

Assignment 5

TRIGGER

Name: Kaveri Zanzane

Roll No: 242

Batch: C4

STATEMENT:

For the following relational schema: Emp(eid: integer, ename: string, age: integer, salary: real, did: integer) , Dept(did: integer, dname: string, budget: real, managerid: integer) Write a trigger on Emp table to ensure that the employee's salary does not exceed the Department Budget.

1. CREATE EMPLOYEE TABLE

```
CREATE TABLE Emp (  
    eid INTEGER PRIMARY KEY,  
    ename VARCHAR(50),  
    age INTEGER,  
    salary REAL,  
    did INTEGER,  
    FOREIGN KEY (did) REFERENCES Dept(did)  
);
```

EID	ENAME	AGE	SALARY	DID
1001	Kaveri	30	50000	1
1002	Swara	25	60000	1
1003	Shivani	40	70000	2
1004	Vaishnavi	35	80000	2
1005	Sakshi	28	90000	3
1006	Aishwarya	45	100000	3
1007	Arya	32	110000	3
1008	Sejal	27	120000	4

2. CREATE DEPARTMENT TABLE



SQL Worksheet

```
1 v CREATE TABLE Dept (  
2     did INTEGER PRIMARY KEY,  
3     dname VARCHAR(50),  
4     budget REAL,  
5     managerid INTEGER  
6 );  
7  
8 v INSERT INTO Dept (did, dname, budget, managerid)  
9 VALUES(1, 'Sales', 1000000, 101);  
0 v INSERT INTO Dept (did, dname, budget, managerid)  
1 VALUES(2, 'Marketing', 800000, 102);  
2 v INSERT INTO Dept (did, dname, budget, managerid)  
3 VALUES(3, 'Engineering', 1500000, 103);  
4 v INSERT INTO Dept (did, dname, budget, managerid)  
5 VALUES(4, 'Human Resources', 500000, 104);  
6  
7
```

DID	DNAME	BUDGET	MANAGERID
1	Sales	1000000	101
2	Marketing	800000	102
3	Engineering	1500000	103
4	Human Resources	500000	104

Download CSV

4 rows selected.

3. CREATE TRIGGER



The screenshot shows a web-based SQL editor interface. At the top, there's a header with a hamburger menu, a 'Live SQL' logo, and links for 'Feedback', 'Help', and a user profile 'kaverizanzanel'. Below the header, the main area is titled 'SQL Worksheet' and contains a code editor with a line number margin on the left (48 to 65). The code defines a trigger named 'Emp_salary_check' that fires before insert or update on the 'Emp' table. It declares two variables, 'total_salary' and 'dept_budget' of type 'REAL'. The trigger logic calculates the sum of salaries for the new employee and compares it to the department budget. If the total salary exceeds the budget, it raises an application error. Below the code editor, a status bar shows two error messages: 'ORA-20001: Total employee salaries cannot exceed department budget' and 'ORA-06512: at "SQL_RTCSCUVXDTMMWXYXXBSUTHNK.EMP_SALARY_CHECK", line 10'. At the bottom right, there is a button labeled 'Activate Window'.

```
48
49 CREATE OR REPLACE TRIGGER Emp_salary_check
50 BEFORE INSERT OR UPDATE ON Emp
51 FOR EACH ROW
52 DECLARE
53     total_salary REAL;
54     dept_budget REAL;
55 BEGIN
56     SELECT SUM(salary) INTO total_salary FROM Emp WHERE did = :NEW.did;
57
58     SELECT budget INTO dept_budget FROM Dept WHERE did = :NEW.did;
59
60 IF total_salary > dept_budget THEN
61     RAISE_APPLICATION_ERROR(-20001, 'Total employee salaries cannot exceed department budget');
62 END IF;
63 END;
64
65
```

ORA-20001: Total employee salaries cannot exceed department budget
ORA-06512: at "SQL_RTCSCUVXDTMMWXYXXBSUTHNK.EMP_SALARY_CHECK", line 10
ORA-06512: at "SYS.DBMS_SQL", line 1721

Activate Window

QUERIES:

```
CREATE TABLE Dept (  
    did INTEGER PRIMARY KEY,  
    dname VARCHAR(50),  
    budget REAL,  
    managerid INTEGER  
)
```

```
INSERT INTO Dept (did, dname, budget, managerid)  
VALUES(1, 'Sales', 1000000, 101),  
INSERT INTO Dept (did, dname, budget, managerid)  
VALUES(2, 'Marketing', 800000, 102),  
INSERT INTO Dept (did, dname, budget, managerid)  
VALUES(3, 'Engineering', 1500000, 103),
```

```
|INSERT INTO Dept (did, dname, budget, managerid)
|VALUES(4, 'Human Resources', 500000, 104)
```

```
|INSERT INTO Dept (did, dname, budget, managerid)
|VALUES(1, 'Sales', 1000000, 101)
```

```
|INSERT INTO Dept (did, dname, budget, managerid)
|VALUES(2, 'Marketing', 800000, 102)
```

```
|INSERT INTO Dept (did, dname, budget, managerid)
|VALUES(3, 'Engineering', 1500000, 103)
```

```
|INSERT INTO Dept (did, dname, budget, managerid)
|VALUES(4, 'Human Resources', 500000, 104)
```

```
|CREATE TABLE Emp (
|  eid INTEGER PRIMARY KEY,
|  ename VARCHAR(50),
|  age INTEGER,
|  salary REAL,
|  did INTEGER,
|  FOREIGN KEY (did) REFERENCES Dept(did)
|)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
|VALUES(1001, 'Kaveri', 30, 50000, 1)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1002, 'Swara', 25, 60000, 1)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1003, 'Shivani', 40, 70000, 2)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1004, 'Vaishnavi', 35, 80000, 2)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1005, 'Sakshi', 28, 90000, 3)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1006, 'Aishwarya', 45, 100000, 3)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1007, 'Arya', 32, 110000, 3)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1008, 'Sejal', 27, 120000, 4)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1009, 'Nikita', 38, 130000, 4)
```

```
|INSERT INTO Emp (eid, ename, age, salary, did)
```

```
|VALUES(1010, 'Pooja', 33, 140000, 4)
```

```
|CREATE OR REPLACE TRIGGER Emp_salary_check
```

```
|BEFORE INSERT OR UPDATE ON Emp
```

```
|FOR EACH ROW
```

```
|DECLARE
```

```
| total_salary REAL;
```

```
| dept_budget REAL;
```

```
|BEGIN
```

```
| SELECT SUM(salary) INTO total_salary FROM Emp WHERE did = :NEW.did;
|
| SELECT budget INTO dept_budget FROM Dept WHERE did = :NEW.did;
|
| IF total_salary > dept_budget THEN
|   RAISE_APPLICATION_ERROR(-20001, 'Total employee salaries cannot exceed
| department budget');
| END IF;
| END;
```

```
| CREATE OR REPLACE TRIGGER Emp_salary_check
| BEFORE INSERT OR UPDATE ON Emp
| FOR EACH ROW
| DECLARE
|   total_salary REAL;
|   dept_budget REAL;
| BEGIN
|   SELECT SUM(salary) INTO total_salary FROM Emp WHERE did = :NEW.did;
|
|   SELECT budget INTO dept_budget FROM Dept WHERE did = :NEW.did;
|
|   IF total_salary > dept_budget THEN
|     RAISE_APPLICATION_ERROR(-20001, 'Total employee salaries cannot exceed
| department budget');
|   END IF;
| END;
```

```
| INSERT INTO Emp VALUES (26733, 'Kaveri Zanzane', 20, 20000000, 4)
```
