

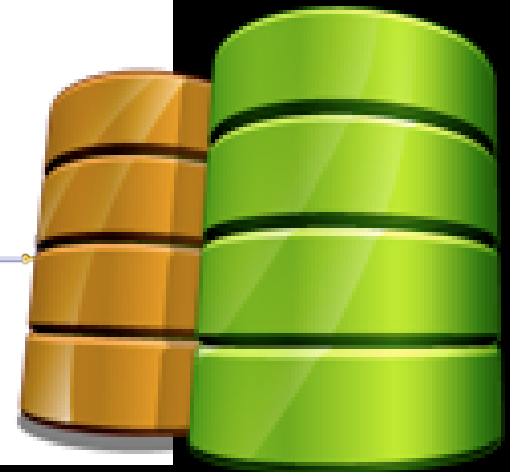
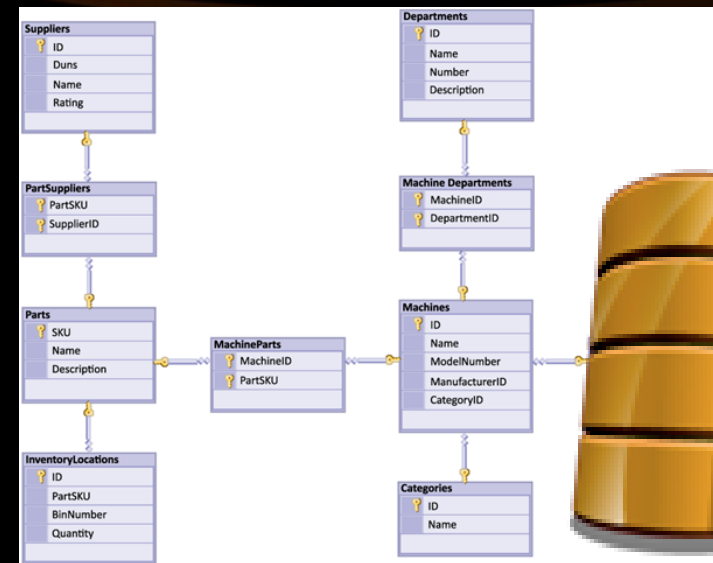
Съхранени процедури



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>





Съхранени процедури

Логика за обработка в базата данни

Какво са съхранените процедури?

- Съхранените процедури
 - Капсулират повтаряща 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:

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

Име на процедура

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;
```



Съхранени процедури

С параметри

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

Използване

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

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 ;
```

Създаване на процедура

Изпълнение на процедура

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

-- 11

Изпечатване на резултати

Задача: изтегляне на пари

- Създайте съхранена процедура **usp_withdraw_money** (account_id, money_amount) която използва транзакции.
- Влидарайте **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)

Име на процедурата

Параметри

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


Решение: Изтегляне на пари (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;
```

Връщане назад

```
END IF;  
COMMIT;  
END $$
```

Запазване на промените

Съхранени процедури



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

