

LAPORAN
PRAKTEK BASIS DATA
FUNGSI NUMERIK



SEMESTER 3

SULIS TIYAH

2001081002

TEKNIK KOMPUTER 2B

JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI TEKNIK KOMPUTER
POLITEKNIK NEGERI PADANG

2021

A. DASAR TEORI

Fungsi Operator Aritmatika dasar yang dimiliki oleh MySQL adalah; penjumlahan, pengurangan, perkalian, dan pembagian. Operator aritmetika tersebut disertakan dalam perintah select yang bertujuan untuk memanipulasi data yang sifatnya sederhana. Untuk kebutuhan yang lebih kompleks guna memanipulasi data secara aritmetika, MySQL juga menyediakan fungsi-fungsi built-in.

Adapun fungsi aritmatika [numerik] built-in yang telah disediakan oleh MySQL:

FUNGSI	KETERANGAN	CONTOH
ABS (x)	Digunakan untuk mendapatkan nilai absolut dari suatu data numerik.	Select ABS (-12); ⇒ 12
ACOS (x)	Fungsi ini dapat menghasilkan nilai arc cosine x, apabila x tidak bernilai antara -1 dan 1 maka akan menghasilkan NULL.	Select ACOS (-0.98765); ⇒ 2,9842681197855
ASIN (x)	Dapat menghasilkan arc sinus x, juga akan menghasilkan NULL jika nilai x tidak bernilai antara -1 dan 1.	Select ASIN (-0.98765); ⇒ -1,4134717929906
ATAN (x)	Menghasilkan arc tangent x.	Select ATAN (5); ⇒ 1,373400766945
ATAN2 (x,y)	Fungsi ini menghasilkan arc tangent x berdasarkan parameter y untuk menentukan kuadrant.	Select ATAN2 (5,3); ⇒ 1.0303768265243
BIN (x)	Fungsi ini digunakan untuk mendapatkan nilai biner dari bilangan x dengan tipe data BIGINT. Hasil dari fungsi ini memiliki tipe data STRING.	Select BIN (5); ⇒ 101
CEIL (x) / CEILING (x)	Nilai yang dihasilkan dari fungsi ini dikonversikan menjadi BIGINT berupa nilai bilangan bulat terbesar yang tidak lebih kecil dari x.	Select CEILING (3.123); ⇒ 4
CONV (x,daribasis,kebasis)	Digunakan untuk melakukan konversi suatu bilangan dari suatu	Select CONV (7, 15, 3); ⇒ 21

	basis bilangan ke basis bilangan lainnya. Hasil dari fungsi ini adalah STRING.	
COS (x)	Digunakan untuk menghitung nilai cosinus x dalam radian.	Select COS (7); ⇒ 0,7539022543433
COT (x)	Menghitung cotangent x.	Select COT (7); ⇒ 1,1475154224051
DEGREES (x)	Dapat menghasilkan besar sudut dalam derajat, dari suatu x dalam radian.	Select DEGREES (7); ⇒ 401,07045659158
EXP (x)	Menghitung e pangkat x.	Select EXP (7); ⇒ 1096,6331584285
FLOOR (x)	Membulatkan bilangan ke integer terdekat yang tidak lebih besar dari x.	Select FLOOR (7.98); ⇒ 7
FORMAT (num,dec)	Membuat format desimal "x.xxx.xxx.xx" dengan membulatkan angka yang didefinisikan dengan dec.	Select FORMAT (12345.67, 5); ⇒ 12,345.67000
GREATEST (x,y)	Fungsi ini digunakan untuk mendapatkan nilai terbesar dari daftar bilangan.	Select GREATEST (9, 200, 344, 5, 3, 2); ⇒ 344
HEX (x)	Untuk menghasilkan bilangan hexadesimal dengan tipe data STRING dari x dengan tipe data BIGINT.	Select HEX(17); ⇒ 11
LEAST (x,y,z,...)	Fungsi ini digunakan untuk mendapatkan nilai terkecil dari daftar bilangan.	Select LEAST (17, 3, 0, 100, 355); ⇒ Hasil: 0
LOG (x)	Untuk mendapatkan nilai logaritma natural dari x.	Select LOG (17); ⇒ 2,8332133440562
LOG10 (x)	Menghasilkan logaritma dasar 10.	Select LOG10 (17); ⇒ 1,2304489213783

MOD (x,y)	Fungsi yang sama ketika menggunakan operator % yaitu untuk mendapatkan sisa bagi.	Select MOD (15, 2); ⇒ 1
OCT (x)	Memberikan bilangan oktal dari suatu bilangan x dengan tipe data BIGINT. Nilai yang dihasilkan bertipe STRING.	Select OCT (15); ⇒ 17
PI ()	Tentu saja digunakan untuk mendapatkan nilai konstanta PI.	Select PI (); ⇒ 3,141593
POW (x,y) / POWER (x,y)	Menghitung x pangkat y.	Select POW (5, 3); ⇒ 125
RADIANS (x)	Digunakan untuk menghasilkan nilai radian dari x dalam derajat. Merupakan kebalikan dari fungsi DEGREES(x).	Select RADIANS (180); ⇒ 3,1415926535898
RAND () dan RAND (seed)	Untuk mendapatkan bilangan random pada pecahan 0 sampai 1. Jika diberikan parameter seed maka jika seed sama bilangan randomnya juga sama dan bisa diprediksi.	Select RAND (); ⇒ 0,89384155151792 Select RAND (180); ⇒ 0,18818008358421
ROUND (x)	Membulatkan bilangan x menjadi bilangan bulat integer.	Select ROUND (5.43); ⇒ 5
ROUND (x,y)	Membulatkan angka termasuk dibelakang koma jika y didefinisikan.	Select ROUND (5.4315, 3); ⇒ 5.432
SIGN (x)	Digunakan untuk menentukan apakah nilai x bernilai positif, negatif atau nol. Fungsi ini akan memberikan nilai -1 untuk data negatif, 1 untuk data positif dan 0 untuk data nol.	Select SIGN (-4.5637); ⇒ -1
SIN (x)	Digunakan untuk menghitung Sinus x dalam radian.	Select SIN (360); ⇒ 0.95891572341431

SQRT (x)	Untuk mencari akar kuadrat bilangan x.	Select SQRT (49); ⇒ 7
TAN (x)	Digunakan untuk menghitung tangent x dalam radian.	Select TAN (180); ⇒ 1.3386902103512
TRUNCATE(x,y)	Memotong bilangan desimal x hingga sejumlah angka dibelakang koma y.	Select TRUNCATE (1234.56789, 3); ⇒ 1234.567

B. ALAT DAN BAHAN

1. PC / Laptop
2. Windows
3. Software XAMPP Control Panel

C. LANGKAH KERJA

1. Aktifkan fasilitas MySQL yang ada pada XAMPP Control Panel

♠ DNPENJUALAN

1. Tampilkan data-data pelanggan yang tidak memiliki kode pos.

```

MariaDB [dbpenjualan]> select * from tpelanggan
-> where kdpos IS NULL;
+-----+-----+-----+-----+-----+-----+
| kdlgn | nmlgn | almlgn | kota   | kdpos | telp  |
+-----+-----+-----+-----+-----+-----+
| P010  | Rudi  | NULL   | Solok  | NULL  | NULL  |
| P015  | Citra | NULL   | Padang | NULL  | NULL  |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.004 sec)

```

♠ CLASSICMODELS

1. Tampilkan data customers yang terdiri dari customername, addressline1, addressline2, city, state yang nama customernya ada unsur Inc, yang addressline2 dan state tidak kosong. --> 5 records

```
MariaDB [classicmodels]> select customerName, addressLine1, addressLine2, city, state
-> from customers
-> where customerName like '%Inc%' and ( addressLine2 IS NOT NULL and state IS NOT NULL);
```

customerName	addressLine1	addressLine2	city	state
Muscle Machine Inc	4092 Furth Circle	Suite 400	NYC	NY
American Souvenirs Inc	149 Spinnaker Dr.	Suite 101	New Haven	CT
Vitachrome Inc.	2678 Kingston Rd.	Suite 101	NYC	NY
Classic Legends Inc.	5905 Pompton St.	Suite 750	NYC	NY
Microscale Inc.	5290 North Pendale Street	Suite 200	NYC	NY

```
5 rows in set (0.019 sec)
```

2. Tampilkan data customer terdiri dari nama, negara dan salesrepemployeenumber untuk customer yang salesrepemployeenumber nya tidak kosong dan tdk berasal dari negara Jerman, Spanyol, USA, UK, Perancis dan Australia --> 34

```
MariaDB [classicmodels]> select customerName , Country, salesRepEmployeeNumber
-> from customers
-> where salesRepEmployeeNumber IS NOT NULL and country NOT IN ('Germany','Spain','USA','UK','France','Australia');
```

customerName	Country	salesRepEmployeeNumber
Baane Mini Imports	Norway	1504
Volvo Model Replicas, Co	Sweden	1504
Danish Wholesale Imports	Denmark	1401
Dragon Souveniers, Ltd.	Singapore	1621
Handji Gifts& Co	Singapore	1612
Herkku Gifts	Norway	1504
Osaka Souveniers Co.	Japan	1621
Toys of Finland, Co.	Finland	1501
Clover Collections, Co.	Ireland	1504
Canadian Gift Exchange Network	Canada	1323
King Kong Collectables, Co.	Hong Kong	1621
Heintze Collectables	Denmark	1401
Qu[-bec Home Shopping Network	Canada	1286
Amica Models & Co.	Italy	1401
Royal Canadian Collectables, Ltd.	Canada	1323
Rovelli Gifts	Italy	1401
Vida Sport, Ltd	Switzerland	1702
Norway Gifts By Mail, Co.	Norway	1504
Oulu Toy Supplies, Inc.	Finland	1501
Petit Auto	Belgium	1401
Down Under Souveniers, Inc	New Zealand	1612
Suominen Souveniers	Finland	1501
GiftsForHim.com	New Zealand	1612
Precious Collectables	Switzerland	1702
Royale Belge	Belgium	1401
Salzburg Collectables	Austria	1401
Cruz & Sons Co.	Philippines	1621
L'ordine Souveniers	Italy	1401
Tokyo Collectables, Ltd	Japan	1621
Extreme Desk Decorations, Ltd	New Zealand	1612
Scandinavian Gift Ideas	Sweden	1504
Mini Auto Werke	Austria	1401
Frau da Collezione	Italy	1401
Kelly's Gift Shop	New Zealand	1612

```
34 rows in set (0.005 sec)
```

3. Tampilkan 10 data customers (terdiri dari customername, addressline1, addressline2, city dan creditlimit), untuk customer yang tidak punya alamat alternatif (addressline 2 kosong) dan creditlimitnya 0. Tampilkan terurut berdasarkan nama --> 10

```

MariaDB [classicmodels]> select customerName, addressLine1, addressLine2, city, creditLimit
-> from customers
-> where addressLine2 IS NULL
-> AND creditLimit = 0
-> ORDER BY customerName limit 10;
+-----+-----+-----+-----+-----+
| customerName | addressLine1 | addressLine2 | city | creditLimit |
+-----+-----+-----+-----+-----+
| ANG Resellers | Gran Vía, 1 | NULL | Madrid | 0.00 |
| Anton Designs, Ltd. | c/ Gobelás, 19-1 Urb. La Florida | NULL | Madrid | 0.00 |
| Asian Treasures, Inc. | 8 Johnstown Road | NULL | Cork | 0.00 |
| BG&E Collectables | Rte des Arsenaux 41 | NULL | Fribourg | 0.00 |
| Cramer Spezialitäten, Ltd | Maubelstr. 90 | NULL | Brandenburg | 0.00 |
| Der Hund Imports | Obere Str. 57 | NULL | Berlin | 0.00 |
| Feuer Online Stores, Inc | Heerstr. 22 | NULL | Leipzig | 0.00 |
| Franken Gifts, Co | Berliner Platz 43 | NULL | München | 0.00 |
| Havel & Zbyszek Co | ul. Filtrowa 68 | NULL | Warszawa | 0.00 |
| Kommission Auto | Luisenstr. 48 | NULL | Münster | 0.00 |
+-----+-----+-----+-----+-----+
10 rows in set (0.006 sec)
```

♠ CEIL () / CEILING ()

```

MariaDB [mysql]> SELECT CEIL (6.6);
+-----+
| CEIL (6.6) |
+-----+
|          7 |
+-----+
1 row in set (0.004 sec)

MariaDB [mysql]> SELECT CEIL (6.1);
+-----+
| CEIL (6.1) |
+-----+
|          7 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT CEILING (6.001);
+-----+
| CEILING (6.001) |
+-----+
|          7 |
+-----+
1 row in set (0.000 sec)
```

♠ FLOOR ()

```
MariaDB [mysql]> SELECT FLOOR (6.9);
+-----+
| FLOOR (6.9) |
+-----+
|           6 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT FLOOR (6.3);
+-----+
| FLOOR (6.3) |
+-----+
|           6 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT FLOOR (6.99);
+-----+
| FLOOR (6.99) |
+-----+
|           6 |
+-----+
1 row in set (0.000 sec)
```

♠ ROUND ()

1. Satu argumen

```
MariaDB [mysql]> SELECT ROUND (5.9);
+-----+
| ROUND (5.9) |
+-----+
|           6 |
+-----+
1 row in set (0.004 sec)

MariaDB [mysql]> SELECT ROUND (5.4);
+-----+
| ROUND (5.4) |
+-----+
|           5 |
+-----+
1 row in set (0.000 sec)
```


2. Dua argumen

```
MariaDB [mysql]> SELECT ROUND (2.673,1);
+-----+
| ROUND (2.673,1) |
+-----+
|                2.7 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT ROUND (2.648,1);
+-----+
| ROUND (2.648,1) |
+-----+
|                2.6 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT ROUND (2.648,2);
+-----+
| ROUND (2.648,2) |
+-----+
|                2.65 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT ROUND (3.5,3);
+-----+
| ROUND (3.5,3) |
+-----+
|              3.500 |
+-----+
1 row in set (0.001 sec)
```

♠ TRUNCATE ()

```
MariaDB [mysql]> SELECT TRUNCATE (2.673,1);
+-----+
| TRUNCATE (2.673,1) |
+-----+
|                2.6 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT TRUNCATE (2.648,1);
+-----+
| TRUNCATE (2.648,1) |
+-----+
|                2.6 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT TRUNCATE (2.648,2);
+-----+
| TRUNCATE (2.648,2) |
+-----+
|                2.64 |
+-----+
1 row in set (0.001 sec)

MariaDB [mysql]> SELECT TRUNCATE (2.648,0);
+-----+
| TRUNCATE (2.648,0) |
+-----+
|                2 |
+-----+
1 row in set (0.000 sec)
```

♠ DIV ()

```
MariaDB [mysql]> SELECT 10 DIV 3;
+-----+
| 10 DIV 3 |
+-----+
|          3 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT 3 DIV 10;
+-----+
| 3 DIV 10 |
+-----+
|          0 |
+-----+
1 row in set (0.000 sec)
```

♠ MOD ()

```
MariaDB [mysql]> SELECT 10 MOD 3;
+-----+
| 10 MOD 3 |
+-----+
|          1 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT 3 MOD 10;
+-----+
| 3 MOD 10 |
+-----+
|          3 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT 10 % 3;
+-----+
| 10 % 3 |
+-----+
|          1 |
+-----+
1 row in set (0.000 sec)
```

♠ Sqrt ()

```
MariaDB [mysql]> SELECT SQRT(81);
+-----+
| SQRT(81) |
+-----+
|          9 |
+-----+
1 row in set (0.004 sec)

MariaDB [mysql]> SELECT ROUND (SQRT(20),2);
+-----+
| ROUND (SQRT(20),2) |
+-----+
|                4.47 |
+-----+
1 row in set (0.004 sec)
```

♠ POW () atau POWER ()

```
MariaDB [mysql]> SELECT POW(5,3);
+-----+
| POW(5,3) |
+-----+
|        125 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT ROUND (POW(2.5,5),1);
+-----+
| ROUND (POW(2.5,5),1) |
+-----+
|                97.7 |
+-----+
1 row in set (0.001 sec)
```

♠ ABS ()

```
MariaDB [mysql]> SELECT ABS (10);
+-----+
| ABS (10) |
+-----+
|         10 |
+-----+
1 row in set (0.000 sec)

MariaDB [mysql]> SELECT ABS (-10);
+-----+
| ABS (-10) |
+-----+
|         10 |
+-----+
1 row in set (0.008 sec)
```

D. TUGAS DAN PEMBAHASAN

DBPENJUALAN

1. Semua barang mendapatkan diskon sebesar 5% dari total harga(jml*hrp).
Tampilkan nonota, kdbrg, totalharga sebelum diskon, diskon dan total harga setelah diskon. Semua perhitungan gunakan pembulatan keatas

```

MariaDB [dbpenjualan]> select nonota, kdbrg, CEIL (jml*hrp) AS total,
-> CEIL (0.05*jml*hrp) AS diskon,
-> CEIL ((jml*hrp)-(0.05*jml*hrp)) AS total_pembayaran
-> from ttrans;

```

nonota	kdbrg	total	diskon	total_pembayaran
T001	B001	375000	187500	3562500
T001	B002	875000	43750	831250
T001	B002	875000	43750	831250
T001	B003	500000	25000	475000
F002	B002	2625000	131250	2493750
F002	B005	500000	25000	475000
F001	B004	2200000	110000	2090000
K001	B001	15000000	750000	14250000
K001	B002	3500000	175000	3325000
K002	B002	6750000	337500	6412500
K002	B005	1500000	75000	1425000
T004	B001	7500000	375000	7125000
T011	B001	875000	43750	831250
T004	B001	3750000	187500	3562500
T004	B002	175000	8750	166250
T004	B003	75000	3750	71250
K003	B001	9000000	450000	8550000
F010	B001	7500000	375000	7125000
F010	B004	3300000	165000	3135000
K005	B001	30000000	1500000	28500000

20 rows in set (0.001 sec)

```

MariaDB [dbpenjualan]> select nonota, kdbrg, CEIL (jml*hrp) AS total,
-> CEIL (0.05*jml*hrp) AS diskon,
-> CEIL (0.95*jml*hrp) AS total_pembayaran
-> from ttrans;

```

nonota	kdbrg	total	diskon	total_pembayaran
T001	B001	375000	187500	3562500
T001	B002	875000	43750	831250
T001	B002	875000	43750	831250
T001	B003	500000	25000	475000
F002	B002	2625000	131250	2493750
F002	B005	500000	25000	475000
F001	B004	2200000	110000	2090000
K001	B001	15000000	750000	14250000
K001	B002	3500000	175000	3325000
K002	B002	6750000	337500	6412500
K002	B005	1500000	75000	1425000
T004	B001	7500000	375000	7125000
T011	B001	875000	43750	831250
T004	B001	3750000	187500	3562500
T004	B002	175000	8750	166250
T004	B003	75000	3750	71250
K003	B001	9000000	450000	8550000
F010	B001	7500000	375000	7125000
F010	B004	3300000	165000	3135000
K005	B001	30000000	1500000	28500000

20 rows in set (0.001 sec)

- Untuk total pembelian yang lebih dari Rp.1.000.000,- akan mendapatkan diskon sebesar 5%. Tampilkan nonota, kdbrg, total harga sebelum diskon, diskon, total harga setelah diskon (pembulatan kebawah)

```

MariaDB [dbpenjualan]> select nonota, kdbrg,
-> FLOOR (jml*hrp) as 'Total Harga Sebelum Diskon',
-> FLOOR (0.05*jml*hrp) as 'Diskon',
-> FLOOR ((jml*hrp)-(0.05*jml*hrp)) as 'Total Harga Setelah Diskon'
-> FROM ttrans
-> where jml*hrp > 1000000;

```

nonota	kdbrg	Total Harga Sebelum Diskon	Diskon	Total Harga Setelah Diskon
T001	B001	3750000	187500	3562500
F002	B002	2625000	131250	2493750
F001	B004	2200000	110000	2090000
K001	B001	15000000	750000	14250000
K001	B002	3500000	175000	3325000
K002	B002	6750000	337500	6412500
K002	B005	1500000	75000	1425000
T004	B001	7500000	375000	7125000
T004	B001	3750000	187500	3562500
K003	B001	9000000	450000	8550000
F010	B001	7500000	375000	7125000
F010	B004	3300000	165000	3135000
K005	B001	30000000	1500000	28500000

13 rows in set (0.000 sec)

```

MariaDB [dbpenjualan]> select nonota, kdbrg,
-> FLOOR (jml*hrp) as 'Total Harga Sebelum Diskon',
-> FLOOR (0.05*jml*hrp) as 'Diskon',
-> FLOOR (0.95*jml*hrp) as 'Total Harga Setelah Diskon'
-> FROM ttrans
-> where jml*hrp > 1000000;

```

nonota	kdbrg	Total Harga Sebelum Diskon	Diskon	Total Harga Setelah Diskon
T001	B001	3750000	187500	3562500
F002	B002	2625000	131250	2493750
F001	B004	2200000	110000	2090000
K001	B001	15000000	750000	14250000
K001	B002	3500000	175000	3325000
K002	B002	6750000	337500	6412500
K002	B005	1500000	75000	1425000
T004	B001	7500000	375000	7125000
T004	B001	3750000	187500	3562500
K003	B001	9000000	450000	8550000
F010	B001	7500000	375000	7125000
F010	B004	3300000	165000	3135000
K005	B001	30000000	1500000	28500000

13 rows in set (0.001 sec)

- untuk pelanggan dengan kode P002 diharuskan membayar pajak barang sebanyak 10% dari total harga. Tapi hanya barang yang dibeli ditahun 2005 dan barang yang harganya diatas Rp.500.000,-. tampilkan data transaksi lengkap dengan besaran pajaknya. (lakukan pembulatan)

```

MariaDB [dbpenjualan]> select * , ROUND(0.1*jml*hrng) as 'Pajak Barang'
-> FROM ttrans
-> WHERE tgl BETWEEN '20050101' AND '20051231'
-> AND hrg > 500000;
+-----+-----+-----+-----+-----+-----+-----+
| nonota | tgl      | kdign | kdbrg | jml  | hrg    | Pajak Barang |
+-----+-----+-----+-----+-----+-----+-----+
| F010   | 2005-03-01 | P002  | B001  | 10   | 750000 | 750000 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

CLASSICMODELS

1. Untuk pelanggan yang melakukan pembayaran pada bulan januari 2003 dari tabel payments, wajib membayar pajak sebanyak 10% dari amount. Tampilkan semua field dan besaran pajak lakukan pembulatan keatas dan urutkan berdasarkan paymentdate.

```

MariaDB [classicmodels]> select * , CEIL(0.1*amount) as 'Pajak'
-> FROM payments
-> WHERE paymentDate BETWEEN '20030101' AND '20030131'
-> ORDER BY paymentDate;
+-----+-----+-----+-----+-----+
| customerNumber | checkNumber | paymentDate | amount | Pajak |
+-----+-----+-----+-----+-----+
| 363            | IS232033   | 2003-01-16  | 10223.83 | 1023 |
| 128            | DI925118   | 2003-01-28  | 10549.01 | 1055 |
| 181            | GQ132144   | 2003-01-30  | 5494.78  | 550  |
+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)

```

2. Carilah berapa total harga yang didapatkan Untuk penjualan product yang bernama Harley dari tabel products, tetapi hanya untuk buyprice yang nilainya dibawah 50. Lakukan pembulatan.

```

MariaDB [classicmodels]> SELECT productName, buyPrice, quantityInStock,
-> ROUND (buyPrice * quantityInStock) AS 'Total Price'
-> FROM products
-> where productName LIKE '%Harley%' AND buyPrice < 50;
+-----+-----+-----+-----+
| productName | buyPrice | quantityInStock | Total Price |
+-----+-----+-----+-----+
| 1969 Harley Davidson Ultimate Chopper | 48.81 | 7933 | 387210 |
| 1936 Harley Davidson El Knucklehead | 24.23 | 4357 | 105570 |
+-----+-----+-----+-----+
2 rows in set (0.001 sec)

```

3. Pada tabel orders terdapat orderdate dan shippeddate. Tampilkan lamanya waktu tunggu dari order hingga pengiriman, untuk status yang barangnya telah dikirim (shipped) dan terjadi pada tahun 2004, urutkan dari bulan order desember.

```
MariaDB [classicmodels]> SELECT orderDate, shippedDate, ABS (orderDate - shippedDate) AS 'Waktu Tunggu'  
-> FROM orders  
-> where status = 'shipped' AND shippedDate BETWEEN '20040101' AND '20041231'  
-> ORDER BY orderDate desc;
```

orderDate	shippedDate	Waktu Tunggu
2004-12-17	2004-12-20	3
2004-12-16	2004-12-18	2
2004-12-15	2004-12-18	3
2004-12-10	2004-12-16	6
2004-12-10	2004-12-14	4
2004-12-09	2004-12-12	3
2004-12-07	2004-12-13	6
2004-12-04	2004-12-05	1
2004-12-04	2004-12-05	1
2004-12-03	2004-12-07	4
2004-12-03	2004-12-09	6
2004-12-02	2004-12-05	3
2004-12-01	2004-12-03	2
2004-11-29	2004-11-30	1
2004-11-29	2004-11-30	1
2004-11-25	2004-11-26	1
2004-11-25	2004-11-29	4
2004-11-24	2004-11-29	5
2004-11-24	2004-11-25	1
2004-11-24	2004-11-29	5
2004-11-24	2004-11-26	2
2004-11-23	2004-11-30	7
2004-11-22	2004-11-27	5
2004-11-21	2004-11-26	5
2004-11-20	2004-11-24	4
2004-11-19	2004-11-23	4
2004-11-18	2004-11-20	2
2004-11-17	2004-11-18	1
2004-11-17	2004-11-23	6
2004-11-16	2004-11-21	5
2004-11-15	2004-11-16	1
2004-11-12	2004-11-18	6
2004-11-09	2004-11-10	1
2004-11-05	2004-11-08	3
2004-11-05	2004-11-08	3
2004-11-05	2004-11-09	4
2004-11-04	2004-11-07	3
2004-11-04	2004-11-10	6
2004-11-03	2004-11-07	4
2004-11-03	2004-11-06	3
2004-11-02	2004-11-07	5

2004-11-02	2004-11-08	6	
2004-11-01	2004-11-05	4	
2004-11-01	2004-11-07	6	
2004-10-29	2004-10-30	1	
2004-10-22	2004-10-23	1	
2004-10-22	2004-10-25	3	
2004-10-21	2004-10-23	2	
2004-10-16	2004-10-18	2	
2004-10-16	2004-10-20	4	
2004-10-15	2004-10-18	3	
2004-10-15	2004-10-20	5	
2004-10-14	2004-10-20	6	
2004-10-14	2004-10-17	3	
2004-10-13	2004-10-15	2	
2004-10-11	2004-10-17	6	
2004-10-06	2004-10-09	3	
2004-09-30	2004-10-01	71	
2004-09-27	2004-10-01	74	
2004-09-16	2004-09-21	5	
2004-09-15	2004-09-16	1	
2004-09-10	2004-09-14	4	
2004-09-10	2004-09-14	4	
2004-09-09	2004-09-14	5	
2004-09-08	2004-09-11	3	
2004-09-08	2004-09-14	6	
2004-09-07	2004-09-13	6	
2004-09-03	2004-09-04	1	
2004-09-01	2004-09-05	4	
2004-08-30	2004-09-01	71	
2004-08-28	2004-09-01	73	
2004-08-27	2004-08-31	4	
2004-08-21	2004-08-26	5	
2004-08-20	2004-08-23	3	
2004-08-20	2004-08-22	2	
2004-08-19	2004-08-23	4	
2004-08-17	2004-08-19	2	
2004-08-09	2004-08-15	6	
2004-08-06	2004-08-09	3	
2004-08-04	2004-08-05	1	
2004-08-02	2004-08-08	6	
2004-07-23	2004-07-29	6	
2004-07-21	2004-07-22	1	
2004-07-21	2004-07-22	1	
2004-07-20	2004-07-22	2	
2004-07-20	2004-07-23	3	
2004-07-19	2004-07-24	5	
2004-07-16	2004-07-18	2	
2004-07-12	2004-07-14	2	
2004-07-07	2004-07-09	2	

2004-07-06	2004-07-10	4
2004-07-02	2004-07-07	5
2004-06-30	2004-07-01	71
2004-06-28	2004-07-02	74
2004-06-17	2004-06-22	5
2004-06-15	2004-06-17	2
2004-06-15	2004-06-23	8
2004-06-14	2004-06-15	1
2004-06-08	2004-06-10	2
2004-06-04	2004-06-09	5
2004-06-03	2004-06-04	1
2004-05-26	2004-05-29	3
2004-05-18	2004-05-24	6
2004-05-11	2004-05-15	4
2004-05-08	2004-05-11	3
2004-05-05	2004-05-08	3
2004-05-05	2004-05-06	1
2004-05-04	2004-05-09	5
2004-04-29	2004-05-04	75
2004-04-26	2004-04-28	2
2004-04-20	2004-04-25	5
2004-04-13	2004-04-19	6
2004-04-13	2004-04-20	7
2004-04-12	2004-04-17	5
2004-04-09	2004-04-10	1
2004-04-05	2004-04-10	5
2004-04-03	2004-04-08	5
2004-04-02	2004-04-06	4
2004-03-30	2004-04-02	72
2004-03-29	2004-04-02	73
2004-03-20	2004-03-25	5
2004-03-19	2004-03-25	6
2004-03-15	2004-03-20	5
2004-03-11	2004-03-12	1
2004-03-10	2004-03-13	3
2004-03-02	2004-03-08	6
2004-02-26	2004-03-02	76
2004-02-22	2004-02-24	2
2004-02-21	2004-02-26	5
2004-02-20	2004-02-24	4
2004-02-19	2004-02-20	1
2004-02-18	2004-02-19	1
2004-02-12	2004-02-16	4
2004-02-10	2004-02-12	2
2004-02-09	2004-02-11	2
2004-02-04	2004-02-06	2
2004-02-02	2004-02-04	2
2004-01-29	2004-02-01	72
2004-01-26	2004-01-29	3

2004-01-22	2004-01-27	5
2004-01-16	2004-01-18	2
2004-01-15	2004-01-18	3
2004-01-12	2004-01-20	8
2004-01-09	2004-01-12	3
2004-01-02	2004-01-04	2

145 rows in set (0.001 sec)

4. MSRP adalah manufacturer suggested retail price atau harga eceran yang disarankan. Total harga masing2 product adalah quantity instock * msrp. Nilai keuntungan bisa didapatkan dari msrp dikurangi dengan buy price. Kira-kira berapa keuntungan yang didapatkan dari penjualan masing-masing product. Tampilkan msrp, quantityinstock, total, keuntungan, productvendor untuk product yang diproduksi oleh Min Lin Diecast. Lakukan pembulatan kebawah.

```
MariaDB [classicmodels]> SELECT MSRP, quantityInStock,  
-> FLOOR (quantityInStock * MSRP) AS 'Total Price',  
-> FLOOR ((MSRP - buyPrice) * quantityInStock) AS 'Profit', productVendor  
-> FROM products  
-> where productVendor LIKE '%Min Lin Diecast%';
```

MSRP	quantityInStock	Total Price	Profit	productVendor
95.70	7933	759188	371978	Min Lin Diecast
142.25	9772	1390067	472573	Min Lin Diecast
62.46	5649	352836	158793	Min Lin Diecast
86.02	4259	366359	146552	Min Lin Diecast
80.41	1917	154145	60117	Min Lin Diecast
143.62	5545	796372	286676	Min Lin Diecast
88.51	9173	811902	381596	Min Lin Diecast
91.34	5841	533516	234749	Min Lin Diecast

8 rows in set (0.001 sec)

5. Profit diperoleh dari selisih msrp dan buyprice. Tampilkan 10 data profit tertinggi untuk produk dengan jenis(productline) Classic cars. Tampilkan productname, productline, msrp, buyprice dan profit. Lakukan pembulatan matematis (round) utk msrp, buyprice dan profit

```
MariaDB [classicmodels]> SELECT productName, productLine,  
-> ROUND (MSRP), ROUND (buyPrice), ROUND (MSRP - buyPrice) AS 'Profit'  
-> FROM products  
-> WHERE productLine LIKE '%Classic Cars%'  
-> ORDER BY (MSRP - buyPrice) DESC limit 10;
```

productName	productLine	ROUND (MSRP)	ROUND (buyPrice)	Profit
1952 Alpine Renault 1300	Classic Cars	214	99	116
2001 Ferrari Enzo	Classic Cars	208	96	112
1968 Ford Mustang	Classic Cars	195	95	99
1992 Ferrari 360 Spider red	Classic Cars	169	78	91
1969 Ford Falcon	Classic Cars	173	83	90
1948 Porsche Type 356 Roadster	Classic Cars	141	62	79
1957 Corvette Convertible	Classic Cars	149	70	79
1999 Indy 500 Monte Carlo SS	Classic Cars	132	57	75
1976 Ford Gran Torino	Classic Cars	147	73	74
1998 Chrysler Plymouth Prowler	Classic Cars	164	102	62

10 rows in set (0.001 sec)

6. Totalprice adalah hasil kali dari Quantityordered dengan priceeach pada tabel orderdetails. Diskon diberikan 10% dari totalprice. Tampilkan ordernumber, productcode, quantityordered, priceeach, total, diskon, payment (total-diskon) untuk 10 data dengan payment tertinggi yang quantityorderednya diantara 20 sampai 30 dan nomor order nya bukan 10420, 10423, 10425 Lakukan pembulatan keatas untuk priceeach dan payment.

```
MariaDB [classicmodels]> SELECT orderNumber, productCode, quantityOrdered, CEIL(priceEach),
-> (quantityOrdered * priceEach) AS 'Total Price',
-> (0.1 * quantityOrdered * priceEach) AS 'Diskon',
-> CEIL ((quantityOrdered * priceEach) - (0.1 * quantityOrdered * priceEach)) AS 'Payment'
-> FROM orderdetails
-> WHERE quantityOrdered BETWEEN '20' AND '30' AND orderNumber NOT IN ('10420','10423','10425')
-> ORDER BY Payment DESC limit 10;
```

orderNumber	productCode	quantityOrdered	CEIL(priceEach)	Total Price	Diskon	Payment
10228	S10_1949	29	215	6214.70	621.470	5594
10112	S10_1949	29	198	5717.64	571.764	5146
10347	S10_1949	30	189	5657.40	565.740	5092
10103	S10_1949	26	215	5571.80	557.180	5015
10342	S18_3232	30	168	5029.50	502.950	4527
10214	S18_1749	30	167	4998.00	499.800	4499
10333	S10_1949	26	189	4903.08	490.308	4413
10169	S12_1099	30	164	4903.20	490.320	4413
10299	S10_4698	29	165	4773.69	477.369	4297
10356	S18_2795	30	159	4758.90	475.890	4284

10 rows in set (0.012 sec)

7. Untuk pelanggan yang melakukan pembayaran pada bulan januari 2003 dari tabel payments, wajib membayar pajak sebanyak 10% dari amount. Tampilkan semua field dan besaran pajak lakukan pembulatan keatas dan urutkan berdasarkan paymentdate.

```
MariaDB [classicmodels]> SELECT *, CEIL (0.1 * amount) AS 'Pajak'
-> FROM payments
-> WHERE paymentDate BETWEEN '20030101' AND '20030131'
-> ORDER BY paymentDate;
```

customerNumber	checkNumber	paymentDate	amount	Pajak
363	IS232033	2003-01-16	10223.83	1023
128	DI925118	2003-01-28	10549.01	1055
181	GQ132144	2003-01-30	5494.78	550

3 rows in set (0.001 sec)

E. REFERENSI

- ♠ [Job Sheet Basis Data.pdf](#)
- ♠ <https://artechinnovation.wordpress.com/2012/11/21/fungsi-dalam-mysql-fungsiaritmatikanumerik/>
- ♠ <http://algoritmamamu.blogspot.com/2017/01/fungsi-numerik-dalam-sql.html>
- ♠ <http://sribututorial.blogspot.com/2016/09/penggunaan-fungsi-numerik-di-mysql-bag1.html>
- ♠ <http://iuminaroh.blogspot.com/2017/09/fungsi-numerik-pda-mysql.html>