5

Pernyataan SELECT (Single Row Function)

Function-Function di MySQL

- Function Numerik
- Function String/Text
- Function Date/Time
- Function Alur Kontrol (Control Flow Function)
- Function Komparasi/Perbandingan
- Function Konversi Tipe Data
- Function Enkripsi dan Kompresi
- Function Lain

FUNCTION NUMERIK

ABS(X)

Mengembalikan nilai absolut dari sebuah bilangan

```
mysql> SELECT ABS(2);
    -> 2

mysql> SELECT ABS(-32);
    -> 32
```

CEILING(X)

Pembulatan ke integer yang lebih besar.

CEIL() = CEILING

```
mysql> SELECT CEILING(1.23);
       -> 2
mysql> SELECT CEILING(-1.23);
       -> -1
```

FLOOR(X)

Pembulatan ke integer yang lebih kecil. Membulatkan ke bawah.

```
mysql> SELECT FLOOR(1.23);
        -> 1
mysql> SELECT FLOOR(-1.23);
        -> -2
```

ROUND(X), ROUND(X,Y)

ROUND(X) : Pembulatan nilai X ke integer terdekatnya

ROUND(X,Y) : Pembulatan nilai X dengan Y angka pecahan

```
mysql> SELECT ROUND(-1.23);
        -> -1
mysql> SELECT ROUND(-1.58);
        -> -2
mysql> SELECT ROUND(1.58);
        -> 2
mysql> SELECT ROUND(1.298, 1);
        -> 1.3
mysql> SELECT ROUND(1.298, 0);
        -> 1
mysql> SELECT ROUND(23.298, -1);
        -> 20
```

CONV(N,from_base,to_base)

Mengkonversi bilangan dari suatu bilangan ke bilangan dengan basis lain.

```
mysql> SELECT CONV('a',16,2);
        -> '1010'
mysql> SELECT CONV('6E',18,8);
        -> '172'
mysql> SELECT CONV(-17,10,-18);
        -> '-H'
mysql> SELECT CONV(10+'10'+'10'+0xa,10,10);
        -> '40'
```

PI()

Mengembalikan Nilai PI (22/7)

```
mysql> SELECT PI();
        -> 3.141593
mysql> SELECT PI()+0.0000000000000000;
        -> 3.14159265358793116
```

RADIANS(X), DEGREES(X)

Radians() = Konversi dari Derajat ke Radian

Degrees() = Konversi dari Radian ke Derajat

```
mysql> SELECT DEGREES(1);
        -> 57.2957795130823
mysql> SELECT DEGREES(PI() / 2);
        -> 90

mysql> SELECT RADIANS(90);
        -> 1.5707963267949
```

SIN(X), COS(X), TAN(X)

Mengembalikan nilai sinus, cosinus, dan tangent. X harus dalam radian, bukan derajat.

```
mysql> SELECT SIN(PI());
        -> 1.224606353822e-16
mysql> SELECT ROUND(SIN(PI()));
        -> 0

mysql> SELECT COS(RADIANS(180));
        -> -1

mysql> SELECT TAN(PI());
        -> -1.224606353822e-16
mysql> SELECT TAN(PI()+1);
        -> 1.5574077246549
```

POW(X,Y), POWER(X,Y)

Mengembalikan nilai X pangkat Y

```
mysql> SELECT POW(2,2);
        -> 4
mysql> SELECT POWER(2,-2);
        -> 0.25
```

RAND()

Menghasilkan Nilai acak antara 0 <= n<1

```
mysql> SELECT RAND();
        -> 0.032589455165179
```

SQRT(X)

Mengembalikan nilai akar quadrat

```
mysql> SELECT SQRT(4);
        -> 2
mysql> SELECT SQRT(20);
        -> 4.4721359549996
mysql> SELECT SQRT(-16);
        -> NULL
```

FUNCTION STRING/TEXT

ASCII(X), ORD(X)

Mengembalikan kode karakter dari karakter pertama suatu string

```
mysql> SELECT ASCII('2');
        -> 50
mysql> SELECT ASCII(2);
        -> 50
mysql> SELECT ASCII('dx');
        -> 100
```

```
mysql> SELECT ORD('2');
        -> 50
mysql> SELECT ORD(2);
        -> 50
mysql> SELECT ORD('dx');
        -> 100
```

BIN(X)

Konversi bilangan desimal ke bilangan biner

```
mysql> SELECT BIN(12);
    -> '1100'
```

OCT(X)

Konversi dari bilangan desimal ke bilangan octal

```
mysql> SELECT OCT(12);
    -> 14
```

HEX(N or S)

HEX(N) : Konversi dari bilangan desimal ke hexadesimal

HEX(S) : Konvesi dari string ke hexadesimal

```
mysql> SELECT HEX(255);
        -> 'FF'
mysql> SELECT 0x616263;
        -> 'abc'
mysql> SELECT HEX('abc');
        -> 616263
```

UNHEX(*str*)

Mengkonversi deret karakter hexa menjadi karakter

```
mysql> SELECT UNHEX('4D7953514C');
        -> 'MySQL'
mysql> SELECT 0x4D7953514C;
        -> 'MySQL'
mysql> SELECT UNHEX(HEX('string'));
        -> 'string'
mysql> SELECT HEX(UNHEX('1267'));
        -> '1267'
```

CHAR_LENGTH(S), LENGTH(S)

LENGTH(S) : Banyaknya byte dalam string.

CHAR_LENGTH(S) : Banyaknya karakter dalam string

```
mysql> SELECT LENGTH('text');
        -> 4
mysql> SELECT CHAR_LENGTH('text');
        -> 4
```

Jika ada sebuah karakter yang multibyte, maka 2 byte bisa dianggap 1 karakter oleh CHAR_LENGTH(), tetapi tetap akan dianggap 2 byte oleh LENGTH()

CHAR(X)

- Mengkonversikan urutan bilangan-bilangan ke dalam suatu string. Jika inputnya berupa string, maka akan dianggap bilangan bulat.

```
mysql> SELECT CHAR(77,121,83,81,'76');
        -> 'MySQL'
mysql> SELECT CHAR(77,77.3,'77.3');
        -> 'MMM'
```

CONCAT(S1, S2, ...)

Menggabungkan sekumpulan string menjadi sebuah string.

```
mysql> SELECT CONCAT('My', 'S', 'QL');
        -> 'MySQL'
mysql> SELECT CONCAT('My', NULL, 'QL');
        -> NULL
mysql> SELECT CONCAT(14.3);
        -> '14.3'
```

CONCAT_WS(X, S1, S2, ...)

Menggabungkan sekumpulan string menjadi sebuah string dengan memberikan string pemisah (X)

```
mysql> SELECT CONCAT_WS(',','First name',
                                'Second name','Last Name');
        -> 'First name,Second name,Last Name'
mysql> SELECT CONCAT_WS(',','First name',NULL,'Last Name');
        -> 'First name,Last Name'
```

FORMAT(N,D)

Memformat bilangan (N) ke dalam string dengan pemisah ribuan (format #,###,###.##) dengan banyaknya pecahan sebanyak D.

```
mysql> SELECT FORMAT(12332.123456, 4);
        -> '12,332.1235'
mysql> SELECT FORMAT(12332.1,4);
        -> '12,332.1000'
mysql> SELECT FORMAT(12332.2,0);
        -> '12,332'
```

INSERT(*str*, *pos*, *len*, *newstr*)

Mereplace suatu string *str* dengan string lain *newstr* dimulai pada posisi *pos* sampai panjang string mencapai *len*. Jika *len* melebihi sisa panjang maka seluruh string mulai dari pos akan direplace dengan newstr

```
mysql> SELECT INSERT('Quadratic', 3, 4, 'What');
        -> 'QuWhattic'
mysql> SELECT INSERT('Quadratic', -1, 4, 'What');
        -> 'Quadratic'
mysql> SELECT INSERT('Quadratic', 3, 100, 'What');
        -> 'QuWhat'
```

INSTR(*str*, *substr*)

Mengembalikan posisi string *substr* pada string *str*

```
mysql> SELECT INSTR('foobarbar', 'bar');
        -> 4
mysql> SELECT INSTR('xbar', 'foobar');
        -> 0
```

LEFT(S,N)

Mengambil N karakter dari string S

```
mysql> SELECT LEFT('foobarbar', 5);
        -> 'fooba'
```

LOCATE(*substr, str*), LOCATE(*substr, str, pos*), POSITION(*substr* IN *str*)

LOCATE(substr, str) : Mencari posisi *substr* pada *str*

LOCATE(substr, str, pos) : Mencari posisi *substr* pada *str* mulai posisi *pos*

POSITION(substr IN str) = LOCATE(substr, str)

```
mysql> SELECT LOCATE('bar', 'foobarbar');
        -> 4
mysql> SELECT LOCATE('xbar', 'foobar');
        -> 0
mysql> SELECT LOCATE('bar', 'foobarbar', 5);
        -> 7
mysql> SELECT POSITION('bar' IN 'foobarbar');
        -> 4
```

LOWER(S), LCASE(S)

Mengkonversi string S ke bentuk huruf kecil

```
mysql> SELECT LOWER('QUADRATICALLY');
        -> 'quadratically'

mysql> SELECT LCASE('QUADRATICALLY');
        -> 'quadratically'
```

UPPER(S), UCASE(S)

Mengkonversi string S ke bentuk huruf kapital

```
mysql> SELECT UPPER('quadratically');
        -> 'QUADRATICALLY'

mysql> SELECT UCASE('quadratically');
        -> 'QUADRATICALLY'
```

LPAD(*str*,*len*,*padstr*)

Mengembalikan string *str* sepanjang *len*, jika panjang string *str* masih kurang dari *len*, maka akan ditambah string *padstr* pada bagian kiri (awal)

```
mysql> SELECT LPAD('hi',4,'?');
        -> '??hi'
mysql> SELECT LPAD('hi',1,'??');
        -> 'h'
```

RPAD(*str*,*len*,*padstr*)

Mengembalikan string *str* sepanjang *len*, jika panjang string *str* masih kurang dari *len*, maka akan ditambah string *padstr* pada bagian kanan (akhir)

```
mysql> SELECT RPAD('hi',4,'?');
        -> 'hi??'
mysql> SELECT RPAD('hi',1,'?');
        -> 'h'
```

LTRIM(*s*), RTRIM(*s*), TRIM(*s*)

LTRIM(s) : Menghapus spasi pada kiri (awal) string

RTRIM(s) : Menghapus spasi pada kanan (akhir) string

TRIM(s) : Menghapus spasi pada kiri dan kanan string

TRIM(s) = LTRIM(RTRIM(s))

```
mysql> SELECT RTRIM('barbar   ');
        -> 'barbar'
mysql> SELECT LTRIM('   barbar ');
        -> 'barbar '
mysql> SELECT TRIM('   barbar   ');
        -> 'barbar'
```

SUBSTRING(*str*,*pos*), SUBSTRING(*str* FROM *pos*), SUBSTRING(*str*,*pos*,*len*), SUBSTRING(*str* FROM *pos* FOR *len*),

SUBSTRING(str, pos) = SUBSTRING(str FROM pos) : mengambil string dari string *str* mulai posisi *pos*.

SUBSTRING(str, *pos*, *len*) = *SUBSTRING(str* FROM *pos* FOR *len*) : Mengambil string dari string *str* mulai posisi *pos* sebanyak *len*

```
mysql> SELECT SUBSTRING('Quadratically',5);
        -> 'ratically'
mysql> SELECT SUBSTRING('foobarbar' FROM 4);
        -> 'barbar'
mysql> SELECT SUBSTRING('Quadratically',5,6);
        -> 'ratica'
mysql> SELECT SUBSTRING('Sakila', -3);
        -> 'ila'
mysql> SELECT SUBSTRING('Sakila', -5, 3);
        -> 'aki'
mysql> SELECT SUBSTRING('Sakila' FROM -4 FOR 2);
        -> 'ki'
```

Jika *pos* negatif, berarti dimulai dari kanan

Function SUBSTR = SUBSTRING
Function MID = SUBSTRING(s,p,l)

REPEAT(*str*,*count*)

Mengulang string *str* sebanyak *count* kali

```
mysql> SELECT REPEAT('MySQL', 3);
    -> 'MySQLMySQLMySQL'
```

REPLACE(*str*,*from_str*,*to_str*)

Mengganti string *from_str* yan ada dalam string *str* dengan string *to_str*

```
mysql> SELECT REPLACE('www.mysql.com', 'w', 'Ww');
        -> 'WwWwWw.mysql.com'
```

REVERSE(*str*)

Mengubah urutan karakter dalam string menjadi terbalik

```
mysql> SELECT REVERSE('abc');
    -> 'cba'
```

RIGHT(*str*,*len*)

Mengambil sebanyak *len* karakter dari string *str* dimulai dari posisi kanan

```
mysql> SELECT RIGHT('foobarbar', 4);
        -> 'rbar'
```

SOUNDEX(*str*)

Mengembalikan soundex dari suatu str

```
mysql> SELECT SOUNDEX('Hello'), SOUNDEX('Hallo');
        -> 'H400', 'H400'
mysql> SELECT SOUNDEX('Quadratically');
        -> 'Q36324'
```

SPACE(N)

Mengembalikan string yang berisi spasi sebanyak N karakter

```
mysql> SELECT SPACE(6);
    -> '      '
```

LATIHAN-LATIHAN

Tabel : world.country

1. Cari negara yang nama negaranya 6 s/d 9 karakter
2. Tampilkan nama negara lengkap dengan localnamenya dengan format “nama negara(localname)”, contoh untuk negara mesir ditampilkan sebagai : 'Egypt(Misr)‘
3. Tampilkan nama negara dan populasinya. Populasi dibulatkan ke seribuan terdekat. Contoh : jika populasi adalah 123456 menjadi 123000
4. Tampilkan nama negara dan populasinya. Populasi dibulatkan ke seribuan diatasnya. Contoh : jika populasi adalah 123456 menjadi 124000
5. Carilah nama negara yang 3 huruf terakhirnya sama dengan 3 huruf terakhir nama benuanya.

LATIHAN-LATIHAN

Tabel : world.city

1. Tampilkan ID kota dan nama kotanya. Untuk ID kota tampilkan dalam 5 digit yang diawali dengan 0. Contoh : untuk kota dengan ID 123 harus ditampilkan 00123
2. Tampilkan kota dan populasinya. Tampilkan populasi dalam format yang menggunakan pemisah ribuan. Tampilkan dalam ribuan. Contoh : Jika populasinya 12345 maka ditampilkan 12 Ribu. Jika populasinya 12345678 maka akan ditampilkan 12,345 Ribu
3. Mirip dengan nomor 1, tetapi jangan menggunakan fungsi LPAD. Buatlah alternatif dari LPAD.

LATIHAN-LATIHAN

Tabel : classicmodels.employees

1. Tampilkan gabungan field FirstName dengan Lastname menjadi field NamaLengkap
2. Seperti nomor 1, lengkapi dengan field email. Tetapi untuk email tidak boleh ditampilkan dalam bentuk email biasa (agar tidak bisa dibaca oleh robot spammer), tetapi dengan mengganti @ dengan [at] dan titik (.) dengan [dot]. Contoh : andri@heryandi.net menjadi andri[at]heryandi[dot]net

FUNCTION DATE/TIME

INTERVAL UNIT

Unit	Format
MICROSECOND	MICROSECONDS
SECOND	SECONDS
MINUTE	MINUTES
HOUR	HOURS
DAY	DAYS
WEEK	WEEKS
MONTH	MONTHS
QUARTER	QUARTERS
YEAR	YEARS

Unit	Format
SECOND_MICROSECOND	'SECONDS.MICROSECONDS'
MINUTE_MICROSECOND	'MINUTES:SECONDS.MICROSECONDS'
MINUTE_SECOND	'MINUTES:SECONDS'
HOUR_MICROSECOND	'HOURS:MINUTES:SECONDS.MICROSECONDS'
HOUR_SECOND	'HOURS:MINUTES:SECONDS'
HOUR_MINUTE	'HOURS:MINUTES'
DAY_MICROSECOND	'DAYS HOURS:MINUTES:SECONDS.MICROSECONDS'
DAY_SECOND	'DAYS HOURS:MINUTES:SECONDS'
DAY_MINUTE	'DAYS HOURS:MINUTES'
DAY_HOUR	'DAYS HOURS'
YEAR_MONTH	'YEARS-MONTHS'

Perhatikan tanda petik tunggal (') pada format.
Contoh :
Penggunaan HOUR_MINUTE ditulis dengan ' 4:30 ' bukan 4:30 untuk menyatakan 4 jam 30 menit.

ADDDATE(*date*,INTERVAL *expr unit*), ADDDATE(*expr*,*days*)

Menambah suatu nilai interval waktu ke sebuah DATE

```
mysql> SELECT ADDDATE('2008-01-02', INTERVAL 31 DAY);
        -> '2008-02-02'

mysql> SELECT ADDDATE('2008-01-02', 31);
        -> '2008-02-02'
```

Jika ADDDATE menggunakan interval, maka mempunyai fungsi yang sama dengan DATE_ADD()

```
mysql> SELECT DATE_ADD('2008-01-02', INTERVAL 31 DAY);
        -> '2008-02-02'
```

SUBDATE(*date*,INTERVAL *expr unit*), SUBDATE(*expr*,*days*)

Mengurangkan suatu nilai interval waktu ke sebuah DATE

```
mysql> SELECT SUBDATE('2008-01-02 12:00:00', 31);
        -> '2007-12-02 12:00:00'

mysql> SELECT SUBDATE('2008-01-02 12:00:00', 31);
        -> '2007-12-02 12:00:00'
```

Jika SUBDATE menggunakan interval, maka mempunyai fungsi yang sama dengan DATE_SUB()

```
mysql> SELECT DATE_SUB('2008-01-02', INTERVAL 31 DAY);
        -> '2007-12-02'
```

ADDTIME(*expr1*,*expr2*)

Menambah nilai expr1 dengan expr2.

Expr1 berupa DATE atau DATETIME, dan expr2 merupa format waktu

```
mysql> SELECT ADDTIME('2007-12-31 23:59:59.999999', '1 1:1:1.000002');
        -> '2008-01-02 01:01:01.000001'
mysql> SELECT ADDTIME('01:00:00.999999', '02:00:00.999998');
        -> '03:00:01.999997'
```

Contoh Format Waktu :

'1 2:3:4.000005' = 1 hari 2 jam 3 menit 4 detik 5 mikrodetik

SUBTIME(*expr1,expr2*)

Mengurangi nilai expr1 dengan expr2.

Expr1 berupa DATE atau DATETIME, dan expr2 merupa format waktu

```
mysql> SELECT SUBTIME('2007-12-31 23:59:59.999999','1 1:1:1.000002');
        -> '2007-12-30 22:58:58.999997'
mysql> SELECT SUBTIME('01:00:00.999999', '02:00:00.999998');
        -> '-00:59:59.999999'
```

Contoh Format Waktu :

'1 2:3:4.000005' = 1 hari 2 jam 3 menit 4 detik 5 mikrodetik

CURRENT_DATE(), CURRENT_DATE, CURDATE()

Mengembalikan tanggal sekarang dalam format ‘YYYY-MM-DD’ atau YYYYMMDD, tergantung cara penggunaan fungsinya apakah sebagai string atau angka.

CURRENT_DATE() dan CURRENT_DATE = CURDATE()

```
mysql> SELECT CURDATE();
        -> '2008-06-13'
mysql> SELECT CURDATE() + 0;
        -> 20080613
```

CURRENT_TIME(), CURRENT_TIME, CURTIME()

Mengembalikan waktu sekarang dalam format ‘HH:MM:SS’ atau HHMMSS.uuuuuu, tergantung cara penggunaan fungsinya apakah sebagai string atau angka.

CURRENT_TIME() dan CURRENT_TIME = CURTIME()

```
mysql> SELECT CURTIME();
        -> '23:50:26'
mysql> SELECT CURTIME() + 0;
        -> 235026.000000
```

CURRENT_TIMESTAMP() , SYSDATE(), CURRENT_TIMESTAMP, NOW()

Mengembalikan nilai DATETIME waktu sekarang. Format yang dikembalikan bisa berupa string atau angka.

Untuk format string akan menampilkan '**YYYY-MM-DD HH:MM:SS.uuuuuu**' dan untuk format angka akan menampilkan **YYYYMMDDHHMMSS.uuuuuu**

CURRENT_TIMESTAMP(), CURRENT_TIMESTAMP, SYSDATE() = NOW()

```
mysql> SELECT NOW();
        -> '2007-12-15 23:50:26'
mysql> SELECT NOW() + 0;
        -> 20071215235026.000000
```

LOCALTIME, LOCALTIME(), LOCALTIMESTAMP, LOCALTIMESTAMP()

Sama Dengan NOW()

UTC_DATE(), UTC_TIME(), UTC_TIMESTAMP()

Mengambil informasi DATE, TIME dan TIMESTAMP berdasarkan waktu UTC (GMT)

```
mysql> SELECT UTC_DATE(), UTC_DATE() + 0;
        -> '2003-08-14', 20030814
mysql> SELECT UTC_TIME(), UTC_TIME() + 0;
        -> '18:07:53', 180753.000000
mysql> SELECT UTC_TIMESTAMP(), UTC_TIMESTAMP() + 0;
        -> '2003-08-14 18:08:04', 20030814180804.000000
Contoh : Indonesia +07:00
mysql> SELECT UTC_DATE(), NOW();
        -> '2009-11-25 17:39:58', '2009-11-26 00:39:58'
```

DATE(*datetime*)

Mengembalikan bagian tanggal dari suatu DATE atau DATETIME.

```
mysql> SELECT DATE('2003-12-31 01:02:03');
        -> '2003-12-31'
```

TIME(*datetime*)

Mengembalikan bagian waktu dari suatu TIME atau DATETIME.

```
mysql> SELECT TIME('2003-12-31 01:02:03');
        -> '01:02:03'
mysql> SELECT TIME('2003-12-31 01:02:03.000123');
        -> '01:02:03.000123'
'
```

DAY(*date*), DAYNAME(*date*), DAYOFMONTH(*date*), DAYOFWEEK(*date*), DAYOFYEAR(*date*)

Mengekstrak informasi tertentu dari date yang berupa DATE/DATETIME

DAYNAME() : Mengambil nama hari

DAY(), DAYOFMONTH() : Ekstrak bagian tanggal

DAYOFYEAR() : Nomor urut hari dalam tahun

DAYOFWEEK() : Nomor urut hari dalam minggu (1=Minggu, 2=Senin, …, 7=Sabtu)

DAY(*date*), DAYNAME(*date*), DAYOFMONTH(*date*), DAYOFWEEK(*date*), DAYOFYEAR(*date*)

```
mysql> SELECT DAYNAME('2007-02-03');
        -> 'Saturday'
mysql> SELECT DAYOFMONTH('2007-02-03');
        -> 3
mysql> SELECT DAYOFYEAR('2007-02-03');
        -> 34
mysql> SELECT DAYOFWEEK('2007-02-03');
        -> 7
```

MONTH(*date*), MONTHNAME(*date*)

MONTH() : mengembalikan nomor bulan (1 .. 12)

MONTHNAME() : mengembalikan nama bulan

```
mysql> SELECT MONTH('2008-02-03');
        -> 2

mysql> SELECT MONTHNAME('2008-02-03');
        -> 'February'
```

YEAR(*date*)

Mengembalikan nilai tahun dari sebuah tanggal

```
mysql> SELECT YEAR('1987-01-01');
        -> 1987
```

WEEK(*date*), YEARWEEK(*date*)

WEEK() : Mengambil nomor minggu dalam tahun tersebut

YEARWEEK() : Menampilkan tahun dan nomor minggu

```
mysql> SELECT WEEK('2008-02-20');
        -> 7
mysql> SELECT YEARWEEK('2008-02-20');
        -> 200807
```

HOUR(*time*), MINUTE(*time*), SECOND(*time*), MICROSECOND(*time*)

HOUR() : Mengembalikan bagian jam

MINUTE() : Mengembalikan bagian menit

SECOND() : Mengembalikan bagian detik

MICROSECOND() : Mengambalikan bagian mikro detik

```
mysql> SELECT HOUR('10:05:03'), HOUR('272:59:59');
        -> 10, 272
mysql> SELECT MINUTE('2008-02-03 10:05:03');
        -> 5
mysql> SELECT SECOND('10:05:03');
        -> 3
mysql> SELECT MICROSECOND('12:00:00.123456');
        -> 123456
```

EXTRACT(*unit* FROM *date*)

Mengambil bagian dari tanggal

```
mysql> SELECT EXTRACT(YEAR FROM '2009-07-02');
       -> 2009
mysql> SELECT EXTRACT(YEAR_MONTH FROM '2009-07-02 01:02:03');
       -> 200907
mysql> SELECT EXTRACT(DAY_MINUTE FROM '2009-07-02 01:02:03');
       -> 20102
mysql> SELECT EXTRACT(MICROSECOND
                      FROM '2003-01-02 10:30:00.000123');
        -> 123
```

QUARTER(*date*)

Mengambil informasi suatu tanggal berada quarter berapa.
Range nilainya 1 .. 4

```
mysql> SELECT QUARTER('2008-04-01');
        -> 2
```

MAKEDATE(*year*,*dayofyear*)

Mengembalikan nilai berupa DATE yang berasal dari tahun dan nomor hari.

```
mysql> SELECT MAKEDATE(2011,31), MAKEDATE(2011,32);
        -> '2011-01-31', '2011-02-01'
mysql> SELECT MAKEDATE(2011,365), MAKEDATE(2012,365);
        -> '2011-12-31', '2012-12-30'
```

MAKETIME(*hour,minute,second*)

Mengembalikan data berupa TIME sesuai parameter inputnya.

```
mysql> SELECT MAKETIME(12,15,30);
        -> '12:15:30'
```

LAST_DAY(*date*)

Mengembalikan tanggal terakhir pada bulan yang ada pada paremeter date.

```
mysql> SELECT LAST_DAY('2003-02-05');
        -> '2003-02-28'
mysql> SELECT LAST_DAY('2004-02-05');
        -> '2004-02-29'
mysql> SELECT LAST_DAY('2004-01-01 01:01:01');
        -> '2004-01-31'
```

UNIX_TIMESTAMP(), UNIX_TIMESTAMP(*date*)

UNIX_TIMESTAMP() : Mengembalikan waktu sekarang berdasarkan UTC dalam detik dimulai dari tanggal **'1970-01-01 00:00:00'**

UNIX_TIMESTAMP(date) : Mengembalikan banyak detik dari tanggal yang jadi parameter dimulai dari **'1970-01-01 00:00:00'** UTC

```
mysql> SELECT UNIX_TIMESTAMP();
        -> 1196440210
mysql> SELECT UNIX_TIMESTAMP('2007-11-30 10:30:19');
        -> 1196440219
```

EXTRACT(*unit* FROM *date*)

Mengekstrak bagian tertentu dari sebuah tanggal

```
mysql> SELECT EXTRACT(YEAR FROM '2009-07-02');
        -> 2009
mysql> SELECT EXTRACT(YEAR_MONTH FROM '2009-07-02 01:02:03');
        -> 200907
mysql> SELECT EXTRACT(DAY_MINUTE FROM '2009-07-02 01:02:03');
        -> 20102
mysql> SELECT EXTRACT(MICROSECOND
    ->                  FROM '2003-01-02 10:30:00.000123');
         -> 123
```

DATEDIFF(*date1,date2*)

Mengembalikan selisih antara 2 tanggal

```
mysql> SELECT DATEDIFF('2007-12-31 23:59:59','2007-12-30');
        -> 1
mysql> SELECT DATEDIFF('2010-11-30 23:59:59','2010-12-31');
        -> -31
```

PERIOD_ADD(*P*,*N*)

Menambah suatu Yeardate (P) sebanyak N bulan.

Hasil dari function ini bukanlah date.

```
mysql> SELECT PERIOD_ADD(200801,2);
    -> 200803
```

PERIOD_DIFF(*P1,P2*)

Mengambil selisih bulan antara P1 dan P2.

Hasil dari function ini adalah angka.

```
mysql> SELECT PERIOD_DIFF(200802,200703);
    -> 11
```

SEC_TO_TIME(*seconds*)

Mengkonversi dari detik ke format TIME

```
mysql> SELECT SEC_TO_TIME(2378);
        -> '00:39:38'
mysql> SELECT SEC_TO_TIME(2378) + 0;
        -> 3938
```

TIME_TO_SEC(*time*)

Mengkonversi dari format TIME ke detik.

```
mysql> SELECT TIME_TO_SEC('22:23:00');
        -> 80580
mysql> SELECT TIME_TO_SEC('00:39:38');
        -> 2378
```

TO_DAYS(*date*)

Konversi dari DATE ke angka, dimulai dari tahun 0.

Tahun 0000 bulan 01 tanggal 01 == 1

```
mysql> SELECT TO_DAYS(950501);
        -> 728779
mysql> SELECT TO_DAYS('2007-10-07');
        -> 733321
```

FROM_DAYS(*day*)

Konversi dari angka ke DATE, dimulai dari tahun 0.

1 = Tahun 0000 bulan 01 tanggal 01

```
mysql> SELECT FROM_DAYS(730669);
        -> '2007-07-03'
```

CONVERT_TZ(*date,from_tz,to_tz*)

Konversi DATETIME dari suatu time zone ke time zone lain.

```
mysql> SELECT CONVERT_TZ('2004-01-01 12:00:00','+00:00','+10:00');
        -> '2004-01-01 22:00:00'
mysql> SELECT CONVERT_TZ('2004-01-01 12:00:00','GMT','MET');
        -> '2004-01-01 13:00:00'
```

Untuk menggunakan contoh ke-dua yang menggunakan nama time zone, maka tabel mysql.timezone harus terisi terlebih dahulu.

DATE_FORMAT(*date*,*format*)

Mengkonversi tanggal menjadi string berdasarkan format tertentu. Beberapa format yang sering digunakan adalah :

Specifier	Deskripsi
%a	Abbreviated weekday name (Sun..Sat)
%b	Abbreviated month name (Jan..Dec)
%c	Month, numeric (0..12)
%D	Day of the month with English suffix (0th, 1st, 2nd, 3rd, …)
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%f	Microseconds (000000..999999)
%H, %k	Hour (00..23)
%h, %l	Hour (01..12)
%i	Minutes, numeric (00..59)
%j	Day of year (001..366)
%M	Month name (January..December)
%m	Month, numeric (00..12)

DATE_FORMAT(*date,format*)

Specifier	Deskripsi
%p	AM or PM
%r	Time, 12-hour (hh:mm:ss followed by AM or PM)
%S, %s	Seconds (00..59)
%T	Time, 24-hour (hh:mm:ss)
%U	Week (00..53), where Sunday is the first day of the week
%u	Week (00..53), where Monday is the first day of the week
%V	Week (01..53), where Sunday is the first day of the week; used with %X
%v	Week (01..53), where Monday is the first day of the week; used with %x
%W	Weekday name (Sunday..Saturday)
%w	Day of the week (0=Sunday..6=Saturday)
%X	Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
%x	Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
%Y	Year, numeric, four digits
%y	Year, numeric (two digits)
%%	A literal “%” character

DATE_FORMAT(*date*,*format*)

```
mysql> SELECT DATE_FORMAT('2009-10-04 22:23:00', '%W %M %Y');
        -> 'Sunday October 2009'
mysql> SELECT DATE_FORMAT('2007-10-04 22:23:00', '%H:%i:%s');
        -> '22:23:00'
mysql> SELECT DATE_FORMAT('1900-10-04 22:23:00',
    ->                    '%D %y %a %d %m %b %j');
        -> '4th 00 Thu 04 10 Oct 277'
mysql> SELECT DATE_FORMAT('1997-10-04 22:23:00',
    ->                    '%H %k %I %r %T %S %w');
        -> '22 22 10 10:23:00 PM 22:23:00 00 6'
mysql> SELECT DATE_FORMAT('1999-01-01', '%X %V');
        -> '1998 52'
mysql> SELECT DATE_FORMAT('2006-06-00', '%d');
        -> '00'
```

TIME_FORMAT(*time*,*format*)

Sama dengan DATE_FORMAT(), tetapi hanya memperbolehkan untuk format jam, menit, detik dan mikrodetik saja. Format selain itu akan menjadikan hasil menjadi NULL atau nol.

STR_TO_DATE(*str,format*)

Konversi dari string menjadi DATE / DATETIME sesuai dengan formatnya.

```
mysql> SELECT STR_TO_DATE('01,5,2013','%d,%m,%Y');
        -> '2013-05-01'
mysql> SELECT STR_TO_DATE('May 1, 2013','%M %d,%Y');
        -> '2013-05-01'
```

FROM_UNIXTIME(*unix_timestamp*), FROM_UNIXTIME(*unix_timestamp*,*format*)

Konversi dari UNIX TIMESTAMP menjadi DATE atau TIME atau string.

```
mysql> SELECT FROM_UNIXTIME(1196440219);
        -> '2007-11-30 10:30:19'
mysql> SELECT FROM_UNIXTIME(1196440219) + 0;
        -> 20071130103019.000000
mysql> SELECT FROM_UNIXTIME(UNIX_TIMESTAMP(),
    ->                      '%Y %D %M %h:%i:%s %x');
        -> '2007 30th November 10:30:59 2007'
```

FUNCTION ALUR CONTROL

CASE

Melakukan operasi seperti Switch Case atau Case Of.
Sebenarnya CASE bukanlah sebuah function.

```
CASE value WHEN [compare_value] THEN result [WHEN [compare_value] THEN result ...] [ELSE result] END

                                    ATAU

CASE WHEN [condition] THEN result [WHEN [condition] THEN result ...] [ELSE result] END
```

```
select Name,Population,
       case when population>=100000000 then 'Banyak'
            when population>=10000000 then 'Lumayan'
            else 'Sedikit' end Keterangan
from world.country;
```

IF(*expr1,expr2,expr3*)

Jika expr1 bernilai TRUE, maka expr2 akan dieksekusi dan jika expr1 bernilai FALSE maka expr3 akan dieksekusi.

```
mysql> SELECT IF(1>2,2,3);
        -> 3
mysql> SELECT IF(1<2,'yes','no');
        -> 'yes'
```

IFNULL(*expr1,expr2*)

Jika expr1 bernilai TRUE, maka function ini akan mereturnkan nilai NULL, jika expr1 bernilai FALSE maka function ini akan mereturnkan nilai sesuai expr2.

```
mysql> SELECT IFNULL(1,0);
        -> 1
mysql> SELECT IFNULL(NULL,10);
        -> 10
mysql> SELECT IFNULL(1/0,10);
        -> 10
mysql> SELECT IFNULL(1/0,'yes');
        -> 'yes'
```

NULLIF(*expr1,expr2*)

Mengembalikan nilai NULL jika expr1 = expr2.

```
mysql> SELECT NULLIF(1,1);
        -> NULL
mysql> SELECT NULLIF(1,2);
        -> 1
```

FUNCTION PERBANDINGAN

GREATEST(*value1,value2,...*)

Mencari nilai terbesar dari suatu list

```
mysql> SELECT GREATEST(2,0);
        -> 2
mysql> SELECT GREATEST(34.0,3.0,5.0,767.0);
        -> 767.0
mysql> SELECT GREATEST('B','A','C');
        -> 'C'
```

LEAST(*value1,value2,...*)

Mencari nilai terkecil dari suatu list

```
mysql> SELECT LEAST(2,0);
        -> 0
mysql> SELECT LEAST(34.0,3.0,5.0,767.0);
        -> 3.0
mysql> SELECT LEAST('B','A','C');
        -> 'A'
```

INTERVAL(*N,N1,N2,N3,...*)

Mengembalikan nilai 0 jika N < N1, 1 jika N < N2, dst. Jika N bernilai NULL maka akan menghasilkan nilai -1

```
mysql> SELECT INTERVAL(23, 1, 15, 17, 30, 44, 200);
        -> 3
mysql> SELECT INTERVAL(10, 1, 10, 100, 1000);
        -> 2
mysql> SELECT INTERVAL(22, 23, 30, 44, 200);
        -> 0
```

ISNULL(*expr*)

Akan menghasilkan nilai TRUE (1) jika expr bernilai NULL atau FALSE (0) jika expr bernilai NOT NULL

```
mysql> SELECT ISNULL(1+1);
        -> 0
mysql> SELECT ISNULL(1/0);
        -> 1
```

COALESCE(*value1, value2, value3,...*)

Mengembalikan nilai pertama dari parameter yang bernilai tidak null

```
mysql> SELECT COALESCE(NULL,1);
        -> 1
mysql> SELECT COALESCE(NULL,NULL,NULL);
        -> NULL
mysql> SELECT COALESCE(IndepYear,Capital)
              FROM World.Country
```

STRCMP(*expr1,expr2*)

Membandingkan string expr1 dengan expr2. Nilai yang akan direturnkan adalah :

0 : jika expr1 = expr2

-1 : jika expr1 < expr2

1 : jika expr1 > expr2

```
mysql> SELECT STRCMP('text', 'text2');
        -> -1
mysql> SELECT STRCMP('text2', 'text');
        -> 1
mysql> SELECT STRCMP('text', 'text');
        -> 0
```

Function Konversi Tipe Data

Name	Description
BINARY	Cast a string to a binary string
CAST()	Cast a value as a certain type
Convert()	Cast a value as a certain type

BINARY

Mengkonversi string ke dalam bentuk BINARY. Jika anda membandingkan nilai dalam bentuk binary, maka berlaku sebagai case sensitive.

```
mysql> SELECT 'a' = 'A';
        -> 1
mysql> SELECT BINARY 'a' = 'A';
        -> 0
mysql> SELECT 'a' = 'a ';
        -> 1
mysql> SELECT BINARY 'a' = 'a ';
        -> 0
```

CONVERT(*expr*,*type*)

Konversi suatu ekspresi ke suatu tipe tertentu.

Tipe dapat berupa

- **BINARY[(*N*)]**
- **CHAR[(*N*)]**
- **DATE , DATETIME , TIME**
- **DECIMAL[(*M*[,*D*])]**
- **SIGNED [INTEGER]**
- **UNSIGNED [INTEGER]**

```
mysql> SELECT 1-2,CONVERT(1-2,UNSIGNED)
    -> -1,18446744073709551615
mysql> SELECT CONVERT(CONVERT(1-2,UNSIGNED),SIGNED);
    -> -1
```

CAST(*expr* AS *type*)

Sama dengan CONVERT(expr,type)

```
mysql> SELECT 1-2,CAST(1-2 AS UNSIGNED)
        -> -1,18446744073709551615
mysql> SELECT CAST(CAST(1-2 AS UNSIGNED) AS SIGNED);
        -> -1
```

FUNCTION ENKRIPSI DAN DEKRIPSI

AES_ENCRYPT(*str*,*key_str*), AES_DECRYPT(*crypt_str*,*key_str*)

Enkripsi/dekripsi menggunakan algoritma AES (Advanced Encryption Standard) terhadap string str dengan kunci key_str

```
mysql > SELECT AES_ENCRYPT('text','password');
     -> 'ö½¨Ü·øÍJ'

Mysql > SELECT AES_DECRYPT(AES_ENCRYPT('text','password'),'password');
     -> 'text'
```

AES_ENCRYPT(*str*,*key_str*), AES_DECRYPT(*crypt_str*,*key_str*)

Enkripsi/dekripsi menggunakan algoritma AES (Advanced Encryption Standard) terhadap string str dengan kunci key_str

```
mysql > SELECT AES_ENCRYPT('text','password');
     -> 'ö½¨Ü·øÍJ'

Mysql > SELECT AES_DECRYPT(AES_ENCRYPT('text','password'),'password');
     -> 'text'
```

DES_ENCRYPT(str), DES_DECRYPT(crypt_str)

Melakukan enkripsi dengan algoritma DES

```
mysql > SELECT DES_ENCRYPT('text');
     -> 'ûùì'

Mysql > SELECT DES_DECRYPT(DES_ENCRYPT('text'));
     -> 'text'
```

ENCODE(*str,pass_str*), DECODE(*crypt_str,pass_str*)

ENCODE() : Enkripsi str dengan pass_str

DECODE() : Decripsi crypt dengan pass_str

```
mysql > SELECT ENCODE('text', 'pass');
     -> 'F}^i'

Mysql > SELECT DECODE(ENCODE('text', 'pass'), 'pass');
     -> 'text'
```

MD5(*str*)

Menghitung MD5 128-bit checksum untuk sebuah string. Hasilnya tidak bisa dibalikan ke string asli (one-way)

```
mysql> SELECT MD5('testing');
        -> 'ae2b1fca515949e5d54fb22b8ed95575'
```

PASSWORD(*str*), OLD_PASSWORD(*str*)

Membuat enkripsi password str. OLD_PASSWORD() merupakan versi terdahulunya. One-Way-Encryption

```
mysql> SELECT PASSWORD('badpwd');
        -> '*AAB3E285149C0135D51A520E1940DD3263DC008C'
mysql> SELECT OLD_PASSWORD('badpwd');
        -> '7f84554057dd964b'
```

COMPRESS(*string_to_compress*)

Kompress String.

```
mysql> SELECT LENGTH(COMPRESS(REPEAT('a',1000)));
        -> 21
mysql> SELECT LENGTH(COMPRESS(''));
        -> 0
mysql> SELECT LENGTH(COMPRESS('a'));
        -> 13
mysql> SELECT LENGTH(COMPRESS(REPEAT('a',16)));
        -> 15
```

UNCOMPRESS(*string_to_uncompress*)

Uncompress string asli dari hasil kompresi

```
mysql> SELECT UNCOMPRESS(COMPRESS('any string'));
        -> 'any string'
mysql> SELECT UNCOMPRESS('any string');
        -> NULL
```

UNCOMPRESSED_LENGTH(*compressed_string*)

Menghitung panjang string hasil uncompress dari string yang telah terkompressi

```
mysql> SELECT UNCOMPRESSED_LENGTH(COMPRESS(REPEAT('a',30)));
        -> 30
```

FUNCTION LAIN-LAIN

CONNECTION_ID()

Mengambil ID Koneksi ke Server MySQL

```
mysql> SELECT CONNECTION_ID();
        -> 23786
```

CURRENT_USER(), CURRENT_USER

Mengembalikan kombinasi nama user dan nama host untuk account MySQL yang digunakan oleh server untuk melakukan autentifikasi client yang sedang aktif.

```
mysql> SELECT CURRENT_USER();
        -> 'davida@localhost'
```

USER(), SESSION_USER(), SYSTEM_USER()

Mengambil nama user dan nama host dari user yang sedang aktif.

```
mysql> SELECT USER(),SESSION_USER(), SYSTEM_USER();
       -> 'root@localhost', 'root@localhost', 'root@localhost'
```

DATABASE(), SCHEMA()

Mengambil nama database yang sedang dibuka

```
mysql> SELECT DATABASE(), SCHEMA();
        -> 'test', 'test'
```

LAST_INSERT_ID()

Mengambil nilai dari field AUTO_INCREMENT dari insert terakhir.

```
mysql> Perintah SQL insert yang menyertakan field auto_increment

mysql> SELECT LAST_INSERT_ID();
        -> 195
```