```
pragma solidity ^0.8.0;

contract EmployeeData {
    struct Employee {
        uint id;
        string name;
        uint age;
        string position;
        uint salary;
    }

    Employee[] public employees;
    mapping(uint => uint) private employeeIndex;

    fallback() external payable {}

    function addEmployee(uint _id, string memory _name, uint _age, string memory _position, uint _salary) public {
        require(employeeIndex[_id] == 0, "Employee ID already exists");
        employees.push(Employee(_id, _name, _age, _position, _salary));
        employeeIndex[_id] = employees.length;
    }

    function getEmployee(uint _id) public view returns (Employee memory) {
        require(employeeIndex[_id] > 0, "Employee does not exist");
        return employees[employeeIndex[_id] - 1];
    }

    function updateEmployee(uint _id, string memory _name, uint _age, string memory _position, uint _salary) public {
        require(employeeIndex[_id] > 0, "Employee does not exist");
        uint index = employeeIndex[_id] - 1;
        employees[index].name = _name;
        employees[index].age = _age;
        employees[index].position = _position;
        employees[index].salary = _salary;
    }

    function deleteEmployee(uint _id) public {
        require(employeeIndex[_id] > 0, "Employee does not exist");
        uint index = employeeIndex[_id] - 1;
        employees[index] = employees[employees.length - 1];
        employeeIndex[employees[index].id] = index + 1;
        employees.pop();
        delete employeeIndex[_id];
    }

    function getTotalEmployees() public view returns (uint) {
        return employees.length;
    }
}
```