARCHAR2(6)  
  CONSTRAINT nomor_pesanan_fk REFERENCES  
  pesanan(nomor_pesanan),  
  nama_barang VARCHAR2(20),  
  jumlah NUMBER(6));
```

Foreign key di tabel `barang_pesanan` ini kita beri nama sama yaitu `nomor_pesanan`. Sedang nama constraint-nya adalah `nomor_pesanan_fk`.

Kemudian kita masukkan dua buah data:

```
INSERT INTO barang_pesanan VALUES (1071, 'Pensil');  
INSERT INTO barang_pesanan VALUES (1073, 'Papan tulis');
```

SQL statement yang kedua akan gagal.

Karena sifat menghalangi inilah, maka key di tabel parent (primary dan unique) diistilahkan penjaga integritas (integrity constraint) Perlu dicatat bahwa ada jenis constraint lain didalam database relasional, selain penggunaan key yang dibahas disini.

Tetapi, bila kita menggunakan Unique key dari tabel parent sebagai Foreign key di tabel child, Oracle tidak menghalangi adanya nilai null didalam Foreign key ini – maka akan ada *barang_pesanan* yang tidak memiliki pesannya! Maka kita harus berhati-hati bila menggunakan Unique key sebagai Foreign key!

Foreign key, didalam tabel child boleh digunakan sebagai Primary atau Unique key, misalnya karena Foreign key ini akan digunakan oleh tabel yang lain sebagai Foreign key (hubungan ber-rantai)

Efek Samping

Oracle otomatis membuat index untuk semua key, primary maupun unique. Index memerlukan tempat (didalam disk).

Selain membutuhkan tempat, index bisa memperlambat respon (performance), karena setiap kali ada data baru maka selain ditabelnya, di-index-nya juga harus ditambahkan key dari data baru bersangkutan; demikian juga bila kita mengubah nilai key yang sudah ada. (Mengubah nilai key sangat tidak dianjurkan, salah satu alasan adalah karena semua tabel yang terhubung juga harus segera di-update!) Ada teknik-teknik untuk menangani masalah respon dan perubahan nilai key ini, dan Oracle juga menyediakan fasilitasnya. (Tetapi tidak dibahas di tulisan ini)

Saran Pemakaian Key

Pilih kolom yang tepat, tentukan macam (primary dan/atau unique) dan jumlahnya, sesuai spesifikasi (peraturan pelaksanaan bisnis dan kebutuhan datanya), untuk menghubungkan tabel-tabel didalam database. Terapkan dengan menggunakan fasilitas bersangkutan yang tersedia didalam

database yang dipakai. Fasilitas key dan sifat-sifatnya di Microsoft SQL Server misalnya tidak sama persis dengan Oracle.