CS 1555

Lecture 17

**Database Programming at Large (continued)**

ANSI exceptions

- Signaling of exception conditions and declaring handlers for exceptions

- declare out\_of\_stock condition

declare exit handler for out\_of\_stock

begin

signal out\_of\_stock

end;

- The handler here is exit – causes enclosing begin..end to be exited

- Other actions possible on exception

Exception handling in PL/SQL

- EXCEPTION clause at the end of a block

- BEGIN

Statements;

EXCEPTION

WHEN condition THEN handler\_statements;

[WHEN OTHERS THEN …]

END;

Three parts to program

- Declare variables

- Program body

- Handle exceptions

- RAISE NOTICE is basically a print statement

PL/SQL: var & const

- DECLARE introduces variables, constants & records

- Variables & constants

- <variable\_name> datatype [NOT NULL := value];

- <constant\_name> CONSTANT datatype := VALUE;

**-** Declaration of variables/constants based on column from database table

- <variable\_name> table\_name column\_name%type;

- Ex: y student.SID%type;

PL/SQL: records

- Record type: TYPE <record\_type\_name> IS RECORD (<1st\_col\_name> datatype, …);

- Declare fields based on a column from database table: col\_name table\_name.column\_name%type;

- Record variable declaration

- User defined: record\_name record\_type\_name;

- Database defined: record\_name table\_name%ROWTYPE;

PL/pgSQL: “record”

- Rec\_name RECORD;

- Has no predefined structure

- Substructure is set when it is assigned a value

Cursors: multiple tuple retrieval

- If more than one tuples are selected, then tuples must be processed one at a time by means of a cursor

- A cursor is a “pointer” to a tuple in the result of a query

- Current tuple with respect to a cursor is the tuple pointed to by the cursor

PL/pgSQL cursors

- <cursor\_name> CURSOR {IS | FOR} <query>

- Declares a cursor by defining a query to be associated with a cursor with it

- OPEN <cursor\_name> brings the query result from the database and positions the cursor before the first tuple

- Before a cursor can be used, it must be opened

- CLOSE <cursor\_name> closes the named cursor and deletes the associated result table

- FETCH copies into variables the current tuple and advances the cursor

- FETCH [direction {FROM | IN}] cursor INTO target;

- Special variable FOUND will be set to true if a row is returned from the fetch

- MOVE repositions cursor without retrieving any data

Implicit cursor in Postgres

- FOR reservation\_record IN SELECT \* FROM reservation;

External language functions/procedures

- Suppose that increment function was implemented in funcs.c and compiled

- CREATE FUNCTION add\_one(integer) RETURNS integer

AS ‘DIRECTORY/funcs’, ‘add\_one’

LANGUAGE C STRICT;

- Then add\_one can be called as usual

External routines: performance vs security

- Benefits of external language functions/procedures

- More efficient

- More expressive power

- Drawbacks

- Code to implement function may need to be executed in database system’s address space

- Risk accidental corruption of database structures

- Security risk

- Use sandbox techniques (a safe language like Java)

- Direct execution in database system’s space is used when efficiency is more important than security

Embedded SQL (ESQL)

- SQL statements are embedded by enclosing them:

- between “&SQL(“ and “)”;

- between “EXECSQL” and “END-EXEC”

- between “EXECSQL” and “;”

- Two types of statement-level embedding

- Static SQL: complete SQL statements

- Dynamic SQL: statements are created during execution

Host data structure

- Program variables used within an SQL command are declared within a DECLARE SECTION

- Host structures/records (i.e. C struct) must match tuple formats exactly

- Field order is important

- Strings in C/C++ are terminated with NULL character

Scope rules

- Arguments used in an SQL statement could be constants or program variables

- Program variables within an SQL command are prefixed with “:”

Status of execution

- SQLCODE represents status of SQL command

- 0 if command was successful

- Positive if warning was generated

- Negative if command failed