**Pertemuan ke-2**

**FUNGSI-FUNGSI DALAM PERNYATAAN SQL**

Pernyataan SQL dapat memasukan pernyataan aritmatika, yang dapat bisa memasukkan fungsi-fungsi seperti:

MAX -- nilai maksimum dari sebuah himpunan nilai

MIN -- nilai minimum dari sebuah himpunan nilai

SUM -- jumlah dari sebuah himpunan nilai

AVE -- rata-rata dari sebuah himpunan nilai

COUNT -- menghitung banyak baris

Fungsi fungsi di atas dapat digunakan dalam pernyataan SQL. Dengan demikian:

SELECT COUNT (\*)

FROM PARTS;

Akan menghasilkan jumlah baris dalam relasi (tabel) PARTS, yaitu jumlah parts, dalam hal ini adalah 5.

Klausa SELECT boleh merisi sembarang ekspresi aritmatika dan berisikan sembarang fungsi di atas. Pernyataan SQL akan memilih baris dari sebuah tabel dan mengevaluasi pernyatan yang menerapkan fungsi pada kolom dalam baris terpilih. Sebagai contoh:

SELECT SUM(QTY-NEEDED)

FROM REQ-LINES

WHERE PART-NO=’pc6’

Akan menemukan jumlah baris part ‘pc6’ yang tercatat dalam REQ-LINES. Pada contoh kita outputnya adalah 152.

SELECT SUM(QTY-NEEDED – QTY-FILLED)

FROM REQ-LINES

WHERE PART-NO=’pc6’

akan menemukan jumlah part ‘pc6’ yang tersisa untuk diisi. Dalam contoh kita akan menghasilkan 137.

Juga memungkinkan untuk mengelompokan baris dalam sebuah relasi dengan satu kolom dan menerapkan fungsi untuk setiap grup. Sebagai contoh:

SELECT PART-N0, SUM(QTY-NEEDED)

FROM REQ-LINES

GROUP BY PART-NO;

Query ini mengelompokan baris. Terdapat satu grup untuk setiap PART-NO dan semua baris dalam gurup mempunyai nilai yang sama dari PART-NO. Fungsi SUM kemudian diterapkan pada setiap grup sehingga output berisikan satu baris untuk setiap PART-NO dan total dari setiap part inventory. Output dari pernyataan ini adalah:

PART-NO SUM(QTY-NEEDED)

Pc6 162

Jw3 160

Mx7 100

Bb11 70

Terdapat satu batasan dalam pernyataan SQL yang memuat phrase GROUP. Sembarang kolom dalam klausa SELECT yang bukanlah argument sebuah fungsi, sebagi contoh PART-NO, harus mencul dalam klausa GROUP. Dengan demikian PART-NO, karena ini bagian dari klausa GROUP, dapat muncul dalam KLAUSA SELECT tapi DESCRIPTION, WIGHT atau MAX-DIM tidak bisa.

**Fungsi dalam klausa WHERE**

Fungsi dan ekpresi aritmatika dapat juga muncul dalam klausa kondisi. Ini dilakukan dengan memperluas pernyataan SELECT dengan klausa HAVING. Sebagai contoh pernyataan SQL berikut digunakan untuk menemukan bagian yang total kuantitas yang dicari melampaui 150:

SELECT PART-NO

FROM REQ-LINES

GROUP BY PART-NO

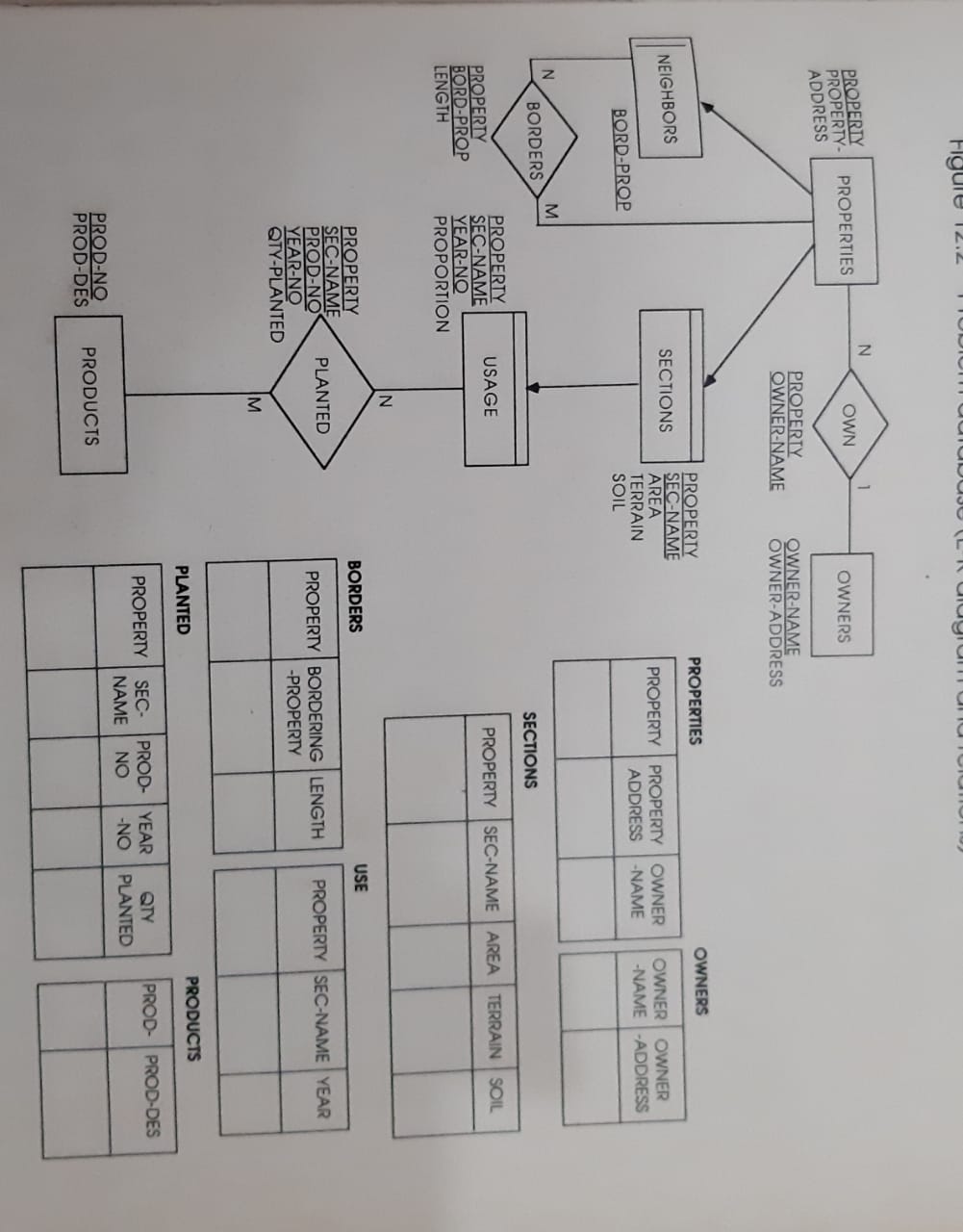
HAVING SUM(QTY-NEEDED> 150;

Sekarang semua baris dalam REQ-LINES digrupkan oleh PART-NO. Jumlah yang diminta dievalusi pada setiap grup, dan output PART-NO jika dijumlahkan melampaui 150. Pada contoh ini adalah pc6 dan jw3.

Latihan

Tulis pernyataan SQL untuk mengambil informasi berikut dari database di bawah ini (gambar 2.1).

1. Jumlah property dalam basis data tersebut
2. total jumlah produksi pada sebuah produk pada tahun tertentu
3. daftar produk
4. total area dari property yang diberikan
5. daftar property dan ukurannya
6. property yang diproduksi tak lebih dari 10.000 dari produk yang diberikan
7. property yang lebih besar dari 25.000



Gambar 2.1: Problem database (E-R diagram dan relasinya)

**PENYIMPANAN PERNYATAAN SQL**

Sering query yang sama mungkin digunakan secara berulang. Ini kemudian menjadi melelahkan untuk menginputkan ini setiap kali dibutuhkan. Kebanyakan editor SQL dapat menyimpan pernyataan SQL. Pernyataan yang disimpan dapat dieksekusi pada kemudian hari.

Pernyataan yang digunakan untuk menyimpan dan menjalankan query tergantung kepada jenis DBMS dan editor SQLnya. Sistem IBM menggunakan pernyataan

STORE <nama pernyataan>

untuk menyimpan pernyataan SQL.

Pernyataan SQL yang sudah disimpan dapat dieksekusi dengan memasukan

START <nama\_pernyataan>

Pernyataan juga dapat disimpan dengan menggunakan parapmeter selain nilai sebenarnya dalam klausa WHERE. Pada kasus tersebut pernyataan yang sama dapat dieksekusi pemeilihan baris yang berbeda setiap waktu. Dengan demikian misalkan kita membutuh sebuah pernyataan yang akan mengambil PART-NO dan QTY-NEEDED untuk ditampikan. Informasi tentang daftar permintaan yang berbeda dapat dibutuhkan setiap waktu yang dieksekusi dari pernyataan tersebut. Pada kasus tersebut kita membuat pernyataan:

SELECT PART-NO, QTY-NEEDED

FROM FREQ-LINES

WHERE REQ-NO = & 1;

Disini &1 adalah sebuah parameter yang dapat mempunyai nilai berbeda pada setiap waktu pernyataan ini dijalankan pernyataan ini disimpan dengan sebagai

STORE REQ-DETAILS

Jika kita membutuhkan detail permintaan 1 maka kita dapat memasukannya dengan

START REQ-DETAILS(1)

Untuk mendapatkan detail permintaan 2 kita masukan

START REQ-DETAILS(2)

UPDATING BASISDATA

Terdapat sejumlah cara untuk merubah isi sebuah tabel. Salah satu cara untuk menggunakan pernyataan SQL untuk merubah basisdata secara interaktif. Cara yang lain adalah menggunakan program yang dikembangkan menggunakan bahasa tingkat tinggi.

Merubah basisdata menggunakan interaktif SQL

SQL menyertakan perintah untuk memasukan baris ke dalam tabel, menghapus baris dari tabel atau merubah nilai dari baris atribut.

**Memasukan sebuah baris baru**

Untuk memasukan sebuah baris baru perlu untuk menyebutkan secara jelas nama kolom dan nilai yang ditetapkan pada nama kolom. Ini dilakukan menggunakan pernyataan INSERT

I

NSERT INTO REQUISITIONS

(REG-N0, PROJ-NO,DATA-NEEDED, WHERE-NEEDED)

VALUES (7,’PR3,870719);

Pernyataan ini memasukan sebuah baris baru pada relasi REQUSITIONS. Nama-nama pada tanda kurung menyebutkna nama kolom. Nilai dalam kurung adalah digunakan untuk atribut. Nilai diberikan mengikuti urutan yang sama dengan nama atribut.

Juga memungkinkan untuk memasukan baris yang diperoleh dari sebuah pernyataan SELECT ke dalam sebuah tabel. Sebagai contoh, misalkan kita mempunyai relasi

TEMP (REQ-NO, LOCATION)

Dan masukan pernyataan

INSERT INTO TEMP

SELECT REQ-NO, WHERE-NEEDED

FROM REQUISITIONS

WHERE PROJ-NO = ‘pr1’;

Semua baris yang diambil oleh pernyataan SELECT akan dimasukan kedalam relasi TEMP. catatan, akan tetapi, kamu tidak bisa mengambil sebuah baris dari sebuah relasi dan memasukannya kedalam relasi yang sama.

Metoda insersi ini digunakan untuk dua alasan. Pertama, metoda ini dapat digunakan untuk menyimpan haris dari pengembilan untuk keperluan berikutnya. Kedua, dapat digunakan untuk menghitung nilai baru menggunakan ekspresi matematika dan menyimnya dalam sebuah relasi baru.

Updating Baris

Untuk merubah sebuah nilai sebuah kolom perlu untuk mendefinikkan nilai baru dan memilih baris yang akan dirubah. Ini dapat dilakukan dengan pernyataan UPDATE.

UPDATE REQUISITIONS

SET DATE-NEDDED = 870720

WHERE REQ-NO=3;

Pernyataan di atas merubah nilai DATE-NEEDED dalam perminataan 3 menjadi 870720. Dapat dicatat bahwa sebuah update dapat merubah lebih dari satu baris. Ini akan merubah semua baris yang memenuhi kondisi pada WHERE.

Menghapus Baris

Disini diperlukan untuk memilih baris yang akan dihapus. Ini dilakukan dengan pernyataan DELETE.

DELETE FROM PARTS

WHERE PART-NO=’pc6’;

Pernyataan di atas menghapus PART-N0=’pc6’ dari relasi PARTS. Perintah ini dapat menghapus sebarang nomor baris. Ini akan menghapus semua baris yang ditentukan pada kondisi WHERE.