Triggers

Lecture 14

Robb T. Koether

Hampden-Sydney College

Wed, Feb 14, 2018

Outline

Triggers

Triggers

```
CREATE TRIGGER trigger_name
trigger_time trigger_event
ON table_name
FOR EACH ROW
trigger_action
```

- A trigger is an event-action pair.
- When the `trigger_event` occurs, the `trigger_action` is taken.
- The `trigger_time` is either `BEFORE` or `AFTER`.
- The `trigger_event` is `INSERT`, `DELETE`, or `UPDATE`.

Triggers

- The `trigger_action` is a statement (query) or series of statements to be executed.
- If there is more than one statement, then they must be enclosed in a `BEGIN`-`END` block and separated by semicolons.
- The action is executed once for each tuple that is affected by the event.
    - For an insertion, the affected tuples are the inserted tuples.
    - For a deletion, the affected tuples are the deleted tuples.
    - For an update, the affected tuples are the updated tuples.

Triggers

- The keyword `OLD` refers to a deleted or updated tuple.
- The keyword `NEW` refers to an added or updated tuple.
- For an update event, `OLD` and `NEW` will always refer to the same tuple, but before and after the update.

Triggers

Triggers

```
CREATE TRIGGER fire_emp
AFTER DELETE ON employees
FOR EACH ROW
DELETE FROM dependents
WHERE dependents.ssn = OLD.ssn
```

- For example, when an employee is deleted from `employees`, all corresponding tuples must also be deleted from `dependents`.
- What about the table `works_on`?

Triggers

Triggers

```
CREATE TRIGGER drop_emp
AFTER DELETE ON employees
FOR EACH ROW
BEGIN
DELETE FROM dependents WHERE dependents.ssn = OLD.ssn;
DELETE FROM works WHERE works.ssn = OLD.ssn;
END;
```

- However, the semicolon will prematurely end the CREATE command.
- What to do?

Triggers

Triggers

```
DELIMITER #
CREATE TRIGGER drop_emp
AFTER DELETE ON employees
FOR EACH ROW
BEGIN
DELETE FROM dependents WHERE dependents.ssn = OLD.ssn;
DELETE FROM works WHERE works.ssn = OLD.ssn;
END#
DELIMITER ;
```

- To handle that situation, we need to temporarily redefine the delimiter, which has been the semicolon.

Outline

Cascading Triggers

- Suppose that we delete a department from the `departments` table.
- Then all employees in that department should be deleted from the `employees` table.
- But then all dependents of those employees should be deleted from the `dependents` table.

Cascading Triggers

- Therefore, we need two triggers.
  - One trigger will delete all employees from the `employees` table when the department is deleted from the `departments` table.
  - The other trigger will delete all dependents of an employee from the `dependents` table when the employee is deleted from the `employees` table.
- We see from this example that triggers may cascade.

Cascading Triggers

Cascading Triggers

```
CREATE TRIGGER drop_dept
AFTER DELETE ON departments
FOR EACH ROW
DELETE FROM employees
WHERE employees.dept = OLD.dept;

CREATE TRIGGER drop_emp
AFTER DELETE ON employees
FOR EACH ROW
DELETE FROM dependents
WHERE dependents.ssn = OLD.ssn;
```

- The trigger `drop_dept` will automatically invoke the trigger `drop_emp`.

Outline

Update Triggers

Updating a Social Security Number

```
CREATE TRIGGER update_ssn
AFTER UPDATE ON employee
FOR EACH ROW
UPDATE dependents SET dependents.ssn = NEW.ssn
WHERE dependents.ssn = OLD.ssn;
```

- This trigger will update the dependents' SSN's when the employees' SSN's are updated.

Update Triggers

Updating a Social Security Number

```
CREATE TRIGGER update_ssn
AFTER UPDATE ON employee
FOR EACH ROW
UPDATE dependents SET dependents.ssn = NEW.ssn
WHERE dependents.ssn = OLD.ssn;
```

- This trigger will update the dependents' SSN's when the employees' SSN's are updated.
- Note the use of `NEW` and `OLD`.

Update Triggers

Insert a Friend

```
CREATE TRIGGER make_friend
AFTER INSERT ON friends
FOR EACH ROW
INSERT INTO friends
VALUES (NEW.user2, NEW.user1);
```

- In the Tigerface database, this trigger in intended to insert the same pair in reverse order.
- It does not work.

Update Triggers

Insert a Friend

```
CREATE TRIGGER make_friend
AFTER INSERT ON friends
FOR EACH ROW
INSERT INTO friends
VALUES (NEW.user2, NEW.user1);
```

- In the Tigerface database, this trigger in intended to insert the same pair in reverse order.
- It does not work. Why not?

Update Triggers

Insert a User

```
CREATE TRIGGER update_stats
AFTER INSERT ON users
FOR EACH ROW
UPDATE user_stats
SET user_count = user_count + 1;
```

- Suppose we have another table `user_stats` in which we record, among other things, the number of users.
- Write a similar trigger for deletions from `users`.

Outline

DisplayingTriggers

Displaying Triggers

```
SHOW TRIGGERS;
```

- We may display all triggers.

Displaying and Dropping Triggers

Dropping Triggers

```
DROP TRIGGER trigger_name;
```

- Triggers are given names so that they can be dropped.

Outline

Examples

- Using the company database, create triggers for
  - Deleting a project.
  - Updating a department number.