1. Retrieve the first 5 employees by highest salary.

Select * from employees order by salary desc limit 5;

2. Retrieve the 5 employees with the lowest salary, skipping the first 10 records.

Select * from employees order by salary asc limit 5 OFFSET 10;

3. Display each department's total salary, but only show departments where the total salary exceeds $30,000.

1. Employee(eid,enames,e.salary,deptId)

2. Department(id,name)

Select d.department_id, d.department_name,sum(e.salary) as Salary from employees e

join departments d on e.department_id=d.department_id

group by d.department_id

having sum(e.salary) > 30000;

4. Conditional Logic (CASE Statement)

For each employee, display their salary and a note if it is above or below $7000.

```
Select employee_id,Concat(first_name,' ',last_name) as fullName, salary,
CASE
when salary > 7000 then "Above 7000 Salary"
when salary < 7000 then "Below 7000 Salary"
ELSE "Exactly 7000 Salary"
end as salary_note
from employees;
```

5.List the projects that started in the last 6 months.

```
Select project_name, start_date,end_date from projects where start_date >=
date_sub(curdate(),Interval 6 month);
```

6.Display all projects, including those with no employees assigned.

```
SELECT  ep.employee_id,ep.project_id
FROM employeeprojects ep
RIGHT JOIN employees e ON ep.employee_id=e.employee_id;
```

7.Increase the salary of all employees in the 'Finance' department by 12%.

UPDATE employees SET salary = salary * 1.12 where department_id=3;

8.Retrieve the first and last names of employees whose last names start with 'S'.

Select e.first_name,e.last_name from employees e where e.last_name like 's%';

9. Count the number of employees assigned to each department.

Select count(employee_id), department_id from employees group by department_id;

10.Find the total number of hours worked by employees on each project.

Select p.project_id, p.project_name, SUM(ep.hours_worked) AS total_hours FROM projects p LEFT JOIN employeeprojects ep ON p.project_id = ep.project_id

group by p.project_id, p.project_name;

11. Select the employees who earn more than the average salary.

Select * from employees where salary > (Select AVG(salary) from employees);

12. Display each project's start date in the format 'Month Day, Year' (e.g., January 01, 2024).

select project_id, project_name, DATE_FORMAT(start_date, '%M %d, %Y') AS Date from projects;

13.List all employees, their department names, and the projects they have worked on.

Select e.employee_id, e.first_name, e.last_name, d.department_name, p.project_name

from employees e

JOIN departments d ON e.department_id = d.department_id

LEFT JOIN employeeprojects ep ON e.employee_id = ep.employee_id

LEFT JOIN projects p ON ep.project_id = p.project_id;

14.Find the average salary of employees in each department, but only show departments where more than 5 employees work.

Select d.department_id, d.department_name, AVG(e.salary) AS average_salary FROM departments d join employees e ON d.department_id = e.department_id

group by d.department_id, d.department_name having COUNT(e.employee_id) > 5;