

Упражнения: Общи упражнения

Problem 1. Част I – Заявки за Company_database

Problem 2. Служители със заплата над 35000

Създайте съхранена процедура `usp_get_employees_salary_above_35000` която връща **първото и последното име на всички служители**, които имат заплата над 35000. Резултатът трябва да бъде сортиран по **first_name**, а след това и по **last_name** в азбучен ред.

Пример

first_name	last_name
Amy	Alberts
Brian	Welcker
Dan	Wilson
...	...

Problem 3. Служители със заплата над...

Създайте съхранена процедура `usp_get_employees_salary_above`, която **приема число** като параметър и връща **първото и последното име на всички служители**, които имат заплата **равна на поне** даденото число. Резултатът трябва да се сортира по **first_name**, а след това и по **last_name** в азбучен ред.

Пример

Числото в този пример е 48100.

first_name	last_name
Amy	Alberts
Brian	Welcker
Dylan	Miller
...	...

Problem 4. Градове започващи със...

Създайте съхранена процедура `usp_get_towns_starting_with`, която **приема низ** като параметър и връща **всички имена на градове започващи с този низ**. Резултатът трябва да бъде сортиран по **името на града** по азбучен ред.

Пример

Това е списъкът с всички градове започващи с "b".

town_name
Bellevue
Berlin
Bordeaux
Bothell

Problem 5. Служители от град

Напишете съхранена процедура `usp_get_employees_from_town` която приема **името на град** като параметър и връща **първото и последното име на всички служители, които живеят в дадения град**. Резултатът трябва да бъде сортиран по **first_name**, а след това по **last_name** в азбучен ред.

Пример

Ето списък на служителите живеещи в Sofia.

first_name	last_name
George	Denchev
Martin	Kulov
Svetlin	Nakov

Problem 6. Функция за ниво на заплата

Напишете функция `ufn_get_salary_level`, която получава заплата на служител и връща нивото на заплатата.

- Ако заплатата е < 30000 връща "Low"
- Ако заплатата е между 30000 и 50000 (вкл) връща "Average"
- Ако заплатата е > 50000 връща "High"

Пример

salary	salary_Level
13500.00	Low
43300.00	Average
125500.00	High

Problem 7. Дефинирайте функция

Дефинирайте функция `ufn_is_word_comprised(set_of_letters, word)`, която връща `true` или `false` според това дали думата е съставена от даденото множество от букви.

Пример

set_of_letters	word	result
Oistmiahf	Sofia	1
Oistmiahf	halves	0
Bobr	Rob	1
Pppp	Guy	0

PART II – Заявки с банковата база от данни

Problem 8. Изтегляне на пари

Напишете съхранена процедура `usp_withdraw_money` (`account_id`, `money_amount`), която работи чрез транзакции.

Уверете се, че изтеглянето е приключило само ако балансът е достатъчен и `money_amount` е валидно положително число. Работете с точност до 4-тия знак след десетичната запетая.

Пример

Това е резултатът при `account_id = 1` и `money_amount = 10`.

account_id	account_holder_id	balance
1	1	113.1200

Problem 9. Банков превод

Напишете съхранена процедура `usp_transfer_money(from_account_id, to_account_id, amount)`, която трансферира пари от една сметка към друга. Разгледайте случаите, когато едно от `account_id`-тата е невалидно, сумата от пари е отрицателно число, балансът не е достатъчен или трансферите от/към един и същ акаунт. Уверете се, че цялата процедура минава без грешки и ако се случи грешка, то това не изменя базата данни.

Пример

`from_account_id = 1`, `to_account_id = 2` и `money_amount = 10`.

account_id	account_holder_id	balance
1	1	113.1200
2	3	4364.2300

Problem 10. Логове за банкови сметки

Създайте нова таблица – `logs` (`log_id`, `account_id`, `old_sum`, `new_sum`). Добавете тригер към таблицата `accounts`, който да добавя нов запис в `logs` всеки път, когато има промяна по банкова сметка.

Пример

log_id	account_id	old_sum	new_sum
1	1	123.12	113.12
...

Problem 11. Тригер за мейли

Създайте нова таблица – `notification_emails`(`id`, `recipient`, `subject`, `body`). Добавете тригер към таблицата с логовете от предната задача, за да създадете нов мейл всеки път, когато бъде добавен запис в `logs` таблицата. Следната информация е нужна за всеки мейл:

- **recipient** – `account_id`
- **subject** – “Balance change for account: {`account_id`}”
- **body** - “On {`date`} your balance was changed from {`old`} to {`new`}.”

Пример

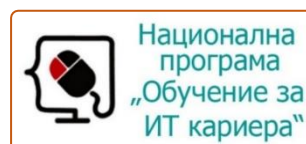
id	recipient	subject	body
1	1	Balance change for account: 1	On Sep 15 2016 at 11:44:06 AM your balance was changed from 133 to 143.
...

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** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).

