

PRAKTIKUM SISTEM BASIS DATA

Modul 10: Operator dan Fungsi SQL

Tujuan Praktikum:

1. Mahasiswa dapat menjelaskan operator-operator yang bisa diterapkan dalam statemen query.
2. Mahasiswa dapat menjelaskan berbagai macam fungsi untuk mengolah data *string*, *numeric*, *date*, *cast*, dan *control-flow*.
3. Mahasiswa dapat menerapkan operator atau fungsi yang tepat sesuai dengan kebutuhan informasi dari basis data.

Alat dan Bahan:

1. Modul Praktikum
2. Komputer
3. MySQL phpmyadmin, dengan mengimpor tabel sampel [mahasiswa.sql](#)
4. Viewer LCD
5. NetSupport System (untuk Pretest)

Materi Dan Pembagian Waktu:

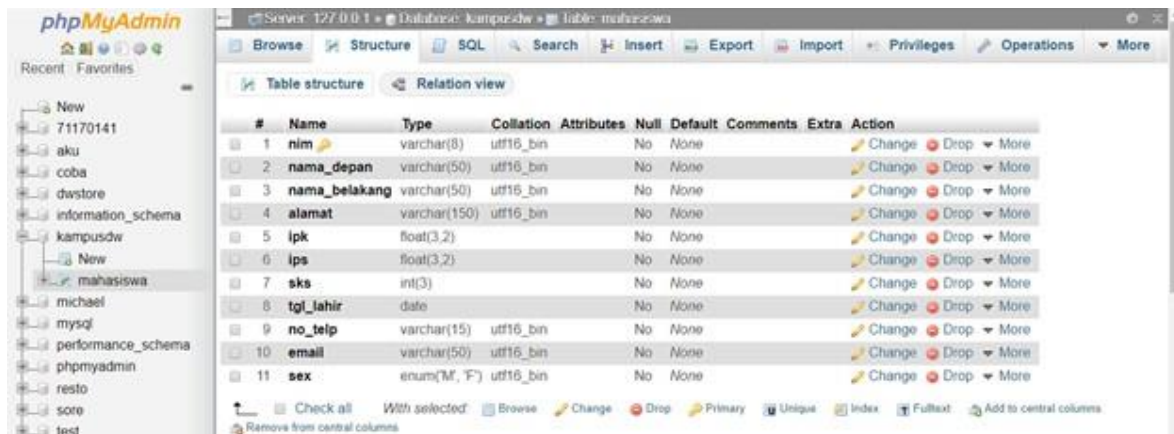
Materi dan pembagian waktu pelaksanaan praktikum meliputi:

No	Materi	Waktu
1	Pretest	10 menit
2	Operator	20 menit
3	String Functions	20 menit
4	Numeric Functions	15 menit
5	Date Functions	15 menit
6	Cast Functions	10 menit
7	Control Flow Functions	15 menit
8	Posttest	45 menit
Total		150 menit

Materi Praktikum

Pada bahasan kali ini, kita akan belajar mengenai berbagai macam operator dan fungsi untuk *men-tuning* statemen query agar lebih efisien dan efektif dalam menyaring data untuk menghasilkan informasi yang diinginkan.

Mahasiswa dipersilakan menyiapkan tabel yang akan digunakan sebagai sampel uji dari materi praktikum kali ini, dengan cara import table. Silakan buat database terlebih dahulu dengan nama **PERT_10_nama** kemudian import file **mahasiswa.sql** yang telah disiapkan di e-class. Amati struktur tabel dengan mengklik tab Jelajahi dan Struktur. Bila melalui console bisa dengan **DESC mahasiswa** dan **SELECT * FROM mahasiswa**.



Gambar 1. Browse & Describe Table Structure

1. Operators

Berikut operator-operator yang sering digunakan:

Operator	Deskripsi
AND, &&	Logical AND
BETWEEN ... AND ...	Mengecek apakah suatu nilai berada pada range nilai.
/	Operator pembagian
=	Operator penyamaan
>, >=	Lebih besar, lebih besar sama dengan
<, <=	Lebih kecil, lebih kecil sama dengan
IS	Menguji suatu nilai terhadap suatu boolean
IS NOT	Menguji suatu nilai terhadap suatu boolean
IS NOT NULL	Menguji nilai NOT NULL
IS NULL	Menguji nilai NULL
LIKE	Pembandingan pola yang sederhana
-	Operator pengurangan
%, MOD	Operator sisa hasil bagi
NOT, !	Menegasikan nilai
NOT BETWEEN ... AND ...	Mengecek apakah satu nilai tidak berada pada range nilai
, OR	Logical OR
+	Operator Penjumlahan
REGEXP	Pencocokan pola menggunakan <i>regular expressions</i>
!=, <>	Operator tidak sama dengan
*	Operator perkalian
XOR	Logical XOR

Latihan:

- Tampilkan mahasiswa perempuan yang berasal dari **Yogyakarta**.
- Tampilkan mahasiswa yang memiliki IPK antara **3.00** hingga **3.50**.
- Tampilkan mahasiswa yang sudah menyelesaikan SKS lebih dari atau sama dengan **70**.
- Tampilkan mahasiswa yang memiliki nama depannya diawali **"Mi.."** atau nama belakangnya berakhiran **"..ya"**.
- Tampilkan NIM, nama depan dan jumlah SKS yang harus ditempuh untuk lulus (mencapai **144** SKS), diurutkan dari yang paling mendekati kelulusan.

2. String Functions

Fungsi string sering digunakan untuk mengelola kolom dan isi data yang akan ditampilkan (digunakan di klausa SELECT) maupun untuk melengkapi filter data (digunakan di klausa WHERE). Berikut ini operator string yang sering digunakan:

Fungsi	Deskripsi
CHAR_LENGTH()	Mengembalikan jumlah karakter
CONCAT(<i>str1</i> , <i>str2</i> , ...)	Menggabungkan string
INSERT(<i>str</i> , <i>pos</i> , <i>Len</i> , <i>newstr</i>)	Menyisipkan <i>newstr</i> pada posisi dan panjang tertentu dari string
LOWER(), LCASE()	Mengubah string menjadi lower case
LEFT(<i>str</i> , <i>Len</i>)	Mengembalikan sekian karakter dari sisi kiri string
LOCATE(<i>substr</i> , <i>str</i> , <i>pos</i>)	Mengembalikan letak penemuan pertama dari suatu substring dimulai dari posisi tertentu
LPAD(<i>str</i> , <i>Len</i> , <i>padstr</i>)	Menempelkan <i>padstr</i> ke kiri <i>str</i> dengan total panjang karakter setelah penempelan sejumlah <i>len</i>
LTRIM(<i>str</i>)	Menghapus spasi di awal
MID(<i>str</i> , <i>pos</i> , <i>Len</i>), SUBSTR(<i>str</i> , <i>pos</i> , <i>Len</i>)	Mengembalikan suatu substring dimulai dari posisi tertentu
POSITION(<i>substr</i> IN <i>str</i>)	Sama dengan LOCATE(<i>substr</i> , <i>str</i>)
REPEAT(<i>str</i> , <i>count</i>)	Mengulangi string sebanyak <i>count</i> yang ditentukan
REPLACE(<i>str</i> , <i>from_str</i> , <i>to_str</i>)	Mengembalikan <i>str</i> dengan menggantikan string awal <i>from_str</i> menjadi <i>to_str</i>
REVERSE(<i>str</i>)	Membalik susunan karakter dari suatu string
RIGHT(<i>str</i> , <i>Len</i>)	Mengembalikan sekian karakter dari sisi kanan string
RPAD(<i>str</i> , <i>Len</i> , <i>padstr</i>)	Menempelkan <i>padstr</i> ke kanan <i>str</i> dengan total panjang karakter setelah penempelan sejumlah <i>len</i>
RTRIM(<i>str</i>)	Menghapus spasi di belakang string
STRCMP()	Membandingkan dua strings
SPACE(<i>N</i>)	Mengembalikan spasi sebanyak <i>N</i>
SUBSTRING_INDEX(<i>str</i> , <i>delim</i> , <i>count</i>)	Mengembalikan suatu substring dari string sebelum sejumlah delimiter muncul. Nilai <i>count</i> minus berarti dihitung dari kanan
TRIM({BOTH LEADING TRAILING} [<i>remstr</i>] FROM <i>str</i>)	Menghapus spasi atau <i>remstr</i> dari <i>str</i> , dari sisi kiri (leading), sisi kanan (trailing) atau keduanya (both)
UPPER(<i>str</i>)	Mengubah string menjadi upper case

Latihan:

- Tampilkan NIM dan gabungan nama depan dan nama belakang sebagai nama lengkap dari mahasiswa diurutkan NIM.
- Tambahkan data email di column email dengan ketentuan "nama depan" + "." + "nama belakang"+"@gmail.ac.id" , contoh : andika.rangga@gmail.ac.id. Semua lowercase.
- Ubahlah data email dengan mengganti kata "gmail" menjadi "ukdw" , contoh : andika.rangga@ukdw.ac.id.
- Tampilkan mahasiswa yang merupakan mahasiswa prodi SI (72...), dengan menggunakan column NIM tanpa menggunakan wildcard % maupun _.
- Tampilkan mahasiswa yang merupakan mahasiswa angkatan 2017 dengan menggunakan column NIM tanpa menggunakan wildcard % maupun _.
- Tampilkan NIM dan hanya nama kabupaten/kota saja dari kolom alamat.

3. Numeric Functions

Fungsi numerik digunakan untuk mengelola nilai data. Fungsi ini bisa digunakan di klausa SELECT maupun untuk melengkapi filter data di klausa WHERE. Berikut ini fungsi numerik yang sering digunakan:

Fungsi	Deskripsi
ABS(<i>x</i>)	Mengembalikan nilai absolut
ACOS(<i>x</i>) , ASIN(<i>x</i>) , ATAN(<i>x</i>)	Mengembalikan arc cosinus, arc sinus, arc tangen
CEILING(<i>x</i>)	Mengembalikan pembulatan ke atas
COS(<i>x</i>) , SIN(<i>x</i>) , TAN(<i>x</i>) , COT(<i>x</i>)	Mengembalikan cosinus, sinus, tangen, cotangen. Nilai <i>x</i> dalam radian.
DEGREES(<i>x</i>)	Mengkonversi radian ke derajat
DIV	Pembagian integer
EXP(<i>x</i>)	Mengembalikan bilangan <i>e</i> pangkat <i>x</i>
FLOOR(<i>x</i>)	Mengembalikan pembulatan ke bawah
LN(<i>x</i>)	Mengembalikan logaritma natural dari <i>x</i>
LOG(<i>b</i>, <i>x</i>)	Mengembalikan logaritma basis <i>b</i> dari <i>x</i>
MOD(<i>n</i>, <i>m</i>)	Mengembalikan sisa hasil bagi dari <i>n</i> dibagi <i>m</i>
PI()	Mengembalikan nilai pi
POWER(<i>x</i>, <i>y</i>)	Mengembalikan nilai <i>x</i> pangkat <i>y</i>
RADIANS(<i>x</i>)	Mengkonversi derajat ke radian
RAND()	Mengembalikan nilai floating-point acak
ROUND(<i>x</i>, <i>d</i>)	Membulatkan <i>x</i> hingga sampai <i>d</i> angka desimal
SIGN(<i>x</i>)	Mengembalikan tanda positif/negatif
SQRT(<i>x</i>)	Mengembalikan akar kuadrat dari <i>x</i>
TRUNCATE(<i>x</i>, <i>d</i>)	Memotong angka desimal hingga <i>d</i> angka di belakang koma

Selain yang tercantum di atas, masih ada beberapa fungsi numerik yang disebut sebagai fungsi agregasi antara lain SUM, AVG, COUNT, dan lain-lain yang akan kita bahas pada pertemuan berikutnya.

Latihan:

- Hitung luas lingkaran dengan jari-jari 5 cm. Hasil harus menuliskan satuan cm dan dibulatkan dua angka di belakang koma.
- Hitung nilai sinus dan kosinus dari sudut 120°.

4. Date & Time Functions

Struktur data penanggalan dan pewaktuan di MySQL lebih mudah dikelola dibandingkan di Oracle maupun DBMS lainnya, karena kita bisa memperlakukannya layaknya karakter biasa. Namun tentunya untuk menghindari kesalahan pengolahan, kita tetap membutuhkan fungsi-fungsi khusus penanggalan dan pewaktuan.

Fungsi	Deskripsi
ADDDATE (<i>date</i> , INTERVAL <i>n</i> <i>expr_unit</i>)	Menambah <i>n expr_unit</i> dari <i>date</i> yang dimasukkan. <i>Expr_unit</i> bisa berupa DAY/WEEK/MONTH/QUARTER/YEAR
ADDTIME (<i>expr1</i> , <i>expr2</i>)	Menambah waktu <i>expr2</i> terhadap waktu <i>expr1</i>
CONVERT_TZ (<i>datetime</i> , <i>from_tz</i> , <i>to_tz</i>)	Mengkonversi zona waktu
CURDATE (), CURRENT_DATE ()	Mengembalikan tanggal saat perintah dijalankan, diberikan dalam bentuk YYYY-MM-DD
CURTIME (), CURRENT_TIME ()	Mengembalikan waktu saat perintah dijalankan
CURRENT_TIMESTAMP (), NOW ()	Mengembalikan timestamp saat perintah dijalankan
DATE (<i>expr</i>)	Mengekstrak date dari format datetime
DATEDIFF (<i>expr1</i> , <i>expr2</i>)	Menghitung selisih hari antara <i>expr1</i> dan <i>expr2</i>
DATE_FORMAT (<i>date</i> , <i>format</i>)	Memformat date berdasarkan format tertentu. Penulisan format dalam tanda petik “.
	%a Singkatan nama hari (Sun..Sat)
	%b Singkatan nama bulan (Jan..Dec)
	%c Bulan, numerik (0..12)
	%D Tanggal dalam bentuk ke- (1st, 2nd, ...)
	%d Tanggal, numerik (00..12)
	%e Tanggal, numerik (0..12)
	%f Mikrodetik
	%H Jam, 24 jam (00..23)
	%h Jam, 12 jam (01..12)
	%i Menit, numerik (00..59)
	%j Hari dari tahun (001..366)
	%k Jam (0..23)
	%l Jam (1..12)
	%M Nama bulan (Januari..December)
	%m Bulan, numerik (00..12)
	%p AM atau PM
	%r Waktu, 12 jam diikuti AM atau PM
	%S Detik (00..59)
	%s Detik (00..59)
	%T Waktu, 24 jam (hh:mm:ss)
	%U Pekan (00..53), Sunday hari pertama
	%u Pekan (00..53), Monday hari pertama
	%V Pekan (01..53), Sunday hari pertama
	%v Pekan (01..53), Monday hari pertama
	%W Nama weekday (Sunday..Saturday)
	%w Hari dari pekan (0=Sunday..6=Saturday)
	%X Tahun untuk pekan, Sunday hari pertama
	%x Tahun untuk pekan, Monday hari pertama
	%Y Tahun, numerik 4 digit
	%y Tahun, numerik 2 digit
DAY (<i>date</i>), DAYOFMONTH (<i>date</i>)	Mengembalikan tanggal dari bulan
DAYOFWEEK (<i>date</i>)	Mengembalikan hari ke- dari pekan (1..7)
DAYOFYEAR (<i>date</i>)	Mengembalikan hari ke- dari tahun (1..366)

EXTRACT(<i>unit</i> FROM <i>date</i>)	Mengekstrak <i>unit</i> dari <i>date</i> . Unit berupa DAY/MONTH/YEAR/HOUR/MINUTE/SECOND atau gabungan (misal YEAR_MONTH atau DAY_MINUTE)
HOUR(<i>time</i>)	Mengekstrak jam dari <i>time</i>
LAST_DAY(<i>date</i>)	Mengembalikan tanggal terakhir dari bulan
MINUTE(<i>time</i>)	Mengekstrak menit dari <i>time</i>
MONTH(<i>date</i>)	Mengekstrak bulan dari <i>date</i> (1..12)
MONTHNAME(<i>date</i>)	Mengekstrak nama bulan dari <i>date</i>
QUARTER(<i>date</i>)	Mengembalikan paruh semester dari <i>date</i>
SECOND(<i>time</i>)	Mengembalikan detik dari <i>time</i>
STR_TO_DATE(<i>str</i>, <i>format</i>)	Mengubah string menjadi date format tertentu. Contoh: STR_TO_DATE('May 1, 2017', '%M %d, %Y') menghasilkan 2017-03-01
SUBDATE(<i>date</i>, INTERVAL <i>n</i> <i>expr_unit</i>)	Kebalikan dari ADDDATE
SUBTIME(<i>expr1</i>, <i>expr2</i>)	Kebalikan dari ADDTIME
SYSDATE()	Hampir mirip dengan NOW(), namun SYSDATE() bisa memberikan hasil yang berbeda walau dijalankan dalam satu statemen
TIME(<i>datetime</i>)	Mengembalikan <i>time</i>
TIMEDIFF(<i>datetime1</i>, <i>datetime2</i>)	Mengurangkan <i>datetime1</i> dengan <i>datetime2</i> dalam bentuk <i>time</i>
TIMESTAMP(<i>date</i>)	Membentuk <i>date</i> menjadi YYYY-MM-DD HH:MM:SS
TIMESTAMPDIFF(<i>unit</i>, <i>datetime1</i>, <i>datetime2</i>)	Mengurangkan <i>datetime1</i> dengan <i>datetime2</i> dalam <i>unit</i> yang diinginkan. Unit bisa berupa DAY/MONTH/YEAR/HOUR/MINUTE/SECOND
TIMEFORMAT(<i>time</i>, <i>format</i>)	Cara kerja mirip DATEFORMAT
TO_DAYS(<i>date</i>)	Mengembalikan hitungan hari dimulai dari tahun 0 gregorian (1582 Masehi)
UTC_DATE()	Mengembalikan tanggal universal (Temps Universel Coordonne)
UTC_TIME()	Mengembalikan waktu universal
UTC_TIMESTAMP()	Mengembalikan timestamp universal
WEEK(<i>date</i>)	Mengembalikan pekan (hari pertama Sunday, 0..53)
WEEKDAY(<i>date</i>)	Mengembalikan indeks weekday (0=Monday..6=Sunday)
WEEKOFYEAR(<i>date</i>)	Sama dengan WEEK dengan hari pertama Monday, 1..53
YEAR(<i>date</i>)	Mengembalikan tahun
YEARWEEK(<i>date</i>)	Mengembalikan tahun dan pekan

Latihan:

- Tampilkan mahasiswa yang lahir sebelum tahun 2000.
- Tampilkan NIM, nama lengkap, tanggal lahir dengan format Sun, 6 October 1999.
- Tampilkan mahasiswa yang lahir di bulan ini (gunakan NOW() atau yang ekuivalen).
- Tampilkan data mahasiswa yang lahir di hari Senin.
- Tampilkan NIM, nama_depan, nama_belakang, serta usia setiap mahasiswa.

5. Cast Functions

Fungsi cast adalah fungsi untuk memaksakan perubahan tipe data dari tipe data yang sudah berlaku sebelumnya. Satu cast function yang paling sering dipakai adalah **CONVERT(*expr*, *type*)** dengan *type* sebagai berikut:

Type CONVERT	Deskripsi
BINARY(<i>N</i>)	Menghasilkan string dengan tipe data binary
CHAR(<i>N</i>)	Menghasilkan string dengan tipe data char
DATE	Menghasilkan nilai date
DATETIME	Menghasilkan nilai datetime
DECIMAL(<i>M</i>,<i>D</i>)	Menghasilkan nilai desimal dengan total panjang digit <i>M</i> dan menggunakan <i>D</i> angka di belakang koma
JSON	Menghasilkan nilai JSON
SIGNED INTEGER	Menghasilkan nilai integer bertanda
UNSIGNED INTEGER	Menghasilkan nilai integer tak bertanda
TIME	Menghasilkan nilai time

Latihan:

- Ambil nilai terbesar dari NIM dari prodi Informatika, lalu lakukan increment nilai tersebut sebanyak 1 untuk menciptakan NIM mahasiswa yang baru saja mendaftar.

6. Control-Flow Functions

Seperti bahasa pemrograman pada umumnya, SQL juga memiliki fungsi kontrol alur.

Fungsi	Deskripsi
CASE WHEN <i>condition</i> THEN <i>result</i> [WHEN <i>condition</i> THEN <i>result</i>] ... [ELSE <i>result</i>] END	Fungsi selector CASE pada umumnya. Contoh penggunaan: SELECT CASE WHEN 1>0 THEN 'true' ELSE 'false' END
IF(<i>expr1</i>, <i>expr2</i>, <i>expr3</i>)	Fungsi percabangan IF pada umumnya. Jika <i>expr1</i> TRUE, maka akan mengembalikan <i>expr2</i> , jika FALSE maka akan mengembalikan <i>expr3</i> .
IFNULL(<i>expr1</i>, <i>expr2</i>)	Jika <i>expr1</i> tidak NULL, maka akan mengembalikan <i>expr2</i> . Jika NULL maka akan mengembalikan <i>expr2</i> .
NULLIF(<i>expr1</i>, <i>expr2</i>)	Mengembalikan NULL jika <i>expr1</i> = <i>expr2</i> adalah TRUE. Jika FALSE maka akan mengembalikan <i>expr1</i> .

Latihan:

- Tampilkan NIM, nama lengkap, dan prodi. Jika dia memiliki NIM berawalkan "71" maka prodinya "Informatika", jika NIM berawalkan "72" maka prodinya "Sistem Informasi".
- Periksalah apakah nama depan dari mahasiswa dengan NIM 72180351 dengan 71180390 sama, jika benar munculkan pesan "Nama depan sama", jika beda "Nama depan beda".
- Tampilkan data NIM, nama depan, nama belakang, dan nomor telephone, tapi jika nomor telepon NULL gantilah dengan kata-kata "No Telephone Tidak Tersedia"

REFERENSI:

MySQL Operators and Functions - <https://dev.mysql.com/doc/refman/5.7/en/functions.html>

MySQL Functions - https://www.w3schools.com/sql/sql_ref_mysql.asp

Lynn Beighley, "Head First SQL", 2007, O'Reilly. ISBN: 978-0-596-52684-9