SET SERVEROUTPUT ON;

DBMS\_OUTPUT.NEW\_LINE();

DBMS\_OUTPUT.PUT\_LINE(‘saf’);

CHR(10) : ASCII code 10 (enter)

TO\_CHAR(v\_num, ‘9999.99’);

VARIABLE g\_var VARCHAR2(1000) : Global variable

PRINT g\_var

DECLARE

CURSOR var\_cur IS

SELECT ……

BEGIN

OPEN var\_cur;

LOOP

FETCH var\_cur INTO col1, col2;

EXIT WHEN var\_cur%NOTFOUND;

END LOOP;

CLOSE var\_cur;

END;

/

v\_value%FOUND : true, a row exists

v\_value%NOTFOUND; : true, no more row

v\_value%ISOPEN; : always return false for implicit cursors

v\_value%ROWCOUNT; : number of row

BEGIN

FOR v\_row IN var\_cur LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_row.col1 || ‘, ‘ || v\_row.col2);

END LOOP;

END;

/

IF (expression) THEN

ELSIF (expression) THEN

ELSE

END IF

CASE var

WHEN ‘value1’ THEN

Do something;

WHEN ‘value2’ THEN

Do something;

ELSE

Do something else;

END CASE;

DECLARE

exp\_toomany EXCEPTION;

BEGIN

…

…

RAISE exp\_toomany; throw a exception named “exp\_toomany”

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

Do something;

WHEN exp\_toomany THEN

Do something;

WHEN OTHERS THEN

ROLLBACK WORK;

END;

/

SQLCODE : it has an error code. It starts from 0 and when an error occurs it gets error code

SQLERRM(SQLCODE) : show error message

ex\_noname EXCEPTION;

PRAGMA EXCEPTION\_INIT(ex\_noname, -1311(error\_code)); initialize personal exception;

CREATE OR REPLACE PROCEDURE procedure1 (

param\_id VARCHAR2 := ‘0’,

param\_name VARCHAR2

param\_value1 IN // can be a value

param\_value2 OUT // must be a variable

param\_value3 IN OUT // must be a variable

)

AS

BEGIN

DMBS\_OUTPUT.PUT\_LINE(pl\_id || ‘ ‘ || pl\_name);

END;

/

CALL procedure1(‘myId’, pl\_name => ‘kim’);

CREATE OR REPLACE FUNCTION getTotalDiscounts(

param\_date IN VARCHAR2

)

RETURN NUMBER

AS

v\_total NUMBER;

BEGIN

SELECT SUM(discount)

INTO v\_total

FROM saleInv

WHERE saleDate > param\_date;

RETURN v\_total;

END;

/