PL/SQL Exceptional handling

DECLARE

s\_regno student.regno%type := &regno;

s\_name student.name%type;

s\_addr student.addr%type;

--user defined exception

ex\_invalid\_regno EXCEPTION;

BEGIN

IF s\_regno <=0 THEN

RAISE ex\_invalid\_regno;

ELSE

SELECT regno,name,addr into s\_regno,s\_name,s\_addr from student where regno = s\_regno;

dbms\_output.put\_line(s\_regno||' '||s\_name||' '||s\_addr);

END IF;

EXCEPTION

WHEN ex\_invalid\_regno THEN

dbms\_output.put\_line('regno must be greater than zero!');

WHEN no\_data\_found THEN

dbms\_output.put\_line('no such student!');

WHEN others THEN

dbms\_output.put\_line('error!');

END;

PL/SQL Cursor

DECLARE

cursor all\_emps is select empno,ename from emp1 order by empno;

emp1 all\_emps%rowtype;

begin

open all\_emps;

loop

exit when all\_emps%notfound;

fetch all\_emps into emp1;

dbms\_output.put\_line(emp1.ename||' '||emp1.empno);

end loop;

close all\_emps;

end;

table creation:

create table emp1(empno number(4), Ename varchar(10), addr varchar(20));

PL/SQL Trigger

CREATE OR REPLACE TRIGGER show\_salary\_difference

BEFORE DELETE OR INSERT OR UPDATE ON employees

FOR EACH ROW

WHEN (NEW.EMP\_ID>0)

DECLARE

sal\_diff number;

BEGIN

sal\_diff:=:NEW.salary -:OLD.salary;

dbms\_output.put\_line('old salary:'||:OLD.salary);

dbms\_output.put\_line('New salary:'||:NEW.salary);

dbms\_output.put\_line('Salary difference:'|| sal\_diff);

END;

DECLARE

total\_rows number(2);

BEGIN

UPDATE employees

SET salary = salary + 5000;

if sql%notfound THEN

dbms\_output.put\_line('no employees updated');

ELSIF sql%found THEN

total\_rows :=sql%rowcount;

dbms\_output.put\_line(total\_rows || 'employees updated');

END IF;

END;

PL/SQL Packages

CREATE PACKAGE cust\_sal AS

PROCEDURE find\_sal(c\_id customer.id%type);

END

cust\_sal;

package created.

CREATE OR REPLACE PACKAGE BODY cust\_sal AS

PROCEDURE find\_sal(c\_id customer.id%type)IS c\_sal

customer.salary%type;

BEGIN

SELECT salary INTO c\_sal

FROM customer WHERE id=c\_id;

dbms\_output.put\_line('Salary:'||c\_sal);

END find\_sal;

END cust\_sal;

Package Body created

DECLARE

code customer.id%type :=&cc\_id;

BEGIN

cust\_sal.find\_sal(code);

end;