Database Systems Laboratory 13

DATABASE TRIGGER

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

DATABASE TRIGGER

1. Database Trigger
2. Types of Triggers
3. : OLD and : NEW Qualifiers
4. INSTEAD OF Trigger

Database Trigger

Database Trigger

- Trigger is a stored procedure that is fired when a DML (INSERT, DELETE, UPDATE) statement is issued against an associated table
- Trigger is mainly used to enforce checks, search and backing up data
- A trigger can't use a TCL statement like COMMIT, ROLLBACK, or SAVEPOINT
- A variable in a trigger can't be declared with LONG or LONG RAW data type

- Components of Trigger:
  - Triggering SQL statement
  - Trigger action
  - Trigger restriction

Types of Triggers

BEFORE, AFTER & INSTEAD OF Triggers

- **BEFORE Triggers**: These triggers are fired before the triggering SQL statement is executed
- **AFTER Triggers**: These triggers are fired after the triggering SQL statement is executed
- **INSTEAD OF Triggers**: These triggers are used to make the non-updateable views updateable

Statement level & Row level Triggers

- **Statement level or Table level Triggers**: These triggers are fired for each DML operation being performed on a table. This is the default type
- **Row level Triggers**: These triggers are fired for each and every record which is inserted or deleted or updated from a table. This is the common type

Database Trigger...

Database Trigger Syntax

```
CREATE OR REPLACE TRIGGER triggername BEFORE/
      AFTER/INSTEAD OF Triggerevent ON tablename/viewname
      [FOR EACH ROW] [WHEN cond]
DECLARE
      declaration
BEGIN
      executable statements
EXCEPTION
      exception handling statements
END;
```

: OLD and : NEW Qualifiers

: OLD and : NEW Qualifiers

: OLD and : NEW qualifiers are related to ROW level triggers

Value of a column before the data change is referenced by prefixing OLD

Value of a column after the data change is referenced by prefixing NEW

Database Triggers...

EMP_TRIG(ename, sal)

EMP_BKUP(ename, sal, deletedate)

Create a trigger which checks the value of sal before INSERT or UPDATE statement and ensures that sal below 500 is not inserted

```
CREATE OR REPLACE TRIGGER MIN_SAL_CHK BEFORE
      INSERT OR UPDATE ON EMP_TRIG FOR EACH ROW
      WHEN(NEW.sal<500)
BEGIN
      RAISE_APPLICATION_ERROR(-20000,'sal must be above 500');
END;

.....
INSERT INTO EMP_TRIG VALUES('Prasant',2000);

INSERT INTO EMP_TRIG VALUES('Jack',300);
```

Database Triggers...

EMP_TRIG(ename, sal)

EMP_BKUP(ename, sal, deletedate)

Create a trigger which keeps backup of deleted records of EMP_TRIG table. Deleted records are inserted in EMP_BKUP table

```
CREATE OR REPLACE TRIGGER BKUP_REC AFTER
      DELETE ON EMP_TRIG FOR EACH ROW
BEGIN
      INSERT INTO EMP_BKUP VALUES( : OLD.ename,
          : OLD.sal, SYSDATE);
END;

.....
DELETE FROM EMP_TRIG WHERE ENAME='Prasant';
```

Database Triggers...

EMP_TRIG(ename, sal)

EMP_BKUP(ename, sal, deletedate)

Create a trigger which checks for any duplicate value and disallow insertion

```
CREATE OR REPLACE TRIGGER UNIQUE_VAL BEFORE
      INSERT ON EMP_TRIG FOR EACH ROW
DECLARE
      vc NUMBER(2);
BEGIN
      SELECT COUNT(*) INTO vc FROM EMP_TRIG
          WHERE ename=: NEW.ename;
      IF vc=1 THEN
           RAISE_APPLICATION_ERROR(-20024,'You
                have entered duplicate ename');
      END IF;
END;
```

INSTEAD OF Trigger

INSTEAD OF Trigger

Normally, we can't perform any of the DML operations on non-updateable views. But, INSTEAD OF Trigger can be used to make the non-updateable views updateable

It is used to modify a table that can't be modified through a view

```
CREATE OR REPLACE VIEW stud_faculty AS SELECT *
FROM student LEFT OUTER JOIN faculty ON
faculty.facultyid= student.facultyid;

.....
DELETE FROM stud_faculty WHERE facultyid=235;
```

INSTEAD OF Trigger...

INSTEAD OF Trigger...

```
CREATE OR REPLACE TRIGGER fac_delete INSTEAD OF
      DELETE ON stud_faculty FOR EACH ROW
BEGIN
      DELETE FROM faculty WHERE facultyid= : OLD.facultyid;
END;

.............
DELETE FROM stud_faculty WHERE facultyid=235;
```