1) Consider the following tables:

```
EMP (
    EmployeeID INT PRIMARY KEY,

    Name VARCHAR(100),

    Salary DECIMAL(15, 2),

    DepartmentNumber INT,

    ManagerID INT,

    JobTitle VARCHAR(100),

    FOREIGN KEY (DepartmentNumber) REFERENCES DEPARTMENT(DepartmentID),

    FOREIGN KEY (ManagerID) REFERENCES EMP(EmployeeID)
);
```

```
DEPARTMENT (
DepartmentID INT PRIMARY KEY,

DepartmentName VARCHAR(100),

Location VARCHAR(100)
);
```

```
JOB (

JobTitle VARCHAR(100)

PRIMARY KEY );
```

Write a SQL query to

i) Retrieve the names and salaries of all employees from the `EMP` table.
ii) Retrieve the employee IDs and names of all employees whose salary is greater than 50,000.
iii) Find the average salary of employees in the `EMP` table.
iv) Retrieve the department numbers and the count of employees in each department from the `EMP` table.
v) Retrieve the names of employees along with the names of their respective managers. Assume the `EMP` table has a `ManagerID` column that references the `EmployeeID`.
vi) Retrieve the employee IDs of employees whose salary is greater than the average salary of the company.
vii) Increase the salary of all employees in department 10 by 10%.
viii) Delete all employees from the `EMP` table who are in department 5 and have a salary less than 40,000.
ix) Create a table `DEPARTMENT` with columns `DepartmentID`, `DepartmentName`, and `Location`.
x) nsert a new employee with the following details into the `EMP` table: EmployeeID = 101, Name = 'John Doe', Salary = 60000, DepartmentNumber = 1, ManagerID = 100.
xi) Retrieve the employee IDs and a new column `SalaryCategory` which classifies salaries

into 'High', 'Medium', and 'Low'. Use thresholds of 70,000 for High, 50,000 for Medium, and below 50,000 for Low.

xii) Retrieve the employee IDs and names of employees who have the highest salary in their respective departments.

xiii) Retrieve distinct department numbers from the `EMP` table.

xiv) Retrieve all distinct job titles from the `EMP` table and another table `JOB` which has a `JobTitle` column.

xv) Retrieve the employee IDs of employees who work in a department with at least one other employee.

**Solutions:**

i) SELECT Name, Salary FROM EMP;

ii) SELECT EmployeeID, Name FROM EMP WHERE Salary > 50000;

iii) SELECT AVG(Salary) AS AverageSalary FROM EMP;

iv) SELECT DepartmentNumber, COUNT(*) AS EmployeeCount FROM EMP GROUP BY DepartmentNumber;

v) SELECT E1.Name AS EmployeeName, E2.Name AS ManagerName FROM EMP E1 JOIN EMP E2 ON E1.ManagerID = E2.EmployeeID;

vi) SELECT EmployeeID FROM EMP WHERE Salary > (SELECT AVG(Salary) FROM EMP);

vii) UPDATE EMP SET Salary = Salary * 1.10 WHERE DepartmentNumber = 10;

viii) DELETE FROM EMP WHERE DepartmentNumber = 5 AND Salary < 40000;

ix) CREATE TABLE DEPARTMENT ( DepartmentID INT PRIMARY KEY, DepartmentName VARCHAR(100), Location VARCHAR(100) );

x) INSERT INTO EMP (EmployeeID, Name, Salary, DepartmentNumber, ManagerID) VALUES (101, 'John Doe', 60000, 1, 100);

xi) SELECT EmployeeID, CASE WHEN Salary >= 70000 THEN 'High' WHEN Salary >= 50000 THEN 'Medium' ELSE 'Low' END AS SalaryCategory FROM EMP;

xii) SELECT EmployeeID, Name FROM EMP WHERE Salary = (SELECT MAX(Salary) FROM EMP E2 WHERE E2.DepartmentNumber = EMP.DepartmentNumber);

xiii) SELECT DISTINCT DepartmentNumber FROM EMP;

xiv) SELECT JobTitle FROM EMP UNION SELECT JobTitle FROM JOB;

xv) SELECT E1.EmployeeID FROM EMP E1 WHERE EXISTS ( SELECT 1 FROM EMP E2 WHERE E2.DepartmentNumber = E1.DepartmentNumber AND E2.EmployeeID != E1.EmployeeID );

2) Write a query to retrieve the names and department names of all employees.

**EMP Table:**

| EmployeeID | Name | DepartmentNumber |
|---|---|---|
| 101 | John | 1 |
| 102 | Jane | 2 |
| 103 | Alice | 1 |

**DEPARTMENT Table:**

| DepartmentID | DepartmentName |
|---|---|
| 1 | IT |
| 2 | Finance |

**Solution:**

```
SELECT E.Name, D.DepartmentName
FROM EMP E
JOIN DEPARTMENT D ON E.DepartmentNumber = D.DepartmentID;
```

3) Given the following data, write a query to find the output.

**Solution:**

```
SELECT E.Name, D.DepartmentName
FROM EMP E
JOIN DEPARTMENT D ON E.DepartmentNumber = D.DepartmentID;
```

4) Write a query to retrieve the Cartesian product of the `EMP` and `DEPARTMENT` tables.

**EMP Table:**

| EmployeeID | Name | DepartmentNumber |
|---|---|---|
| 101 | John | 1 |
| 102 | Jane | 2 |
| 103 | Alice | 1 |

**DEPARTMENT Table:**

| DepartmentID | DepartmentName |
|---|---|
| 1 | IT |
| 2 | Finance |

**Solution:**

```
SELECT E.Name, D.DepartmentName
FROM EMP E, DEPARTMENT D;
```

**Output:** This query will produce a Cartesian product of all combinations of employee names and department names.

5) Write a query to retrieve the names of employees who belong to department 1 or department 2.

```
SELECT Name
FROM EMP
WHERE DepartmentNumber IN (1, 2);
```

6) Write a query to retrieve the names of employees whose salary is greater than any employee in department 1.

**EMP Table:**

| EmployeeID | Name | Salary | DepartmentNumber |
|---|---|---|---|
| 101 | John | 60000 | 1 |

| | | | |
|---|---|---|---|
| 102 | Jane | 55000 | 2 |
| 103 | Alice | 50000 | 1 |

Solution:

```
SELECT Name
FROM EMP E1
WHERE Salary > ANY (
    SELECT Salary
    FROM EMP E2
    WHERE E2.DepartmentNumber = 1
);
```

**Output:**

| Name |
|---|
| John |
| Jane |

7) Write a query to retrieve the names of employees whose name starts with 'J'.

sql
Copy code

```
SELECT Name
FROM EMP
WHERE Name LIKE 'J%';
```

8) Consider the following tables:

**Authors Table**

| author_id | name | birth_year |
|---|---|---|
| 1 | J.K. Rowling | 1965 |
| 2 | George Orwell | 1903 |
| 3 | J.R.R. Tolkien | 1892 |

**Books Table**

| book_id | title | author_id | published_year | price |
|---|---|---|---|---|
| 1 | Harry Potter | 1 | 1997 | 20.99 |
| 2 | 1984 | 2 | 1949 | 15.99 |
| 3 | The Hobbit | 3 | 1937 | 25.99 |
| 4 | Animal Farm | 2 | 1945 | 12.99 |
| 5 | The Lord of the Rings | 3 | 1954 | 35.99 |

**Sales Table**

| sale_id | book_id | quantity | sale_date |
|---|---|---|---|
| 1 | 1 | 5 | 2023-01-15 |
| 2 | 3 | 2 | 2023-02-20 |
| 3 | 1 | 3 | 2023-03-01 |
| 4 | 2 | 4 | 2023-03-15 |
| 5 | 4 | 6 | 2023-03-30 |
| 6 | 5 | 1 | 2023-04-05 |

i) SELECT a.name, SUM(s.quantity * b.price) AS total_sales
FROM Authors a
JOIN Books b ON a.author_id = b.author_id
JOIN Sales s ON b.book_id = s.book_id
GROUP BY a.name
ORDER BY total_sales DESC;

Question: What is the output of the above query?

A) J.K. Rowling - $167.92, George Orwell - $128.90, J.R.R. Tolkien - $87.97
B) J.K. Rowling - $167.92, George Orwell - $103.92, J.R.R. Tolkien - $103.96
C) J.K. Rowling - $125.94, George Orwell - $103.92, J.R.R. Tolkien - $103.96
D) J.K. Rowling - $125.94, George Orwell - $128.90, J.R.R. Tolkien - $87.97

**Solution:**
To solve this, we need to calculate the total sales for each author:

J.K. Rowling:

Harry Potter: (5 * 20.99) + (3 * 20.99) = 104.95 + 62.97 = $167.92
George Orwell:

1984: (4 * 15.99) = $63.96
Animal Farm: (6 * 12.99) = $77.94
Total = 63.96 + 77.94 = $141.90
J.R.R. Tolkien:

The Hobbit: (2 * 25.99) = $51.98
The Lord of the Rings: (1 * 35.99) = $35.99
Total = 51.98 + 35.99 = $87.97

Correct Answer: A) J.K. Rowling - $167.92, George Orwell - $141.90, J.R.R. Tolkien - $87.97

ii) SELECT b.title, SUM(s.quantity) AS total_quantity
FROM Books b
JOIN Sales s ON b.book_id = s.book_id
GROUP BY b.title
ORDER BY total_quantity DESC
LIMIT 1;
Question: What is the output of the above query?

A) Harry Potter - 8
B) 1984 - 4
C) Animal Farm - 6
D) The Hobbit - 2

**Solution:**
To find the most sold book:

Harry Potter: 5 + 3 = 8
1984: 4
Animal Farm: 6
The Hobbit: 2
The Lord of the Rings: 1
Correct Answer: A) Harry Potter - 8

iii) SELECT a.name
FROM Authors a
LEFT JOIN Books b ON a.author_id = b.author_id
LEFT JOIN Sales s ON b.book_id = s.book_id
WHERE s.sale_id IS NULL;

Question: What is the output of the above query?

A) J.K. Rowling
B) George Orwell
C) J.R.R. Tolkien
D) No authors

**Solution**:
To find authors with no sales:

All authors have sales because there are no NULL entries in the Sales table for the given Books.
J.K. Rowling has sales for "Harry Potter".
George Orwell has sales for "1984" and "Animal Farm".
J.R.R. Tolkien has sales for "The Hobbit" and "The Lord of the Rings".
Correct Answer: D) No authors

iv)
```
SELECT b.title
FROM Books b
LEFT JOIN Sales s ON b.book_id = s.book_id
WHERE s.sale_id IS NULL;
```

**Question:** What is the output of the above query?

1. A) Harry Potter, 1984, The Hobbit, Animal Farm, The Lord of the Rings
2. B) Harry Potter
3. C) No books
4. D) The Lord of the Rings

**Solution:** To find the books that have not been sold:

- Check each book in the Sales table to see if there is a corresponding sale:
    - All books have corresponding sales in the Sales table except "The Lord of the Rings".

Correct Answer: **D) The Lord of the Rings**

v)
```
SELECT a.name
FROM Authors a
JOIN Books b ON a.author_id = b.author_id
```

```
GROUP BY a.name
HAVING COUNT(b.book_id) > 1;
```

**Question:** What is the output of the above query?

1. A) J.K. Rowling
2. B) George Orwell
3. C) J.R.R. Tolkien
4. D) George Orwell and J.R.R. Tolkien

**Solution:** To find authors with more than one book:

- J.K. Rowling: 1 book
- George Orwell: 2 books (1984, Animal Farm)
- J.R.R. Tolkien: 2 books (The Hobbit, The Lord of the Rings)

Correct Answer: **D) George Orwell and J.R.R. Tolkien**

vi)
```
SELECT b.title
FROM Books b
JOIN Sales s ON b.book_id = s.book_id
WHERE b.published_year > 1950
GROUP BY b.title;
```

**Question:** What is the output of the above query?

1. A) Harry Potter, The Lord of the Rings
2. B) Harry Potter
3. C) 1984
4. D) The Hobbit

**Solution:** To find books published after 1950 that have sales:

- Harry Potter: published in 1997, has sales
- 1984: published in 1949
- Animal Farm: published in 1945
- The Hobbit: published in 1937
- The Lord of the Rings: published in 1954, has sales

Correct Answer: **A) Harry Potter, The Lord of the Rings**

vii)
```
SELECT b.title, SUM(s.quantity * b.price) AS total_sales
FROM Books b
JOIN Sales s ON b.book_id = s.book_id
WHERE b.published_year < 1950
```

```
GROUP BY b.title;
```

**Question:** What is the output of the above query?

1. A) 1984 - $63.96, Animal Farm - $77.94, The Hobbit - $51.98
2. B) 1984 - $63.96, Animal Farm - $77.94
3. C) 1984 - $63.96, The Hobbit - $51.98
4. D) Animal Farm - $77.94, The Hobbit - $51.98

**Solution:** To calculate total sales for books published before 1950:

- 1984: (4 * 15.99) = $63.96
- Animal Farm: (6 * 12.99) = $77.94
- The Hobbit: (2 * 25.99) = $51.98

Correct Answer: **A) 1984 - $63.96, Animal Farm - $77.94, The Hobbit - $51.98**

viii)
```
SELECT b.title AS name
FROM Books b
WHERE b.published_year > 1950
UNION
SELECT a.name
FROM Authors a
WHERE a.birth_year > 1950;
```

**Question:** What is the output of the above query?

1. A) Harry Potter, J.K. Rowling, The Lord of the Rings
2. B) Harry Potter, The Lord of the Rings
3. C) J.K. Rowling, George Orwell
4. D) Harry Potter, The Lord of the Rings, J.K. Rowling

**Solution:** To find the union of book titles published after 1950 and author names born after 1950:

- Books published after 1950: Harry Potter, The Lord of the Rings
- Authors born after 1950: J.K. Rowling

Correct Answer: **D) Harry Potter, The Lord of the Rings, J.K. Rowling**

ix)
```
SELECT b.title AS name
FROM Books b
WHERE b.published_year < 1950
INTERSECT
```

```
SELECT a.name
FROM Authors a
WHERE a.birth_year < 1950;
```

**Question:** What is the output of the above query?

1. A) 1984
2. B) George Orwell
3. C) No results
4. D) Animal Farm

**Solution:** To find the intersect of book titles published before 1950 and author names born before 1950:

- Books published before 1950: 1984, Animal Farm, The Hobbit
- Authors born before 1950: George Orwell, J.R.R. Tolkien
- There is no direct match between the titles of the books and the names of the authors.

Correct Answer: **C) No results**