

Payroll Calculation

Problem statement:

You are a part of the HR department in a company, and you have been asked to calculate the monthly payout for each employee based on their experience and performance.

Objective:

The objective is to design a database to retrieve the detailed salary divisions of each employee in the organization.

1. -- create a database named SQL basics:

```
create database sqlbasics;
```

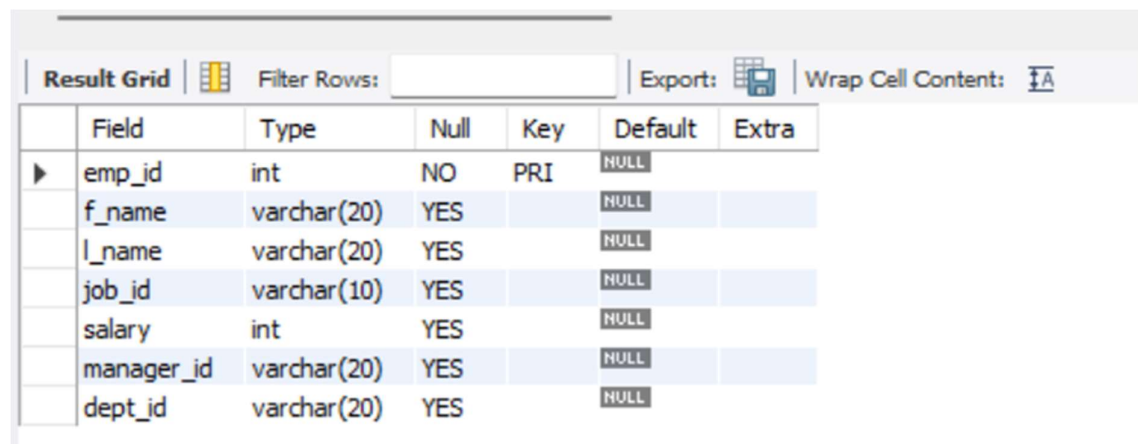
2. -- select SQL basics:

```
use sqlbasics;
```

3. -- Write a query to create the employee and department tables

```
Create table employee_datasets (  
emp_id int not null primary key,  
f_name varchar(20),  
l_name varchar(20),  
job_id varchar(10),  
salary int,  
manager_id varchar(20),  
dept_id varchar(20));
```

```
describe employee_datasets;
```



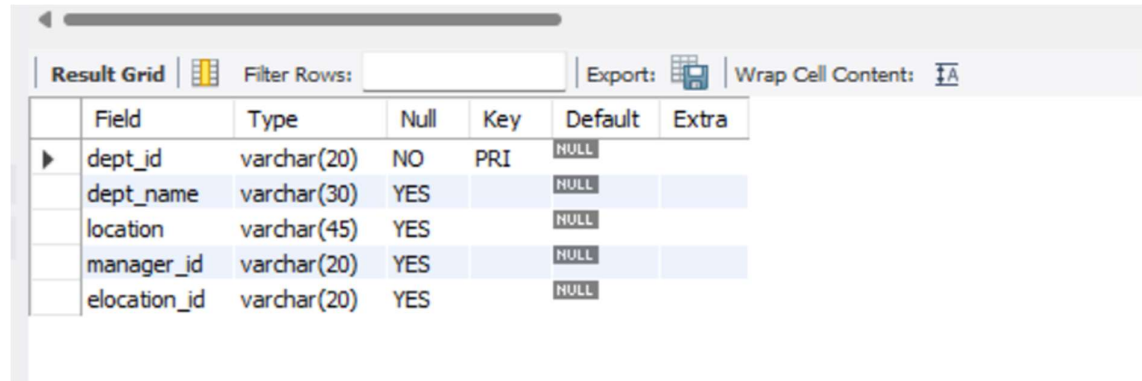
The screenshot shows a database management tool interface. At the top, there is a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below the toolbar is a table representing the structure of the 'employee_datasets' table. The table has columns for Field, Type, Null, Key, Default, and Extra. The rows list the fields: emp_id (int, NOT NULL, PRIMARY KEY), f_name (varchar(20), NULL), l_name (varchar(20), NULL), job_id (varchar(10), NULL), salary (int, NULL), manager_id (varchar(20), NULL), and dept_id (varchar(20), NULL).

| | Field | Type | Null | Key | Default | Extra |
|---|------------|-------------|------|-----|---------|-------|
| ▶ | emp_id | int | NO | PRI | NULL | |
| | f_name | varchar(20) | YES | | NULL | |
| | l_name | varchar(20) | YES | | NULL | |
| | job_id | varchar(10) | YES | | NULL | |
| | salary | int | YES | | NULL | |
| | manager_id | varchar(20) | YES | | NULL | |
| | dept_id | varchar(20) | YES | | NULL | |

```
create table department_datasets (  
dept_id varchar(20) not null primary key,  
dept_name varchar(30),
```

```
location varchar(45),
manager_id varchar(20),
elocation_id varchar(20));
```

```
describe department_datasets;
```



| Field | Type | Null | Key | Default | Extra |
|--------------|-------------|------|-----|---------|-------|
| dept_id | varchar(20) | NO | PRI | NULL | |
| dept_name | varchar(30) | YES | | NULL | |
| location | varchar(45) | YES | | NULL | |
| manager_id | varchar(20) | YES | | NULL | |
| elocation_id | varchar(20) | YES | | NULL | |

4. -- Write a query to insert values into the employee and department tables:

```
insert into employee_datasets (emp_id, f_name, l_name, job_id, salary, manager_id,
dept_id)
values (101, 'Ankit', 'Jain', 'HP124', 200000, 2, 24),
(102, 'Sarvesh', 'Patel', 'HP123', 150000, 2, 24),
(103, 'Krishna', 'Gee', 'HP125', 500000, 5, 44),
(104, 'Rana', 'Gee', 'HP122', 250000, 3, 54),
(105, 'Soniya', 'Jain', 'HP121', 400000, 1, 22),
(106, 'Nithin', 'Kumar', 'HP120', 300000, 4, 34),
(107, 'Karan', 'Patel', 'HP126', 300001, 2, 34),
(108, 'Shilpa', 'Jian', 'HP127', 300001, 5, 24),
(109, 'Mukesh', 'Singh', 'HP128', 300001, 4, 44);
```

```
select * from employee_datasets;
```

| Result Grid | | | | | | | |
|--------------|--------|---------|--------|--------|--------|------------|---------|
| Filter Rows: | | | | | | | |
| | emp_id | f_name | l_name | job_id | salary | manager_id | dept_id |
| ▶ | 101 | Ankit | Jain | HP124 | 200000 | 2 | 24 |
| | 102 | Sarvesh | Patel | HP123 | 150000 | 2 | 24 |
| | 103 | Krishna | Gee | HP125 | 500000 | 5 | 44 |
| | 104 | Rana | Gee | HP122 | 250000 | 3 | 54 |
| | 105 | Soniya | Jain | HP121 | 400000 | 1 | 22 |
| | 106 | Nithin | Kumar | HP120 | 300000 | 4 | 34 |
| | 107 | Karan | Patel | HP126 | 300001 | 2 | 34 |
| | 108 | Shilpa | Jian | HP127 | 300001 | 5 | 24 |
| | 109 | Mukesh | Singh | HP128 | 300001 | 4 | 44 |
| * | NULL | NULL | NULL | NULL | NULL | NULL | NULL |

employees_datasets4

```
insert into department_datasets (dept_id, dept_name, location, manager_id,
elocation_id)
```

```
values (22, 'Administration', 'uk', 1, 218),
(24, 'Production', 'india', 2, 212),
(34, 'development', 'india', 4, 212),
(44, 'communication', 'usa', 5, 220),
(54, 'Maintenance', 'usa', 3, 220);
```

```
select *from department_datasets;
```

| Result Grid | | | | | |
|--------------|---------|----------------|----------|------------|--------------|
| Filter Rows: | | | | | |
| | dept_id | dept_name | location | manager_id | elocation_id |
| ▶ | 22 | Administration | uk | 1 | 218 |
| | 24 | Production | india | 2 | 212 |
| | 34 | development | india | 4 | 212 |
| | 44 | communication | usa | 5 | 220 |
| | 54 | Maintenance | usa | 3 | 220 |
| * | NULL | NULL | NULL | NULL | NULL |

5. -- Write a query to create a view of the employee and department tables:

```
create view emp_1 as select emp_id, f_name, l_name, dept_name, location, salary
from employee_datasets, department_datasets where salary>300000;
```

```
select * from emp_1;
```

| Result Grid | | | | | | |
|-------------|--------|--------------|--------|----------------|----------|--------------------|
| | | Filter Rows: | | Export: | | Wrap Cell Content: |
| | emp_id | f_name | l_name | dept_name | location | salary |
| ▶ | 109 | Mukesh | Singh | Administration | uk | 300001 |
| | 108 | Shilpa | Jian | Administration | uk | 300001 |
| | 107 | Karan | Patel | Administration | uk | 300001 |
| | 105 | Soniya | Jain | Administration | uk | 400000 |
| | 103 | Krishna | Gee | Administration | uk | 500000 |
| | 109 | Mukesh | Singh | Production | india | 300001 |
| | 108 | Shilpa | Jian | Production | india | 300001 |
| | 107 | Karan | Patel | Production | india | 300001 |
| | 105 | Soniya | Jain | Production | india | 400000 |
| | 103 | Krishna | Gee | Production | india | 500000 |

6. -- Write a query to display the first and last names of every employee in the employee table whose salary is greater than the average salary of the employees listed in the SQL basics table:

```
select f_name, l_name from employee_datasets where salary > (select avg(salary)
from employee_datasets);
```

| Result Grid | | |
|-------------|---------|--------------|
| | | Filter Rows: |
| | f_name | l_name |
| ▶ | Krishna | Gee |
| | Soniya | Jain |
| | Karan | Patel |
| | Shilpa | Jian |
| | Mukesh | Singh |

7. -- Write a query to change the delimiter to //:
delimiter //

8. -- Write a query to create a stored procedure in the employee table for every employee whose salary is greater than or equal to 3,00,000:

```
drop procedure if exists salaries;
delimiter &&
create procedure salaries()
begin
select * from employee_datasets where salary>=300000;
end &&
```

9. -- Write a query to execute the stored procedure:

call salaries();

| Result Grid | | | | | | | |
|-------------|--------|--------------|--------|---------|--------|---------------------------------------|---------|
| | | Filter Rows: | | Export: | | Wrap Cell Content: IA | |
| | emp_id | f_name | l_name | job_id | salary | manager_id | dept_id |
| ▶ | 103 | Krishna | Gee | HP125 | 500000 | 5 | 44 |
| | 105 | Soniya | Jain | HP121 | 400000 | 1 | 22 |
| | 106 | Nithin | Kumar | HP120 | 300000 | 4 | 34 |
| | 107 | Karan | Patel | HP126 | 300001 | 2 | 34 |
| | 108 | Shilpa | Jian | HP127 | 300001 | 5 | 24 |
| | 109 | Mukesh | Singh | HP128 | 300001 | 4 | 44 |

10. -- Write a query to create and execute a stored procedure with one parameter using the order by function in descending order of the salary earned:

delimiter &&

create procedure sort_desc_salary(in num int)

begin

select * from employee_datasets order by salary desc limit num;

end &&

call sort_desc_salary(5);

| Result Grid | | | | | | | |
|-------------|--------|--------------|--------|---------|--------|---------------------------------------|---------|
| | | Filter Rows: | | Export: | | Wrap Cell Content: IA | |
| | emp_id | f_name | l_name | job_id | salary | manager_id | dept_id |
| ▶ | 103 | Krishna | Gee | HP125 | 500000 | 5 | 44 |
| | 105 | Soniya | Jain | HP121 | 400000 | 1 | 22 |
| | 107 | Karan | Patel | HP126 | 300001 | 2 | 34 |
| | 108 | Shilpa | Jian | HP127 | 300001 | 5 | 24 |
| | 109 | Mukesh | Singh | HP128 | 300001 | 4 | 44 |