

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Інститут прикладної математики та фундаментальних наук  
Кафедра прикладної математики

### **Звіт**

про виконання лабораторної роботи №3  
з курсу «Організація баз даних та знань»

**Виконав:**

студент групи ПМ-22

Янів П.В.

**Перевірив:**

Пабірівський В.В.

Львів-2024

## Варіант №11 “Оплата праці”

На підприємстві працюють працівники, кожен з яких належить до певного структурного підрозділу. Кожен структурний підрозділ має свою назву, код, керівника. Кожен працівник працює на певній посаді та характеризується табельним номером, прізвищем, ім'ям, по-батькові, паспортними даними, датою та місцем народження, домашньою адресою.

На підприємстві встановлені такі форми оплати праці

- основна заробітна плата у вигляді погодинної оплати за кожну відпрацьовану годину, причому для кожної посади існує свій тариф за годину,
- додаткова заробітна плата – оплата за роботу у понад встановлені норми, яка полягає в подвійній оплаті за кожну додатково відпрацьовану годину та потрійній – за кожну відпрацьовану годину у вихідний (субота, неділя) та святковий день.

Працівник кожного року має відпустку. Система повинна зберігати дані про всі відпустки працівників. Одночасно перебувати в відпустці можуть не більше 15% працівників одного відділу. Оплата відпусток проводиться так:  $S = M / (365 - C) * N$ , де  $S$  – сума відпускних,  $M$  – сумарний заробіток працівника за останні 12 місяців, 365 – кількість днів у році,  $C$  – число святкових днів,  $N$  – тривалість відпустки в календарних днях.

Система повинна вести облік робочого часу (фактично відпрацьовані години працівником за кожен день), нараховувати місячну зарплату всім працівникам, розраховувати для кожного працівника заробітну плату до видачі на руки (заробітна плата до видачі на руки – це нарахована заробітна плата мінус соціальний податок 20%).

Система повинна вести облік виплати зарплати працівника.

В системі повинні діяти такі додаткові обмеження: працівник не може працювати в декількох відділах та на декількох посадах одночасно; основна зарплата працівника не може бути меншою від встановленої мінімальної зарплати, працівнику забороняється нараховувати зарплату після його звільнення; виплата зарплати працівникам повинна здійснюватись не пізніше 10 числа наступного місяця, який слідує за відпрацьованим, при порушенні цього правила працівнику додається до зарплати за поточний місяць пеня в розмірі 0,1% від суми заборгованості за кожен прострочений день.

- розрахункову відомість заробітної плати для відділу,
- відомість на отримання заробітної плати для відділу,
- відомість про роботу працівника впродовж певного періоду,
- відомість про борг підприємства перед працівниками.

## Завдання:

### Лабораторна робота № 3. “Підпрограми СУБД Oracle”

Завдання лабораторної роботи виконувати в призначеній предметній області. При написанні процедур повинні бути використані лише базові таблиці відповідно до створеної у ЛР №1 схеми БД, тобто результати роботи процедур повинні записуватися у відповідні поля існуючих таблиць БД та *не можна* створювати жодних тимчасових таблиць. Змістовне призначення процедур є наступним:

**Варіант 11.** Написати процедуру нарахування зарплати за вказаний місяць заданому працівнику (в тому числі врахувати нарахування відпускних). Написати процедуру, яка б викликала першу процедуру для всіх працівників.

## У звіті має бути наведено:

Для кожної процедури / функції:

- 1) Призначення процедури / функції.
- 2) Опис алгоритму роботи (концепція роботи та покроковий опис, можливо блок-схема).

- 3) Опис вхідних / вихідних параметрів (їх призначення, типи, варіанти застосувань).
- 4) Опис внутрішніх ідентифікаторів (їх призначення, типи, застосування).
- 5) Результати виконання.

## Процедури:

Створив процедуру PaySalary з двома параметрами.

```
CREATE PROCEDURE [PaySalary] (@EmployeeID INT, @MounthPay varchar(255))
AS
BEGIN
INSERT INTO [dbo].[SalaryGive] ([EmployeeID] , [Salary] , [PayGiveData])
SELECT @EmployeeID,
Sum(( [HoursWorked].[HoursWorkedDay]+[HoursWorked].[AddHoursWorkedDay])*[SalaryPerHours].[Salary]) + ISNULL((
SELECT Vacation.VacationPay
FROM Vacation
WHERE (Vacation.EmployeeID = @EmployeeID AND Vacation.VacationMounth = @MounthPay) OR
Vacation.VacationMounth IS NULL
),0
),@MounthPay
FROM HoursWorked, SalaryPerHours
WHERE [HoursWorked].[EmployeeID]=@EmployeeID AND
CONVERT(VARCHAR,[SalaryPerHours].[PositionName])=CONVERT(VARCHAR, (SELECT Position
FROM EmployeeINFO
WHERE EmployeeID=@EmployeeID))
AND ([HoursWorked].[MounthWork] LIKE '___.'+ @MounthPay)

END;
```

Параметр @EmployeeID це табельний номер працівника для нарахування зарплати та @MounthPay вказує місяць за який нараховується зарплата

Дані про видану зарплату записується у створену таблицю SalaryGive, записується табельний номер працівника, сума зарплати, та місяць за який була видана зарплата.

Зарплата обчислюється за допомогою SQL-запитів.

```
Sum(( [HoursWorked].[HoursWorkedDay]+[HoursWorked].[AddHoursWorkedDay])*[SalaryPerHours].[Salary])
```

Цей запит додає відпрацьовані години до додаткових відпрацьованих годин та множить їх на погодинну оплату яку отримує посада на якій записаний працівник.

Дані про відпрацьовані години береться з таблиці HoursWorked де записані дані про відпрацьовані години кожного працівника. Дані про оплату відпрацьованої години для кожної посади береться з таблиці SalaryPerHours.

```

ISNULL((
SELECT Vacation.VacationPay
FROM Vacation
WHERE (Vacation.EmployeeID = @EmployeeID AND Vacation.VacationMounth =@MounthPay) OR
Vacation.VacationMounth IS NULL
),0
)

```

Цей запит додає до порахованої оплати відпрацьованих годин ще оплату відпустки за вказаний місяць, якщо вона існує, з таблиці Vacation

Далі за допомогою WHERE беру з таблиць тільки ті дані які подрібні.

Для виконання наступної частини лабораторної створив процедуру PaySalaryForAllEmployee яка викликає попередню процедуру для усіх працівників

```

CREATE PROCEDURE [PaySalaryForAllEmployee] (@Mounth varchar(255))
AS
BEGIN

DECLARE @Count as INT = 1;
WHILE (@Count<=16)
BEGIN
EXEC PaySalary @EmployeeID = @Count, @MounthPay = @Mounth;
SET @Count = @Count+1
END;
END;

```

Ця процедура приймає тільки один параметр який визначає місяць за який нараховується зарплата.

За допомогою циклу WHILE та індекса @Count викликаю попередню процедуру для усіх працівників.

Також для того щоб не було ідентичних записів у таблиці SalaryGive, в яку записуються усі видані зарплати, задав обмеження що кожне поле є унікальне

## Перевірка роботи процедур:

Спочатку використаю запит:

```
EXEC PaySalary @EmployeeID = 1 , @MounthPay = '03.2024'
```

Для нарахування зарплати працівнику, з табельним номером 1, за березень

Далі перевіряю чи з'явився запит у таблиці:

```
SELECT TOP (1000) [EmployeeID]
      ,[Salary]
      ,[PayGiveData]
FROM [YanivPM22Var11].[dbo].[SalaryGive]
```

136 %

Results Messages

	EmployeeID	Salary	PayGiveData
1	1	360	03.2024

Дані записані правильно.

Тепер перевірю другу процедуру:

**EXEC** PaySalaryForAllEmployee @Mounth = '03.2024'

Далі перевіряю чи з'явився запит у таблиці:

```
SELECT TOP (1000) [EmployeeID]
      ,[Salary]
      ,[PayGiveData]
FROM [YanivPM22Var11].[dbo].[SalaryGive]
```

136 %

Results Messages

	EmployeeID	Salary	PayGiveData
1	1	360	03.2024
2	2	46	03.2024
3	3	360	03.2024
4	4	213	03.2024
5	5	710	03.2024
6	6	540	03.2024
7	7	299	03.2024
8	8	230	03.2024
9	9	575	03.2024
10	10	150	03.2024
11	11	425	03.2024
12	12	300	03.2024
13	13	270	03.2024
14	14	330	03.2024
15	15	720	03.2024
16	16	330	03.2024

Query executed successfully.

Дані записані правильно та не дублюються.

**GitHub репозиторій:**

<https://github.com/PetroYaniv/DBSQLYanivPetroPM22.git>

**Висновок:** Для виконання лабораторної роботи №3 створив дві процедури для нарахування зарплати. Задав обмеження в таблицях щоб записи не повторювались. Процедури беруть дані з таблиць створених у лабораторній роботі №1. За допомогою параметрів процедури беруть з таблиць тільки ті дані які потрібні для коректної роботи. Результат роботи процедур є запис у задану таблицю.