

# DAY 16

PRN : 200243020003

## Triggers

1. The rows in the JOBS table store a minimum salary and a maximum salary allowed for different JOB\_ID values. You are asked to write code to ensure that employees salaries fall within the range allowed for their job type, for insert and update operations. Write a procedure called CHECK\_SALARY that accepts two parameters, one for an employee's job ID string and the other for the salary. The procedure uses the job ID to determine the minimum and maximum salary for the specified job. If the salary parameter does not fall within the salary range of the job, inclusive of the minimum and maximum, then it should raise an application exception, with the message Invalid salary . Salaries for job must be between and . Replace the various items in the message with values supplied by parameters and variables populated by queries. Save the file. [Use table Jobs] Create a trigger called CHECK\_SALARY\_TRG on the copy\_emp table that fires before an INSERT or UPDATE operation on each row. The trigger must call the CHECK\_SALARY procedure to carry out the business logic. The trigger should pass the new job ID and salary to the procedure parameters.

```
create
  OR REPLACE PROCEDURE check_salary (jid copy_emp.job_id % TYPE, sal
number) AS mn
  number;
mx number;
sal_exp
EXCEPTION;
BEGIN
  SELECT
    min_sal,
    max_sal INTO
mn,
mx
FROM
  jobs
WHERE
  job_id = jid;

IF sal NOT BETWEEN mn AND MX THEN
  raise sal_exp;

END IF;

EXCEPTION
WHEN sal_exp THEN
  DBMS_OUTPUT.PUT_LINE ('salary not in valid range');
```

```

END;

/ CREATE OR REPLACE TRIGGER check_salary_trg BEFORE INSERT
  OR UPDATE ON copy_emp FOR EACH ROW
BEGIN
    check_salary (:new.job_id, :new.salary);
END;

/
BEGIN
    INSERT INTO copy_emp
        values(& eid, '&fname', '&lname', '&email', '&phno', '&hdate',
        '&job', & sal, & comm, & mid, & did);
END;

/

```

2. You are asked to prevent employees from being deleted during business hours. Write a statement trigger called DELETE\_EMP\_TRG on the employees\_copy table to prevent rows from being deleted during weekday business hours, which are from 9:00 a.m. to 6:00 p.m. And day MON to FRI". Now, try deleting employee 205 from employees\_copy table.

```

CREATE OR REPLACE TRIGGER delete_emp_trg BEFORE DELETE ON copy_emp
BEGIN
    IF to_char(sysdate, 'DY')
        NOT IN('SAT', 'SUN') AND to_number(to_char(sysdate, 'HH24'))
        BETWEEN 09 AND 18 THEN
        raise_application_error (- 20202,
            'cannot perform delete operation b/w 9am to 6pm');
    END IF;
END;

/

```