


# **ПРОЦЕДУРЫ И ФУНКЦИИ**

**ЛАБОРАТОРНАЯ РАБОТА №5**

# ПОРЯДОК ВЫПОЛНЕНИЯ

1. Изучить теоретический материал (см. лекцию [«Процедурный SQL»](#)).
2. Установить демонстрационную [БД «Проездные школьника»](#).
3. Изучить примеры из текста лабораторной работы.
4. Используя примеры, создать в демонстрационной БД функцию `vozrast()` и хранимые процедуры `all_classes_info ()`.
5. Разработать функции и процедуры из заданий для самостоятельного выполнения.
6. В качестве результата работы предоставить 4 sql-файла (по одному для каждого задания), содержащих код для создания функции/процедуры. В качестве имени использовать ***N\_FamiliyaO.sql***, где *N* – номер задания.



# **ХРАНИМЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ**

# ХРАНИМЫЕ ПРОЦЕДУРЫ

```
CREATE PROCEDURE <имя_процедуры>
( [ [IN | OUT | INOUT ] параметр | [,...] ] )
{[NOT] DETERMINISTIC}
{ CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
BEGIN
    <тело_процедуры>
END
```

# ФУНКЦИИ

```
CREATE FUNCTION <имя_функции> ( [<параметр|> тип_данных [...]] )  
RETURNS тип_данных  
[NOT] DETERMINISTIC  
{ CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }  
BEGIN  
    <тело_функции>  
END
```

# ПРОЦЕДУРА ДЛЯ СУММЫ ДВУХ ЧИСЕЛ

```
sum_a_b
```

The name of statement. The

```
1 • CREATE DEFINER=`root`@`localhost` PROCEDURE `sum_a_b`(in a int, in b int/*, out sum int*/)
2 • begin
3 •   declare sum int;
4 •   set sum=a+b;
5 •   select sum as 'Сумма';
6 • end
```

```
1 • set @sum=0;
2 • call sum_a_b(15, 12, @sum);
3 • select @sum;
```

<

Result Grid | Filter Rows: | Export:

@sum
27

# ФУНКЦИЯ ДЛЯ СУММЫ ДВУХ ЧИСЕЛ

```
get_sum
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `get_sum`(a int, b int) RETURNS int
2     DETERMINISTIC
3 begin
4     declare res int;
5     set res=a+b;
6     return res;
7 end
```

```
1 • select get_sum(15,12);
```

<	
Result Grid	
	get_sum(15,12)
▶	27

A decorative wavy line in yellow and white on the left side of the image.

# **ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ**



# ЗАДАНИЕ №1

Изменить функцию `vozrast()` таким образом, чтобы при ее вызове рассчитывался возраст ученика на текущую дату (по его id).

```
1 • CREATE DEFINER=`root`@`localhost` FUNCTION `vozrast` (b_date DATE) RETURNS int
2     DETERMINISTIC
3     begin
4     declare vozrast INT ;
5     set vozrast=0;
6     set vozrast=DATE_FORMAT(FROM_DAYS(TO_DAYS(now()) - TO_DAYS(b_date)), '%Y') + 0;
7     -- select b_date as 'Дата рождения', vozrast as 'Возраст на сегодня';
8     return vozrast;
9     end
```

# ЗАДАНИЕ №2

## СОЗДАТЬ ФУНКЦИЮ ДЛЯ РАСЧЕТА ВЫПЛАТЫ НА РУКИ З/П

На вход поступает следующая информация:

- количество рабочих дней за месяц,
- количество отработанных дней в отчетном периоде,
- оклад.

***зарплата=оклад\*отработанные\_дни/всего\_рабочих.***

***Сумма на руки=зарплата\*(1-0,13-0,01)=зарплата\*0,86.***

Пример расчета: месяц – май, рабочих дней в месяце (за исключением выходных и праздничных) – 10, всего рабочих дней в месяце мае – 23, оклад – 25751,00 рубль.

$$ЗП=25751*10/23=11196,09$$

$$На\ руки=ЗП*0,86=11196,09*0,86=9628,63.$$

# ЗАДАНИЕ №3

## СУММА, ВЫДАННАЯ НА РУКИ

На вход поступает:

- количество рабочих дней за месяц,
- количество отработанных дней в отчетном периоде,
- сумма оклада.

Вывести результат в виде (прописать заголовки):

Оклад	отработанных дней/рабочих дней	Всего начислено	Подходный налог	Профсоюзные отчисления	К выдаче
25751.00	10/23	11196.08	-1455.49	-111.96	9628.63

# ЗАДАНИЕ №4

Вывести отформатированную информацию о количестве учеников в заданном классе (по id):

- Класс – год обучения и литеру (3 «а»),
- Номер школы (9),
- Количество учеников.

