

Dodatek do procedur

Procedura jest podprogramem utworzonym w SQL, która zawiera kod SQL oraz kod sterujący jej wykonaniem.

Procedury są przydatne wtedy, gdy chcemy wykonać jakieś często powtarzające się operacje.

Ogólna postać:

```
CREATE PROCEDURE nazwa_procedury (IN zmienna typ, OUT zmienna typ)
BEGIN
    -- polecenia i kod SQL
END
```

Przykład użycia:

```
DELIMITER //

CREATE PROCEDURE przecena (IN nazwa_towaru VARCHAR(50))
BEGIN
    UPDATE produkt
    SET cena = cena * 0.9
    WHERE nazwa_produkту = nazwa_towaru;
END //

DELIMITER ;

call przecena('Pan Tadeusz');
```

Przykład 1:

Na bazie danych na tabeli pracownik:

```
CREATE TABLE `pracownik` (
  `id_pracownik` int(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  `imie` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `nazwisko` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `zawod` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `pensja` float NOT NULL
);

Na danych:

INSERT INTO `pracownik` (`id_pracownik`, `imie`, `nazwisko`, `zawod`, `pensja`) VALUES
(1, 'Jan', 'Kowalski', 'nauczyciel', 4200),
```

(2, 'Ewa', 'Nowak', 'lekarz', 9300),
 (3, 'Edyta', 'Mach', 'nauczyciel', 5300),
 (4, 'Ewelina', 'Koniec', 'lekarz', 11200),
 (5, 'Justyna', 'Baran', 'kasjer', 4700),
 (6, 'Adam', 'Malec', 'kasjer', 6200),
 (7, 'Ewelina', 'Kanil', 'wdrozeniowiec IT', 8200);

Zaprojektować procedurę wynagrodzenia, która dla podanego tytułu zawodowego obliczy, ile pracowników pracuje w danym zawodzie, jakie jest minimalne, maksymalne i średnie wynagrodzenie w tej grupie zawodowej.

```
SELECT COUNT(id_pracownik) as pracownicy, MIN(pensja) as minimalne, MAX(pensja) as
maksymalne, AVG(pensja) as srednie
FROM pracownik
WHERE zawod = zawod1
GROUP BY zawod
```

Szczegóły

Nazwa procedury wynagrodzenia

Typ PROCEDURE

Kierunek	Nazwa	Typ	Długość/Wartości	Opcje
IN	zawod1	V	50	Koi - Usun

Parametry

Dodaj parametr

```
1 SELECT COUNT(id_pracownik) as pracownicy, MIN(pensja) as
2 minimalne, MAX(pensja) as maksymalne, AVG(pensja) as
3 srednie
4 FROM pracownik
5 WHERE zawod = zawod1
6 GROUP BY zawod
```

Całość procedury:

```
DELIMITER $$
CREATE DEFINER='root'@'localhost' PROCEDURE `wynagrodzenia`(IN `zawod1`
VARCHAR(50))
SELECT COUNT(id_pracownik) as pracownicy, MIN(pensja) as minimalne, MAX(pensja) as
maksymalne, AVG(pensja) as srednie
FROM pracownik
WHERE zawod = zawod1
GROUP BY zawod$$
DELIMITER ;
```

✓ Pokazano wiersze 0 - 0 (1 ogółem, Wykonaj)

```
call wynagrodzenia('nauczyciel');
```

[Edytuj w linii] [Edytuj] [Utwórz kod PHP]

☐ Pokaż wszystko | Liczba wierszy: 25

+ Opcje

pracownicy	minimalne	maksymalne	srednie
2	4200	5300	4750

Funkcje różnią się od procedur tym, że nie muszą być wskazywane parametry wejścia oraz wyjścia (funkcje bezparametrowe). Ponadto funkcja zwraca jakiś typ danych z dostępnych w SQL.

Ogólna postać:

```
CREATE FUNCTION nazwa_f(parametry) RETURNS typ
BEGIN
    -- polecenia
    RETURN wynik
END
```

Przykład:

Zaprojektować funkcję symulacja_wynagrodzenia, która dla podanego parametru zasymuluje ile będzie wynosiło wynagrodzenie po podwyżce.

DELIMITER \$\$

```
CREATE DEFINER=`root`@`localhost` FUNCTION `symulacja_wynagrodzenia`(`pensja`
FLOAT, `podwyzka_procent` FLOAT) RETURNS float
BEGIN
```

```
    DECLARE nowaPensja float;
    SET nowaPensja = pensja * podwyzka_procent;
    RETURN nowaPensja;
END$$
DELIMITER ;
```

Nazwa procedury	symulacja_wynagrodzenia			
Typ	FUNCTION			
Parametry		Nazwa	Typ	Długość/Wartości
	↑	pensja	FLOAT	
	↑	podwyżka_p	FLOAT	
	Dodaj parametr			
Zwracany typ	FLOAT			
Powrót długości/wartości				
Powrót opcji				
	<pre> 1 BEGIN 2 DECLARE nowaPensja float; 3 SET nowaPensja = pensja * podwyżka_procent; 4 RETURN nowaPensja; 5 END </pre>			
Pokazano wiersze 0 - 0 (1 ogółem, Wykonanie zapytania trwało 0,0006 sekund(y).)				
<pre>SELECT `symulacja_wynagrodzenia`(5200, 1.25) AS `symulacja_wynagrodzenia`;</pre>				
☐ Profilowanie [Edytuj w linii] [Edytuj] [Wyjaśnij SQL] [Utwórz kod PHP] [Odśwież]				
☐ Pokaż wszystko Liczba wierszy: 25 Filtrowanie wierszy: Przeszukaj				

+ Opcje

symulacja_wynagrodzenia

6500

FORMULARZ NA STRONIE WWW Z SYMULACJA WYNAGRODZENIA:

```

<body>
  <form method="POST">
    Pensja: <input type="number" name="podwyżka_procent"><br>
    Podwyżka: <input type="number" step="0.1" name="pensja"><br>
    <input type="submit" value="Policz">
  </form>
  <?php
  if (isset($_POST['podwyżka_procent']) && isset($_POST['pensja'])) {
    $podwyżka_procent = $_POST['podwyżka_procent'];
    $pensja = $_POST['pensja'];
    $c = mysqli_connect('localhost', 'root', '', 'klasa4_procedury');
    $r = mysqli_query($c, "SELECT symulacja_wynagrodzenia($pensja,
    $podwyżka_procent);");
  }
  }

```

```

        echo 'Wynagrodzenie: ';
        echo mysqli_fetch_row($r)[0];
    }
    ?>
</body>

```

```

<body>
    <form method="POST">
        Pensja: <input type="number" name="podwyzka_procent"><br>
        Podwyżka: <input type="number" step="0.1" name="pensja"><br>
        <input type="submit" value="Policz">
    </form>
    <?php
    if (isset($_POST['podwyzka_procent']) && isset($_POST['pensja'])) {
        $podwyzka_procent = $_POST['podwyzka_procent'];
        $pensja = $_POST['pensja'];
        $c = mysqli_connect('localhost', 'root', '', 'klasa4_procedury');
        $r = mysqli_query($c, "SELECT symulacja_wynagrodzenia($pensja, $podwyzka_procent);");
        echo 'Wynagrodzenie: ';
        echo mysqli_fetch_row($r)[0];
    }
    ?>
</body>

```

Pensja:

Podwyżka: