SQL Assignment - Query Analysis and Solutions

Table Data Overview

Departments

ID	Name	Location
1	HR	New York
2	IT	San Jose
3	Finance	Chicago
4	Marketing	New York

Employee

ID	Name	Dept_ID	Salary	Hire_Date
101	Alice	2	80000	2020-01-15
102	Bob	2	90000	2019-03-20
103	Charlie	1	60000	2021-07-30
104	David	3	75000	2018-11-10
105	Eve	4	72000	2022-05-01
106	Frank	4	55000	2023-02-14

Projects

ID	Name	Start_Date	End_Date	Budget
201	Alpha	2023-01-01	2023-06-30	500000
202	Beta	2022-05-01	2022-12-31	300000
203	Gamma	2021-09-15	2022-03-15	250000
204	Delta	2023-03-01	2023-08-30	600000

Employee_Projects (Employee_ID, Project_ID, Hours)

- All employees are assigned to projects (no unassigned employees)

Query Analysis Summary

❌ Major Issues Found

- **Query 1 & 9**: Looking for 'John' who doesn't exist in the data
- **Query 5**: Incorrect logic and non-existent 'Sales' department
- **Query 6**: Incorrect expectation - all employees are assigned to projects

✅ Queries Working Correctly

- **Query 2 & 10**: Average salary queries
- **Query 3**: Departments with fewer than 3 employees
- **Query 4**: Highest salary employee
- **Query 7**: Employees in New York departments
- **Query 8**: Projects starting after latest hire date

Detailed Query Analysis

Query 1 & 9: Find employees who work in the same department as 'John'

❌ **FUNDAMENTAL ISSUE:** There is no employee named 'John' in the data!

Available employees: Alice, Bob, Charlie, David, Eve, Frank

Original Query:

```sql
select *
from employee
where department_id = (select department_id
                       from employee
                       where name = 'JHON') -- or 'Jhon'
```

Result: No rows selected (correct result - John doesn't exist)

If we wanted employees in same dept as 'Alice':

```sql
SELECT *
FROM employee
WHERE department_id = (SELECT department_id
                       FROM employee
                       WHERE name = 'Alice')
```

Would return: Alice and Bob (both in IT department)

Query 2 & 10: List employees earning more than average salary ✅

Status: CORRECT

Average salary calculation: (80000+90000+60000+75000+72000+55000)/6 = 72,000

Query:

sql

```sql
select name
from employee
where salary > (select avg(salary)
                from employee);
```

Correct Result: Alice (80000), Bob (90000), David (75000) **Your Result:** Alice, Bob, David ✅

Query 3: Find all departments with fewer than 3 employees ✅

Status: CORRECT after fix

Department employee counts:

- HR (1): Charlie (1 employee)
- IT (2): Alice, Bob (2 employees)
- Finance (3): David (1 employee)
- Marketing (4): Eve, Frank (2 employees)

All departments have fewer than 3 employees

Final Query:

sql

```sql
select count(*), department_id
from employee
group by department_id
having count(*) < 3
```

Your Result: All 4 departments returned ✅

Query 4: Get employee with highest salary ✅

Status: CORRECT

Highest salary: Bob with 90,000

Query:

```sql
select name
from employee
where salary = (select max(salary)
                from employee);
```

Your Result: Bob ✅

Query 5: List employees who joined after earliest hire date in Sales department

❌ **FUNDAMENTAL ISSUE:** There is no 'Sales' department!

Available departments: HR, IT, Finance, Marketing

Original Query Logic: Also incorrect - uses `=` instead of `>`

If we fix for 'IT' department (earliest hire: Bob on 2019-03-20):

```sql
SELECT *
FROM employee
WHERE hire_date > (SELECT MIN(hire_date)
                   FROM employee
                   WHERE department_id = (SELECT department_id
                                          FROM departments
                                          WHERE department_name = 'IT'));
```

Would return: Alice (joined 2020-01-15, after Bob's 2019-03-20)

Query 6: Display employees not assigned to any project

❌ **EXPECTATION ISSUE:** All employees are assigned to projects

Employee-Project assignments:

- Alice: Projects 201, 202
- Bob: Projects 201, 203
- Charlie: Project 204
- David: Project 202
- Eve: Project 204
- Frank: Project 204

Query:

```sql
select e.*, ep.*
from employee e left join employee_projects ep
on e.employee_id = ep.employee_id
where ep.employee_id is null;
```

Your Result: No rows selected ✅ (Correct - no unassigned employees)

Query 7: Find employees in departments located in 'New York' ✅

Status: CORRECT after typo fix

New York departments: HR (1), Marketing (4)

Employees in New York: Charlie (HR), Eve (Marketing), Frank (Marketing)

Final Query:

```sql
select *
from employee
where department_id in (select department_id
                        from departments
                        where location = 'New York')
```

Your Result: Charlie, Eve, Frank ✅

Query 8: Find projects that started after latest employee hire date ✅

Status: CORRECT after fixes

Latest hire date: Frank on 2023-02-14

Projects starting after 2023-02-14:

- Only Delta (2023-03-01) starts after Frank's hire date

Final Query:

```sql
SELECT *
FROM projects
WHERE start_date > (SELECT MAX(hire_date) FROM employee);
```

Your Result: Delta project ✅

Summary of Corrections Needed

Queries That Need Data-Based Fixes:

1. **Query 1 & 9:** Change 'John' to an existing employee name
2. **Query 5:** Change 'Sales' to an existing department name AND fix the logic (= to >)

Queries That Worked Correctly:

- Query 2, 3, 4, 6, 7, 8, 10

Key Insights:

- Your SQL syntax and logic were mostly correct
- Main issues were data mismatches (non-existent names/departments)
- The "no rows selected" results were actually correct given the data
- All employees are assigned to projects, so Query 6 correctly returns nothing

Recommended Test Cases:

sql

```sql
-- Alternative Query 1: Find employees in same dept as Alice
SELECT * FROM employee
WHERE department_id = (SELECT department_id FROM employee WHERE name = 'Alice');

-- Alternative Query 5: Employees who joined after earliest in IT
SELECT * FROM employee
WHERE hire_date > (SELECT MIN(hire_date) FROM employee
                   WHERE department_id = (SELECT department_id FROM departments
                                          WHERE department_name = 'IT'));
```