

ZADANIE 1

The screenshot shows the MySQL Workbench interface. The main window displays the execution results of a stored procedure named 'UDZIELONE_POZYCZKI'. The results show a single row with the value 9500 for the column SUMA_POZYCZEK. The 'Edit routine' dialog is open, showing the details of the 'UDZIELONE_POZYCZKI' procedure. The routine is a PROCEDURE with two parameters: 'nazwa_fi' (VARCHAR, 40) and 'SUMA_P' (INT). The definition of the procedure is as follows:

```
1 BEGIN
2   SELECT SUM POZYCZKI.kwota INTO SUMA_POZYCZEK
3   FROM POZYCZKI
4   WHERE POZYCZKI.firma = nazwa_firma;
5 END
```

The 'Edit routine' dialog also shows the following options:

- Is deterministic: ☐
- Adjust privileges: ☒
- Definer: 'kubasz' '@' '%'
- Security type: DEFINER
- SQL data access: NO SQL
- Comment: (empty)

ZADANIE 2

The screenshot shows the MySQL Workbench interface. The main window displays the execution results of a stored function named 'UDZIELONE_KREDYTY'. The results show a single row with the value 9500 for the column UDZIELONE_KREDYTY. The 'Edit routine' dialog is open, showing the details of the 'UDZIELONE_KREDYTY' function. The routine is a FUNCTION with one parameter: 'nazwa_firma' (VARCHAR, 40). The return type is INT. The definition of the function is as follows:

```
1 BEGIN
2   DECLARE suma INT DEFAULT 0;
3   SELECT SUM POZYCZKI.kwota INTO suma
4   FROM POZYCZKI
5   WHERE POZYCZKI.firma = nazwa_firma;
6   RETURN (suma);
7 END
```

The 'Edit routine' dialog also shows the following options:

- Is deterministic: ☐
- Adjust privileges: ☒
- Definer: 'kubasz' '@' '%'
- Security type: DEFINER
- SQL data access: NO SQL
- Comment: (empty)

ZADANIE 3

Server: localhost:3306 Database: zsl

Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0026 seconds.)

```
CALL KLASYFIKUJ("BOGACKI", @bogacki);
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CALL KLASYFIKUJ("PROWIDENT", @prowident);
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0009 seconds.)

```
CALL KLASYFIKUJ("BOCIEK", @bociek);
```

[Edit inline] [Edit] [Create PHP code]

✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

```
SELECT @bogacki AS BOGACKI, @prowident AS PROWIDENT, @bociek AS BOCIEK
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

| BOGACKI | PROWIDENT | BOCIEK |
|---------|-----------|--------|
| plotka | rekin | rekin |

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Cancel

Edit routine

Details

Routine name: KLASYFIKUJ

Type: PROCEDURE

| Direction | Name | Type | Length/Values | Options |
|-----------|----------|---------|---------------|--------------|
| IN | NAZWA_F | VARCHAR | 40 | Charset Drop |
| OUT | REZULTAT | VARCHAR | 10 | Charset Drop |

Add parameter

Definition

```
1 BEGIN
2
3 DECLARE suma INT DEFAULT 0;
4 SET suma = UDZIELONE_KREDYTY(NAZWA_FIRMY);
5
6 IF suma > 8000 THEN
7     SET REZULTAT = "rekin";
8 ELSE
9     SET REZULTAT = "plotka";
10 END IF;
11
12 END
```

Is deterministic: ☐

Adjust privileges: ☒

Definer: 'kubasz'@'%'

Security type: DEFINER

SQL data access: NO SQL

Comment:

Go Close

ZADANIE 4

Server: localhost:3306 • Database: zsl

✓ Your SQL query has been executed successfully.

```
SET @p0='500'; SET @p1='15'; SELECT 'OPROCENTOWANIE' (@p0, @p1) AS 'OPROCENTOWANIE';
```

Execution results of routine 'OPROCENTOWANIE'

| Name | Action | Type | Returns |
|--------------------|--|-----------|--------------|
| KLASYFIKUJ | Edit Execute Export Drop | PROCEDURE | |
| UDZIELONE_KREDYTY | Edit Execute Export Drop | FUNCTION | int |
| UDZIELONE_POZYCZKI | Edit Execute Export Drop | PROCEDURE | |
| OPROCENTOWANIE | Edit Execute Export Drop | FUNCTION | decimal(3,1) |

New

[Add routine](#)

Edit routine

Details

Routine name: OPROCENTOWANIE

Type: FUNCTION

| Parameters | Name | Type | Length/Values | Options |
|------------|------------|------|---------------|----------------------|
| 1 | KWOTA_POZ | INT | | Drop |
| 2 | OKRES_SPLA | INT | | Drop |

[Add parameter](#)

Return type: DECIMAL

Return length/values: 3,1

Return options:

Definition:

```
1 BEGIN
2   IF KWOTA_POZYCZKI <= 1000 THEN
3     IF OKRES_SPLATY <= 12 THEN
4       RETURN 10.0;
5     ELSEIF OKRES_SPLATY <= 36 THEN
6       RETURN 9.0;
7     ELSE
8       RETURN 8.5;
9   END IF;
10  ELSEIF KWOTA_POZYCZKI <= 2000 THEN
11    IF OKRES_SPLATY <= 12 THEN
12      RETURN 9.5;
13    ELSEIF OKRES_SPLATY <= 36 THEN
14      RETURN 9.0;
15    ELSE
16      RETURN 8.0;
17  END IF;
18 END
```

Is deterministic: ☐

Adjust privileges: ☒

Definer: 'kubasz'@'%'

Security type: DEFINER

SQL data access: NO SQL

Comment:

[Go](#) [Close](#)

ZADANIE 5

Server: localhost:3306 • Database: zsl

✓ Your SQL query has been executed successfully.

0 rows affected by the last statement inside the procedure.

```
SET @p0='1'; SET @p1='Bociek'; SET @p2='1234'; SET @p3='16'; CALL 'NOWA_POZYCZKA' (@p0, @p1, @p2, @p3);
```

Execution results of routine 'NOWA_POZYCZKA'

MySQL returned an empty result set (i.e. zero rows).

Routines

| Name | Action | Type | Returns |
|--------------------|--|-----------|--------------|
| KLASYFIKUJ | Edit Execute Export Drop | PROCEDURE | |
| OPROCENTOWANIE | Edit Execute Export Drop | FUNCTION | decimal(3,1) |
| UDZIELONE_KREDYTY | Edit Execute Export Drop | FUNCTION | int |
| UDZIELONE_POZYCZKI | Edit Execute Export Drop | PROCEDURE | |
| NOWA_POZYCZKA | Edit Execute Export Drop | PROCEDURE | |

New

[Add routine](#)

Edit routine

Details

Routine name: NOWA_POZYCZKA

Type: PROCEDURE

| Parameters | Direction | Name | Type | Length/Values | Options |
|------------|-----------|----------|------|---------------|----------------------|
| 1 | IN | ID_KLIEN | INT | | Drop |
| 2 | IN | FIRMA | VARC | 40 | Drop |
| 3 | IN | KWOTA | INT | | Drop |
| 4 | IN | OKRES_S | INT | | Drop |

[Add parameter](#)

Definition:

```
1 BEGIN
2   DECLARE id INT DEFAULT 0;
3   DECLARE procent DECIMAL(3,1);
4   SELECT MAX(id_posyczki) INTO id FROM POZYCZKI;
5   SET procent = OPROCENTOWANIE(KWOTA, OKRES_SPLATY);
6   INSERT INTO POZYCZKI(id_posyczki, id_klienta, firma, kwota,
7     okres_splaty, oprocentowanie) VALUES(id+1, ID_KLIENTA, FIRMA,
8     KWOTA, OKRES_SPLATY, procent);
9 END
```

Is deterministic: ☐

Adjust privileges: ☒

Definer: 'kubasz'@'%'

Security type: DEFINER

SQL data access: NO SQL

Comment:

[Go](#) [Close](#)

```

DELIMITER $$
CREATE DEFINER=`kubasz`@`%` PROCEDURE `KLASYFIKUJ`(IN `NAZWA_FIRMY` VARCHAR(40), OUT `REZULTAT` VARCHAR(10))
    NO SQL
BEGIN

DECLARE suma INT DEFAULT 0;
SET suma = UDZIELONE_KREDYTY(NAZWA_FIRMY);

IF suma > 8000 THEN
    SET REZULTAT = "rekin";
ELSE
    SET REZULTAT = "plotka";
END IF;

END$$
DELIMITER ;

```

```

DELIMITER $$
CREATE DEFINER=`kubasz`@`%` PROCEDURE `UDZIELONE_POZYCZKI`(IN `nazwa_firma` VARCHAR(40), OUT `SUMA_POZYCZEK` INT)
    NO SQL
BEGIN
    SELECT SUM(POZYCZKI.kwota) INTO SUMA_POZYCZEK
    FROM POZYCZKI
    WHERE POZYCZKI.firma = nazwa_firma;
END$$
DELIMITER ;

```

```

DELIMITER $$
CREATE DEFINER=`kubasz`@`%` FUNCTION `OPROCENTOWANIE`(`KWOTA_POZYCZKI`
INT, `OKRES_SPLATY` INT) RETURNS decimal(3,1)
    NO SQL
BEGIN

IF KWOTA_POZYCZKI <= 1000 THEN
    IF OKRES_SPLATY <= 12 THEN
        RETURN(10.0);
    ELSEIF OKRES_SPLATY <= 36 THEN
        RETURN(9.0);
    ELSE
        RETURN(8.5);
    END IF;
ELSEIF KWOTA_POZYCZKI <= 2000 THEN
    IF OKRES_SPLATY <= 12 THEN
        RETURN(9.5);
    ELSEIF OKRES_SPLATY <= 36 THEN
        RETURN(8.0);
    ELSE
        RETURN(7.0);
    END IF;
END IF;

END$$
DELIMITER ;

```

```

DELIMITER $$
CREATE DEFINER=`kubasz`@`%` FUNCTION `UDZIELONE_KREDYTY`(`nazwa_firmy`
VARCHAR(40)) RETURNS int
    NO SQL
BEGIN

DECLARE suma INT DEFAULT 0;

SELECT SUM(POZYCZKI.kwota) INTO suma
FROM POZYCZKI
WHERE POZYCZKI.firma = nazwa_firmy;

RETURN(suma);

END$$
DELIMITER ;

```

```
DELIMITER $$

CREATE DEFINER=`kubasz`@`%` PROCEDURE `NOWA_POZYCZKA`(IN `ID_KLIENTA` INT, IN `FIRMA` VARCHAR(40), IN `KWOTA` INT, IN `OKRES_SPLATY` INT)
    NO SQL
BEGIN

DECLARE id INT DEFAULT 0;
DECLARE procent DECIMAL(3,1);

SELECT MAX(id_pozyczki) INTO id FROM POZYCZKI;
SET procent = OPROCENTOWANIE(KWOTA, OKRES_SPLATY);

INSERT INTO POZYCZKI(id_pozyczki, id_klienta, firma, kwota, okres_splat
y, oprocentowanie) VALUES (id+1, ID_KLIENTA, FIRMA, KWOTA, OKRES_SPLATY
, procent);

END$$
DELIMITER ;
```