// create employee details  
const employees = [  
  { id: 1, name: "Madavan", department: "HR", salary: 40000, bonusPercentage: 10 },  
  { id: 2, name: "Kannan", department: "Engineering", salary: 60000, yearsOfExperience: 3 },  
  { id: 3, name: "Ananthi", department: "Sales", salary: 50000, sales: 200000 },  
  { id: 1, name: "Madavan", department: "HR", salary: 40000, bonusPercentage: 10 }, // Duplicate  
];  
// Department Bonus Criteria  
const bonusCriteria = {  
  HR: (employee) => (employee.salary < 50000 ? 0.1 : 0),  
  Engineering: (employee) => (employee.yearsOfExperience > 2 ? 0.15 : 0),  
  Sales: (employee) => {  
    if (employee.sales < 100000) return 0.05;  
    if (employee.sales <= 500000) return 0.1;  
   else  
 return 0.2;  
  },  
};  
  
// Remove duplicates based on id  
const uniqueEmployees = Array.from(new Set(employees.map((emp) => [emp.id](http://emp.id/" \t "_blank))))  
  .map((id) => employees.find((emp) => [emp.id](http://emp.id/" \t "_blank) === id));  
 console.log (uniqueEmployees);  
  
// Filter employees who belong to the "Sales" department  
const SalesEmployees = employees.filter((employee) => employee.department === "Sales");  
console.log(SalesEmployees);  
// calculation of baseSalaries  
const baseSalaries = employees.map((employee) => employee.salary);  
console.log({baseSalaries});  
// calculation of totalBaseSalary    
const totalBaseSalary = employees.reduce((total, employee) => total + employee.salary, 0);  
console.log({totalBaseSalary});  
//  totalSalary of all department  
for (let i = 0; i < employees.length; i++) {  
  if (employees[i].department === "HR" && i === employees.findIndex((emp) => emp.department === "HR")) {  
    const totalSalaryHrDepartment = employees.filter((employee) => employee.department === "HR").reduce((total, employee) => total + employee.salary, 0);  
    console.log({ totalSalaryHrDepartment });  
  } else if (employees[i].department === "Engineering") {  
    const totalSalaryEngineeringDepartment = employees.filter((employee) => employee.department === "Engineering").reduce((total, employee) => total + employee.salary, 0);  
    console.log({ totalSalaryEngineeringDepartment });  
  } else if (employees[i].department === "Sales") {  
    const totalSalarySalesDepartment = employees.filter((employee) => employee.department === "Sales").reduce((total, employee) => total + employee.salary, 0);  
    console.log({ totalSalarySalesDepartment });  
  }  
}  
// TotalCompensation  
const employeesWithTotalCompensation = uniqueEmployees.map((employee) => {const departmentBonusPercentage = bonusCriteria[employee.department]? bonusCriteria[employee.department](employee) : 0;  
const totalBonusPercentage = (employee.bonusPercentage || 0) + departmentBonusPercentage;  
const bonus = employee.salary \* totalBonusPercentage;  
const totalCompensation = employee.salary + bonus;  
  
  return { ...employee, bonus, totalCompensation };  
});  
  
console.log(employeesWithTotalCompensation);