PL/SQL

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

25. PL/SQL

PL/SQL ? What is it ? Why do we need it ?

PL/SQL is a procedural extension of SQL
PL/SQL = procedural Programming
Language+ SQL

We need PL/SQL to bridge a gap between high level declarative query language and procedural programming language

© Janusz R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009

25. PL/SQL

© Janusz R. Getta

### 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

© Janusz R. Getta

CSCI235/MCS9235 Databases, SCSSE, Spring 2009

25. PL/SQL

### **Program structure**

PL/SQL is a block-structured language

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

Logical blocks can be nested to any level

Logical blocks consist of <u>declarative</u>, <u>executable</u>, and exception components

© Janusz R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009

25. PL/SQL

## **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';

© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009

25. PL/SQL

## **Executable components**

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

student\_name VARCHAR(40);

BEGIN

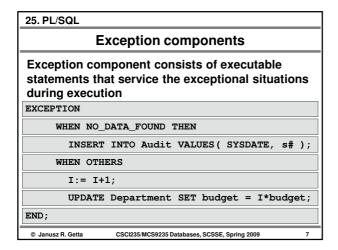
student\_num := 910000;

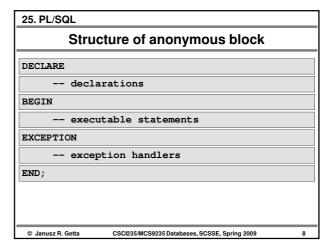
SELECT name INTO student\_name
FROM Student
WHERE s# = student\_num;

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

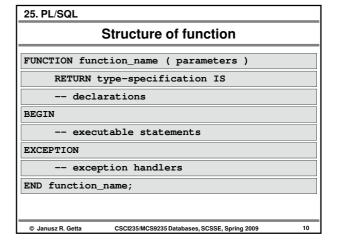
© Janusz R. Getta CSCl235/MCS9235 Databases, SCSSE, Spring 2009

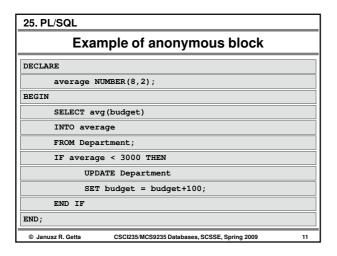
1

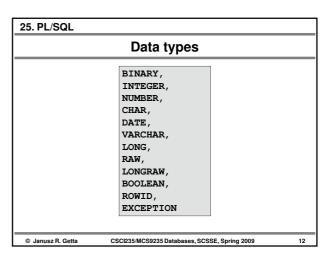


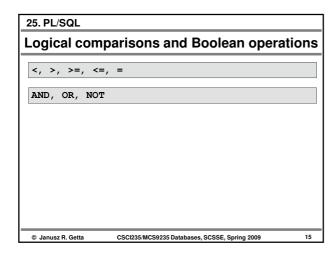


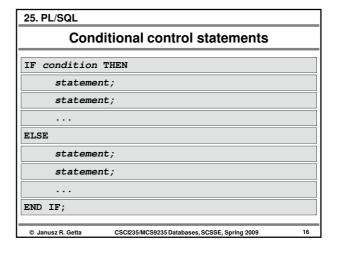
25. PL/SQL	
Structure of procedure	
PROCEDURE procedure_name ( parameters )	
declarations	
BEGIN	
executable statements	
EXCEPTION	
exception handlers	
END procedure_name;	
© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009	9











Conditional control statements

IF condition THEN

...

ELSIF condition THEN

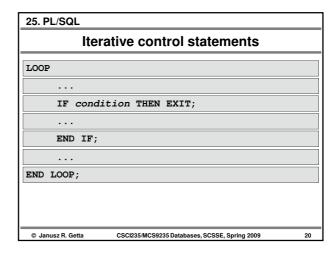
...

Description THEN

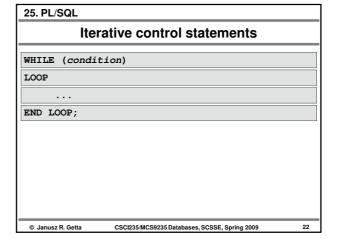
...

END IF;

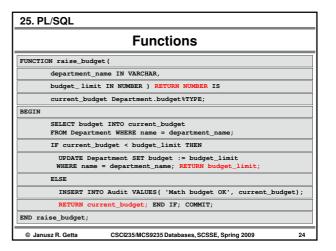
# Anonymous block DECLARE current\_budget Department.budget%TYPE; budget\_limit NUMBER(6) := 400000; BEGIN SELECT budget INTO current\_budget FROM Department WHERE name = 'Math'; If current\_budget < budget\_limit THEN UPDATE Department SET budget := budget\_limit WHERE name = 'Math'; ELSE INSERT INTO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



25. PL/SQL	
Iterative control statements	
FOR variable IN scope	
LOOP	
END LOOP;	
FOR variable IN REVERSE scope	
LOOP	
•••	
END LOOP;	
© Janusz R. Getta CSCI235/MCS9235 Databases, SCSSE, Spring 2009	21



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

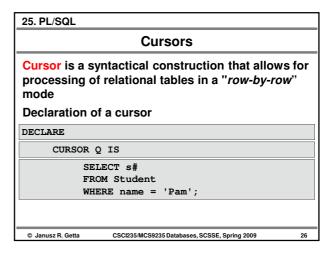
DECLARE

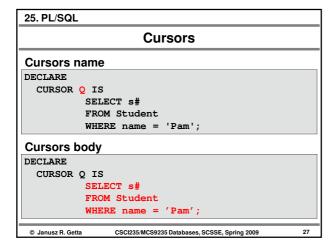
student_no STUDENT.s#%TYPE;

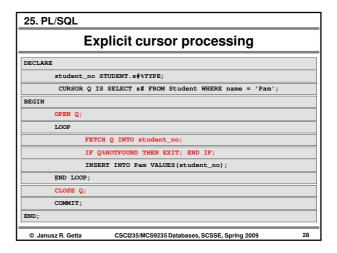
BEGIN

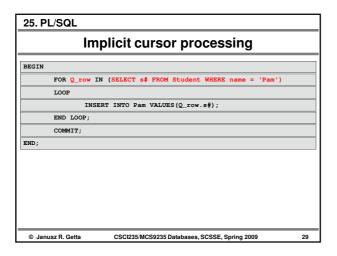
SELECT s#
INTO student_no
FROM Student
WHERE name = 'Pam';
...

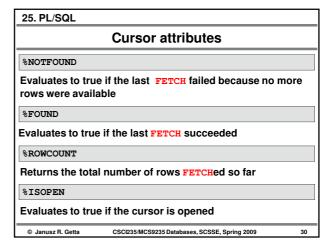
ERROR at line 1:
ORA-06503: PL/SQL: error 0 - Unhandled exception ORA-
01427: single-row subquery returns more than one row
which was raised in a statement ending at line 6
```



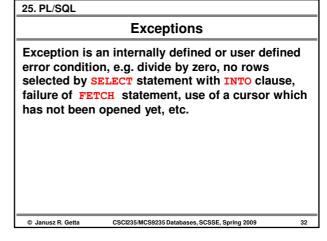


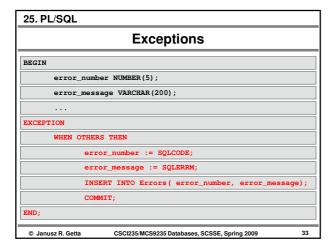


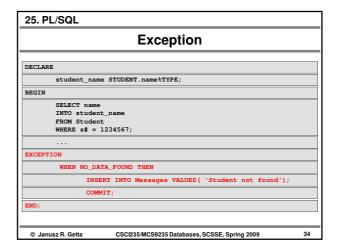




# DECLARE student\_no STUDENT.s#\text{\text{\*\*}} TYPE; CURSOR Q IS SELECT s# FROM Student WHERE name = 'Pam'; BEGIN OPEN Q; LOOP FETCH Q INTO student\_no; IF Q\noticed then exit end if; INSERT INTO Pam VALUES(student\_no); END LOOP; IF Q\text{\text{\*\*}} TROMOTION THEN EXIT END IF; INSERT INTO Pam VALUES(student\_no); END LOOP; IF Q\text{\text{\*\*}} 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







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

