

**LAPORAN BASIS DATA TINGKAT LANJUT**  
**STORED PROCEDURE & FUNCTION**



**Oleh :**

Ali Arsa      18615008

**TEKNIK INFORMATIKA**

**TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI SAMARINDA**

**2020**

## DAFTAR ISI

DAFTAR ISI.....	i
Exercise 1: Simple stored procedures.....	1
Exercise 2: Input and Output of stored procedures .....	5
Exercise 3: Simple stored function.....	9
Exercise 4: Built-in function .....	14
Exercise 5: Variables .....	20

## Exercise 1: Simple stored procedures

- 1.0 Pada **simple stored procedures** ini, pertama-tama kita masuk dulu ke interface awal mysql dengan menggunakan perintah **mysql -u root -p**, selanjutnya kita eksekusi file **person\_data.sql** yang ada pada komputer kita yang telah di download melalui classroom dengan perintah **SOURCE C:\..direktori..\person\_data.sql**. File tersebut akan memasukkan **5 record** kedalam table **person**. Lalu, gunakan perintah **SELECT** untuk memastikan 5 record tersebut sudah masuk.

```
C:\xamppbaru\mysql\bin>mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.4.10-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\person_data.sql
Database changed
Query OK, 0 rows affected (0.217 sec)

Query OK, 0 rows affected (0.262 sec)

Query OK, 5 rows affected (0.065 sec)
Records: 5  Duplicates: 0  Warnings: 0

MariaDB [mydb]> SELECT * FROM person;
+-----+-----+-----+-----+
| person_id | first_name | last_name | age |
+-----+-----+-----+-----+
| 1 | sang | shin | 88 |
| 2 | kelly | jones | 22 |
| 3 | jack | kennedy | 56 |
| 4 | paul | kennedy | 34 |
| 5 | daniel | song | 24 |
+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

- 1.1 Gunakan perintah **DELIMITER \$\$** untuk menjadikan karakter **\$\$** untuk dijadikan pernyataan akhir sebuah statement dan agar karakter **;** bisa digunakan didalam procedure. Lalu buat sebuah procedure baru seperti bawah. Disini kita membuat procedure bernama **count\_people**. Selanjutnya kembalikan fungsi karakter **;** agar menjadi seperti semula menggunakan perintah **DELIMITER ;**. Selanjutnya **eksekusi procedure count\_people** tadi menggunakan perintah **CALL count\_people;** tanpa **()**, atau **CALL count\_people();** dengan **()**. Keduanya sama saja karena tidak disebutkan parameternya IN, OUT, ataupun INOUT. Lalu, gunakan perintah **SHOW CREATE PROCEDURE** seperti dibawah, perfungsi untuk menunjukkan detail procedure yang telah kita buat, termasuk isi statement procedure tersebut.

```

MariaDB [mydb]> DELIMITER $$
MariaDB [mydb]> CREATE PROCEDURE count_people()
  -> BEGIN
  -> SELECT COUNT(person_id) FROM person;
  -> END $$
Query OK, 0 rows affected (0.219 sec)

MariaDB [mydb]> DELIMITER ;
MariaDB [mydb]> CALL count_people;
+-----+
| COUNT(person_id) |
+-----+
|                5 |
+-----+
1 row in set (0.045 sec)

Query OK, 0 rows affected (0.051 sec)

MariaDB [mydb]> CALL count_people();
+-----+
| COUNT(person_id) |
+-----+
|                5 |
+-----+
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.007 sec)

MariaDB [mydb]> SHOW CREATE PROCEDURE count_people\G
***** 1. row *****
      Procedure: count_people
      sql_mode: NO_ZERO_IN_DATE,NO_ZERO_DATE,NO_ENGINE_SUBSTITUTION
      Create Procedure: CREATE DEFINER='root'@'localhost' PROCEDURE `count_people`()
BEGIN
SELECT COUNT(person_id) FROM person;
END
character_set_client: cp850
collation_connection: cp850_general_ci
      Database Collation: utf8mb4_general_ci
1 row in set (0.000 sec)

```

- 1.2.1 Selanjutnya eksekusi file **procedures\_school.sql**, file tersebut akan mengeksekusi 2 procedure, **create\_school\_table()** dan **drop\_school\_table()**. Untuk memasikannya, lakukan perintah **SHOW PROCEDURE STATUS** seperti dibawah ini.

```

MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\procedur
es_school.sql
Query OK, 0 rows affected (0.088 sec)

Query OK, 0 rows affected (0.088 sec)

MariaDB [mydb]> SHOW PROCEDURE STATUS LIKE '%school%'\G
***** 1. row *****
      Db: mydb
      Name: create_school_table
      Type: PROCEDURE
      Definer: root@localhost
      Modified: 2020-04-11 10:26:35
      Created: 2020-04-11 10:26:35
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
***** 2. row *****
      Db: mydb
      Name: drop_school_table
      Type: PROCEDURE
      Definer: root@localhost
      Modified: 2020-04-11 10:26:35
      Created: 2020-04-11 10:26:35
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
2 rows in set (0.099 sec)

```

1.2.2 Lalu periksa table yang ada dengan perintah **SHOW TABLES;**, perhatikan daftar table-table tersebut. Lalu lakukan perintah **CALL** untuk procedure **create\_school\_table()**. Dan Periksa table lagi dengan perintah **SHOW TABLES;**. Jika bertambah table **school\_table**. Maka procedure **create\_school\_table()** yang berisi statement pembuatan table **school\_table** berhasil dijalankan.

Selanjutnya, lakukan perintah **CALL** untuk procedure **drop\_school\_table()** dan periksa kembali table-tablenya. Jika table **school\_table** hilang, maka procedure **drop\_school\_table()** yang berisi statement penghapusan table **school\_table** berhasil dijalankan.

```

MariaDB [mydb]> SHOW TABLES;
+-----+
| Tables_in_mydb |
+-----+
| checking        |
| employees       |
| person          |
| savings         |
| student         |
| transaction1_innodb |
| transaction2_myisam |
+-----+
7 rows in set (0.001 sec)

MariaDB [mydb]> CALL create_school_table();
Query OK, 0 rows affected (0.292 sec)

MariaDB [mydb]> SHOW TABLES;
+-----+
| Tables_in_mydb |
+-----+
| checking        |
| employees       |
| person          |
| savings         |
| school_table    |
| student         |
| transaction1_innodb |
| transaction2_myisam |
+-----+
8 rows in set (0.001 sec)

MariaDB [mydb]> CALL drop_school_table();
Query OK, 0 rows affected (0.188 sec)

MariaDB [mydb]> SHOW TABLES;
+-----+
| Tables_in_mydb |
+-----+
| checking        |
| employees       |
| person          |
| savings         |
| student         |
| transaction1_innodb |
| transaction2_myisam |
+-----+
7 rows in set (0.001 sec)

```

## Exercise 2: Input and Output of stored procedures

- 2.1.1 Eksekusi file **procedures\_person\_in.sql**, file tersebut berisi 2 procedure, **get\_peron()** dan **get\_person2()**. Lalu periksa untuk memasikan procedure tersebut telah di eksekusi dengan perintah **SHOW PROCEDURE STATUS** seperti dibawah dengan spesifikasi khusus hanya yang mengandung nama **get\_person**. Jika muncul 2 detail procedure, artinya sudah berhasil.

```
MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\procedures_person_in.sql
Query OK, 0 rows affected (0.100 sec)

Query OK, 0 rows affected (0.088 sec)

MariaDB [mydb]> SHOW PROCEDURE STATUS LIKE '%get_person%'\G
***** 1. row *****
      Db: mydb
      Name: get_person
      Type: PROCEDURE
      Definer: root@localhost
      Modified: 2020-04-11 10:38:15
      Created: 2020-04-11 10:38:15
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
***** 2. row *****
      Db: mydb
      Name: get_person2
      Type: PROCEDURE
      Definer: root@localhost
      Modified: 2020-04-11 10:38:15
      Created: 2020-04-11 10:38:15
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
2 rows in set (0.029 sec)
```

- 2.1.2 Selanjutnya panggil procedure **get\_person** dengan atribut didalamnya berangka 2, yang artinya berdasarkan statement procedure tersebut adalah **id dari person**, menggunakan perintah **CALL** seperti dibawah. Jika muncul data **person dengan person\_id nomor 2**, artinya procedure berhasil dipanggil.

Lalu lakukan pemanggilan lagi seperti sebelumnya untuk procedure **get\_person2** dengan atribut didalamnya berangka 2 lalu 30, yang artinya berdasarkan statement procedure tersebut adalah **angka pertama merupakan id lebih dari dan angka kedua, age lebih dari**. Jika muncul data seperti dibawah, maka pemanggilan procedure telah berhasil.

```
MariaDB [mydb]> CALL get_person(2);
+-----+-----+-----+-----+
| person_id | first_name | last_name | age |
+-----+-----+-----+-----+
|          2 | kelly      | jones     | 22  |
+-----+-----+-----+-----+
1 row in set (0.001 sec)

Query OK, 0 rows affected (0.008 sec)

MariaDB [mydb]> CALL get_person2(2, 30);
+-----+-----+-----+-----+
| person_id | first_name | last_name | age |
+-----+-----+-----+-----+
|          3 | jack      | kennedy   | 56  |
|          4 | paul      | kennedy   | 34  |
+-----+-----+-----+-----+
2 rows in set (0.007 sec)

Query OK, 0 rows affected (0.019 sec)
```

- 2.2 Berikutnya, eksekusi file **procedures\_person\_out.sql**. File tersebut berisi procedure dengan statemaent menampilkan **first\_name** melalui **id** menggunakan parameter pada nama procedure. Lalu tampilkan status procedure yang mengandung nama **get\_person\_name** menggunakan perintah seperti dibawah.

```
MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\procedur
es_person_out.sql
Query OK, 0 rows affected (0.141 sec)

MariaDB [mydb]> SHOW PROCEDURE STATUS LIKE '%get_person_name%'\G
***** 1. row *****
               Db: mydb
              Name: get_person_name
              Type: PROCEDURE
             Definer: root@localhost
            Modified: 2020-04-11 10:45:10
             Created: 2020-04-11 10:45:10
          Security_type: DEFINER
             Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
1 row in set (0.058 sec)
```



Lalu lakukan pemanggilan pada procedure **get\_person\_name(3, @myname)**. Angka **3** sebagai **id**, dan **@myname** sebagai pengganti **f\_name**. Dapat dilihat pada file sql tadi. Selanjutnya tampilan **@myname**; maka akan tampil **data f\_name** sebagai **@myname**. Namun yang tampil hanya orang dengan id nomor 3, karena pada pemanggilan procedure tadi hanya memanggil orang dengan id nomor 3.

```
MariaDB [mydb]> CALL get_person_name(3, @myname);
Query OK, 1 row affected (0.001 sec)

MariaDB [mydb]> SELECT @myname;
+-----+
| @myname |
+-----+
| jack    |
+-----+
1 row in set (0.000 sec)
```

- 2.3 Eksekusi file **procedures\_compute\_inout.sql**, file tersebut berisi procedure berparameter **INOUT number INT** dengan statement **number dikali number** dan akan **dijadikan number lagi**. Lalu tampilkan status procedure yang mengandung kata **compute**.

```
MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\procedures_compute_inout.sql
Query OK, 0 rows affected (0.160 sec)

MariaDB [mydb]> SHOW PROCEDURE STATUS LIKE '%compute%'\G
***** 1. row *****
               Db: mydb
               Name: compute_square
               Type: PROCEDURE
               Definer: root@localhost
               Modified: 2020-04-11 10:47:34
               Created: 2020-04-11 10:47:34
               Security_type: DEFINER
               Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
1 row in set (0.021 sec)
```

Berikutnya, **atur nilai sebuah variable**, disini diberi nama **var** dengan **nilai 7**. Lalu lakukan pemanggilan procedure **compute\_square(@var)**; yang artinya kita memanggil procedure tersebut dengan nilai dari **@var** sebagai **number**. Jika sudah, tampilkan **@var** yang menjadi pengganti number. Jika berhasil maka nilainya akan menjadi **49**. Karena nilai **var** adalah **7**, lalu sesuai dengan statement procedure, **7 x 7** maka **nilainya akan menjadi 49**.

```
MariaDB [mydb]> SET @var=7;
Query OK, 0 rows affected (0.000 sec)

MariaDB [mydb]> CALL compute_square(@var);
Query OK, 1 row affected (0.055 sec)

MariaDB [mydb]> SELECT @var;
+-----+
| @var |
+-----+
|    49 |
+-----+
1 row in set (0.000 sec)
```

### Exercise 3: Simple stored functions

- 3.1 Lakukan eksekusi file **functions\_compute.sql**. File tersebut berisi procedure yang berisi fungsi perhitungan, perhitungan luas segiempat pada procedure **compute\_square\_function**, dan perhitungan luas lingkaran pada procedure **compute\_circle\_area**. Lalu tampilkan status function yang mengandung kata **compute** dengan perintah seperti dibawah.

```
MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\function
s_compute.sql
Query OK, 0 rows affected (0.119 sec)

Query OK, 0 rows affected (0.143 sec)

MariaDB [mydb]> SHOW FUNCTION STATUS LIKE '%compute%'\G
***** 1. row *****
      Db: mydb
      Name: compute_circle_area
      Type: FUNCTION
      Definer: root@localhost
      Modified: 2020-04-11 10:59:15
      Created: 2020-04-11 10:59:15
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
***** 2. row *****
      Db: mydb
      Name: compute_square_function
      Type: FUNCTION
      Definer: root@localhost
      Modified: 2020-04-11 10:59:15
      Created: 2020-04-11 10:59:15
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
2 rows in set (0.038 sec)
```

Lalu tampilkan data procedure **compute\_square\_function** dengan nilai sisi **3**. Maka procedure akan melakukan fungsi perhitungan **sisi \* sisi** yang pada procedure kali ini menggunakan number sebagai pengganti sisi. Jadi **3\*3=9**. Maka data procedure dengan nilai 3 akan menampilkan **angka 9**.

Begitu juga dengan procedure **compute\_circle\_area**, tampilkan dengan nilai **3** sebagai **radius atau jari-jarinya**. Maka perhitungannya akan menjadi **pi\*3\*3=28.2743**

```

MariaDB [mydb]> SELECT compute_square_function(3);
+-----+
| compute_square_function(3) |
+-----+
| 9 |
+-----+
1 row in set (0.001 sec)

MariaDB [mydb]> SELECT compute_circle_area(3);
+-----+
| compute_circle_area(3) |
+-----+
| 28.2743 |
+-----+
1 row in set (0.053 sec)

```

Lalu lakukan pemanggilan procedure **compute\_square\_function(3);**, disini akan terjadi error karena procedure tersebut tidak ada.

```

MariaDB [mydb]> CALL compute_square_function(3);
ERROR 1305 (42000): PROCEDURE mydb.compute_square_function does not exist

```

Selanjutnya kita akan **menghapus kedua function** tadi menggunakan perintah **DROP FUNCTION namafunction;** Maka function akan terhapus.

```

MariaDB [mydb]> DROP FUNCTION compute_square_function;
Query OK, 0 rows affected (0.121 sec)

MariaDB [mydb]> DROP FUNCTION compute_circle_area;
Query OK, 0 rows affected (0.101 sec)

```

- 3.2 Eksekusi file **functions\_date.sql**, yang berisi 2 procedure dengan fungsi **menampilkan tanggal** dan **menampilkan waktu**. Lalu tampilkan status function yang mengandung kata **today**. Maka status kedua function tadi akan ditampilkan.

```

MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\function
s_date.sql
Query OK, 0 rows affected (0.097 sec)

Query OK, 0 rows affected (0.088 sec)

MariaDB [mydb]> SHOW FUNCTION STATUS LIKE '%today%'\G
***** 1. row *****
      Db: mydb
      Name: today_date
      Type: FUNCTION
      Definer: root@localhost
      Modified: 2020-04-11 11:02:37
      Created: 2020-04-11 11:02:37
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
***** 2. row *****
      Db: mydb
      Name: today_time
      Type: FUNCTION
      Definer: root@localhost
      Modified: 2020-04-11 11:02:37
      Created: 2020-04-11 11:02:37
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
2 rows in set (0.059 sec)

```

Lakukan perintah **SELECT** untuk menampilkan fungsi kedua procedure seperti dibawah ini. Pada function **today\_date** akan muncul **tanggal hari ini**, sedangkan pada fungsi **today\_time** akan muncul **waktu saat ini**.

```

MariaDB [mydb]> SELECT today_date();
+-----+
| today_date() |
+-----+
| Saturday April 2020 |
+-----+
1 row in set (0.060 sec)

MariaDB [mydb]> SELECT today_time();
+-----+
| today_time() |
+-----+
| 11:03:41 |
+-----+
1 row in set (0.000 sec)

```

- 3.3 Eksekusi file **functions\_table.sql** yang berisi 2 procedure yang pertama berisi statement **menambahkan data** kedalam table **person**, yang kedua berisi statement **menghapus data tersebut berdasarkan idnya**. Jika sudah di eksekusi, tampilkan status fungsi yang mengandung kata **insert\_person**, selanjutnya yang mengandung kata **delete\_person**.

```
MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\function
s_table.sql
Query OK, 0 rows affected (0.090 sec)

Query OK, 0 rows affected (0.122 sec)

MariaDB [mydb]> SHOW FUNCTION STATUS WHERE Name='insert_person'\G
***** 1. row *****
      Db: mydb
      Name: insert_person
      Type: FUNCTION
      Definer: root@localhost
      Modified: 2020-04-11 11:04:22
      Created: 2020-04-11 11:04:22
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
1 row in set (0.026 sec)

MariaDB [mydb]> SHOW FUNCTION STATUS WHERE Name='delete_person'\G
***** 1. row *****
      Db: mydb
      Name: delete_person
      Type: FUNCTION
      Definer: root@localhost
      Modified: 2020-04-11 11:04:22
      Created: 2020-04-11 11:04:22
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
1 row in set (0.021 sec)
```

Tampilkan fungsi **insert\_person** lengkap dengan paramater sesuai data yang ingin diisi, seperti dibawah contohnya **131 sebagai id, lily sebagai f\_name, maria sebagai last\_name, 33 sebagai age**. Maka akan tampil 1 yang artinya data tersebut berjumlah 1. Tampilkan data pada table person untuk memastikan data tersebut sudah masuk atau belum.

Selanjutnya jalankan perintah menampilkan fungsi **delete\_person** dengan parameter **3**. Yang artinya kita akan menjalankan fungsi hapus untuk id nomor 3, dimana orang yang memiliki id nomor 3 bernama **jack kennedy**. Jika sudah, tampilkan data table person untuk memastikan data tersebut sudah berhasil dihapus.

```
MariaDB [mydb]> SELECT insert_person(131, 'lily', 'maria', 33);
```

insert_person(131, 'lily', 'maria', 33)
1

```
1 row in set (0.104 sec)
```

```
MariaDB [mydb]> SELECT * FROM person;
```

person_id	first_name	last_name	age
1	sang	shin	88
2	kelly	jones	22
3	jack	kennedy	56
4	paul	kennedy	34
5	daniel	song	24
131	lily	maria	33

```
6 rows in set (0.000 sec)
```

```
MariaDB [mydb]> SELECT delete_person(3);
```

delete_person(3)
1

```
1 row in set (0.080 sec)
```

```
MariaDB [mydb]> SELECT * FROM person;
```

person_id	first_name	last_name	age
1	sang	shin	88
2	kelly	jones	22
4	paul	kennedy	34
5	daniel	song	24
131	lily	maria	33

```
5 rows in set (0.000 sec)
```

## Exercise 4: Built-in functions

- 4.1 Perintah pertama adalah menampilkan data **rata-rata** usia(age) pada table person menggunakan fungsi **AVG**.  
Perintah kedua adalah menampilkan data **nilai terbesar** usia(age) pada table person menggunakan fungsi **MAX**, dan nilai **terkecil** menggunakan fungsi **MIN**.  
Perintah ketiga adalah menampilkan **banyaknya data atau banyaknya baris data** pada table person menggunakan fungsi **COUNT**.  
Perintah keempat adalah menampilkan **jumlah** usia(age) pada table person jika ditotal menggunakan fungsi **SUM**.

```
MariaDB [mydb]> SELECT AVG(age) average_age FROM person;
+-----+
| average_age |
+-----+
|      40.2000 |
+-----+
1 row in set (0.046 sec)

MariaDB [mydb]> SELECT MAX(age) max_age, MIN(age) min_age FROM person;
+-----+-----+
| max_age | min_age |
+-----+-----+
|      88 |      22 |
+-----+-----+
1 row in set (0.014 sec)

MariaDB [mydb]> SELECT COUNT(*) AS total_person FROM person;
+-----+
| total_person |
+-----+
|           5 |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT SUM(age) AS sum_age FROM person;
+-----+
| sum_age |
+-----+
|      201 |
+-----+
1 row in set (0.000 sec)
```



- 4.2 Perintah pertama terdapat 2 fungsi, yaitu fungsi **CEILING** untuk **pembulatan ke atas**, dan fungsi **FLOOR** untuk pembulatan **kebawah**.  
Perintah kedua dengan fungsi **ABS** adalah fungsi untuk merubah nilai sebuah data menjadi **nilai absolut**, yaitu bilangan riil atau bilangan tanpa plus minus.  
Perintah ketiga dengan fungsi **PI** adalah untuk menampilkan nilai dari **PI**.  
Perintah keempat dengan fungsi **RAND** adalah fungsi untuk **membuat sebuah nilai acak**.  
Perintah kelima dengan fungsi **SQRT** adalah fungsi untuk menghitung **akar kuadrat** dari suatu nilai.  
Perintah keenam dengan fungsi **POWER** adalah fungsi untuk menghitung **pemangkatan**.  
Perintah ketujuh dengan fungsi **ROUND** adalah fungsi untuk **mengatur secara spesifik penjang sebuah nilai sebelum atau setelah koma**. Jika nilainya positif maka nilai setelah koma akan tampil, jika negatif maka nilai sebelum koma akan menjadi angka 0.

```

MariaDB [mydb]> SELECT CEILING(210.21*2) AS ceiling_value, FLOOR(5.4) AS floor_value;
+-----+-----+
| ceiling_value | floor_value |
+-----+-----+
|          421 |          5 |
+-----+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT ABS(-20) as arg1,ABS(30) AS arg2;
+-----+-----+
| arg1 | arg2 |
+-----+-----+
|    20 |    30 |
+-----+-----+
1 row in set (0.011 sec)

MariaDB [mydb]> SELECT PI() pi;
+-----+
| pi      |
+-----+
| 3.141593 |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT RAND() AS random_value;
+-----+
| random_value      |
+-----+
| 0.39219171962904903 |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT SQRT(100) AS sqrt_of_100;
+-----+
| sqrt_of_100 |
+-----+
|          10 |
+-----+
1 row in set (0.014 sec)

MariaDB [mydb]> SELECT POWER(10,3) AS 10_power_of_3;
+-----+
| 10_power_of_3 |
+-----+
|          1000 |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT ROUND(12.019 ,1) positive_rounded_value,
-> ROUND(314.239 ,-1) negative_rounded_value;
+-----+-----+
| positive_rounded_value | negative_rounded_value |
+-----+-----+
|          12.0 |          310 |
+-----+-----+
1 row in set (0.000 sec)

```

- 4.3 Perintah pertama dengan fungsi **LENGTH** didalam fungsi **MAX** digunakan untuk **menghitung panjang karakter terbanyak** dari first\_name.
- Perintah kedua dengan fungsi **CONCAT** berfungsi untuk **menggabungkan lebih dari 1 string menjadi 1 string didalam satu kolom yang sama**.
- Perintah ketiga dengan fungsi **UPPER** berguna untuk merubah string didalamnya menjadi huruf **KAPITAL**, dan fungsi **LOWER** berguna untuk merubah string didalamnya menjadi huruf **KECIL**.
- Perintah keempat dengan fungsi **REPLACE** berfungsi untuk **menimpa atau mengganti sebuah string dengan string yang baru**.
- Perintah kelima dengan fungsi **SUBSTRING** berfungsi untuk **membaca string dimulai dari karakter keberapa yang telah ditentukan**. Jadi karakter sebelumnya tidak ditampilkan.
- Perintah keenam dengan fungsi **SUBSTRING** juga sama, namun parameter selanjutnya berisi nilai jumlah **karakter yang ingin ditampilkan**. Maka, karakter setelahnya tidak akan ditampilkan.
- Perintah ketujuh dengan fungsi **ASCII** berfungsi untuk **menemukan nilai numerik dari sebuah karakter** yang mana nilai tersebut memiliki range mulai dari 0-255.
- Perintah ketujuh dengan fungsi **CHAR** berfungsi untuk **menemukan nilai karakter dari sebuah nilai numerik**. Fungsi ini adalah kebalikan dari fungsi ASCII.

```

MariaDB [mydb]> SELECT MAX(LENGTH(first_name)) AS longest_firtsname FROM person;
+-----+
| longest_firtsname |
+-----+
| 6 |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT CONCAT(first_name,':',age) AS firstname_age FROM person;
+-----+
| firstname_age |
+-----+
| sang:88 |
| kelly:22 |
| paul:34 |
| daniel:24 |
| lily:33 |
+-----+
5 rows in set (0.000 sec)

MariaDB [mydb]> SELECT UPPER('Bill'), LOWER('Jones');
+-----+-----+
| UPPER('Bill') | LOWER('Jones') |
+-----+-----+
| BILL | jones |
+-----+-----+
1 row in set (0.054 sec)

MariaDB [mydb]> SELECT REPLACE('life_is bad','bad','good') AS replaced_life;
+-----+
| replaced_life |
+-----+
| life_is good |
+-----+
1 row in set (0.000 sec)

```

```

MariaDB [mydb]> SELECT SUBSTRING('Life is worth living with Passion!',10) AS my_substring;
+-----+
| my_substring |
+-----+
| orth living with Passion! |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT SUBSTRING('Life is worth living with Passion!',10,7) AS my_substring2;
+-----+
| my_substring2 |
+-----+
| orth li      |
+-----+
1 row in set (0.000 sec)

MariaDB [mydb]> SELECT ASCII('a') AS lower_case_a,
-> ASCII('A') AS upper_case_A;
+-----+-----+
| lower_case_a | upper_case_A |
+-----+-----+
|          97 |          65 |
+-----+-----+
1 row in set (0.041 sec)

MariaDB [mydb]> SELECT CHAR(65) AS ch_65,
-> CHAR(97) AS ch_97;
+-----+-----+
| ch_65 | ch_97 |
+-----+-----+
| A     | a     |
+-----+-----+
1 row in set (0.092 sec)

```

## Exercise 5: Variables

- 5.1 Eksekusi file **variables\_in\_procedures.sql** yang berisi 2 procedure bernama **declare\_variables** dan **compute\_something\_with\_variables**. Lalu tampilkan status like yang mengandung kata **variables**. Akan terdapat 2 baris status procedure.

```
MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\variables_in_procedures.sql
Query OK, 0 rows affected (0.094 sec)

Query OK, 0 rows affected (0.088 sec)

MariaDB [mydb]> SHOW PROCEDURE STATUS LIKE '%variables%'\G
***** 1. row *****
      Db: mydb
      Name: compute_something_with_variables
      Type: PROCEDURE
      Definer: root@localhost
      Modified: 2020-04-11 11:20:51
      Created: 2020-04-11 11:20:51
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
***** 2. row *****
      Db: mydb
      Name: declare_variables
      Type: PROCEDURE
      Definer: root@localhost
      Modified: 2020-04-11 11:20:51
      Created: 2020-04-11 11:20:51
      Security_type: DEFINER
      Comment:
character_set_client: cp850
collation_connection: cp850_general_ci
  Database Collation: utf8mb4_general_ci
2 rows in set (0.022 sec)
```

Lakukan pemanggilan procedure **compute\_something\_with\_variables** dengan nilai **4**. Maka procedure tersebut akan menghitung **4\*9=36**. **4** didapat dari variable **counter** yang kita inputkan tadi, sedangkan **9** didapatkan dari **nilai default my\_value** yang telah diatur pada statement procedure tersebut. Selanjutnya lakukan pemanggilan yang sama namun dengan nilai **5**. Maka hasilnya akan menjadi **45** karena menurut perhitungannya adalah **5x9=45**.

```

MariaDB [mydb]> CALL compute_something_with_variables(4);
+-----+
| @counter * my_value |
+-----+
|          36 |
+-----+
1 row in set (0.001 sec)

Query OK, 0 rows affected (0.003 sec)

MariaDB [mydb]> CALL compute_something_with_variables(5);
+-----+
| @counter * my_value |
+-----+
|          45 |
+-----+
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.002 sec)

```

- 5.2 Eksekusi file **variables\_in\_function.sql** yang berisi fungsi untuk **menghitung luas lingkaran**, bedanya dengan procedure luas lingkaran pada exercise sebelumnya adalah, pada kali ini **pi** dijadikan sebuah **variable** bernama **my\_pi**. Tampilkan procedure **compute\_circle\_area2** dengan nilai **3**. Maka akan tampil **28.2743**. Hasil tersebut didapat dari perhitungan **pi\*3\*3=28.2743**.

```

MariaDB [mydb]> SOURCE C:\Users\acer\Downloads\mysql_procedures\sqlscripts\variables_in_functions.sql
Query OK, 0 rows affected (0.118 sec)

```

```

MariaDB [mydb]> SELECT compute_circle_area2(3);
+-----+
| compute_circle_area2(3) |
+-----+
|          28.2743 |
+-----+
1 row in set (0.000 sec)

```