

Procedury składowane — podstawy

Czas: 2 × 45 min

praca indywidualna lub w parach

Cel: Nauczenie się tworzenia i wywoływania procedur składowanych z parametrami IN, OUT i INOUT, zmiennymi lokalnymi i instrukcją IF.

Po zajęciach potrafisz:

- tworzy procedurę z ogranicznikiem DELIMITER w konsoli i w phpMyAdmin
- stosuje parametry IN, OUT i INOUT oraz zmienne sesji @zmienna
- deklaruje zmienne lokalne, przypisuje wynik zapytania przez SELECT ... INTO i używa IF/ELSEIF
- pisze procedurę dodającą dane i procedurę raportu
- przegląda i usuwa procedury (SHOW PROCEDURE STATUS, SHOW CREATE PROCEDURE, DROP PROCEDURE)

Sprzęt i materiały

- komputer w pracowni z XAMPP (MariaDB 10.4) lub MySQL 8.4
- baza sklep zaimportowana z pliku `sklep.sql` (folder Pliki_do_zadan)

Oprogramowanie

- **XAMPP (Apache + MariaDB + phpMyAdmin)** — panel sterowania XAMPP, konsola: przycisk Shell albo `C:\xampp\mysql\bin\mysql.exe -u root`
apachefriends.org/download.html
- **phpMyAdmin** — zakładka SQL do uruchamiania zapytań; wchodzi w skład XAMPP
localhost/phpmyadmin
- **VS Code lub Notepad++** — do pisanie pliku .sql z zapytaniami
code.visualstudio.com

Bezpieczeństwo

- Procedury zmieniające dane (Z8, Z10) testuj na świeżo zaimportowanej bazie, a Z10 w transakcji zakończonej ROLLBACK.
- Usuwać tylko procedury, które sam utworzyłeś.

Przebieg zadania

1 Pierwsza procedura

Średnik kończy polecenia wewnątrz procedury, więc na czas jej tworzenia zmieniamy znak końca polecenia na //. W phpMyAdmin zamiast DELIMITER wpisz // w polu „Ogranicznik” pod oknem SQL.

```
DELIMITER //
DROP PROCEDURE IF EXISTS nazwa //
CREATE PROCEDURE nazwa(IN p_param TYP, OUT p_wynik TYP)
BEGIN
    DECLARE v_zmienna TYP;          -- zmienna lokalna (tylko w procedurze)
    ...
END //
DELIMITER ;

CALL nazwa(wartosc, @wynik);      -- @wynik to zmienna sesji
SELECT @wynik;
```

1. Zaimportuj `sklep.sql`. Utwórz plik `r2_08_nazwisko.sql`; wszystkie procedury twórz według szkieletu z ramki.
2. Z1. Utwórz procedurę `lista_kategorii()` wypisującą kategorie alfabetycznie i wywołaj ją.
3. Z2. Utwórz procedurę `produkty_kategorii(IN p_kategoria VARCHAR(50))` wypisującą nazwę, cenę i stan produktów podanej kategorii od najtańszego. Wywołaj dla kategorii Monitory.
4. Z3. Utwórz procedurę `produkty_w_cenie(IN p_od, IN p_do)` (typ DECIMAL(10,2)) i wywołaj ją dla przedziału 300–700 zł.

2 Parametry OUT i INOUT

1. Z4. Utwórz procedurę `liczba_zamowien(IN p_klient INT, OUT p_liczba INT)`. Wywołaj ją dla klienta 1 ze zmienną `@n` i wyświetl wynik.
2. Z5. Utwórz procedurę `statystyka_klienta(IN p_klient, OUT p_liczba, OUT p_suma)` zwracającą liczbę i łączną wartość zamówień nieanulowanych klienta (SELECT ... INTO dwóch parametrów). Wywołaj dla klientów 4 i 3.
3. Z6. Utwórz procedurę `dołącz_vat(INOUT p_kwota DECIMAL(10,2))`, która mnoży kwotę przez 1,23. Ustaw `SET @k = 100;`, wywołaj procedurę dwa razy i za każdym razem wyświetl `@k`. Wyjaśnij wynik.

3 Zmienne lokalne i IF

1. Z7. Utwórz procedurę `ocena_stanu(IN p_produkt INT)`: w zmiennych lokalnych zapamiętaj nazwę i stan produktu, a potem ustal ocenę: „nie ma takiego produktu” (brak wiersza), „brak — zamów” (stan 0), „niski stan” (poniżej 10), „stan dobry”. Wypisz id, nazwę, stan i ocenę. Sprawdź produkty 3, 2, 7 i 99.
2. Z8. Utwórz procedurę `dodaj_produkt(IN p_nazwa, IN p_kategoria, IN p_cena, IN p_stan, OUT p_id)`. Kategorię podaje się nazwą: jeśli nie istnieje, procedura ją dodaje. Data dodania — dzisiejsza. W `p_id` zwróć numer nowego produktu (`LAST_INSERT_ID()`). Dodaj „Kamera internetowa” (Akcesoria, 219.00, 15 szt.) i „Router Wi-Fi 7” (Sieci, 899.00, 6 szt.) i sprawdź wynik złączeniem produktów z kategoriami.

4 Raport, zmiana danych i zarządzanie procedurami

1. Z9. Utwórz procedurę `raport_miesiaca(IN p_rok INT, IN p_miesiac INT)` wypisującą zamówienia z danego miesiąca: numer, datę, klienta, status i wartość. Wywołaj dla kwietnia i grudnia 2026.
2. Z10. Utwórz procedurę `podwyzka(IN p_procent, IN p_stanowisko, OUT p_zmienionych)` podnoszącą pensję na stanowisku o podany procent i zwracającą liczbę zmienionych wierszy (`ROW_COUNT()`). W transakcji daj sprzedawcom 5%, sprawdź pensje, wykonaj `ROLLBACK`.
3. Z11. Wyświetl listę swoich procedur: `SHOW PROCEDURE STATUS WHERE Db = 'sklep'`; oraz z tabeli `information_schema.ROUTINES`. Wyświetl definicję jednej procedury: `SHOW CREATE PROCEDURE dodaj_produkt;`. Znajdź procedury w phpMyAdmin (baza sklep → Procedury).
4. Z12. Wywołaj `CALL liczba_zamowien(1);` (za mało parametrów), usuń procedurę `lista_kategorii` i spróbuj ją wywołać. Zapisz oba kody błędów.

W sprawozdaniu umieść

- ☐ plik `.sql` z definicjami procedur i wywołaniami Z1–Z12
- ☐ zrzuty ekranu wywołań Z5, Z7 (cztery produkty) i Z8
- ☐ wyjaśnienie różnicy między parametrem OUT, INOUT i zmienną sesji @
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Po co przed utworzeniem procedury zmieniamy ogranicznik poleceń (`DELIMITER`)?
2. Czym różni się zmienna lokalna `DECLARE v_x` od zmiennej sesji `@x`?
3. Dlaczego w Z8 trzeba zapamiętać `LAST_INSERT_ID()` zaraz po wstawieniu kategorii, a nie dopiero na końcu?
4. Podaj dwie zalety trzymania operacji (np. dodawania produktu) w procedurze zamiast w kodzie aplikacji.

Kryteria oceny

Kryterium	Punkty
Procedury z parametrami IN (Z1–Z3)	3
Parametry OUT i INOUT (Z4–Z6)	4
Zmienne lokalne, IF i procedura dodająca dane (Z7–Z8)	5
Raport, procedura zmieniająca dane, zarządzanie procedurami (Z9–Z12)	5
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik `.docx` o tym samym numerze).