

Constraints & Triggers

Triggers – Introduction

Triggers

“Event-Condition-Action Rules”

When *event* occurs, check *condition*; if true, do *action*

1) Move monitoring logic from apps into DBMS

2) Enforce

- Backward compatibility
- Automatic constraint repair

This intro: **SQL standard**

Demo: **SQLite**

* Implementations vary significantly

Triggers in SQL

```
Create Trigger name  
Before | After | Instead Of events  
[ referencing-variables ]  
[ For Each Row ]  
When ( condition )  
action
```

insert on T new
delete on T old
update [of C1...Cn]
on T old, new

once for each modified tuple
old row as var
new row as var
old table as var
new table as var

like SQL where
→ general assertion

SQL statement

Referential Integrity:

✗ R.A references S.B, cascaded delete

Create Trigger **Cascade**

After Delete On **S** ←

Referencing Old Row As **O** ←

For Each Row ←

[no condition]

Delete From **R** where **A = O.B** ←

Referential Integrity:

R.A references S.B, cascaded delete

Create Trigger **Cascade**

After Delete On **S**

Referencing Old Row As **O**

[For Each Row]**X**

[no condition]

Delete From **R** where **A = O.B**

Referential Integrity:

R.A references S.B, cascaded delete

```
Create Trigger Cascade  
After Delete On S  
Referencing Old Table As OT  
[ For Each Row ]  
[ no condition ]  
Delete From R where A = O.B
```

*deleted
tuples*

Referential Integrity:

R.A references S.B, cascaded delete

Create Trigger **Cascade**

After Delete On **S**

Referencing **Old Table** As **OT**

[For Each Row]

[no condition]

Delete From **R** where **A** in (select **B** from **OT**)

Tricky Issues

- Row-level vs. Statement-level
 - New/Old Row and New/Old Table
 - Before, Instead Of
- Multiple triggers activated at same time
- Trigger actions activating other triggers (chaining)
 - Also self-triggering, cycles, nested invocations
- Conditions in when vs. as part of action
- ✳ Implementations vary significantly

Which goes first?

T(K,V) — K key, V value

Create Trigger **IncreaseInserts**

After Insert On T

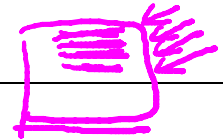
Referencing New Row As NR, New Table As NT

~~X~~ For Each Row ←

When (Select Avg(V) From T) <
(Select Avg(V) From NT)

stable 25

Update T set V=V+10 where K=NR.K



- No statement-level equivalent
- Nondeterministic final state

Triggers

“Event-Condition-Action Rules”

When *event* occurs, check *condition*; if true, do *action*

- 1) Move monitoring logic from apps into DBMS
- 2) Enforce constraints
 - Beyond what constraint system supports
 - Automatic constraint “repair”

* Implementations vary significantly