

```
1  -- Get Into EMPLOYEE Database
2  Use EMPLOYEE;
3
4  -- Retrieving All Records From Each Table for Verification and Analysis
5
6  -- View All Departments
7  SELECT * FROM Department;
8
9  -- View All Job Titles
10 SELECT * FROM JobTitle;
11
12 -- View All Employees with Associated Details
13 SELECT * FROM Employee;
14
15 -- View Salary Information for Each Employee
16 SELECT * FROM Salary;
17
18 -- View Attendance Records of Employees
19 SELECT * FROM Attendance;
20
21 -- View Details of Projects
22 SELECT * FROM Project;
23
24 -- View Employee to Project Allocations
25 SELECT * FROM ProjectAllocation;
26
27 --      QUESTIONS
28
29 -- 1. Retrieve the first and last names of all employees.
30 SELECT FirstName, LastName
31 FROM Employee;
32
33 -- 2. Retrieve the first and last names of employees who work as 'Software Engineer'.
34 SELECT FirstName, LastName
35 FROM Employee
36 WHERE JobTitleID = (SELECT JobTitleID FROM JobTitle WHERE JobTitleName =
37                    'Software Engineer');
38
39 -- 3. Retrieve first names and last names of last 7 hires
40 SELECT TOP 7 FirstName, LastName, HireDate
41 FROM Employee
42 ORDER BY HireDate DESC;
43
44 -- 4. Get the count of employees in each job title.
45 SELECT JobTitleName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
46 FROM Employee
47 INNER JOIN JobTitle
48 ON Employee.JobTitleID = JobTitle.JobTitleID
49 GROUP BY JobTitleName;
50
51 -- 5. Retrieve the full name & other personal info of employees who work in
52    the 'Engineering' department.
53 SELECT CONCAT(FirstName, ' ', LastName) AS 'FULL NAME', PhoneNumber, Email,
54    Gender, DateOfBirth, DepartmentName
55 FROM Employee
```

```
53 INNER JOIN Department
54 ON Employee.DepartmentID = Department.DepartmentID
55 WHERE DepartmentName = 'Engineering';
56
57 -- 6. List job titles that have more than 3 employees.
58 SELECT JobTitleName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
59 FROM Employee
60 INNER JOIN JobTitle
61 ON Employee.EmployeeID = JobTitle.JobTitleID
62 GROUP BY JobTitleName
63 HAVING COUNT(EmployeeID) > 3;
64
65 -- 7. Retrieve all employee names along with their department names.
66 SELECT FirstName, LastName, DepartmentName
67 FROM Employee
68 INNER JOIN Department
69 ON Employee.DepartmentID = Department.DepartmentID;
70
71 -- 8. Retrieve the first names of employees and the projects they are working
    on, along with their role in the project.
72 SELECT FirstName, ProjectName, JobTitleName AS ROLE
73 FROM Employee
74 INNER JOIN ProjectAllocation
75 ON Employee.EmployeeID = ProjectAllocation.EmployeeID
76 INNER JOIN Project
77 ON ProjectAllocation.ProjectID = Project.ProjectID
78 INNER JOIN JobTitle
79 ON Employee.JobTitleID = JobTitle.JobTitleID;
80
81 -- 9. Get the count of employees in each department
82 SELECT DepartmentName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
83 FROM Employee
84 INNER JOIN Department
85 ON Employee.DepartmentID = Department.DepartmentID
86 GROUP BY DepartmentName;
87
88 -- 10. List all departments with more than 5 employees.
89 SELECT DepartmentName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
90 FROM Employee
91 INNER JOIN Department
92 ON Employee.DepartmentID = Department.DepartmentID
93 GROUP BY DepartmentName
94 HAVING COUNT(EmployeeID) > 5;
95
96 -- 11. Retrieve the full names of employees and their managers.
97 SELECT CONCAT(E.FirstName, ' ', E.LastName) AS 'FULL NAME',
98         CONCAT(M.FirstName, ' ', M.LastName) AS 'MANAGER NAME'
99 FROM Employee E
100 INNER JOIN Employee M
101 ON E.EmployeeID = M.ManagerID;
102
103 -- 12. Which manager is managing more employees and how many
104 SELECT TOP 1
105         CONCAT(M.FirstName, ' ', M.LastName) AS 'FULL NAME',
106         COUNT(E.EmployeeID) AS 'NUMBER OF EMPLOYEES'
107 FROM Employee E
```

```
108 INNER JOIN Employee M
109 ON E.EmployeeID = M.ManagerID
110 GROUP BY M.EmployeeID, M.FirstName, M.LastName
111 ORDER BY 2 DESC;
112
113 -- 13. Retrieve names of employees working on projects as 'Software Engineer', ↗
      ordered by project start date
114 SELECT FirstName, LastName, ProjectName, StartDate
115 FROM Employee
116 INNER JOIN ProjectAllocation
117 ON Employee.EmployeeID = ProjectAllocation.EmployeeID
118 INNER JOIN Project
119 ON ProjectAllocation.ProjectID = Project.ProjectID
120 INNER JOIN JobTitle
121 ON Employee.JobTitleID = JobTitle.JobTitleID
122 WHERE JobTitleName = 'Software Engineer'
123 ORDER BY 4;
124
125 -- 14. Retrieve the names of employees who are working on 'Project Delta'.
126 SELECT FirstName, LastName
127 FROM Employee
128 WHERE EmployeeID IN (SELECT EmployeeID FROM ProjectAllocation
129                      WHERE ProjectID = (SELECT ProjectID FROM Project WHERE ↗
                                           ProjectName = 'Project Delta'));
130
131 -- 15. Retrieve the names of employees, department name, and total salary, ↗
      ordered by total salary in descending order
132 SELECT FirstName, LastName, DepartmentName, (BaseSalary + Bonus - Deductions) ↗
      AS 'TOTAL SALARY'
133 FROM Employee
134 INNER JOIN Department
135 ON Employee.DepartmentID = Department.DepartmentID
136 INNER JOIN Salary
137 ON Employee.EmployeeID = Salary.EmployeeID
138 ORDER BY 4 DESC ;
139
140 -- 16. Create a function to find employees with a birthday in the given month ↗
      and calculate their age
141 CREATE FUNCTION dbo.fn_GetBirthday(@Month INT)
142 RETURNS TABLE
143 AS
144 RETURN
145 (
146     SELECT FirstName, LastName, DateOfBirth,
147            MONTH(DateOfBirth) AS BirthMonth,
148            YEAR(GETDATE()) - YEAR(DateOfBirth) Age
149     FROM Employee
150     WHERE MONTH(DateOfBirth) = @Month
151 );
152
153 -- Find employees who have a birthday in November and their age
154 SELECT * FROM DBO.fn_GetBirthday(11);
155
156 -- Find employees who have a birthday in March and their age
157 SELECT * FROM DBO.fn_GetBirthday(3);
158
```

```
159 -- 17. Create a function to find employees in a specified department and calculate their years of service
160 CREATE FUNCTION fn_GetEmployeesByDept(@DeptName VARCHAR(100))
161 RETURNS TABLE
162 AS
163 RETURN
164 (
165     SELECT EmployeeID, FirstName, LastName, DepartmentName, HireDate,
166            DATEDIFF(YEAR, HireDate, GETDATE()) AS 'YEARS OF SERVICE'
167     FROM Employee
168     INNER JOIN Department
169     ON Employee.DepartmentID = Department.DepartmentID
170     WHERE DepartmentName = @DeptName
171 );
172 -- Find employees in the IT department and their years of service
173 SELECT * FROM DBO.fn_GetEmployeesByDept('Engineering');
174
175 -- Find employees in the HR department and their years of service
176 SELECT * FROM DBO.fn_GetEmployeesByDept('Human Resources');
```