Basic Level

Question:1

Retrieve the names of employees who earn more than the average salary of all employees.

Answer:

```
SELECT name
FROM Employee
WHERE salary > (
  SELECT AVG(salary)
  FROM Employee
);
```

Explanation

The subquery calculates the average salary, and the main query selects employees earning more than that average.

Question:2

Find the employees who belong to the department with the highest average salary.

Answer:

```
SELECT name

FROM Employee

WHERE department_id = (

  SELECT department_id

  FROM Employee

  GROUP BY department_id

  ORDER BY AVG(salary) DESC

  LIMIT 1

);
```

Explanation

First, the subquery finds the department with the highest average salary.
Then, the main query selects employees from that department.

Question:3

list all employees who have made at least one sale**.**

Answer:

```
SELECT DISTINCT e.name

FROM Employee e

INNER JOIN Sales s

ON e.emp_id = s.emp_id;
```

Explanation

INNER JOIN selects only those employees whose emp_id exists in the Sales table.

Question:4

Find the employee with the highest sale amount.

Answer:

```
SELECT e.name, s.sale_amount

FROM Employee e

JOIN Sales s

ON e.emp_id = s.emp_id

WHERE s.sale_amount = (
```

```
  SELECT MAX(sale_amount)
  FROM Sales
);
```

Explanation

The subquery finds the highest sale amount, and the main query returns the employee who made that sale.

Question:5

Retrieve the names of employees whose salaries are higher than Shubham’s salary.

Answer:

```
SELECT name
FROM Employee
WHERE salary > (
  SELECT salary
  FROM Employee
  WHERE name = 'Shubham'
);
```

Explanation

The subquery gets Shubham’s salary, and the main query selects employees earning more than that.

Intermediate Level

Question:1
Find employees who work in the same department as Abhishek.

Answer:

```
SELECT name
FROM Employee
WHERE department_id = (
  SELECT department_id
  FROM Employee
  WHERE name = 'Abhishek'
)
AND name <> 'Abhishek';
```

Explanation

The subquery finds Abhishek's department, and the main query selects other employees from the same department.

Question:2
List departments that have at least one employee earning more than ₹60,000.

Answer:

```
SELECT DISTINCT d.department_name
FROM Department d
JOIN Employee e
ON d.department_id = e.department_id
WHERE e.salary > 60000;
```

Explanation

This query joins Department and Employee tables and selects departments where at least one employee earns more than ₹60,000.

Question:3
Find the department name of the employee who made the highest sale.

Answer:

```
SELECT d.department_name
FROM Department d
JOIN Employee e
ON d.department_id = e.department_id
JOIN Sales s
ON e.emp_id = s.emp_id
WHERE s.sale_amount = (
  SELECT MAX(sale_amount)
  FROM Sales
);
```

Explanation

The subquery finds the highest sale amount, and the joins give the department of the employee who made that sale.

Question:4
Retrieve employees who have made sales greater than the average sale amount.

Answer:

```
SELECT DISTINCT e.name
FROM Employee e
JOIN Sales s
ON e.emp_id = s.emp_id
WHERE s.sale_amount > (
  SELECT AVG(sale_amount)
  FROM Sales
);
```

Explanation

The subquery calculates the average sale amount, and the main query selects employees whose sales are above that average.

Question:5
Find the total sales made by employees who earn more than the average salary.

Answer:

```
SELECT SUM(s.sale_amount) AS total_sales
FROM Sales s
JOIN Employee e
ON s.emp_id = e.emp_id
WHERE e.salary > (
  SELECT AVG(salary)
  FROM Employee
);
```

Explanation

First, the subquery finds the average salary.
Then, sales are summed only for employees earning more than that average.

Advanced Level

Question: 1
Find employees who have not made any sales.

Answer:
```
SELECT e.name
FROM Employee e
LEFT JOIN Sales s
ON e.emp_id = s.emp_id
WHERE s.emp_id IS NULL;
```

Explanation

LEFT JOIN keeps all employees.
Employees with no matching record in Sales are those who have not made any sales.

Question: 2
List departments where the average salary is above ₹55,000.

Answer:
```
SELECT d.department_name
FROM Department d
JOIN Employee e
ON d.department_id = e.department_id
GROUP BY d.department_name
HAVING AVG(e.salary) > 55000;
```

Explanation

Salaries are grouped by department, and HAVING filters departments with average salary above ₹55,000.

Question:3

Retrieve department names where the total sales exceed ₹10,000.

Answer:

```
SELECT d.department_name

FROM Department d

JOIN Employee e

ON d.department_id = e.department_id

JOIN Sales s

ON e.emp_id = s.emp_id

GROUP BY d.department_name

HAVING SUM(s.sale_amount) > 10000;
```

Explanation

Sales are grouped by department, and only those departments with total sales above ₹10,000 are selected.

Question: 4

Find the employee who has made the second-highest sale.

Answer:

```
SELECT e.name, s.sale_amount

FROM Employee e

JOIN Sales s

ON e.emp_id = s.emp_id
```

```
WHERE s.sale_amount = (
  SELECT MAX(sale_amount)
  FROM Sales
  WHERE sale_amount < (
    SELECT MAX(sale_amount)
    FROM Sales
  )
);
```

Explanation

First subquery finds the highest sale.
Second subquery finds the next highest sale below it, and the employee is selected.

Question: 5

Retrieve the names of employees whose salary is greater than the highest sale amount recorded.

Answer:

```
SELECT name
FROM Employee
WHERE salary > (
  SELECT MAX(sale_amount)
  FROM Sales
);
```

Explanation

The subquery gets the highest sale amount, and the main query selects employees whose salary is higher than that value.