ASSIGNMENT – 01

DATABASE MANAGEMENT SYSTEM

AIM: To run queries on the table where it defines a relation as “Employees are working for Departments and also for Colleges”.

DESCRIPTION:

In a Database Management System (DBMS), when modelling data related to employees working for departments and also working for a college, we can create a structure that reflects the relationships between employees, departments, and the college itself.

Here is a basic description of how this might be structured:

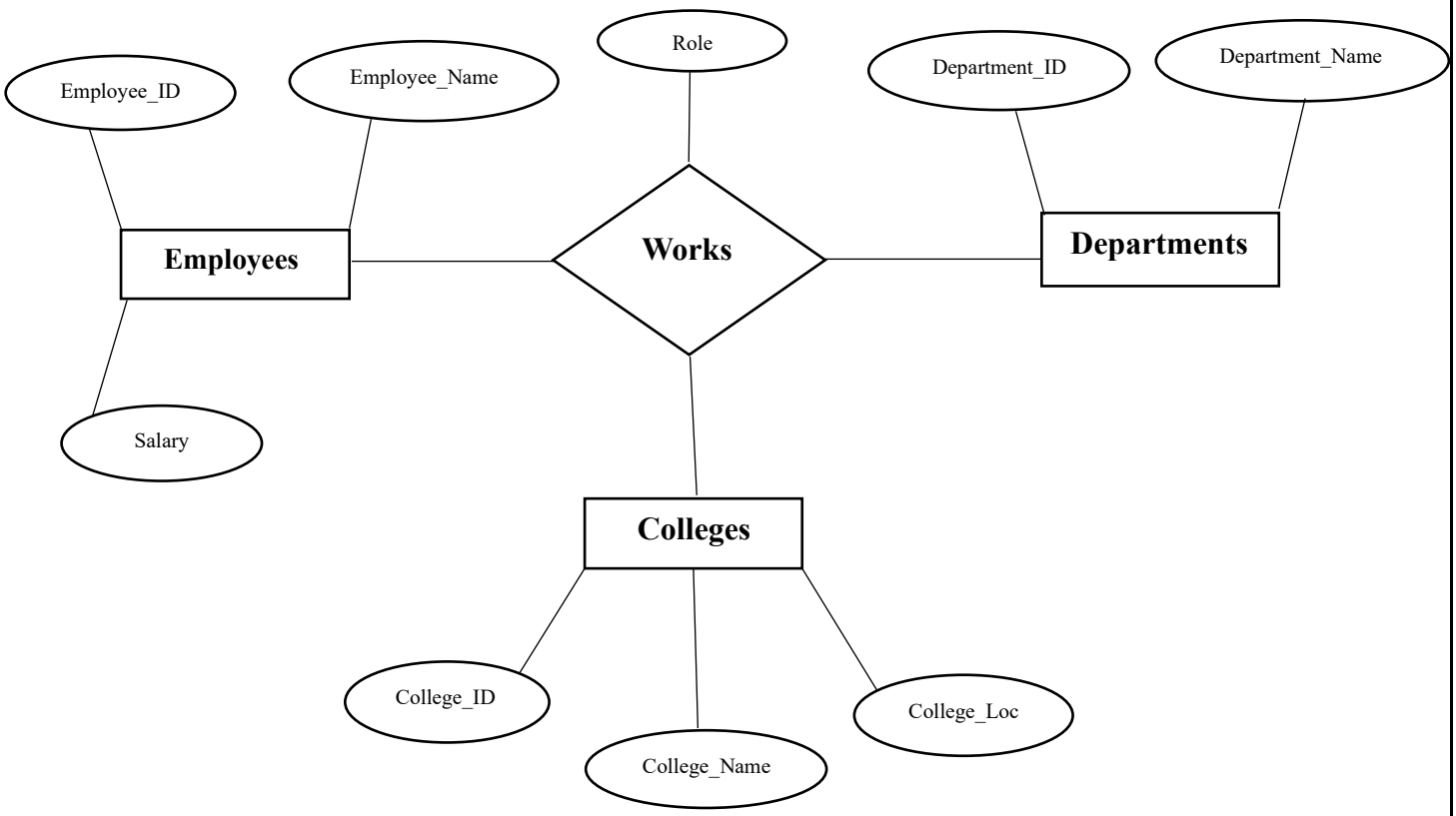
Entities and Their Attributes:

1. Employee
    - Employee ID (Primary Key): Unique identifier for each employee.
    - FirstName: The first name of the employee.
    - LastName: The last name of the employee.
    - Email: The email address of the employee.
    - Phone Number: The phone number of the employee.
    - Hire Date: The date the employee was hired.
    - Job Title: The title or role of the employee within the organization.
    - Department ID (Foreign Key): Links to the department the employee works for.
2. Department
    - Department ID (Primary Key): Unique identifier for each department.
    - Department Name: The name of the department (e.g., Computer Science, Mathematics, etc.).
    - Location: The location where the department operates (e.g., building name, floor number).
    - College ID (Foreign Key): Links to the college where the department is located.
3. College
    - College ID (Primary Key): Unique identifier for each college.
    - College Name: The name of the college (e.g., XYZ University).
    - Address: The address of the college.
    - Phone Number: The contact phone number for the college.

Relationships Between Entities:

1. Employee-Department Relationship:
    - An employee works for a department. This is a many-to-one relationship, meaning many employees can work for the same department.
    - Each employee can only belong to one department (unless they are working part-time in multiple departments, but for simplicity, we assume one department per employee).
    - The Department ID attribute in the Employee table serves as a foreign key, linking an employee to the department they work in.
2. Department-College Relationship:
    - A department belongs to one college. This is a many-to-one relationship because a college can have many departments, but each department belongs to only one college.
    - The College ID attribute in the Department table serves as a foreign key, linking the department to its college.

ER DIAGRAM:



SOURCE CODE:

Employee table

```
CREATE TABLE EMPLOYEE (
                                    EMPLOYEE_ID VARCHAR2(20) PRIMARY KEY,
                                    EMPLOYEE_NAME VARCHAR (100)
                                  );
```

Department table

```
CREATE TABLE DEPARTMENT (
                                     DEPARTMENT_ID VARCHAR2(20) PRIMARY KEY,
                                     DEPARTMENT_NAME VARCHAR (100)
                                    );
```

College table

```
CREATE TABLE COLLEGE (
                                   COLLEGE_ID VARCHAR2(20) PRIMARY KEY,
                                   COLLEGE_NAME VARCHAR (100)
                                 );
```

Employee_Department relationship table (many-to-many)

```
CREATE TABLE EMPLOYEE_DEPATMENT (
                EMPLOYEE_ID VARCHAR2(20),
                DEPARTMENT_ID VARCHAR2(20),
                PRIMARY KEY (EMPLOYEE_ID, DEPARTMENT_ID),
                FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE(EMPLOYEE_ID),
                FOREIGN KEY (DEPARTMENT_ID) REFERENCES DEPARTMENT(DEPARTMENT_ID)
            );
```

Employee_College relationship table (many-to-many)

```
CREATE TABLE EMPLOYEE_COLLEGE (
                 EMPLOYEE_ID VARCHAR2(20),
                 COLLEGE_ID VARCHAR2(20),
                 PRIMARY KEY (EMPLOYEE_ID, COLLEGE_ID),
```

```
    FOREIGN KEY (EMPLOYEE_ID) REFERENCES EMPLOYEE (EMPLOYEE_ID),
    FOREIGN KEY (COLLEGE_ID) REFERENCES COLLEGE (COLLEGE_ID)
);
```

Sample data insertion:

```
INSERT INTO EMPLOYEE (EMPLOYEE_ID, EMPLOYEE_NAME) VALUES (1, 'ALICE'), (2, 'BOB'), (3, 'CHARLIE');

INSERT INTO DEPARTMENT (DEPARTMENT_ID, DEPARTMENT_NAME) VALUES (101, 'HR'), (102, 'IT');

INSERT INTO COLLEGE (COLLEGE_ID, COLLEGE_NAME) VALUES (201, 'ENGINEERING COLLEGE'), (202, 'BUSINESS COLLEGE');

INSERT INTO EMPLOYEE_DEPARTMENT (EMPLOYEE_ID, DEPARTMENT_NO) VALUES (1, 101), (2, 102), (3, 101);

INSERT INTO EMPLOYEE_COLLEGE (EMPLOYEE_ID, DEPATMENT_ID) VALUES (1, 201), (2, 202), (3, 201);
```

Sample queries:

1. List all employees and their departments

```
SELECT E. EMPLOYEE_NAME, D. DEPARTMENT_NAME
FROM EMPLOYEE E
JOIN EMPLOYEE_DEPARTMENT ED ON E. EMPLOYEE_ID= ED. EMPLOYEE_ID
JOIN Department d ON ED. DEPARTMENT_ID= D. DEPARTMENT_ID;
```

2. List all employees and their colleges

```
SELECT E. EMPLOYEE_NAME, C. COLLEGE_NAME
FROM EMPLOYEE E
JOIN EMPLOYEE_COLLEGE EC ON E. EMPLOYEE_ID= EC. EMPLOYEE_ID
JOIN COLLEGE C ON EC. COLLEGE_ID=C.COLLEGE_ID;
```

3. Find all departments an employee works for (by employee name)

```
SELECT D. DEPARTMENT_NAME
FROM DEPARTMENT D
JOIN EMPLOYEE_DEPARTMENT ED ON D. DEPARTMENT_ID= ED. DEPARTMENT_ID
JOIN   EMPLOYEE E ON ED. EMPLOYEE_ID= E. EMPLOYEE_ID;
WHERE E. EMPLOYEE_NAME= 'ALICE';
```

4. Find all colleges an employee works for (by employee name)

```
SELECT C. COLLEGE_NAME
FROM COLLEGE C
JOIN EMPLOYEE_COLLEGE EC ON E. COLLEGE_ID=EC.COLLEGE_ID
JOIN EMPLOYEE E ON EC. EMPLOYEE_ID= E. EMPLOYEE_ID;
WHERE E. EMPLOYEE_NAME = 'BOB ';
```

5. Count the number of employees in each department

```
SELECT D. DEPARTMENT_NAME, COUNT (ED. EMPLOYEE_ID) AS EMPLOYEE_COUNT
FROM DEPARTMENT D
LEFT JOIN EMPLOYEE_DEPARTMENT ED ON D. DEPARTMENT_ID= ED. DEPARTMENT_ID
GROUP BY D. DEPARTMENT_NAME;
```

6. Count the number of employees in each college

```
SELECT C. COLLEGE_NAME, COUNT (EC. EMPLOYEE_ID) AS EMPLOYEE_COUNT
FROM COLLEGE C
LEFT JOIN EMPLOYEE_COLLEGE EC ON C. COLLEGE_ID=EC.COLLEGE_ID
GROUP BY C. COLLEGE_NAME;
```

RESULT:

The queries on the table where it defines a relation as “Employees are working for Departments and also for Colleges” was executed successfully.

Prepared by Team-01

A. Lalasa(23KB1A0506)

A. Lasya(23KB1A0507)

A. Chandhana(23KB1A0508)

SK. Masthan Bee(23KB1A05K6)

T. Mothi Sri(23KB1A05M2)

Y.Srija(23KB1A05P8)