SQL Data Definition Language II

Tom Kelliher, CS 318

Feb. 4, 2002

1 Administrivia

Announcements

Assignment

Read 5.1–4.

From Last Time

Outline

1. Event-driven constraints.
2. Altering a database's schema.
3. Access control and views.

Coming Up

The E/R model.

2 Event-Driven Constraints

1. We're all familiar with events.

2. Event-driven constraints are actions taken as the result of the occurrence of an event.

2.1 Reactive Constraints

1. Static constraints backed by actions.

2. Idea: `ON Event Action`

3. Example:

```
CREATE TABLE Teaching (
   ProfId         INTEGER,
   CrsCode        CHAR(6),
   Semester       CHAR(6),
   PRIMARY KEY (CrsCode, Semester),
   FOREIGN KEY (ProfId) REFERENCES Professor
      ON DELETE NO ACTION
      ON UPDATE CASCADE,
   FOREIGN KEY (CrsCode) REFERENCES Course
      ON DELETE SET NULL
      ON UPDATE CASCADE );
```

4. Events:

   (a) Delete.

   (b) Update.

5. Actions:

   (a) No action.

   (b) Cascade.

   (c) Set null.

2.2 Triggers

1. Reactive foreign key constraints are not general enough — suppose the referenced set of attributes are not a candidate key or, for many DBMSs, are not the primary key?

2. Solution: triggers — a general event/action mechanism.

3. `CREATE TRIGGER` specifies an event an associates code — a stored procedure — with it.

4. PostgreSQL example — verify no one given more than a 5% raise:

```
CREATE FUNCTION RaiseCheck () RETURNS OPAQUE AS '
   BEGIN
      IF NEW.Salary > 1.05 * OLD.Salary THEN
         -- Excessive salary raise - abort transaction.
         RAISE EXCEPTION ''% given an excessive raise'', OLD.EmpName;
      END IF;
      -- Salary increase OK, proceed with transaction.
      RETURN NEW;
   END;
' LANGUAGE 'plpgsql';

CREATE TRIGGER LimitRaises BEFORE UPDATE ON Employee
   FOR EACH ROW EXECUTE PROCEDURE RaiseCheck();
```

   (a) What kind of constraint (static, dynamic) does this implement?

   (b) Alternatively, the code in the if block could have capped the increase at 5%: `NEW.Salary := 1.05 * OLD.Salary;`.

3 Altering a Database's Schema

Examples:

```
DROP TABLE AllMyLifesWork;   -- Whoops.

DROP DATABASE kelliher;
```

```
ALTER USER kelliher WITH PASSWORD 'BitterSweetMint';

ALTER TABLE People
   ADD COLUMN FavIceCream CHAR(6);

ALTER TABLE People
   ADD FOREIGN KEY (FavIceCream) REFERENCES IceCream (Code);
```

PostgreSQL limited on alter functionality. Copy table adding/deleting appropriately.

4 Access Control and Views

1. Access control does what it says.
2. Initially, only the owner of the database has access privileges.
3. Privileges can be granted to other database users:
   (a) Select.
   (b) Insert.
   (c) Update.
   (d) Delete.
   (e) Rule.
   (f) All.
4. Examples:

```
GRANT SELECT, UPDATE ON Employee
   TO GROUP Personnel;

GRANT SELECT ON Books
   TO PUBLIC;
```

```
GRANT ALL ON BillG
   TO kelliher;
```

The *object* of a grant can be a table, view, or sequence (PostgreSQL).

5. Views are just virtual tables created by queries (a query returns a table):

```
CREATE VIEW ProfStud AS
SELECT Te.ProfId AS Prof, Tr.StuID AS Stud
FROM Transcript Tr, Teaching Te
WHERE Tr.CrsCode = Te.CrsCode
   AND Tr.Semester = Te.Semester;
```

PostgreSQL version of 4.5.

6. Views in PostgreSQL are read-only.

7. Obviously, views can restrict tables or combine tables to create new tables.

8. Combining grant and views allows us to specify who can access what in a database and how.