

# Employee Management System Using SQL: Insights and Visualization



# Introduction

This report demonstrates the insights derived from SQL queries and visualization in Power BI. It includes the SQL query, its resulting dataset , and the corresponding visualization for each use case.

# Goals

## 1 Actionable Insights

To provide actionable insights from the employee management dataset.

## 2 Integration Showcase

To showcase the integration of SQL queries with Power BI visualizations.

## 3 Decision Support

To assist in effective decision-making through data-driven reporting.

## 4 Framework Establishment

To establish a robust framework for scalable and repeatable reporting practices.

# Key Metrics and Executive Summary

## Key Metrics:

- **Employee Attendance:** Daily and monthly presence patterns.
- **Department Workforce:** Number of employees per department.
- **Project Associations:** Departments linked to specific projects.
- **Compensation:** Salary, bonuses, and net salary details.
- **Employee Engagement:** Ranking departments by workforce and total salaries.

## Executive Summary:

This report uncovers critical insights into workforce dynamics, compensation distribution, and project alignment. By leveraging SQL queries and Power BI visualizations, we provide clear, actionable data for leadership to make informed decisions. High-performing departments and individuals are highlighted, along with areas needing improvement, such as attendance consistency and project management efficiency.

# Query And Visualization Sections:

## Query 1. Get the attendance status of employees on a specific date "2023-01-11"

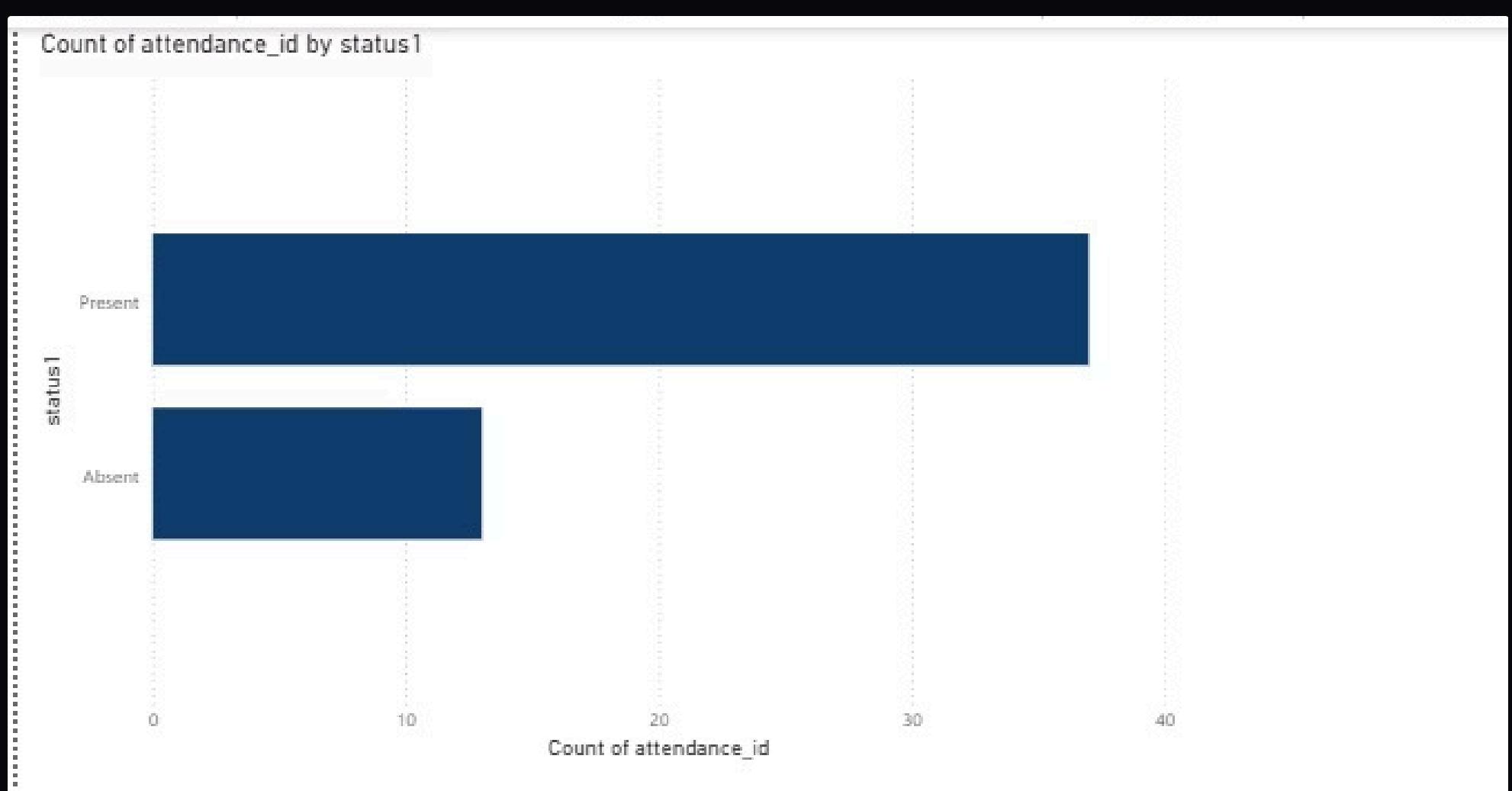
SQL Query:

```
1 • use employee_management_system;
2 • select employee_id, status1,date1
3 • from attendance
4 • where date1= "2023-01-11";
```

Query Result:

	employee_id	status1	date1
1	1	Absent	2023-01-11
2	2	Present	2023-01-11
3	3	Present	2023-01-11
4	4	Present	2023-01-11
5	5	Present	2023-01-11
6	6	Absent	2023-01-11
7	7	Present	2023-01-11
8	8	Present	2023-01-11
9	9	Present	2023-01-11
10	10	Present	2023-01-11
11	11	Absent	2023-01-11
12	12	Present	2023-01-11
13	13	Present	2023-01-11
14	14	Present	2023-01-11

Visualization:



## Query 2. Find all employees in a specific department (dept\_id=102):

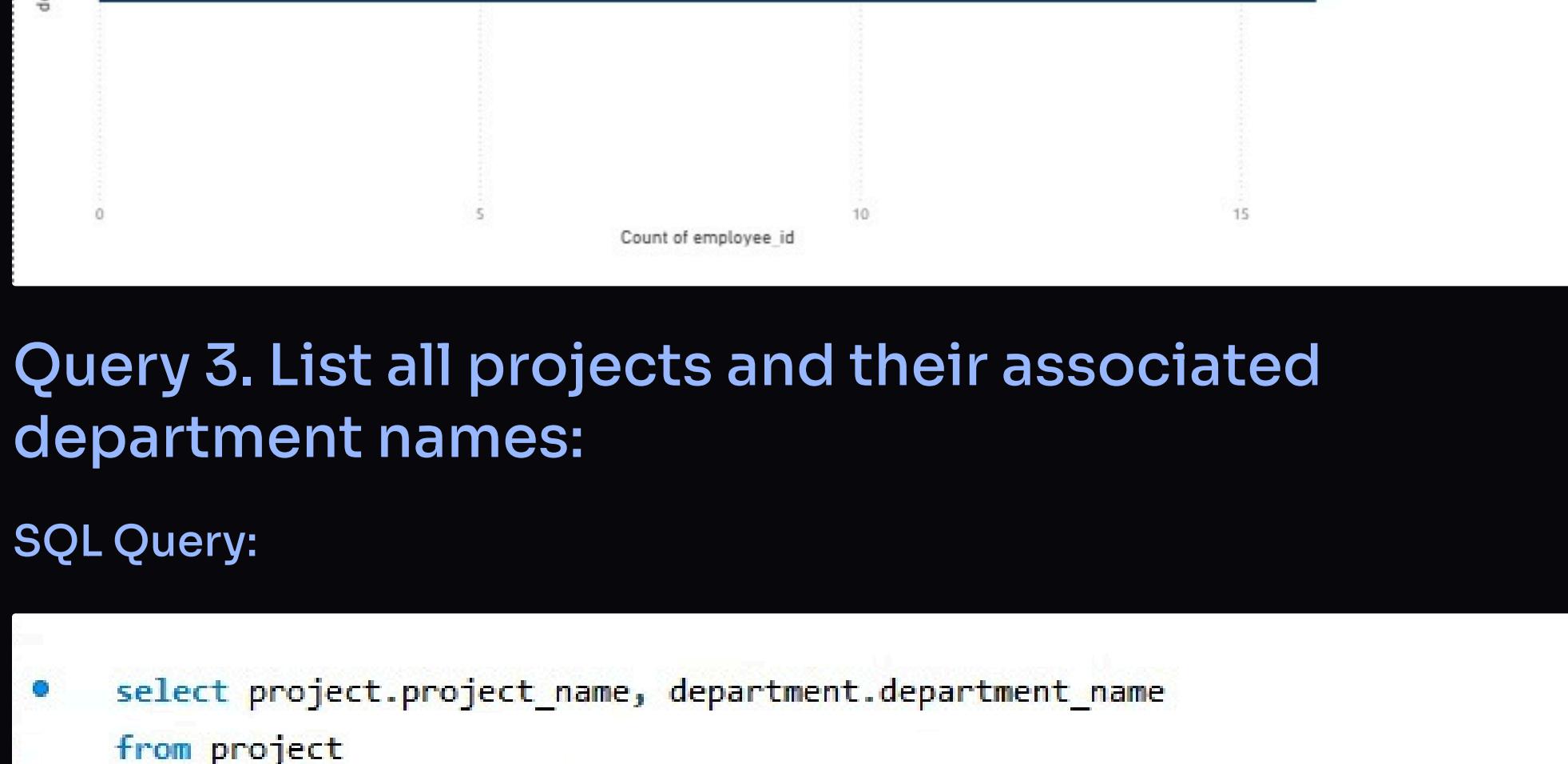
SQL Query:

```
• select employee_id, first_name, last_name, department_id  
  from employee  
 where department_id=102;
```

Query Result:

employee_id	first_name	last_name	department_id
2	Jane	Smith	102
5	Sarah	Davis	102
8	David	Harris	102
12	Benjamin	Walker	102
15	Mia	King	102
18	Ethan	Cooper	102
20	Noah	Jenkins	102
23	Chloe	Perez	102
27	Lily	Bell	102
30	Isaac	Clark	102

Visualization:



## Query 3. List all projects and their associated department names:

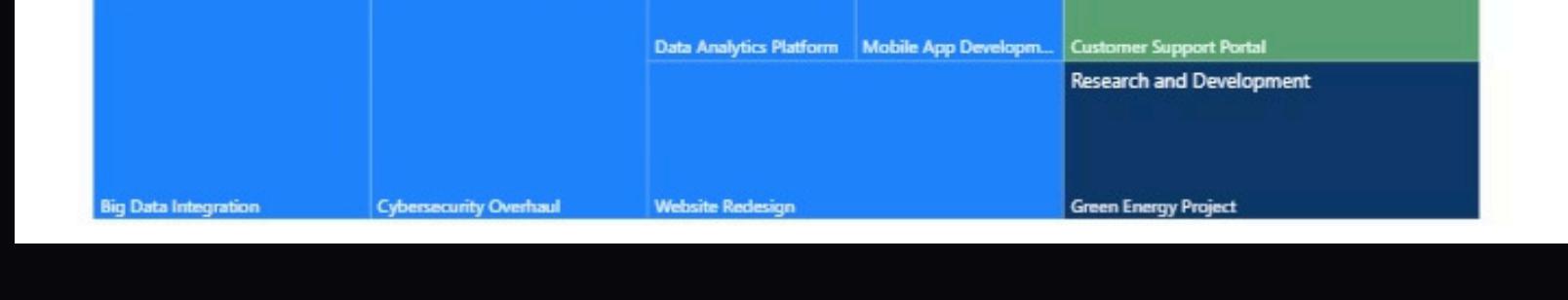
SQL Query:

```
• select project.project_name, department.department_name  
  from project  
 join department on project.department_id = department.department_id;
```

Query Result:

project_name	department_name
Employee Database System	Human Resources
Website Redesign	IT Department
Marketing Campaign	Sales and Marketing
Mobile App Development	IT Department
HR Software Implementation	Human Resources
New Product Launch	Sales and Marketing
Cloud Migration	IT Department
International Expansion	Sales and Marketing
Supply Chain Optimization	Human Resources
Data Analytics Platform	IT Department
Customer Support Portal	Customer Support
Cybersecurity Overhaul	IT Department
Employee Wellness Program	Human Resources

Visualization:



## Query 4. Get the salary details of employees along with their bonuses:

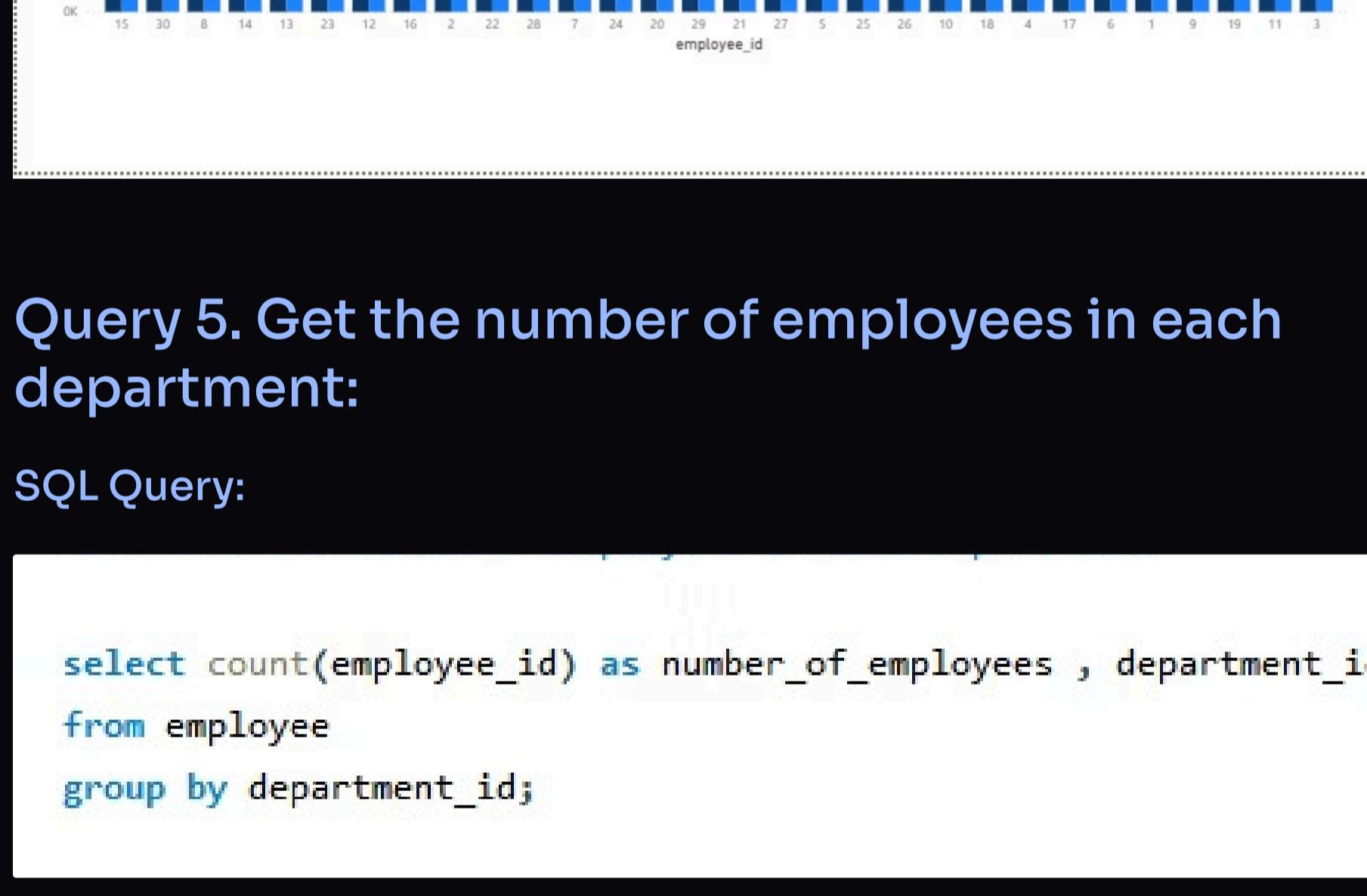
SQL Query:

```
• select employee.first_name, employee.last_name, employee.salary, salary.bonus  
from employee  
join salary on employee.employee_id = salary.employee_id;
```

Query Result:

	first_name	last_name	salary	bonus
▶	John	Doe	60000	5000
	Jane	Smith	75000	7000
	Emily	Johnson	58000	4500
	Michael	Brown	65000	6000
	Sarah	Davis	72000	6500
	William	Taylor	67000	5500
	Olivia	Wilson	71000	7000
	David	Harris	79000	8000
	Sophia	Martinez	62000	5000
	James	Clark	64000	6000
	Isabella	Lewis	63000	4500
	Benjamin	Walker	68000	7000
	Charlotte	Allen	69000	7500
	All	All	74000	7000

Visualization:



## Query 5. Get the number of employees in each department:

SQL Query:

```
select count(employee_id) as number_of_employees , department_id  
from employee  
group by department_id;
```

Query Result:

	number_of_employees	department_id
	11	101
	10	102
	9	103

Visualization:



## Query 7. Get the total number of days each employee was present in a specific month and year (01, 2023).

SQL Query:

```
select employee.employee_id,
       concat (employee.first_name, ' ', employee.last_name) as employee_name,
       count(attendance.attendance_id) as days_present
  from attendance
 join employee on employee.employee_id = attendance.employee_id
 where attendance.status1= 'present' and month(attendance.date1)= 01 and year(attendance.date1) =2023
 group by employee_id, first_name, last_name;
```

Query Result:

	employee_id	employee_name	days_present
	1	John Doe	1
	2	Jane Smith	2
	3	Emily Johnson	1
	4	Michael Brown	2
	5	Sarah Davis	1
	6	William Taylor	1
	7	Olivia Wilson	2
	8	David Harris	1
	9	Sophia Martinez	2
	10	James Clark	2
	11	Isabella Lewis	1
	12	Benjamin Walker	2
	13	Charlotte Allen	1

## Query 8. Calculate the net salary of all employees after bonuses and deductions, and list those whose net salary exceeds a certain threshold.(60000/-)

SQL Query:

```
• select employee.employee_id,
       concat (employee.first_name , ' ', employee.last_name) as employee_name,
       salary.salary as base_salary,
       salary.bonus, salary.deduction,
       (salary.salary+ salary.bonus- salary.deduction) as net_salary
  from salary
 join employee on salary.employee_id = employee.employee_id
 where (salary.salary+ salary.bonus - salary.deduction)> 60000;
```

Query Result:

	employee_id	employee_name	base_salary	bonus	deduction	net_salary
	1	John Doe	60000	5000	1000	64000
	2	Jane Smith	75000	7000	1200	80800
	3	Emily Johnson	58000	4500	800	61700
	4	Michael Brown	65000	6000	1000	70000
	5	Sarah Davis	72000	6500	1100	77400
	6	William Taylor	67000	5500	900	71600
	7	Olivia Wilson	71000	7000	1300	76700
	8	David Harris	79000	8000	1400	85600
	9	Sophia Martinez	62000	5000	1000	66000
	10	James Clark	64000	6000	950	69050
	11	Isabella Lewis	63000	4500	800	66700
	12	Benjamin Walker	68000	7000	1200	73800
	13	Charlotte Allen	69000	7500	1100	75400

## Query 9. Rank departments based on the number of active employees and their total salaries.

SQL Query:

```
• select department.department_id,
       department.department_name,
       count(employee.employee_id) as active_employees,
       sum(employee.salary) as total_salary,
       rank() over (order by count(employee.employee_id) desc , sum(employee.salary) desc ) as department_rank
  from department
  join employee
    on department.department_id = employee.department_id
   where employee.hire_date <= current_date
  group by department.department_id , department.department_name
  order by department_rank;
```

Query Result:

	department_id	department_name	active_employees	total_salary	department_rank
▶	101	Human Resources	11	730000	1
	102	IT Department	10	737000	2
	103	Sales and Marketing	9	623000	3

## 5. Insights and Recommendations

### Attendance Patterns:

Ensure employees are adhering to attendance policies by monitoring high absenteeism rates.

### Resource Allocation:

Optimize workforce deployment based on department sizes and project needs.

### Financial Planning:

Utilize salary and bonus data for better budgeting and payroll management.

### Departmental Growth:

Support underperforming departments by analyzing workforce size and salary distribution.

## 6. Conclusion

The insights presented in this report highlight the power of combining SQL queries with Power BI visualizations to provide actionable data for decision-making. Each section demonstrates how raw data can be transformed into meaningful visuals that are both insightful and easy to interpret.

## 7. Appendices

### Appendix A: SQL Query List

1. Query 1. Get the attendance status of employees on a specific date.
2. Query 2: Find all employees in a specific department.
3. Query 3: List all projects and their associated department names.
4. Query 4: Get the salary details of employees along with their bonuses.
5. Query 5: Get the number of employees in each department.
6. Query 6: Get the total number of days each employee was present in a specific month and year.
7. Query 7: Calculate the net salary of employees exceeding a threshold.
8. Query 8: Rank departments based on the number of active employees and their total salaries.

### Appendix B: Glossary

- **Net Salary:** Total salary after bonuses and deductions.
- **Active Employees:** Employees currently employed and not on leave or inactive status.
- **Threshold Salary:** A benchmark salary used for specific analyses.