

Operasi Matriks Menggunakan SQL pada MySQL

Oleh : Yusup
Fakultas Ilmu Komputer, Universitas AKI Semarang

Abstract

Matrix algebra operations can be performed by using the SQL SELECT statement in particular. To practice SQL commands using the MySQL software.

Keywords : SQL, Matrices, Matrix Operations, MySQL.

Pendahuluan

Matriks adalah suatu himpunan bilangan yang berbentuk persegi panjang, atau Matriks adalah himpunan skalar (bilangan riil atau bilangan kompleks) yang disusun secara empat persegi panjang menurut baris dan kolom atau suatu matriks adalah himpunan unsur-unsur yang disusun menurut baris dan kolom, sehingga berbentuk empat persegi panjang, dimana panjangnya dan lebarnya ditunjukkan oleh banyaknya kolom-kolom dan baris-baris.

Notasi matriks biasanya menggunakan huruf besar A, B, C Misalkan A adalah suatu matriks yang terdiri dari m buah baris dan n buah kolom, maka matriks A mempunyai ordo/dimensi/ukuran (mxn) dan a_{ij} merupakan elemen-elemen/unsur-unsur pada baris ke-i dan kolom ke-j dari matriks A maka secara lengkap sebuah matriks dapat ditulis dengan

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m1} & \cdots & a_{mn} \end{bmatrix}$$

dimana a = elemen matriks, i = nomor baris = 1,2,3,..., m, j = nomor kolom = 1,2,3,..., n

Dua matriks (matriks $A = [a_{ij}]$ dan matriks $B = [b_{ij}]$) dikatakan **sama** ($A = B$) jika kedua matriks tersebut mempunyai ukuran (dimensi/ordo) yang sama (mxn) dan elemen-elemen yang bersangkutan (satu letak) di dalam kedua matriks tersebut sama ($a_{ij} = b_{ij}$) untuk setiap $i = 1,2,...,m$ dan $j = 1,2,...,n$.

Operasi Aljabar Matriks

a. Penjumlahan dan pengurangan matriks

Syaratnya adalah matriks yang akan dijumlahkan/dikurangkan harus mempunyai **ordo yang sama**.

Misalkan $A = [a_{ij}]$, $B = [b_{ij}]$, $C = [c_{ij}]$ maka $A \pm B = C \Leftrightarrow [a_{ij}] \pm [b_{ij}] = [c_{ij}]$

a. **Data Definition Language (DDL) :**

Digunakan untuk mendefinisikan data (Database, Tabel, View, Index) dengan menggunakan perintah-perintah: *CREATE, ALTER, DROP*.

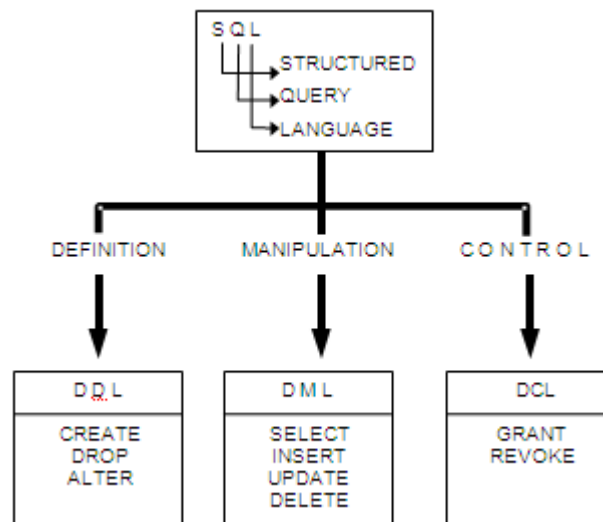
b. **Data Manipulation Language (DML) :**

Digunakan untuk memanipulasi data dengan menggunakan perintah : *SELECT, UPDATE, INSERT, DELETE*. Data Manipulation Language merupakan bagian terpadu bahasa SQL. Perintah-perintahnya dapat dibuat secara

interaktif atau ditempelkan pada sebuah program aplikasi. Pemakai hanya perlu menentukan 'APA' yang ia inginkan, DBMS menentukan 'BAGAIMANA' cara mendapatkannya.

c. **Data Control Language (DCL) :**

Digunakan untuk mengontrol hak para pemakai data dengan perintah : *GRANT, REVOKE*.

**Data Definition Language (DDL)**

DDL merupakan salah satu bahasa SQL yang digunakan untuk membangun objek-objek dalam database seperti pembuatan database, pembuatan tabel, perubahan stuktur tabel, serta penghapusan database (tabel, view)..

1. Perintah Create

Perintah create digunakan untuk membuat database, Table, Index, View.

Membuat Database

Bentuk Umum : **CREATE DATABASE**
<nama database>;

Membuat Tabel

Bentuk Umum perintah membuat tabel:

```
CREATE TABLE (Nama_Tabel) (
(Nama_field-1) (tipe
```

```
data(ukuran)) [UNSIGNED] [NOT
NULL] [PRIMARY KEY] [DEFAULT
nilai_awal] [AUTO_INCREMENT],
>Nama_field-2) (tipe
data(ukuran)) [UNSIGNED] [NOT
NULL] [DEFAULT nilai_awal], . .
., (Nama_field-n) (tipe
data(ukuran)) [UNSIGNED] [NOT
NULL] [DEFAULT nilai_awal]);
```

2. Perintah Alter

Perintah Alter digunakan untuk memperbaharui (mengubah, menambah, menghapus) field/ kolom (struktur tabel) atau mengubah nama tabel. Ada empat macam perubahan dalam struktur tabel, yaitu :

a. Perubahan nama field/kolom

Bentuk Umum : ALTER TABLE (*nama tabel*) **CHANGE** (*nama field lama*) (*nama field baru*) (*tipe data*);

b. Perubahan tipe data

Bentuk Umum : ALTER TABLE (*nama tabel*) **MODIFY** (*nama field*) (*tipe data baru*);

Atau

```
ALTER TABLE (nama tabel)
CHANGE (nama field lama) (nama field baru) (tipe data);
```

Dengan memberi nama field baru sama dengan nama field lama.

c. Penambahan field

Bentuk Umum :

```
ALTER TABLE (nama tabel) ADD
(nama field) (tipe data)
```

```
[FIRST]/[AFTER nama field
sebelumnya];
```

d. Penghapusan field

Bentuk Umum : ALTER TABLE (*nama tabel*) **DROP COLUMN** (*nama field*);

3. Perintah DROP

Perintah drop digunakan untuk menghapus database dan menghapus tabel. Jadi perintah DROP merupakan kebalikan dari perintah CREATE.

a). Menghapus Tabel

Bentuk Umum : DROP TABLE (*nama tabel*);

b). Menghapus Database

Bentuk Umum : DROP DATABASE (*Nama database*);

Data Manipulation Language (DML)

DML merupakan salah satu bahasa SQL yang digunakan untuk membangun objek-objek dalam database seperti menampilkan data / baris / record, memasukkan / menambah data / baris / record, memperbaiki data / baris / record serta menghapus data / baris / record.

a). Memasukkan Data (*Insert*)

Sebelum memasukkan data ke dalam suatu tabel, terdapat beberapa hal yang harus diperhatikan, yaitu:

1. Data-data yang akan dimasukkan ke dalam suatu tabel harus sesuai (*match*) dengan struktur tabelnya (ERROR 1136: Column count doesn't match value count at row 1). Sehingga sebelum memasukkan data ke dalam suatu tabel akan lebih baik periksa dulu struktur dari suatu tabel yang akan diisi data dengan menggunakan perintah DESC atau SHOW COLUMN atau SHOW FIELDS.
2. Apabila data yang akan dimasukkan bertipe selain numerik harus diapit oleh tanda petik (' ');

Terdapat dua cara yang dapat digunakan untuk memasukkan data ke dalam tabel, yaitu:

1.

```
INSERT INTO (nama_tabel)
values (isi_field_2, ...,
isi_field_n);
```
2.

```
INSERT INTO (nama_tabel)
(nama_field_1, nama_field_2,
..., nama_field_n) VALUES
(isi_field_1, isi_field_2, ...,
isi_field_n);
```

b). Mengubah Data (*Update*)

Jika data yang tersimpan dalam suatu tabel dirasa sudah tidak relevan atau sudah kadaluarsa atau terjadi kesalahan dalam memasukkan data, maka data tersebut dapat diperbaharui (*di-update*). Adapun perintah untuk memperbaharui data tersebut adalah sebagai berikut:

```
UPDATE nama_tabel SET
nama_field_1 = isi_baru_1,
nama_field_2= isi_baru_2,...,
nama_field_n = isi_baru_n
WHERE kriteria atau kondisi;
```

c). Menghapus Data (*Delete*)

Jika terdapat data yang salah atau data sudah tidak digunakan lagi, maka data dapat dihapus dengan perintah *DELETE*. Dalam menghapus data harus hati-hati, karena kalau salah menentukan syarat atau lupa menambahkan syarat, data akan terhapus secara langsung (tidak ada konfirmasi). Secara sederhana perintah *DELETE* adalah sebagai berikut:

```
DELETE FROM (Nama tabel) WHERE
kriteria;
```

d). Menampilkan Data (*Select*)

Perintah *SELECT* digunakan untuk menampilkan data di dalam suatu tabel, perintah ini mempunyai cakupan yang luas.

Secara umum struktur penulisan perintah SELECT yang sering digunakan untuk menampilkan suatu data adalah sebagai berikut:

```
SELECT      [fields]      FROM
[nama_tabel] WHERE [kondisi]
ORDER BY [nama_field] ASC / DESC
GROUP BY [nama_field]
```

Perintah SELECT yang paling minimal (tidak error secara penulisan) jika terdiri dari:

```
SELECT (Nama Field) FROM (Nama
Tabel);
```

MySQL

MySQL (bisa dibaca dengan **mai-es-ki-el** atau bisa juga **mai-se-kuel**) adalah suatu perangkat lunak database relasi atau *Relational Database Management System* (RDBMS) yang didistribusikan gratis dibawah lisensi **GPL** (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam

database yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukkan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Matriks dengan SQL

Sebuah matriks dapat didefinisikan dengan SQL dengan menyatakannya dalam sebuah array. Diketahui sebuah matriks $A =$

$$\begin{bmatrix} 2 & -3 & 4 \\ -1 & 0 & 2 \end{bmatrix}.$$

Matriks tersebut dapat dinyatakan dengan statement SQL dalam MySQL sebagai berikut:

```
mysql> select * from MatriksA;
+---+---+---+
| i | k | NilElemen |
+---+---+---+
| 1 | 1 |          2 |
| 1 | 2 |         -3 |
| 1 | 3 |          4 |
| 2 | 1 |         -1 |
| 2 | 2 |          0 |
| 2 | 3 |          2 |
+---+---+---+
```

Adapun struktur dari matriks A tersebut adalah sebagai berikut:

```
+---+---+---+---+---+---+
| Field      | Type      | Null | Key | Default | Extra |
+---+---+---+---+---+---+
| i          | int(11)   | NO   | PRI | NULL    |       |
| k          | int(11)   | NO   | PRI | NULL    |       |
| NilElemen  | int(11)   | NO   |     | NULL    |       |
+---+---+---+---+---+---+
```

Untuk mendefinisikan dan memasukkan data dari matriks tersebut adalah dengan menggunakan statement SQL berikut:

```
mysql> create table MatriksA (i
Integer Not Null Check (i
between 1 and 10), k Integer
Not Null Check (k between 1 and
10), Nilelemen Integer Not
Null, Primary Key(i,k));

mysql> insert into MatriksA
Values (1,1,2), (1,2,-3),
(1,3,4), (2,1,-1), (2,2,0),
(2,3,2);
```

Operasi Matriks dengan SQL dalam MySQL

1. Kesamaan Matriks

Diketahui Matriks A =

$$\begin{bmatrix} 2 & -3 & 4 \\ -1 & 0 & 2 \end{bmatrix}$$
 dan A1 =

$$\begin{bmatrix} -3 & 4 & 2 \\ -1 & 2 & -3 \end{bmatrix},$$

 dan A2 =

$$\begin{bmatrix} 2 & -3 & 4 \\ -1 & 0 & 2 \end{bmatrix}$$

Maka kesamaan matriks A dengan Matriks A2 dapat dinyatakan dengan perintah SQL sebagai berikut:

```
mysql> select count(*) from
matriksA union
-> select count(*) from
matriksA2 union
-> select count(*) from
matriksA A,matriksA2 A2
-> where a.i=a2.i
-> and a.k=a2.k
-> and a.nilelemen =
a2.nilelemen;
```

count(*)
6

Hasilnya hanya satu nilai = 6 yaitu jumlah elemen dari matriks tersebut. Sedangkan jika matriks tersebut tidak sama, maka hasilnya adalah sebagai berikut:

```
mysql> select count(*) from
matriksA union select
count(*) from matriksA1 union
select count(*) from matriksA
A,matriksA1 A1 where
a.i=a1.i and a.k=a1.k and
a.nilelemen=a1.nilelemen;
```

count(*)
6
1

2. Penjumlahan dan Pengurangan Matriks

Untuk menjumlahkan matriks A dengan Matriks A1 dapat dilakukan dengan menggunakan perintah SQL sebagai berikut:

```
mysql> select
a.i,a.k,(a.nilelemen +
a1.nilelemen) JmlMat from
matriksA A,matriksA1 A1
where a.i=a1.i and a.k =
a1.k;
```

i	k	JmlMat
1	1	-1
1	2	1
1	3	6
2	1	-2
2	2	2
2	3	-1

Atau dapat juga dengan menggunakan perintah sebagai berikut:

```
mysql> select A.i,A.k,
(A.nilElemen+A1.nilElemen)
JmlMat from matriksA
A,matriksA1 A1 where A.i=A1.i
and A.k = A1.k and ((Select
count(*) from MatriksA) =
(Select Count(*) from
MatriksA1) and (Select Max(i)
from MatriksA) = (Select
Max(i) from MatriksA1) and
(Select Max(k) from MatriksA)
= (Select Max(k) from
MatriksA1));
```

i	k	JmlMat
1	1	-1
1	2	1
1	3	6
2	1	-2
2	2	2
2	3	-1

Sedangkan untuk mengurangi matriks tersebut caranya sama hanya tinggal mengganti tanda + menjadi -.

3. Perkalian Matriks dengan Skalar

Perkalian matriks A dengan skalar 2 dapat dilakukan dengan menggunakan perintah SQL sebagai berikut:

```
mysql> select i, k,
2*nilElemen from MatriksA2;
```

i	k	2*nilElemen
1	1	4
1	2	-6
1	3	8
2	1	-2
2	2	0
2	3	4

4. Perkalian Matriks dengan Matriks

Perkalian matriks $A = \begin{bmatrix} 2 & -3 & 4 \\ -1 & 0 & 2 \end{bmatrix}$

dengan matriks $B = \begin{bmatrix} -1 & 2 & 3 \\ 0 & 1 & 7 \\ 1 & 1 & -2 \end{bmatrix}$,

dapat dilakukan dengan menggunakan perintah SQL sebagai berikut:

```
mysql> Select i, j,
Sum(MatriksA.NilElemen *
MatriksB.NilElemen) as
KaliElemen From MatriksA,
MatriksB where
matriksA.k=MatriksB.k
group by i,j;
```

i	j	KaliElemen
1	1	2
1	2	5
1	3	-23
2	1	3
2	2	0
2	3	-7

5. Transpose Matriks

Transpose dari matriks A dapat diperoleh dengan perintah SQL sebagai berikut:

```
mysql> select k, i,
nilElemen from matriksA;
```

k	i	nilElemen
1	1	2
2	1	-3
3	1	4
1	2	-1
2	2	0
3	2	2

Kesimpulan

Operasi-operasi dalam matriks khususnya operasi aljabar diantaranya *kesamaan dua matriks, penjumlahan dan pengurangan matriks, perkalian matriks dengan skalar, perkalian matriks dengan matriks, dan transpose matriks* dapat dilakukan dengan menggunakan perintah-perintah SQL khususnya perintah SELECT. Dalam mempraktekkan perintah-perintah SQL tersebut menggunakan Software MySQL.

Daftar Pustaka

D. Suryadi HS, S. Harini M, *Teori dan Soal Pendahuluan Aljabar Linier*, PT. Ghalia Indonesia, Jakarta, 1984.

Howard Anton, *Aljabar Linier Elementer*, edisi kelima, Penerbit Erlangga, Jakarta, 1991.

Joe Celko's, *SQL for Smarties Advanced SQL Programming, Fourth Edition*, Morgan Kaufmann Publishers, USA, 2011.

MySQL 5.1 Reference Manual, MySQL AB, 2009, Sun Microsystems, Inc.

Raghu Ramakrishnan, Johannes Gehrke, *Sistem Manajemen Database*, Edisi 3, McGraw Hill Companies Inc. 2003, Diterjemahkan Tim Penerbit Andi, Yogyakarta.