CS 1555

Lecture 16

**Transactions in SQL (continued)**

Concurrent execution of transactions

- Maybe disrupted somewhere along the way, gives chance for another transaction to be interleaved

- Can cause errors with executing in the wrong order

Write/exclusive lock

- In SELECT statement, can specify FOR UPDATE OF <attribute>

- Prevents it from reading new data while current data is still being used 🡪 don’t let anyone else read the table while current statement is updating it

**Database Programming at Large**

Database programming

- Access a database from an application program (as opposed to interactive interfaces)

Database programming approaches

- Embedded commands: database commands are embedded in general-purpose language

- Library of database functions: API

- Brand new, full-fledged language: PL/SQL (procedural language extensions to SQL)

Approach 3: SQL/PL

- Functions/procedures can be written in SQL itself, or in an external programming language

- Functions are very useful with specialized data types

- Some databases support table-valued functions, which can return a relation as a result

ANSI SQL functions

- Using authors(author, title, author\_order)

- Definition of a function

- CREATE OR REPLACE FUNCTION author\_count(name varchar(20)) return integer

- a\_count integer; -- local variable declaration

- begin… end; -- function body

- SELECT COUNT(author) INTO a\_count;

- INTO is a tuple assignment operator

- ‘/’ executes a PL/SQL block

- Invocation: use it anywhere it is appropriate in a SELECT statement

PL/pgSQL function

- Create a function statement

- CREATE FUNCTION <name> RETURN(S) type AS $$ BEGIN… END; $$ LANGUAGE <language>

- Drop function: DROP FUNCTION IF EXISTS

More on triggers in Postgres

- Can create a constraint trigger

- Constraint triggers must be after row triggers

- SET CONSTRAINTS trig-name <EVALUATION MODE>

ANSI SQL procedures

- Definition: CREATE OR REPLACE PROCEDURE <name> (in <name + type>, out <name + type>)

- Output automatically created as local variable

- Rest is the same: BEGIN… END;

- ‘/’ to execute again

- Parameter options: IN, OUT, INOUT

PostgreSQL stored procedures

- If you want to end a procedure early, you can use RETURN with nothing following it

ANSI/PGSQL procedures: invocation

- Procedures can either be invoked within a trigger, an SQL procedure, or from embedded SQL, using the call statement

- SQL3 allows name overloading for functions and procedures, as long as the number or types of arguments is different

PostgreSQL PL/pgSQL

- Based on ADA  
 - Assignments: direct (:=) and retrieval (INTO)

- Conditional statements: IF-THEN-ELSIF-THEN-ELSE-END IF

- Iterative statements

- Simple loop: LOOP <body> EXIT [or EXIT WHEN] END LOOP