

13-3: Creating DML Triggers: Part II

Vocabulary

- 1.Compound Trigger
- 2.INSTEAD OF trigger
- 3.Conditional Predicate
- 4.:OLD and :NEW qualifiers
- 5.DML Row Trigger

Try It / Solve It

2.

```
CREATE OR REPLACE TRIGGER emp_audit_trigg
```

```
AFTER INSERT OR DELETE ON employees  
BEGIN
```

```
IF INSERTING THEN INSERT INTO audit_table (action) VALUES ('Inserting');  
ELSIF DELETING THEN INSERT INTO audit_table (action) VALUES ('Deleting')
```

```
END IF;
```

```
END;
```

3.

```
ALTER TABLE audit_table
```

```
ADD(emp_id NUMBER(6));
```

```
CREATE OR REPLACE TRIGGER emp_audit_trigg
```

```
AFTER INSERT OR DELETE ON employees FOR EACH ROW
```

```
BEGIN
```

```
IF INSERTING THEN INSERT INTO audit_table (action,emp_id) VALUES  
('Inserted',:NEW.employee_id);
```

```
ELSIF DELETING THEN INSERT INTO audit_table (action,emp_id) VALUES  
('Deleted', :OLD.employee_id);
```

```
END IF;
```

```
END;
```

4.

C.

```
CREATE OR REPLACE TRIGGER emp_trigger
    INSTEAD OF INSERT OR DELETE ON emp_vu
    BEGIN
        IF INSERTING THEN
            update dept_count set count_emps = count_emps+1 where dept_id
= :NEW.department_id;
        ELSIF DELETING THEN
            update dept_count set count_emps = count_emps-1 where dept_id
= :OLD.department_id;
        END IF;
    END;
```

5.

A.

```
CREATE OR REPLACE TRIGGER emp_audit_trigg
    FOR UPDATE OF salary on employees
    COMPOUND TRIGGER
        old_sal NUMBER(6) := 0;
        new_sal NUMBER(6) := 0;
        BEFORE EACH ROW IS
        BEGIN
            old_sal := :OLD.salary;
            INSERT INTO audit_table(action) VALUES ('Updating');
        END BEFORE EACH ROW;

        AFTER EACH ROW IS
        BEGIN
```

```
new_sal := :NEW.salary;
```

```
INSERT INTO audit_table(action) VALUES ('Update complete:old salary = '||  
old_sal||' new salary ='||new_sal);
```

```
END AFTER EACH ROW;
```

```
END;
```

B

```
update employees set salary = 1200 where employee_id = 124;
```

```
SELECT * FROM AUDIT_TABLE;
```