```sql
-- Get Into EMPLOYEE Database
Use EMPLOYEE;

-- Retrieving All Records From Each Table for Verification and Analysis

-- View All Departments
SELECT * FROM Department;

-- View All Job Titles
SELECT * FROM JobTitle;

-- View All Employees with Associated Details
SELECT * FROM Employee;

-- View Salary Information for Each Employee
SELECT * FROM Salary;

-- View Attendance Records of Employees
SELECT * FROM Attendance;

-- View Details of Projects
SELECT * FROM Project;

-- View Employee to Project Allocations
SELECT * FROM ProjectAllocation;

--     QUESTIONS

-- 1. Retrieve the first and last names of all employees.
SELECT FirstName, LastName
FROM Employee;

-- 2. Retrieve the first and last names of employees who work as 'Software Engineer'.
SELECT FirstName, LastName
FROM Employee
WHERE JobTitleID = (SELECT JobTitleID FROM JobTitle WHERE JobTitleName = 'Software Engineer');

-- 3. Retrieve first names and last names of last 7 hires
SELECT TOP 7 FirstName, LastName, HireDate
FROM Employee
ORDER BY HireDate DESC;

-- 4. Get the count of employees in each job title.
SELECT JobTitleName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
FROM Employee
INNER JOIN JobTitle
ON Employee.JobTitleID = JobTitle.JobTitleID
GROUP BY JobTitleName;

-- 5. Retrieve the full name & other personal info of employees who work in the 'Engineering' department.
SELECT CONCAT(FirstName,' ',LastName) AS 'FULL NAME', PhoneNumber, Email, Gender, DateOfBirth, DepartmentName
FROM Employee
```

```sql
INNER JOIN Department
ON Employee.DepartmentID = Department.DepartmentID
WHERE DepartmentName = 'Engineering';

-- 6. List job titles that have more than 3 employees.
SELECT JobTitleName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
FROM Employee
INNER JOIN JobTitle
ON Employee.EmployeeID = JobTitle.JobTitleID
GROUP BY JobTitleName
HAVING COUNT(EmployeeID) > 3;

-- 7. Retrieve all employee names along with their department names.
SELECT FirstNAME, LastName, DepartmentName
FROM Employee
INNER JOIN Department
ON Employee.DepartmentID = Department.DepartmentID;

-- 8. Retrieve the first names of employees and the projects they are working on, along with their role in the project.
SELECT FirstName, ProjectName, JobTitleName AS ROLE
FROM Employee
INNER JOIN ProjectAllocation
ON Employee.EmployeeID = ProjectAllocation.EmployeeID
INNER JOIN Project
ON ProjectAllocation.ProjectID = Project.ProjectID
INNER JOIN JobTitle
ON Employee.JobTitleID = JobTitle.JobTitleID;

-- 9. Get the count of employees in each department
SELECT DepartmentName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
FROM Employee
INNER JOIN Department
ON Employee.DepartmentID = Department.DepartmentID
GROUP BY DepartmentName;

-- 10. List all departments with more than 5 employees.
SELECT DepartmentName, COUNT(EmployeeID) AS 'EMPLOYEE COUNT'
FROM Employee
INNER JOIN Department
ON Employee.DepartmentID = Department.DepartmentID
GROUP BY DepartmentName
HAVING COUNT(EmployeeID) > 5;

-- 11. Retrieve the full names of employees and their managers.
SELECT CONCAT(E.FirstName,' ',E.LastName) AS 'FULL NAME',
       CONCAT(M.FirstName,' ',M.LastName) AS 'MANAGER NAME'
FROM Employee E
INNER JOIN Employee M
ON E.EmployeeID = M.ManagerID;

-- 12. Which manager is managing more employees and how many
SELECT TOP 1
    CONCAT(M.FirstName,' ',M.LastName) AS 'FULL NAME',
    COUNT(E.EmployeeID) AS 'NUMBER OF EMPLOYEES'
FROM Employee E
```

```sql
INNER JOIN Employee M
ON E.EmployeeID = M.ManagerID
GROUP BY M.EmployeeID, M.FirstName, M.LastName
ORDER BY 2 DESC;

-- 13. Retrieve names of employees working on projects as 'Software Engineer', ordered by project start date
SELECT FirstName, LastName, ProjectName, StartDate
FROM Employee
INNER JOIN ProjectAllocation
ON Employee.EmployeeID = ProjectAllocation.EmployeeID
INNER JOIN Project
ON ProjectAllocation.ProjectID = Project.ProjectID
INNER JOIN JobTitle
ON Employee.JobTitleID = JobTitle.JobTitleID
WHERE JobTitleName = 'Software Engineer'
ORDER BY 4;

-- 14. Retrieve the names of employees who are working on 'Project Delta'.
SELECT FirstName, LastName
FROM Employee
WHERE EmployeeID IN (SELECT EmployeeID FROM ProjectAllocation
                    WHERE ProjectID = (SELECT ProjectID FROM Project WHERE ProjectName = 'Project Delta'));

-- 15. Retrieve the names of employees, department name, and total salary, ordered by total salary in descending order
SELECT FirstName, LastName, DepartmentName, (BaseSalary + Bonus - Deductions) AS 'TOTAL SALARY'
FROM Employee
INNER JOIN Department
ON Employee.DepartmentID = Department.DepartmentID
INNER JOIN Salary
ON Employee.EmployeeID = Salary.EmployeeID
ORDER BY 4 DESC ;

-- 16. Create a function to find employees with a birthday in the given month and calculate their age
CREATE FUNCTION dbo.fn_GetBirthday(@Month INT)
RETURNS TABLE
AS
RETURN
(
    SELECT FirstName, LastName, DateOfBirth,
           MONTH(DateOfBirth) AS BirthMonth,
           YEAR(GETDATE()) - YEAR(DateOfBirth) Age
    FROM Employee
    WHERE MONTH(DateOfBirth) = @Month
);

  -- Find employees who have a birthday in November and their age
SELECT * FROM DBO.fn_GetBirthday(11);

  -- Find employees who have a birthday in March and their age
SELECT * FROM DBO.fn_GetBirthday(3);

```

```sql
-- 17. Create a function to find employees in a specified department and calculate their years of service
CREATE FUNCTION fn_GetEmployeesByDept(@DeptName VARCHAR(100))
RETURNS TABLE
AS
RETURN
(
    SELECT EmployeeID, FirstName, LastName, DepartmentName, HireDate,
           DATEDIFF(YEAR, HireDate, GETDATE()) AS 'YEARS OF SEVICE'
    FROM Employee
    INNER JOIN Department
    ON Employee.DepartmentID = Department.DepartmentID
    WHERE DepartmentName = @DeptName
);
  -- Find employees in the IT department and their years of service
SELECT * FROM DBO.fn_GetEmployeesByDept('Engineering');

  -- Find employees in the HR department and their years of service
SELECT * FROM DBO.fn_GetEmployeesByDept('Human Resources');
```