1. Choose a title:

Factory

2. Create a list of data items that will be stored in your project.

managerID	salary	name
111	50000	Alex
222	60000	Jane
333	55000	Paul
444	55000	Yin

employeeID	name	position	managerID
101	Abdullah	laborer	111
102	Saad	laborer	222
103	Jake	laborer	333
104	Linda	Worker	222
105	Shan	Worker	222
601	Gab	Operator	444
602	Farhad	Operator	444
603	Sid	Operator	444

departmentID	name	location	managerID
301	HR	Building A	111
302	Drilling	Building B	222
303	Operator	Building C	444
304	Labour	Building D	333

machineID	name	departmentID
501	Drilling Machine	301
502	Krane	302
503	Compressor	303

partNumber	quantity	machineID	color
1001	50	501	Red
1002	30	502	Green
1003	20	503	Blue

operatorID	name	machineID
601	Gab	501
602	Farhad	502
603	Sid	503

3. Find and list all entities:

Manager Entity

Employee Entity

Machine Entity

Department Entity

Machine Operator Entity

Part Entity (weak entity)

4. Find the attributes of entities.

Manager Entity
Attributes: managerID, salary, managerName

Employee Entity
Attributes: employeeID, employeeName, position

Machine Entity
Attributes: machineID, machineName

Department Entity
Attributes: departmentID, departmentName, location

Machine Operator Entity
Attributes: operatorID, operatorName

Part Entity (weak entity)
Attributes: partNumber, quantity, color

5. Define the data type for each attribute.

Manager Entity
Attributes:
managerID (INT),
Salary (DECIMAL),
managerName (VARCHAR),

Employee Entity
Attributes:
employeeID(INT),
employeeName (VARCHAR),
Position (VARCHAR),

Machine Entity
Attributes:
machineID (INT),
machineName (VARCHAR)

Department Entity
Attributes:
departmentID (INT),
departmentName (VARCHAR),
location (VARCHAR)

Machine Operator Entity
Attributes:
operatorID (INT),
operatorName (VARCHAR)

Part Entity (weak entity)
Attributes:
partNumber(INT),
quantity (INT),
color(VARCHAR)

6. Define relationships between entities. You should have at least one weak entity, one recursive relationship and a class hierarchy.

Employee (super class) **ISA** Manager (child class)

Manager **manages** Department

Machine **assignedTo** Department

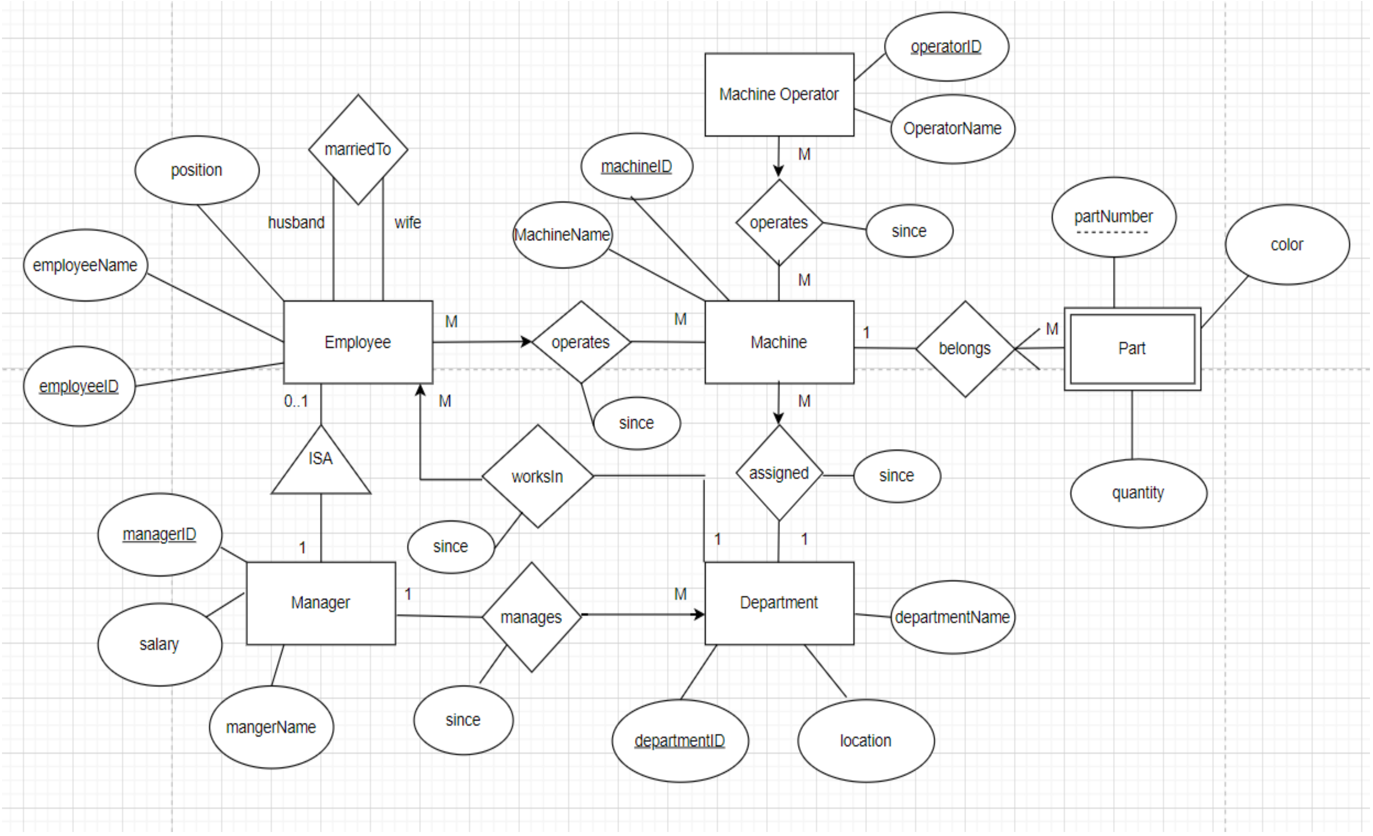
Machine Operator **operates** Machine

Employee **operates** Machine

Part (weak entity) **belongs** Machine

Employee **marriedTo** Employee (recursive relationship)

7. Create UML or ER model of your database.



8. Convert the UML or ER model to relational data model (database schemas). Show Primary Keys and Foreign Keys for each table (total 6-8 tables).

ISA:

```
Employee (
   employeeID INT PRIMARY KEY,
   employeeName VARCHAR(255),
   position VARCHAR(255),
   managerID INT
   FOREIGN KEY (managerID) REFERENCES Manager(managerID)
);
```

ISA:

```
Manager (
   managerID INT PRIMARY KEY,
   salary DECIMAL,
   mangerName VARCHAR(255)
);

Machine (
   machineID INT PRIMARY KEY,
   machineName VARCHAR(255),
   departmentID INT,
   FOREIGN KEY (departmentID) REFERENCES Department(departmentID)
);

Department (
   departmentID INT PRIMARY KEY,
   departmentName VARCHAR(255),
   location VARCHAR(255),
   managerID INT,
   FOREIGN KEY (managerID) REFERENCES Manager(managerID)
);

MachineOperator (
   operatorID INT PRIMARY KEY,
   operatorName VARCHAR(255),
   machineID INT,
   FOREIGN KEY (machineID) REFERENCES Machine(machineID)
);
```

Weak Entity:

```
PartBelong (
   partNumber INT,
   quantity INT,
   color VARCHAR
   machineID INT,
   PRIMARY KEY (partNumber, machineID),
   FOREIGN KEY (machineID) REFERENCES Machine(machineID)
);
```

Recursive relationship:

```
MarriedTo (
   wifeID INT,
   husbandID INT,
   PRIMARY KEY (wifeID, husbandID),
   FOREIGN KEY (wifeID) REFERENCES Employee(employeeID),
   FOREIGN KEY (husbandID) REFERENCES Employee(employeeID)
);
```

9. Design necessary queries by the following requirements:

(a) Three queries for one table.

Query 1: Find the sum of all the managers' salary
Query 2: Find the average quantity of all parts
Query 3: Find the lowest salary of out of all managers

(b) Three queries for two tables.

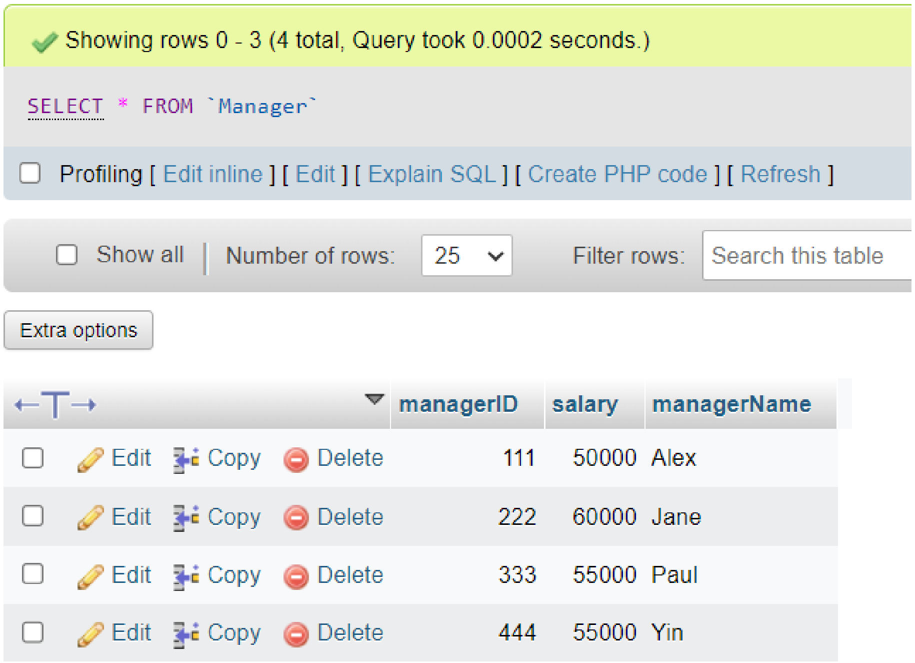
Query 1: Find the employee IDs and names of employees who work in a department located in 'Building A'.
Query 2: Find the names of machines who do not use a 'Red' color part
Query 3: Find the managerID and names of the managers who manage a department located in Building A and in Building B.

(c) Three queries for three tables.

Query 1: Find the number of all parts with their machines and machine operators
Query 2: Find the employee with ID 101 and Manager named Alex and the department location Building A
Query 3: Find the machine operator and machine who use the highest quantity of parts

10. Write necessary SQL commands to create tables and input data to each table.

```
CREATE TABLE Manager (
   managerID INT PRIMARY KEY,
   salary DECIMAL,
   managerName VARCHAR(255)
);
```



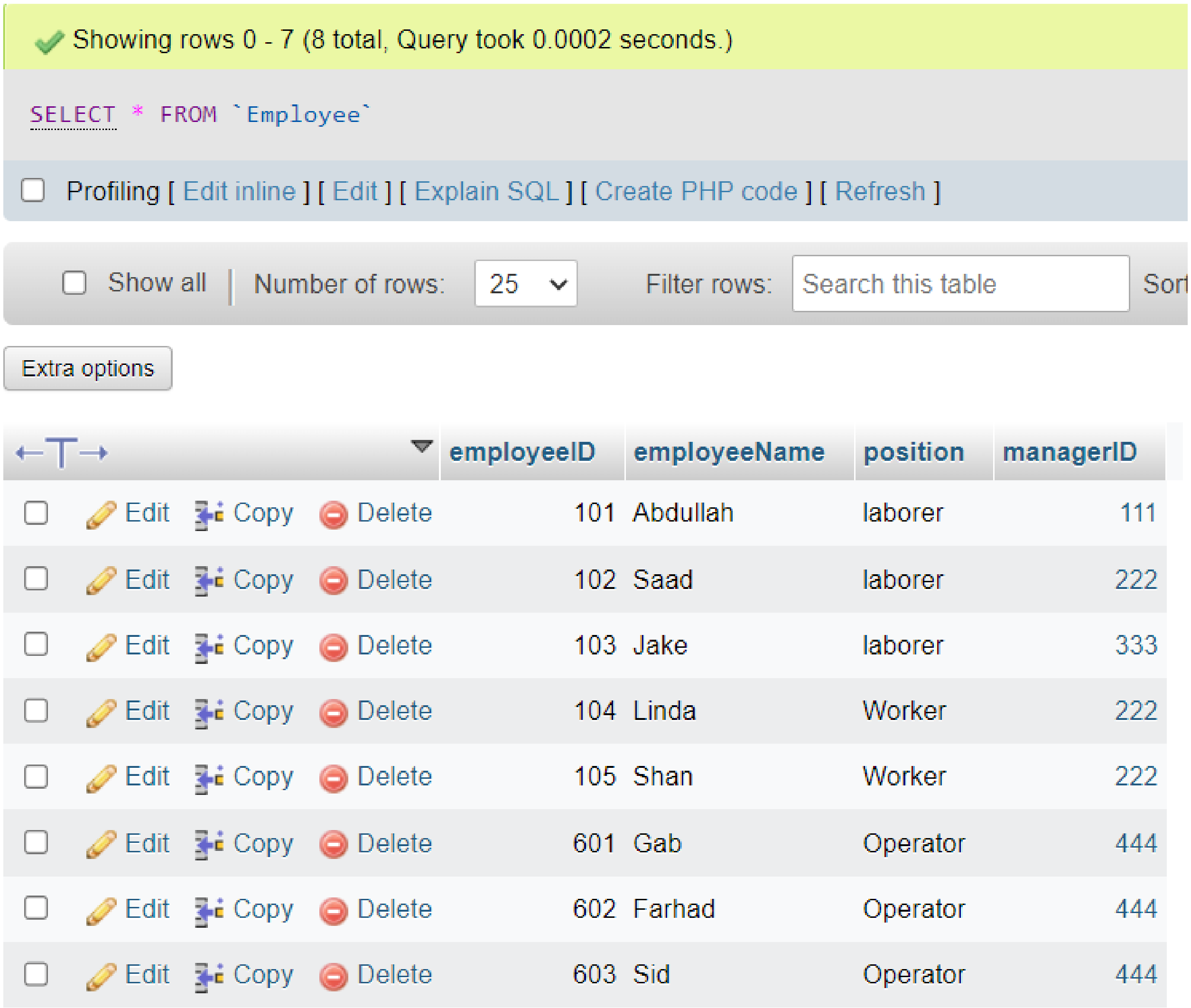
✓ Showing rows 0 - 3 (4 total, Query took 0.0002 seconds.)

```
SELECT * FROM `Manager`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

←T→				managerID	salary	managerName
□	Edit	Copy	Delete	111	50000	Alex
□	Edit	Copy	Delete	222	60000	Jane
□	Edit	Copy	Delete	333	55000	Paul
□	Edit	Copy	Delete	444	55000	Yin

```
CREATE TABLE Employee (
    employeeID INT PRIMARY KEY,
    employeeName VARCHAR(255),
    position VARCHAR(255),
    managerID INT,
    FOREIGN KEY (managerID) REFERENCES Manager(managerID));
```



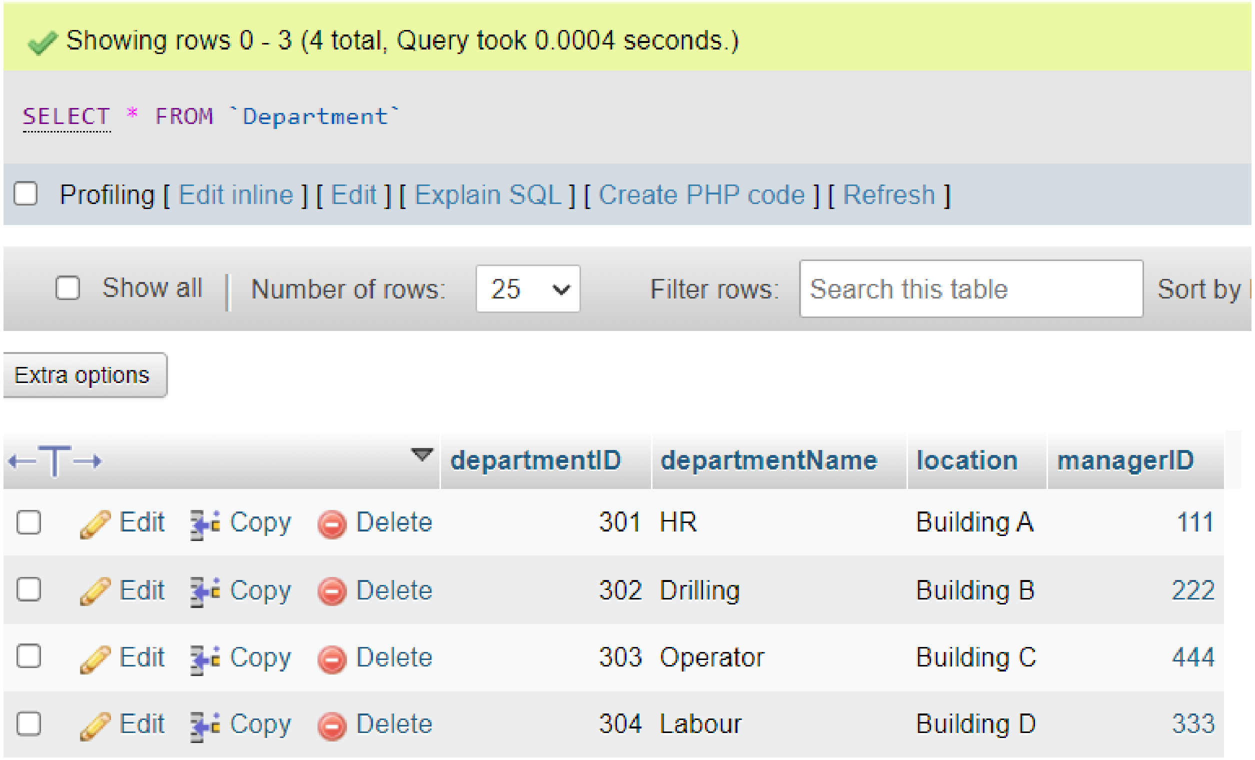
✓ Showing rows 0 - 7 (8 total, Query took 0.0002 seconds.)

```
SELECT * FROM `Employee`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Sort
Extra options

←T→	employeeID	employeeName	position	managerID
□ Edit Copy Delete	101	Abdullah	laborer	111
□ Edit Copy Delete	102	Saad	laborer	222
□ Edit Copy Delete	103	Jake	laborer	333
□ Edit Copy Delete	104	Linda	Worker	222
□ Edit Copy Delete	105	Shan	Worker	222
□ Edit Copy Delete	601	Gab	Operator	444
□ Edit Copy Delete	602	Farhad	Operator	444
□ Edit Copy Delete	603	Sid	Operator	444

```
CREATE TABLE Department (
   departmentID INT PRIMARY KEY,
   departmentName VARCHAR(255),
   location VARCHAR(255),
   managerID INT,
   FOREIGN KEY (managerID) REFERENCES Manager(managerID)
);
```



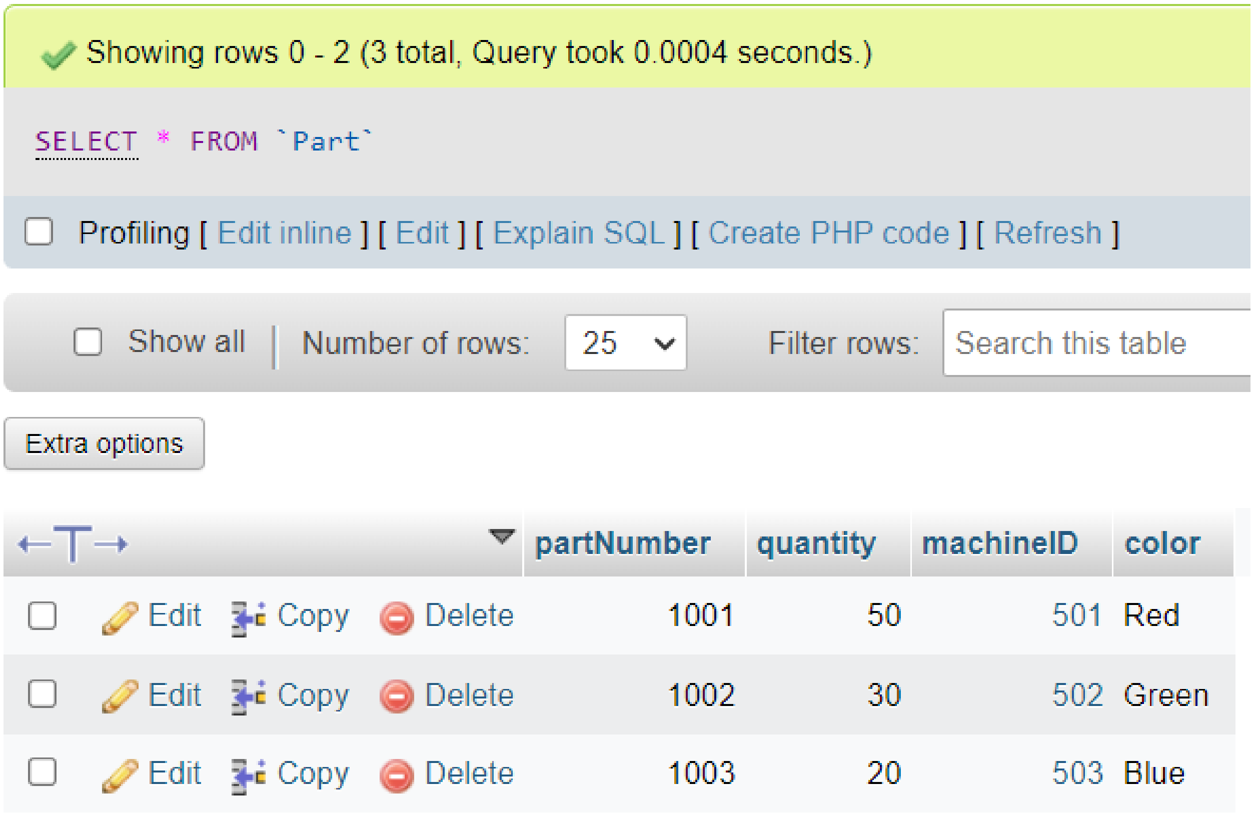
✓ Showing rows 0 - 3 (4 total, Query took 0.0004 seconds.)

```
SELECT * FROM `Department`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
☐ Show all
Number of rows: 25
Filter rows: Search this table
Sort by
Extra options
←T→

	departmentID	departmentName	location	managerID
☐ Edit Copy Delete	301	HR	Building A	111
☐ Edit Copy Delete	302	Drilling	Building B	222
☐ Edit Copy Delete	303	Operator	Building C	444
☐ Edit Copy Delete	304	Labour	Building D	333

```
CREATE TABLE Part (
    partNumber INT,
    quantity INT,
    machineID INT,
    color VARCHAR(255)
    PRIMARY KEY (partNumber, machineID),
    FOREIGN KEY (machineID) REFERENCES Machine(machineID)
);
```



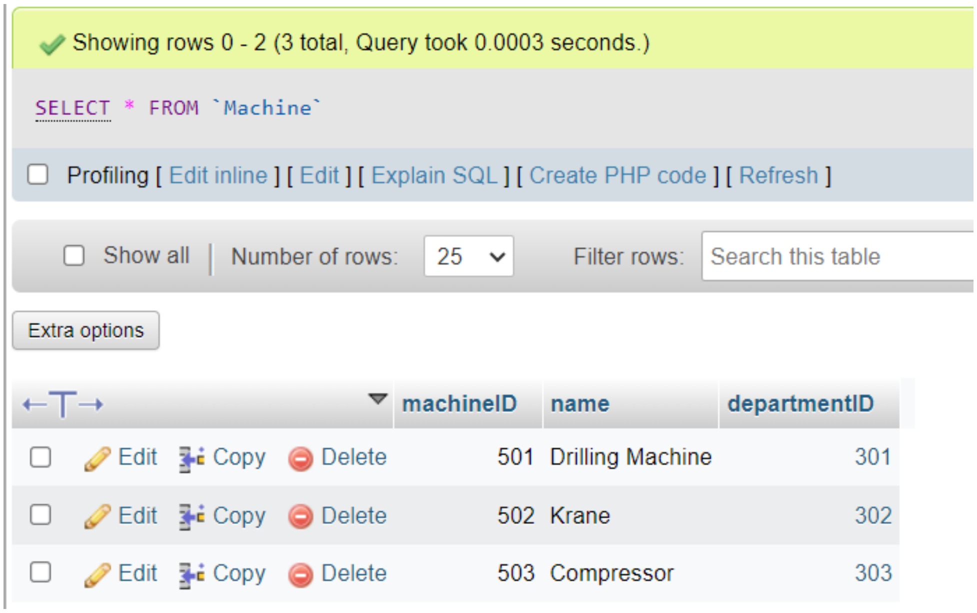
✓ Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `Part`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options
←T→

	partNumber	quantity	machineID	color
□ Edit Copy Delete	1001	50	501	Red
□ Edit Copy Delete	1002	30	502	Green
□ Edit Copy Delete	1003	20	503	Blue

```
CREATE TABLE Machine (
   machineID INT PRIMARY KEY,
   machineName VARCHAR(255),
   departmentID INT,
   FOREIGN KEY (departmentID) REFERENCES Department(departmentID)
);
```



✓ Showing rows 0 - 2 (3 total, Query took 0.0003 seconds.)

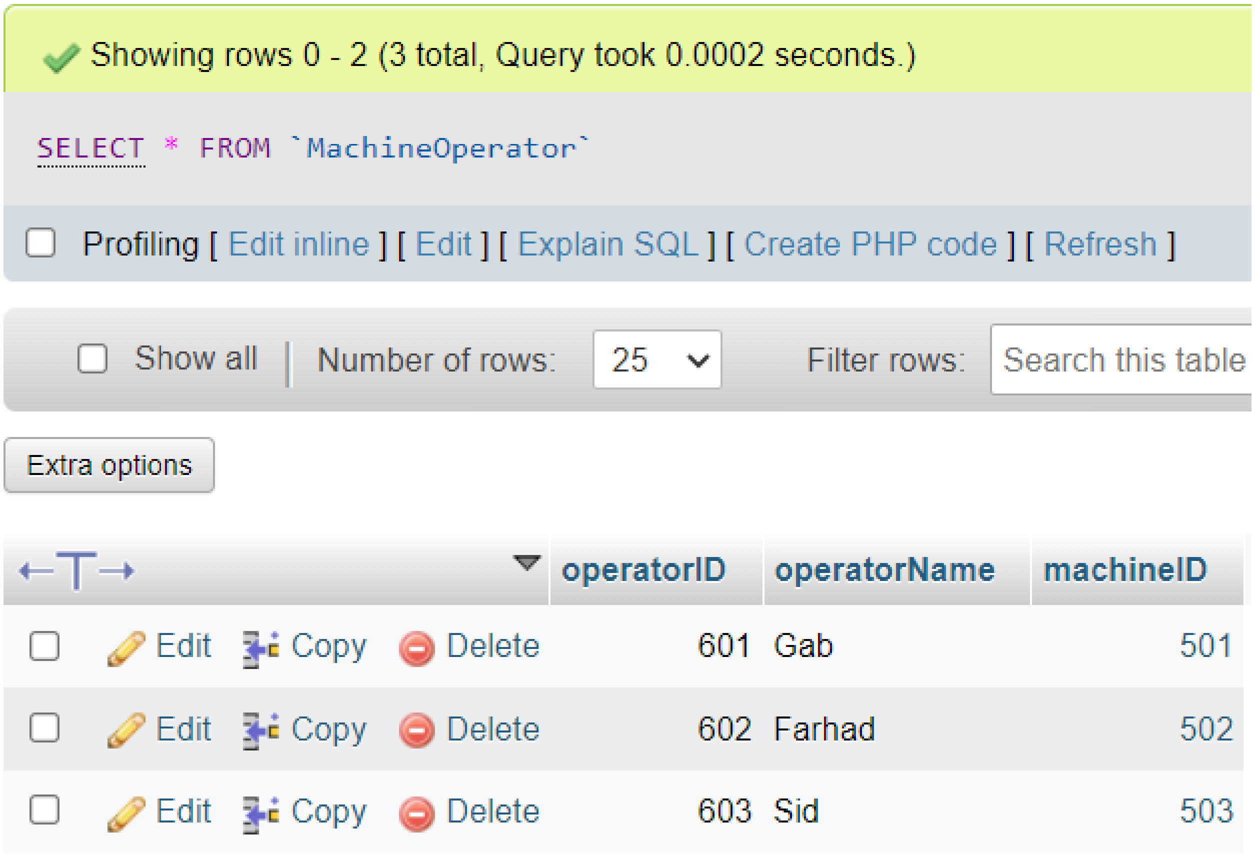
```
SELECT * FROM `Machine`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows:
Search this table
Extra options

Number of rows: 25

	machineID	name	departmentID
□ Edit Copy Delete	501	Drilling Machine	301
□ Edit Copy Delete	502	Krane	302
□ Edit Copy Delete	503	Compressor	303

```
CREATE TABLE MachineOperator (
    operatorID INT PRIMARY KEY,
    operatorName VARCHAR(255),
    machineID INT,
    FOREIGN KEY (machineID) REFERENCES Machine(machineID)
);
```



✓ Showing rows 0 - 2 (3 total, Query took 0.0002 seconds.)

```
SELECT * FROM `MachineOperator`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows:
Search this table
Extra options
←T→

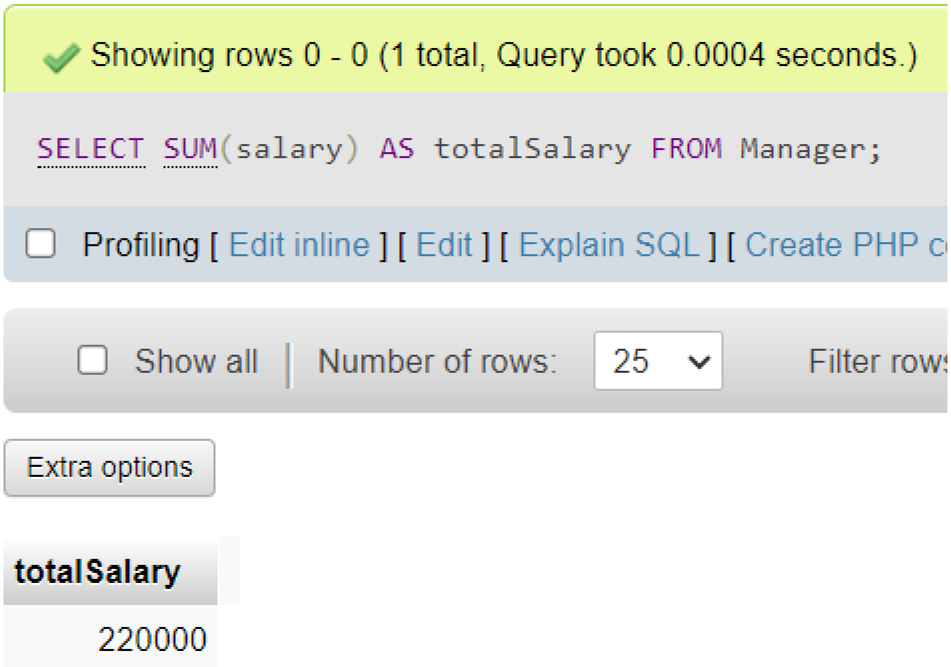
Number of rows: 25

	operatorID	operatorName	machineID
□ Edit Copy Delete	601	Gab	501
□ Edit Copy Delete	602	Farhad	502
□ Edit Copy Delete	603	Sid	503

11. Write the queries from step 9 using SQL statements. Show the output of your SQL queries using screen shot

Query 1: Find the sum of all the managers' salary

```
SELECT SUM(salary) AS totalSalary FROM Manager;
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

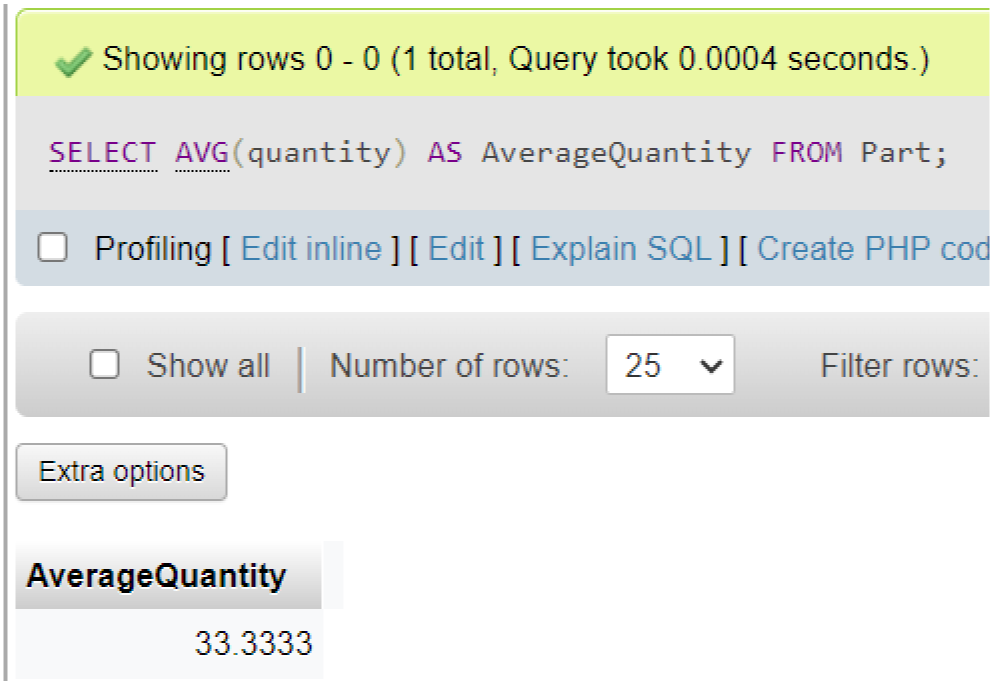
```
SELECT SUM(salary) AS totalSalary FROM Manager;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP c
□ Show all | Number of rows: 25
Filter rows
Extra options

totalSalary
220000

Query 2: Find the average quantity of all parts

```
SELECT AVG(quantity) AS AverageQuantity FROM Part;
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

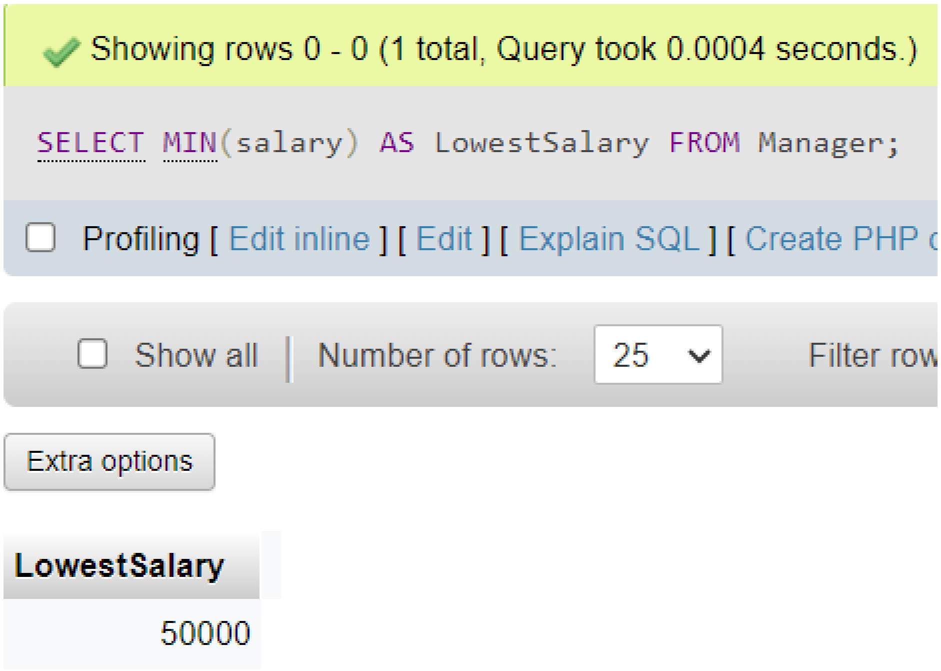
```
SELECT AVG(quantity) AS AverageQuantity FROM Part;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP cod
□ Show all | Number of rows: 25 Filter rows:
Extra options

AverageQuantity
33.3333

Query 3: Find the lowest salary of out of all managers

```
SELECT MIN(salary) AS LowestSalary FROM Manager;
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT MIN(salary) AS LowestSalary FROM Manager;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP c
□ Show all
Filter row
Extra options

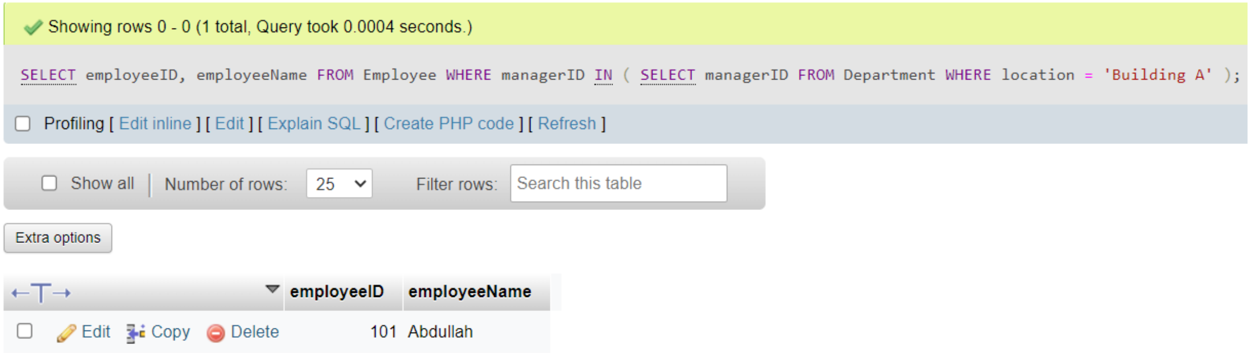
Number of rows: 25

LowestSalary
50000

(b) Three queries for two tables.

Query 1: Find the employee IDs and names of employees who work in a department located in 'Building A'.

```
SELECT employeeID, employeeName FROM Employee WHERE managerID IN (
SELECT managerID FROM Department WHERE location = 'Building A' );
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

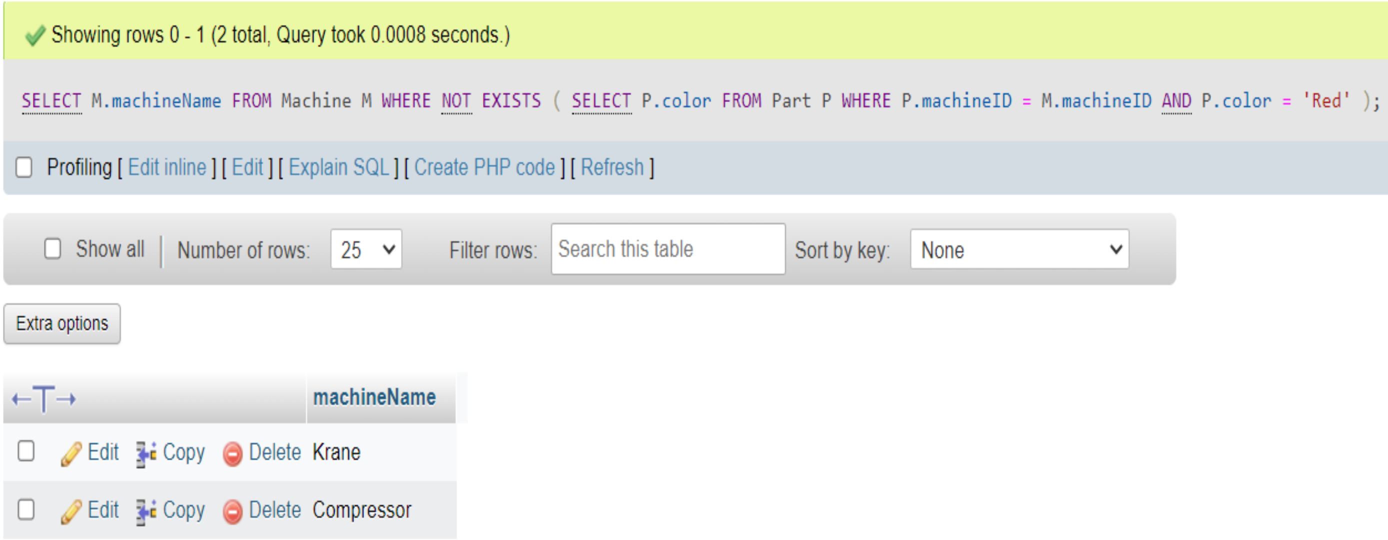
```
SELECT employeeID, employeeName FROM Employee WHERE managerID IN ( SELECT managerID FROM Department WHERE location = 'Building A' );
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options
←T→
□ Edit Copy Delete

employeeID	employeeName
101	Abdullah

Query 2: Find the names of machines who do not use a 'Red’ color part

```
SELECT M.machineName FROM Machine M WHERE NOT EXISTS ( SELECT P.color
FROM Part P WHERE P.machineID = M.machineID AND P.color = 'Red' );
```



✓ Showing rows 0 - 1 (2 total, Query took 0.0008 seconds.)

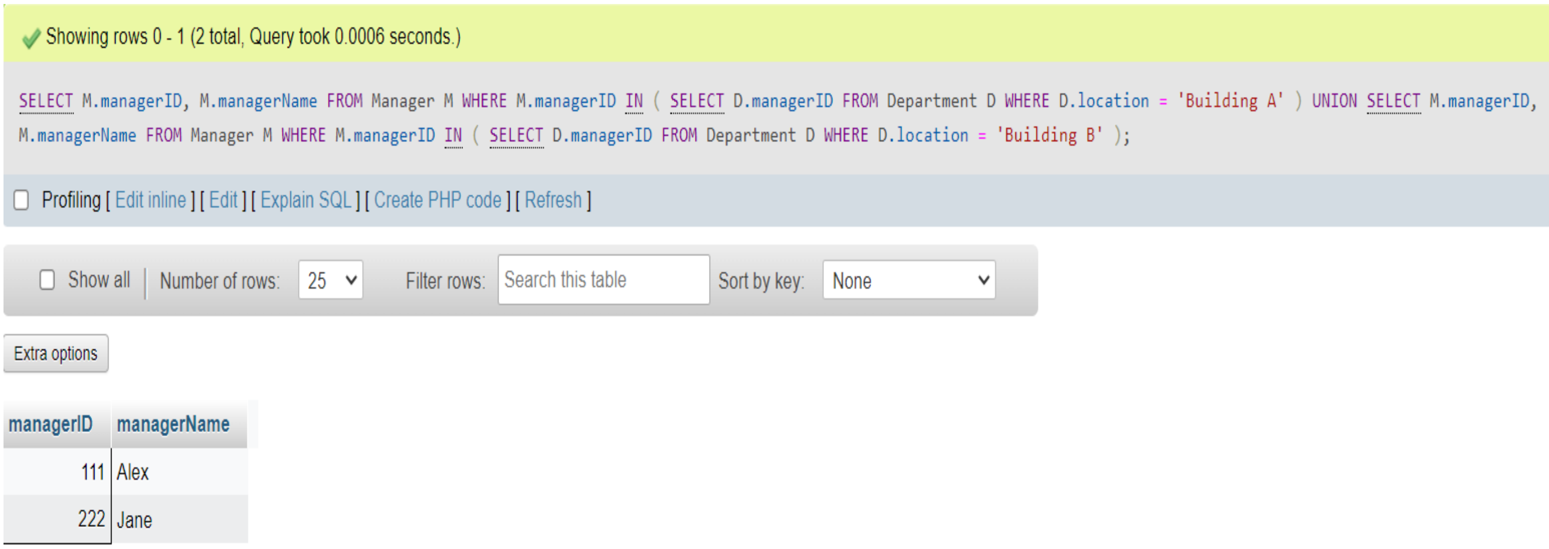
```
SELECT M.machineName FROM Machine M WHERE NOT EXISTS ( SELECT P.color FROM Part P WHERE P.machineID = M.machineID AND P.color = 'Red' );
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
Extra options
←T→
□ Edit Copy Delete
□ Edit Copy Delete

machineName
Krane
Compressor

Query 3: Find the managerID and names of the managers who manage a department located in Building A and in Building B.

```
SELECT M.managerID, M.managerName FROM Manager M WHERE M.managerID IN
( SELECT D.managerID FROM Department D WHERE D.location = 'Building A' )
UNION SELECT M.managerID, M.managerName FROM Manager M WHERE
M.managerID IN ( SELECT D.managerID FROM Department D WHERE D.location =
'Building B' );
```



✓ Showing rows 0 - 1 (2 total, Query took 0.0006 seconds.)

```
SELECT M.managerID, M.managerName FROM Manager M WHERE M.managerID IN ( SELECT D.managerID FROM Department D WHERE D.location = 'Building A' ) UNION SELECT M.managerID,
M.managerName FROM Manager M WHERE M.managerID IN ( SELECT D.managerID FROM Department D WHERE D.location = 'Building B' );
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Filter rows: Search this table
Extra options

Number of rows: 25

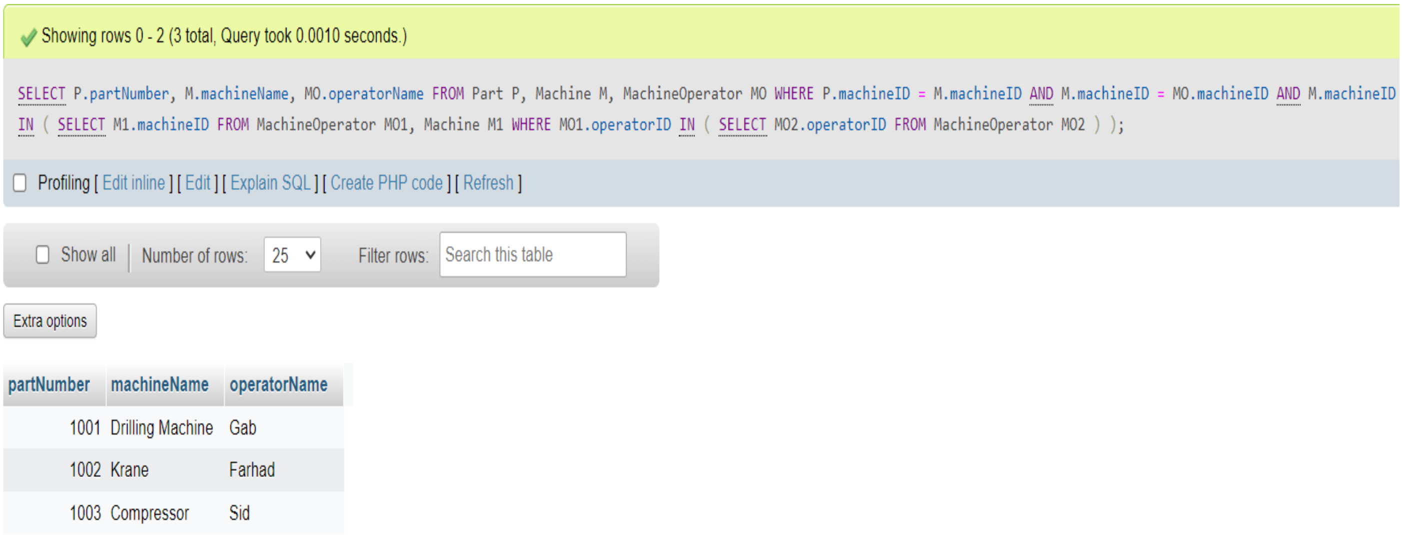
Sort by key: None

managerID	managerName
111	Alex
222	Jane

(c) Three queries for three tables.

Query 1: Find the number of all parts with their machines and machine operators

```
SELECT P.partNumber, M.machineName, MO.operatorName FROM Part P, Machine
M, MachineOperator MO WHERE P.machineID = M.machineID AND M.machineID =
MO.machineID AND M.machineID IN ( SELECT M1.machineID FROM
MachineOperator MO1, Machine M1 WHERE MO1.operatorID IN ( SELECT
MO2.operatorID FROM MachineOperator MO2 ) );
```



✓ Showing rows 0 - 2 (3 total, Query took 0.0010 seconds.)

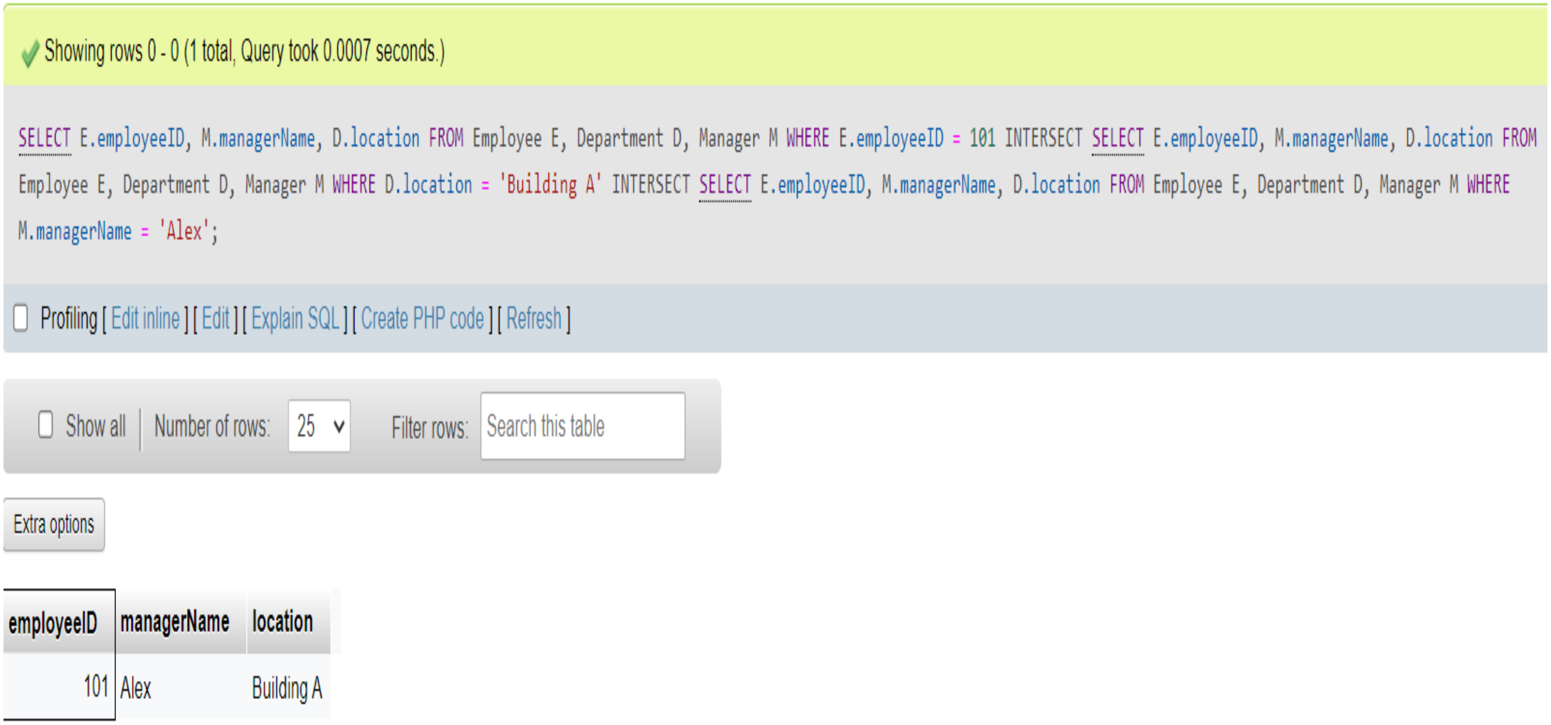
```
SELECT P.partNumber, M.machineName, MO.operatorName FROM Part P, Machine M, MachineOperator MO WHERE P.machineID = M.machineID AND M.machineID = MO.machineID AND M.machineID
IN ( SELECT M1.machineID FROM MachineOperator MO1, Machine M1 WHERE MO1.operatorID IN ( SELECT MO2.operatorID FROM MachineOperator MO2 ) );
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

partNumber	machineName	operatorName
1001	Drilling Machine	Gab
1002	Krane	Farhad
1003	Compressor	Sid

Query 2: Find the employee with ID 101 and Manager named Alex and the department location Building A

SELECT E.employeeID, M.managerName, D.location FROM Employee E, Department D, Manager M WHERE E.employeeID = 101 INTERSECT SELECT E.employeeID, M.managerName, D.location FROM Employee E, Department D, Manager M WHERE D.location = 'Building A' INTERSECT SELECT E.employeeID, M.managerName, D.location FROM Employee E, Department D, Manager M WHERE M.managerName = 'Alex';



✓ Showing rows 0 - 0 (1 total, Query took 0.0007 seconds.)

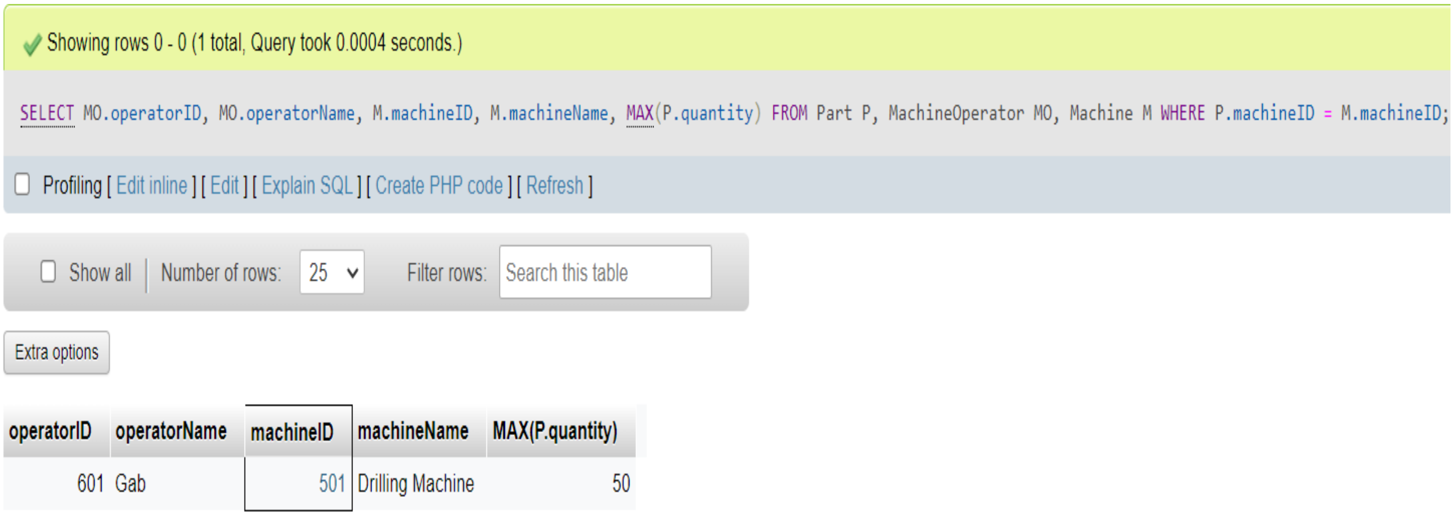
```
SELECT E.employeeID, M.managerName, D.location FROM Employee E, Department D, Manager M WHERE E.employeeID = 101 INTERSECT SELECT E.employeeID, M.managerName, D.location FROM
Employee E, Department D, Manager M WHERE D.location = 'Building A' INTERSECT SELECT E.employeeID, M.managerName, D.location FROM Employee E, Department D, Manager M WHERE
M.managerName = 'Alex';
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

employeeID	managerName	location
101	Alex	Building A

Query 3: find the machine operator and machine who use the highest quantity of parts

```
SELECT MO.operatorID, MO.operatorName, M.machineID, M.machineName,
MAX(P.quantity) FROM Part P, MachineOperator MO, Machine M WHERE P.machineID
= M.machineID;
```



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT MO.operatorID, MO.operatorName, M.machineID, M.machineName, MAX(P.quantity) FROM Part P, MachineOperator MO, Machine M WHERE P.machineID = M.machineID;
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all
Number of rows: 25
Filter rows: Search this table
Extra options

operatorID	operatorName	machineID	machineName	MAX(P.quantity)
601	Gab	501	Drilling Machine	50