

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра «Системи штучного інтелекту»



Звіт до лабораторної роботи №10
З дисципліни «Організація Баз Даних»

Виконав:
студент групи КН-208
Дерев'янний Андрій

Прийняла:
Мельникова Н.І

Львів-2020

Тема: Написання збережених процедур на мові SQL

Мета: навчитися розробляти та виконувати збережені процедури та функції у MySQL

Завдання:

Розробити та виконати збережені процедури та функції у MySQL

Виконання завдання:

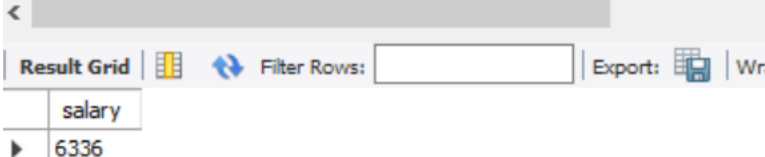
1) Створимо функцію для обрахунку зарплати за місяць на основі щоденної оплати(8 робочих годин) і кількості робочих днів:

```
3 • CREATE FUNCTION calculate_salary (daily_pay float(9), paid_days int(2))
4 RETURNS float deterministic
5 RETURN (daily_pay*paid_days*0.9);
6
7 • select calculate_salary(320, 22) as salary;
```

де 320 – заробіток за день, 22 – кількість робочих днів

Результат роботи функції:

```
7 • select calculate_salary(320, 22) as salary;
```




salary
6336

Тепер створимо ще одну функцію. Цього разу будемо розраховувати зарплату за місяць з врахуванням понаднормових годин:

```
9 • CREATE FUNCTION calculate_overtime (daily_pay float(9), paid_days int(2), overtime int(3))
10 RETURNS float deterministic
11 RETURN (calculate_salary(daily_pay, paid_days)+(daily_pay*overtime*0.3));
12
```

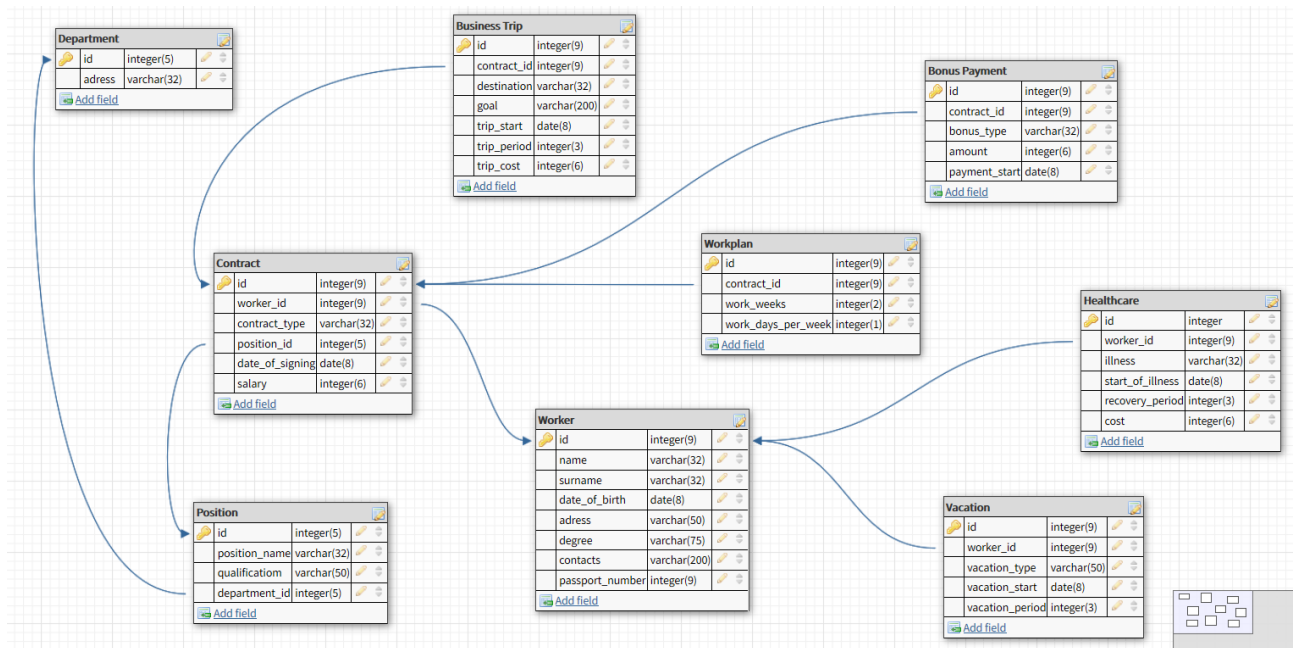
Результат роботи функції:

```
13 • select calculate_overtime(320, 22, 44) as overtime;
```



overtime
10560

2) Діаграма бази даних(відділ кадрів підприємства):



Створимо процедуру виведення найважливішої інформації про робітника: ім'я, прізвище, тип контракту, зарплата, посада, адреса офісу

```

3  DELIMITER //
4
5  • CREATE PROCEDURE worker_info (IN id int(9))
6  BEGIN
7      DECLARE error char;
8      SET error = '0';
9      IF (id > 0) THEN
10         BEGIN
11             CREATE TABLE IF NOT EXISTS db.worker_info (name varchar(32), surname varchar(32), contract varchar(32), salary int(6));
12             TRUNCATE db.worker_info;
13             INSERT INTO db.worker_info SELECT name, surname, contract_type as contract, salary, position_name as position, department_id as department_address
14             FROM ((worker INNER JOIN contract) INNER JOIN position) INNER JOIN department
15             ON worker.id=contract.worker_id
16             AND contract.position_id=position.id
17             AND position.department_id=department.id
18             AND worker.id=id;
19         END;
20         ELSE SELECT error;
21         END IF;
22     END//
23     DELIMITER ;

```

Результат роботи процедури для робітника з ідентифікаційним номером 6:

```

25 • call worker_info(6);
26
27 • select * from worker_info;

```

name	surname	contract	salary	position	department_address
chloe	price	some_type	8000	junior_position	someAddress3

Висновок:

За час виконання лабораторної роботи я навчився розробляти та виконувати збережені процедури та функції у MySQL

Додаток:

SQL-скрипт бази даних:

```
create database db;
```

```
use db;
```

```
CREATE TABLE `Department` (  
    `id` INT(5) NOT NULL AUTO_INCREMENT,  
    `adress` varchar(32) NOT NULL,  
    PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE `Worker` (  
    `id` INT(9) NOT NULL AUTO_INCREMENT,  
    `name` varchar(32) NOT NULL,  
    `surname` varchar(32) NOT NULL,  
    `date_of_birth` DATE NOT NULL,  
    `adress` varchar(50) NOT NULL,  
    `degree` varchar(75) NOT NULL,  
    `contacts` varchar(200) NOT NULL,  
    `passport_number` INT(9) NOT NULL,  
    PRIMARY KEY (`id`)  
);
```

```
CREATE TABLE `Vacation` (  
    `id` INT(9) NOT NULL AUTO_INCREMENT,  
    `worker_id` INT(9) NOT NULL,  
    `vacation_type` varchar(50),  
    `vacation_start` DATE,  
    `vacation_period` INT(3),  
    PRIMARY KEY (`id`),
```

```
CONSTRAINT fk0 FOREIGN KEY (worker_id)  
    REFERENCES Worker(id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `Position` (  
    `id` INT(5) NOT NULL AUTO_INCREMENT,  
    `position_name` varchar(32) NOT NULL,  
    `qualification` varchar(50) NOT NULL,  
    `department_id` INT(5) NOT NULL,  
    PRIMARY KEY (`id`),  
  
    CONSTRAINT fk1 FOREIGN KEY (department_id)  
        REFERENCES Department(id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `Contract` (  
    `id` INT(9) NOT NULL AUTO_INCREMENT,  
    `worker_id` INT(9) NOT NULL,  
    `contract_type` varchar(32) NOT NULL,  
    `position_id` INT(5) NOT NULL,  
    `date_of_signing` DATE NOT NULL,  
    `salary` INT(6) NOT NULL,  
    PRIMARY KEY (`id`),  
  
    CONSTRAINT fk2 FOREIGN KEY (worker_id)  
        REFERENCES Worker(id)  
        ON DELETE CASCADE ON UPDATE CASCADE,  
  
    CONSTRAINT fk3 FOREIGN KEY (position_id)  
        REFERENCES `Position`(id)  
        ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE `Workplan` (  
    `id` INT(9) NOT NULL AUTO_INCREMENT,  
    `contract_id` INT(9) NOT NULL,  
    `work_weeks` INT(2),  
    `work_days_per_week` INT(1),
```

PRIMARY KEY (`id`),

CONSTRAINT fk4 FOREIGN KEY (contract_id)
REFERENCES Contract(id)
ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE `Bonus_Payment` (
 `id` INT(9) NOT NULL AUTO_INCREMENT,
 `contract_id` INT(9) NOT NULL,
 `bonus_type` varchar(32),
 `amount` INT(6),
 `payment_start` DATE,
 PRIMARY KEY (`id`),

 CONSTRAINT fk5 FOREIGN KEY (contract_id)
 REFERENCES Contract(id)
 ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE `Healthcare` (
 `id` INT NOT NULL AUTO_INCREMENT,
 `worker_id` INT(9) NOT NULL,
 `illness` varchar(32),
 `start_of_illness` DATE,
 `recovery_period` INT(3),
 `cost` INT(6),
 PRIMARY KEY (`id`),

 CONSTRAINT fk6 FOREIGN KEY (worker_id)
 REFERENCES Worker(id)
 ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE `Business_Trip` (
 `id` INT(9) NOT NULL AUTO_INCREMENT,

`contract_id` INT(9) NOT NULL,

`destination` varchar(32),

`goal` varchar(200),

`trip_start` DATE,

`trip_period` INT(3),

`trip_cost` INT(6),

PRIMARY KEY (`id`),

CONSTRAINT fk7 FOREIGN KEY (contract_id)

REFERENCES Contract(id)

ON DELETE CASCADE ON UPDATE CASCADE

);