1. π\_emp\_id, first\_name, last\_name, salary, department\_id(σdepartment\_name='Engineering'(employees ⋈ departments))
2. Projection to display only the first names and salaries of all employees: π\_first\_name, salary(employees)
3. Find employees who are managers: π\_emp\_id, first\_name, last\_name, salary, department\_id(employees ⋈ σ\_manager\_id=emp\_id(departments))
4. Retrieve employees earning a salary greater than ?60000: σ\_salary > 60000(employees)
5. Join employees with their respective departments: employees ⋈ departments
6. Cartesian product between employees and projects: employees × projects
7. Find employees who are not managers: π\_emp\_id, first\_name, last\_name, salary, department\_id(employees - (employees ⋈ σ\_manager\_id=emp\_id(departments)))
8. Natural join between departments and projects: departments ⋈ projects
9. Project the department names and locations from departments table: π\_department\_name, location(departments)
10. Retrieve projects with budgets greater than ?100000: σ\_budget > 100000(projects)
11. Find employees who are managers in the 'Sales' department: π\_emp\_id, first\_name, last\_name, salary, department\_id(σ\_department\_name='Sales'(employees) ⋈ σ\_manager\_id=emp\_id(departments))
12. Union operation between two sets of employees from the 'Engineering' and 'Finance' departments: π\_emp\_id, first\_name, last\_name, salary, department\_id(σ\_department\_name='Engineering'(employees)) ∪ π\_emp\_id, first\_name, last\_name, salary, department\_id(σ\_department\_name='Finance'(employees))
13. Find employees who are not assigned to any projects: π\_emp\_id, first\_name, last\_name, salary, department\_id(employees - (employees ⋈ projects))
14. Join operation to display employees along with their project assignments: employees ⋈ projects
15. Find employees whose salaries are not within the range ?50000 to ?70000: π\_emp\_id, first\_name, last\_name, salary, department\_id(σ\_salary < 50000 ∨ salary > 70000(employees))