# Практическая работа №1.1

Студент: Болдинов Алексей ЭФМО-02-24

1. Создайте таблицу РАБОЧИЙ ДЕНЬ, в которой будет храниться информация о входе/выходе сотрудника через систему контроля доступа посредством электронной карты. Считается, что в таблице хранятся данные за одну неделю. Для каждого сотрудника фикси руется время входа (значение = 1), время выхода (значение = 2). Рабочий день начинается в 9:00 часов и заканчивается в 18:00 часов. Время обеда: с 13:00 до 14:00 часов. Опоздание – это вход позднее 9:00 часов, уход на обед до 13:00, возвращение с обеда после 14:00, уход в конце рабочего дня до 18:00 часов. Заполните таблицу тестовыми данными (указать для нескольких сотрудников время входа/выхода для 1-2 рабочих дней). Фрагмент тестовых данных представлен ниже.

CREATE TABLE work\_day (

employee\_id bigint

datetime timestamp

reader\_value smallint,

PRIMARY KEY (employee\_id, datetime)

);

1. Создайте функцию, которая выводит общее количество часов, отработанное со трудником за прошедшую неделю. Если сотрудник отработал меньше 40 часов, сообщить «Меньше нормы», если 40 часов – «Норма», если больше 40 часов – «Больше нормы».

DECLARE

ei bigint := employee\_id;

total\_seconds bigint := 0; -- Суммарное количество секунд

total\_hours bigint;

entry\_time timestamp; -- Время входа

exit\_time timestamp; -- Время выхода

BEGIN

-- Проходим по всем данным для данного сотрудника

FOR entry\_time, exit\_time IN

SELECT r.datetime,

LEAD(r.datetime) OVER (PARTITION BY r.employee\_id ORDER BY r.datetime) AS exit\_time

FROM work\_day r

WHERE r.employee\_id = ei

ORDER BY r.datetime

LOOP

-- Проверяем, что это вход (1) и выход (2)

IF entry\_time IS NOT NULL AND exit\_time IS NOT NULL THEN

-- Расчет времени между входом и выходом (в секундах)

total\_seconds := total\_seconds + EXTRACT(epoch FROM (exit\_time - entry\_time));

END IF;

END LOOP;

-- Преобразуем секунды в часы

total\_hours := total\_seconds / 3600;

-- Проверяем отработанное время и возвращаем результат

IF total\_hours < 40 THEN

RETURN 'Меньше нормы';

ELSIF total\_hours = 40 THEN

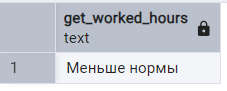
RETURN 'Норма';

ELSE

RETURN 'Больше нормы';

END IF;

END;



;

1. Создайте функцию, которая производит расчет заработной платы сотрудников по формуле:

ЗАРПЛАТА=ОКЛАД + ОКЛАД\*A

ОКЛАД – базовый оклад предприятия А - при отсутствии опоздания у сотрудника А=1, за каждые 10 мин. опоздания А уменьшается на 0,05 (А >= 0).

DECLARE

ei bigint := employee\_id;

entry\_time time;

max\_allowed\_time time := '09:00:00'; -- Ожидаемое время входа (9:00)

delay\_minutes double precision := 0;

coefficient double precision := 1; -- Изначальный коэффициент A

BEGIN

-- Получаем время первого входа сотрудника

SELECT datetime INTO entry\_time

FROM work\_day

WHERE ei = ei AND reader\_value = 1

ORDER BY datetime

LIMIT 1;

-- Если вход позже 9:00, вычисляем время опоздания

IF entry\_time > max\_allowed\_time THEN

-- Вычисляем количество минут опоздания

delay\_minutes := EXTRACT(EPOCH FROM (entry\_time - max\_allowed\_time)) / 60;

-- Уменьшаем коэффициент за каждые 10 минут опоздания

coefficient := 1 - (delay\_minutes / 10) \* 0.05;

-- Коэффициент не может быть меньше 0

IF coefficient < 0 THEN

coefficient := 0;

END IF;

END IF;

-- Рассчитываем заработную плату

RETURN base\_salary + (base\_salary \* coefficient);

END;

SELECT calculate\_salary(100, 50000);

