

CSLR51 DBMS Session 7

2. Write the following as Cursors on the corresponding Schema.

Employee Schema

q. Develop a stored procedure to insert a new attribute 'address' in DEPENDENT and update the same as that of the employee's address.

Query:

```
CREATE PROCEDURE addAddr()
BEGIN
DECLARE DONE INT DEFAULT FALSE;
DECLARE temp_essn CHAR(10);
DECLARE temp_addr VARCHAR(50);
DECLARE temp_dep_name VARCHAR(15);
DECLARE addr_addition CURSOR FOR SELECT Essn,Dependent_name FROM Dependent;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;
ALTER TABLE new_dependent ADD COLUMN emp_addr VARCHAR(50);
OPEN addr_addition;
loop_label : LOOP
    IF done THEN
        LEAVE loop_label;
    END IF;
    FETCH addr_addition INTO temp_essn,temp_dep_name;
    SELECT Addr INTO temp_addr FROM Employee WHERE Ssn = temp_essn;
    UPDATE new_dependent SET emp_addr = temp_addr WHERE Essn = temp_essn AND Dependent_name =
temp_dep_name;
END LOOP loop_label;
SELECT * FROM new_dependent;
CLOSE addr_addition;
END;
```

Output:

```
mysql> CALL addAddr();
+-----+-----+-----+-----+-----+-----+
| Essn      | Dependent_name | Gender | Bdate      | Relationship | emp_addr                                |
+-----+-----+-----+-----+-----+-----+
| 123456789 | Alice          | F      | 1988-12-30 | Daughter    | 731 Fondren, Houston, TX              |
| 123456789 | Elizabeth      | F      | 1967-05-05 | Spouse      | 731 Fondren, Houston, TX              |
| 123456789 | Michae        | M      | 1988-01-04 | Son         | 731 Fondren, Houston, TX              |
| 333445555 | Alice          | F      | 1986-04-05 | Daughter    | 638 Voss, Houston, TX                 |
| 333445555 | Joy           | F      | 1958-05-03 | Spouse      | 638 Voss, Houston, TX                 |
| 333445555 | Theodore      | M      | 1983-10-25 | Son         | 638 Voss, Houston, TX                 |
| 987654321 | Abner          | M      | 1942-02-28 | Spouse      | 291 Berry, Bellaire, TX               |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.06 sec)
```

r. Develop a stored procedure to display the fname, ssn and salary, grade of an employee. Handle the condition such that if salary of an employee is 1 - 10000, assign grade3, grade2 if salary in between 10000 and 50000 and grade1 if salary > 50000. Handle exception with an error message when an invalid case occurs.

Query:

```
CREATE PROCEDURE getGrade(IN emp_ssn CHAR(10))
BEGIN
DECLARE DONE INT DEFAULT FALSE;
DECLARE temp_fname VARCHAR(15);
DECLARE temp_ssn CHAR(10);
DECLARE temp_salary DECIMAL(10,2);
DECLARE FOUND INT DEFAULT FALSE;
DECLARE find_grade CURSOR FOR SELECT Fname, Ssn, Salary FROM Employee;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;
OPEN find_grade;
loop_label : LOOP
    IF DONE THEN
        LEAVE loop_label;
    END IF;
    FETCH find_grade INTO temp_fname, temp_ssn, temp_salary;
    IF temp_ssn = emp_ssn THEN
        SET FOUND = TRUE;
        IF temp_salary IS NULL THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Salary of the employee not assigned!";
        ELSE
            SELECT temp_fname, temp_ssn, temp_salary,
                CASE WHEN temp_salary BETWEEN 1 AND 9999 THEN "Grade 3"
                     WHEN temp_salary BETWEEN 10000 AND 50000 THEN "Grade 2"
                     WHEN temp_salary > 50000 THEN "Grade 1"
                END
            AS Grade;
            LEAVE loop_label;
        END IF;
    END IF;
END LOOP;
CLOSE find_grade;
IF NOT FOUND THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Given SSN is not found the DB";
END IF;
END;
```

Output:

```
mysql> CALL getGrade("453453453");
+-----+-----+-----+-----+
| temp_fname | temp_ssn | temp_salary | Grade |
+-----+-----+-----+-----+
| Joyce      | 453453453 | 25000.00 | Grade 2 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

s. Create a stored procedure to display deptno, avgsalary and #employees in each department. Handle exceptions with an error message when invalid deptno is given.

Query:

```
CREATE PROCEDURE getDeptInfo(IN dep_no INT)
BEGIN
DECLARE DONE INT DEFAULT FALSE;
DECLARE temp_dno INT;
DECLARE FOUND INT DEFAULT FALSE;
DECLARE dept_info CURSOR FOR SELECT Dnumber FROM Department;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;
OPEN dept_info;
loop_label : LOOP
    FETCH dept_info INTO temp_dno;
    IF dep_no = temp_dno THEN
        SET FOUND = TRUE;
        IF DONE THEN
            LEAVE loop_label;
        END IF;
        SELECT dep_no,AVG(Salary) AS avg_salary,COUNT(*) AS "#employees" FROM Employee WHERE Dno =
temp_dno GROUP BY dep_no;
        LEAVE loop_label;
    END IF;
END LOOP;
CLOSE dept_info;
IF NOT FOUND THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Given department number is not found in the DB";
END IF;
END;
```

Output:

```
mysql> CALL getDeptInfo(5);
+-----+-----+-----+
| dep_no | avg_salary  | #employees |
+-----+-----+-----+
|      5 | 33250.000000 |          4 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Flight Schema

t. Develop a stored procedure to update an employee record given the employee id. Print a message after the update is successfully done with an exception handling of a invalid employee id.

Query:

```
CREATE PROCEDURE alterEmp(IN param_eid INT)
BEGIN
DECLARE DONE INT DEFAULT FALSE;
DECLARE FOUND INT DEFAULT FALSE;
DECLARE temp_eid INT;
DECLARE emp_data CURSOR FOR SELECT eid FROM employees;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;
OPEN emp_data;
loop_label : LOOP
    IF DONE THEN
        LEAVE loop_label;
    END IF;
    FETCH emp_data INTO temp_eid;
    IF temp_eid = param_eid THEN
        SET FOUND = TRUE;
        UPDATE employees SET SALARY = 1000 WHERE eid = temp_eid;
        LEAVE loop_label;
    END IF;
END LOOP;
CLOSE emp_data;
IF NOT FOUND THEN
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "Given EID number is not found in the DB";
END IF;
END;
```

Output:

```
mysql> table employees;
+-----+-----+-----+
| eid | ename   | salary |
+-----+-----+-----+
| 101 | Albert  | 10000  |
| 102 | Bob     | 250000 |
| 103 | Clair   | 10000  |
| 104 | Douglas | 450000 |
| 105 | Einstein | 30000  |
| 106 | Franklin | 1500000 |
| 107 | George  | 500000 |
| 108 | Harry   | 100000 |
| 109 | Jack    | 250000 |
| 110 | Lincon  | 75000  |
| 111 | James   | 50000  |
| 112 | Jobs    | 125000 |
| 113 | Maria   | 40000  |
| 201 | Morris  | 50000  |
| 202 | Nick    | 100000 |
| 203 | Parker  | 1000000 |
| 204 | Robert  | 150000 |
| 205 | Sam     | 500000 |
+-----+-----+-----+
18 rows in set (0.00 sec)

mysql> CALL alterEmp(111);
Query OK, 1 row affected (0.00 sec)

mysql> table employees;
+-----+-----+-----+
| eid | ename   | salary |
+-----+-----+-----+
| 101 | Albert  | 10000  |
| 102 | Bob     | 250000 |
| 103 | Clair   | 10000  |
| 104 | Douglas | 450000 |
| 105 | Einstein | 30000  |
| 106 | Franklin | 1500000 |
| 107 | George  | 500000 |
| 108 | Harry   | 100000 |
| 109 | Jack    | 250000 |
| 110 | Lincon  | 75000  |
| 111 | James   | 1000   |
| 112 | Jobs    | 125000 |
| 113 | Maria   | 40000  |
| 201 | Morris  | 50000  |
| 202 | Nick    | 100000 |
| 203 | Parker  | 1000000 |
| 204 | Robert  | 150000 |
| 205 | Sam     | 500000 |
+-----+-----+-----+
18 rows in set (0.01 sec)
```

u. Develop a stored procedure to display the name, salary of each employee from employee table. Handle the condition such that if salary of an employee is above 50,000 rank them as Grade 'A' else as Grade 'B'.

Query:

```
CREATE PROCEDURE showEmp()
BEGIN
DECLARE DONE INT DEFAULT FALSE;
DECLARE temp_eid INT;
DECLARE emp_info_data CURSOR FOR SELECT eid FROM employees;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;
OPEN emp_info_data;
CREATE TABLE temp_emp_info LIKE employees;
INSERT INTO temp_emp_info SELECT * FROM employees;
ALTER TABLE temp_emp_info ADD COLUMN Grade CHAR(1);
loop_label : LOOP
    FETCH emp_info_data INTO temp_eid;
    IF DONE THEN
        LEAVE loop_label;
    END IF;
    UPDATE temp_emp_info SET Grade = CASE WHEN salary >= 50000 THEN "A" ELSE "B" END
        WHERE eid = temp_eid;
END LOOP;
SELECT * FROM temp_emp_info;
DROP TABLE temp_emp_info;
CLOSE emp_info_data;
END;
```

Output:

```
mysql> CALL showEmp();
+-----+-----+-----+-----+
| eid | ename  | salary | Grade |
+-----+-----+-----+-----+
| 101 | Albert | 10000  | B     |
| 102 | Bob    | 250000 | A     |
| 103 | Clair  | 10000  | B     |
| 104 | Douglas | 450000 | A     |
| 105 | Einstein | 30000  | B     |
| 106 | Franklin | 1500000 | A     |
| 107 | George | 500000 | A     |
| 108 | Harry  | 100000 | A     |
| 109 | Jack   | 250000 | A     |
| 110 | Lincon  | 75000  | A     |
| 111 | James  | 1000   | B     |
| 112 | Jobs   | 125000 | A     |
| 113 | Maria  | 40000  | B     |
| 201 | Morris | 50000  | A     |
| 202 | Nick   | 100000 | A     |
| 203 | Parker | 1000000 | A     |
| 204 | Robert | 150000 | A     |
| 205 | Sam    | 500000 | A     |
+-----+-----+-----+-----+
18 rows in set (0.12 sec)

Query OK, 0 rows affected (0.13 sec)
```

v. Develop a stored procedure that builds a name list of all employees who are certified for a Boeing aircraft and handle an exception with an error message.

Query:

```
CREATE PROCEDURE showBoeing()
BEGIN
DECLARE DONE INT DEFAULT FALSE;
DECLARE temp_eid INT;
DECLARE temp_ename VARCHAR(50);
DECLARE count INT DEFAULT 0;
DECLARE boeing_info CURSOR FOR SELECT eid,ename FROM employees;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET DONE = TRUE;
CREATE TABLE temp_emp_boeing(
    emp_name VARCHAR(50)
);
OPEN boeing_info;
loop_label : LOOP
    FETCH boeing_info INTO temp_eid,temp_ename;
    IF DONE THEN
        LEAVE loop_label;
    END IF;
    IF EXISTS(SELECT eid FROM certified NATURAL JOIN aircraft WHERE eid = temp_eid AND aname = "Boeing")
    THEN
        INSERT INTO temp_emp_boeing VALUES (temp_ename);
        SET COUNT = COUNT + 1;
    END IF;
END LOOP;
CLOSE boeing_info;
IF COUNT > 0 THEN
    SELECT * FROM temp_emp_boeing;
    DROP TABLE temp_emp_boeing;
ELSE
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = "No pilots are piloting Boeing aircraft!";
END IF;
END;
```

Output:

```
mysql> CALL showBoeing();
+-----+
| emp_name |
+-----+
| Albert   |
| Bob      |
| Clair    |
| Douglas  |
| Einstein |
| George   |
| Harry    |
| Jack     |
| Lincon   |
+-----+
9 rows in set (0.07 sec)

Query OK, 0 rows affected (0.08 sec)
```