

SQL: Complex Integrity Constraints, Triggers

Chapter 5.7-5.9

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 1

Integrity Constraints

- Describes conditions that must be satisfied by every legal instance
- Types of integrity constraints
 - Domain constraints
 - Primary key constraints
 - Foreign key constraints
 - **General constraints**

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 2

Table Constraints

- More general than key constraints
- Can use a query to express constraint
 - Constraints checked each time table updated
 - CHECK constraint always true for empty relation

```

CREATE TABLE Sailors
  ( sid INTEGER,
    sname CHAR(10),
    rating INTEGER,
    age REAL,
    PRIMARY KEY (sid),
    CHECK ( rating >= 1
           AND rating <= 10 )
CREATE TABLE Reserves
  ( sname CHAR(10),
    bid INTEGER,
    day DATE,
    PRIMARY KEY (bid,day),
    CONSTRAINT noInterlakeRes
    CHECK ('Interlake' <>
           (SELECT B.bname
            FROM Boats B
            WHERE B.bid=bid)))

```

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 3

Constraints Over Multiple Relations

- For general constraint over multiple tables, use an **assertion**

Number of boats plus number of sailors is < 100

```
CREATE ASSERTION smallClub
CHECK
((SELECT COUNT (S.sid) FROM Sailors S) +
 (SELECT COUNT (B.bid) FROM Boats B) < 100)
```

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 4

Active Databases & Triggers

- Trigger**: procedure that starts automatically if specified changes occur to the DBMS
- Three parts:
 - Event (activates the trigger)
 - Condition (tests whether the trigger should run)
 - Action (what happens if the trigger runs)
 - Before and After Triggers
- Trigger Execution
 - Row-level Triggers: Once per modified row
 - Statement-level Triggers: Once per SQL statement

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 5

Example

- Event: Insert new row into Students
- Condition: New student's age ≥ 18
- Event: Update university's total enrollment

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 6

Oracle Trigger Basic Syntax

```
CREATE [OR REPLACE] TRIGGER <trigger_name>
{BEFORE | AFTER} {INSERT | DELETE | UPDATE} ON <table_name>
[REFERENCING [NEW AS <new_row>] [OLD AS <old_row>]]
[FOR EACH ROW [WHEN (<condition>)]]
<trigger_body>
```

In Oracle, <trigger_body> is a PL/SQL block

Oracle Triggers Not quite the same As SQL Standard

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 7

Oracle Trigger Example

Please note that the example on this slide has been updated

```
CREATE TRIGGER print_on_insert AFTER INSERT ON Student /* Event */
FOR EACH ROW
WHEN (new.age >= 18)
BEGIN /* Action */
    DBMS_OUTPUT.PUT_LINE('Inserting student over 18');
END;
.
RUN;
```

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 8

Triggers

- First trigger executed *before* the activating statement, second executes *after* the activating statement.
- In combination with:
 - "FOR EACH ROW" - execute once per modified record
 - (default) - execute once per activating statement.
- In combination with:
 - "INSERT"
 - "DELETE"
 - "UPDATE"

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 9

Triggers

- Referencing old and new outside the when clause

The NEW and OLD keywords, when specified in the WHEN clause, are not considered bind variables, so are not preceded by a colon (:). However, you must precede NEW and OLD with a colon in all references other than the WHEN clause.

- For more options see:
http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/statements_7004.htm#i2235611

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 10

Oracle Trigger Example

```
CREATE TABLE T1 (a INTEGER, b CHAR(10));
CREATE TABLE T2 (c CHAR(10), d INTEGER);

CREATE TRIGGER trigger1
AFTER INSERT ON T1
FOR EACH ROW
WHEN (NEW.a <= 10)
BEGIN
  INSERT INTO T2 VALUES(:NEW.b, :NEW.a);
END;
.
RUN;
```

What does this trigger do?

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 11

Triggers: Pitfalls and Pain

- Triggers can be recursive!
 - Chain of triggers can be hard to predict, which makes triggers difficult to understand and debug.
- Errors with "mutating" table
 - A table that is currently being modified by an UPDATE, DELETE, or INSERT statement, or a table that might be updated by the effects of a DELETE CASCADE constraint.
 - The session that issued the triggering statement cannot query or modify a mutating table.
 - Used to prevent a trigger from seeing inconsistent data.

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 12

Triggers

```
CREATE OR REPLACE TRIGGER Emp_count
AFTER DELETE ON Emp_tab
FOR EACH ROW
DECLARE
  n INTEGER;
BEGIN
  SELECT COUNT(*) INTO n FROM Emp_tab;
  DBMS_OUTPUT.PUT_LINE(' There are now ' || n || ' employees. ');
END;
```

DELETE FROM Emp_tab WHERE Empno = 7499;

ORA-04091: table Emp_tab is mutating, trigger/function may not see it

- Example from http://download-west.oracle.com/docs/cd/B10501_01/appdev.920/a96590/adq13trq.htm#786
- Also see <http://asktom.oracle.com/tkyte/Mutate/index.html>

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 13

Triggers vs. Constraints

- Often used to maintain consistency
 - Can you use a foreign key?
 - Foreign keys are defined declaratively
- Constraints are easier to understand than triggers
- Triggers are more powerful.
 - Often used to fill out fields in a form
 - Check complex actions (such as credit limit in a shopping application)
 - Check preferred customer status
 - Generate logs for auditing and security checks.

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 14

More Notes on JDBC, Database Application Programming

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 15

SQL Injection – Example

- Simple, easily-prevented, yet surprisingly common vulnerability

The diagram illustrates the flow of data from a user interface to a database. On the left, a yellow box labeled 'Web Front-End' contains two input fields: 'Name' and 'Password'. A double-headed arrow connects this box to a purple cloud labeled 'JDBC Application'. Another double-headed arrow connects the 'JDBC Application' to an orange cylinder labeled 'Grades Database'.

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 16

SQL Injection – Example

JDBC
Authenticate (String n, String p) {
...
String query = "SELECT grade FROM students WHERE name = ' " + n + "' AND password = ' " + p + "'";
...
}

Input:
name = bart; password = mypwd
name = lisa ; password = n' OR 'x'='x
name = foo
password = n'; UPDATE students SET grade = 'A'

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 17

SQL Injection

What is a simple way to prevent this kind of attack?

Validate input!
Use a prepared statement!!

2/2/09 EECS 484: Database Management Systems, Kristen LeFevre 18



Announcements

- Optional Exercises: 5.1, 5.3
