create database jdbc\_practice;

use jdbc\_practice;

create table accounts

(

id int auto\_increment primary key,

username varchar(255),

password varchar(255),

role enum ('admin','hr') not null,

create\_at timestamp default current\_timestamp(),

status boolean

);

drop table accounts;

create table departments

(

dp\_id int auto\_increment primary key,

dp\_name varchar(255) not null unique check ( char\_length(dp\_name) > 0 && char\_length(dp\_name) < 100 ),

dp\_description varchar(255),

dp\_status boolean

);

create table employee

(

employee\_id varchar(5) primary key,

full\_name varchar(150) not null check (char\_length(full\_name) between 15 and 150),

email varchar(255) not null unique,

phone\_number varchar(15) not null,

gender enum ('male', 'female', 'other') not null,

salary\_level int not null check (salary\_level > 0),

salary decimal(15, 2) not null check (salary > 0),

birth\_date date not null,

address text not null,

status enum ('active', 'inactive', 'onleave', 'policyleave') not null,

department\_id int not null,

foreign key (department\_id) references departments (dp\_id),

check (employee\_id regexp '^E[0-9]{4}$'),

check (email regexp '^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'),

check (phone\_number regexp '^0[0-9]{9}$')

);

delimiter \\

create procedure isLogin(

username\_in varchar(255),

password\_in varchar(255)

)

begin

select \* from accounts where username = username\_in and password = password\_in;

end \\

delimiter \\;

delimiter \\

create procedure get\_all\_department()

begin

select \*from departments;

end \\

delimiter \\;

INSERT INTO departments (dp\_name, dp\_description, dp\_status)

VALUES ('Phòng Nhân sự', 'Quản lý nhân sự và tuyển dụng', true),

('Phòng Kế toán', 'Theo dõi tài chính và kế toán', true),

('Phòng Kỹ thuật', 'Phát triển phần mềm và hạ tầng', true),

('Phòng Marketing', 'Chiến lược quảng bá và truyền thông', true),

('Phòng Bán hàng', 'Quản lý bán hàng và chăm sóc khách hàng', true),

('Phòng IT', 'Hỗ trợ kỹ thuật và quản trị hệ thống', true),

('Phòng Pháp chế', 'Tư vấn và xử lý pháp lý', true),

('Phòng R&D', 'Nghiên cứu và phát triển sản phẩm mới', true),

('Phòng Thiết kế', 'Thiết kế giao diện và nhận diện thương hiệu', true),

('Phòng Điều hành', 'Điều phối hoạt động doanh nghiệp', true),

('Phòng Hành chính', 'Quản lý văn phòng và tài sản', true),

('Phòng Đào tạo', 'Đào tạo và phát triển nhân sự', true),

('Phòng Chăm sóc Khách hàng', 'Hỗ trợ khách hàng sau bán', true),

('Phòng Sản xuất', 'Quản lý sản xuất và vận hành', true),

('Phòng Logistics', 'Quản lý kho vận và vận chuyển', true),

('Phòng Kiểm thử', 'Kiểm tra chất lượng phần mềm', true),

('Phòng Dự án', 'Quản lý và giám sát tiến độ dự án', false),

('Phòng An ninh', 'Đảm bảo an toàn thông tin và dữ liệu', false),

('Phòng CSR', 'Trách nhiệm xã hội của doanh nghiệp', false),

('Phòng Tài nguyên', 'Quản lý tài nguyên và môi trường', true);

delimiter \\

create procedure get\_department\_by\_page(page int, size int)

begin

select \*from departments limit page offset size;

end \\

delimiter \\

drop procedure get\_department\_by\_page;

delimiter \\

create procedure add\_new\_department(

name varchar(255), description varchar(255), staus boolean

)

begin

insert into departments (dp\_name, dp\_description, dp\_status)

values (name, description, staus);

end \\

delimiter \\;

delimiter \\

create procedure update\_department(id\_in int, name\_in varchar(255), des\_in varchar(255), status\_in boolean)

begin

update departments

set dp\_name = name\_in,

dp\_description = des\_in,

dp\_status = status\_in

where dp\_id = id\_in;

end \\

delimiter \\

delimiter \\

create procedure delete\_department(id int)

begin

delete from departments where dp\_id = id;

end \\

delimiter \\

delimiter \\

create procedure get\_all\_employees()

begin

select \*from employee;

end \\

delimiter \\

delimiter \\

create procedure get\_department\_by\_name(name varchar(255))

begin

select \*from departments where dp\_name = name;

end \\

delimiter \\;

INSERT INTO employee (employee\_id, full\_name, email, phone\_number, gender, salary\_level, salary, birth\_date, address,

status, department\_id)

VALUES ('E0001', 'Nguyễn Văn An ', 'an.nguyen01@example.com', '0123456789', 'male', 2, 12000000.00, '1990-05-12',

'Hà Nội', 'active', 11),

('E0002', 'Trần Thị Bích Hạnh ', 'hanh.tran02@example.com', '0987654321', 'female', 3, 15000000.00,

'1988-07-25', 'TP HCM', 'active', 2),

('E0003', 'Lê Minh Tuấn ', 'tuan.le03@example.com', '0912345678', 'male', 1, 10000000.00, '1995-11-30',

'Đà Nẵng', 'active', 3),

('E0004', 'Phạm Thị Hương ', 'huong.pham04@example.com', '0909123456', 'female', 2, 12000000.00,

'1992-03-18', 'Hải Phòng', 'onleave', 4),

('E0005', 'Đỗ Văn Cường ', 'cuong.do05@example.com', '0934567890', 'male', 4, 18000000.00, '1987-09-05',

'Cần Thơ', 'active', 5),

('E0006', 'Ngô Thị Mai ', 'mai.ngo06@example.com', '0967890123', 'female', 1, 9000000.00, '1993-12-01',

'Huế', 'active', 6),

('E0007', 'Vũ Minh Khôi ', 'khoi.vu07@example.com', '0978901234', 'male', 2, 11500000.00, '1990-06-20',

'Quảng Ninh', 'inactive', 7),

('E0008', 'Bùi Thị Lan ', 'lan.bui08@example.com', '0943210987', 'female', 3, 14000000.00, '1989-01-15',

'Nam Định', 'active', 8),

('E0009', 'Hoàng Văn Đức ', 'duc.hoang09@example.com', '0923456789', 'male', 2, 12500000.00, '1991-08-10',

'Nghệ An', 'policyleave', 9),

('E0010', 'Lý Thị Kim Ngân ', 'ngan.ly10@example.com', '0956789012', 'female', 2, 13000000.00, '1994-04-22',

'Bình Dương', 'active', 10);

SELECT \*

FROM departments;

delimiter \\

create procedure get\_employee\_by\_page(page int, size int)

begin

select \*from employee limit page offset size;

end \\

delimiter \\

delimiter \\

create procedure add\_new\_employee(

in p\_employee\_id varchar(5),

in p\_full\_name varchar(150),

in p\_email varchar(255),

in p\_phone\_number varchar(15),

in p\_gender enum ('male', 'female', 'other'),

in p\_salary\_level int,

in p\_salary decimal(15, 2),

in p\_birth\_date date,

in p\_address text,

in p\_status enum ('active', 'inactive', 'onleave', 'policyleave'),

in p\_department\_id int

)

begin

insert into employee (employee\_id, full\_name, email, phone\_number, gender,

salary\_level, salary, birth\_date, address, status, department\_id)

values (p\_employee\_id, p\_full\_name, p\_email, p\_phone\_number, p\_gender,

p\_salary\_level, p\_salary, p\_birth\_date, p\_address, p\_status, p\_department\_id);

end \\

delimiter ;

delimiter \\

create procedure update\_employee(

in p\_employee\_id varchar(5),

in p\_full\_name varchar(150),

in p\_email varchar(255),

in p\_phone\_number varchar(15),

in p\_gender enum ('male', 'female', 'other'),

in p\_salary\_level int,

in p\_salary decimal(15, 2),

in p\_birth\_date date,

in p\_address text,

in p\_status enum ('active', 'inactive', 'onleave', 'policyleave'),

in p\_department\_id int

)

begin

update employee

set full\_name = p\_full\_name,

email = p\_email,

phone\_number = p\_phone\_number,

gender = p\_gender,

salary\_level = p\_salary\_level,

salary = p\_salary,

birth\_date = p\_birth\_date,

address = p\_address,

status = p\_status,

department\_id = p\_department\_id

where employee\_id = p\_employee\_id;

end \\

delimiter ;

delimiter \\

create procedure delete\_employee(in p\_employee\_id varchar(5))

begin

delete from employee where employee\_id = p\_employee\_id;

end \\

delimiter ;

delimiter \\

create procedure set\_employee\_inactive(emp\_id varchar(10))

begin

update employee

set status = 'inactive'

where employee\_id = emp\_id;

end \\

delimiter \\;

delimiter \\

create procedure search\_employee\_by\_name(emp\_name varchar(100))

begin

select \*

from employee

where full\_name like concat('%', emp\_name, '%');

end \\

delimiter \\

DELIMITER //

CREATE PROCEDURE search\_employee\_by\_age\_range(IN min\_age INT, IN max\_age INT)

BEGIN

SELECT \*,

TIMESTAMPDIFF(YEAR, birth\_date, CURDATE()) AS age

FROM employee

WHERE TIMESTAMPDIFF(YEAR, birth\_date, CURDATE()) BETWEEN min\_age AND max\_age;

END //

DELIMITER ;

drop procedure search\_employee\_by\_age\_range;

delimiter \\

create procedure sort\_employees\_by\_salary\_desc()

begin

select \*

from employee

order by salary desc;

end \\

delimiter \\

delimiter \\

create procedure sort\_employees\_by\_name\_asc()

begin

select \*

from employee

order by full\_name asc;

end \\

delimiter \\

delimiter //

create procedure calculator\_employee\_group\_by\_department()

begin

select d.dp\_id as department\_id,

d.dp\_name as department\_name,

count(e.employee\_id) as employee\_count

from departments d

left join employee e on d.dp\_id = e.department\_id

group by d.dp\_id, d.dp\_name;

end //

delimiter ;

delimiter //

create procedure count\_employee()

begin

select count(employee\_id) as total\_employee from employee;

end //

delimiter ;

delimiter //

create procedure most\_staffed\_office()

begin

select d.dp\_name as Dapartment\_name, count(e.employee\_id) as count\_employee

from departments d

join employee e on d.dp\_id = e.department\_id

group by d.dp\_id

order by count\_employee desc

limit 1;

end //

delimiter ;

delimiter \\

create procedure most\_salary\_department()

begin

select d.dp\_name as DepartmentName,

sum(e.salary) as totalSalary

from departments d

join employee e on d.dp\_id = e.department\_id

group by d.dp\_id

order by totalSalary desc

limit 1;

end \\

delimiter \\;