

MySQL

Mengurutkan hasil Query

Oleh :

Janner Simarmata

sijanner@yahoo.com

<http://simarmata.cogia.net>

*Dipublikasikan dan didedikasikan
untuk perkembangan pendidikan di Indonesia melalui*

MateriKuliah.Com

Lisensi Pemakaian Artikel:

*Seluruh artikel di **MateriKuliah.Com** dapat digunakan, dimodifikasi dan disebarakan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut Penulis. Hak Atas Kekayaan Intelektual setiap artikel di **MateriKuliah.Com** adalah milik Penulis masing-masing, dan mereka bersedia membagikan karya mereka semata-mata untuk perkembangan pendidikan di Indonesia. **MateriKuliah.Com** sangat berterima kasih untuk setiap artikel yang sudah Penulis kirimkan.*

1.1. Pendahuluan

Bab ini meliputi sorting (*pengurutan*), suatu operasi yang sangat penting untuk mengontrol bagaimana MySQL menampilkan hasil dari statement SELECT. Sorting dilakukan dengan menambahkan klausa ORDER BY pada sebuah query. Tanpa klausa semacam itu, MySQL akan menampilkan hasil dengan sembarang urutan (*Sorting juga bisa dilakuka dengan menggunakan klausa GROUP BY*).

Berikut ini adalah dua tabel yang digunakan dalam contoh :

```
mysql> SELECT * FROM driver_log;
```

rec_id	name	trav_date	miles
1	Ben	2001-11-30	152
2	Suzi	2001-11-29	391
3	Henry	2001-11-29	300
4	Henry	2001-11-27	96
5	Ben	2001-11-29	131
6	Henry	2001-11-26	115
7	Suzi	2001-12-02	502
8	Henry	2001-12-01	197
9	Ben	2001-12-02	79
10	Henry	2001-11-30	203

```
mysql> SELECT * FROM mail;
```

t	srcuser	srchost	dstuser	dsthost	size
2001-05-11 10:15:08	barb	saturn	tricia	mars	58274
2001-05-12 12:48:13	tricia	mars	gene	venus	194925
2001-05-12 15:02:49	phil	mars	phil	saturn	1048
2001-05-13 13:59:18	barb	saturn	tricia	venus	271
2001-05-14 09:31:37	gene	venus	barb	mars	2291
2001-05-14 11:52:17	phil	mars	tricia	saturn	5781
2001-05-14 14:42:21	barb	venus	barb	venus	98151
2001-05-14 17:03:01	tricia	saturn	phil	venus	2394482
2001-05-15 07:17:48	gene	mars	gene	saturn	3824
2001-05-15 08:50:57	phil	venus	phil	venus	978
2001-05-15 10:25:52	gene	mars	tricia	saturn	998532
2001-05-15 17:35:31	gene	saturn	gene	mars	3856
2001-05-16 09:00:28	gene	venus	barb	mars	613
2001-05-16 23:04:19	phil	venus	barb	venus	10294
2001-05-17 12:49:23	phil	mars	tricia	saturn	873
2001-05-19 22:21:51	gene	saturn	gene	venus	23992

1.2. Menggunakan ORDER BY untuk Mengurutkan Hasil Query Penyelesaian

Tambahkan klausa ORDER BY pada query.

Pembahasan

Ketika anda memilih record, mereka akan dikeluarkan dari database dan dikembalikan dalam urutan apapun yang telah digunakan oleh server. Untuk mengatur baris dari hasil query dalam urutan tertentu, urutkanlah dengan menambah klausa ORDER BY ke statement SELECT. Tanpa

ORDER BY, urutan hasil query akan berubah ketika isi tabel dirubah. Dengan klausa ORDER BY, MySQL akan selalu mengurutkan baris seperti yang diinginkan.

ORDER BY mempunyai sifat-sifat umum sebagai berikut :

- Anda dapat mengurutkan menggunakan satu maupun banyak nilai kolom.
- Anda dapat mengurutkan sembarang kolom baik dalam urutan menaik (*ascending*, *default*) atau menurun (*descending*)
- Anda dapat mengacu ke kolom-kolom yang diurutkan berdasarkan nama, letak dalam daftar kolom, atau menggunakan alias

1.3 Penamaan Kolom yang Diurutkan dan Menentukan Arah Pengurutan

Berikut ini adalah contoh yang menunjukkan bagaimana mengurutkan satu kolom atau banyak kolom dan bagaimana mengurutkan dalam urutan menaik atau menurun.

Query berikut ini menghasilkan urutan berdasarkan satu kolom menggunakan driver nama :

```
mysql> SELECT * FROM driver_log ORDER BY name;
```

rec_id	name	trav_date	miles
1	Ben	2001-11-30	152
5	Ben	2001-11-29	131
9	Ben	2001-12-02	79
3	Henry	2001-11-29	300
4	Henry	2001-11-27	96
6	Henry	2001-11-26	115
8	Henry	2001-12-01	197
10	Henry	2001-11-30	203
2	Suzi	2001-11-29	391
7	Suzi	2001-12-02	502

Urutan secara default adalah menaik. Anda dapat membuat urutan menaik secara eksplisit dengan menambah ASC setelah nama kolom yang akan diurutkan :

```
SELECT * FROM driver_log ORDER BY name ASC;
```

Pengurutan sebaliknya yaitu urutan menurun, dapat dilakukan dengan menambahkan DESC setelah nama kolom yang akan diurutkan :

```
mysql> SELECT * FROM driver_log ORDER BY name DESC;
```

rec_id	name	trav_date	miles
2	Suzi	2001-11-29	391
7	Suzi	2001-12-02	502
3	Henry	2001-11-29	300
4	Henry	2001-11-27	96
6	Henry	2001-11-26	115
8	Henry	2001-12-01	197
10	Henry	2001-11-30	203
1	Ben	2001-11-30	152
5	Ben	2001-11-29	131
9	Ben	2001-12-02	79

Kalau diperhatikan keluaran dari query, baris untuk nama tertentu tidak diurutkan. Ini karena MySQL tidak mengurutkan sesuatu yang tidak disebutkan secara eksplisit :

- Urutan secara keseluruhan yang dikembalikan oleh query tidak ditentukan kecuali anda menggunakan klausa ORDER BY.
- Dengan cara yang sama, dalam kelompok baris yang diurutkan berdasarkan pada nilai pada suatu kolom, urutan nilai pada kolom lain juga tidak akan ditentukan kecuali anda menyebutkan nama kolom tersebut dalam klausa ORDER BY.

Untuk dapat melakukan pengurutan dengan banyak kolom, gunakan koma. Query berikut ini mengurutkan dalam urutan menaik berdasarkan nama dan trav_date dalam baris setiap nama tertentu :

```
mysql> SELECT * FROM driver_log ORDER BY name, trav_date;
```

rec_id	name	trav_date	miles
5	Ben	2001-11-29	131
1	Ben	2001-11-30	152
9	Ben	2001-12-02	79
6	Henry	2001-11-26	115
4	Henry	2001-11-27	96
3	Henry	2001-11-29	300
10	Henry	2001-11-30	203
8	Henry	2001-12-01	197
2	Suzi	2001-11-29	391
7	Suzi	2001-12-02	502

Dengan banyak kolom juga dapat mengurutkan secara menurun, tapi DESC harus ditulis setelah setiap nama kolom :

```
mysql> SELECT * FROM driver_log ORDER BY name DESC, trav_date DESC;
```

rec_id	name	trav_date	miles
7	Suzi	2001-12-02	502
2	Suzi	2001-11-29	391
8	Henry	2001-12-01	197
10	Henry	2001-11-30	203
3	Henry	2001-11-29	300
4	Henry	2001-11-27	96
6	Henry	2001-11-26	115
9	Ben	2001-12-02	79
1	Ben	2001-11-30	152
5	Ben	2001-11-29	131

Pengurutan dengan banyak kolom dapat melakukan pengurutan campuran dimana beberapa kolom diurutkan menaik dan lainnya diurutkan secara menurun. Query berikut mengurutkan berdasarkan nama secara menurun, kemudian berdasarkan trav_date secara menaik untuk masing-masing nama :

```
mysql> SELECT * FROM driver_log ORDER BY name DESC, trav_date;
```

rec_id	name	trav_date	miles
2	Suzi	2001-11-29	391
7	Suzi	2001-12-02	502
6	Henry	2001-11-26	115

4	Henry	2001-11-27	96
3	Henry	2001-11-29	300
10	Henry	2001-11-30	203
8	Henry	2001-12-01	197
5	Ben	2001-11-29	131
1	Ben	2001-11-30	152
9	Ben	2001-12-02	79

1.4 Cara untuk Mengacu pada Kolom yang Diurutkan

Klausa `ORDER BY` dalam query yang ditunjukkan mengacu pada kolom yang diurutkan berdasarkan nama. Anda dapat juga menyebutkan kolom-kolom berdasarkan posisinya dalam daftar kolom keluaran atau menggunakan alias. Posisi pada daftar output dimulai dari 1. Query berikut menghasilkan kolom output ketiga yaitu miles :

```
mysql> SELECT name, trav_date, miles FROM driver_log ORDER BY 3;
```

name	trav_date	miles
Ben	2001-12-02	79
Henry	2001-11-27	96
Henry	2001-11-26	115
Ben	2001-11-29	131
Ben	2001-11-30	152
Henry	2001-12-01	197
Henry	2001-11-30	203
Henry	2001-11-29	300
Suzi	2001-11-29	391
Suzi	2001-12-02	502

Jika kolom output mempunyai alias, anda dapat mengacu ke alias pada klausa `ORDER BY` :

```
mysql> SELECT name, trav_date, miles AS distance FROM driver_log
-> ORDER BY distance;
```

name	trav_date	distance
Ben	2001-12-02	79
Henry	2001-11-27	96
Henry	2001-11-26	115
Ben	2001-11-29	131
Ben	2001-11-30	152
Henry	2001-12-01	197
Henry	2001-11-30	203
Henry	2001-11-29	300
Suzi	2001-11-29	391
Suzi	2001-12-02	502

Kolom yang ditentukan berdasarkan posisi atau alias dapat diurutkan baik secara menaik atau menurun seperti kolom bernama :

```
mysql> SELECT name, trav_date, miles FROM driver_log ORDER BY 3 DESC;
```

name	trav_date	miles
Suzi	2001-12-02	502
Suzi	2001-11-29	391
Henry	2001-11-29	300

Henry	2001-11-30	203
Henry	2001-12-01	197
Ben	2001-11-30	152
Ben	2001-11-29	131
Henry	2001-11-26	115
Henry	2001-11-27	96
Ben	2001-12-02	79

1.5 Mengurutkan Bagian dari Tabel

Penyelesaian

Tambahkan klausa WHERE yang hanya memilih rekord yang anda ingin lihat.

Pembahasan

ORDER BY tidak peduli berapa banyak baris yang ada. Dia akan mengurutkan baris apapun yang dikeluarkan oleh query. Jika anda tidak ingin menampilkan seluruh tabel, tambahkan klausa WHERE untuk menyatakan baris yang mana yang anda pilih. Sebagai contoh, untuk mengurutkan rekord untuk hanya satu sopir, dapat digunakan query berikut ini :

```
mysql> SELECT trav_date, miles FROM driver_log WHERE name = 'Henry'
-> ORDER BY trav_date;
+-----+-----+
| trav_date | miles |
+-----+-----+
| 2001-11-26 | 115   |
| 2001-11-27 | 96    |
| 2001-11-29 | 300   |
| 2001-11-30 | 203   |
| 2001-12-01 | 197   |
+-----+-----+
```

Kolom bernama pada klausa ORDER BY tidak perlu sama dengan pada klausa WHERE.

1.6 Mengurutkan Hasil dari Ekspresi

Penyelesaian

Letakkan ekspresi yang menghitung nilai pada klausa ORDER BY.

Pembahasan

Berikut ini adalah tabel mail yang salah satu kolomnya menunjukkan ukuran dari surat.

```
mysql> SELECT * FROM mail;
+-----+-----+-----+-----+-----+-----+
| t          | srcuser | srchost | dstuser | dsthost | size  |
+-----+-----+-----+-----+-----+-----+
| 2001-05-11 10:15:08 | barb   | saturn  | tricia  | mars    | 58274 |
| 2001-05-12 12:48:13 | tricia | mars    | gene    | venus   | 194925 |
| 2001-05-12 15:02:49 | phil   | mars    | phil    | saturn  | 1048   |
| 2001-05-13 13:59:18 | barb   | saturn  | tricia  | venus   | 271    |
...

```

Untuk mengurutkan rekord berdasarkan ukuran surat dalam kilobyte, nilai yang akan diurutkan dihitung dengan menggunakan ekspresi. Anda dapat menggunakan ORDER BY untuk mengurutkan hasil dari ekspresi. Untuk melakukan ini, tuliskan ekspresi pada daftar kolom output dan mengacu baik berdasarkan posisi atau memberi alias dan mengacu pada alias tersebut :

```
mysql> SELECT t, srcuser, FLOOR((size+1023)/1024)
-> FROM mail WHERE size > 50000 -> ORDER BY 3;
```

t	srcuser	FLOOR((size+1023)/1024)
2001-05-11 10:15:08	barb	57
2001-05-14 14:42:21	barb	96
2001-05-12 12:48:13	tricia	191
2001-05-15 10:25:52	gene	976
2001-05-14 17:03:01	tricia	2339

```
mysql> SELECT t, srcuser, FLOOR((size+1023)/1024) AS kilobytes
-> FROM mail WHERE size > 50000 -> ORDER BY kilobytes;
```

t	srcuser	kilobytes
2001-05-11 10:15:08	barb	57
2001-05-14 14:42:21	barb	96
2001-05-12 12:48:13	tricia	191
2001-05-15 10:25:52	gene	976
2001-05-14 17:03:01	tricia	2339

```
mysql> SELECT t, srcuser, FLOOR((size+1023)/1024)
-> FROM mail WHERE size > 50000 -> ORDER BY FLOOR((size+1023)/1024);
```

t	srcuser	FLOOR((size+1023)/1024)
2001-05-11 10:15:08	barb	57
2001-05-14 14:42:21	barb	96
2001-05-12 12:48:13	tricia	191
2001-05-15 10:25:52	gene	976
2001-05-14 17:03:01	tricia	2339

Ada dua alasan mengapa sebaiknya menggunakan alias dari pada menuliskan ekspresi :

- Lebih mudah untuk menulis klausa ORDER BY menggunakan alias dari pada mengulangi ekspresi.
- Alias bisa berguna untuk menampilkan hasil, untuk menyediakan label kolom yang lebih berarti.

1.7 Menampilkan Satu Kumpulan Nilai Sembari Mengurutkan yang Lain

Penyelesaian

Anda dapat menggunakan kolom pada klausa ORDER BY yang tidak tampak dalam daftar output kolom.

Pembahasan

ORDER BY tidak terbatas pada pengurutan hanya kolom-kolom bernama pada daftar output kolom. ORDER BY juga dapat mengurutkan nilai yang tersembunyi. Cara ini biasanya digunakan ketika anda mempunyai nilai yang dapat dinyatakan dengan cara lain dan anda ingin menampilkan satu tipe nilai tapi mengurutkan berdasarkan yang lain. Sebagai contoh, anda ingin menampilkan ukuran surat tidak dalam byte tapi dalam K atau kilobytes. Anda dapat merubahnya dengan menggunakan ekspresi :

```
CONCAT(FLOOR((size+1023)/1024),'K')
```

Tapi nilainya menjadi string, sehingga tidak bisa diurutkan secara numeris :

```
mysql> SELECT t, srcuser, CONCAT(FLOOR((size+1023)/1024),'K') AS size_in_K  
-> FROM mail WHERE size > 50000 ORDER BY size_in_K;
```

t	srcuser	size_in_K
2001-05-12 12:48:13	tricia	191K
2001-05-14 17:03:01	tricia	2339K
2001-05-11 10:15:08	barb	57K
2001-05-14 14:42:21	barb	96K
2001-05-15 10:25:52	gene	976K

Untuk mendapatkan urutan yang diinginkan, tampilkan stringnya, tapi gunakan ukuran numeris sebenarnya untuk mengurutkan :

```
mysql> SELECT t, srcuser, CONCAT(FLOOR((size+1023)/1024),'K') AS size_in_K  
-> FROM mail WHERE size > 50000 ORDER BY size;
```

t	srcuser	size_in_K
2001-05-11 10:15:08	barb	57K
2001-05-14 14:42:21	barb	96K
2001-05-12 12:48:13	tricia	191K
2001-05-15 10:25:52	gene	976K
2001-05-14 17:03:01	tricia	2339K

Menampilkan nilai sebagai string tapi mengurutkan sebagai angka juga bisa menimbulkan masalah. Misalnya nomor punggung pemain bisa 0 atau 00. Keduanya akan dibaca sama yaitu 0. Untuk mengatasi masalah ini, dapat dilakukan dengan menyimpan nomor punggung sebagai string :

```
CREATE TABLE roster  
(  
  name CHAR(30), # player name  
  jersey_num CHAR(3) # jersey number  
);
```

Kemudian nomor punggung akan menampilkan apapun yang anda masukkan, dan 0 dan 00 akan diperlakukan sebagai nilai yang berbeda. Tapi ini juga masih menimbulkan masalah. Misalkan sebuah tim terdiri dari pemain sebagai berikut :

```
mysql> SELECT name, jersey_num FROM roster;
```

name	jersey_num
Lynne	29
Ella	0
Elizabeth	100
Nancy	00
Jean	8
Sherry	47

Masalah timbul ketika anda mencoba untuk mengurutkan anggota tim berdasarkan nomor punggung. Jika nomor -nomor itu disimpan sebagai string, mereka akan diurutkan secara leksikal.


```
mysql> SELECT name, jersey_num FROM roster ORDER BY jersey_num;
```

name	jersey_num
Ella	0
Nancy	00
Elizabeth	100
Lynne	29
Sherry	47
Jean	8

Nilai 100 dan 8 bertukar tempat. Tapi hal ini dapat diselesaikan dengan mudah. Tampilkan nilai string, tapi gunakan nilai numeric untuk pengurutan. Untuk melakukan ini, tambahkan nol ke nilai jersey_num untuk memaksa konversi dari string ke angka :

```
mysql> SELECT name, jersey_num FROM roster ORDER BY jersey_num+0;
```

name	jersey_num
Ella	0
Nancy	00
Jean	8
Lynne	29
Sherry	47
Elizabeth	100

Cara untuk menampilkan nilai tapi mengurutkan berdasarkan lainnya juga berguna ketika anda ingin menampilkan nilai komposit yang dibentuk dari banyak kolom tapi tidak urut seperti yang diinginkan. Misalnya, tabel mail menampilkan daftar pengirim pesan menggunakan nilai srcuser dan srchost. Jika anda ingin menampilkan pengirim pesan dari tabel mail sebagai alamat email dalam format srcuser@srchost dengan username terlebih dulu, anda dapat membuat nilai tersebut menggunakan ekspresi berikut :

```
CONCAT(srcuser, '@', srchost)
```

Tapi nilai ini tidak baik untuk melakukan pengurutan jika anda ingin memperlakukan hostname sebagai username yang lebih penting. Lebih baik mengurutkan hasil menggunakan nilai-nilai kolom yang asli dari pada menggunakan nilai komposit :

```
mysql> SELECT t, CONCAT(srcuser, '@', srchost) AS sender, size
-> FROM mail WHERE size > 50000 ORDER BY srchost, srcuser;
```

t	sender	size
2001-05-15 10:25:52	gene@mars	998532
2001-05-12 12:48:13	tricia@mars	194925
2001-05-11 10:15:08	barb@saturn	58274
2001-05-14 17:03:01	tricia@saturn	2394482
2001-05-14 14:42:21	barb@venus	98151

1.8 Mengurutkan dan Nilai NULL

Penyelesaian

Penempatan nilai NULL pada daftar yang diurutkan telah berubah dan tergantung pada versi MySQL anda. Jika nilai NULL tidak keluar pada posisi yang diinginkan pada urutan, lakukan trik sehingga bisa keluar seperti yang anda inginkan.

Pembahasan

Ketika kolom yang diurutkan mengandung nilai NULL, MySQL meletakkannya berdekatan dalam urutan. Kelihatannya aneh kalau nilai NULL dikelompokkan seperti ini karena mereka mungkin tidak sama :

```
mysql> SELECT NULL = NULL;
```

```
+-----+
| NULL = NULL |
+-----+
| NULL        |
+-----+
```

Dilain pihak, nilai NULL secara konsep sangar mirip satu sama lain dari pada dengan nilai bukan NULL, dan tidak ada cara yang baik untuk membedakan satu NULL dari yang lain. Tapi meskipun nilai NULL dikelompokkan bersama, mereka bisa ditempatkan pada awal atau akhir urutan, tergantung apda versi MySQL.

Tapi anda dapat memaksa untuk menempatkan nilai NULL pada awal atau akhir urutan apapun versi MySQL anda. Misalkan anda mempunyai tabel t dengan isi berikut ini :

```
mysql> SELECT val FROM t;
```

```
+-----+
| val |
+-----+
| 3    |
| 100  |
| NULL |
| NULL |
| 9    |
+-----+
```

Biasanya, pengurutan meletakkan nilai NULL pada awal :

```
mysql> SELECT val FROM t ORDER BY val;
```

```
+-----+
| val |
+-----+
| NULL |
| NULL |
| 3    |
| 9    |
| 100  |
+-----+
```

Untuk meletakkannya pada akhir urutan, gunakan ekstra kolom ORDER BY yang memetakan nilai NULL ke nilai yang lebih tinggi dari pada nilai bukan NULL :

```
mysql> SELECT val FROM t ORDER BY IF(val IS NULL,1,0), val;
```

```
+-----+
| val |
+-----+
| 3    |
| 9    |
+-----+
```

100
NULL
NULL

Ini juga bisa dilakukan dengan DESC :

```
mysql> SELECT val FROM t ORDER BY IF(val IS NULL,1,0), val DESC;
```

val
100
9
3
NULL
NULL

Jika MySQL meletakkan nilai NULL pada akhir urutan dan anda ingin meletakkan pada awal, gunakan teknik yang sama, tapi baliklah argumen kedua dan ketiga dari fungsi IF () untuk memetakan nilai NULL ke nilai yang lebih rendah dari nilai bukan NULL :

```
IF(val IS NULL,0,1)
```

1.9 Mengontrol Case Sensitivity dari Pengurutan String

Penyelesaian

Rubahlah **case sensitivity** dari nilai yang diurutkan.

Pembahasan

Sifat ini terbawa dalam pengurutan : ORDER BY menghasilkan pengurutan leksikal yang case sensitive untuk string biner tapi tidak untuk string bukan biner. Tabel textblob.val berikut mengandung kolom TEXT tstr dan kolom BLOB bstr yang akan menunjukkan hal berikut :

```
mysql> SELECT * FROM textblob_val;
```

tstr	bstr
aaa	aaa
AAA	AAA
bbb	bbb
BBB	BBB

Kedua kolom mengandung nilai yang sama. Tapi mereka menghasilkan hasil pengurutan yang berbeda, karena kolom TEXT tidak case sensitive tapi kolom BLOB case sensitive :

```
mysql> SELECT tstr FROM textblob_val ORDER BY tstr;
```

tstr
aaa
AAA
bbb
BBB

```
mysql> SELECT bstr FROM textblob_val ORDER BY bstr;
```

bstr

```
+-----+
| AAA |
| BBB |
| aaa |
| bbb |
+-----+
```

Untuk melakukan pengurutan case sensitive untuk string yang tidak case sensitive, rubahlah kolom yang diurutkan ke bentuk string biner menggunakan kata kunci BINARY :

```
mysql> SELECT tstr FROM textblob_val ORDER BY BINARY tstr;
```

```
+-----+
| tstr |
+-----+
| AAA |
| BBB |
| aaa |
| bbb |
+-----+
```

Kemungkinan lain untuk merubah kolom keluaran ke biner dan mengurutkan adalah :

```
mysql> SELECT BINARY tstr FROM textblob_val ORDER BY 1;
```

```
+-----+
| BINARY tstr |
+-----+
| AAA |
| BBB |
| aaa |
| bbb |
+-----+
```

Anda juga dapat menggunakan fungsi CAST () yang tersedia pada MySQL 4.0.2 :

```
mysql> SELECT tstr FROM textblob_val ORDER BY CAST(tstr AS BINARY);
```

```
+-----+
| tstr |
+-----+
| AAA |
| BBB |
| aaa |
| bbb |
+-----+
```

Hal yang berbeda adalah untuk mengurutkan string biner dalam cara non case sensitive. Untuk melakukan ini, rubahlah nilai ke uppercase atau lowercase dengan UPPER () atau LOWER () :

```
mysql> SELECT bstr FROM textblob_val ORDER BY UPPER(bstr);
```

```
+-----+
| bstr |
+-----+
| aaa |
| AAA |
| bbb |
| BBB |
+-----+
```

Cara lain adalah anda dapat merubah kolom keluaran dan mengurutkannya, tapi melakukan hal ini bisa mempengaruhi nilai yang ditampilkan, kemungkinan dalam cara yang tidak dikehendaki :

```
mysql> SELECT UPPER(bstr) FROM textblob_val ORDER BY 1;
```

```
+-----+
| UPPER(bstr) |
+-----+
```

AAA
AAA
BBB
BBB

1.10 Pengurutan Berdasarkan Tanggal

Penyelesaian

Pengurutan menggunakan tipe kolom tanggal atau waktu, mengabaikan bagian nilai yang tidak berhubungan jika perlu.

Pembahasan

Banyak tipe informasi yang mengikutsertakan informasi tanggal atau waktu dan sering perlu untuk mengurutkan hasilnya dalam urutan waktu. MySQL tahu bagaimana mengurutkan tipe kolom waktu, sehingga tidak ada trik khusus untuk mengurutkan nilai dalam kolom DATE, DATETIME, TIME atau TIMESTAMP. Berikut adalah tabel yang akan digunakan dalam contoh :

```
mysql> SELECT * FROM temporal_val;
```

d	dt	t	ts
1970-01-01	1884-01-01 12:00:00	13:00:00	19800101020000
1999-01-01	1860-01-01 12:00:00	19:00:00	20210101030000
1981-01-01	1871-01-01 12:00:00	03:00:00	19750101040000
1964-01-01	1899-01-01 12:00:00	01:00:00	19850101050000

Menggunakan klausa ORDER BY dengan setiap kolom akan mengurutkan nilai ke urutan yang sesuai :

```
mysql> SELECT * FROM temporal_val ORDER BY d;
```

d	data	t	ts
1964-01-01	1899-01-01 12:00:00	01:00:00	19850101050000
1970-01-01	1884-01-01 12:00:00	13:00:00	19800101020000
1981-01-01	1871-01-01 12:00:00	03:00:00	19750101040000
1999-01-01	1860-01-01 12:00:00	19:00:00	20210101030000

```
mysql> SELECT * FROM temporal_val ORDER BY dt;
```

d	dt	t	ts
1999-01-01	1860-01-01 12:00:00	19:00:00	20210101030000
1981-01-01	1871-01-01 12:00:00	03:00:00	19750101040000
1970-01-01	1884-01-01 12:00:00	13:00:00	19800101020000
1964-01-01	1899-01-01 12:00:00	01:00:00	19850101050000

```
mysql> SELECT * FROM temporal_val ORDER BY t;
```

d	dt	t	ts
1964-01-01	1899-01-01 12:00:00	01:00:00	19850101050000
1981-01-01	1871-01-01 12:00:00	03:00:00	19750101040000

1970-01-01	1884-01-01 12:00:00	13:00:00	19800101020000
1999-01-01	1860-01-01 12:00:00	19:00:00	20210101030000

```
mysql> SELECT * FROM temporal_val ORDER BY ts;
```

d	dt	t	ts
1981-01-01	1871-01-01 12:00:00	03:00:00	19750101040000
1970-01-01	1884-01-01 12:00:00	13:00:00	19800101020000
1964-01-01	1899-01-01 12:00:00	01:00:00	19850101050000
1999-01-01	1860-01-01 12:00:00	19:00:00	20210101030000

Kadang-kadang pengurutan waktu hanya menggunakan bagian dari kolom tanggal atau waktu. Dalam kasus ini, anda dapat mengambil bagian yang anda butuhkan dan menggunakannya untuk mengurutkan hasilnya.

1.11 Mengurutkan Berdasarkan Hari Kalender

Penyelesaian

Mengurutkan menggunakan bulan dan tanggal dari suatu tanggal, mengabaikan tahunnya.

Pembahasan

Mengurutkan dalam urutan kalender berbeda dari mengurutkan tanggal. Anda mengabaikan bagian tahun dari tanggal dan mengurutkan hanya menggunakan bulan dan hari untuk mengurutkan record dalam hal dimana mereka jatuh selama tahun kalender. Misalkan anda mempunyai tabel event yang tampak seperti ini ketika nilai diurutkan berdasarkan tanggal kejadian aktual :

```
mysql> SELECT date, description FROM event ORDER BY date;
```

date	description
1215-06-15	Signing of the Magna Carta
1732-02-22	George Washington's birthday
1776-07-14	Bastille Day
1789-07-04	US Independence Day
1809-02-12	Abraham Lincoln's birthday
1919-06-28	Signing of the Treaty of Versailles
1944-06-06	D-Day at Normandy Beaches
1957-10-04	Sputnik launch date
1958-01-31	Explorer 1 launch date
1989-11-09	Opening of the Berlin Wall

Untuk meletakkan kejadian-kejadian ini dalam urutan kalender, urutkan berdasarkan bulan, kemudian berdasarkan hari dalam bulan :

```
mysql> SELECT date, description FROM event ORDER BY MONTH(date), DAYOFMONTH(date);
```

date	description
1958-01-31	Explorer 1 launch date
1809-02-12	Abraham Lincoln's birthday
1732-02-22	George Washington's birthday
1944-06-06	D-Day at Normandy Beaches

1215-06-15	Signing of the Magna Carta
1919-06-28	Signing of the Treaty of Versailles
1789-07-04	US Independence Day
1776-07-14	Bastille Day
1957-10-04	Sputnik launch date
1989-11-09	Opening of the Berlin Wall

Anda juga bisa menggunakan fungsi DAYOFYEAR() yang bisa digunakan untuk mengurutkan hari kalender :

```
mysql> SELECT date, description FROM event ORDER BY DAYOFYEAR(date);
```

date	description
1958-01-31	Explorer 1 launch date
1809-02-12	Abraham Lincoln's birthday
1732-02-22	George Washington's birthday
1944-06-06	D-Day at Normandy Beaches
1215-06-15	Signing of the Magna Carta
1919-06-28	Signing of the Treaty of Versailles
1789-07-04	US Independence Day
1776-07-14	Bastille Day
1957-10-04	Sputnik launch date
1989-11-09	Opening of the Berlin Wall

Tapi dengan menggunakan fungsi DAYOFYEAR() bisa menimbulkan masalah. Dia dapat menghasilkan nilai yang sama untuk hari kalender yang berbeda. Misalnya, 29 Februari dari tahun kabisat dan 1 Maret dari tahun bukan kabisat muncul sebagai hari yang sama :

```
mysql> SELECT DAYOFYEAR('1996-02-29'), DAYOFYEAR('1997-03-01');
```

DAYOFYEAR('1996-02-29')	DAYOFYEAR('1997-03-01')
60	60

Sifat ini berarti bahwa DAYOFYEAR() tidak selalu menghasilkan hasil yang benar dalam pengurutan kalender. Dia dapat mengelompokkan tanggal-tanggal yang sebenarnya terjadi pada hari kalender yang berbeda. Jika suatu tabel menyatakan tanggal-tanggal menggunakan kolom tahun, bulan dan tanggal yang terpisah, pengurutan kalender tidak perlu mengambil hanya bagian tanggalnya. Anda hanya perlu mengurutkan kolom yang sesuai secara langsung. Kita gunakan tabel master sebagai contoh :

```
mysql> SELECT lastname, firstname, birthyear, birthmonth, birthday
-> FROM master;
```

lastname	firstname	birthyear	birthmonth	birthday
AARON	HANK	1934	2	5
AARON	TOMMIE	1939	8	5
AASE	DON	1954	9	8
ABAD	ANDY	1972	8	25
ABADIE	JOHN	1854	11	4
ABBATICCHIO	ED	1877	4	15
ABBEY	BERT	1869	11	29
ABBEY	CHARLIE	1866	10	14

...

Untuk mengurutkan rekord-rekord ini dalam urutan kalender, gunakan kolom birthmont dan birthday. Tentu saja, hal ini akan menyebabkan rekord tidak terurut pada suatu hari, sehingga anda bisa juga menambah kolom tambahan untuk mengurutkan. Query berikut ini memilih pemain dengan tanggal lahir yang diketahui, urutkan berdasarkan urutan kalender dan berdasarkan berdasarkan nama untuk setiap hari kalender :

```
mysql> SELECT lastname, firstname, birthyear, birthmonth, birthday
-> FROM master
-> WHERE birthmonth IS NOT NULL AND birthday IS NOT NULL
-> ORDER BY birthmonth, birthday, lastname, firstname;
```

lastname	firstname	birthyear	birthmonth	birthday
ALLEN	ETHAN	1904	1	1
BEIRNE	KEVIN	1974	1	1
BELL	RUDY	1881	1	1
BERTHRONG	HARRY	1844	1	1
BETHEA	BILL	1942	1	1
BISHOP	CHARLIE	1924	1	1
BOBB	RANDY	1948	1	1
BRUCKMILLER	ANDY	1882	1	1

...

Untuk dataset yang besar, pengurutan menggunakan kolom bagian tanggal yang terpisah dapat lebih cepat dari pada mengurutkan berdasarkan bagian nilai yang diambil dari DATE.

1.12. Mengurutkan Berdasarkan Hari dalam Minggu

Penyelesaian

Gunakan DAYOFWEEK() untuk merubah kolom tanggal ke nilai numerik dari hari dalam minggu.

Pembahasan

Pengurutan hari dalam minggu mirip dengan pengurutan hari kalender, kecuali anda menggunakan fungsi yang berbeda untuk mendapatkan nilai urutan yang sesuai. Anda dapat mendapatkan hari dalam minggu menggunakan DAYNAME(), tapi ini akan menghasilkan string yang mengurutkan secara leksikal dari pada dalam urutan hari dalam minggu (*Minggu, Senin, Selasa, dst*). Disini bisa digunakan cara dengan menampilkan satu nilai tapi mengurutkan dengan yang lainnya. Tampilkan nama hari menggunakan DAYNAME(), tapi urutkan dalam urutan hari dalam minggu menggunakan DAYOFWEEK(), yang mengembalikan nilai numerik dari 1 hingga 7 untuk Minggu hingga Sabtu :

```
mysql> SELECT DAYNAME(date) AS day, date, description
-> FROM event
-> ORDER BY DAYOFWEEK(date);
```

day	date	description
Sunday	1776-07-14	Bastille Day
Sunday	1809-02-12	Abraham Lincoln's birthday
Monday	1215-06-15	Signing of the Magna Carta
Tuesday	1944-06-06	D-Day at Normandy Beaches
Thursday	1989-11-09	Opening of the Berlin Wall
Friday	1957-10-04	Sputnik launch date
Friday	1958-01-31	Explorer 1 launch date
Friday	1732-02-22	George Washington's birthday

Saturday	1789-07-04	US Independence Day	
Saturday	1919-06-28	Signing of the Treaty of Versailles	
+-----+-----+-----+-----+			

Jika anda ingin mengurutkan dalam urutan hari dalam minggu, tapi memperlakukan Senin sebagai hari pertama dalam minggu dan Minggu sebagai hari terakhir, anda dapat menggunakan fungsi MOD () untuk memetakan Senin ke 0, Selasa ke 1, ..., Minggu ke 6 :

```
mysql> SELECT DAYNAME(date), date, description
-> FROM event ORDER BY MOD(DAYOFWEEK(date) + 5, 7);
```

DAYNAME(date)	date	description	
+-----+-----+-----+-----+			
Monday	1215-06-15	Signing of the Magna Carta	
Tuesday	1944-06-06	D-Day at Normandy Beaches	
Thursday	1989-11-09	Opening of the Berlin Wall	
Friday	1957-10-04	Sputnik launch date	
Friday	1958-01-31	Explorer 1 launch date	
Friday	1732-02-22	George Washington's birthday	
Saturday	1789-07-04	US Independence Day	
Saturday	1919-06-28	Signing of the Treaty of Versailles	
Sunday	1776-07-14	Bastille Day	
Sunday	1809-02-12	Abraham Lincoln's birthday	
+-----+-----+-----+-----+			

Tabel berikut menunjukkan ekspresi DAYOFWEEK () yang bisa digunakan untuk meletakkan sembarang hari sebagai hari pertama dalam satu minggu :

Minggu	DAYOFWEEK(date)
Senin	MOD(DAYOFWEEK(date) + 5, 7)
Selasa	MOD(DAYOFWEEK(date) + 4, 7)
Rabu	MOD(DAYOFWEEK(date) + 3, 7)
Kamis	MOD(DAYOFWEEK(date) + 2, 7)
Jum'at	MOD(DAYOFWEEK(date) + 1, 7)
Sabtu	MOD(DAYOFWEEK(date) + 0, 7)

Fungsi lain yang bisa anda gunakan untuk pengurutan hari dalam minggu adalah WEEKDAY () , meskipun dia akan mengembalikan kumpulan nilai yang berbeda (0 untuk Senin hingga 6 untuk Minggu).

1.13 Mengurutkan Berdasarkan Waktu Hari

Penyelesaian

Ambil jam, menit dan detik dari kolom yang mengandung waktu dan gunakanlah untuk mengurutkan.

Pembahasan

Pengurutan berdasarkan waktu hari dapat dilaksanakan dalam beberapa cara, tergantung pada tipe kolomnya. Jika nilainya disimpan dalam kolom TIME, anda tinggal mengurutkannya. Untuk meletakkan nilai DATETIME atau TIMESTAMP dalam urutan waktu hari, ambil bagian waktu dan urutkanlah. Misalnya, tabel mail mengandung nilai DATETIME yang dapat diurutkan berdasarkan waktu hari seperti berikut ini :

```
mysql> SELECT * FROM mail ORDER BY HOUR(t), MINUTE(t), SECOND(t);
```

t	srcuser	srchost	dstuser	dsthost	size
2001-05-15 07:17:48	gene	mars	gene	saturn	3824
2001-05-15 08:50:57	phil	venus	phil	venus	978
2001-05-16 09:00:28	gene	venus	barb	mars	613
2001-05-14 09:31:37	gene	venus	barb	mars	2291
2001-05-11 10:15:08	barb	saturn	tricia	mars	58274
2001-05-15 10:25:52	gene	mars	tricia	saturn	998532
2001-05-14 11:52:17	phil	mars	tricia	saturn	5781
2001-05-12 12:48:13	tricia	mars	gene	venus	194925

...

Anda juga dapat menggunakan `TIME_TO_SEC()` yang memotong bagian tanggal dan mengembalikan bagian waktu yang dinyatakan dalam detik :

```
mysql> SELECT * FROM mail ORDER BY TIME_TO_SEC(t);
```

t	srcuser	srchost	dstuser	dsthost	size
2001-05-15 07:17:48	gene	mars	gene	saturn	3824
2001-05-15 08:50:57	phil	venus	phil	venus	978
2001-05-16 09:00:28	gene	venus	barb	mars	613
2001-05-14 09:31:37	gene	venus	barb	mars	2291
2001-05-11 10:15:08	barb	saturn	tricia	mars	58274
2001-05-15 10:25:52	gene	mars	tricia	saturn	998532
2001-05-14 11:52:17	phil	mars	tricia	saturn	5781
2001-05-12 12:48:13	tricia	mars	gene	venus	194925

...

1.14 Mengurutkan Menggunakan Substring Nilai Kolom

Penyelesaian

Ambil bagian yang anda inginkan dan urutkan mereka secara terpisah.

Pembahasan

Ini adalah aplikasi pengurutan berdasarkan nilai ekspresi. Jika anda ingin mengurutkan record menggunakan hanya bagian tertentu dari suatu nilai kolom, ambil substring yang anda butuhkan dan gunakanlah dalam klausa `ORDER BY`. Ini akan lebih mudah jika substringnya mempunyai posisi dan panjang tetap pada kolom tersebut. Untuk substring dengan posisi dan panjang yang berubah-ubah, anda masih bisa menggunakannya untuk mengurutkan jika ada cara yang tepat untuk mengidentifikasinya.

1.15 Mengurutkan Berdasarkan Substring dengan Panjang Tetap

Penyelesaian

Ambil bagian yang anda butuhkan dengan `LEFT()`, `MID()`, atau `RIGHT()` dan urutkan.

Pembahasan

Misalkan anda mempunyai tabel `housewares` yang berfungsi sebagai katalog untuk peralatan rumah tangga, dan yang barang-barangnya diidentifikasi dengan nilai ID berjumlah 11 karakter dan yang terdiri dari tiga sub bagian: tiga karakter kategori singkatan (*seperti DIN untuk "dinning room" atau KIT untuk "kitchen"*), lima digit nomor seri, dan dua karakter kode negara mengindikasikan negara pembuatnya :

```
mysql> SELECT * FROM housewares;
```

id	description
DIN40672US	dining tabel
KIT00372UK	garbage disposal
KIT01729JP	microwave oven
BED00038SG	bedside lamp
BTH00485US	shower stall
BTH00415JP	lavatory

Jika anda ingin mengurutkan rekord dari tabel ini berdasarkan nilai id, anda dapat langsung menggunakan seluruh nilai kolom :

```
mysql> SELECT * FROM housewares ORDER BY id;
```

id	description
BED00038SG	bedside lamp
BTH00415JP	lavatory
BTH00485US	shower stall
DIN40672US	dining tabel
KIT00372UK	garbage disposal
KIT01729JP	microwave oven

Tapi anda bisa juga ingin mengurutkan berdasarkan satu dari tiga sub bagian tersebut (*misalnya, untuk mengurutkan negara pembuat*). Untuk operasi semacam itu, sangat berguna untuk menggunakan fungsi yang mengambil bagian dari sebuah kolom, seperti `LEFT()`, `MID()`, dan `RIGHT()`. Fungsi-fungsi ini dapat digunakan untuk memisahkan nilai id menjadi tiga komponen :

```
mysql> SELECT id,
-> LEFT(id,3) AS category,
-> MID(id,4,5) AS serial,
-> RIGHT(id,2) AS country
-> FROM housewares;
```

id	category	serial	country
DIN40672US	DIN	40672	US
KIT00372UK	KIT	00372	UK
KIT01729JP	KIT	01729	JP
BED00038SG	BED	00038	SG
BTH00485US	BTH	00485	US
BTH00415JP	BTH	00415	JP

Setiap substring dari nilai id yang mempunyai panjang tetap itu dapat digunakan untuk mengurutkan, baik sendiri maupun dikombinasikan. Untuk mengurutkan kategori produk, ambil nilai kategori dan gunakan dalam klausa `ORDER BY` :

```
mysql> SELECT * FROM housewares ORDER BY LEFT(id,3);
```

id	description
BED00038SG	bedside lamp
BTH00485US	shower stall
BTH00415JP	lavatory
DIN40672US	dining tabel
KIT00372UK	garbage disposal
KIT01729JP	microwave oven

Untuk mengurutkan baris berdasarkan nomor seri produk, gunakan MID() untuk mengambil lima karakter ditengah dari nilai id, dimulai dari digit keempat :

```
mysql> SELECT * FROM housewares ORDER BY MID(id,4,5);
```

id	description
BED00038SG	bedside lamp
KIT00372UK	garbage disposal
BTH00415JP	lavatory
BTH00485US	shower stall
KIT01729JP	microwave oven
DIN40672US	dining tabel

Ini nampak seperti pengurutan numerik, tapi ini sebenarnya pengurutan string, karena MID() mengembalikan string. Nampak bahwa hasil pengurutan mengurutkan karakter dan numerik. Ini terjadi karena fakta bahwa nomor mempunyai angka nol untuk membuat panjang karakter menjadi sama. Untuk mengurutkan berdasarkan kode negara, gunakan dua karakter paling kanan dari nilai id :

```
mysql> SELECT * FROM housewares ORDER BY RIGHT(id,2);
```

id	description
KIT01729JP	microwave oven
BTH00415JP	lavatory
BED00038SG	bedside lamp
KIT00372UK	garbage disposal
DIN40672US	dining tabel
BTH00485US	shower stall

Anda juga dapat menggunakan kombinasi substring. Misalnya, untuk mengurutkan berdasarkan kode negara dan nomor seri, query nya nampak sebagai berikut :

```
mysql> SELECT * FROM housewares ORDER BY RIGHT(id,2), MID(id,4,5);
```

id	description
BTH00415JP	lavatory
KIT01729JP	microwave oven
BED00038SG	bedside lamp
KIT00372UK	garbage disposal

BTH00485US	shower stall
DIN40672US	dining tabel

1.16 Mengurutkan Berdasarkan Substring dengan Panjang Tidak Tetap

Penyelesaian

Cari cara untuk mengidentifikasi bagian yang anda butuhkan sehingga anda dapat mengambilnya.

Pembahasan

Jika substring yang ingin anda gunakan untuk mengurutkan panjangnya tidak tetap, anda membutuhkan alat untuk mengambil hanya bagian dari nilai kolom yang anda inginkan. Untuk melihat apakah ini bisa bekerja, dibuat tabel `housewares2` :

```
mysql> SELECT * FROM housewares2;
```

id	description
DIN40672US	dining tabel
KIT372UK	garbage disposal
KIT1729JP	microwave oven
BED38SG	bedside lamp
BTH485US	shower stall
BTH415JP	lavatory

Bagian kategori dan negara dari nilai id dapat diambil dan diurutkan menggunakan `LEFT()` dan `RIGHT()`, seperti yang dilakukan pada tabel `housewares`. Tapi sekarang bagian numerik dari nilai mempunyai panjang yang berbeda dan tidak dapat diambil dan diurutkan menggunakan pemanggilan `MID()` sederhana. Anda harus menggunakan `SUBSTRING()` untuk meloncati tiga karakter pertama dan mengembalikan sisanya dimulai dari karakter keempat (*digit pertama*) :

```
mysql> SELECT id, SUBSTRING(id,4) FROM housewares2;
```

id	SUBSTRING(id,4)
DIN40672US	40672US
KIT372UK	372UK
KIT1729JP	1729JP
BED38SG	38SG
BTH485US	485US
BTH415JP	415JP

Kemudian ambil semuanya selain dua kolom paling kanan. Salah satu caranya adalah sebagai berikut :

```
mysql> SELECT id, LEFT(SUBSTRING(id,4),LENGTH(SUBSTRING(id,4))-2))
-> FROM housewares2;
```

id	LEFT(SUBSTRING(id,4),LENGTH(SUBSTRING(id,4))-2))
DIN40672US	40672
KIT372UK	372
KIT1729JP	1729

BED38SG	38
BTH485US	485
BTH415JP	415

Tapi pekerjaan diatas terlalu komplek dari seharusnya. Fungsi SUBSTRING() mempunyai argumen ketiga yang opsional yang menentukan panjang hasil yang diinginkan, dan kita tahu bahwa panjang bagian tengah sama dengan panjang string minus lima (*tiga untuk karakter didepan dan dua untuk karakter dibelakang*). Query berikut ini menunjukkan bagaimana untuk mendapatkan bagian numerik ditengan yang dimulai dengan ID kemudian menghilangkan akhiran yang dikanan :

```
mysql> SELECT id, SUBSTRING(id,4), SUBSTRING(id,4,LENGTH(id)-5)
-> FROM housewares2;
```

id	SUBSTRING(id,4)	SUBSTRING(id,4,LENGTH(id)-5)
DIN40672US	40672US	40672
KIT372UK	372UK	372
KIT1729JP	1729JP	1729
BED38SG	38SG	38
BTH485US	485US	485
BTH415JP	415JP	415

Sayangnya, meskipun ekspresi akhir mengambil bagian numerik dengan benar dari ID, namun nilai hasil berupa string. Konsekuensinya, mereka akan diurutkan secara leksikal bukannya secara numerik :

```
mysql> SELECT * FROM housewares2
-> ORDER BY SUBSTRING(id,4,LENGTH(id)-5);
```

id	description
KIT1729JP	microwave oven
KIT372UK	garbage disposal
BED38SG	bedside lamp
DIN40672US	dining tabel
BTH415JP	lavatory
BTH485US	shower stall

Bagaimana untuk menanganinya? Salah satu cara adalah dengan menambah nol yang mengatakan pada MySQL untuk melakukan konversi dari string ke angka yang menghasilkan pengurutan numerik dari nilai nomor seri :

```
mysql> SELECT * FROM housewares2
-> ORDER BY SUBSTRING(id,4,LENGTH(id)-5)+0;
```

id	description
BED38SG	bedside lamp
KIT372UK	garbage disposal
BTH415JP	lavatory
BTH485US	shower stall
KIT1729JP	microwave oven

```
| DIN40672US | dining tabel |
+-----+-----+
```

Tapi pada kasus khusus ini, ada solusi yang lebih baik. Tidak perlu untuk menghitung panjang dari bagian numerik dari string, karena operasi konversi string ke angka akan menghilangkan akhiran yang bukan numerik dan menyediakan nilai yang dibutuhkan untuk mengurutkan bagian nomor seri dengan panjang tidak tetap dari nilai id. Ini berarti bahwa argumen ketiga dari `SUBSTRING()` tidak diperlukan :

```
mysql> SELECT * FROM housewares2
-> ORDER BY SUBSTRING(id,4)+0;
+-----+-----+
| id          | description |
+-----+-----+
| BED38SG     | bedside lamp |
| KIT372UK    | garbage disposal |
| BTH415JP    | lavatory    |
| BTH485US    | shower stall |
| KIT1729JP   | microwave oven |
| DIN40672US  | dining tabel |
+-----+-----+
```

Pada contoh didepan, kemampuan untuk mengambil substring dengan panjang tidak tetap didasarkan pada bermacam karakter yang berbeda yang berada ditengah nilai ID, dibandingkan dengan karakter dibagian akhir (*yaitu digit lawan non-digit*). Pada kasus lain, anda dapat menggunakan karakter pemisah untuk mengambil nilai kolom. Untuk contoh berikutnya kita menggunakan tabel `housewares3` dengan nilai id seperti tampak dibawah ini :

```
mysql> SELECT * FROM housewares3;
+-----+-----+
| id          | description |
+-----+-----+
| 13-478-92-2 | dining tabel |
| 873-48-649-63 | garbage disposal |
| 8-4-2-1     | microwave oven |
| 97-681-37-66 | bedside lamp |
| 27-48-534-2 | shower stall |
| 5764-56-89-72 | lavatory    |
+-----+-----+
```

1.17 Mengurutkan Hostnames dalam Urutan Domain

Penyelesaian

Pisahkan nama dan urutkan bagian-bagiannya dari kanan ke kiri.

Pembahasan

Hostname adalah string dan oleh karena itu pengurutannya adalah leksikal. Tapi, sering diinginkan untuk mengurutkan hostname berdasarkan urutan domain, dimana bagian paling kanan dari nilai hostname lebih penting dari bagian paling kiri. Misalkan anda mempunyai tabel hostname yang mempunyai nama-nama sebagai berikut :

```
mysql> SELECT name FROM hostname ORDER BY name;
+-----+
| name          |
+-----+
| cvs.php.net   |
+-----+
```

dbi.perl.org
jakarta.apache.org
lists.mysql.com
mysql.com
www.kitebird.com

Query di muka menunjukkan pengurutan leksikal berdasarkan nilai nama. Ini berbeda dari urutan domain seperti ditunjukkan pada tabel berikut ini :

Urutan Leksikal	Urutan Domain
cvs.php.net	www.kitebird.com
dbi.perl.org	mysql.com
jakarta.apache.org	lists.mysql.com
lists.mysql.com	cvs.php.net
mysql.com	jakarta.apache.org
<u>www.kitebird.com</u>	dbi.perl.org

Menghasilkan keluaran urutan berdasarkan domain adalah masalah pengurutan substring, dimana diperlukan untuk mengambil masing-masing bagian dari nama sehingga mereka dapat diurutkan dari kiri ke kanan. Ada juga komplikasinya jika nilai anda mengandung jumlah segmen yang berbeda, seperti contoh hostname disini. (*Sebagian besar hostname itu mempunyai tiga segmen, tapi mysql.com hanya punya dua.*). Untuk mengambil bagian dari hostname, mulailah dengan menggunakan SUBSTRING_INDEX(). Nilai hostname mempunyai maksimum tiga segmen, dari segmen inilah bagian-bagian dapat diambil dari kiri ke kanan seperti berikut ini :

```
SUBSTRING_INDEX(SUBSTRING_INDEX(name, '.', -3), '.', 1)
SUBSTRING_INDEX(SUBSTRING_INDEX(name, '.', -2), '.', 1)
SUBSTRING_INDEX(name, '.', -1)
```

Ekspresi ini bekerja dengan baik selama semua hostname mempunyai tiga komponen. Tapi jika ada nama yang kurang dari tiga, tidak akan didapat hasil yang benar, seperti query berikut ini :

```
mysql> SELECT name,
-> SUBSTRING_INDEX(SUBSTRING_INDEX(name, '.', -3), '.', 1) AS leftmost,
-> SUBSTRING_INDEX(SUBSTRING_INDEX(name, '.', -2), '.', 1) AS middle,
-> SUBSTRING_INDEX(name, '.', -1) AS rightmost
-> FROM hostname;
```

name	leftmost	middle	rightmost
cvs.php.net	cvs	php	net
dbi.perl.org	dbi	perl	org
lists.mysql.com	lists	mysql	com
mysql.com	mysql	mysql	com
jakarta.apache.org	jakarta	apache	org
www.kitebird.com	www	kitebird	com

Perhatikan bahwa keluaran dari baris mysql.com. Dia mempunyai mysql untuk nilai kolom paling kiri, dimana ini seharusnya string kosong. Ekspresi pengambilan segmen bekerja dengan cara mengambil sejumlah segmen yang paling kanan, kemudian mengembalikan segmen paling kiri sebagai hasilnya. Sumber dari masalah untuk mysql.com adalah bahwa jika tidak ada sejumlah segmen, ekspresinya akan mengembalikan segmen paling kiri berapapun jumlah segmennya.

Untuk memperbaiki masalah ini, tambahkan sejumlah titik ke nilai hostname untuk menjamin bahwa mereka mempunyai jumlah segmen yang dibutuhkan :

```
mysql> SELECT name,
-> SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT('..',name),'.',-3),'.',1)
-> AS leftmost,
-> SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT('.',name),'.',-2),'.',1)
-> AS middle,
-> SUBSTRING_INDEX(name, '.',-1) AS rightmost
-> FROM hostname;
```

name	leftmost	middle	rightmost
cvs.php.net	cvs	php	net
dbi.perl.org	dbi	perl	org
lists.mysql.com	lists	mysql	com
mysql.com		mysql	com
jakarta.apache.org	jakarta	apache	org
www.kitebird.com	www	kitebird	com

Ini kelihatan jelek. Tapi ekspresi ini tetap membantu mengambil substring yang dibutuhkan untuk mengurutkan nilai hostname dengan benar dari kiri ke kanan :

```
mysql> SELECT name FROM hostname
-> ORDER BY
-> SUBSTRING_INDEX(name, '.',-1),
-> SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT('.',name),'.',-2),'.',1),
-> SUBSTRING_INDEX(SUBSTRING_INDEX(CONCAT('..',name),'.',-3),'.',1);
```

name
www.kitebird.com
mysql.com
lists.mysql.com
cvs.php.net
jakarta.apache.org
dbi.perl.org

Jika anda mempunyai hostname dengan empat segmen, anda harus menambahkan ke klausa ORDER BY ekspresi SUBSTRING_INDEX() yang lain yang menambahkan tiga titik ke nilai hostname.

1.18 Mengurutkan Nilai IP Dotted-Quad dalam Urutan Numerik

Penyelesaian

Pisahkan string dan urutkan bagian-bagiannya secara numerik. Atau cukup gunakan INET_ATON() .

Pembahasan

Jika sebuah tabel mengandung nomor IP yang disajikan sebagai string dalam notasi dotted-quad (*misalnya, 111.122.133.144*), mereka akan diurutkan secara leksikal bukan numerik. Untuk menghasilkan urutan numerik, anda dapat mengurutkannya sebagai nilai empat bagian dimana

masing-masing bagian diurutkan secara numerik. Untuk mencapai hal ini, gunakan teknik yang mirip dengan yang digunakan untuk mengurutkan hostname, tapi dengan perbedaan berikut ini :

- Dotted quad selalu mempunyai empat segmen, sehingga tidak perlu untuk menambah titik ke nilai sebelum mengambil substring.
- Dotted quad mengurutkan dari kiri ke kanan, sehingga urutan yang digunakan substring dalam klausa ORDER BY adalah kebalikan dari yang digunakan untuk mengurutkan hostname.
- Segmen dari nilai dotted-quad adalah bilangan, sehingga tambahkan nol ke masing-masing substring untuk meminta MySQL untuk menggunakan pengurutan numerik dari pada leksikal.

Misalkan anda mempunyai tabel hostip dengan kolom ip yang bernilai string dan mengandung nomor IP :

```
mysql> SELECT ip FROM hostip ORDER BY ip;
```

```
+-----+
| ip      |
+-----+
| 127.0.0.1 |
| 192.168.0.10 |
| 192.168.0.2 |
| 192.168.1.10 |
| 192.168.1.2 |
| 21.0.0.1 |
| 255.255.255.255 |
+-----+
```

Query diatas menghasilkan keluaran yang diurutkan dalam urutan leksikal. Untuk mengurutkan nilai ip secara numerik, anda dapat mengambil masing-masing segmen dan tambahkan nol untuk mengkonversinya ke bilangan menggunakan klausa ORDER BY seperti ini :

```
mysql> SELECT ip FROM hostip
-> ORDER BY
-> SUBSTRING_INDEX(ip, '.',1)+0,
-> SUBSTRING_INDEX(SUBSTRING_INDEX(ip, '.',-3), '.',1)+0,
-> SUBSTRING_INDEX(SUBSTRING_INDEX(ip, '.',-2), '.',1)+0,
-> SUBSTRING_INDEX(ip, '.',-1)+0;
```

```
+-----+
| ip      |
+-----+
| 21.0.0.1 |
| 127.0.0.1 |
| 192.168.0.2 |
| 192.168.0.10 |
| 192.168.1.2 |
| 192.168.1.10 |
| 255.255.255.255 |
+-----+
```

Ada juga penyelesaian yang lebih sederhana jika anda mempunyai MySQL 3.23.15 atau lebih tinggi. Kemudian anda dapat mengurutkan nilai IP menggunakan fungsi INET_ATON() yang secara langsung mengkonversi alamat jaringan ke bentuk numerik :

```
mysql> SELECT ip FROM hostip ORDER BY INET_ATON(ip);
```

ip
21.0.0.1
127.0.0.1
192.168.0.2
192.168.0.10
192.168.1.2
192.168.1.10
255.255.255.255

Jika anda tergoda untuk mengurutkan dengan hanya menambah nol ke nilai ip dan menggunakan ORDER BY pada hasilnya, pertimbangkan nilai yang akan dihasilkan dari konversi string ke angka seperti itu :

```
mysql> SELECT ip, ip+0 FROM hostip;
```

ip	ip+0
127.0.0.1	127
192.168.0.2	192.168
192.168.0.10	192.168
192.168.1.2	192.168
192.168.1.10	192.168
255.255.255.255	255.255
21.0.0.1	21

Konversi hanya mempertahankan nilai sebanyak yang dapat diinterpretasikan sebagai angka. Sisanya tidak akan dipakai dalam pengurutan.

1.19 Mengembangkan Nilai Tertentu menjadi Kepala atau Ekor dari Urutan Penyelesaian

Tambahkan kolom pengurutan lain ke klausa ORDER BY yang menempatkan nilai-nilai itu ditempat yang anda inginkan.

Pembahasan

Jika anda ingin mengurutkan sebuah hasil secara normal kecuali jika anda ingin nilai tertentu lebih dulu, buat kolom pengurutan tambahan yaitu 0 untuk nilai itu dan 1 untuk yang lainnya. Misalkan anda ingin mengurutkan tabel mail dalam urutan pengirim atau penerima, dengan perkecualian yaitu anda ingin meletakkan pesan untuk phil lebih dulu. Anda dapat melakukannya seperti ini :

```
mysql> SELECT t, srcuser, dstuser, size
-> FROM mail
-> ORDER BY IF(srcuser='phil',0,1), srcuser, dstuser;
```

t	srcuser	dstuser	size
2001-05-16 23:04:19	phil	barb	10294
2001-05-12 15:02:49	phil	phil	1048
2001-05-15 08:50:57	phil	phil	978
2001-05-14 11:52:17	phil	tricia	5781

2001-05-17 12:49:23	phil	tricia	873
2001-05-14 14:42:21	barb	barb	98151
2001-05-11 10:15:08	barb	tricia	58274
2001-05-13 13:59:18	barb	tricia	271
2001-05-14 09:31:37	gene	barb	2291
2001-05-16 09:00:28	gene	barb	613
2001-05-15 07:17:48	gene	gene	3824
2001-05-15 17:35:31	gene	gene	3856
2001-05-19 22:21:51	gene	gene	23992
2001-05-15 10:25:52	gene	tricia	998532
2001-05-12 12:48:13	tricia	gene	194925
2001-05-14 17:03:01	tricia	phil	2394482

Nilai dari kolom pengurutan tambahan adalah 0 untuk baris dimana nilai srcuser adalah phil dan 1 untuk baris lainnya. Dengan membuatnya menjadi kolom pengurutan yang paling penting, record untuk pesan yang dikirim oleh phil mengambang ke puncak dari keluaran. Untuk menenggelamkannya ke dasar, anda dapat mengurutkan kolom itu dalam urutan terbalik menggunakan DESC atau membalik urutan argumen kedua dan ketiga dari fungsi IF (). Anda juga dapat menggunakan teknik ini untuk kondisi tertentu, tidak hanya nilai tertentu. Untuk meletakkan lebih dulu record dimana orang mengirimkan pesan ke dirinya sendiri, lakukan ini :

```
mysql> SELECT t, srcuser, dstuser, size
-> FROM mail
-> ORDER BY IF(srcuser=dstuser,0,1), srcuser, dstuser;
```

t	srcuser	dstuser	size
2001-05-14 14:42:21	barb	barb	98151
2001-05-15 07:17:48	gene	gene	3824
2001-05-15 17:35:31	gene	gene	3856
2001-05-19 22:21:51	gene	gene	23992
2001-05-12 15:02:49	phil	phil	1048
2001-05-15 08:50:57	phil	phil	978
2001-05-11 10:15:08	barb	tricia	58274
2001-05-13 13:59:18	barb	tricia	271
2001-05-14 09:31:37	gene	barb	2291
2001-05-16 09:00:28	gene	barb	613
2001-05-15 10:25:52	gene	tricia	998532
2001-05-16 23:04:19	phil	barb	10294
2001-05-14 11:52:17	phil	tricia	5781
2001-05-17 12:49:23	phil	tricia	873
2001-05-12 12:48:13	tricia	gene	194925
2001-05-14 17:03:01	tricia	phil	2394482

Jika anda mengetahui isi tabel anda, anda dapat juga kadang-kadang menghilangkan kolom pengurutan tambahan. Misalnya, srcuser tidak pernah NULL dalam tabel mail, sehingga query sebelumnya dapat ditulis ulang sebagai berikut dengan mengurangi satu kolom dalam klausa ORDER BY (dengan asumsi bahwa nilai NULL diurutkan didepan dari nilai bukan NULL) :

```
mysql> SELECT t, srcuser, dstuser, size
-> FROM mail
-> ORDER BY IF(srcuser=dstuser,NULL,srcuser), dstuser;
```

t	srcuser	dstuser	size
---	---------	---------	------

2001-05-14 14:42:21	barb	barb	98151
2001-05-15 07:17:48	gene	gene	3824
2001-05-15 17:35:31	gene	gene	3856
2001-05-19 22:21:51	gene	gene	23992
2001-05-12 15:02:49	phil	phil	1048
2001-05-15 08:50:57	phil	phil	978
2001-05-11 10:15:08	barb	tricia	58274
2001-05-13 13:59:18	barb	tricia	271
2001-05-14 09:31:37	gene	barb	2291
2001-05-16 09:00:28	gene	barb	613
2001-05-15 10:25:52	gene	tricia	998532
2001-05-16 23:04:19	phil	barb	10294
2001-05-14 11:52:17	phil	tricia	5781
2001-05-17 12:49:23	phil	tricia	873
2001-05-12 12:48:13	tricia	gene	194925
2001-05-14 17:03:01	tricia	phil	2394482

1.20 Mengurutkan dalam Urutan yang Ditentukan Pengguna Penyelesaian

Gunakan `FIELD()` untuk memetakan nilai kolom pada urutan yang menempatkan nilai-nilai dalam urutan yang diinginkan.

Pembahasan

Bagian sebelumnya menunjukkan bagaimana untuk menempatkan kelompok baris tertentu ke urutan atas dari urutan. Jika anda ingin melakukan pengurutan tertentu pada semua nilai dalam suatu kolom, gunakan fungsi `FIELD()` untuk memetakannya ke sebuah daftar nilai numerik dan gunakan angkanya untuk mengurutkan. `FIELD()` membandingkan argumen pertamanya dengan argumen berikutnya dan mengembalikan sebuah angka yang mengindikasikan yang mana yang cocok. Pemanggilan `FIELD()` berikut ini membandingkan nilai terhadap `str1`, `str2`, `str3`, dan `str4` dan mengembalikna 1, 2, 3, atau 4 tergantung pada nilai yang mana yang sama dengan `FIELD(nilai, str1, str2, str3, str4)`.

Jumlah nilai perbandingan tidak perlu empat. `FIELD()` mengambil daftar argumen dengan panjang tidak tetap. Jika nilai adalah `NULL` atau tidak ada nilai yang cocok, `FIELD()` mengembalikan 0. `FIELD()` dapat digunakan untuk mengurutkan sekumpulan nilai acak menjadi terurut sesuai dengan keinginan anda. Sebagai contoh, untuk menampilkan rekord `driver_log` untuk Henry, Suzi, dan Ben, dalam urutan itu, lakukan yang berikut ini :

```
mysql> SELECT * FROM driver_log
-> ORDER BY FIELD(name, 'Henry', 'Suzi', 'Ben');
```

rec_id	name	trav_date	miles
3	Henry	2001-11-29	300
4	Henry	2001-11-27	96
6	Henry	2001-11-26	115
8	Henry	2001-12-01	197
10	Henry	2001-11-30	203
2	Suzi	2001-11-29	391
7	Suzi	2001-12-02	502
1	Ben	2001-11-30	152
5	Ben	2001-11-29	131

9	Ben	2001-12-02	79
---	-----	------------	----

Anda juga dapat menggunakan FIELD() dengan kolom substring. Untuk mengurutkan record dari tabel housewares berdasarkan negara pembuat menggunakan urutan US, UK, JP, SG, lakukan yang berikut ini :

```
mysql> SELECT id, description FROM housewares
-> ORDER BY FIELD(RIGHT(id,2),'US','UK','JP','SG');
```

id	description
DIN40672US	dining tabel
BTH00485US	shower stall
KIT00372UK	garbage disposal
KIT01729JP	microwave oven
BTH00415JP	lavatory
BED00038SG	bedside lamp

Umumnya, FIELD() dapat digunakan untuk mengurutkan sembarang nilai yang berdasarkan kategori kedalam urutan tertentu jika kategori itu tidak dapat diurutkan secara alamiah menjadi urutan yang berguna.

1.21 Mengurutkan Nilai ENUM

Penyelesaian

Pelajari bagaimana mereka bekerja, dan eksploitasi sifat-sifat itu.

Pembahasan

ENUM dianggap sebagai tipe kolom string, tapi nilai ENUM mempunyai sifat khusus yaitu mereka disimpan secara numerik dengan nilai-nilainya diurutkan dengan cara yang sama seperti mereka ditulis dalam definisi tabel. Nilai numerik ini mempengaruhi bagaimana enumerasi disimpan, yang dapat sangat berguna. Misalkan anda mempunyai sebuah tabel bernama weekday yang mengandung sebuah enumerasi kolom day yang mempunyai nama hari sebagai anggotanya :

```
CREATE TABLE weekday
(
day ENUM('Sunday','Monday','Tuesday','Wednesday',
'Thursday','Friday','Saturday')
);
```

Secara internal, MySQL mendefinisikan nilai enumerasi Minggu hingga Sabtu untuk mempunyai nilai numerik dari 1 hingga 7. Untuk melihat hal ini, buatlah tabel menggunakan definisi diatas, kemudian masukkan record hari-hari dalam satu minggu. Tapi untuk membedakan pengurutan pemasukan dari pengurutan biasa, tambahkan hari dalam keadaan tidak terurut :

```
mysql> INSERT INTO weekday (day) VALUES('Monday'),('Friday'),
-> ('Tuesday'), ('Sunday'), ('Thursday'), ('Saturday'), ('Wednesday');
```

Kemudian pilihlah nilainya, baik sebagai string dan sebagai nilai numerik internal (nilai ini didapat dengan menggunakan +0 untuk memberi efek konversi dari string ke angka) :

```
mysql> SELECT day, day+0 FROM weekday;
```

day	day+0
Monday	2
Friday	6
Tuesday	3
Sunday	1
Thursday	5
Saturday	7
Wednesday	4

Perlu dicatat bahwa karena query tidak mengikursertakan klausa ORDER BY, record dikembalikan dalam keadaan tidak terurut. Jika anda menambahkan klausa ORDER BY day, ini akan menjadi jelas bahwa MySQL menggunakan nilai numerik untuk mengurutkan :

```
mysql> SELECT day, day+0 FROM weekday ORDER BY day;
```

day	day+0
Sunday	1
Monday	2
Tuesday	3
Wednesday	4
Thursday	5
Friday	6
Saturday	7

Bagaimana dengan kejadian ketika anda ingin mengurutkan nilai ENUM dalam urutan leksikal ? Nilai-nilai itu harus dianggap sebagai string untuk mengurutkan menggunakan fungsi CONCAT(). CONCAT() biasanya mengambil banyak argumen dan menggabungkannya menjadi satu string. Tapi ini dapat digunakan dengan hanya satu argumen, yang berguna ketika yang anda inginkan adalah cara mendapatkan hasil string :

```
mysql> SELECT day, day+0 FROM weekday ORDER BY CONCAT(day);
```

day	day+0
Friday	6
Monday	2
Saturday	7
Sunday	1
Thursday	5
Tuesday	3
Wednesday	4

Jika anda selalu (*atau hampir selalu*) mengurutkan kolom bukan enumerasi dalam urutan non leksikal tertentu, pertimbangkan untuk merubah tipe kolom ke ENUM, dengan nilainya tercatat dalam urutan yang diinginkan. Untuk melihat bagaimana hal ini bekerja, buat tabel color yang berisi kolom string dan isilah dengan beberapa baris contoh :

```
mysql> CREATE TABEL color (name CHAR(10));
mysql> INSERT INTO color (name) VALUES ('blue'),('green'),
-> ('indigo'),('orange'),('red'),('violet'),('yellow');
```

Mengurutkan berdasarkan kolom nama pada titik nik akan menghasilkan urutan leksikal karena kolomnya mengandung nilai CHAR :

```
mysql> SELECT name FROM color ORDER BY name;
```

name
blue
green
indigo
orange
red
violet
yellow

Sekarang misalkan anda ingin mengurutkan kolom berdasarkan urutan dimana warna muncul dalam urutan pelangi. Salah satu cara untuk mendapatkan urutan pelangi adalah menggunakan FIELD() :

```
mysql> SELECT name FROM color
```

```
-> ORDER BY
```

```
-> FIELD(name, 'red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet');
```

name
red
orange
yellow
green
blue
indigo
violet

Untuk mendapatkan hasil yang sama tanpa FIELD(), gunakan ALTER TABEL untuk mengganti kolom name ke ENUM yang menampilkan warna dalam urutan yang dikehendaki :

```
mysql> ALTER TABEL color
```

```
-> MODIFY name
```

```
-> ENUM('red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet');
```

Setelah mengubah tabel, mengurutkan berdasarkan kolom name akan menghasilkan urutan pelangi secara alamiah dengan tidak memakai perlakuan khusus :

```
mysql> SELECT name FROM color ORDER BY name;
```

name
red
orange
yellow
green
blue
indigo
violet

Referensi:

Paul DuBois, “*MySQL Cookbook*”, O'Reilly - Oktober 2002

BIOGRAFI PENULIS

Janner Simarmata. Lahir di Aek Nabara, 07 Januari 1976. Tamat dari STM GKPS Pematang Siantar tahun 1995. Menyelesaikan program S1 pada jurusan Teknik Informatika di STMIK BANDUNG pada tahun 2000. Pernah mengajar di beberapa Perguruan Tinggi Swasta seperti: STMIK Mikroskil, STMIK Multimedia Prima, Unika Santo Thomas Sumatera Utara. Pada tahun 2004 melanjutkan studi pada program S2 (M.Kom) pada jurusan Ilmu Komputer Universitas Gadjagra Mada sampai sekarang.

Informasi lebih lanjut tentang penulis:

KEYWORD: *Janner Simarmata*

Email: *sijanner@yahoo.com*