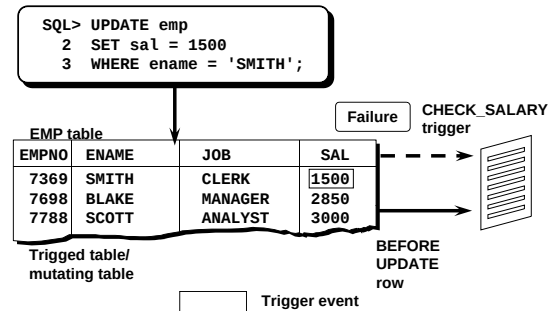


More Trigger Concepts

Reading Data from a Mutating Table



Mutating Table: Example

```
SQL> CREATE OR REPLACE TRIGGER check_salary
2 BEFORE INSERT OR UPDATE OF sal, job ON emp
3 FOR EACH ROW
4 WHEN (new.job <> 'PRESIDENT')
5 DECLARE
6   v_minsalary emp.sal%TYPE;
7   v_maxsalary emp.sal%TYPE;
8 BEGIN
9   SELECT MIN(sal), MAX(sal)
10  INTO   v_minsalary, v_maxsalary
11  FROM   emp
12  WHERE  job = :new.job;
13  IF :new.sal < v_minsalary OR
14     :new.sal > v_maxsalary THEN
15     RAISE_APPLICATION_ERROR(-20505, 'Out of range');
16  END IF;
17 END;
18 /
```

Mutating Table: Example

```
SQL> UPDATE emp
2 SET sal = 1500
3 WHERE ename = 'SMITH';
*
ERROR at line 2:
ORA-04091: table EMP is mutating, trigger/function
may not see it
ORA-06512: at "CHECK_SALARY", line 5
ORA-04088: error during execution of trigger
'CHECK_SALARY'
```

Implementation of Triggers

- Security
- Auditing
- Data integrity
- Referential integrity
- Table replication
- Derived data
- Event logging

Controlling Security Within the Server

```
SQL> GRANT SELECT, INSERT, UPDATE, DELETE
2 ON emp
3 TO CLERK; -- database role
SQL> GRANT CLERK TO SCOTT;
```

Controlling Security with a Database Trigger

```
SQL>CREATE OR REPLACE TRIGGER secure_emp
2 BEFORE INSERT OR UPDATE OR DELETE ON emp
3 DECLARE
4   v_dummy VARCHAR2(1);
5 BEGIN
6   IF (TO_CHAR (sysdate, 'DY') IN ('SAT','SUN'))
7   THEN RAISE_APPLICATION_ERROR (-20506,
8    'You may only change data during normal business
9    hours.');
```

```
10 END IF;
11 SELECT COUNT(*) INTO v_dummy FROM holiday
12 WHERE holiday_date = TRUNC (sysdate);
13 IF v_dummy > 0 THEN RAISE_APPLICATION_ERROR (-20507,
14 'You may not change data on a holiday.');
```

```
15 END IF;
16 END;
17 /
```

Auditing Using the Server Facility

```
SQL> AUDIT INSERT, UPDATE, DELETE
2 ON emp
3 BY ACCESS
4 WHENEVER SUCCESSFUL;
```

Oracle will store the audit information in a data dictionary table or operating system file.

Auditing Using a Trigger

```
SQL>CREATE OR REPLACE TRIGGER audit_emp_values
2 AFTER DELETE OR INSERT OR UPDATE ON emp
3 FOR EACH ROW
4 BEGIN
5   IF audit_emp_package.g_reason IS NULL THEN
6     RAISE_APPLICATION_ERROR (-20059, 'Specify a reason
7     for the data operation with the procedure
8     SET_REASON before proceeding.');
```

```
9   ELSE
10    INSERT INTO audit_emp_table (user_name, timestamp, id,
11    old_last_name, new_last_name, old_title, new_title,
12    old_salary, new_salary, comments)
13    VALUES (user, sysdate, :old.empno, :old.ename,
14    :new.ename, :old.job, :new.job, :old.sal,
15    :new.sal, :audit_emp_package.g_reason);
16 END IF;
17 END;
18 /
```

```
SQL>CREATE TRIGGER cleanup_audit_emp
2 AFTER INSERT OR UPDATE OR DELETE ON emp
3 BEGIN
4   audit_emp_package.g_reason := NULL;
5 END;
6 /
```

Enforce Data Integrity Within the Server

```
SQL> ALTER TABLE emp ADD
2 CONSTRAINT ck_salary CHECK (sal >= 500);
```

Protect Data Integrity with a Trigger

```
SQL>CREATE OR REPLACE TRIGGER check_salary
2 BEFORE UPDATE OF sal ON emp
3 FOR EACH ROW
4 WHEN (new.sal < old.sal)
5 BEGIN
6   RAISE_APPLICATION_ERROR (-20508,
7   'Do not decrease salary.');
```

```
8 END;
9 /
```

Enforce Referential Integrity Within the Server

```
SQL> ALTER TABLE emp
2 ADD CONSTRAINT emp_deptno_fk
3 FOREIGN KEY (deptno) REFERENCES dept(deptno)
4 ON DELETE CASCADE;
```

Protect Referential Integrity with a Trigger

```
SQL>CREATE OR REPLACE TRIGGER cascade_updates
2 AFTER UPDATE OF deptno ON dept
3 FOR EACH ROW
4 BEGIN
5     UPDATE emp
6     SET     emp.deptno = :new.deptno
7     WHERE   emp.deptno = :old.deptno;
8 END;
9 /
```

Replicate a Table Within the Server

```
SQL> CREATE SNAPSHOT emp_copy AS
2  SELECT * FROM emp@ny;
```

Replicate a Table with a Trigger

```
SQL>CREATE OR REPLACE TRIGGER emp_replica
2 BEFORE INSERT OR UPDATE ON emp
3 FOR EACH ROW
4 BEGIN /*Only proceed if user init. data operation,
5     NOT the casc. trigger.*/
6     IF INSERTING THEN
7         IF :new.flag IS NULL THEN
8             INSERT INTO emp@sf VALUES (:new.empno,
9             :new.ename,...,'B');
10            :new.flag := 'A';
11        END IF;
12    ELSE /* Updating. */
13        IF :new.flag = :old.flag THEN
14            UPDATE emp@sf SET ename = :new.ename, ...,
15            flag = :new.flag
16            WHERE empno = :new.empno;
17        END IF;
18        IF :old.flag = 'A' THEN :new.flag := 'B';
19        ELSE :new.flag := 'A';
20        END IF;
21    END IF;
22 END;
23 /
```

Compute Derived Data Within the Server

```
SQL> UPDATE dept
2  SET total_sal = (SELECT SUM(salary)
3                  FROM emp
4                  WHERE emp.deptno = dept.deptno);
```

Compute Derived Values with a Trigger

```
SQL>CREATE OR REPLACE PROCEDURE increment_salary
2  (v_id IN dept.deptno%TYPE,
3   v_salary IN dept.total_salary%TYPE)
4 IS
5 BEGIN
6     UPDATE dept
7     SET     total_sal = NVL (total_sal,0)+ v_salary
8     WHERE  deptno = v_id;
9 END increment_salary;
10 /
```

```
SQL>CREATE OR REPLACE TRIGGER compute_salary
2 AFTER INSERT OR UPDATE OF sal OR DELETE ON emp
3 FOR EACH ROW
4 BEGIN
5     IF     DELETING
6     THEN increment_salary(:old.deptno, -1 * :old.sal);
7     ELSEIF UPDATING
8     THEN increment_salary(:new.deptno, :new.sal-:old.sal);
9     ELSE increment_salary(:new.deptno, :new.sal);--INSERT
10 END IF;
11 END;
12 /
```

Log Events with a Trigger

```
CREATE OR REPLACE TRIGGER notify_reorder_rep
BEFORE UPDATE OF amount_in_stock, reorder_point
ON inventory FOR EACH ROW
DECLARE
v_descrip product.descrip%TYPE;
v_msg_text VARCHAR2(2000);
stat_send number(1);
BEGIN
IF :new.amount_in_stock <= :new.reorder_point THEN
SELECT descrip INTO v_descrip
FROM PRODUCT WHERE prodid = :new.product_id;
v_msg_text := 'ALERT: INVENTORY LOW ORDER:'||CHR(10)||
... 'Yours,' ||CHR(10) ||user || '.'|| CHR(10)|| CHR(10);
ELSIF
:old.amount_in_stock<:new.amount_in_stock THEN NULL;
ELSE
v_msg_text := 'Product #'||... CHR(10);
END IF;
DBMS_PIPE.PACK_MESSAGE(v_msg_text);
stat_send := DBMS_PIPE.SEND_MESSAGE('INV_PIPE');
END;
```

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.