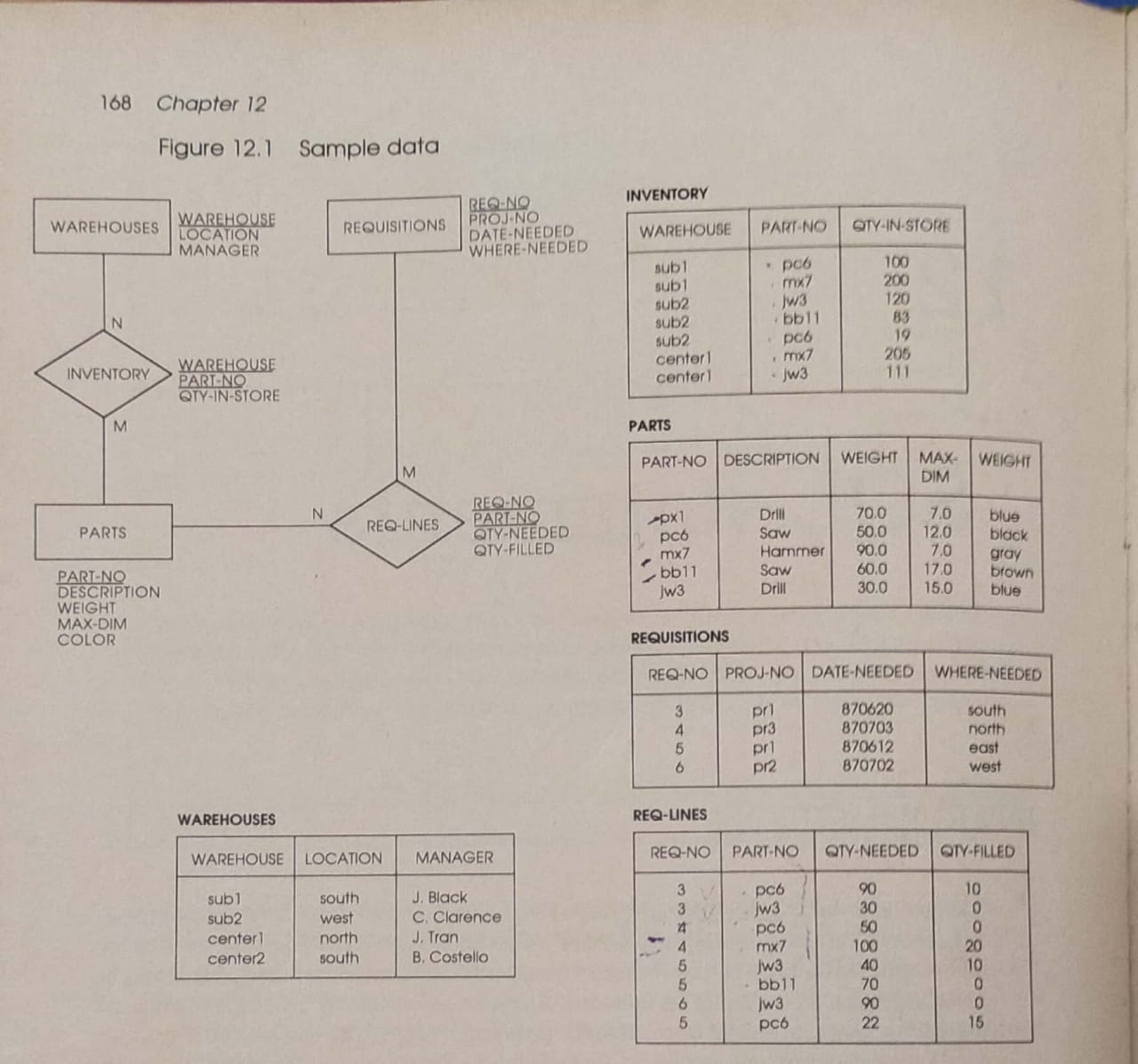
PERNYATAAN SQL

**PENGENALAN**

SQL merupakan bahasa standar digunakan untuk mengakses relasi dalam RDBMS. Dapat digunakan untuk mengambil atau memperbaharui database. SQL dapat digunakan secara interaktif atau ditanamkan dalam bahasa level tinggi. Chapter ini mengenalkan SQL dan menguraikan bagaimana pernyataan SQL digunakan untuk mengambil data dari suatu basisdata. SQL adalah bahasa yang kuat yang dapat digunakan untuk menarik berbagai data dengan satu perintah. SQL melayani permintaan sederhana yang membutuhkan satu baris untuk diambl dari satu relasi. Juga mempunyai pernyataan kuat untuk melakukan query pada hubungan yang komplek antara baris-baris dalam banyak relasi.

Kekuatan ekpresif adalah salah satu isu dalam penggunaan SQL. Isu penting lainnya adalah kinerja. Memungkinkan bahwa beberapa pengambilan mungkin membutuhkan pertimbangan waktu karena mereka mencari secara berutuan pada banyak tabel besar. Dengan demikian beberapa pertimbangan harus diberikan untuk menampilkan ketika merumuskan permintaan.



Chapter ini menggunakan database yang ditunjukan oleh Gambar 12.1 untuk mengilustrasikan perintah SQL. Database ini menyimpan data permintaan bagian-bagian. Data permintaan disimpan dalam relasi REQUSITIONS dan isi data permintaan disimpan dalam tabel REQ-LINES. Data permintaan termasuk jumlah projek yang membuat permintaan, tanggal ketika item permintaan dibutuhkan dan lokasi dimana mereka dibutuhkan. Setiap baris dalam REQ-LINES mewakili sebuh baris dalam permintaan. Kolom REQ-NO mendefinsikan permintaan, dan PART-NO mendefinisikan bagian dalam baris tersebut. Setiap baris permintaan untuk satu bagian dan terdapat banyak baris permintaan dalam satu permintaan. Diskripsi bagian disimpan dalam relasi PARTS. Relasi INVENTORY menyimpan kuantitas setiap bagian dalam Gudang organisasi.

Ide dibalik basis data ini adalah bahwa perminta dibuat dari bagian-bagian. Sebuah pencarian dibuat dari tabel inventory untuk menentukan Gudang yang dapat mengantarkan bagian-bagian (part) ini.

**PERNYATAAN SQL**

Sebuah pernyataan SQL dimasukan kapanpun informasi diambil dari sebuah database. Pernyataan seperti itu dapat digunakan satu kali, atau dapat disimpan dan digunakan kembali kapanpun informasi yang sama diambil secara berulang.

Konstruksi utama digunakan dalam SQL adalah pernyataan SELECT. Bentuk paling dasar dari pernyataan SELECT adalah:

SELECT <kolom-kolom yang akan ditampikan>

FROM <relasi yang memuat kolom tersebut>

WHERE <kondisi yang mesti dipenuhi oleh baris ketika nilai kolom ditampilkan>

Pernyataan SELECT mendefinisikan baris yang diambil dari relasi. Ini juga mendefinisikan nilai kolom untuk ditampilkan pada baris-baris yang diambil. Bagian FROM dari pernyataan pilih mendefinisikan relasi yang baris-barisnya dipilih. Bagian WHERE mendefinisikan kondisi untuk dipenuhi oleh baris terpilih dan bagian SELECT mendifinisikan kolom-kolom yang akan ditampilkan.

Walaupun sederhana dalam ide pernyataan select dapat digunakan untuk merumuskan query secara kuat. Mereka dapat menampilkan sembarang kolom dan menghitung nilai menggunakan fungsi-fungsi yang standar. Mereka juga dapat merujuk pada lebih dari satu relasi, dan menentukan kondisi lintas relasi.

**Mengambil data dari sebuah tabel**

Ide dari SQL adalah untuk menguji sebuah baris tabel pada satu waktu, memilih baris yang memenuhi kondisi WHERE dan mencetak nilai kolom yang ditentukan oleh kondsi SELECT. SQL menyediakan berbagai metoda untuk menentukan pemilih baris oleh kondisi WHERE. Barangkali sederhanya adalah dimana keseluruhan table ada untuk diambil. Ini ditentukan oleh:

SELECT \* FROM PARTS;

Disiini arterisk (\*) menentukan semua kolom menjadi output. Pernyataan FROM menentukan relasi PART yang akan dicari. Ketidakhadiran klausa WHERE berarti bahwa semua baris adalah output. Klausa WHERE harus dimasukan untuk memeilih bagian dari baris dari sebuah tabel. Sebagai contoh, semua baris dengan REQ-NO = 3 dalam relasi REQ-LINES diambil dengan :

SELECT \*

FROM REQ-LINES

WHERE REQ-NO = 3;

Pernyataan tersebut akan menampilkan baris berikut dari REQ-LINES

REQ-NO PART-NO QTY-NEEDED QTY FILLED

3 pe6 90 10

3 jw3 30 0

Jika hanya beberapa kolom dibutuhkan sebagai output, kolom-kolom ini ditentukan dalam klausa SELECT. Sebagai contoh, jika semua yang dibutuhkan adalah daftar bagian dengan PART-NO dan WEIGHT maka pernyataan SQL yang digunakan adalah sebagai berikut:

SELECT PART-N0, WEIGHT

FROM PARTS;

Disini hanya kolom PART-NO dan WEIGHT dari tabel PARTS muncul dalam output. Tentu saja memungkinkan untuk memilih nilai kolom dalam baris yang dipilih dengan menentukan baris dalam klausa WHERE dan kolom dalam SELECT. Dengan demikian pernyataan SQL berikut dapat digunakan untuk menemukan semua PART-NO dan QTY-NEEDED dalam satu permintaan:

SELECT PART-NO, QTY-NEEDED

FROM REQ-LINES

WHERE REQ-NO = 3;

Keluaran dari pernyataan ini adalah

PART-NO QTY-NEEDED

Pc6 90

Jw3 30

Ekspresi aritmatika dapat duga dimasukan dalam klausa SELECT. Sebagai contoh, (QTY-NEEDED – QTY-FILLED) dapat dimasukan dalam klausa SELECT dari pernyataan sebelumnya.

**Pengurutan dan Duplikasi**

Output dari pernyataan sebelumnya adalah dalam urutan yang sama yaitu mereka disimpan dalam tabel. Jika nilai yang sama dari kolom muncul dalam lebih satu baris terpilih, maka nilai-nilai tersebut akan muncul lebih dari satu kali dalam output. SQL memasukan metoda untuk mengontrol urutan outpu dan mengeliminasi duplikat dari output. Urutan ditentukan oleh klausa ORDER sebagai berikut:

SELECT PART-NO, QTY-NEEDED

FROM REQ-LINES

WHERE REQ-NO=3

ORDER BY PART-NO;

Pernyataan ini menghasilkan output yang sama seperti pernyataan sebelumnya tapi dalam urutan yang menaik dari PART-NO. Barisan menurun dapat diperoleh dengan PART-NO dalam urutan klausa ORDER BY dengan DESC.

Duplikasi dihilangkan dengan memasukan DISTINCT dalam klausa SELECT. Sebagai contoh:

SELECT DISTINCT WHERE-NEEDED

FROM REQUISITIONS

WHERE PROJ-NO=’pr1’;

Kata kunci DISCTINCT menentukan nilai atribut tidak boleh duplikasi pada output. Dengan demikian pernyataan ini mengeluarkan lokasi WHERE-NEEDED untuk sebuah projek. Sebuah lokasi hanya akan keluar satu kali jika tidak terdapat lebih dari satu permintaan dalam project ‘pr1’ untuk lokasi tersebut.

Jika terdapat lebih dua kolom dalam klausa SELECT maka DISTINCT diberlakukan pada kombinasi nilai dari kolom-kolom.

**Kondisi Seleksi**

Kondisi seleksi dapat memasukan berbagai kondisi Boolean. Bentuk-bentuk kondisi disediakan memasukan komperasi artimatika seperti lebih besar, kecil dari. Bentuk kondisi juga memasukan operator berikut:

BETWEEN

WHERE WEIGHT BETWEEN 10 AND 17

Memilih baris dimana berat 10 atau lebih dan 17 atau kurang.

LIKE

WHERE DESCRIPTION LIKE ‘h%’

Memilih baris dimana deskripsi dimulai dengan huruf ‘h’. Pencocokan string dapat dinyakan pada awal, akhir atau semberang tempat dalam string. Dengan demikian %h akan menemukan nilai berakhiran dengan ‘h’ dan %h% akan menemukan nilai yang memasukan ‘h’ pada sembarang tempat dalam string.

IN dan NOT IN

WHERE DESCRIPTION IN(‘Saw’,’Drill’)

Mencari semua baris dimana deskripsi adalah Saw atau Drill

WHERE DESCRIPTION NOT IN(‘Saw’)

Mencari semua baris yang deskripsinya selain ‘Saw’. Ingat, SQL menguji satu baris pada satu waktu. Dengan demikian:

SELECT REQ-NO

FROM REQ-LINES

WHERE PART-NO NOT IN(‘pc6’);

Tidak menemukan permintaan yang didak memasukan part ‘pc6’. Itu menemukan permintaan dimana PART-NO adalah sebuah nilai selain ‘pc6’. Sebagai altertif itu menemukan permintaan yang memasukan sebuah garis permintaan tidak untuk ‘pc6’, atau, permintaan tersebut yang memasukan sekurang-kurangnya satu bagian selain dari ‘pc6’.

**Kondisi AND dan OR**

Sering dibutuhkan untuk memlih baris yang memenuhi lebih dari satu kondisi. Sembarang jumlah kondisi yang demikian dapat ditentukan dalam pernyataan SQL, dipisahkan oleh AND dan OR untuk menentukan logika kondisi. Sebagai contoh:

SELECT PART-NO

FROM PARTS

WHERE WEIGHT >55

AND MAX-DIM >10;

Output semua bagian lebih berat dari 55 dan lebih besar dari 10 dalam dimensi maksimum. Contoh lain penggunaan kondisi OR adalah:

SELECT PART-NO

FROM PARTS

WHERE DESCRIPTION =’Saw’

OR DESCRIPTION =’Hammer’;

Pernyataan SQL ini mengambil semua part yaitu saw or hammer. Anda harus mencatat bahwa alternative lain untuk menggunakan klausa IN – WHERE DESCRIPTION IN (‘Saw’,’Hammer’). alternative selanjutnya adalah untuk menggunakan kondisi UNION, yang didiskusikan nanti.