CSCI235 – Database Systems

Stored PL/SQL

sjapit@uow.edu.au

2 October 2021

Acknowledgements

The following presentation were adapted from the lecture slides of:

CSCI235 – Database Systems, 08StoredPLSQL

By Dr Janusz R. Getta, University of Wollongong, Australia

Outline

- Stored PL/SQL. What is it?
- Applications
- CREATE OR REPLACE PROCEDURE statement
- CREATE OR REPLACE FUNCTION statement
- GRANT statement revisited

Stored PL/SQL. What is it?

- **Stored PL/SQL** means **PL/SQL procedures** and **PL/SQL functions** pre-compiled and stored in a **data dictionary** ready to be processed
- **Stored procedures** and **functions** can be referenced or called any number of times by multiple applications processing the relational tables
- **Stored procedures** and **functions** can accept parameters when processed (called)
- **Stored procedures** can be processed (called) with **EXECUTE** statement

Stored PL/SQL. What is it?

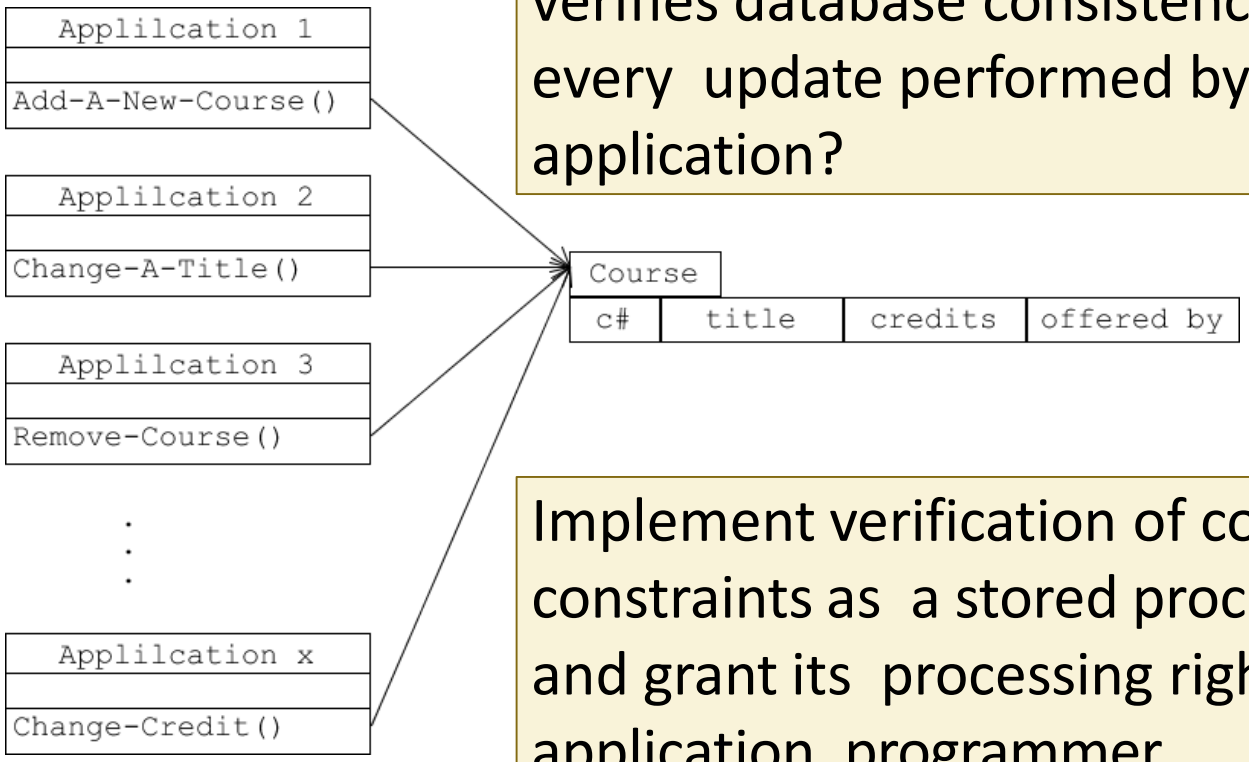
- **Stored functions** can be processed (called) in SQL statement wherever a function can be used, e.g. as row functions in **SELECT** statement
- **Stored procedures** and **stored functions** can be used to extend the functionality of data retrieval and data manipulation statements of SQL **(extensibility)** and to eliminate duplication of code in the database applications **(reuseability)**
- **Stored procedures** and **functions** are created with **CREATE OR REPLACE PROCEDURE** and **CREATE OR REPLACE FUNCTION** SQL statements

Stored PL/SQL

Applications

Applications - reusability

How to avoid duplication of PL/SQL code that verifies database consistency constraints after every update performed by a database application?



Implement verification of consistency constraints as a stored procedure or function and grant its processing rights to a database application programmer.

Applications - extensibility

- Find the names of all departments together with a list of courses offered by each department, display the results in the following form:

Department Name	List of Courses offered
Math	Calculus, Topology, Logic, Algebra
Comp Sci	Python, Java, Databases
Biol	
Phys	Relativity, Mechanics
Astro	Astrology

Applications - extensibility

- Implement a function LCOURSES(dept_name) that returns a list of courses offered by a department whose name is a value of a parameter dept_name.
- Use a function LCOURSES as a row function in SELECT statement

```
SELECT      dname, LCOURSES( dname )
FROM        DEPARTMENT;
```

The function LCOURSES is called for every row retrieved from a relational table DEPARTMENT like any standard row function, e.g., UPPER function.

Stored PL/SQL

CREATE or REPLACE PROCEDURE statement

CREATE or REPLACE PROCEDURE statement

- **CREATE OR REPLACE PROCEDURE** statement compiles and stores PL/SQL **procedure** in a **data dictionary**
- The following **stored procedure INSERT_COURSE** converts the values of string parameters to upper case and inserts a row into a relational table **COURSE**.

```
CREATE OR REPLACE PROCEDURE INSERT_COURSE (
                                cnumber      IN NUMBER,
                                ctitle       VARCHAR2,
                                ccredits     NUMBER,
                                coffer       VARCHAR2) is
```

CREATE or REPLACE PROCEDURE statement

```
BEGIN
        INSERT INTO COURSE VALUES ( cnumber, UPPER(ctitle),
        ccredits, UPPER(coffer) );
        COMMIT;
END INSERT_COURSE;
```

EXECUTE statement is used to process the procedure INSERT_COURSE

EXECUTE INSERT_COURSE(666, 'Java for kids', 6, 'Comp Sci');

Stored PL/SQL

CREATE or REPLACE
FUNCTION statement

CREATE or REPLACE FUNCTION statement

- **CREATE OR REPLACE FUNCTION** statement compiles and stores PL/SQL **function** in a **data dictionary**
- The following **stored function LCOURSES** lists the names of departments together with the titles of courses offered by each department

```
CREATE OR REPLACE FUNCTION LCOURSES(
                dept_name VARCHAR ) RETURN VARCHAR IS
```

CREATE or REPLACE FUNCTION statement

```
    course_list VARCHAR(300);
BEGIN
course_list = ' ';
FOR course_cur_rec  IN (
    SELECT title
    FROM  COURSE
    WHERE offered_by = dept_name;
```

CREATE or REPLACE FUNCTION statement

```
    LOOP
        Course_list := course_list || course_cur_rec.title || ' ';
    END LOOP;
    RETURN course_list;
END LCOURSESD;
```

The stored function LCOURSES is called as a row function in SELECT statement

```
SELECT      dname, LCOURSES( dname )
FROM        COURSE;
```

Stored PL/SQL

GRANT statement revisited

GRANT statement revisited

- In addition to **read** and **write** access rights it is possible to grant **EXECUTE** rights on **stored procedures** and **functions**
- For example, a user **scott** grants execution rights on **INSERT_COURSE** to a user **janusz**

```
GRANT EXECUTE ON INSERT_COURSE TO janusz;
```

- Now, the user janusz executes a stored procedure INSERT_COURSE.

References

- Database PL/SQL Language Reference
- Database SQL Language Reference, CREATE PROCEDURE
- Database SQL Language Reference, CREATE FUNCTION
- Database SQL Language Reference, GRANT
- T. Connoly, C. Begg, Database Systems, A Practical Approach to Design, Implementation, and Management, Chapter 8 Advanced SQL, Pearson Education Ltd, 2015