

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA**

KHOA KHOA HỌC & KĨ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN 2 - MÔN HỆ CƠ SỞ DỮ LIỆU (CO2013) - HK241

PHẦN MỀM QUẢN LÝ NHÂN SỰ VÀ CHẤM CÔNG

Nhóm: LỚP L11

Giảng viên hướng dẫn: Trần Thị Quế Nguyệt

Sinh viên thực hiện:
Nguyễn Danh Thành - 2114782
Nguyễn An Khang - 2211441
Trần Anh Khôi - 2211694
Hồ Quốc Khương - 2211709

Thành phố Hồ Chí Minh, tháng 12 năm 2024

Mục lục

1	Danh sách thành viên & Phân chia công việc	1
2	Tạo bảng và dữ liệu mẫu	2
2.1	Sơ đồ mô hình dữ liệu quan hệ	2
2.2	Hiện thực các bảng và tạo dữ liệu mẫu	3
2.2.1	Bảng ChiNhanh	3
2.2.2	Bảng PhongBan	6
2.2.3	Bảng NhanVien	9
2.2.4	Bảng Sdt_NhanVien	12
2.2.5	Bảng NhanVienToanThoiGian	14
2.2.6	Bảng NhanVienBanThoiGian	16
2.2.7	Bảng DuAn	17
2.2.8	Bảng NhanVienThamGiaDuAn	20
2.2.9	Bảng NguoiPhuThuoc	22
2.2.10	Bảng BangChamCong	25
2.2.11	Bảng LanRaVao	27
2.2.12	Bảng LichLamViec	29
2.2.13	Bảng BangThietLapLuong	31
2.2.14	Bảng BangLuong	32
3	Hiện thực ứng dụng	34
3.1	Thủ tục INSERT/UPDATE/DELETE dữ liệu vào 1 bảng dữ liệu	34
3.1.1	Thủ tục INSERT	34
3.1.2	Thủ tục UPDATE	37
3.1.3	Thủ tục DELETE	42
3.2	Trigger	44
3.2.1	Nhóm Trigger 1	44
3.2.2	Nhóm Trigger 2	47
3.3	Thủ tục	50
3.3.1	Thủ tục 1	50
3.3.2	Thủ tục 2	51
3.4	Hàm	52
3.4.1	Hàm 1	52

3.4.2 Hàm 2	55
4 Ứng dụng kết nối tới cơ sở dữ liệu	58
4.1 Tổng quan công nghệ	58
4.2 Khai báo kết nối đến MySQL	59
4.3 Các màn hình demo	60
4.3.1 Màn hình 1	60
4.3.2 Màn hình 2	76
4.3.3 Màn hình 3	84
5 Phần thêm	88
5.1 Người thực hiện: Trần Anh Khôi - 2211694	88
5.1.1 Function tonggiolamviec	88
5.1.2 Procedure View_Bangluong	90
6 Source Code	94

Danh sách hình vẽ

1 Mô hình dữ liệu quan hệ của hệ thống	2
2 Dữ liệu bảng ChiNhanh	5
3 Dữ liệu bảng PhongBan	8
4 Dữ liệu bảng NhanVien	11
5 Dữ liệu bảng Sdt_NhanVien	13
6 Dữ liệu bảng NhanVienToanThoiGian	15
7 Dữ liệu bảng NhanVienBanThoiGian	16
8 Dữ liệu của bảng DuAn	19
9 Dữ liệu của bảng NhanVienThamGiaDuAn	21
10 Dữ liệu của bảng NguoiPhuThuoc	24
11 Dữ liệu của bảng BangChamCong	26
12 Dữ liệu của bảng LanRaVao	28
13 Dữ liệu của bảng LichLamViec	30
14 Dữ liệu của bảng BangThietLapLuong	31
15 Dữ liệu của bảng BangLuong	33
16 Sử dụng thủ tục ThemNhanVien nhưng sai định dạng MaNV	36
17 Sử dụng thủ tục ThemNhanVien nhưng sai định dạng TenLoi	36

18	Sử dụng thủ tục ThemNhanVien nhưng sai định dạng Email	36
19	Sử dụng thủ tục ThemNhanVien nhưng mã phòng ban cung cấp không tồn tại	36
20	Sử dụng thủ tục ThemNhanVien với các tham số đúng định dạng	36
21	Sử dụng thủ tục SuaNhanVien trong trường hợp MaNV không tồn tại	40
22	Sử dụng thủ tục SuaNhanVien trong trường hợp MaNV sai định dạng	40
23	Sử dụng thủ tục SuaNhanVien trong trường hợp Họ sai định dạng	40
24	Sử dụng thủ tục SuaNhanVien trong trường hợp HeSoPhatDiTre bé hơn 0	40
25	Trước khi sử dụng thủ tục SuaNhanVien	41
26	Sử dụng thủ tục SuaNhanVien thành công	41
27	Kết quả sau khi thành công sử dụng thủ tục SuaNhanVien, email đã được thay đổi	41
28	Thực thi thủ tục XoaNhanVien thành công	43
29	Kết quả thực thi thủ tục XoaNhanVien	43
30	Thông tin của phòng ban 'PB0101' ban đầu, số nhân viên hiện tại là 5	45
31	Thực hiện thêm 3 nhân viên vào phòng ban 'PB0101'	45
32	Thông tin của phòng ban 'PB0101', số nhân viên đã tăng từ 5 lên 8	45
33	Thông tin của phòng ban 'PB0201' ban đầu, số nhân viên hiện tại là 4	45
34	Thực hiện việc chuyển hai nhân viên từ phòng ban 'PB0101' sang phòng ban 'PB0201'	46
35	Thông tin của phòng ban 'PB0101', số nhân viên đã giảm từ 8 xuống 6	46
36	Thông tin của phòng ban 'PB0201', số nhân viên đã tăng từ 4 lên 6	46
37	Thực hiện việc xóa một nhân viên thuộc phòng ban 'PB0201'	46
38	Thông tin của phòng ban 'PB0201', số nhân viên đã giảm từ 6 xuống 5	46
39	Thông tin ban đầu của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 là 0	48
40	Thực hiện thêm các lần ra vào công ty của nhân viên 'NV0001'	48
41	Thông tin của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 đã được cập nhật thành 7.37	48
42	Thực hiện giảm giờ ra của một thực thể trong bảng LanRaVao đi 15 phút	48
43	Thông tin của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 đã giảm từ 7.37 xuống 7.12 (giảm 0.25 giờ = 15 phút)	49
44	Thực hiện xóa một thực thể trong bảng LanRaVao, tổng thời gian bị xóa là 3 giờ 30 phút	49
45	Thông tin của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 đã giảm từ 7.12 xuống 3.62 (giảm 3.5 giờ = 3 giờ 30 phút)	49
46	Kết quả sau khi thực thi thủ tục LocPhongBanCoSoLuongNhanVienCoMatNhieuNhat	50
47	Kết quả sau khi thực thi thủ tục LocPhongBanCoSoLuongNhanVienLonHon với số lượng nhân viên tối thiểu là 10	51

48	Thông tin về nhân viên và giờ làm việc của nhân viên	54
49	Kết quả sau khi gọi hàm tinhluong	54
50	Kết quả truy vấn	57
51	Kết quả sau khi gọi hàm KiemTraTangCa	57
52	Kết quả sau khi gọi hàm KiemTraTangCa	57
53	Hiển thị thông tin bảng NhanVien, được sắp xếp mặc định theo mã nhân viên theo thứ tự a-z	60
54	Màn hình hiển thị trang thêm nhân viên	60
55	Màn hình thêm nhân viên - thất bại do sai định dạng email	61
56	Màn hình thêm nhân viên - thành công	61
57	Tra cứu thành công, nhân viên vừa được thêm ở trang web đã được cập nhật bên phía back-end	61
58	Màn hình cập nhật thông tin nhân viên	62
59	Màn hình thông báo lỗi lúc cập nhật thông tin nhân viên	62
60	Màn hình trước khi nhấn vào nút 'Cập nhật'	63
61	Màn hình thông báo cập nhật thông tin nhân viên 'NV0003' thành công	63
62	Thông tin của nhân viên 'NV0003' đã được cập nhật thành công, với GioiTinh được đổi thành 'Nam', HeSoPhatDiTre được đổi thành 0.08 và LuongTheoGio được đổi thành 40000	63
63	Khi click vào icon thùng rác trong hàng chứa mã nhân viên 'NV0003', màn hình hiện lên bảng xác nhận	64
64	Màn hình sau khi xóa nhân viên có mã 'NV0003'	64
65	Kết quả tra cứu rỗng, chứng tỏ nhân viên 'NV0003' đã bị xóa	64
66	Hiển thị thông tin bảng PhongBan, cùng với các nút chức năng	76
67	Màn hình lọc phòng ban có số lượng nhân viên lớn hơn 5	77
68	Màn hình sắp xếp phòng ban dựa trên số ngày có mặt của nhân viên	77
69	Màn hình ban đầu, do chưa xác định thời gian nên hiện bảng rỗng	84
70	Bảng lương của toàn bộ nhân viên tính từ ngày 01/11/2024 đến ngày 30/11/2024 . .	84
71	Bảng lương của các nhân viên toàn thời gian thuộc phòng ban 'PB0303'	85
72	Bảng lương của các nhân viên toàn thời gian thuộc phòng ban '0101'	85
73	Kết quả câu lệnh tra cứu thông thường	88
74	Kết quả câu lệnh gọi hàm tonggiolamviec	89
75	Bảng lương của tất cả các nhân viên thuộc phòng ban 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024	93

76	Bảng lương của các nhân viên toàn thời gian thuộc phòng ban 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024	93
77	Bảng lương của các nhân viên bán thời gian thuộc phòng ban 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024	93

Danh sách bảng

1	Danh sách thành viên và phân chia công việc	1
---	---	---

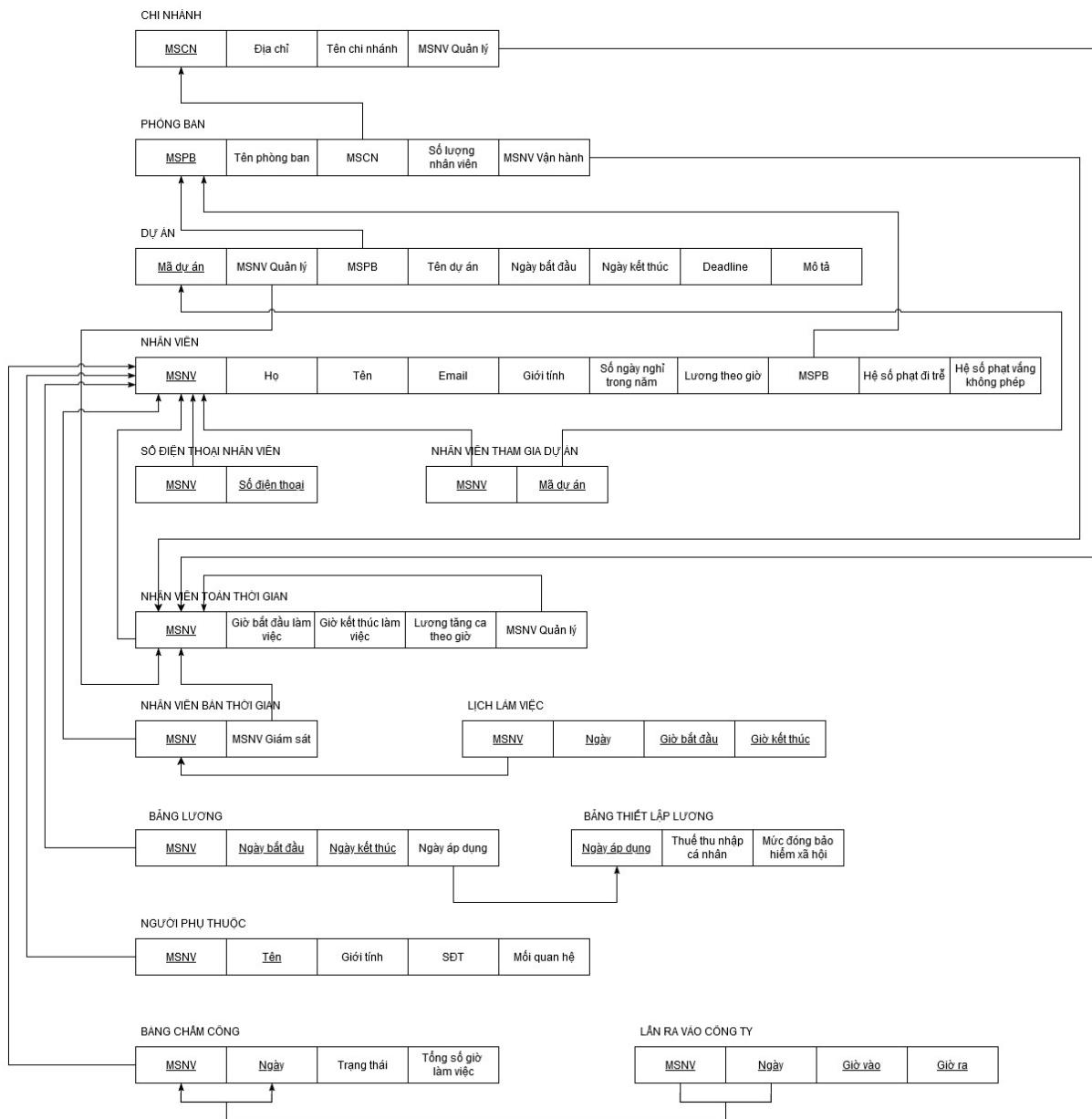
1 Danh sách thành viên & Phân chia công việc

STT	Họ tên	MSSV	Công việc đảm nhận	Ký tên
1	Nguyễn Danh Thành	2114782	Các thủ tục ở mục 3.1	
			Hiện thực màn hình 1	
2	Nguyễn An Khang	2211441	Tạo bảng và khởi tạo dữ liệu	
			Các trigger ở mục 3.2	
3	Trần Anh Khôi	2211694	Tạo bảng và khởi tạo dữ liệu	
			Các hàm ở mục 3.4	
			Hiện thực màn hình 3	
4	Hồ Quốc Khuong	2211709	Các thủ tục ở mục 3.3	
			Hiện thực màn hình 2	

Bảng 1: Danh sách thành viên và phân chia công việc

2 Tạo bảng và dữ liệu mẫu

2.1 Sơ đồ mô hình dữ liệu quan hệ



Hình 1: Mô hình dữ liệu quan hệ của hệ thống

2.2 Hiệu thực các bảng và tạo dữ liệu mẫu

2.2.1 Bảng ChiNhanh

- Câu lệnh tạo bảng

```
1  CREATE TABLE ChiNhanh (
2      MaChiNhanh CHAR(4) PRIMARY KEY,
3      TenChiNhanh NVARCHAR(100) NOT NULL,
4      DiaChi NVARCHAR(200) NOT NULL,
5      MSNV_QuanLy CHAR(6)
6  );
7  -- Sau khi đã import đủ data thì thêm thuộc tính khóa ngoại
8  ALTER TABLE chinhanh
9      ADD FOREIGN KEY (MSNV_QuanLy) REFERENCES
    ↳ nhanvientoanthoigian(MaNV);
```

- Câu lệnh ràng buộc

```

1   -- Trigger để kiểm tra định dạng của dữ liệu trước khi thực hiện
2   → INSERT
3   DELIMITER //
4   CREATE TRIGGER IF NOT EXISTS checkChiNhanhFormat
5   BEFORE INSERT on ChiNhanh
6   FOR EACH ROW
7   BEGIN
8       IF NEW.MaChiNhanh NOT REGEXP '^CN[0-9]{2}$' THEN
9           SIGNAL SQLSTATE '45000'
10          SET MESSAGE_TEXT = 'Mã chi nhánh phải ở định dạng CNxx, với
11          → xx là hai chữ số bất kỳ';
12      END IF;
13
14      IF NEW.TenChiNhanh NOT REGEXP '^[:alnum:][:space:].[\\-]*$'
15          → THEN
16          SIGNAL SQLSTATE '45000'
17          SET MESSAGE_TEXT = 'Tên chi nhánh không được chứa ký tự đặc
18          → biệt';
19      END IF;
20
21      IF NEW.DiaChi NOT REGEXP '^[:alnum:][:space:].[\\-]*$' THEN
22          SIGNAL SQLSTATE '45000'
23          SET MESSAGE_TEXT = 'Địa chỉ không được chứa ký tự đặc biệt';
24      END IF;
25
26      IF (NEW.MSNV_QuanLy IS NOT NULL) AND (NOT EXISTS (SELECT 1 from
27          → nhanvien WHERE nhanvien.`MaNV` = NEW.MSNV_QuanLy)) THEN
28          SIGNAL SQLSTATE '45000'
29          SET MESSAGE_TEXT = 'Mã số nhân viên quản lý không tồn tại';
30      END IF;
31  END //
32  DELIMITER ;

```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO ChiNhanh (MaChiNhanh, TenChiNhanh, DiaChi)
2     VALUES
3       ('CN01', 'Chi nhánh Cần Thơ', '321 Đường Cần Thơ, Cần Thơ'),
4       ('CN02', 'Chi nhánh Hải Phòng', '654 Đường Hải Phòng, Hải
5         → Phòng'),
6       ('CN03', 'Chi nhánh Huế', '987 Đường Huế, Huế'),
7       ('CN04', 'Chi nhánh Vinh', '234 Đường Vinh, Nghệ An');

8   -- Cập nhật thuộc tính MSNV_QuanLy sau khi đã thêm dữ liệu ở bảng
9     → NhanVienToanThoiGian
10  UPDATE chinhanh SET `MSNV_QuanLy` = 'NV3058' WHERE `MaChiNhanh` =
11    → 'CN01';
12  UPDATE chinhanh SET `MSNV_QuanLy` = 'NV9225' WHERE `MaChiNhanh` =
13    → 'CN02';
14  UPDATE chinhanh SET `MSNV_QuanLy` = 'NV9433' WHERE `MaChiNhanh` =
15    → 'CN03';
16  UPDATE chinhanh SET `MSNV_QuanLy` = 'NV1452' WHERE `MaChiNhanh` =
17    → 'CN04';

```

- Kết quả dữ liệu của bảng

	* MaChiNhan char(4)	* TenChiNhanh varchar(100)	* DiaChi varchar(200)	MSNV_Quar char(6)
	Filter	Filter	Filter	Filter
>	CN01	Chi nhánh Cần Thơ	321 Đường Cần Thơ, Cần Thơ	NV3058
>	CN02	Chi nhánh Hải Phòng	654 Đường Hải Phòng, Hải Phòng	NV9225
>	CN03	Chi nhánh Huế	987 Đường Huế, Huế	NV9433
>	CN04	Chi nhánh Vinh	234 Đường Vinh, Nghệ An	NV1452

Hình 2: Dữ liệu bảng ChiNhanh

2.2.2 Bảng PhongBan

- Câu lệnh tạo bảng

```
1   CREATE TABLE PhongBan (
2       MaPhongBan CHAR(6) PRIMARY KEY,
3       TenPhongBan NVARCHAR(100) NOT NULL,
4       MaChiNhanh CHAR(4) NOT NULL,
5       SoLuongNhanVien INT DEFAULT 0,
6       MSNV_VanHanh CHAR(6),
7       FOREIGN KEY (MaChiNhanh) REFERENCES ChiNhanh(MaChiNhanh)
8   );
9   -- Thêm thuộc tính khóa ngoại sau khi import đủ thông tin ở bảng
10  -- khác
11 ALTER TABLE phongban
12     ADD FOREIGN KEY (MSNV_VanHanh) REFERENCES
13         nhanvientoanthoigian(MaNV);
```

- Câu lệnh ràng buộc

```

1   CREATE TRIGGER IF NOT EXISTS checkPhongBanFormat
2     BEFORE INSERT on PhongBan
3     FOR EACH ROW
4     BEGIN
5       IF NEW.MaPhongBan NOT REGEXP '^PB[0-9]{4}$' THEN
6         SIGNAL SQLSTATE '45000'
7         SET MESSAGE_TEXT = 'Mã phòng ban phải ở định dạng PBxxxx,
8           ↪ với xxxx là bốn chữ số bất kỳ';
9         END IF;
10
11      IF NEW.TenPhongBan NOT REGEXP '^[:alnum:][:space:].[\\-]*$'
12        ↪ THEN
13        SIGNAL SQLSTATE '45000'
14        SET MESSAGE_TEXT = 'Tên phòng ban không được chứa ký tự đặc
15           ↪ biệt';
16        END IF;
17
18      IF NOT EXISTS (SELECT 1 FROM chinhhanh where
19        ↪ chinhhanh.`MaChiNhanh`=NEW.MaChiNhanh) THEN
20        SIGNAL SQLSTATE '45000'
21        SET MESSAGE_TEXT = 'Mã chi nhánh không tồn tại';
22        END IF;
23
24      IF (NEW.MSNV_VanHanh IS NOT NULL) AND (NOT EXISTS (SELECT 1 from
25        ↪ nhanvien WHERE nhanvien.`MaNV`=NEW.MSNV_VanHanh)) THEN
26        SIGNAL SQLSTATE '45000'
27        SET MESSAGE_TEXT = 'Mã số nhân viên vận hành không tồn tại';
28        END IF;
29
30      END //
```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO PhongBan (MaPhongBan, TenPhongBan, MaChiNhanh)
2     VALUES
3       ('PB0101', 'Phòng Kế toán', 'CN01'),
4       ('PB0102', 'Phòng Nhân sự', 'CN01'),
5       ('PB0103', 'Phòng Kỹ thuật', 'CN01'),
6       ('PB0104', 'Phòng IT', 'CN01'),
7       ('PB0105', 'Phòng Marketing', 'CN01'),
8       ...
9
10      -- Cập nhật thuộc tính MSNV_VanHanh sau khi đã thêm dữ liệu ở bảng
11      -- NhanVienToanThoiGian
12      UPDATE phongban SET `MSNV_VanHanh` = 'NV8120' WHERE `MaPhongBan` =
13          → 'PB0101';
14      UPDATE phongban SET `MSNV_VanHanh` = 'NV6786' WHERE `MaPhongBan` =
15          → 'PB0102';
16      UPDATE phongban SET `MSNV_VanHanh` = 'NV7040' WHERE `MaPhongBan` =
17          → 'PB0103';
18      UPDATE phongban SET `MSNV_VanHanh` = 'NV4036' WHERE `MaPhongBan` =
19          → 'PB0104';
20      UPDATE phongban SET `MSNV_VanHanh` = 'NV1404' WHERE `MaPhongBan` =
21          → 'PB0105';
22      ...

```

- Kết quả dữ liệu của bảng

* MaPhongBan char(6)	* TenPhongBan varchar(100)	* MaChiNhanh char(4)	SoLuongNhanVien int	MSNV_VanHanh char(6)
Filter	Filter	Filter	Filter	Filter
PB0101	Phòng Kế toán	CN01	5	NV8120
PB0102	Phòng Nhân sự	CN01	7	NV6786
PB0103	Phòng Kỹ thuật	CN01	9	NV7040
PB0104	Phòng IT	CN01	4	NV4036
PB0105	Phòng Marketing	CN01	3	NV1404
PB0106	Phòng Chăm sóc khách hàng	CN01	11	NV3614
PB0201	Phòng Kế toán	CN02	4	NV9225
PB0202	Phòng Nhân sự	CN02	13	NV1472
PB0203	Phòng Kỹ thuật	CN02	7	NV5136
PB0204	Phòng IT	CN02	7	NV3795

Hình 3: Dữ liệu bảng PhongBan

2.2.3 Bảng NhanVien

- Câu lệnh tạo bảng

```
1   CREATE TABLE NhanVien (
2       MaNV CHAR(6) PRIMARY KEY,
3       Ho NVARCHAR(10) NOT NULL,
4       TenLot NVARCHAR(10),
5       Ten NVARCHAR(10) NOT NULL,
6       GioiTinh ENUM('Nam', 'Nữ', 'Khác') NOT NULL,
7       Email VARCHAR(100) NOT NULL UNIQUE,
8       HeSoPhatDiTre DECIMAL(5, 2) DEFAULT 1,
9       HeSoPhatVangKhongPhep DECIMAL(5, 2) DEFAULT 1,
10      SoNgayNghi INT DEFAULT 12,
11      LuongTheoGio DECIMAL(10, 2) NOT NULL,
12      MaPhongBan CHAR(6),
13      FOREIGN KEY (MaPhongBan) REFERENCES PhongBan(MaPhongBan)
14  );
```

- Câu lệnh ràng buộc

```

1  DELIMITER //
2  CREATE TRIGGER IF NOT EXISTS checkNhanVienFormat
3    BEFORE INSERT ON nhanvien
4    FOR EACH ROW
5    BEGIN
6      IF NEW.MaNV NOT REGEXP '^NV[0-9]{4}$' THEN
7          SIGNAL SQLSTATE '45000'
8          SET MESSAGE_TEXT = 'Mã nhân viên phải có định dạng NVxxxxx,
9             → với 4 chữ số đằng sau';
10     END IF;
11
12     IF NEW.Ho NOT REGEXP '^[[:alnum:]]*$' THEN
13         SIGNAL SQLSTATE '45000'
14         SET MESSAGE_TEXT = 'Họ chỉ gồm một từ và không được chứa ký
15             → tự đặc biệt';
16     END IF;
17
18     IF NEW.TenLot NOT REGEXP '^[[:alnum:]][[:space:]]*$' THEN
19         SIGNAL SQLSTATE '45000'
20         SET MESSAGE_TEXT = 'Tên lót không được chứa ký tự đặc biệt';
21     END IF;
22
23     IF NEW.Ten NOT REGEXP '^[[:alnum:]]*$' THEN
24         SIGNAL SQLSTATE '45000'
25         SET MESSAGE_TEXT = 'Tên chỉ gồm một từ và không được chứa ký
26             → tự đặc biệt';
27     END IF;
28
29     IF NEW.Email NOT REGEXP
30       → '^[[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}+$' THEN
31         SIGNAL SQLSTATE '45000'
32         SET MESSAGE_TEXT = 'Email không đúng định dạng.';
33     END IF;
34
35     ...
36     IF EXISTS (SELECT 1 FROM nhanvien WHERE nhanvien.Email =
37       → NEW.Email) THEN
38         SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Email nhân viên
39             → đã tồn tại';
40     END IF;
41
42     END //
43     DELIMITER ;

```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO NhanVien (MaNV, Ho, TenLot, Ten, GioiTinh, Email,
2     → LuongTheoGio, MaPhongBan)
3   VALUES
4     ('NV8058', 'Văn', 'Phú', 'Ân', 'Nam', 'vanphuan@gmail.com',
5       → 90000, 'PB0106'),
6     ('NV9750', 'Kim', 'Hải', 'Sơn', 'Nam', 'kimhaison@gmail.com',
7       → 60000, 'PB0205'),
8     ('NV2034', 'Thi', 'Dũng', 'Trí', 'Nữ', 'thidungtri@gmail.com',
9       → 80000, 'PB0301'),
10    ('NV2790', 'Tô', 'Đức', 'Duy', 'Nam', 'toducduy@gmail.com',
11      → 25000, 'PB0401'),
12    ('NV1452', 'Lực', 'Thuận', 'Thành', 'Khác',
13      → 'lucthuanthanh@gmail.com', 100000, 'PB0402'),
14    ('NV9714', 'Sái', 'Thế', 'An', 'Nữ', 'saithean@gmail.com',
15      → 60000, 'PB0204'),
16    ('NV9858', 'Nguyễn', 'Việt', 'Khôi', 'Khác',
17      → 'nguyenvietkhoi@gmail.com', 50000, 'PB0106'),
18    ('NV7733', 'Đoàn', 'Hữu', 'Trung', 'Nữ',
19      → 'doanhhuutrung@gmail.com', 25000, 'PB0101'),
20    ('NV9240', 'Phạm', 'Anh', 'Tài', 'Khác',
21      → 'phamanhtai@gmail.com', 40000, 'PB0206')
22    ...;

```

- Kết quả dữ liệu của bảng

* MaNV	* Họ	* Họ	TênL	TênL	Ten	GioiTinh	Email	HeSoPhatDiTre	HeSoPhatVangKhongPhep	SoNgayNghi	* LuongTheoGio	MaPhongBan
char(6)	varchar	varchar	varcl	varcl	varchar	enum('Nam','Nữ')	varchar(100)	decimal(5,2)	decimal(5,2)	int	decimal(10,2)	char(6)
NV1108	Bùi	Bích	Như	Nam	bulbichnhu@gmail.com	Nam	bulbichnhu@gmail.com	1.00	1.00	12	25000.00	PB0403
NV1234	Lực	Quang	Hưng	Nữ	lucquanghung@gmail.com	Nữ	lucquanghung@gmail.com	1.00	1.00	12	60000.00	PB0405
NV1252	Châu	Phương	Chi	Nữ	chauphuongchi@gmail.com	Nữ	chauphuongchi@gmail.com	1.00	1.00	12	35000.00	PB0301
NV1271	Trần	Hồng	Trúc	Nữ	tranhongtruc@gmail.com	Nữ	tranhongtruc@gmail.com	1.00	1.00	12	90000.00	PB0301
NV1282	Lương	Điệu	Ái	Khác	luongdieuai@gmail.com	Khác	luongdieuai@gmail.com	1.00	1.00	12	70000.00	PB0206
NV1371	Nguyễn	Thúy	Anh	Nam	nguyenthuyanh@gmail.com	Nam	nguyenthuyanh@gmail.com	1.00	1.00	12	80000.00	PB0301
NV1404	La	Khải	Cá	Nam	lakhaica@gmail.com	Nam	lakhaica@gmail.com	1.00	1.00	12	70000.00	PB0105
NV1452	Lực	Thuận	Thành	Khác	lucthuanthanh@gmail.com	Khác	lucthuanthanh@gmail.com	1.00	1.00	12	100000.00	PB0402
NV1472	Phạm	Khắc	Minh	Khác	phamkhacminh@gmail.com	Khác	phamkhacminh@gmail.com	1.00	1.00	12	90000.00	PB0202
NV1492	Lực	Thanh	Tuấn	Nữ	lucthanhtuan@gmail.com	Nữ	lucthanhtuan@gmail.com	1.00	1.00	12	100000.00	PB0203

Hình 4: Dữ liệu bảng NhanVien

2.2.4 Bảng Sdt_NhanVien

- Câu lệnh tạo bảng

```

1   CREATE TABLE Sdt_NhanVien (
2       MaNV CHAR(6),
3       SoDienThoai VARCHAR(10),
4       PRIMARY KEY (MaNV, SoDienThoai),
5       Foreign Key (MaNV) REFERENCES nhanvien(MaNV)
6   );

```

- Câu lệnh ràng buộc

```

1   CREATE Trigger if NOT EXISTS checkSdtFormat
2       BEFORE INSERT ON Sdt_NhanVien
3       FOR EACH ROW
4       BEGIN
5           IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
6                           → nhanvien.`MaNV` = NEW.MaNV)) THEN
7               SIGNAL SQLSTATE '45000'
8               SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
9           END IF;
10
11
12           IF NEW.SoDienThoai NOT REGEXP '^0[0-9]{9}' THEN
13               SIGNAL SQLSTATE '45000'
14               SET MESSAGE_TEXT = 'Số điện thoại phải bắt đầu bằng số 0 và
15                           → có chính xác 10 chữ số';
16           END IF;
17
18
19           IF EXISTS (SELECT 1 from nhanvien NATURAL INNER JOIN
20                           → Sdt_NhanVien WHERE `MaNV` = NEW.MaNV AND
21                           → `SoDienThoai` = NEW.SoDienThoai) THEN
22               SIGNAL SQLSTATE '45000'
23               SET MESSAGE_TEXT = 'Nhân viên vừa nhập đã có số điện thoại
24                           → trên';
25           END IF;
26       END //
```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO sdt_nhanvien
2     VALUES
3       ('NV8058', '0649835291'),
4       ('NV9750', '0429234781'),
5       ('NV2034', '0111708152'),
6       ('NV2790', '0720862287'),
7       ('NV1452', '0819584592'),
8       ('NV9714', '0688341035'),
9       ('NV9858', '0235295264'),
10      ('NV7733', '0437707384'),
11      ('NV9240', '0248130737'),
12      ('NV3993', '0290786034'),
13      ... ;

```

- Kết quả dữ liệu của bảng

* MaNV	* SoDienThoai
char(6)	varchar(10)
Filter	Filter
NV1108	0756211072
NV1108	0995508868
NV1234	0263148435
NV1234	0306888516
NV1234	0421132825
NV1252	0547056594
NV1252	0853769290
NV1271	0605223195
NV1282	0297851155
NV1371	0710842471

Hình 5: Dữ liệu bảng Sdt_NhanVien

2.2.5 Bảng NhanVienToanThoiGian

- Câu lệnh tạo bảng

```
1   CREATE TABLE NhanVienToanThoiGian (
2       MaNV CHAR(6) PRIMARY KEY,
3       GioBatDau TIME,
4       GioKetThuc TIME,
5       LuongTangCa DECIMAL(10, 2),
6       MaNVQuanLy CHAR(6) DEFAULT NULL,
7           FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),
8           Foreign Key (MaNVQuanLy) REFERENCES NhanVienToanThoiGian(MaNV)
9   );
```

- Câu lệnh ràng buộc

```
1   -- Tối thêm sau
```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO nhanvientoanthoigian(`MaNV`, `LuongTangCa`,
2     → `GioBatDau`, `GioKetThuc`)
VALUES
3   ('NV1234', 90000.0, '08:00:00', '17:00:00'),
4   ('NV1271', 135000.0, '08:00:00', '17:00:00'),
5   ('NV1282', 105000.0, '08:00:00', '17:00:00'),
6   ('NV1371', 120000.0, '08:00:00', '17:00:00'),
7   ('NV1404', 105000.0, '08:00:00', '17:00:00'),
8   ('NV1452', 150000.0, '08:00:00', '17:00:00'),
9   ('NV1472', 135000.0, '08:00:00', '17:00:00'),
10  ('NV1492', 150000.0, '08:00:00', '17:00:00'),
11  ('NV1778', 135000.0, '08:00:00', '17:00:00'),
12  ('NV1816', 75000.0, '08:00:00', '17:00:00),
13  ...;
```

- Kết quả dữ liệu của bảng

* MaNV char(6)	GioBatDau time	GioKetThuc time	LuongTangCa decimal(10,2)	MaNVQuanLy char(6)
Filter	Filter	Filter	Filter	Filter
NV1234	08:00:00	17:00:00	90000.00	NV2494
NV1271	08:00:00	17:00:00	135000.00	NV9433
NV1282	08:00:00	17:00:00	105000.00	NV3265
NV1371	08:00:00	17:00:00	120000.00	NV1271
NV1404	08:00:00	17:00:00	105000.00	NV3058
NV1452	08:00:00	17:00:00	150000.00	NV5031
NV1472	08:00:00	17:00:00	135000.00	NV9225
NV1492	08:00:00	17:00:00	150000.00	NV5136
NV1778	08:00:00	17:00:00	135000.00	NV1472
NV1816	08:00:00	17:00:00	75000.00	NV7040

Hình 6: Dữ liệu bảng NhanVienToanThoiGian

2.2.6 Bảng NhanVienBanThoiGian

- Câu lệnh tạo bảng

```

1   CREATE TABLE NhanVienBanThoiGian (
2       MaNV CHAR(6) PRIMARY KEY,
3       MaNV_GiamSat CHAR(6),
4       FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),
5       Foreign Key (MaNV_GiamSat) REFERENCES NhanVienToanThoiGian(MaNV)
6   );

```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO nhanvienbanthoigian(`MaNV`, `MaNV_GiamSat`)
2   VALUES
3       ('NV7733', 'NV9606'), ('NV8232', 'NV9606'),
4       ('NV9199', 'NV9606'), ('NV8084', 'NV6786'),
5       ('NV1540', 'NV4355'), ('NV8519', 'NV4355'),
6       ('NV9375', 'NV1404'), ('NV3184', 'NV2357'),
7       ('NV7938', 'NV2357'), ('NV9007', 'NV2357'),
8       ...;

```

- Kết quả dữ liệu của bảng

MaNV char(6)	MaNV_GiamSat char(6)
Filter	Filter
NV1252	NV1371
NV9375	NV1404
NV2099	NV1967
NV6452	NV1967
NV6043	NV2105
NV9741	NV2105
NV3184	NV2357
NV7938	NV2357
NV9007	NV2357
NV3140	NV2494

Hình 7: Dữ liệu bảng NhanVienBanThoiGian

2.2.7 Bảng DuAn

- Câu lệnh tạo bảng

```
1   CREATE TABLE DuAn (
2       MaDuAn CHAR(6) PRIMARY KEY,
3       TenDuAn NVARCHAR(100) NOT NULL,
4       MoTa TEXT,
5       NgayBatDau DATE NOT NULL,
6       NgayKetThuc DATE,
7       Deadline DATE NOT NULL,
8       MaPhongBan CHAR(6),
9       MaQuanLy CHAR(6),
10      FOREIGN KEY (MaPhongBan) REFERENCES PhongBan(MaPhongBan),
11      FOREIGN KEY (MaQuanLy) REFERENCES NhanVienToanThoiGian(MaNV)
12  );
```

- Câu lệnh ràng buộc

```

1  CREATE TRIGGER checkDuAnFormat
2   BEFORE INSERT ON DuAn
3   FOR EACH ROW
4   BEGIN
5     IF NEW.MaDuAn NOT REGEXP '^DA[0-9]{4}$' THEN
6       SIGNAL SQLSTATE '45000'
7       SET MESSAGE_TEXT = 'Mã dự án phải có định dạng DAxxxx, với 4
8       ↪ chữ số đăng sau';
9     END IF;
10
11    IF EXISTS (SELECT 1 from duan WHERE duan.MaDuAn=NEW.MaDuAn) THEN
12      SIGNAL SQLSTATE '45000'
13      SET MESSAGE_TEXT = 'Mã dự án đã tồn tại';
14    END IF;
15
16    IF NEW.TenDuAn NOT REGEXP '^[:alnum:][:space:].[\\-]*$' THEN
17      SIGNAL SQLSTATE '45000'
18      SET MESSAGE_TEXT = 'Tên dự án không được chứa ký tự đặc
19       ↪ biệt';
20    END IF;
21
22    IF NEW.MoTa NOT REGEXP '^[:alnum:][:space:][:punct:]*$' THEN
23      SIGNAL SQLSTATE '45000'
24      SET MESSAGE_TEXT = 'Mô tả không được chứa ký tự đặc biệt';
25    END IF;
26
27    IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
28      ↪ nhanvien.`MaNV`=NEW.MaQuanLy)) THEN
29      SIGNAL SQLSTATE '45000'
30      SET MESSAGE_TEXT = 'Mã số nhân viên quản lý dự án không tồn
31       ↪ tại';
32    END IF;
33
34    IF (NOT EXISTS (SELECT 1 from phongban WHERE
35      ↪ phongban.`MaPhongBan`=NEW.MaPhongBan)) THEN
36      SIGNAL SQLSTATE '45000'
37      SET MESSAGE_TEXT = 'Mã phòng ban không tồn tại';
38    END IF;
39
40  END //

```

- Câu lệnh thêm dữ liệu

```
1      INSERT INTO duan(`MaDuAn`, `TenDuAn`, `MoTa`, `NgayBatDau`,  
2      ↵    `Deadline`, `NgayKetThuc`, `MaPhongBan`, `MaQuanLy`)  
3      VALUES  
4          ('DA0001', 'Theodore', 'Lorem ipsum d...t non varius.',  
5          ↵    '2023-12-01', '2024-01-08', '2024-01-05', 'PB0101',  
6          ↵    'NV8120'),  
7          ('DA0002', 'Daphne', 'Sed arcu met...bus molestie.',  
8          ↵    '2024-01-10', '2024-01-25', NULL, 'PB0102', 'NV3058'),  
9          ('DA0003', 'Cora', 'Nulla plac...inibus.', '2024-02-10',  
10         ↵    '2024-03-13', '2024-03-13', 'PB0103', 'NV1816'),  
11         ('DA0004', 'Atticus', 'Praesent id m...nenatis.', '2024-03-15',  
12         ↵    '2024-03-23', '2024-03-23', 'PB0104', 'NV4036'),  
13         ('DA0005', 'Iris', 'Curabit...lectus.', '2024-03-20',  
14         ↵    '2024-04-15', '2024-04-15', 'PB0105', 'NV1404'),  
15         ('DA0006', 'Atlas', 'Vivamu...lla egestas dignissim.',  
16         ↵    '2024-03-21', '2024-04-23', '2024-04-23', 'PB0106',  
17         ↵    'NV2357'),  
18         ('DA0007', 'Phoebe', 'Aenean nunc ...in semper nisl.',  
19         ↵    '2024-04-23', '2024-04-30', '2024-04-28', 'PB0201',  
20         ↵    'NV9225'),  
21         ('DA0008', 'Ophelia', 'Curabi...dictum tortor.', '2024-04-30',  
22         ↵    '2024-05-16', '2024-05-10', 'PB0202', 'NV1472'),  
23         ('DA0009', 'Elias', 'Cras tempor ma...alesuada fames ac turpis  
24         ↵    egestas.', '2024-05-10', '2024-06-05', '2024-06-05',  
25         ↵    'PB0203', 'NV1492'),  
26         ('DA0010', 'Zephyr', 'Integer ne...ces suscipit.',  
27         ↵    '2024-05-13', '2024-06-23', '2024-06-23', 'PB0204',  
28         ↵    'NV3795'),  
29         ...;
```

- Kết quả dữ liệu của bảng

* MaDuAn char(6)	* TenDuAn varchar(100)	MoTa text	* NgayBatDau date	NgayKetThuc date	* Deadline date	MaPhongBan char(6)	MaQuanLy char(6)
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
DA0001	Theodore	Lorem ipsum dolor sit amet	2023-12-01	2024-01-05	2024-01-08	PB0101	NV8120
DA0002	Daphne	Sed arcu metus, commodo	2024-01-10	(NULL)	2024-01-25	PB0102	NV3058
DA0003	Cora	Nulla placerat fermentum ju	2024-02-10	2024-03-13	2024-03-13	PB0103	NV1816
DA0004	Atticus	Praesent id massa at ipsum	2024-03-15	2024-03-23	2024-03-23	PB0104	NV4036
DA0005	Iris	Curabitur quis semper dui. /	2024-03-20	2024-04-15	2024-04-15	PB0105	NV1404
DA0006	Atlas	Vivamus scelerisque vitae ri	2024-03-21	2024-04-23	2024-04-23	PB0106	NV2357
DA0007	Phoebe	Aenean nunc erat, blandit u	2024-04-23	2024-04-28	2024-04-30	PB0201	NV9225
DA0008	Ophelia	Curabitur vehicula neque a	2024-04-30	2024-05-10	2024-05-16	PB0202	NV1472
DA0009	Elias	Cras tempor mauris libero, i	2024-05-10	2024-06-05	2024-06-05	PB0203	NV1492
DA0010	Zephyr	Integer nec lorem placerat,	2024-05-13	2024-06-23	2024-06-23	PB0204	NV3795

Hình 8: Dữ liệu của bảng DuAn

2.2.8 Bảng NhanVienThamGiaDuAn

- Câu lệnh tạo bảng

```

1   CREATE TABLE NhanVienThamGiaDuAn (
2       MaNhanVien CHAR(6),
3       MaDuAn CHAR(6),
4       PRIMARY KEY (MaNhanVien, MaDuAn),
5       Foreign Key (MaNhanVien) REFERENCES NhanVien(MaNV),
6       Foreign Key (MaDuAn) REFERENCES DuAn(MaDuAn)
7   );

```

- Câu lệnh ràng buộc

```

1   CREATE TRIGGER checkNhanVienThamGiaDuAnFormat
2       BEFORE INSERT ON NhanVienThamGiaDuAn
3       FOR EACH ROW
4       BEGIN
5           IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
6               → nhanvien.`MaNV` = NEW.MaNhanVien)) THEN
7               SIGNAL SQLSTATE '45000'
8               SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
9           END IF;
10
11          IF (NOT EXISTS (SELECT 1 from DuAn WHERE
12              → DuAn.`MaDuAn` = NEW.MaDuAn)) THEN
13              SIGNAL SQLSTATE '45000'
14              SET MESSAGE_TEXT = 'Mã số dự án không tồn tại';
15          END IF;
16
17          IF EXISTS (SELECT 1 FROM NhanVienThamGiaDuAn WHERE
18              → NhanVienThamGiaDuAn.`MaNhanVien` = NEW.`MaNhanVien` AND
19              → NhanVienThamGiaDuAn.`MaDuAn` = NEW.`MaDuAn`) THEN
20              SIGNAL SQLSTATE '45000'
21              SET MESSAGE_TEXT = 'Nhân viên vừa nhập đã tham gia dự án
22                  trên';
23          END IF;
24      END //
```

- Câu lệnh thêm dữ liệu

```

1
2 INSERT INTO nhanvienthamgiaduan(`MaDuAn`, `MaNhanVien`)
3 VALUES
4   ('DA0001', 'NV8120'),
5   ('DA0002', 'NV3058'),
6   ('DA0003', 'NV1816'),
7   ('DA0004', 'NV4036'),
8   ('DA0005', 'NV1404'),
9   ('DA0006', 'NV2357'),
10  ('DA0007', 'NV9225'),
11  ('DA0008', 'NV1472'),
12  ('DA0009', 'NV1492'),
13  ('DA0010', 'NV3795'),
14  ('DA0011', 'NV9750'),
15  ('DA0012', 'NV7288'),
16  ...

```

- Kết quả dữ liệu của bảng

* ↗ MaNhan' ↓ ↘ char(6)	* ↗ MaD' ↓ ↘ char(6)
Filter	Filter
NV5235	DA0001
NV8112	DA0001
NV8120	DA0001
NV8207	DA0001
NV8232	DA0001
NV9741	DA0001
NV3058	DA0002
NV3709	DA0002
NV5775	DA0002
NV6452	DA0002

Hình 9: Dữ liệu của bảng NhanVienThamGiaDuAn

2.2.9 Bảng NguoiPhuThuoc

- Câu lệnh tạo bảng

```
1 CREATE TABLE NguoiPhuThuoc (
2     Ten NVARCHAR(50),
3     GioiTinh ENUM('Nam', 'Nữ', 'Khác') NOT NULL,
4     SoDienThoai VARCHAR(10) NOT NULL,
5     MoiQuanHe NVARCHAR(50) NOT NULL,
6     MaNV CHAR(6),
7     PRIMARY KEY (Ten, MaNV),
8     FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)
9 );
```

- Câu lệnh ràng buộc

```

1   CREATE TRIGGER IF NOT EXISTS checkNguoiPhuThuocFormat
2     BEFORE INSERT on NguoiPhuThuoc
3     FOR EACH ROW
4     BEGIN
5       IF NEW.Ten NOT REGEXP '^[[:alnum:][:space:]]*$' THEN
6         SIGNAL SQLSTATE '45000'
7         SET MESSAGE_TEXT = 'Tên người phụ thuộc không được chứa
8           ↳ những ký tự đặc biệt';
9         END IF;
10
11      IF NEW.MoiQuanHe NOT REGEXP '^[[:alnum:][:space:]]*$' THEN
12        SIGNAL SQLSTATE '45000'
13        SET MESSAGE_TEXT = 'Tên mối quan hệ không được chứa những ký
14           ↳ tự đặc biệt';
15        END IF;
16
17      IF NEW.SoDienThoai NOT REGEXP '^0[0-9]{9}' THEN
18        SIGNAL SQLSTATE '45000'
19        SET MESSAGE_TEXT = 'Số điện thoại phải bắt đầu bằng số 0 và
20           ↳ có chính xác 10 chữ số';
21        END IF;
22
23      IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
24           ↳ nhanvien.MaNV=NEW.MaNV)) THEN
25        SIGNAL SQLSTATE '45000'
26        SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
27      END IF;
28    END //
```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO nguoiPhuThuoc(`MaNV`, `Ten`, `SoDienThoai`, `GioiTinh`,
2     → `MoiQuanHe`)
3   VALUES
4     ('NV7733', 'Nguyễn Thị Mỹ Ngọc', '0318269338', 'Nữ', 'Chồng'),
5     ('NV8120', 'Nguyễn Lê Hoài Ân', '0891190428', 'Nữ', 'Cha'),
6     ('NV8232', 'Nguyễn Như Mười', '0452543489', 'Nữ', 'Cha'),
7     ('NV9199', 'Lê Anh Hải', '0796135045', 'Nữ', 'Mẹ'),
8     ('NV9606', 'Huỳnh Hữu Thoại', '0164997442', 'Nam', 'Con'),
9     ('NV3058', 'Mai Trúc Anh', '0341137858', 'Nữ', 'Chồng'),
10    ('NV3189', 'Ngô Minh Mẫn', '0839219566', 'Nữ', 'Vợ'),
11    ('NV3308', 'Lê Quyết Tiến', '0229990080', 'Nữ', 'Mẹ'),
12    ('NV3577', 'Nguyễn Văn Dũng', '0130696033', 'Nữ', 'Vợ'),
13    ('NV4286', 'Trần Công Danh', '0464749368', 'Nam', 'Chồng'),
14    ...
15

```

- Kết quả dữ liệu của bảng

* Ten varchar(50)	* GioiTinh enum('Nam','Nữ','Khác')	* SoDienThoai varchar(10)	* MoiQuanHe varchar(50)	* MaN char(6)
Filter	Filter	Filter	Filter	Filter
Đỗ Nhật Nam	Nữ	0566551802	Con	NV1108
Nguyễn Cao Thiên Trí	Nữ	0781893349	Cha	NV1234
Lê Nguyễn Đông Triều	Nam	0461605454	Cha	NV1234
Lê Văn Quảng	Nữ	0274955100	Con	NV1252
Tạ Hoài Nam	Nam	0842457249	Mẹ	NV1271
Lê Hoàng Hữu Nghị	Nữ	0690565824	Cha	NV1282
Phạm Thanh Vũ	Nữ	0275350125	Cha	NV1371
Phan Thị Thùy Trúc	Nam	0893547294	Chồng	NV1404
Lê Ngọc Thi	Nữ	0242702035	Mẹ	NV1452
Đào Tiến Lâm	Nữ	0773321017	Cha	NV1472

Hình 10: Dữ liệu của bảng NguoiPhuThuoc

2.2.10 Bảng BangChamCong

- Câu lệnh tạo bảng

```
1   CREATE TABLE BangChamCong (
2       MaNV CHAR(6),
3       Ngay DATE,
4       TrangThai ENUM('Có mặt', 'Vắng có phép', 'Vắng không phép'),
5       TongSoGioLam DECIMAL(5, 2) DEFAULT 0,
6       PRIMARY KEY (MaNV, Ngay),
7       FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)
8   );
```

- Câu lệnh ràng buộc

```
1   CREATE TRIGGER checkBangChamCongFormat
2       BEFORE INSERT ON BangChamCong
3       FOR EACH ROW
4       BEGIN
5           IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
6                           nhanvien.`MaNV` = NEW.MaNV)) THEN
7               SIGNAL SQLSTATE '45000'
8               SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
9           END IF;
10      END //
```

- Câu lệnh thêm dữ liệu

```

1  INSERT INTO bangchamcong(`MaNV`, `Ngay`, `TrangThai`)
2  VALUES
3      ('NV1234', '2024-10-21', 'Có mặt'),
4      ('NV1234', '2024-10-22', 'Có mặt'),
5      ('NV1234', '2024-10-23', 'Có mặt'),
6      ('NV1234', '2024-10-24', 'Có mặt'),
7      ('NV1234', '2024-10-25', 'Có mặt'),
8      ('NV1234', '2024-10-28', 'Có mặt'),
9      ('NV1234', '2024-10-29', 'Vắng không phép'),
10     ('NV1234', '2024-10-30', 'Vắng không phép'),
11     ('NV1234', '2024-10-31', 'Có mặt'),
12     ('NV1234', '2024-11-01', 'Có mặt'),
13     ...

```

- Kết quả dữ liệu của bảng

* MaNV	Ngay	TrangThai	TongSoGioLam
char(6)	date	enum('Có mặt','Vắng có phép')	decimal(5,2)
Filter	Filter	Filter	Filter
NV1108	2024-10-22	Có mặt	3.57
NV1108	2024-10-29	Có mặt	4.00
NV1108	2024-11-01	Có mặt	4.00
NV1108	2024-11-04	Có mặt	2.88
NV1108	2024-11-05	Vắng có phép	0.00
NV1108	2024-11-07	Vắng không phép	0.00
NV1108	2024-11-11	Vắng có phép	0.00
NV1108	2024-11-14	Có mặt	2.87
NV1108	2024-11-15	Vắng không phép	0.00
NV1108	2024-11-18	Có mặt	4.00

Hình 11: Dữ liệu của bảng BangChamCong

2.2.11 Bảng LanRaVao

- Câu lệnh tạo bảng

```
1   CREATE TABLE LanRaVao (
2       MaNV CHAR(6),
3       Ngay DATE,
4       GioVao TIME,
5       GioRa TIME,
6       PRIMARY KEY (MaNV, Ngay, GioVao, GioRa),
7       FOREIGN KEY (MaNV, Ngay) REFERENCES BangChamCong(MaNV, Ngay)
8   );
```

- Câu lệnh ràng buộc

```
1   CREATE TRIGGER checkLanRaVaoFormat
2       BEFORE INSERT ON LanRaVao
3       FOR EACH ROW
4       BEGIN
5           IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
6                           nhanvien.`MaNV` = NEW.MaNV)) THEN
7               SIGNAL SQLSTATE '45000'
8               SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
9           END IF;
10      END //
```

- Câu lệnh thêm dữ liệu

```

1  INSERT INTO Lanravao (`MaNV`, `Ngay`, `GioVao`, `GioRa`)
2  VALUES
3      ('NV1234', '2024-10-21', '08:00', '12:00'),
4      ('NV1234', '2024-10-21', '13:00', '17:00'),
5      ('NV1234', '2024-10-22', '08:00', '12:00'),
6      ('NV1234', '2024-10-22', '13:00', '17:00'),
7      ('NV1234', '2024-10-23', '08:18', '12:00'),
8      ('NV1234', '2024-10-23', '13:00', '17:00'),
9      ('NV1234', '2024-10-24', '08:00', '11:07'),
10     ('NV1234', '2024-10-24', '13:00', '17:00'),
11     ('NV1234', '2024-10-25', '08:00', '12:00'),
12     ('NV1234', '2024-10-25', '13:00', '17:00'),
13     ...

```

- Kết quả dữ liệu của bảng

* MaNV	* Ngay	* GioVao	* GioRa
Filter	Filter	Filter	Filter
NV1108	2024-10-22	08:00:00	11:34:00
NV1108	2024-10-29	08:00:00	12:00:00
NV1108	2024-11-01	13:00:00	17:00:00
NV1108	2024-11-04	08:37:00	11:30:00
NV1108	2024-11-14	08:23:00	11:15:00
NV1108	2024-11-18	08:00:00	12:00:00
NV1108	2024-11-19	08:00:00	11:41:00
NV1108	2024-11-21	08:57:00	11:17:00
NV1108	2024-11-27	08:00:00	11:31:00
NV1108	2024-12-06	13:00:00	17:00:00

Hình 12: Dữ liệu của bảng LanRaVao

2.2.12 Bảng LichLamViec

- Câu lệnh tạo bảng

```
1   CREATE TABLE LichLamViec (
2       MaNV CHAR(6),
3       Ngay DATE,
4       GioBatDau TIME,
5       GioKetThuc TIME,
6       PRIMARY KEY (MaNV, Ngay, GioBatDau, GioKetThuc),
7       FOREIGN KEY (MaNV) REFERENCES NhanVienBanThoiGian(MaNV)
8   );
```

- Câu lệnh ràng buộc

```
1   CREATE TRIGGER checkLichLamViecFormat
2       BEFORE INSERT ON LichLamViec
3       FOR EACH ROW
4       BEGIN
5           IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
6                           nhanvien.`MaNV` = NEW.MaNV)) THEN
7               SIGNAL SQLSTATE '45000'
8               SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
9           END IF;
10      END //
```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO lichlamviec(`MaNV`, `Ngay`, `GioBatDau`, `GioKetThuc`)
2     VALUES
3       ('NV1252', '2024-10-21', '08:00', '12:00'),
4       ('NV1252', '2024-10-23', '13:00', '17:00'),
5       ('NV1252', '2024-10-24', '13:00', '17:00'),
6       ('NV1252', '2024-10-29', '08:00', '12:00'),
7       ('NV1252', '2024-11-01', '13:00', '17:00'),
8       ('NV1252', '2024-11-04', '08:00', '12:00'),
9       ('NV1252', '2024-11-06', '13:00', '17:00'),
10      ('NV1252', '2024-11-08', '08:00', '12:00'),
11      ('NV1252', '2024-11-12', '13:00', '17:00'),
12      ('NV1252', '2024-11-15', '08:00', '12:00'),
13      ...

```

- Kết quả dữ liệu của bảng

* MaNV char(6)	* Ngay date	* GioBatDau time	* GioKetThuc time
Filter	Filter	Filter	Filter
NV1108	2024-10-22	08:00:00	12:00:00
NV1108	2024-10-29	08:00:00	12:00:00
NV1108	2024-11-01	13:00:00	17:00:00
NV1108	2024-11-04	08:00:00	12:00:00
NV1108	2024-11-05	13:00:00	17:00:00
NV1108	2024-11-07	13:00:00	17:00:00
NV1108	2024-11-11	13:00:00	17:00:00
NV1108	2024-11-14	08:00:00	12:00:00
NV1108	2024-11-15	13:00:00	17:00:00
NV1108	2024-11-18	08:00:00	12:00:00

Hình 13: Dữ liệu của bảng LichLamViec

2.2.13 Bảng BangThietLapLuong

- Câu lệnh tạo bảng

```

1   CREATE TABLE BangThietLapLuong (
2       NgayApDung DATE PRIMARY KEY,
3       ThueSuat DECIMAL(5, 2),
4       BaoHiemXH DECIMAL(5, 2)
5   );

```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO bangthietlapluong(`NgayApDung`, `ThueSuat`, `BaoHiemXH`)
2   VALUES
3       ('2024-10-01', 0.10, 0.08),
4       ('2024-11-25', 0.08, 0.08);

```

- Kết quả dữ liệu của bảng

* NgayApDung date	ThueSuat decimal(5,2)	BaoHiemXH decimal(5,2)
Filter	Filter	Filter
2024-10-01	0.10	0.08
2024-11-25	0.08	0.08

Hình 14: Dữ liệu của bảng BangThietLapLuong

2.2.14 Bảng BangLuong

- Câu lệnh tạo bảng

```
1 CREATE TABLE BangLuong (
2     MaNV CHAR(6),
3     NgayBatDau DATE,
4     NgayKetThuc DATE,
5     NgayApDung DATE,
6     PRIMARY KEY (MaNV, NgayBatDau, NgayKetThuc),
7     FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV),
8     FOREIGN KEY (NgayApDung) REFERENCES
9         BangThietLapLuong(NgayApDung)
);
```

- Câu lệnh ràng buộc

```
1 CREATE TRIGGER checkBangLuongFormat
2 BEFORE INSERT ON BangLuong
3 FOR EACH ROW
4 BEGIN
5     IF (NOT EXISTS (SELECT 1 from nhanvien WHERE
6         nhanvien.`MaNV` = NEW.MaNV)) THEN
7         SIGNAL SQLSTATE '45000'
8         SET MESSAGE_TEXT = 'Mã số nhân viên không tồn tại';
9     END IF;
END //
```

- Câu lệnh thêm dữ liệu

```

1   INSERT INTO bangluong(`MaNV`, `NgayBatDau`, `NgayKetThuc`,
2     → `NgayApDung`)
3   VALUES
4     ('NV1108', '2024-10-21', '2024-10-31', '2024-10-01'),
5     ('NV1108', '2024-11-01', '2024-11-30', '2024-11-25'),
6     ('NV1234', '2024-10-21', '2024-10-31', '2024-10-01'),
7     ('NV1234', '2024-11-01', '2024-11-30', '2024-11-25'),
8     ('NV1252', '2024-10-21', '2024-10-31', '2024-10-01'),
9     ('NV1252', '2024-11-01', '2024-11-30', '2024-11-25'),
10    ('NV1271', '2024-10-21', '2024-10-31', '2024-10-01'),
11    ('NV1271', '2024-11-01', '2024-11-30', '2024-11-25'),
12    ('NV1282', '2024-10-21', '2024-10-31', '2024-10-01'),
13    ('NV1282', '2024-11-01', '2024-11-30', '2024-11-25'),
14    ...
15

```

- Kết quả dữ liệu của bảng

* MaNV	* NgayBatDau	* NgayKetThuc	NgayApDung
Filter	Filter	Filter	Filter
NV1108	2024-10-21	2024-10-31	2024-10-01
NV1234	2024-10-21	2024-10-31	2024-10-01
NV1252	2024-10-21	2024-10-31	2024-10-01
NV1271	2024-10-21	2024-10-31	2024-10-01
NV1282	2024-10-21	2024-10-31	2024-10-01
NV1371	2024-10-21	2024-10-31	2024-10-01
NV1404	2024-10-21	2024-10-31	2024-10-01
NV1452	2024-10-21	2024-10-31	2024-10-01
NV1472	2024-10-21	2024-10-31	2024-10-01
NV1492	2024-10-21	2024-10-31	2024-10-01

Hình 15: Dữ liệu của bảng BangLuong

3 Hiện thực ứng dụng

3.1 Thủ tục INSERT/UPDATE/DELETE dữ liệu vào 1 bảng dữ liệu

Bảng dữ liệu được chọn: bảng NhanVien

3.1.1 Thủ tục INSERT

Mô tả thủ tục: Thủ tục có nhiệm vụ thực hiện lệnh thêm vào bảng NhanVien thông tin của nhân viên. Thủ tục hoạt động qua hai quá trình:

- Đầu tiên là kiểm tra định dạng của các tham số được truyền vào, nếu sai định dạng thì sẽ thông báo lỗi và ngừng việc thực hiện thủ tục.
- Cuối cùng, sau khi đảm bảo về định dạng, thủ tục sẽ thêm vào bảng NhanVien một thực thể NhanVien mới, với các giá trị là các tham số được cung cấp.

Input:

- Tham số p_MaNV, có kiểu CHAR(6)
- Tham số p_Ho, có kiểu NVARCHAR(10)
- Tham số p_TenLot, có kiểu NVARCHAR(10)
- Tham số p_Ten, có kiểu NVARCHAR(10)
- Tham số p_GioiTinh, có kiểu ENUM('Nam', 'Nữ', 'Khác')
- Tham số p_Email, có kiểu VARCHAR(100)
- Tham số p_LuongTheoGio, có kiểu DECIMAL(10, 2)
- Tham số p_MaPhongBan, có kiểu CHAR(6)

Output: Không có. **Câu lệnh tạo thủ tục:**

```

1 CREATE PROCEDURE ThemNhanVien(
2     IN p_MaNV CHAR(6),
3     IN p_Ho NVARCHAR(10),
4     IN p_TenLot NVARCHAR(10),
5     IN p_Ten NVARCHAR(10),
6     IN p_GioiTinh ENUM('Nam', 'Nữ', 'Khác'),
7     IN p_Email VARCHAR(100),
8     IN p_LuongTheoGio DECIMAL(10, 2),
9     IN p_MaPhongBan CHAR(6)
10    )

```

```

11 BEGIN
12   -- Kiểm tra dữ liệu hợp lệ
13   IF p_MaNV NOT REGEXP '^NV[0-9]{4}$' THEN
14     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Mã nhân viên phải có
15     → định dạng NVxxxx, với 4 chữ số đằng sau';
16   END IF;
17   IF p_Ho REGEXP '[^a-zA-ZÀ-܂]' THEN
18     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Họ không được chứa
19     → số hoặc ký tự đặc biệt.';
20   END IF;
21   IF p_TenLot IS NOT NULL AND p_TenLot REGEXP '[^a-zA-ZÀ-܂]' THEN
22     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tên lót không được
23     → chứa số hoặc ký tự đặc biệt.';
24   END IF;
25   IF p_Ten REGEXP '[^a-zA-ZÀ-܂]' THEN
26     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tên không được chứa
27     → số hoặc ký tự đặc biệt.';
28   END IF;
29   IF p_Email NOT REGEXP
30     → '^[_a-zA-Z0-9.%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}+$' THEN
31     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Email không đúng
32     → định dạng.';
33   END IF;
34   -- Kiểm tra sự tồn tại của MaPhongBan trong bảng PhongBan
35   IF NOT EXISTS (SELECT 1 FROM PhongBan WHERE MaPhongBan =
36     → p_MaPhongBan) THEN
37     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Mã phòng ban không
38     → tồn tại.';
39   END IF;
40   IF p_LuongTheoGio <= 0 THEN
41     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Lương theo giờ phải
42     → lớn hơn 0.';
43   END IF;
44
45   -- Thêm nhân viên
46   INSERT INTO NhanVien (MaNV, Ho, TenLot, Ten, GioiTinh, Email,
47     → LuongTheoGio, MaPhongBan)
48   VALUES (p_MaNV, p_Ho, p_TenLot, p_Ten, p_GioiTinh, p_Email,
49     → p_LuongTheoGio, p_MaPhongBan);
50 END //
```

Một vài câu lệnh thực thi trường hợp báo lỗi

The screenshot shows a MySQL Workbench interface with a query editor containing the SQL command: `CALL ThemNhanVien("NVabcd", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", 25000, "PB0101");`. Below the query, an error message is displayed: `Mã nhân viên phải có định dạng NVxxxx, với 4 chữ số đằng sau`.

Hình 16: Sử dụng thủ tục ThemNhanVien nhưng sai định dạng MaNV

The screenshot shows a MySQL Workbench interface with a query editor containing the SQL command: `CALL ThemNhanVien("NV0393", "Trần", "Li-en", "Khôi", "Nam", "trananhkhoitv@gmail.com", 25000, "PB0101");`. Below the query, an error message is displayed: `Tên lót không được chứa số hoặc ký tự đặc biệt.`

Hình 17: Sử dụng thủ tục ThemNhanVien nhưng sai định dạng TenLoi

The screenshot shows a MySQL Workbench interface with a query editor containing the SQL command: `CALL ThemNhanVien("NV0393", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", 25000, "PB0101");`. Below the query, an error message is displayed: `Email không đúng định dạng.`

Hình 18: Sử dụng thủ tục ThemNhanVien nhưng sai định dạng Email

The screenshot shows a MySQL Workbench interface with a query editor containing the SQL command: `CALL ThemNhanVien("NV0393", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", 25000, "PB1101");`. Below the query, an error message is displayed: `Mã phòng ban không tồn tại.`

Hình 19: Sử dụng thủ tục ThemNhanVien nhưng mã phòng ban cung cấp không tồn tại

Câu lệnh thực thi thủ tục trường hợp đúng:

The screenshot shows a MySQL Workbench interface with a query editor containing the SQL command: `CALL ThemNhanVien("NV0001", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", 25000, "PB0101");`. Below the query, the output shows the command again: `CALL ThemNhanVien("NV0001", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", 25000, "PB0101")` and the result: `AffectedRows : 1`.

Hình 20: Sử dụng thủ tục ThemNhanVien với các tham số đúng định dạng

3.1.2 Thủ tục UPDATE

Mô tả thủ tục: Thủ tục có nhiệm vụ thực hiện lệnh cập nhật một thực thể trong bảng NhanVien. Thủ tục hoạt động qua hai quá trình:

- Đầu tiên là kiểm tra định dạng của các tham số được truyền vào, nếu sai định dạng thì sẽ thông báo lỗi và ngừng việc thực hiện thủ tục.
- Cuối cùng, sau khi đảm bảo về định dạng, thủ tục sẽ thêm vào bảng NhanVien một thực thể NhanVien mới, với các giá trị là các tham số được cung cấp.

Input:

- Tham số p_MaNV, có kiểu CHAR(6)
- Tham số p_Ho, có kiểu NVARCHAR(10)
- Tham số p_TenLot, có kiểu NVARCHAR(10)
- Tham số p_Ten, có kiểu NVARCHAR(10)
- Tham số p_GioiTinh, có kiểu ENUM('Nam', 'Nữ', 'Khác')
- Tham số p_Email, có kiểu VARCHAR(100)
- Tham số p_HeSoPhatDiTre, có kiểu DECIMAL(5, 2)
- Tham số p_HeSoPhatVangKhongPhep, có kiểu DECIMAL(5, 2)
- Tham số p_SoNgayNghi, có kiểu INT
- Tham số p_LuongTheoGio, có kiểu DECIMAL(10, 2)
- Tham số p_MaPhongBan, có kiểu CHAR(6)

Output: Không có.

Câu lệnh tạo thủ tục:

```

1 CREATE PROCEDURE SuaNhanVien(
2     IN p_MaNV CHAR(6),
3     IN p_Ho NVARCHAR(10),
4     IN p_TenLot NVARCHAR(10),
5     IN p_Ten NVARCHAR(10),
6     IN p_GioiTinh ENUM('Nam', 'Nữ', 'Khác'),
7     IN p_Email VARCHAR(100),
8     IN p_HeSoPhatDiTre DECIMAL(5, 2),
9     IN p_HeSoPhatVangKhongPhep DECIMAL(5, 2),
10    IN p_SoNgayNghi INT,
11    IN p_LuongTheoGio DECIMAL(10, 2),
12    IN p_MaPhongBan CHAR(6)
13 )

```

```

14 BEGIN
15   -- Kiểm tra dữ liệu hợp lệ
16   IF p_ManV NOT REGEXP '^NV[0-9]{4}$' THEN
17     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Mã nhân viên phải có
18     → định dạng NVxxxx, với 4 chữ số đằng sau';
19 END IF;
20   -- Kiểm tra sự tồn tại của nhân viên trong bảng NhanVien
21   IF NOT EXISTS (SELECT 1 FROM NhanVien WHERE MaNV = p_ManV) THEN
22     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Nhân viên với mã này
23     → không tồn tại.';
24 END IF;
25   IF p_Ho REGEXP '[^a-zA-ZÀ-܂]' THEN
26     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Họ không được chứa số
27     → hoặc ký tự đặc biệt.';
28 END IF;
29   IF p_TenLot IS NOT NULL AND p_TenLot REGEXP '[^a-zA-ZÀ-܂]' THEN
30     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tên lót không được chứa
31     → số hoặc ký tự đặc biệt.';
32 END IF;
33   IF p_Ten REGEXP '[^a-zA-ZÀ-܂]' THEN
34     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Tên không được chứa số
35     → hoặc ký tự đặc biệt.';
36 END IF;
37   IF p_Email NOT REGEXP '^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$'
38     → THEN
39     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Email không đúng định
40     → dạng.';
41 END IF;
42   -- Kiểm tra sự tồn tại của MaPhongBan trong bảng PhongBan
43   IF NOT EXISTS (SELECT 1 FROM PhongBan WHERE MaPhongBan = p_MaPhongBan)
44     → THEN
45     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Mã phòng ban không tồn
46     → tại.';
47 END IF;
48   IF p_LuongTheoGio <= 0 THEN
49     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Lương theo giờ phải lớn
50     → hơn 0.';
51 END IF;
52   IF p_HeSoPhatDiTre < 0 THEN
53     SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Hệ số phạt đi trễ phải lớn
54     → hơn hoặc bằng 0.';
55 END IF;

```

```
46  IF p_HeSoPhatVangKhongPhep < 0 THEN
47    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Hệ số phạt vắng không
48      → phép phải lớn hơn hoặc bằng 0.';
49  END IF;
50  IF p_SoNgayNghi < 0 THEN
51    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Số ngày nghỉ phải lớn
52      → hơn hoặc bằng 0.';
53  END IF;
54  -- Sửa thông tin nhân viên
55  UPDATE NhanVien
56  SET Ho = p_Ho,
57    TenLot = p_TenLot,
58    Ten = p_Ten,
59    GioiTinh = p_GioiTinh,
60    Email = p_Email,
61    HeSoPhatDiTre = p_HeSoPhatDiTre,
62    HeSoPhatVangKhongPhep = p_HeSoPhatVangKhongPhep,
63    SoNgayNghi = p_SoNgayNghi,
64    LuongTheoGio = p_LuongTheoGio,
65    MaPhongBan = p_MaPhongBan
66 WHERE MaNV = p_MaNV;
67 END //
```

Một vài câu lệnh thực thi trường hợp báo lỗi

```
CALL SuaNhanVien("NV0000", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", NULL, NULL, NULL, 25000, "PB0101");
```

Q Search Results | 1 + Export | Cost: 5ms < >

Nhân viên với mã này không tồn tại.

Hình 21: Sử dụng thủ tục SuaNhanVien trong trường hợp MaNV không tồn tại

```
CALL SuaNhanVien("NV123b", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", NULL, NULL, NULL, 25000, "PB0101");
```

Q Search Results | 1 + Export | Cost: 3ms < >

Mã nhân viên phải có định dạng NVxxxx, với 4 chữ số đằng sau

Hình 22: Sử dụng thủ tục SuaNhanVien trong trường hợp MaNV sai định dạng

```
CALL SuaNhanVien("NV0001", "Trần?", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", NULL, NULL, NULL, 25000, "PB0101");
```

Q Search Results | 1 + Export | Cost: 5ms < >

Họ không được chứa số hoặc ký tự đặc biệt.

Hình 23: Sử dụng thủ tục SuaNhanVien trong trường hợp Họ sai định dạng

```
CALL SuaNhanVien("NV0001", "Trần", "Anh", "Khôi", "Nam", "trananhkhoitv@gmail.com", -1, NULL, NULL, 25000, "PB0101");
```

Q Search Results | 1 + Export | Cost: 4ms < >

He số phạt đi trễ phải lớn hơn hoặc bằng 0.

Hình 24: Sử dụng thủ tục SuaNhanVien trong trường hợp HeSoPhatDiTre bé hơn 0

Câu lệnh thực thi thủ tục trường hợp đúng:

SELECT * FROM nhanvien WHERE `MaNV`='NV0001';												
	Q	* MaNV char(6)	* HoTen varchar(20)	TenLoi varchar(20)	* TenLop varchar(20)	* GioiTinh enum('1' for nam, '0' for nhanh)	* Email varchar(100)	HeSoPhu decimal(4,2)	HeSoPhu decimal(4,2)	SoLuong int	* LuongTheoThang decimal(10,2)	MaPhongBan char(6)
		Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
	>	NV0001	Trần	Anh	Khôi	Nam	trananhkhoi@gmail.com	1.00	1.00	12	25000.00	PB0101

Hình 25: Trước khi sử dụng thủ tục SuaNhanVien

CALL SuaNhanVien("NV0001", "Trần", "Anh", "Khôi", "Nam", "trananhkhoi@gmail.com", NULL, NULL, NULL, 25000, "PB0101");
CALL SuaNhanVien("NV0001", "Trần", "Anh", "Khôi", "Nam", "trananhkhoi@gmail.com", NULL, NULL, NULL, 25000, "PB0101")
AffectedRows : 1

Hình 26: Sử dụng thủ tục SuaNhanVien thành công

SELECT * FROM nhanvien WHERE `MaNV`='NV0001';												
	Q	Search Results	✉	①	⊕	+	⊖	⟳	↑	↓	Export	Cost: 5ms < 1 > Total 1
		Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
	>	NV0001	Trần	Anh	Khôi	Nam	trananhkhoi@gmail.com	1.00	1.00	12	25000.00	PB0101

Hình 27: Kết quả sau khi thành công sử dụng thủ tục SuaNhanVien, email đã được thay đổi

3.1.3 Thủ tục DELETE

Mô tả thủ tục: Thủ tục có nhiệm vụ thực hiện lệnh xóa một thực thể trong hệ thống. Thủ tục hoạt động như sau: qua MaNV được cung cấp, thủ tục tiến hành tìm kiếm và xóa các thực thể có liên quan đến nhân viên đó ở mọi bảng trong hệ cơ sở dữ liệu.

Input:

- Tham số p_MaNV, có kiểu CHAR(6)

Output: Không có.

```

1   -- Thủ tục xóa nhân viên
2   DROP PROCEDURE IF EXISTS XoaNhanVien //
3   CREATE PROCEDURE XoaNhanVien(
4       IN p_MaNV CHAR(6)
5   )
6   BEGIN
7       -- Xóa dữ liệu liên quan từ các bảng con trước
8       DELETE FROM LanRaVao WHERE MaNV = p_MaNV;
9       DELETE FROM BangChamCong WHERE MaNV = p_MaNV;
10      DELETE FROM LichLamViec WHERE MaNV = p_MaNV;
11      DELETE FROM NhanVienThamGiaDuAn WHERE MaNhanVien = p_MaNV;
12      DELETE FROM NguoiPhuThuoc WHERE MaNV = p_MaNV;
13      DELETE FROM Sdt_NhanVien WHERE MaNV = p_MaNV;
14      DELETE FROM BangLuong WHERE MaNV = p_MaNV;

15      -- Xóa nhân viên từ bảng cụ thể
16      DELETE FROM NhanVienBanThoiGian WHERE MaNV = p_MaNV;
17      DELETE FROM NhanVienToanThoiGian WHERE MaNV = p_MaNV;

18      -- Cuối cùng, xóa nhân viên từ bảng chính
19      DELETE FROM NhanVien WHERE MaNV = p_MaNV;
20
21  END //

```

Câu lệnh thực thi thủ tục thành công

The screenshot shows the MySQL Workbench interface. In the SQL tab, the following command is entered:

```
CALL XoaNhanVien("NV0001");
```

Below the command, the results show:

```
CALL XoaNhanVien("NV0001")  
AffectedRows : 1
```

The status bar at the bottom indicates a cost of 27ms.

Hình 28: Thực thi thủ tục XoaNhanVien thành công

Kết quả thực thi thủ tục XoaNhanVien, khi truy vấn bảng NhanVien với MaNV là "NV0001" thì không cho kết quả.

The screenshot shows the MySQL Workbench interface. A SELECT query is run:

```
SELECT * FROM nhanvien WHERE `MaNV`='NV0001';
```

The results table shows the following columns and their types:

	* MaNV	* Ho	TenLot	* Ten	* GioiTinh	* Email	HeSoPhatDiTre	HeSoPhatVangKhong
	char(6)	varchar(10)	varchar(10)	varchar(10)	enum('Nam','Nữ','Khác')	varchar(100)	decimal(5,2)	decimal(5,2)
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter

The table is empty, indicating no results were found for the specified MaNV.

Hình 29: Kết quả thực thi thủ tục XoaNhanVien

3.2 Trigger

3.2.1 Nhóm Trigger 1

Mô tả Triggers: Nhóm Triggers 1 dùng để tính toán, cập nhật số lượng nhân viên có trong một phòng ban.

Câu lệnh tạo Triggers

```

1   -- Trigger sau khi thêm nhân viên
2   CREATE TRIGGER update_employee_count_after_insert
3       AFTER INSERT ON NhanVien
4           FOR EACH ROW
5               BEGIN
6                   UPDATE PhongBan
7                       SET SoLuongNhanVien = SoLuongNhanVien + 1
8                       WHERE MaPhongBan = NEW.MaPhongBan;
9               END //
10      -- Trigger sau khi cập nhật nhân viên
11      CREATE TRIGGER update_employee_count_after_update
12          AFTER UPDATE ON NhanVien
13              FOR EACH ROW
14                  BEGIN
15                      -- Nếu phòng ban của nhân viên thay đổi, cập nhật số lượng nhân viên
16                      -- ở phòng ban cũ và mới
17                      IF OLD.MaPhongBan ≠ NEW.MaPhongBan THEN
18                          -- Giảm số lượng nhân viên ở phòng ban cũ
19                          UPDATE PhongBan
20                              SET SoLuongNhanVien = SoLuongNhanVien - 1
21                              WHERE MaPhongBan = OLD.MaPhongBan;
22
23                      -- Tăng số lượng nhân viên ở phòng ban mới
24                      UPDATE PhongBan
25                          SET SoLuongNhanVien = SoLuongNhanVien + 1
26                          WHERE MaPhongBan = NEW.MaPhongBan;
27                  END IF;
28              END //
29      -- Trigger sau khi xóa nhân viên
30      CREATE TRIGGER update_employee_count_after_delete
31          AFTER DELETE ON NhanVien
32              FOR EACH ROW
33                  BEGIN
34                      UPDATE PhongBan
35                          SET SoLuongNhanVien = SoLuongNhanVien - 1
36                          WHERE MaPhongBan = OLD.MaPhongBan;
37      END //

```

Kiểm tra Trigger

- Kiểm tra thao tác thêm nhân viên: Thêm 3 nhân viên mới vào phòng ban có mã phòng ban là 'PB0101'

```
SELECT * from phongban WHERE `MaPhongBan`='PB0101';
```

	Search Results										Export	Cost: 4ms	1	Total 1
	Q	* MaPhongBan char(6)	*	TenPhongBar varchar(100)	*	MaChiNh char(4)		SoLuongNhanVien int		MSNV_VanHa char(6)				
	Q	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter				
	>	PB0101	Phòng Kế toán	CN01		5				NV8120				

Hình 30: Thông tin của phòng ban 'PB0101' ban đầu, số nhân viên hiện tại là 5

```
INSERT INTO NhanVien (MaNV, Ho, TenLot, Ten, GioiTinh, Email, LuongTheoGio, MaPhongBan)
VALUES
('NV0001', 'Trần', 'Văn', 'Hoài', 'Nam', 'example@gmail.com', 25000, 'PB0101'),
('NV0002', 'Nguyễn', 'Thị', 'Ba', 'Nữ', 'example2@gmail.com', 35000, 'PB0101'),
('NV0003', 'Lê', 'Phan', 'Thu', 'Nữ', 'example3@gmail.com', 40000, 'PB0101');
```

```
INSERT INTO NhanVien (MaNV, Ho, TenLot, Ten, GioiTinh, Email, LuongTheoGio, MaPhongBan) VALUES ('NV0001', 'Trần', 'Văn', 'Hoài', 'Nam', 'example@gmail.com', 25000, 'PB0101'), ('NV0002', 'Nguyễn', 'Thị', 'Ba', 'Nữ', 'example2@gmail.com', 35000, 'PB0101'), ('NV0003', 'Lê', 'Phan', 'Thu', 'Nữ', 'example3@gmail.com', 40000, 'PB0101')
```

AffectedRows : 3

Hình 31: Thực hiện thêm 3 nhân viên vào phòng ban 'PB0101'

```
SELECT * from phongban WHERE `MaPhongBan`='PB0101';
```

	Search Results										Export	Cost: 2ms	1	Total 1
	Q	* MaPhongBan char(6)	*	TenPhongBar varchar(100)	*	MaChiNh char(4)		SoLuongNhanVien int		MSNV_VanHa char(6)				
	Q	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter				
	>	PB0101	Phòng Kế toán	CN01		8				NV8120				

Hình 32: Thông tin của phòng ban 'PB0101', số nhân viên đã tăng từ 5 lên 8

- Kiểm tra sau thao tác cập nhật thông tin về phòng ban của nhân viên: Chuyển 2 nhân viên đang trong phòng ban 'PB0101' sang phòng ban 'PB0201'

```
SELECT * FROM phongban WHERE `MaPhongBan`='PB0201';
```

	Search Results										Export	Cost: 3ms	1	Total 1
	Q	* MaPhongBan char(6)	*	TenPhongBar varchar(100)	*	MaChiNh char(4)		SoLuongNhanVien int		MSNV_VanHa char(6)				
	Q	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter				
	>	PB0201	Phòng Kế toán	CN02		4				NV9225				

Hình 33: Thông tin của phòng ban 'PB0201' ban đầu, số nhân viên hiện tại là 4

```
UPDATE nhanvien
SET `MaPhongBan` = 'PB0201'
WHERE `MaNV` = 'NV0003' OR `MaNV` = 'NV0002';
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL command: `UPDATE nhanvien SET `MaPhongBan` = 'PB0201' WHERE `MaNV` = 'NV0003' OR `MaNV` = 'NV0002';`
- Results Tab:** Shows the output: `AffectedRows : 2`.
- Toolbar:** Includes icons for search, export, and cost information (Cost: 8ms).

Hình 34: Thực hiện việc chuyển hai nhân viên từ phòng ban 'PB0101' sang phòng ban 'PB0201'

SELECT * from phongban WHERE `MaPhongBan` = 'PB0101';					
	* MaPhongBan	* TenPhongBar	* MaChiNh	SoLuongNhanVien	MSNV_VanHa
	char(6)	varchar(100)	char(4)	int	char(6)
	Filter	Filter	Filter	Filter	Filter
	PB0101	Phòng Kế toán	CN01	6	NV8120

Hình 35: Thông tin của phòng ban 'PB0101', số nhân viên đã giảm từ 8 xuống 6

SELECT * FROM phongban WHERE `MaPhongBan` = 'PB0201';					
	* MaPhongBan	* TenPhongBar	* MaChiNh	SoLuongNhanVien	MSNV_VanHa
	char(6)	varchar(100)	char(4)	int	char(6)
	Filter	Filter	Filter	Filter	Filter
	PB0201	Phòng Kế toán	CN02	6	NV9225

Hình 36: Thông tin của phòng ban 'PB0201', số nhân viên đã tăng từ 4 lên 6

- Kiểm tra sau thao tác xóa nhân viên: Xóa 1 nhân viên từ phòng ban 'PB0201'

```
DELETE FROM nhanvien
WHERE `MaNV` = 'NV0003';
```

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Contains the SQL command: `DELETE FROM nhanvien WHERE `MaNV` = 'NV0003';`
- Results Tab:** Shows the output: `AffectedRows : 1`.
- Toolbar:** Includes icons for search, export, and cost information (Cost: 5ms).

Hình 37: Thực hiện việc xóa một nhân viên thuộc phòng ban 'PB0201'

SELECT * FROM phongban WHERE `MaPhongBan` = 'PB0201';					
	* MaPhongBan	* TenPhongBar	* MaChiNh	SoLuongNhanVien	MSNV_VanHa
	char(6)	varchar(100)	char(4)	int	char(6)
	Filter	Filter	Filter	Filter	Filter
	PB0201	Phòng Kế toán	CN02	5	NV9225

Hình 38: Thông tin của phòng ban 'PB0201', số nhân viên đã giảm từ 6 xuống 5

3.2.2 Nhóm Trigger 2

Mô tả Triggers: Nhóm Triggers 2 dùng để tính toán tổng số giờ làm việc trong ngày của một nhân viên khi họ thực hiện việc chấm công.

Câu lệnh tạo Triggers

```

1   CREATE TRIGGER update_total_hours_after_insert
2     AFTER INSERT ON LanRaVao
3       FOR EACH ROW
4         BEGIN
5           DECLARE total_hours DECIMAL(5, 2);
6           SET total_hours = TIMESTAMPDIFF(MINUTE, NEW.GioVao, NEW.GioRa) / 60;
7
8           UPDATE BangChamCong
9             SET TongSoGioLam = COALESCE(TongSoGioLam, 0) + total_hours
10            WHERE MaNV = NEW.MaNV AND Ngay = NEW.Ngay;
11        END // 
12
13
14      CREATE TRIGGER update_total_hours_after_update
15        AFTER UPDATE ON LanRaVao
16          FOR EACH ROW
17            BEGIN
18              DECLARE total_hours DECIMAL(5, 2);
19              DECLARE previous_hours DECIMAL(5, 2);
20              SET previous_hours = TIMESTAMPDIFF(MINUTE, OLD.GioVao, OLD.GioRa) /
21                → 60;
22              SET total_hours = TIMESTAMPDIFF(MINUTE, NEW.GioVao, NEW.GioRa) / 60;
23              UPDATE BangChamCong
24                SET TongSoGioLam = COALESCE(TongSoGioLam, 0) + total_hours -
25                  → previous_hours
26                WHERE MaNV = NEW.MaNV AND Ngay = NEW.Ngay;
27            END;
28
29      CREATE TRIGGER update_total_hours_after_delete
30        AFTER DELETE ON LanRaVao
31          FOR EACH ROW
32            BEGIN
33              DECLARE total_hours DECIMAL(5, 2);
34              SET total_hours = TIMESTAMPDIFF(MINUTE, OLD.GioVao, OLD.GioRa) / 60;
35
36              UPDATE BangChamCong
37                SET TongSoGioLam = COALESCE(TongSoGioLam, 0) - total_hours
38                WHERE MaNV = OLD.MaNV AND Ngay = OLD.Ngay;
39            END //

```

Kiểm tra Trigger: Quan sát giá trị của thuộc tính TongSoGioLam trong bảng BangChamCong của nhân viên 'NV0001' trong ngày 13/12/2024 khi thực hiện các thao tác thêm, sửa và xóa trong bảng LanRaVao.

- Kiểm tra thao tác thêm lần ra vào

```
SELECT `MaNV`, `Ho`, `TenLot`, `Ten`, `Ngay`, `TongSoGioLam` from nhanvien NATURAL INNER JOIN bangchamcong WHERE `MaNV`='NV0001';
```

MaNV	Ho	TenLot	Ten	Ngay	TongSoGioLam
NV0001	Trần	Văn	Hoài	2024-12-13	0.00

Hình 39: Thông tin ban đầu của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 là 0

```
INSERT INTO lanravao(`MaNV`, `Ngay`, `GioVao`, `GioRa`)
VALUES
('NV0001', '2024-12-13', '08:23:00', '12:00:00'),
('NV0001', '2024-12-13', '13:00:00', '16:45:00');

AffectedRows : 2
```

Hình 40: Thực hiện thêm các lần ra vào công ty của nhân viên 'NV0001'

MaNV	Ho	TenLot	Ten	Ngay	TongSoGioLam
NV0001	Trần	Văn	Hoài	2024-12-13	7.37

Hình 41: Thông tin của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 đã được cập nhật thành 7.37

- Kiểm tra thao tác cập nhật lần ra vào

```
UPDATE lanravao
SET `GioRa`='16:30:00'
WHERE `MaNV`='NV0001' AND `Ngay`='2024-12-13' AND `GioVao`='13:00:00';

AffectedRows : 1
```

Hình 42: Thực hiện giảm giờ ra của một thực thể trong bảng LanRaVao đi 15 phút

SELECT `MaNV`, `Ho`, `TenLot`, `Ten`, `Ngay`, `TongSoGioLam` from nhanvien NATURAL INNER JOIN bangchamcong WHERE `MaNV` = 'NV0001';						
	MaNV	Ho	TenLot	Ten	Ngay	TongSoGioLam
>	NV0001	Trần	Văn	Hoài	2024-12-13	7.12

Hình 43: Thông tin của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 đã giảm từ 7.37 xuống 7.12 (giảm 0.25 giờ = 15 phút)

- Kiểm tra thao tác xóa lần ra vào

DELETE FROM lanravao WHERE `MaNV`='NV0001' AND `Ngay`='2024-12-13' AND `GioVao`='13:00:00';
DELETE FROM lanravao WHERE `MaNV`='NV0001' AND `Ngay`='2024-12-13' AND `GioVao`='13:00:00'
AffectedRows : 1

Hình 44: Thực hiện xóa một thực thể trong bảng LanRaVao, tổng thời gian bị xóa là 3 giờ 30 phút

SELECT `MaNV`, `Ho`, `TenLot`, `Ten`, `Ngay`, `TongSoGioLam` from nhanvien NATURAL INNER JOIN bangchamcong WHERE `MaNV` = 'NV0001';						
	MaNV	Ho	TenLot	Ten	Ngay	TongSoGioLam
>	NV0001	Trần	Văn	Hoài	2024-12-13	3.62

Hình 45: Thông tin của nhân viên 'NV0001', với tổng số giờ làm trong ngày 13/12/2024 đã giảm từ 7.12 xuống 3.62 (giảm 3.5 giờ = 3 giờ 30 phút)

3.3 Thủ tục

3.3.1 Thủ tục 1

Mô tả thủ tục: Thủ tục có tác dụng sắp xếp các phòng ban theo tiêu chí phòng ban có số ngày nhân viên có mặt.

Input: Không có

Output: Bảng thông tin các phòng ban được sắp xếp theo tiêu chí nói trên.

Câu lệnh tạo thủ tục

```

1 CREATE PROCEDURE LocPhongBanCoSoLuongNhanVienCoMatNhieuNhat()
2 BEGIN
3     SELECT
4         PB.MaPhongBan, PB.TenPhongBan, PB.SoLuongNhanVien,
5             COUNT(NV.MaNV) AS SoLuongNhanVienCoMat
6     FROM
7         PhongBan PB
8     JOIN
9         NhanVien NV ON PB.MaPhongBan = NV.MaPhongBan
10    JOIN
11        BangChamCong BCC ON NV.MaNV = BCC.MaNV
12    WHERE
13        BCC.TrangThai = 'Có mặt'
14    GROUP BY
15        PB.MaPhongBan
16    ORDER BY
17        SoLuongNhanVienCoMat DESC;
18 END //
```

Kết quả:

CALL `LocPhongBanCoSoLuongNhanVienCoMatNhieuNhat`;				
	MaPhongBan	TenPhongBan	SoLuongNhanVien	SoLuongNhanVienCoMat
	Filter	Filter	Filter	Filter
>	PB0202	Phòng Nhân sự	13	478
>	PB0401	Phòng Kế toán	11	421
>	PB0106	Phòng Chăm sóc khách hàng	11	391
>	PB0103	Phòng Kỹ thuật	9	376
>	PB0305	Phòng Marketing	10	357
>	PB0303	Phòng Kỹ thuật	10	316
>	PB0405	Phòng Marketing	9	301
>	PB0404	Phòng IT	8	287
>	PB0304	Phòng IT	8	276
>	PB0102	Phòng Nhân sự	7	262

Hình 46: Kết quả sau khi thực thi thủ tục LocPhongBanCoSoLuongNhanVienCoMatNhieuNhat

3.3.2 Thủ tục 2

Mô tả thủ tục: Thủ tục dùng để lọc ra các phòng ban có số lượng nhân viên lớn hơn một con số được người dùng nhập vào.

Input:

- tham số minEmployeeCount, có kiểu INT

Output: Bảng thông tin của các phòng ban có số lượng nhân viên lớn hơn tham số được truyền vào.

Câu lệnh tạo thủ tục

```

1   --Lọc phòng ban có số lượng nhân viên lớn hơn số lượng cho trước
2   CREATE PROCEDURE LocPhongBanCoSoLuongNhanVienLonHon(IN minEmployeeCount
3     ↵      INT)
4   BEGIN
5     SELECT
6       PB.TenPhongBan, COUNT(NV.MaNV) AS EmployeeCount
7     FROM
8       PhongBan PB
9     LEFT JOIN
10      NhanVien NV ON PB.MaPhongBan = NV.MaPhongBan
11    WHERE
12      PB.SoLuongNhanVien > minEmployeeCount
13    GROUP BY
14      PB.TenPhongBan
15    HAVING
16      EmployeeCount > minEmployeeCount
17    ORDER BY
18      EmployeeCount DESC;
END //

```

Kết quả:

CALL `LocPhongBanCoSoLuongNhanVienLonHon` (10)		
	MaPhongBan	TenPhongBan
>	PB0202	Phòng Nhân sự
>	PB0106	Phòng Chăm sóc khách hàng
>	PB0401	Phòng Kế toán
		13
		11
		11

Hình 47: Kết quả sau khi thực thi thủ tục LocPhongBanCoSoLuongNhanVienLonHon với số lượng nhân viên tối thiểu là 10

3.4 Hàm

3.4.1 Hàm 1

Mô tả hàm 1: Hàm tinhluong có chức năng tính lương trước thuế cho một nhân viên nhất định, với thời gian tính lương được xác định bởi người dùng

Input:

- Tham số MaNV, có kiểu CHAR(6)
- Tham số ngayBatDau, có kiểu DATE
- Tham số ngayKetThuc, có kiểu DATE

Output: Giá trị luong, có kiểu INT, mang ý nghĩa là lương trước thuế của nhân viên trong khoảng thời gian cho trước.

Câu lệnh khởi tạo hàm

```
1 CREATE FUNCTION tinhLuong (MaNV CHAR(6), ngayBatDau DATE, ngayKetThuc
2   → DATE) RETURNS INT DETERMINISTIC
3 BEGIN
4   DECLARE luong INT DEFAULT 0;
5   DECLARE current_workingDate DATE;
6   DECLARE current_luong INT;
7   DECLARE done INT DEFAULT 0;
8
9   -- Declare a cursor to iterate through workdays and daily salaries
10  DECLARE WorkDay_CURSOR CURSOR FOR
11    SELECT `Ngay`, `TongSoGioLam`*`LuongTheoGio`
12    FROM bangchamcong NATURAL INNER JOIN nhanvien
13    WHERE nhanvien.MaNV = MaNV;
14
15   -- Declare a handler for the end of the cursor
16   DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
```

```

16    -- Input validation: Check if MaNV exists
17    IF MaNV IS NULL OR NOT EXISTS (
18        SELECT 1
19        FROM nhanvien
20        WHERE nhanvien.`MaNV` = MaNV
21    ) THEN
22        RETURN -1; -- Invalid employee ID
23    END IF;

24
25    -- Validate the date range
26    IF ngayBatDau IS NULL OR ngayKetThuc IS NULL OR ngayBatDau >
27        ↳ ngayKetThuc THEN
28        RETURN -2; -- Invalid date range
29    END IF;

30    -- Open the cursor
31    OPEN WorkDay_CURSOR;

32
33    -- Loop through the cursor
34    read_loop: LOOP
35        FETCH WorkDay_CURSOR INTO current_workingDate, current_luong;
36        IF done = 1 THEN
37            LEAVE read_loop;
38        END IF;

39
40        -- Check if the working date is within the specified range
41        IF (current_luong IS NOT NULL) AND (current_workingDate BETWEEN
42            ↳ ngayBatDau AND ngayKetThuc) THEN
43            SET luong = luong + current_luong;
44        END IF;
45    END LOOP;

46    -- Close the cursor
47    CLOSE WorkDay_CURSOR;

48
49    RETURN luong;
50    END //

```

Kiểm tra: Tính lương của nhân viên có mã nhân viên là "NV7733" từ ngày 01/11/2024 đến 07/11/2024:

- Thông tin về nhân viên và giờ làm việc của nhân viên trên từ ngày 01/11/2024 đến 07/11/2024:

<pre>SELECT `MaNV`, `Ho`, `TenLot`, `Ten`, `Ngay`, `TongSoGioLam`, `LuongTheoGio` FROM nhanvien NATURAL INNER JOIN bangchamcong WHERE `MaNV` = 'NV7733' AND `Ngay` BETWEEN '2024-11-01' AND '2024-11-07';</pre>							
	MaNV	Ho	TenLot	Ten	Ngay	TongSoGioLam	LuongTheoGio
>	NV7733	Đoàn	Hữu	Trung	2024-11-01	4.00	25000.00
>	NV7733	Đoàn	Hữu	Trung	2024-11-04	4.00	25000.00
>	NV7733	Đoàn	Hữu	Trung	2024-11-05	3.25	25000.00
>	NV7733	Đoàn	Hữu	Trung	2024-11-06	0.00	25000.00
>	NV7733	Đoàn	Hữu	Trung	2024-11-07	4.00	25000.00

Hình 48: Thông tin về nhân viên và giờ làm việc của nhân viên

- Có thể thấy, lương của nhân viên 'NV7733' trong quãng thời gian trên là:

$$(4.00 + 4.00 + 3.25 + 0.00 + 4.00) \times 25000 = 381250$$

- Giá trị mà hàm trả về đúng bằng giá trị vừa tính được:

<pre>SELECT `tinhLuong` ('NV7733', '2024-11-01', '2024-11-07');</pre>							
	`tinhLuong`('NV7733')	int	Filter	Filter	Filter	Filter	Filter
>	381250						

Hình 49: Kết quả sau khi gọi hàm tinhluong

3.4.2 Hàm 2

Mô tả hàm 2: Hàm KiemTraTangCa có chức năng kiểm tra số giờ tăng ca của một nhân viên có vượt quá số giờ được truyền vào không trong một quãng thời gian được người gọi hàm quy định.

Input:

- Tham số MaNV, có kiểu CHAR(6)
- Tham số NgayBatDau, có kiểu DATE
- Tham số NgayKetThuc, có kiểu DATE
- Tham số GioTangCaToiThieu, có kiểu TIME

Output: Giá trị BOOLEAN, trả về 0 nếu thời gian tăng ca là nhỏ hơn, ngược lại trả về 1.

Câu lệnh khởi tạo hàm

```

1   CREATE Function KiemTraTangCa (MaNV CHAR(6), NgayBatDau Date,
2     →  NgayKetThuc Date, GioTangCaToiThieu TIME) RETURNS BOOLEAN
3     →  DETERMINISTIC
4
5 BEGIN
6   DECLARE total_hours TIME DEFAULT 0;
7   DECLARE current_gioTangCa TIME;
8   DECLARE current_ngay DATE;
9   DECLARE done INT DEFAULT 0;
10  -- DECLARE
11  DECLARE nhungLanRaVao CURSOR FOR
12    SELECT TIMEDIFF(lanravao.`GioRa`, lanravao.`GioVao`),
13      →  lanravao.`Ngay`
14    FROM lanravao
15    WHERE `GioVao` > '17:00:00' AND lanravao.`MaNV` = `MaNV`;
16
17  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
18
19  IF MaNV is NULL OR NOT EXISTS (SELECT 1 FROM nhanvientoanthoigian
20    →  WHERE nhanvientoanthoigian.`MaNV` = `MaNV`) THEN
21    RETURN -1; -- MaNV rỗng hoặc MaNV không phải là NV toàn thời
22      →  gian;
23  END IF;

```

```

19   IF NgayBatDau IS NULL OR NgayKetThuc IS NULL OR (NgayKetThuc <
20     ↵ NgayBatDau) THEN
21     RETURN -2; -- NgayBatDau hoặc NgayKetThuc rỗng hoặc phạm vi ngày
22     ↵ không hợp lệ
23   END IF;

24
25   IF GioTangCaToiThieu IS NULL THEN
26     RETURN -3; -- GioTangCaToiThieu rỗng
27   END IF;

28
29   OPEN nhungLanRaVao;
30   read_loop: LOOP
31     FETCH nhungLanRaVao into current_gioTangCa, current_ngay;
32     IF done = 1 THEN
33       LEAVE read_loop;
34     END IF;
35
36   IF current_ngay BETWEEN NgayBatDau AND NgayKetThuc THEN
37     SET total_hours = ADDTIME(total_hours, current_gioTangCa);
38   END IF;
39   END LOOP;

40
41   CLOSE nhungLanRaVao;
42
43   return (total_hours  $\geq$  GioTangCaToiThieu);
44   END //

```

Kiểm tra: Kiểm tra tổng số giờ tăng ca của nhân viên có mã nhân viên là 'NV1271' xem trong thời gian từ ngày 01/11/2024 đến ngày 30/11/2024, thời gian tăng ca của nhân viên đó có vượt qua mốc 7 tiếng và 8 tiếng không.

- Truy vấn tổng thời gian tăng ca bằng câu lệnh SELECT, có thể thấy tổng giờ tăng ca từ ngày 01/11/2024 đến ngày 30/11/2024 của nhân viên 'NV1271' là 7 tiếng 11 phút.

```

SELECT `MaNV`, `Ho`, `TenLot`, `Ten`, SEC_TO_TIME(SUM(TIME_TO_SEC(TIMEDIFF(`GioRa`, `GioVao`)))) as TongGioTangCa
from nhanvien NATURAL INNER JOIN lanravao
WHERE `GioVao` > '17:00:00' AND `MaNV` = 'NV1271'
AND `Ngay` BETWEEN '2024-11-01' AND '2024-11-30'
GROUP BY `MaNV`;
    
```

The screenshot shows the MySQL Workbench interface with the following details:
 - **Query Editor:** Contains the SQL query above.
 - **Results Tab:** Shows the output of the query:

MaNV	Ho	TenLot	Ten	TongGioTangCa
NV1271	Trần	Hồng	Trúc	07:11:00

 - **Toolbar:** Includes icons for search, export, and other database operations.

Hình 50: Kết quả truy vấn

- Gọi hàm để kiểm tra với số giờ là 7 tiếng, kết quả cho ra là 1 (True), đúng với kỳ vọng

```

SELECT `KiemTraTangCa`( 'NV1271', '2024-11-01', '2024-11-30', '07:00:00');
    
```

The screenshot shows the MySQL Workbench interface with the following details:
 - **Query Editor:** Contains the call to the KiemTraTangCa function.
 - **Results Tab:** Shows the output:

Result
1

 - **Toolbar:** Includes icons for search, export, and other database operations.

Hình 51: Kết quả sau khi gọi hàm KiemTraTangCa

- Gọi hàm để kiểm tra với số giờ là 8 tiếng, kết quả cho ra là 0 (False), đúng với kỳ vọng

```

SELECT `KiemTraTangCa`( 'NV1271', '2024-11-01', '2024-11-30', '08:00:00');
    
```

The screenshot shows the MySQL Workbench interface with the following details:
 - **Query Editor:** Contains the call to the KiemTraTangCa function.
 - **Results Tab:** Shows the output:

Result
0

 - **Toolbar:** Includes icons for search, export, and other database operations.

Hình 52: Kết quả sau khi gọi hàm KiemTraTangCa

4 Ứng dụng kết nối tới cơ sở dữ liệu

4.1 Tổng quan công nghệ

Về Front-end:

- Công nghệ sử dụng: Vite và React.
- Giao diện người dùng được xây dựng bằng React, kết hợp với Vite để tối ưu hóa quá trình phát triển và đóng gói. React được sử dụng để tạo các thành phần giao diện động, giúp hiển thị dữ liệu nhân sự và cung cấp các chức năng quản lý như thêm mới, chỉnh sửa, và xóa thông tin nhân viên. Giao diện người dùng giao tiếp với backend thông qua API.

Về Back-end:

- Công nghệ sử dụng: Node.js và Express.js.
- Backend đóng vai trò xử lý logic nghiệp vụ và cung cấp API RESTful để phục vụ yêu cầu từ frontend. Express.js giúp tổ chức các endpoint một cách rõ ràng và dễ mở rộng.

Về Database:

- Công nghệ sử dụng: MySQL
- Dữ liệu nhân sự được lưu trữ trong MySQL, sử dụng các bảng để quản lý thông tin nhân viên, phòng ban, và các mối quan hệ liên quan.

Giao tiếp giữa các phần:

- Frontend và backend giao tiếp qua HTTP/HTTPS thông qua các API RESTful. Dữ liệu được truyền dưới dạng JSON.
- Backend sử dụng thư viện MySQL2 kết nối MySQL để truy xuất và xử lý dữ liệu, sau đó trả về kết quả cho frontend.

4.2 Khai báo kết nối đến MySQL

Kết nối đến MySQL được khởi tạo ở file database.js như sau:

```

1 import mysql from 'mysql2/promise'
2 import { configDotenv } from 'dotenv'
3 configDotenv()
4
5 const connection_info = {
6   host: process.env.MYSQL_HOST,
7   user: process.env.MYSQL_USER,
8   password: process.env.MYSQL_PASSWORD,
9   database: process.env.MYSQL_DATABASE,
10  rowsAsArray: true,
11 }
12
13 export async function ReadQuery(sql, param) {
14   let result = '';
15   let success = true
16   try {
17     const connection = await mysql.createConnection(connection_info)
18     result = await connection.execute(sql, param);
19   } catch (error) {
20     console.log(error.message)
21     success = false;
22   }
23   return result[0];
24 }
```

Để thực hiện một câu lệnh SQL, người dùng cần tạo một API mới trong file api.js và tạo một đường dẫn đến nó trong file index.js. Ví dụ: API để đọc toàn bộ dữ liệu từ bảng NhanVien:

```

1 // ##### api.js #####
2 export const getNhanVien = async () => {
3   const rows = await ReadQuery('SELECT * FROM NhanVien');
4   return rows;
5 };
6 // ##### index.js #####
7
8 // ##### index.js #####
9 app.get('/api/nhanvien', async function (req, res) {
10   const nhanvien = await getNhanVien();
11   res.send({nhanvien})
12 })
13 // ##### index.js #####

```

4.3 Các màn hình demo

4.3.1 Màn hình 1

Bao gồm các màn hình: Xem thông tin nhân viên, thêm nhân viên và sửa nhân viên. Có tính năng phân trang và xóa một nhân viên.

Màn hình hiển thị bảng thông tin tất cả các nhân viên trong công ty

Thông tin nhân viên											
Mã Nhân Viên	Họ	Tên Lót	Tên	Giới Tính	Email	Hộ Số Phát Đì Trẻ	Hộ Vắng Không Phép	Số Ngày Nghỉ	Lương Theo Giờ	Mã Phòng Ban	Tùy chỉnh
NV1108	Bùi	Bích	Như	Nam	buibichnhu@gmail.com	1.00	1.00	12	25000.00	PB0403	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1234	Lục	Quang	Hưng	Nữ	luocquanghung@gmail.com	1.00	1.00	12	60000.00	PB0405	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1271	Trần	Hồng	Trúc	Nữ	tranhongtruc@gmail.com	1.00	1.00	12	90000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1282	Lương	Điệu	Ái	Khác	luongdieua@gmail.com	1.00	1.00	12	70000.00	PB0206	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1371	Nguyễn	Thúy	Anh	Nam	nguyenthuyanh@gmail.com	1.00	1.00	12	80000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1404	La	Khải	Ca	Nam	lakhaiaca@gmail.com	1.00	1.00	12	70000.00	PB0105	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1452	Lục	Thuận	Thánh	Khác	lucthuanthanh@gmail.com	1.00	1.00	12	100000.00	PB0402	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1472	Phạm	Khắc	Minh	Khác	phamkhacminh@gmail.com	1.00	1.00	12	90000.00	PB0202	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1492	Lục	Thanh	Tuấn	Nữ	lucthanhtuan@gmail.com	1.00	1.00	12	100000.00	PB0203	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1540	Úc	Thanh	Quang	Nữ	ucthanhquang@gmail.com	1.00	1.00	12	35000.00	PB0104	<input checked="" type="checkbox"/> <input type="checkbox"/>

Trang trước | Trang 1 / 17 | Trang sau | [Thêm nhân viên](#)

Thông tin cá nhân

Hình 53: Hiển thị thông tin bảng NhanVien, được sắp xếp mặc định theo mã nhân viên theo thứ tự a-z

Màn hình hiển thị trang thêm nhân viên, nơi để người dùng thêm một nhân viên mới vào công ty.

- Màn hình có những input field cho người dùng nhập thông tin nhân viên mới vào.
- Nút 'Quay lại' để quay về trang hiển thị bảng thông tin toàn bộ nhân viên
- Nút 'Thêm nhân viên' dùng để thêm nhân viên, màn hình sẽ thông báo thành công và cập nhật lên cơ sở dữ liệu thông tin về nhân viên mới nếu người dùng nhập đầy đủ thông tin, đúng định dạng và không trùng mã nhân viên, nếu không màn hình sẽ thông báo lỗi và không thực hiện việc thêm nhân viên.
- Nút 'Nhập lại' dùng để xóa toàn bộ những thông tin trong các input field.

Thêm nhân viên mới

Mã nhân viên
NVxxxx

Họ
Nguyễn

Tên lót
Vân

Tên
A

Giới tính

Email
example@domain.com

Lương theo giờ
20000

Mã phòng ban

Quay lại | [Thêm nhân viên](#) | Nhập lại

Thông tin cá nhân

Hình 54: Màn hình hiển thị trang thêm nhân viên

Khi có lỗi xảy ra, như lỗi sai định dạng thông tin của nhân viên, màn hình sẽ thông báo lỗi cho người dùng

Hình 55: Màn hình thêm nhân viên - thất bại do sai định dạng email

Khi đã nhập đúng định dạng và đầy đủ thông tin, khi nhấn nút thêm nhân viên, màn hình sẽ thông báo thành công và cập nhật lên cơ sở dữ liệu

Hình 56: Màn hình thêm nhân viên - thành công

Tra cứu thông tin nhân viên vừa được thêm mới trực tiếp trên cơ sở dữ liệu để kiểm chứng hoạt động

Hình 57: Tra cứu thành công, nhân viên vừa được thêm ở trang web đã được cập nhật bên phía back-end

Để chỉnh sửa thông tin của một nhân viên cụ thể, người dùng phải tìm đến hàng chứa thông tin của nhân viên mình muốn cập nhật và click vào icon cây bút ở trong hàng đó, khi đó màn hình chỉnh

sửa thông tin sẽ được hiện ra.

Màn hình chỉnh sửa thông tin nhân viên

- Màn hình có những input field cho người dùng nhập thông tin vào với những placeholder là thông tin hiện tại của nhân viên đó.
- Người dùng có thể nhập vào bất cứ field nào mà mình muốn cập nhật, không bắt buộc phải nhập hết toàn bộ.
- Nút 'Cập nhật' dùng để tiến hành cập nhật thông tin nhân viên, màn hình sẽ thông báo thành công và cập nhật lên cơ sở dữ liệu nếu người dùng nhập đúng định dạng các field muốn cập nhật, nếu không màn hình sẽ thông báo lỗi và không thực hiện việc cập nhật thông tin nhân viên.
- Nút 'Quay lại' để quay về trang hiển thị bảng thông tin toàn bộ nhân viên

Hình 58: Màn hình cập nhật thông tin nhân viên

Trường hợp cập nhật nhưng lỗi định dạng ở thuộc tính HeSoPhatDiTre.

Hình 59: Màn hình thông báo lỗi lúc cập nhật thông tin nhân viên

BÁO CÁO BÀI TẬP LỚN 2 - MÔN HỆ CƠ SỞ DỮ LIỆU (CO2013) - HK241 - LỚP L11

Cập nhật thông tin của nhân viên có mã nhân viên là 'NV0003' với các thông tin mới.

Hình 60: Màn hình trước khi nhấn vào nút 'Cập nhật'

Kết quả của việc cập nhật thông tin của nhân viên 'NV0003'

Mã Nhân Viên	Họ	Tên Lót	Tên	Giới Tính	Email	He So Phat Di Tre	He So Phat Vang Khong	Số Ngày Nghỉ	Lương Theo Giờ	Mã Phòng Ban	Tùy chỉnh
NV0001	Trần	Vân	Hoài	Nam	example@gmail.com	1.00	1.00	12	25000.00	PB0101	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV0002	Nguyễn	Thị	Bà	Nữ	example2@gmail.com	1.00	1.00	12	35000.00	PB0201	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV0003	Trần	Anh	Khôi	Nam	khoiTran@gmail.com	0.80	1.00	10	40000.00	PB0102	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1108	Bùi	Bích	Như	Nam	buibichnhu@gmail.com	1.00	1.00	12	25000.00	PB0403	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1234	Lục	Quang	Hưng	Nữ	lucquanghung@gmail.com	1.00	1.00	12	60000.00	PB0405	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1252	Châu	Phượng	Chi	Nữ	chauphuongchi@gmail.com	1.00	1.00	12	35000.00	PB0301	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1271	Trần	Hồng	Trúc	Nữ	tranhongtruc@gmail.com	1.00	1.00	12	90000.00	PB0301	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1282	Lương	Điều	Ai	Khác	luongdieua@gmail.com	1.00	1.00	12	70000.00	PB0206	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1371	Nguyễn	Thúy	Anh	Nam	nguyenthuyhanh@gmail.com	1.00	1.00	12	80000.00	PB0301	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
NV1404	La	Khải	Ca	Nam	lakhaiqa@gmail.com	1.00	1.00	12	70000.00	PB0105	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

Hình 61: Màn hình thông báo cập nhật thông tin nhân viên 'NV0003' thành công

Kiểm tra thông tin nhân viên trực tiếp trên cơ sở dữ liệu

MaNV	Họ	Tên lót	Tên	Giới Tính	Email	HeSoPhatDiTre	HeSoPhatVangKhong	SoNgayNghi	LuongTheoGio
NV0003	Trần	Anh	Khôi	Nam	khoiTran@gmail.com	0.80	1.00	10	40000.00

Hình 62: Thông tin của nhân viên 'NV0003' đã được cập nhật thành công, với GioiTinh được đổi thành 'Nam', HeSoPhatDiTre được đổi thành 0.08 và LuongTheoGio được đổi thành 40000

Để xóa một nhân viên xác định, người dùng tìm đến hàng chứa thông tin nhân viên đó và click vào icon thùng rác. Trang web sẽ hiện ra một bảng cho người dùng xác nhận hành động của mình.

BÁO CÁO BÀI TẬP LỚN 2 - MÔN HỆ CƠ SỞ DỮ LIỆU (CO2013) - HK241 - LỚP L11

Thông tin nhân viên											
Mã Nhân Viên	Họ	Tên Lót	Tên	Giới Tính	Email	Hệ Số Phạt ĐI Trễ	Hệ Số Phạt Vắng Không Phép	Số Ngày Nghỉ	Lương Theo Giờ	Mã Phòng Ban	Tùy chỉnh
NV0001	Trần	Văn	Hoài	Nam	example@gmail.com	1.00	1.00	12	25000.00	PB0101	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV0002	Nguyễn	Thị	Bà	Nữ	example2@gmail.com	1.00	1.00	12	35000.00	PB0201	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV0003	Trần	Anh	Khôi	Nam	khoitran@gmail.com	0.80	1.00	10	40000.00	PB0102	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1108	Bùi	Bích	Như	Nam	buibichnhu@gmail.com	1.00	1.00	12	25000.00	PB0403	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1234	Lực	Quang	Hiếu	Nam	luocquanghiu@gmail.com	1.00	1.00	12	60000.00	PB0405	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1252	Châu	Phương	Cẩm	Nữ	chaufangcam@gmail.com	1.00	1.00	12	35000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1271	Trần	Hồng	Tú	Nữ	tunhanh@gmail.com	1.00	1.00	12	90000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1282	Lương	Diệu	Ánh	Nam	nguyenthuyanh@gmail.com	1.00	1.00	12	70000.00	PB0206	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1371	Nguyễn	Thúy	Anh	Nam	nguyenthuyanh@gmail.com	1.00	1.00	12	80000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1404	La	Khải	Ca	Nam	lakhaica@gmail.com	1.00	1.00	12	70000.00	PB0105	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1452	Lực	Thuận	Thành	Khác	lucthuathanh@gmail.com	1.00	1.00	12	100000.00	PB0402	<input checked="" type="checkbox"/> <input type="checkbox"/>

Trang trước Trang 1 / 18 Trang sau

[Thêm nhân viên](#)

Hình 63: Khi click vào icon thùng rác trong hàng chứa mã nhân viên 'NV0003', màn hình hiện lên bảng xác nhận

Sau khi xóa nhân viên thành công, màn hình sẽ hiện thị thông báo

Thông tin nhân viên											
Mã Nhân Viên	Họ	Tên Lót	Tên	Giới Tính	Email	Hệ Số Phạt ĐI Trễ	Hệ Số Phạt Vắng Không Phép	Số Ngày Nghỉ	Lương Theo Giờ	Mã Phòng Ban	Tùy chỉnh
NV0001	Trần	Văn	Hoài	Nam	example@gmail.com	1.00	1.00	12	25000.00	PB0101	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV0002	Nguyễn	Thị	Bà	Nữ	example2@gmail.com	1.00	1.00	12	35000.00	PB0201	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1108	Bùi	Bích	Như	Nam	buibichnhu@gmail.com	1.00	1.00	12	25000.00	PB0403	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1234	Lực	Quang	Hiếu	Nam	luocquanghiu@gmail.com	1.00	1.00	12	60000.00	PB0405	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1252	Châu	Phương	Cẩm	Nữ	chaufangcam@gmail.com	1.00	1.00	12	35000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1271	Trần	Hồng	Tú	Nữ	tunhanh@gmail.com	1.00	1.00	12	90000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1282	Lương	Diệu	Ánh	Nam	nguyenthuyanh@gmail.com	1.00	1.00	12	70000.00	PB0206	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1371	Nguyễn	Thúy	Anh	Nam	nguyenthuyanh@gmail.com	1.00	1.00	12	80000.00	PB0301	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1404	La	Khải	Ca	Nam	lakhaica@gmail.com	1.00	1.00	12	70000.00	PB0105	<input checked="" type="checkbox"/> <input type="checkbox"/>
NV1452	Lực	Thuận	Thành	Khác	lucthuathanh@gmail.com	1.00	1.00	12	100000.00	PB0402	<input checked="" type="checkbox"/> <input type="checkbox"/>

Trang trước Trang 1 / 18 Trang sau

[Thêm nhân viên](#)

Hình 64: Màn hình sau khi xóa nhân viên có mã 'NV0003'

Kiểm tra thông tin nhân viên đã bị xóa

SELECT * FROM nhanvien WHERE `MaNV` = 'NV0003';											
<input checked="" type="checkbox"/> Search Results <input type="checkbox"/> Trang 1 / 1 Total 0											
MaNV	Ho	TenLott	Ten	GiớiTinh	Email	HeSoPhatDiTre	HeSoPhatVangKhong	SoNgayNghi	LuongTheoGio	MaPhongBan	Tùy chỉnh

Hình 65: Kết quả tra cứu rỗng, chứng tỏ nhân viên 'NV0003' đã bị xóa

Đoạn code minh họa việc gọi API và render dữ liệu cho màn hình hiển thị bảng thông tin nhân viên

```

1 // BACKEND
2 // API để truy vấn dữ liệu trong bảng nhân viên
3 export const getNhanVien = async () => {
4     const rows = await ReadQuery('SELECT * FROM NhanVien');
5     return rows;
6 }
7
8 app.get('/api/nhanvien', async function (req, res) {
9     const nhanvien = await getNhanVien();
10    res.send({nhanvien})
11 })

```

```

1 // FRONTEND
2 // lấy dữ liệu từ bảng nhân viên
3 try {
4     const res = await fetch('/api/nhanvien');
5     const data = await res.json();
6     // dữ liệu từ bảng nhân viên được lưu trong object NhanVien dưới dạng
7     // list
8     setNhanVien(data.nhanvien);
9 } catch (error) {
10     console.error(error);
11 }
12 ...
13 // truyền dữ liệu xuống component NhanVienTable
14 <NhanVienTable rowsData={currentRows} setRowsData={setNhanVien}
15     // tại component gốc TableRows, dữ liệu được chuyển thành các một list
16     // các <td> và được styling bằng tailwind css
17     let row = rowData.map((data, index) => <td className='max-w-[400px]
18         text-wrap px-3 py-1 border' key={index}>{data}</td>);
19     // sau đó, chúng được bọc trong element <tr>
20     tableRows.push(<tr key={index}>{row}</tr>)
21     // cuối cùng, chúng sẽ được render trong tag <tbody> ở cuối file
22     // component TableRows.jsx
23     return (
24         <tbody>
25             {tableRows}
26         </tbody>
27     )

```

Đoạn code minh họa việc gọi API và render màn hình thêm nhân viên

```

1 // BACKEND
2 // API thêm nhân viên mới
3 export const insertNhanVien = async (MaNV, Ho, TenLot, Ten, GioiTinh,
4   ↪ Email, LuongTheoGio, MaPhongBan) => {
5   const [result, message] = await WriteQuery(
6     `CALL ThemNhanVien('${MaNV}', '${Ho}', '${TenLot}', '${Ten}', ${GioiTinh},
7       ↪ '${Email}', ${LuongTheoGio},
8       ↪ '${MaPhongBan}')`
9   );
10  console.log("result", result)
11  return [result, message];
12};

13 app.post('/api/nhanvien/insert', async function (req, res) {
14   const { MaNV, Ho, TenLot, Ten, GioiTinh, Email, LuongTheoGio,
15     ↪ MaPhongBan } = req.body;
16   const [result, message] = await insertNhanVien(MaNV, Ho, TenLot,
17     ↪ Ten, GioiTinh, Email, LuongTheoGio, MaPhongBan);
18   if (result === 400) {
19     return res.status(400).json({
20       success: false,
21       message: message
22     });
23   }
24   return res.status(200).json({
25     success: true,
26     message: 'Thêm nhân viên thành công'
27   });
28 });

```

```

1 // FRONTEND
2 // Khởi tạo thực thể newNhanVien, chứa các thông tin nhân viên mới sẽ
   → được thêm vào công ty
3 const [newNhanVien, setNewNhanVien] = useState({
4   MaNV: '',
5   Ho: '',
6   TenLot: '',
7   Ten: '',
8   GioiTinh: '',
9   Email: '',
10  LuongTheoGio: '',
11  MaPhongBan: ''
12 });
13
14 // các input fields và nút 'Thêm nhân viên'
15 // Khi người dùng thay đổi giá trị ở các input fields, giá trị của các
   → trường trong thực thể newNhanVien sẽ được thay đổi dựa trên các hàm
   → trong onChangeHandle
16 <InputField label={'Mã nhân viên'} placeholder={'NVxxxx'} type={'text'}
   → onChangeHandle={handleSetMaNV} />
17 <InputField label={'Họ'} placeholder={'Nguyễn'} type={'text'}
   → onChangeHandle={handleSetHo} />
18 <InputField label={'Tên lót'} placeholder={'Văn'} type={'text'}
   → onChangeHandle={handleSetTenLot} />
19 <InputField label={'Tên'} placeholder={'A'} type={'text'}
   → onChangeHandle={handleSetTen} />
20 <SelectField label={'Giới tính'} options={['Nam', 'Nữ', 'Khác']}
   → onchangeHandler={handleSetGioiTinh} />
21 <InputField label={'Email'} placeholder={'example@domain.com'}
   → type={'email'} onChangeHandle={handleSetEmail} />
22 <InputField label={'Lương theo giờ'} placeholder={'20000'}
   → type={'number'} onChangeHandle={handleSetLuongTheoGio} />
23 <SelectField label={'Mã phòng ban'} options={phongBanOptions}
   → onchangeHandler={handleSetMaPhongBan} />
24
25 <div className='my-3 w-fit border-blue-300 border rounded-md'>
26   <Button label={'Thêm nhân viên'}
     → onClickFunction={handleAddNhanVien} />
27 </div>

```

```

1 // Hàm thực thi lệnh thêm nhân viên
2 const createNhanVien = async (nhanVien) => {
3     const errors = [];
4
5     // Kiểm tra thiếu trường bắt buộc (trừ trường Tên lót)
6     if (!nhanVien.MaNV || !nhanVien.Ho || !nhanVien.Ten ||
7         !nhanVien.GioiTinh || !nhanVien.Email || !nhanVien.LuongTheoGio
8         || !nhanVien.MaPhongBan) {
9         errors.push('Hãy điền vào tất cả các trường.');
10    }
11
12    // Kiểm tra định dạng cụ thể
13    if (nhanVien.MaNV && !nhanVien.MaNV.match(/^NV[0-9]{4}$/)) {
14        errors.push('Mã nhân viên phải có định dạng NVxxxx, với 4 chữ số.');
15    }
16
17    if (nhanVien.Email && !nhanVien.Email.match(/^[a-zA-Z0-9._%+-]+@[a_
18        -zA-Z0-9.-]+\.[a-zA-Z]{2,}\$/)) {
19        errors.push('Email không đúng định dạng.');
20    }
21
22    if (nhanVien.Ho && nhanVien.Ho.match(/[^a-zA-ZÀ-܂ ]/)) {
23        errors.push('Họ không được chứa số hoặc ký tự đặc biệt.');
24    }
25
26    if (nhanVien.Ten && nhanVien.Ten.match(/[^a-zA-ZÀ-܂ ]/)) {
27        errors.push('Tên không được chứa số hoặc ký tự đặc biệt.');
28    }
29
30    if (nhanVien.LuongTheoGio && parseFloat(nhanVien.LuongTheoGio) <= 0)
31        {
32            errors.push('Lương theo giờ phải lớn hơn 0.');
33        }
34
35    // Nếu có lỗi, trả về danh sách lỗi
36    if (errors.length > 0) {
37        return {
38            success: false,
39            message: errors.join('\n') // Kết hợp các lỗi thành một chuỗi
40        };
41    }
42
43    // Nếu không có lỗi, gửi yêu cầu tới API để thêm nhân viên
44    const res = await fetch('/api/nhanvien/insert', {
45        method: 'POST',
46        headers: {'Content-Type': 'application/json'},
47        body: JSON.stringify(nhanVien)
48    });

```

```
45 const data = await res.json();
46 return {
47   success: data.success,
48   message: data.message
49 };
50 }
```

Đoạn code minh họa cho việc gọi API và render màn hình chỉnh sửa nhân viên

```

1 // BACKEND
2 // API để lấy thông tin của một nhân viên theo mã nhân viên
3 export const getNhanVienByMaNV = async (MaNV) => {
4     const rows = await ReadQuery(`SELECT * FROM NhanVien WHERE MaNV =
5         '${MaNV}'`);
6     return rows.length ? rows[0] : null;
7 };
8
9 app.get('/api/nhanvien/:MaNV', async (req, res) => {
10     const { MaNV } = req.params;
11     try {
12         const nhanvien = await getNhanVienByMaNV(MaNV); // Hàm này cần lấy
13             // từ database
14         if (!nhanvien) {
15             return res.status(404).json({ success: false, message: 'Nhân
16                 viên không tồn tại' });
17         }
18         res.status(200).json({ success: true, nhanvien });
19     } catch (error) {
20         console.error(error);
21         res.status(500).json({ success: false, message: 'Lỗi server' });
22     }
23 };
24
25 // API để cập nhật thông tin một nhân viên
26 export const updateNhanVien = async (MaNV, Ho, TenLot, Ten, GioiTinh,
27     Email, HeSoPhatDiTre, HeSoPhatVangKhongPhep, SoNgayNghi,
28     LuongTheoGio, MaPhongBan) => {
29     const [result, message] = await WriteQuery(
30         `CALL SuaNhanVien('${MaNV}', '${Ho}', '${TenLot}', '${Ten}',
31             '${GioiTinh}', '${Email}', ${HeSoPhatDiTre},
32             ${HeSoPhatVangKhongPhep}, ${SoNgayNghi}, ${LuongTheoGio},
33             '${MaPhongBan}')`
34     );
35     console.log("result", result)
36     return [result, message];
37 };

```

```
30 app.put('/api/nhanvien/update/:MaNV', async (req, res) => {
31   const { MaNV, Ho, TenLot, Ten, GioiTinh, Email, HeSoPhatDiTre,
32     → HeSoPhatVangKhongPhep, SoNgayNghi, LuongTheoGio, MaPhongBan } =
33     → req.body;
34   console.log("Dữ liệu nhận được từ frontend:", req.body); // Kiểm tra
35   → dữ liệu từ frontend
36   try {
37     const [result, message] = await updateNhanVien(MaNV, Ho,
38       → TenLot, Ten, GioiTinh, Email, HeSoPhatDiTre,
39       → HeSoPhatVangKhongPhep, SoNgayNghi, LuongTheoGio,
40       → MaPhongBan);
41     console.log("Kết quả cập nhật:", result); // Log kết quả từ
42     → database
43     if (result === 400) {
44       return res.status(400).json({ success: false, message });
45     }
46     res.status(200).json({ success: true, message: 'Cập nhật thành
47       → công' });
48   } catch (error) {
49     console.error(error);
50     res.status(500).json({ success: false, message: 'Lỗi server' });
51   }
52 });
53 );
```

```

1 // Lấy thông tin của nhân viên để hiển thị thông tin hiện tại dưới dạng
2   → placeholder của các input fields
3 const fetchNhanVien = async () => {
4   try {
5     const res = await fetch(`api/nhanvien/${MaNV}`); // API lấy
6       → thông tin nhân viên
7     const data = await res.json();
8     if (data.success) {
9       const nhanVienData = data.nhanvien;
10      setNhanVien({
11        Ho: nhanVienData[1],
12        TenLot: nhanVienData[2],
13        Ten: nhanVienData[3],
14        GioiTinh: nhanVienData[4],
15        Email: nhanVienData[5],
16        HeSoPhatDiTre: nhanVienData[6],
17        HeSoPhatVangKhongPhep: nhanVienData[7],
18        SoNgayNghi: nhanVienData[8],
19        LuongTheoGio: nhanVienData[9],
20        MaPhongBan: nhanVienData[10],
21      });
22    } else {
23      console.error(data.message);
24      alert('Không tìm thấy nhân viên!');
25      navigate('/nhan-vien');
26    }
27  } catch (error) {
28    console.error('Error fetching nhân viên:', error.message);
29    alert('Có lỗi xảy ra khi lấy thông tin nhân viên!');
30  };
31 // Đoạn code hiển thị các input fields
32
33 // Thực thể để lưu thông tin mới của nhân viên
34 // Khi các giá trị trong input fields được thay đổi, các trường của thực
35   → thể nhanVien cũng thay đổi theo dựa trên hàm setNhanVien().
36 const [nhanVien, setNhanVien] = useState({
37   Ho: '',
38   TenLot: '',
39   Ten: '',
40   GioiTinh: '',
41   Email: '',
42   HeSoPhatDiTre: '',
43   HeSoPhatVangKhongPhep: '',
44   SoNgayNghi: '',
45   LuongTheoGio: '',
46   MaPhongBan: '',
47 });

```

```

46 // Hàm thực hiện việc gọi API cập nhật thông tin
47 const handleUpdateNhanVien = async () => {
48     // Các lỗi về nhập liệu sẽ được kiểm tra bằng hàm validateNhanVien
49     // → và được lưu vào thực thể errors
50     const errors = validateNhanVien(nhanVien);
51     if (errors.length > 0) {
52         // Đoạn code thông báo lỗi
53         return;
54     }
55
56     const updatedNhanVien = {
57         MaNV, // Mã nhân viên từ URL (không thay đổi)
58         ...nhanVien,
59         HeSoPhatDiTre: parseFloat(nhanVien.HeSoPhatDiTre),
60         HeSoPhatVangKhongPhep:
61             parseFloat(nhanVien.HeSoPhatVangKhongPhep),
62         SoNgayNghi: parseInt(nhanVien.SoNgayNghi, 10),
63         LuongTheoGio: parseFloat(nhanVien.LuongTheoGio),
64     };
65
66     try {
67         const res = await fetch(`/api/nhanvien/update/${MaNV}`, {
68             method: 'PUT',
69             headers: {
70                 'Content-Type': 'application/json',
71             },
72             body: JSON.stringify(updatedNhanVien),
73         });
74         const data = await res.json();
75
76         if (data.success) {
77             // Đoạn code hiển thị thông báo cập nhật thành công
78             navigate(`/nhan-vien?page=${currentPage}`); // Quay lại
79             // → trang đúng
80         } else {
81             // Đoạn code hiển thị thông báo cập nhật thất bại
82         }
83     } catch (error) {
84         console.error('Error updating nhân viên:', error.message);
85     }
86 };

```

Đoạn code minh họa cho việc gọi API cho thao tác xóa một nhân viên

```

1 // API cho thao tác xóa một nhân viên cụ thể
2 export const deleteNhanVien = async (MaNV) => {
3     const [result, message] = await WriteQuery(`CALL
4         XoaNhanVien('${MaNV}')`);
5     return [result, message];
6 };
7
8 app.delete('/api/nhanvien/delete/:MaNV', async function (req, res) {
9     const { MaNV } = req.params;
10    console.log(`Đang thực hiện yêu cầu xóa nhân viên với MãNV:
11        ${MaNV}`);
12    const [result, message] = await deleteNhanVien(MaNV);
13    if (result === 400) {
14        return res.status(400).json({
15            success: false,
16            message: message
17        });
18    }
19    return res.status(200).json({
20        success: true,
21        message: 'Xóa nhân viên thành công'
22    });
23
24 // Khi click vào icon thùng rác ở một hàng trong bảng nhân viên, Mã nhân
25 // viên của hàng đó sẽ được truyền để làm tham số cho hàm xóa nhân viên
26 // dưới đây
27 const deleteFunction = async (MaNV) => {
28     if (window.confirm('Bạn có chắc chắn muốn xóa nhân viên này
29         không?')) {
30         // Xóa tạm thời hàng khỏi giao diện trước
31         setRowsData((prevRows) => prevRows.filter((row) => row[0] !==
32             MaNV));
33
33     try {
34         const res = await fetch(`/api/nhanvien/delete/${MaNV}`, {
35             method: 'DELETE',
36         });
37         const data = await res.json();
38     }
39 
```

```
34 if (data.success) {
35     alert('Xóa nhân viên thành công');
36 } else {
37     alert('Xóa nhân viên không thành công: ' + data.message);
38     // Khôi phục lại hàng nếu xóa không thành công
39     setRowsData((prevRows) => [...prevRows, rowsData.find((row)
40         => row[0] === MaNV)]);
41 }
42 } catch (error) {
43     console.error('Error deleting employee:', error);
44     // Khôi phục lại hàng nếu có lỗi xảy ra
45     setRowsData((prevRows) => [...prevRows, rowsData.find((row)
46         => row[0] === MaNV)]);
47 }
```

4.3.2 Màn hình 2

Mô tả: Xem thông tin của phòng ban cùng với thông tin của nhân viên vận hành phòng ban đó. Có tính năng lọc các phòng ban theo số lượng nhân viên tối thiểu. Có tính năng sắp xếp các phòng ban theo thứ tự giảm dần tổng số ngày có mặt của các nhân viên thuộc phòng ban đó.

Màn hình hiển thị thông tin phòng ban, cùng với các nút chức năng

The screenshot shows a web-based application for managing departmental information. On the left, there is a sidebar with a user icon and the following menu items:

- Chủ sở hữu công ty
- Thông tin chi nhánh
- Thông tin phòng ban
- Thông tin dự án
- Thông tin nhân viên
- Tính lương

Below the sidebar, there is a section titled "Thông tin cá nhân". The main content area is titled "Thông tin phòng ban". It features a search bar with placeholder "Nhập số lượng nhân viên tối thiểu" and a dropdown menu set to "0". There are two buttons: "Lọc phòng ban" (highlighted in green) and "Phòng ban NV có mặt nhiều nhất".

The main table has the following columns:

Mã Phòng Ban	Tên Phòng Ban	Mã Chi Nhánh	Số Lượng Nhân Viên	MSNV Vận Hành	Tên NV Vận Hành	Tùy chỉnh
PB0101	Phòng Kế toán	CN01	6	NV8120	Lê Diễm Châu	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0102	Phòng Nhân sự	CN01	7	NV6786	Nguyễn Minh Duyên	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0103	Phòng Kỹ thuật	CN01	9	NV7040	Dữu Hồng Đức	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0104	Phòng IT	CN01	4	NV4036	Huỳnh Thanh Phi	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0105	Phòng Marketing	CN01	3	NV1404	La Khải Ca	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0106	Phòng Chăm sóc khách hàng	CN01	11	NV3614	Lê Bá Kỳ	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0201	Phòng Kế toán	CN02	5	NV9225	Dương Phương Loan	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0202	Phòng Nhân sự	CN02	13	NV1472	Phạm Khắc Minh	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0203	Phòng Kỹ thuật	CN02	7	NV5136	Đặng Hiệp Hào	<input checked="" type="checkbox"/> <input type="checkbox"/>
PB0204	Phòng IT	CN02	7	NV3795	Nguyễn Hiền Thực	<input checked="" type="checkbox"/> <input type="checkbox"/>

At the bottom, there are navigation buttons: "Trang trước", "Trang 1 / 3", and "Trang sau".

Hình 66: Hiển thị thông tin bảng PhongBan, cùng với các nút chức năng

Để sử dụng tính năng lọc các phòng ban theo số lượng nhân viên tối thiểu, người dùng tiến hành nhập số lượng tối thiểu và click vào nút 'Lọc phòng ban'. Màn hình sẽ hiển thị thông tin các phòng ban thỏa mãn tiêu chí đó. Người dùng có thể quay lại xem thông tin phòng ban bằng cách nhấn vào nút 'Tắt lọc'.



Chủ sở hữu công ty

- [Thông tin chi nhánh](#)
- [Thông tin phòng ban](#)
- [Thông tin dự án](#)
- [Thông tin nhân viên](#)
- [Tính lương](#)

Thông tin phòng ban

5
Lọc phòng ban
Phòng ban NV có mặt nhiều nhất

Mã Phòng Ban	Tên Phòng Ban	Số Lượng Nhân viên
PB0202	Phòng Nhân sự	13
PB0106	Phòng Chăm sóc khách hàng	11
PB0401	Phòng Kế toán	11
PB0303	Phòng Kỹ thuật	10
PB0305	Phòng Marketing	10
PB0103	Phòng Kỹ thuật	9
PB0405	Phòng Marketing	9
PB0206	Phòng Chăm sóc khách hàng	8
PB0304	Phòng IT	8
PB0404	Phòng IT	8

Trang trước
Trang 1 / 2
Trang sau

Thông tin cá nhân

Hình 67: Màn hình lọc phòng ban có số lượng nhân viên lớn hơn 5

Để sử dụng tính năng sắp xếp các phòng ban theo thứ tự giảm dần tổng số ngày có mặt của các nhân viên, người dùng nhấn vào nút 'Phòng ban NV có mặt nhiều nhất'. Để quay lại bảng thông tin phòng ban bình thường sau khi sử dụng tính năng, người dùng click vào nút 'Tắt hiển thị'.



Chủ sở hữu công ty

- [Thông tin chi nhánh](#)
- [Thông tin phòng ban](#)
- [Thông tin dự án](#)
- [Thông tin nhân viên](#)
- [Tính lương](#)

Thông tin phòng ban

Nhập
Lọc phòng ban
Tắt hiển thị

Mã Phòng Ban	Tên Phòng Ban	Số Lượng Nhân viên	Tổng số ngày có mặt
PB0202	Phòng Nhân sự	13	478
PB0401	Phòng Kế toán	11	421
PB0106	Phòng Chăm sóc khách hàng	11	391
PB0103	Phòng Kỹ thuật	9	376
PB0305	Phòng Marketing	10	357
PB0303	Phòng Kỹ thuật	10	316
PB0405	Phòng Marketing	9	301
PB0404	Phòng IT	8	287
PB0304	Phòng IT	8	276
PB0102	Phòng Nhân sự	7	262

Trang trước
Trang 1 / 3
Trang sau

Thông tin cá nhân

Hình 68: Màn hình sắp xếp phòng ban dựa trên số ngày có mặt của nhân viên

Để hiện thực màn hình trên, chúng em tạo 3 components tương ứng cho 3 bảng với số cột khác nhau, mỗi bảng tương ứng với một trong các chức năng sau:

- Xem thông tin phòng ban và nhân viên vận hành.
- Xem thông tin các phòng ban có số nhân viên lớn hơn một con số cụ thể.
- Hiển thị phòng ban theo thứ tự giảm dần số ngày có mặt của nhân viên.

```

1 // Component để hiển thị bảng thông tin phòng ban và nhân viên vận hành.
2 function PhongBanTable(props) {
3     const headers = ['Mã Phòng Ban', 'Tên Phòng Ban', 'Mã Chi Nhánh',
4         ↪ 'Số Lượng Nhân viên', 'MSNV Vận Hành', 'Tên NV vận hành',
5         ↪ /*'MSNV Quản Lý', 'Tên NV quản lý',*/ "Tùy chỉnh"]
6
7     return (
8         <Table columnHeaders={headers} rowsData={props.rowsData} />
9     )
10 }
11 // Component để hiển thị bảng xem thông tin phòng ban có số nhân viên
12 ↪ lớn hơn một con số cụ thể.
13 function PhongBanTable2({ rowsData }) {
14     const headers = ['Mã Phòng Ban', 'Tên Phòng Ban', 'Số Lượng Nhân
15         ↪ viên'];
16
17     return (
18         <Table columnHeaders={headers} rowsData={rowsData} />
19     );
20 }
21 // Component để hiển thị phòng ban theo thứ tự giảm dần số ngày có mặt
22 ↪ của nhân viên.
23 function PhongBanTable3({ rowsData }) {
24     const headers = ['Mã Phòng Ban', 'Tên Phòng Ban', 'Số Lượng Nhân
25         ↪ viên', 'Tổng số ngày có mặt'];
26
27     return (
28         <Table columnHeaders={headers} rowsData={rowsData} />
29     );
30 }
31
32 // Việc bảng nào sẽ được hiển thị được xác định bởi trạng thái của hai
33 ↪ biến showMaxPresence và filterEnabled
34 {showMaxPresence ? (
35     <PhongBanTable3 rowsData={currentRows} />
36 ) : filterEnabled ? (
37     <PhongBanTable2 rowsData={currentRows} />
38 ) : (
39     <PhongBanTable rowsData={currentRows} />
40 )}
```

Câu lệnh gọi API và render dữ liệu cho bảng thông tin phòng ban và nhân viên vận hành.

```

1 // BACK-END
2 // API để lấy thông tin về phòng ban cùng nhân viên vận hành phòng ban
3 // → đó
4 export const getPhongBanInfo_Manager = async () => {
5     const rows = await ReadQuery(`SELECT nhanvien.\`MaPhongBan\`,
6         \`TenPhongBan\`, \`MaChiNhanh\`, \`SoLuongNhanVien\`,
7         nhanvien.\`MaNV\`, CONCAT (nhanvien.\`Ho\`, ' ',
8         nhanvien.\`TenLot\`, ' ', nhanvien.\`Ten\`) as 'TenVanHanh' from
9         phongban JOIN nhanvien ON
10        phongban.\`MSNV_VanHanh\` = nhanvien.\`MaNV\`;`)
11
12     return rows;
13 }
14
15 app.get('/api/phongban', async function (req, res) {
16     const phongbaninfo = await getPhongBanInfo_Manager();
17     res.send({phongbaninfo})
18 })
19
20
21 // FRONT-END
22 // Gọi API để lấy thông tin
23 const fetchPhongBan = async () => {
24     try {
25         const res = await fetch("/api/phongban");
26         const data = await res.json();
27         console.log(data);
28         setPhongBan(data.phongbaninfo);
29     } catch (error) {
30         console.log(error);
31     }
32 };
33
34 // Các câu lệnh phục vụ việc phân trang
35 // Xác định dữ liệu cho trang hiện tại
36 const totalPages = Math.ceil(phongBan.length / rowsPerPage);
37 // currentRows sẽ chứa thông tin của phòng ban và nhân viên vận hành tại
38 // → trang currentPage
39 const currentRows = phongBan.slice(
40     (currentPage - 1) * rowsPerPage,
41     currentPage * rowsPerPage
42 );
43
44 // Truyền dữ liệu cho component PhongBanTable để render bảng dữ liệu
45 <PhongBanTable rowsData={currentRows} />

```

Câu lệnh gọi API và render dữ liệu cho bảng các phòng ban có số nhân viên lớn hơn một con số cụ thể.

```

1  // BACK-END
2  // API để lấy thông tin về các phòng ban có số nhân viên lớn hơn một con
   → số cụ thể.
3  export const getPhongBanCoSoLuongNhanVienLonHon = async (min) => {
4      const rows = await ReadFromProcedureQuery(`CALL
   ↳ LocPhongBanCoSoLuongNhanVienLonHon(${min})`);
5      return rows;
6  };
7  app.get('/api/phongban-co-nhanvien-lon-hon', async (req, res) => {
8      const { min } = req.query; // Lấy tham số "min" từ query string
9      if (!min) {
10          return res.status(400).json({
11              success: false,
12              message: 'Vui lòng cung cấp tham số "min"',
13          });
14      }
15
16      try {
17          const filteredPhongBan = await
   ↳ getPhongBanCoSoLuongNhanVienLonHon(min);
18          res.status(200).json({
19              success: true,
20              filteredPhongBan,
21          });
22      } catch (error) {
23          console.error(error);
24          res.status(500).json({
25              success: false,
26              message: 'Lỗi server',
27          });
28      }
29  });

```

```
1 // FRONT-END
2 // Input field để người dùng nhập số lượng nhân viên tối thiểu
3 <input
4     type="number"
5     placeholder="Nhập số lượng nhân viên tối thiểu"
6     value={filterValue}
7     onChange={(e) => setFilterValue(e.target.value)}
8   />
9 // Khi người dùng nhấn nút 'Lọc phòng ban', hàm fetchFilteredPhongBan sẽ
10 → được gọi để lấy thông tin
11 const fetchFilteredPhongBan = async () => {
12   try {
13     const res = await fetch(`/api/phongban-co-nhanvien-lon_`_
14       → -hon?min=${filterValue}`);
15     const data = await res.json();
16     if (data.success) {
17       setPhongBan(data.filteredPhongBan);
18     } else {
19       console.error("Không có dữ liệu sau khi lọc.");
20     }
21   } catch (error) {
22     console.error("Lỗi khi lọc dữ liệu:", error);
23   }
24 };
25 // Các câu lệnh phục vụ việc phân trang
26 // Dữ liệu của trang currentPage lúc này được lưu ở thực thể currentRows
27 // Truyền dữ liệu cho component PhongBanTable để render bảng dữ liệu
28 <PhongBanTable2 rowsData={currentRows} />
```

Câu lệnh gọi API và render dữ liệu cho bảng hiển thị phòng ban theo thứ tự giảm dần số ngày có mặt của nhân viên.

```

1  // BACK-END
2  // API để lấy dữ liệu phòng ban theo thứ tự giảm dần số ngày có mặt của
   ↵ nhân viên.
3  export const getPhongBanCoSoLuongNhanVienCoMatNhieuNhat = async () => {
4      try {
5          const rows = await ReadFromProcedureQuery('CALL
   ↵ LocPhongBanCoSoLuongNhanVienCoMatNhieuNhat');
6          return rows;
7      } catch (error) {
8          console.error(error);
9          throw error; // Rerun the error if needed for logging
10     }
11 };
12 app.get('/api/phongban-nhanvien-comatnhieunhat', async (req, res) => {
13     try {
14         const maxPresencePhongBan = await
15             getPhongBanCoSoLuongNhanVienCoMatNhieuNhat(); // Hàm này sẽ
16             ↵ gọi thủ tục bạn đã tạo
17         res.status(200).json({
18             success: true,
19             maxPresencePhongBan,
20         });
21     } catch (error) {
22         console.error(error);
23         res.status(500).json({
24             success: false,
25             message: 'Lỗi khi lấy dữ liệu phòng ban'
26         });
27     }
28 });

```

```

1 // FRONT-END
2 // Khi nhấn nút 'Phòng ban NV có mặt nhiều nhất', hàm sau sẽ được gọi để
3 // → gọi API trên
4 const fetchMaxPresencePhongBan = async () => {
5   try {
6     const res = await
7       → fetch("/api/phongban-nhanvien-comatnhieunhat");
8     const data = await res.json();
9     if (data.success) {
10       setPhongBan(data.maxPresencePhongBan);
11     } else {
12       console.error("Không có dữ liệu phòng ban có mặt nhiều
13         → nhất.");
14     }
15   } catch (error) {
16     console.error("Lỗi khi fetch dữ liệu phòng ban có mặt nhiều
17         → nhất:", error);
18   }
19 };
20

// Các câu lệnh phục vụ việc phân trang
// Dữ liệu của trang currentPage lúc này được lưu ở thực thể currentRows
// Dữ liệu được truyền xuống component hiển thị phòng ban theo thứ tự
// → giảm dần số ngày có mặt của nhân viên để render bảng thông tin.
<PhongBanTable3 rowsData={currentRows}

```

4.3.3 Màn hình 3

Mô tả: Tính lương cho nhân viên trong khoảng thời gian được chọn.

Màn hình ban đầu. Hiện tại đang hiển thị bảng rỗng cho người dùng chưa chọn ngày bắt đầu và ngày kết thúc cho khoảng thời gian tính lương.

Hình 69: Màn hình ban đầu, do chưa xác định thời gian nên hiện bảng rỗng

Sau khi người dùng chọn ngày bắt đầu và ngày kết thúc, màn hình sẽ hiển thị bảng lương của các nhân viên được lọc theo tiêu chí mà người dùng đặt.

MSNV	Tên nhân viên	Tổng số giờ làm việc	Lương tạm tính	Lương thực tế
NV0001	Trần Văn Hoài	0	0	0
NV0002	Nguyễn Thị Ba	0	0	0
NV1108	Bùi Bích Như	23.28	582 000	492 605
NV1234	Lục Quang Hưng	128.22	7 693 200	6 511 524
NV1252	Châu Phương Chi	45.44	1 590 400	1 346 115
NV1271	Trần Hồng Trúc	128.17	11 535 300	9 763 478
NV1282	Lương Diệu Ái	133.39	9 337 300	7 903 091
NV1371	Nguyễn Thúy Anh	109.18	8 734 400	7 392 796
NV1404	La Khải Ca	149.02	10 431 400	8 829 137
NV1452	Lục Thuận Thành	120.99	12 099 000	10 240 594

Hình 70: Bảng lương của toàn bộ nhân viên tính từ ngày 01/11/2024 đến ngày 30/11/2024

Các tiêu chí mà người dùng có thể lọc nhân viên là:

- Lọc ra những nhân viên trong 1 phòng ban nào đó
- Lọc ra những nhân viên theo loại nhân viên (nhân viên toàn thời gian hoặc nhân viên bán thời gian)

Một vài màn hình ví dụ cho bộ lọc:

The screenshot shows a software interface for calculating salaries. At the top, there are dropdown menus for 'Chi nhánh' (selected: CN03 - Chi nhánh Huế), 'Phòng ban' (selected: PB0303 - Phòng Kỹ thuật), and 'Loại nhân viên' (selected: Nhân viên toàn thời gian). Below these are date pickers for 'Ngày bắt đầu tính lương' (01/11/2024) and 'Ngày kết thúc' (30/11/2024). The main area displays a table of employee salaries:

MSNV	Tên nhân viên	Tổng số giờ làm việc	Lương tạm tính	Lương thực tế
NV2994	Thi Thúy Hiền	169.17	10 150 200	8 591 129
NV6713	Chử Chấn Hùng	120.01	7 200 600	6 094 588
NV7597	Nguyễn Thái Yên	117.25	8 207 500	6 946 828
NV8703	Nguyễn Quang Vũ	142.83	12 854 700	10 880 218
NV9964	Vũ Hoài Phương	131.52	11 836 800	10 018 668

At the bottom, there are navigation buttons: 'Trang trước', 'Trang 1 / 1', and 'Trang sau'.

Hình 71: Bảng lương của các nhân viên toàn thời gian thuộc phòng ban 'PB0303'

The screenshot shows a software interface for calculating salaries. At the top, there are dropdown menus for 'Chi nhánh' (selected: CN01 - Chi nhánh Cần Thơ), 'Phòng ban' (selected: PB0101 - Phòng Kế toán), and 'Loại nhân viên' (selected: Nhân viên toàn thời gian). Below these are date pickers for 'Ngày bắt đầu tính lương' (01/11/2024) and 'Ngày kết thúc' (30/11/2024). The main area displays a table of employee salaries:

MSNV	Tên nhân viên	Tổng số giờ làm việc	Lương tạm tính	Lương thực tế
NV8120	Lê Diễm Châu	153.75	12 300 000	10 410 720
NV9606	Nguyễn Hướng Ly	146.05	7 302 500	6 180 836

At the bottom, there are navigation buttons: 'Trang trước', 'Trang 1 / 1', and 'Trang sau'.

Hình 72: Bảng lương của các nhân viên toàn thời gian thuộc phòng ban '0101'

Câu lệnh tạo API phục vụ cho màn hình tính lương

```

1  export const getBangLuong = async (MaPhongBan, EmpType, BeginDate,
2    ↪ EndDate) => {
3      if ((BeginDate === "") || (EndDate === "")) return [];
4      const rows = await ReadFromProcedureQuery(`CALL
5        ↪ View_BangLuong('${MaPhongBan}', ${EmpType}, '${BeginDate}',
6        ↪ '${EndDate}')`);
7      return rows;
8    }
9
10
11 app.get('/api/bang-luong', async (req, res) => {
12   let {MaPhongBan, EmpType, BeginDate, EndDate} = req.query;
13   if (!MaPhongBan) MaPhongBan = "all";
14   try {
15     const BangLuong = await getBangLuong(MaPhongBan, EmpType,
16       ↪ BeginDate, EndDate);
17     // console.log(BangLuong)
18     res.status(200).json({
19       success: true,
20       BangLuong
21     })
22   } catch (error) {
23     console.error(error);
24     res.status(500).json({ success: false, message: 'Lỗi server' });
25   }
26 })

```

Component phục vụ việc render dữ liệu:

```

1  function BangLuong({Data}) {
2    const Headers = ["MSNV", "Tên nhân viên", "Tổng số giờ làm việc",
3      ↪ "Lương tạm tính", "Lương thực tế"]
4    return (
5      <Table columnHeaders={Headers} rowsData={Data} />
6    )
7  }
8
9  export default BangLuong

```

Câu lệnh gọi API và render dữ liệu

```

1  // Khởi tạo các thực thể để lưu giá trị tham số cho câu lệnh gọi API
2  const [beginDate, setBeginDate] = useState("")
3  const [endDate, setEndDate] = useState("")
4  const [maChiNhanh, setMaChiNhanh] = useState("");

```

```

5   const [maPhongBan, setMaPhongBan] = useState("");
6   const [empType, setEmpType] = useState(0);
7   // Khi người dùng thực hiện việc thay đổi giá trị của các input fields
8   // sau, giá trị của các thực thể trên sẽ thay đổi
9   <SelectField label={"Chi nhánh"} placeholder={"Tất cả"}
10  > options={chiNhanhOptions} onChangeHandler={handleMaChiNhanh}>
11 <SelectField label={"Phòng ban"} placeholder={"Tất cả"}
12  > options={phongBanOptions} onChangeHandler={handleMaPhongBan} />
13 <SelectField label={"Loại nhân viên"} placeholder={"Tất cả"}
14  > options={[{"Nhân viên toàn thời gian", "Nhân viên bán thời gian"}]}
15  > onChangeHandler={handleEmpType}>
16 <InputField label={"Ngày bắt đầu tính lương"} type={'date'}
17  > onChangeHandle={handleBeginDate}>
18 <InputField label={"Ngày kết thúc"} type={'date'}
19  > onChangeHandle={handleEndDate}>

20 // Khi đã có đầy đủ hai giá trị beginDate và endDate, câu lệnh gọi API
21 // sẽ được thực hiện để lấy dữ liệu và lưu vào thực thể bangLuongData
22 const [bangLuongData, setBangLuongData] = useState([]);
23 const fetchBangLuong = async (maPhongBan, empType, beginDate, endDate)
24  => {
25   try {
26     const res = await
27       fetch(`api/bang-luong?MaPhongBan=${maPhongBan}&EmpType=$
28         ${empType}&BeginDate=${beginDate}&EndDate=${endDate}`);
29     const data = await res.json();
30     console.log(data.BangLuong)
31     data.BangLuong.forEach(item => {
32       item[3] = numberWithSpaces(item[3])
33       item[4] = numberWithSpaces(Math.round(item[4]))
34     });
35     setBangLuongData(data.BangLuong);
36   } catch (error) {
37     console.log(error);
38   }
39 }

40 // Dữ liệu đang được lưu ở thực thể bangLuongData sẽ được xử lý để phân
41 // trang, các dữ liệu của trang hiện tại sẽ được lưu ở thực thể curData
42 const [curData, setCurData] = useState([]);
43 useEffect(() => {
44   setCurData(bangLuongData.slice(curPage * 10 - 10, curPage * 10));
45 }, [bangLuongData, curPage])

46 // Cuối cùng, truyền dữ liệu của trang hiện tại xuống component
47 // BangLuong để render bảng dữ liệu
48 <BangLuong Data={curData}>

```

5 Phần thêm

5.1 Người thực hiện: Trần Anh Khôi - 2211694

5.1.1 Function tonggiolamviec

Mô tả: Hàm dùng để tính tổng số giờ làm việc của một nhân viên cụ thể, trong khoảng thời gian xác định

Input:

- Tham số MaNV, có kiểu CHAR(6)
- Tham số ngayBatDau, có kiểu DATE
- Tham số ngayKetThuc, có kiểu DATE

Output: Một giá trị có kiểu DECIMAL(5, 2), là tổng số giờ làm việc của nhân viên có mã nhân viên là tham số 'MaNV' trong khoảng thời gian từ ngayBatDau đến ngayKetThuc.

Câu lệnh tạo hàm:

```

1 Create Function tonggiolamviec (MaNV CHAR(6), ngayBatDau DATE,
2   ↪  ngayKetThuc DATE) RETURNS DECIMAL(5, 2) DETERMINISTIC
3 BEGIN
4   DECLARE total_time DECIMAL(5, 2);
5   SET total_time = (SELECT SUM(bangchamcong.`TongSoGioLam`)
6     from bangchamcong
7     WHERE bangchamcong.`MaNV` = MaNV AND
8       bangchamcong.`TongSoGioLam` IS NOT NULL AND
9       bangchamcong.`Ngay` BETWEEN ngayBatDau AND ngayKetThuc
10      GROUP BY bangchamcong.`MaNV`);
11      return total_time;
12  END //
```

Kiểm tra Tính tổng số giờ làm việc của nhân viên có mã nhân viên là 'NV1271' từ ngày 01/11/2024 đến ngày 07/11/2024

- Câu lệnh tra cứu thông thường:

Search Results				
MaNV	Ho	TenLot	Ten	SUM(`TongSoGioLam`)
NV1271	Trần	Hồng	Trúc	30.75

Hình 73: Kết quả câu lệnh tra cứu thông thường

- Gọi hàm tonggiolamviec:

BÁO CÁO BÀI TẬP LỚN 2 - MÔN HỆ CƠ SỞ DỮ LIỆU (CO2013) - HK241 - LỚP L11

The screenshot shows a MySQL command-line interface window. The query entered is:

```
SELECT tonggiolamviec( 'NV1271', "2024-11-01", "2024-11-07");
```

The results pane displays the output of the stored procedure:

	Filter
>	30.75

At the top of the interface, there are various icons for search, export, and navigation. The status bar indicates a cost of 4ms and a total of 1 row.

Hình 74: Kết quả câu lệnh gọi hàm tonggiolamviec

5.1.2 Procedure View_Bangluong

Mô tả: Thủ tục dùng để xem bảng lương trong khoảng thời gian xác định của các nhân viên trong một phòng ban cụ thể, hoặc toàn bộ nhân viên, có thể lọc theo loại nhân viên. Thủ tục này dùng để phục vụ việc hiển thị dữ liệu ở màn hình 3.

input:

- Tham số MaPhongBan, có kiểu CHAR(6)
- Tham số EmpType, có kiểu INT
- Tham số BeginDate, có kiểu DATE
- Tham số EndDate, có kiểu DATE

Output: Bảng chứa thông tin của các nhân viên, số giờ làm việc trong quãng thời gian trên, lương tạm tính và lương thực tế của họ sau khi đã trừ đi thuế và bảo hiểm xã hội.

Câu lệnh khởi tạo thủ tục:

```

1  CREATE PROCEDURE View_BangLuong (MaPhongBan CHAR(6), EmpType INT,
2   → BeginDate DATE, EndDate DATE)
3  BEGIN
4    DECLARE ThueSuat DECIMAL(5, 2);
5    DECLARE BaoHiemXH DECIMAL(5, 2);
6
7    SET ThueSuat = (SELECT bangthietlapluong.`ThueSuat` FROM
8     → bangthietlapluong WHERE `NgayApDung` ≤ EndDate ORDER BY
9     → `NgayApDung` DESC LIMIT 1);
10   SET BaoHiemXH = (SELECT bangthietlapluong.`BaoHiemXH` FROM
11    → bangthietlapluong WHERE `NgayApDung` ≤ EndDate ORDER BY
12    → `NgayApDung` DESC LIMIT 1);
13
14  IF EmpType = 0 THEN
15    IF MaPhongBan = 'all' THEN
16      SELECT nhanvien.`MaNV`, CONCAT(nhanvien.`Ho`, ' ', ,
17       → nhanvien.`TenLot`, ' ', nhanvien.`Ten`) AS TenNhanVien,
18       → tonggiolamviec(nhanvien.`MaNV`, BeginDate, EndDate) AS
19       → TongGioLamViec, tinhluong(nhanvien.`MaNV`, BeginDate,
20       → EndDate) AS LuongTamTinh, ((tinhluong(nhanvien.`MaNV`,
21       → BeginDate, EndDate) - tinhluong(nhanvien.`MaNV`,
22       → BeginDate, EndDate)*(BaoHiemXH)*(1 - ThueSuat)) AS
23       → LuongThucTe FROM nhanvien;
24  ELSE
25    SELECT nhanvien.`MaNV`, CONCAT(nhanvien.`Ho`, ' ', ,
26     → nhanvien.`TenLot`, ' ', nhanvien.`Ten`) AS TenNhanVien,
27     → tonggiolamviec(nhanvien.`MaNV`, BeginDate, EndDate) AS
28     → TongGioLamViec, tinhluong(nhanvien.`MaNV`, BeginDate,
29     → EndDate) AS LuongTamTinh, ((tinhluong(nhanvien.`MaNV`,
30     → BeginDate, EndDate) - tinhluong(nhanvien.`MaNV`,
31     → BeginDate, EndDate)*(BaoHiemXH)*(1 - ThueSuat)) AS
32     → LuongThucTe FROM nhanvien WHERE
33     → nhanvien.`MaPhongBan` = MaPhongBan;
34 END IF;

```

```

15 ELSEIF EmpType = 1 THEN
16   IF MaPhongBan = 'all' THEN
17     SELECT nhanvien.`MaNV`, CONCAT(nhanvien.`Ho`, ' ', 
18       nhanvien.`TenLot`, ' ', nhanvien.`Ten`) AS TenNhanVien,
19       tonggiolamviec(nhanvien.`MaNV`, BeginDate, EndDate) AS
20       TongGioLamViec, tinhluong(nhanvien.`MaNV`, BeginDate,
21       EndDate) AS LuongTamTinh, ((tinhluong(nhanvien.`MaNV`,
22       BeginDate, EndDate) - tinhluong(nhanvien.`MaNV`,
23       BeginDate, EndDate)*(BaoHiemXH)*(1 - ThueSuat)) AS
24       LuongThucTe FROM nhanvien NATURAL INNER JOIN
25       nhanvientoanthoigian;
26   ELSE
27     SELECT nhanvien.`MaNV`, CONCAT(nhanvien.`Ho`, ' ', 
28       nhanvien.`TenLot`, ' ', nhanvien.`Ten`) AS TenNhanVien,
29       tonggiolamviec(nhanvien.`MaNV`, BeginDate, EndDate) AS
30       TongGioLamViec, tinhluong(nhanvien.`MaNV`, BeginDate,
31       EndDate) AS LuongTamTinh, ((tinhluong(nhanvien.`MaNV`,
32       BeginDate, EndDate) - tinhluong(nhanvien.`MaNV`,
33       BeginDate, EndDate)*(BaoHiemXH)*(1 - ThueSuat)) AS
34       LuongThucTe FROM nhanvien NATURAL INNER JOIN
35       nhanvientoanthoigian WHERE
36       nhanvien.`MaPhongBan` = MaPhongBan;
37   END IF;
38 ELSEIF EmpType = -1 THEN
39   IF MaPhongBan = 'all' THEN
40     SELECT nhanvien.`MaNV`, CONCAT(nhanvien.`Ho`, ' ', 
41       nhanvien.`TenLot`, ' ', nhanvien.`Ten`) AS TenNhanVien,
42       tonggiolamviec(nhanvien.`MaNV`, BeginDate, EndDate) AS
43       TongGioLamViec, tinhluong(nhanvien.`MaNV`, BeginDate,
44       EndDate) AS LuongTamTinh, ((tinhluong(nhanvien.`MaNV`,
45       BeginDate, EndDate) - tinhluong(nhanvien.`MaNV`,
46       BeginDate, EndDate)*(BaoHiemXH)*(1 - ThueSuat)) AS
47       LuongThucTe FROM nhanvien NATURAL INNER JOIN
48       nhanvienbanthoigian;
49   ELSE
50     SELECT nhanvien.`MaNV`, CONCAT(nhanvien.`Ho`, ' ', 
51       nhanvien.`TenLot`, ' ', nhanvien.`Ten`) AS TenNhanVien,
52       tonggiolamviec(nhanvien.`MaNV`, BeginDate, EndDate) AS
53       TongGioLamViec, tinhluong(nhanvien.`MaNV`, BeginDate,
54       EndDate) AS LuongTamTinh, ((tinhluong(nhanvien.`MaNV`,
55       BeginDate, EndDate) - tinhluong(nhanvien.`MaNV`,
56       BeginDate, EndDate)*(BaoHiemXH)*(1 - ThueSuat)) AS
57       LuongThucTe FROM nhanvien NATURAL INNER JOIN
58       nhanvienbanthoigian WHERE
59       nhanvien.`MaPhongBan` = MaPhongBan;
60   END IF;
61 END IF;
62 END IF;
63 END //

```

Thực thi thủ tục: Xem bảng lương của các nhân viên thuộc phòng ban có mã phòng ban là 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024, lần lượt lọc theo các loại nhân viên:

- Tất cả các nhân viên thuộc phòng ban 'PB0101'

call `View_BangLuong` ('PB0101', 0, '2024-10-15', '2024-10-31');					
	MaNV string	TenNhanVien varchar	TongGioLamViec decimal	LuongTamTinh int	LuongThucTe decimal
>	NV7733	Đoàn Hữu Trung	11.03	275750	228321.0000
>	NV8120	Lê Diễm Châu	63.25	5060000	4189680.0000
>	NV8232	Mã Hưng Đạo	11.45	286250	237015.0000
>	NV9199	Phùng Tuấn Thành	18.77	469250	388539.0000
>	NV9606	Nguyễn Hương Ly	72.31	3615500	2993634.0000

Hình 75: Bảng lương của tất cả các nhân viên thuộc phòng ban 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024

- Các nhân viên toàn thời gian thuộc phòng ban 'PB0101'

call `View_BangLuong` ('PB0101', 1, '2024-10-15', '2024-10-31');					
	MaNV string	TenNhanVien varchar	TongGioLamViec decimal	LuongTamTinh int	LuongThucTe decimal
>	NV8120	Lê Diễm Châu	63.25	5060000	4189680.0000
>	NV9606	Nguyễn Hương Ly	72.31	3615500	2993634.0000

Hình 76: Bảng lương của các nhân viên toàn thời gian thuộc phòng ban 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024

- Các nhân viên bán thời gian thuộc phòng ban 'PB0101'

call `View_BangLuong` ('PB0101', -1, '2024-10-15', '2024-10-31');					
	MaNV string	TenNhanVien varchar	TongGioLamViec decimal	LuongTamTinh int	LuongThucTe decimal
>	NV7733	Đoàn Hữu Trung	11.03	275750	228321.0000
>	NV8232	Mã Hưng Đạo	11.45	286250	237015.0000
>	NV9199	Phùng Tuấn Thành	18.77	469250	388539.0000

Hình 77: Bảng lương của các nhân viên bán thời gian thuộc phòng ban 'PB0101' từ ngày 15/10/2024 đến ngày 31/10/2024

6 Source Code

Source code được nhóm lưu ở Github: [Github](#)