PL/SQL

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

© Janusz R. Getta

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

© Janusz R. Getta

© Janusz R. Getta

25. PL/SQL

Overview

PL/SQL =
Data Manipulation statements of SQL +
SELECT statement +
variables +
assignment statement +
conditional control statements +
repetition statement +
exception handling +
procedure and function statements +
packages

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

Program structure

PL/SQL is a block-structured language

It means that its basic units such as anonymous blocks, procedures, and functions are the logical blocks

Logical blocks can be nested to any level

Logical blocks consist of declarative, executable, and exception components

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

Declarative components

Declarative components contain declarations of variables, constants, cursors, procedures, and functions

DECLARE

stock_num NUMBER(5);

stock_name VARCHAR(30);

stock_date DATE;

limit CONSTANT NUMBER(11,2) := 2.45;

CURSOR Q IS
SELECT s# FROM Student WHERE name = 'Jo';

Executable components

Declarative components assignment statements, conditional control statements, iterative statements, procedure and function calls, SQL statements

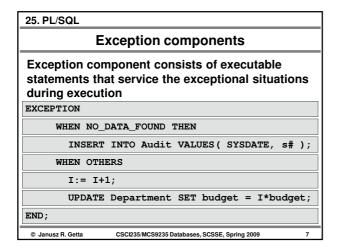
student_name VARCHAR(40);

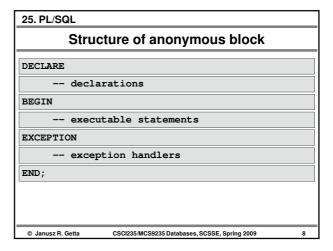
BEGIN

Student_num := 910000;

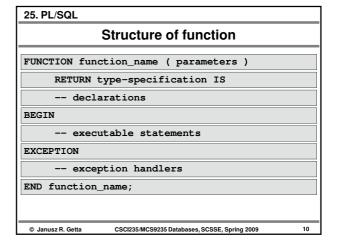
SELECT name INTO student_name
FROM Student
WHERE s# = student_num;

IF a > b THEN a:= a+1 ELSE b:= b+1 END IF;

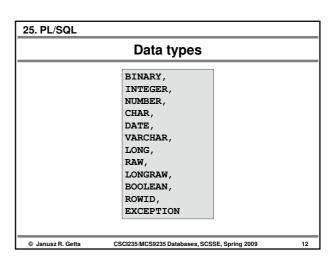


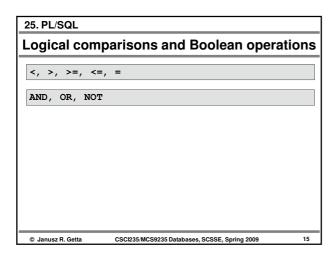


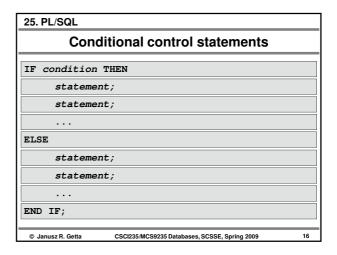
25. PL/SQL	
Structure of procedure	
PROCEDURE procedure_name (parameters)	
declarations BEGIN	
executable statements	
EXCEPTION	
exception handlers	
END procedure_name;	
© Janusz R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009	9

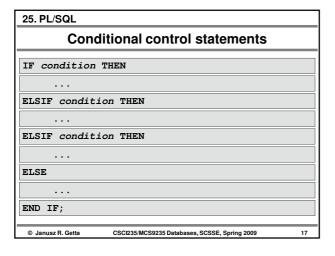


25. PL	/SQL	
	Example of anonymous block	
DECLAR	E	
	average NUMBER(8,2);	
BEGIN		
	SELECT avg(budget)	
	INTO average	
	FROM Department;	
	IF average < 3000 THEN	
	UPDATE Department	
	SET budget = budget+100;	
	END IF	
END;		
© Janus	z R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009 11	





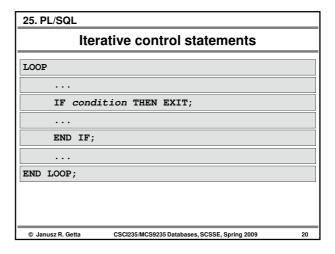


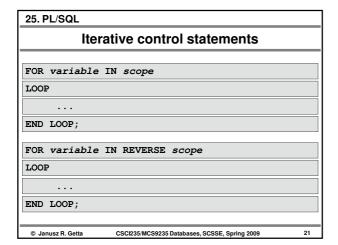


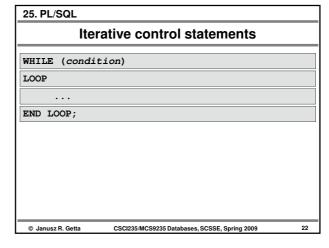
```
CREATE TABLE Department (
name VARCHAR2 (50),
code CHAR(5),
total_staff_number NUMER(2) NOT NULL,
chair VARCHAR(50),
budget NUMER(9,1) NULL,
CONSTRAINT dept_ckey UNIQUE (code),
CONSTRAINT dept_ckey UNIQUE (code),
CONSTRAINT dept_ckey UNIQUE (chairperson),
CONSTRAINT dept_ckey UNIQUE (chairperson),
CONSTRAINT dept_ckey UNIQUE (chairperson),
CONSTRAINT dept_ckey (Chairperson),
CONSTRAINT dept_ckey (NIQUE (chairperson),
CONSTRAINT dept_ckey (NIQUE (chairperson),
CONSTRAINT dept_ckey (NIQUE (chairperson),
CONSTRAINT dept_ckey (NIQUE (chairperson),
CONSTRAINT course Department (NIQUE (CHAIR))
CONSTRAINT course _checkl
CRECK (credits IN (6, 12),
CONSTRAINT course _checkl
CRECK (credits IN (6, 12),
CONSTRAINT course _ckey1 FORBIGN KBY(offered_by)
REFERENCES Department(name));

© Janusz R. Getta CSC1235MCS9235 Databases, SCSSE, Spring 2009 18
```

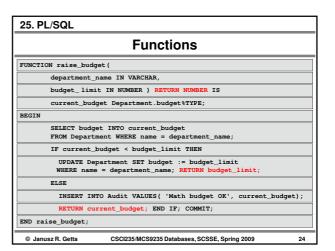
Anonymous block DECLARE current_budget Department.budget%TYPE; budget_limit NUMBER(6) := 400000; BEGIN SELECT budget INTO current_budget FROM Department WHERE name = 'Math'; If current_budget < budget_limit THEN UPDATE Department SET budget := budget_limit WHERE name = 'Math'; ELSE INSERT INTO Audit VALUES('Math budget OK', current_budget); END IF; COMMIT; END; © Janusz R. Getta CSC1235/MCS9235 Databases, SCSSE, Spring 2009 19







25. PL/SQL
Procedures
PROCEDURE raise_budget(
department_name IN VARCHAR,
budget_limit IN NUMBER) IS
current_budget Department.budget%TYPE;
BEGIN
SELECT budget INTO current_budget FROM Department WHERE name = department_name;
IF current_budget < budget_limit THEN
<pre>UPDATE Department SET budget := budget_limit WHERE name = department_name;</pre>
ELSE
INSERT INTO Audit VALUES('Math budget OK', current_budget);
END IF; COMMIT;
END raise_budget;
© Janusz R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009 23



```
CURSORS

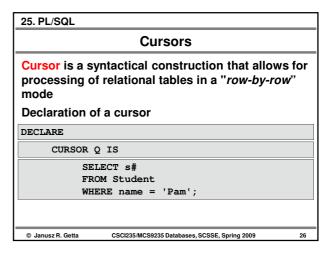
DECLARE

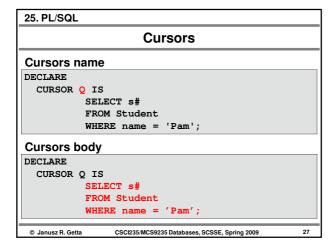
student_no STUDENT.s#%TYPE;

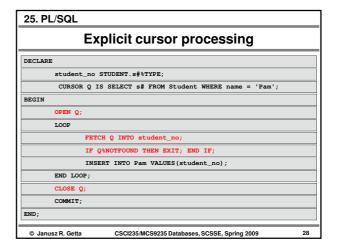
BEGIN

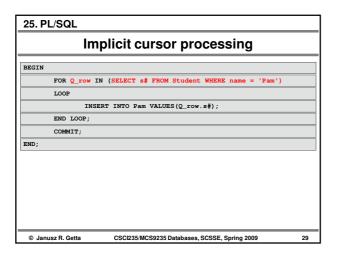
SELECT s#
INTO student_no
FROM Student
WHERE name = 'Pam';
...

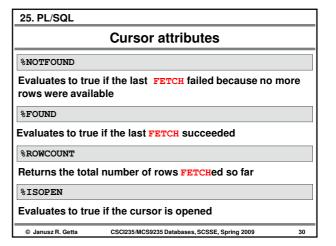
ERROR at line 1:
ORA-06503: PL/SQL: error 0 - Unhandled exception ORA-
01427: single-row subquery returns more than one row
which was raised in a statement ending at line 6
```





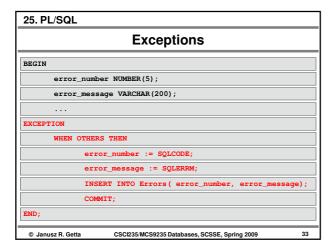


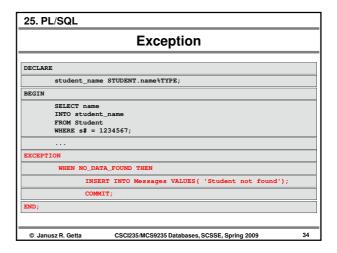




DECLARE student_no STUDENT.s#%TYPE; CURSOR Q IS SELECT s# FROM Student WHERE name = 'Pam'; BEGIN OPEN Q; LOOP FETCH Q INTO student_no; IF Q*NOTFOUND THEN EXIT END IF; INSERT INTO Pam VALUES(student_no); END LOOP; IF Q*ROWCOUNT = 0 THEN INSERT INTO Messages VALUES ('NO ROWS PROCESSED'); END IF; CLOSE Q; COMMIT; END; © Janusz R. Getta CSC1235/MCS9235 Databases, SCSSE, Spring 2009 31

Exceptions Exception is an internally defined or user defined error condition, e.g. divide by zero, no rows selected by SELECT statement with INTO clause, failure of FETCH statement, use of a cursor which has not been opened yet, etc.





25. PL/SQL			
Exceptions			
NO_DATA_FOUND			
Raised when SELECT statement returns no rows			
TOO_MANY_ROWS			
Raised when SELECT statement returns more than one row			
INVALID_CURSOR			
Raised when PL/SQL call specifies an invalid cursor, e.g. closing an unopened cursor			
OTHERS			
Raised when any other exception, not explicitly named happens			
© Janusz R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009	35		

