

Relational Database Design

Triggers

1

Triggers What & Where

□ **Triggers are stored procedures that are invoked by the DBMS in response to database activities (e.g. insert, update and delete).**

□ In DBMSs, the stored procedure are written in a programming extension to SQL. Consequently we have:

- **Error control**
- **Other involved inserts / updates / deletes (single tables)**
- **Complicated inserts / updates / delete (multi tables with multi table spanning integrity constraints)**
- **Some system interaction events (e.g. on logon, on new session)**

□ **Where can we use triggers in:**

- to fire when updating data and checks a validation step;
- trigger overrides updating if outside the data's constraints;
- trigger to check status of a session when a new process starts;
- trigger to initialize a number of activities (e.g. audit) associate to a change of a base database object.

Joseph Vella© - Data Modelling and Database Triggers

Slide 2

2

Triggers

For "aggregate checks"

- Say we have a budget rule that states our salary bill cannot exceed 100,000 liras.

- A **check constraint** on the employees salary will not do - this is an aggregate check!

- A trigger that checks for the total limits is attached to the emp table and is fired on inserts and updates. If the condition (I.e. limit is exceeded) then an application error is raised!

- That is, check once for every INSERT or UPDATE (which can affect one or many tuples).

```
CREATE OR REPLACE TRIGGER emp_sal_trigger
AFTER INSERT OR UPDATE ON emp
DECLARE
    total    NUMBER; max_salary CONSTANT NUMBER := 100000;
BEGIN
    SELECT sum(sal) INTO total FROM emp;
    IF total > max_salary THEN
        RAISE_APPLICATION_ERROR(-20100,'Salary Budget Exceeded!');
    END IF;
END;
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 3

3

Triggers

For "row level rule"

- Say we have a budget rule that states an employee's salary can have an increase of up to 25% at a time.

- A **check constraint** on the employees salary will not do - this is a capping restriction!

- We need to attach a trigger to the sal attribute of emp and time it with update statements. If indeed the change is greater than the allowed we have to override the raise with the 25% limit.

- If the event matches (I.e. an update) then we need to check if the new salary is over the allowed threshold - use the WHEN condition.

```
CREATE OR REPLACE TRIGGER emp_sal_hike_trigger
BEFORE UPDATE OF sal ON emp
FOR EACH ROW
WHEN ( new.sal > ( 1.25 * old.sal ) )
DECLARE
    max_salary NUMBER;
BEGIN
    :new.sal := ( 1.25 * :old.sal );
END;
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 4

4

Triggers

Firing Specification

When do triggers fire? This is determined by (twelve possible combinations):

Action

SQL statement - INSERT, UPDATE or DELETE (any combination)

Some others too! - DDL, Errors, User logon / logoff

Level

either STATEMENT or ROW level triggering

1. How often is the trigger fired (ST - once, ROW - per tuple)

2. Visibility of currently updating row's values (ROW - yes, ST - no)

avoid row level triggers on updates of a large number of rows

Note: for row level triggering we have an important naming mechanism, this includes the new and old values - correlation values.

Timing

whether the trigger fires either BEFORE or AFTER the statement-level or row-level execution.

In the case of an update operation both before and after have access to the OLD and NEW values - but the before can change the course!

Joseph Vella© - Data Modelling and Database Triggers

Slide 5

5

The Twelve Types of Triggers

Insert Trigger	Update Trigger	Delete Trigger
BEFORE INSERT STATEMENT	BEFORE UPDATE STATEMENT	BEFORE DELETE STATEMENT
BEFORE INSERT ROW	BEFORE UPDATE ROW	BEFORE DELETE ROW
AFTER INSERT ROW	AFTER UPDATE ROW	AFTER DELETE ROW
AFTER INSERT STATEMENT	AFTER UPDATE STATEMENT	AFTER DELETE STATEMENT

HOW MANY TRIGGERS PER OBJECT?

Triggers are an overhead! Also some triggers can fire other triggers (through definition of delete cascade or the code of a fired trigger) - cascading triggers ...

Joseph Vella© - Data Modelling and Database Triggers

Slide 6

6

Triggers for Database Design with Stored Procedures

3

Triggers Various Examples

□ The structure of a trigger specification is:

```
CREATE OR REPLACE TRIGGER
    firing specification
BEGIN    body    END;
```

□ Examples (of firing specs):

```
BEFORE UPDATE OF SALARY ON EMP ...
```

```
BEFORE UPDATE OF SALARY ON EMP
FOR EACH ROW
WHEN OLD.SALARY < 50000 ...
```

```
AFTER UPDATE OF EMP
FOR EACH ROW
WHEN NEW.SALARY - OLD.SALARY < 1000 ...
```

□ Example (of trigger body):

```
CREATE OR REPLACE TRIGGER purchase_date
BEFORE INSERT OR UPDATE ON purchase_order
FOR EACH ROW
BEGIN
    :new.purchase_date := TRUNC(:new.purchase_date);
END;
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 7

7

Triggers and Transaction Processing

□ Transaction control (e.g. commit and rollback) are not allowed inside a trigger. Even if a trigger calls a procedure which in turn submits a transaction control command then a run-time error is flagged during the firing of the trigger.

□ The protocol by which a database transaction and a trigger should communicate with each other is through the error control handling.

□ Specifically a trigger should raise an error through the DBMS_STANDARD package procedure:

```
RAISE_APPLICATION_ERROR ( ... );
```

- Important note: make sure all is cleared in the trigger environment before an error is RAISED (we need to have a clean slate for any forthcoming trigger firing)!!

□ The operation whose database transaction has risen an error exception has to cater by an explicit rollback.

Joseph Vella© - Data Modelling and Database Triggers

Slide 8

8

Triggers Administration

Trigger administration:

- Enable / disable all table's triggers:
 - `ALTER TABLE purchase_order ENABLE ALL TRIGGERS;`
 - `ALTER TABLE purchase_order DISABLE ALL TRIGGERS;`
- Enable /disable a table trigger:
 - `ALTER TRIGGER purchase_order_dir ENABLE;`
 - `ALTER TRIGGER purchase_order_dir DISABLE;`
- Purge a trigger:
 - `DROP TRIGGER purchase_order_dir;`
- Compile a trigger:
 - `ALTER TRIGGER purchase_order_dir COMPILE;`

Data dictionary views and triggers:

- `user_triggers` view;
- `user_trigger_cols` view;

Joseph Vella© - Data Modelling and Database Triggers

Slide 9

9

Triggers for Auditing

How about tracing (i.e. an audit trail) the changes to an employee's salary.

□ Assume we have the following tables: □ A typical audit trail trigger would be:

```
emp(enumb number PRIMARY KEY,
    ename varchar2(25),
    sal number not null,
    ...);
```

```
sal_change_trail
(enumb number,
 tran_date date not null,
 user_name varchar2(25),
 old_sal number not null,
 new_sal number not null,
 CONSTRAINT sct
 PRIMARY KEY
 (enumb, tran_date));
```

```
CREATE OR REPLACE TRIGGER salaud
AFTER UPDATE ON emp
FOR EACH ROW
BEGIN
    IF :old.enumb <> :new.enumb
    THEN RAISE APPLICATION_ERROR
        (20000,'PK broken');
    END IF;
    IF :old.sal <> :new.sal THEN
        BEGIN
            INSERT INTO
                sal_change_trail
            ( ... ) VALUES
            ( ... );
        END;
    END IF;
END;
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 11

11

Trigger Example in PostgreSQL (Row Level)

```
CREATE TABLE person (
    id serial PRIMARY KEY,
    first_name varchar not null,
    middle_name varchar not null,
    last_name varchar not null,
    full_name varchar not null
);
CREATE OR REPLACE FUNCTION compute_person_full_name() RETURNS trigger
AS $$
BEGIN
    NEW.full_name := NEW.first_name ||
        CASE WHEN NEW.middle_name = '' THEN ''
              ELSE ' ' || NEW.middle_name END || ' ' ||
              NEW.last_name;
    RETURN NEW;
END
$$ LANGUAGE plpgsql; // r haas 2010
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 12

12

Trigger Example in PostgreSQL (Row Level)

```
CREATE TRIGGER compute_person_full_name
BEFORE INSERT OR UPDATE ON person
FOR EACH ROW
EXECUTE PROCEDURE compute_person_full_name();

INSERT INTO person
(first_name, middle_name, last_name)
VALUES ('Robert', 'E.', 'Scott'),
       ('Harry', '', 'Larrison');

SQL> select full_name from person;
 full_name
-----
Robert E. Scott
Harry Larrison
(2 rows)
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 13

13

Trigger Example in PostgreSQL (Aggregate Level)

```
CREATE TABLE orders (
    id serial primary key,
    customer_name varchar not null,
    number_of_items integer not null default 0,
    total_price numeric(12,2) not null default 0
);

CREATE TABLE order_items (
    order_id integer not null references orders (id),
    item_name varchar not null,
    price numeric(12,2) not null default 0
);
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 14

14

Trigger Example in PostgreSQL (Aggregate Level)

```
CREATE OR REPLACE FUNCTION update_order_stats()
    RETURNS trigger AS $$
BEGIN
    IF (TG_OP IN ('INSERT', 'UPDATE')) THEN
        UPDATE orders
            SET number_of_items = number_of_items + 1,
                total_price = total_price + NEW.price
            WHERE id = NEW.order_id;
    END IF;
    IF (TG_OP IN ('UPDATE', 'DELETE')) THEN
        UPDATE orders
            SET number_of_items = number_of_items - 1,
                total_price = total_price - OLD.price
            WHERE id = OLD.order_id;
    END IF;
    RETURN NULL;
END
$$ LANGUAGE plpgsql;
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 15

15

Trigger Example in PostgreSQL (Aggregate Level)

```
CREATE TRIGGER update_order_stats
  AFTER INSERT OR UPDATE OR DELETE ON order_items
  FOR EACH ROW
    EXECUTE PROCEDURE update_order_stats();
```

Joseph Vella© - Data Modelling and Database Triggers

Slide 16

16