

Bases de Datos I

Procedimientos y Funciones Almacenadas

Procedimientos almacenados

```
Create procedure  
nombreparámetro, ....  
As  
    declaración variable, ...  
    Begin try  
        instrucción, ...  
  
    End  
    Begin catch  
        instrucción, ...  
  
    End
```

Procedimiento que procesa una
fila del cursor

Procedimiento que recorre un
cursor

Programa principal

Procedimientos almacenados

- CREATE PROCEDURE remove_emp (pemployee_id NUMBER)
AS
- BEGIN
- DELETE FROM employees
- WHERE employees.employee_id =pemployee_id;
- END;



Secuencias

- CREATE SEQUENCE s_employee
- START WITH 0
- INCREMENT BY 1
- MINVALUE 0
- MAXVALUE 100000000
- NOCACHE
- NOCYCLE;

Procedimientos almacenados

- CREATE OR REPLACE PROCEDURE remove_emp (employee_id NUMBER) AS
- tot_emps NUMBER;
- BEGIN
- DELETE FROM employees
- WHERE employees.employee_id = remove_emp.employee_id;
- tot_emps := tot_emps - 1;
- END;

SQL inmerso

- SQL puede utilizarse al interior de un lenguaje de programación de alto nivel como C++, Java, otro
- La respuesta a una consulta realizada desde un programa de aplicación es almacenada por el SABD en lo que se conoce como un cursor (una tabla temporal). Además es un puntero que se posiciona en una tupla del conjunto
- Un cursor sirve para hacer el traslado de un modo conjuntista como es el caso en SQL, a un modo de procedimiento como es el caso de un lenguaje de programación

Cursor

- Es un objeto en memoria RAM que contiene las filas producto de ejecutar una consulta a la base de datos dentro de un programa de aplicación
- El cursor puede ser recorrido una fila a la vez para procesarlas

Cursor

- CURSOR c2 (subject_id_in IN varchar2)
- IS
- SELECT course_number
- from courses_tbl
- where subject_id = subject_id_in;

Cursores

- CURSOR c1
- IS
- SELECT course_number
- from courses_tbl
- where course_name = name_in;

Cursor

- OPEN: Abre el cursor
- FETCH: Permite recuperar una fila para que sea utilizada por una aplicación
- CLOSE: Cierra el cursor

Atributos de los cursores

- <http://www.techonthenet.com/oracle/cursors/attributes.php>

Funciones

- CREATE [OR REPLACE] FUNCTION function_name [(parameter [,parameter])]
- RETURN return_datatype IS | AS [declaration_section]
- BEGIN
- executable_section
- [EXCEPTION
- exception_section]
- END [function_name];

Funciones

- CREATE FUNCTION get_bal(acc_no IN NUMBER)
- RETURN NUMBER
- IS acc_bal NUMBER(11,2);
- BEGIN
- SELECT order_total
- INTO acc_bal
- FROM orders
- WHERE customer_id = acc_no;
- RETURN(acc_bal);
- END;

Funciones

- CREATE OR REPLACE Function FindCourse
- (name_in IN varchar2)
- RETURN number
- IS
- cnumber number;
-
- cursor c1 is
- select course_number
- from courses_tbl
- where course_name = name_in;
-
- BEGIN
-
- open c1;
- fetch c1 into cnumber;
-
- if c1%notfound then
- cnumber := 9999;
- end if;
-
- close c1;
-
- RETURN cnumber;
-
- EXCEPTION
- WHEN OTHERS THEN
- raise_application_error(-20001,'An error was encountered - '||SQLCODE||' -ERROR- '||SQLERRM);
- END;

Packages

- CREATE OR REPLACE PACKAGE emp_mgmt IS
- FUNCTION hire (last_name VARCHAR2, job_id VARCHAR2,
- manager_id NUMBER, salary NUMBER,
- commission_pct NUMBER, department_id NUMBER)
- RETURN NUMBER;
- FUNCTION create_dept(department_id NUMBER, location_id NUMBER)
- RETURN NUMBER;
- PROCEDURE remove_emp(employee_id NUMBER);
- PROCEDURE remove_dept(department_id NUMBER);
- PROCEDURE increase_sal(employee_id NUMBER, salary_incr NUMBER);
- PROCEDURE increase_comm(employee_id NUMBER, comm_incr NUMBER);
- END emp_mgmt;

Package Body

- CREATE OR REPLACE PACKAGE BODY emp_mgmt AS
- tot_emps NUMBER;
- tot_depts NUMBER;
- FUNCTION hire
- (last_name VARCHAR2, job_id VARCHAR2,
- manager_id NUMBER, salary NUMBER,
- commission_pct NUMBER, department_id NUMBER)
- RETURN NUMBER IS new_empno NUMBER;
- BEGIN
- SELECT employees_seq.NEXTVAL
- INTO new_empno
- FROM DUAL;
- INSERT INTO employees
- VALUES (new_empno, 'First', 'Last', 'first.last@oracle.com',
- '(123)123-1234', '18-JUN-02', 'IT_PROG', 90000000, 00,
- 100, 110);
- tot_emps := tot_emps + 1;
- RETURN(new_empno);
- END;
- FUNCTION create_dept(department_id NUMBER, location_id NUMBER)
- RETURN NUMBER IS
- new_deptno NUMBER;
- BEGIN
- SELECT departments_seq.NEXTVAL
- INTO new_deptno
- FROM dual;
- INSERT INTO departments
- VALUES (new_deptno, 'department name', 100, 1700);
- tot_depts := tot_depts + 1;
- RETURN(new_deptno);
- END; ...
- END emp_mgmt;

Package Body

- CREATE OR REPLACE PACKAGE BODY emp_mgmt AS
- tot_emps NUMBER;
- tot_depts NUMBER;
- FUNCTION hire
- (last_name VARCHAR2, job_id VARCHAR2,
- manager_id NUMBER, salary NUMBER,
- commission_pct NUMBER, department_id NUMBER)
- RETURN NUMBER IS new_empno NUMBER;
- BEGIN
- SELECT employees_seq.NEXTVAL
- INTO new_empno
- FROM DUAL;
- INSERT INTO employees
- VALUES (new_empno, 'First', 'Last', 'first.last@oracle.com',
- '123123-1234', '18-JUN-02', 'IT_PROG', 90000000, 00,
- 100, 110);
- tot_emps := tot_emps + 1;
- RETURN(new_empno);
- END;
- FUNCTION create_dept(department_id NUMBER, location_id NUMBER)
- RETURN NUMBER IS
- new_deptno NUMBER;
- BEGIN
- SELECT departments_seq.NEXTVAL
- INTO new_deptno
- FROM dual;
- INSERT INTO departments
- VALUES (new_deptno, 'department name', 100, 1700);
- tot_depts := tot_depts + 1;
- RETURN(new_deptno);
- END;

Excepciones

- Command> DECLARE
- > v_lname VARCHAR2 (15);
- > BEGIN
- > SELECT last_name INTO v_lname
- > FROM employees
- > WHERE first_name = 'John';
- > DBMS_OUTPUT.PUT_LINE ('Last name is : ' || v_lname);
- > END;
- > /
- 8507: ORA-01422: exact fetch returns more than requested number of rows
- 8507: ORA-06512: at line 4
- The command failed.

Excepciones

- Command> DECLARE
- > v_lname VARCHAR2 (15);
- > BEGIN
- > SELECT last_name INTO v_lname
- > FROM employees
- > WHERE first_name = 'John';
- > DBMS_OUTPUT.PUT_LINE ('Last name is :' || v_lname);
- > EXCEPTION
- > WHEN TOO_MANY_ROWS THEN
- > DBMS_OUTPUT.PUT_LINE (' Your SELECT statement retrieved multiple
- > rows. Consider using a cursor.');
- > END;
- > /
- Your SELECT statement retrieved multiple rows. Consider using a cursor.
-
- PL/SQL procedure successfully completed.

Excepciones predeterminadas

- http://docs.oracle.com/cd/E13085_01/doc/timesten.1121/e13076/exceptions.htm
- Command> DECLARE v_invalid PLS_INTEGER;
- > BEGIN
- > v_invalid := 100/0;
- > EXCEPTION
- > WHEN ZERO_DIVIDE THEN
- > DBMS_OUTPUT.PUT_LINE ('Attempt to divide by 0');
- > END;
- > /
- Attempt to divide by 0
-
- PL/SQL procedure successfully completed.

Excepciones definidas por el programador

- Command> DECLARE
- > v_deptno NUMBER := 500;
- > v_name VARCHAR2 (20) := 'Testing';
- > e_invalid_dept EXCEPTION;
- > BEGIN
- > UPDATE departments
- > SET department_name = v_name
- > WHERE department_id = v_deptno;
- > IF SQL%NOTFOUND THEN
- > RAISE e_invalid_dept;
- > END IF;
- > ROLLBACK;
- > EXCEPTION
- > WHEN e_invalid_dept THEN
- > DBMS_OUTPUT.PUT_LINE ('No such department');
- > DBMS_OUTPUT.PUT_LINE (SQLERRM);
- > DBMS_OUTPUT.PUT_LINE (SQLCODE);
- > END;
- > /
- No such department
- User-Defined Exception
- 1
- PL/SQL procedure successfully completed.

Ejercicio

- Ejercicio creación de paquetes y test
- Debug

Tablas temporales

- Son más rápidas que las tablas permanentes
- Solo existen en la PGA de la sesión que está utilizándola
- El DML contra tablas temporales no genera Redo
- Puesto que los datos solo persisten por la duración de una sesión (quizás solamente para la duración de una transacción), no hay un propósito en la generación redo

Tablas temporales

- CREATE GLOBAL TEMPORARY TABLE temp_tab_name
- (column datatype [,column datatypeâ€¦])
- [ON COMMIT {DELETE | PRESERVE} ROWS] ;

Tablas temporales

- Laboratorio – Tablas temporales



Triggers

- CREATE TRIGGER schema.trigger_name
- BEFORE
- DELETE OR INSERT OR UPDATE
- ON schema.table_name
- BEGIN
- pl/sql_block;
- END;

Variables de ambiente

- ORACLE_HOME donde se indica el directorio base del usuario oracle (es el directorio home)
- ORACLE_SID nombre de la instancia de base de datos

Gracias - ¿Preguntas?