

Belajar Awal Sebuah Query

Oleh Adi Sumaryadi

Membicarakan tentang query, sangat erat hubungannya dengan cara penulisan query tersebut ke dalam sebuah bentuk bahasa yang mudah dimengerti. Pada umumnya, bahasa query yang digunakan untuk mengekspresikan sebuah pernyataan dari query adalah SQL (Structure Query Language).

Sebuah *query* adalah sebuah ekspresi bahasa yang menggambarkan data yang akan didapatkan kembali dari sebuah database. Dalam hubungannya dengan optimisasi query, seringkali diasumsikan bahwa query-query tersebut dinyatakan dalam sebuah dasar-dasar isi dan sekumpulan cara orientasi, yang memberikan optimizer pilihan-pilihan diantara alternatif prosedur-prosedur evaluasi.

Query dapat digunakan pada beberapa keadaan. Kebanyakan aplikasi nyatanya adalah permintaan-permintaan secara langsung dari *user* yang memerlukan informasi tentang bentuk maupun isi dari database. Apabila permintaan user terbatas pada sekumpulan query-query standar, maka query-query tersebut dapat dioptimisasi secara manual oleh pemrograman prosedur-prosedur pencarian gabungan dan membatasi input dari user pada sebuah ukuran menu. Tetapi bagaimanapun juga, sebuah sistem optimisasi query otomatis menjadi penting apabila query-query khusus ditanyakan dengan menggunakan bahasa query yang digunakan secara umum seperti SQL.

Aplikasi yang kedua dari query terjadi pada transaksi-transaksi yang mengubah data yang disimpan berdasarkan nilainya saat itu. Pada akhirnya, query seperti ekspresi-ekspresi dapat digunakan secara internal dalam sebuah DBMS, sebagai contoh adalah untuk mengecek kebenaran akses dan menyamakan kebenaran akses-akses yang terjadi.

Membicarakan tentang query, sangat erat hubungannya dengan cara penulisan query tersebut ke dalam sebuah bentuk bahasa yang mudah dimengerti. Pada umumnya, bahasa query yang digunakan untuk mengekspresikan sebuah pernyataan dari query adalah SQL (Structure Query Language).

SQL adalah sebuah bahasa database yang luas yang memiliki *statement-statement* (pernyataan) untuk definisi data, query dan *update* data (memperbaharui data). SQL mempunyai satu statement dasar untuk mendapatkan kembali informasi dari sebuah database. Statement dasar dari SQL adalah SELECT.

Bentuk dasar dari statement SELECT biasa disebut dengan blok *select from where* yang terbentuk dari tiga macam klausa yaitu SELECT, FROM dan WHERE yang mempunyai bentuk sebagai berikut :

SELECT <daftar Attribute>

FROM <daftar Tabel>

WHERE <kondisi>

Dimana <daftar attribute> adalah sebuah daftar dari nama-nama attribute yang nilai-nilainya didapatkan oleh query. Sedangkan <daftar tabel> adalah sebuah daftar dari nama-nama relasi yang

diperlukan oleh proses sebuah query. <kondisi> adalah sebuah kondisi ekspresi boolean yang mengidentifikasi tuple-tuple yang akan dikembalikan oleh query.

Hubungan Database Dengan Pemrosesan Query

Database adalah kumpulan dari data-data yang berhubungan satu sama lainnya yang digunakan untuk pencarian suatu data tertentu pada saat SQL query dijalankan. Sebuah database dirancang, dibuat dan ditempati oleh data dengan tujuan tertentu. Di dalam sistem database relasional, tabel-tabel dari database saling berhubungan satu sama lainnya. Dan sebuah tabel database akan selalu memiliki *attribute names* (nama-nama attribute), *relation names* (nama-nama relasi), dan *tuples* (record-record).

Attribute digunakan untuk mengidentifikasi sebuah nama yang diikutsertakan dalam relasi dan menspesifikasikan domain (tipe data sederhana yang menentukan sebuah pemisahan data). Sedangkan *tuples* adalah kumpulan dari record-record di dalam sebuah database. *Relation name* mendefinisikan attribute-attribute yang diperlukan dalam predikat dan mendefinisikan arti dari predikat tersebut.

Optimisasi Query

Optimisasi Query adalah suatu proses untuk menganalisa query untuk menentukan sumber-sumber apa saja yang digunakan oleh query tersebut dan apakah penggunaan dari sumber tersebut dapat dikurangi tanpa merubah output. Atau bisa juga dikatakan bahwa optimisasi query adalah sebuah prosedur untuk meningkatkan strategi evaluasi dari suatu query untuk membuat evaluasi tersebut menjadi lebih efektif. Optimisasi query mencakup beberapa teknik seperti transformasi query ke dalam bentuk logika yang sama, memilih jalan akses yang optimal dan mengoptimalkan penyimpanan data.

Optimisasi query merupakan bagian dasar dari sebuah sistem database dan juga merupakan suatu proses untuk menghasilkan rencana akses yang efisien dari sebuah query di dalam sebuah database. Secara tidak langsung, sebuah rencana akses merupakan sebuah strategi yang nantinya akan dijalankan untuk sebuah query, untuk mendapatkan kembali operasi-operasi yang apabila dijalankan akan menghasilkan database record query. Ada tiga aspek dasar yang ditetapkan dan mempengaruhi optimisasi query, yaitu : *search space*, *cost model* dan *search strategy*.

Search space adalah sekumpulan rencana-rencana akses yang sama secara logika yang dapat digunakan untuk mengevaluasi sebuah query. Semua rencana-rencana dalam search space query mengembalikan hasil yang sama biarpun beberapa rencana lebih efisien dibandingkan dengan rencana yang lainnya.

Cost model menandakan sebuah harga untuk tiap rencana dalam search space. Harga dari rencana tersebut adalah sebuah perkiraan dari sumber-sumber yang digunakan pada saat rencana dijalankan, dimana harga yang lebih rendah, merupakan yang terbaik dari rencana-rencana yang ada.

Search strategy adalah sebuah perincian dari rencana-rencana mana dalam search space yang akan diperiksa. Apabila search space-nya kecil, maka strategi yang dapat diteruskan adalah menghitung dan mengevaluasi setiap rencana. Meskipun kebanyakan search space bahkan untuk query-query yang sederhana adalah sangat besar, akan tetapi query optimizer selalu memerlukan aturan

heuristik untuk mengontrol nomer dari rencana-rencana yang akan diperiksa.

Sejarah Singkat Optimisasi Query

Optimisasi query lahir sejak sebelum tahun 1970. Pada saat itu yang dikenal dengan jaman kegelapan (dark age), optimisasi query masih dilakukan secara manual oleh manusia. Selain itu, pada jaman tersebut database relasional juga belum dikenal sehingga diperlukan seseorang yang benar-benar ahli dalam database untuk melakukan optimisasi query. Jadi pada jaman kegelapan hingga tahun 1970-an, optimisasi query masih dilakukan oleh manusia dan masih menggunakan cara yang benar-benar kuno.

Seiring dengan perkembangan jaman di mana teknologi menjadi semakin maju, maka sekitar pertengahan tahun 1970-an sampai pada pertengahan tahun 1980-an optimisasi query tidak lagi dilakukan secara manual, tetapi dilakukan oleh suatu sistem yang dikenal dengan *sistem R* yang kemudian menjadi berkembang pada optimisasi perintah JOIN, sekumpulan tahapan untuk optimisasi query selanjutnya. Hingga saat ini, optimisasi query terus berkembang bersamaan dengan bermunculannya teknik-teknik baru yang digunakan dalam proses optimisasi query meskipun pada dasarnya hanya ada dua macam teknik utama yang biasanya digunakan dalam mengoptimisasi sebuah query. Bisa dikatakan bahwa optimisasi query merupakan tulang punggung dari sebuah sistem karena ketepatangunaan suatu sistem sangat tergantung dari optimizernya.

Hampir semua DBMS menggunakan sistem optimisasi query untuk mengurangi waktu eksekusi query sehingga kerja sistem dapat dioptimalkan dan penggunaan sumber-sumber dapat diminimumkan. Selain itu, akses dari disk yang sangat lambat dibandingkan dengan akses memory membuat optimisasi query menjadi semakin penting.

Tujuan Optimisasi Query

Prinsip ekonomi yang diperlukan untuk sebuah query adalah mengoptimisasi prosedur-prosedur, mencoba untuk memaksimumkan output dari sejumlah sumber-sumber yang diberikan ataupun untuk meminimumkan penggunaan sumber untuk memberikan output.

Tujuan dari optimisasi query adalah berbeda-beda untuk setiap sistem. Ada yang menggunakan optimisasi query untuk meminimumkan waktu proses sedangkan pada situasi lain bisa juga optimisasi query diperlukan untuk waktu respon, meminimumkan I/O dan meminimumkan penggunaan memory. Tetapi pada dasarnya, tujuan dari optimisasi query adalah menemukan jalan akses yang termurah untuk meminimumkan total waktu pada saat proses sebuah query. Untuk mencapai tujuan tersebut, maka diperlukan *optimizer* untuk melakukan analisa query dan untuk melakukan pencarian jalan akses.

Kata Kunci :