# Copilot

Let's complete the Employee Database with the necessary data and then perform the queries.

### Employee Database

#### Creating the Data

##### Employees

CREATE (:Employee {name: "John Doe"});

CREATE (:Employee {name: "Jane Smith"});

CREATE (:Employee {name: "Robert Brown"});

CREATE (:Employee {name: "Emily Davis"});

CREATE (:Employee {name: "Michael Johnson"});

##### Departments

CREATE (:Department {name: "HR"});

CREATE (:Department {name: "Engineering"});

CREATE (:Department {name: "Marketing"});

##### Projects

CREATE (:Project {name: "Project Alpha"});

CREATE (:Project {name: "Project Beta"});

CREATE (:Project {name: "Project Gamma"});

##### Skills

CREATE (:Skill {name: "Java"});

CREATE (:Skill {name: "Scala"});

CREATE (:Skill {name: "Project Management"});

CREATE (:Skill {name: "Python"});

CREATE (:Skill {name: "Data Analysis"});

##### Relationships

MATCH (e:Employee {name: "John Doe"}), (d:Department {name: "Engineering"})

CREATE (e)-[:WORKS\_IN]->(d);

MATCH (e:Employee {name: "Jane Smith"}), (d:Department {name: "HR"})

CREATE (e)-[:WORKS\_IN]->(d);

MATCH (e:Employee {name: "Robert Brown"}), (d:Department {name: "Marketing"})

CREATE (e)-[:WORKS\_IN]->(d);

MATCH (e:Employee {name: "Emily Davis"}), (d:Department {name: "Engineering"})

CREATE (e)-[:WORKS\_IN]->(d);

MATCH (e:Employee {name: "Michael Johnson"}), (d:Department {name: "HR"})

CREATE (e)-[:WORKS\_IN]->(d);

MATCH (p:Project {name: "Project Alpha"}), (d:Department {name: "Engineering"})

CREATE (p)-[:CONTROLLED\_BY]->(d);

MATCH (p:Project {name: "Project Beta"}), (d:Department {name: "HR"})

CREATE (p)-[:CONTROLLED\_BY]->(d);

MATCH (p:Project {name: "Project Gamma"}), (d:Department {name: "Marketing"})

CREATE (p)-[:CONTROLLED\_BY]->(d);

MATCH (e:Employee {name: "John Doe"}), (s:Skill {name: "Java"})

CREATE (e)-[:HAS\_SKILL]->(s);

MATCH (e:Employee {name: "Jane Smith"}), (s:Skill {name: "Project Management"})

CREATE (e)-[:HAS\_SKILL]->(s);

MATCH (e:Employee {name: "Robert Brown"}), (s:Skill {name: "Data Analysis"})

CREATE (e)-[:HAS\_SKILL]->(s);

MATCH (e:Employee {name: "Emily Davis"}), (s:Skill {name: "Scala"})

CREATE (e)-[:HAS\_SKILL]->(s);

MATCH (e:Employee {name: "Michael Johnson"}), (s:Skill {name: "Python"})

CREATE (e)-[:HAS\_SKILL]->(s);

#### Queries

a) List the names of employees having the same skills as employee "John Doe":

MATCH (e1:Employee {name: "John Doe"})-[:HAS\_SKILL]->(s:Skill)<-[:HAS\_SKILL]-(e2:Employee)

RETURN e2.name;

**Result:** John Doe

b) List the projects controlled by a department "Engineering" and have employees of the same department working in it:

MATCH (p:Project)-[:CONTROLLED\_BY]->(d:Department {name: "Engineering"}), (e:Employee)-[:WORKS\_IN]->(d)

RETURN p.name;

**Result:** Project Alpha

c) List the names of the projects belonging to departments managed by employee "Jane Smith":

MATCH (e:Employee {name: "Jane Smith"})-[:WORKS\_IN]->(d:Department)<-[:CONTROLLED\_BY]-(p:Project)

RETURN p.name;

**Result:** Project Beta

This completes the Employee Database and the queries based on the provided data. Let me know if you need any further assistance!

Source(s)

1. [Employees Sample Database :: 5 Employees Structure - MySQL](https://dev.mysql.com/doc/employee/en/sakila-structure.html)

2. [Employee Sample Database for MySQL - GitHub](https://github.com/DataStudySquad/employee-sample-database-mysql)

3. [Create an Employee Database In 3 Steps - Five](https://five.co/blog/create-an-employee-database-in-3-steps/)

4. [What Is a Database Schema? Types, Use Cases, & Examples](https://estuary.dev/database-schema/)

5. [Employee Database Diagram - Creately](https://creately.com/diagram/example/h32iuc7l1/employee-database-diagram)

6. <https://github.com/datacharmer/test_db>