



Национална програма
"Обучение за ИТ умения и кариера"
<https://it-kariera.mon.bg>

Министерството на
образованието и науката
<https://www.mon.bg>



Съхранени процедури

Бази данни

Какво са съхранените процедури?

- Съхранените процедури
 - Капсулират повторяща се програмна логика.
 - Могат да приемат входни параметри.
 - Могат да връщат изходни резултати.
- Ползите от съхранените процедури.
 - Споделяне на логика.
 - Подобрена производителност.
 - Намалят мрежовия трафик.

Създаване на съхранена процедура

- **CREATE PROCEDURE**

- Пример:

Име на процедура

```
DELIMITER $$  
CREATE PROCEDURE usp_select_employees_by_seniority()  
BEGIN  
    SELECT *  
    FROM employees  
    WHERE ROUND((DATEDIFF(NOW(), hire_date) / 365.25)) < 15;  
END $$
```

Процедурна логика

Изпълняване на съхранени процедури

- Изпълняване на съхранена процедура чрез **CALL**.

```
CALL usp_select_employees_by_seniority();
```

Изтриване на съхранени процедури

- DROP PROCEDURE

```
DROP PROCEDURE
```

```
usp_select_employees_by_seniority;
```

Дефиниране на параметризирани процедури

- За да дефиниране параметризирана процедура, използвайте:

```
CREATE PROCEDURE usp_procedure_name  
(  
    parameter_1_name parameter_type,  
    parameter_2_name parameter_type,...  
)
```

Параметризирани съхранени процедури

Име на процедурата

```
DELIMITER $$  
CREATE PROCEDURE usp_select_employees_by_seniority(min_years_at_work INT)  
BEGIN  
    SELECT first_name, last_name, hire_date,  
           ROUND(DATEDIFF(NOW(),DATE(hire_date)) / 365.25,0) AS 'years'  
    FROM employees  
    WHERE ROUND(DATEDIFF(NOW(),DATE(hire_date)) / 365.25,0) > min_years_at_work  
    ORDER BY hire_date;  
END $$
```

Процедурна
логика

```
CALL usp_select_employees_by_seniority(15);
```

Използване

Връщане на стойности чрез OUTPUT параметри

```
CREATE PROCEDURE usp_add_numbers
(first_number INT,
second_number INT,
  OUT result INT)
BEGIN
  SET result = first_number + second_number
END $$
DELIMITER ;

SET @answer=0;
CALL usp_add_numbers(5, 6, @answer);
SELECT @answer;

-- 11
```

Създаване на
процедура

Изпълнение на
процедура

Опечатване на резултата

Задача: изтегляне на пари

- Създайте съхранена процедура `usp_withdraw_money` (`account_id`, `money_amount`) която използва транзакции.
 - Влидайте **само** ако сметката е съществувана и хвърлете изключение в противен случай.

```
CALL usp_withdraw_money(1,10);  
SELECT * FROM accounts  
WHERE accounts.id=1;
```

Решение: изтегляне на пари [1/2]

Име на процедурата

```
CREATE PROCEDURE usp_withdraw_money  
    (account_id INT, money_amount DECIMAL)  
BEGIN  
    -- Transaction Logic goes here.  
END
```

Параметри

Решение: изтегляне на пари [2/2]

```
BEGIN  
START TRANSACTION
```

Обновяване на баланса

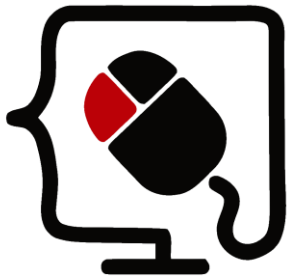
```
UPDATE accounts SET balance = balance - money_amount  
WHERE id = account_id;
```

```
IF(SELECT COUNT(*)  
   FROM accounts  
   WHERE id = account_id) <> 1 THEN ROLLBACK;  
END IF;
```

Връщане назад

```
COMMIT;  
END $$
```

Запазване на промените



Национална програма
"Обучение за ИТ умения и кариера"
<https://it-kariera.mon.bg>

Министерството на
образованието и науката
<https://www.mon.bg>



Документът е разработен за нуждите на Национална програма "Обучение за ИТ умения и кариера" на Министерството на образованието и науката (МОН) и се разпространява под свободен лиценз CC-BY-NC-SA (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).