**COMMENTS:(SQL PLUS)**

SQL> SET SERVEROUTPUT ON;

USE / TO END

Sqlplus / as sysdba

ALTER USER system identified by system;

lsnrctl start

lsnrctl status

**VARIABLES SAMPLE**

Example of initilizing variable

1. **DECLARE**
2. a **integer** := 30;
3. b **integer** := 40;
4. c **integer**;
5. f **real**;
6. **BEGIN**
7. c := a + b;
8. dbms\_output.put\_line('Value of c: ' || c);
9. f := 100.0/3.0;
10. dbms\_output.put\_line('Value of f: ' || f);
11. **END**;

**CONSTANT**

DECLARE

co\_pi CONSTANT REAL := 3.14159;

co\_radius CONSTANT REAL := 10;

co\_area CONSTANT REAL := (co\_pi \* co\_radius\*\*2);

BEGIN

DBMS\_OUTPUT.PUT\_LINE(co\_area);

END;

# **IF Statement**

DECLARE

n\_sales NUMBER := 2000000;

BEGIN

IF n\_sales > 100000 THEN

DBMS\_OUTPUT.PUT\_LINE( 'Sales revenue is greater than 100K ' );

END IF;

END;

DECLARE

A INTEGER :=25;

BEGIN

IF A>200 THEN

DBMS\_OUTPUT.put\_line('CONDITION CHECKED');

ELSIF A>200 THEN

DBMS\_OUTPUT.put\_line('2.CONDITION CHECKED');

ELSE

DBMS\_OUTPUT.PUT\_LINE(A);

END IF;

END;

# **GOTO Statement**

BEGIN

GOTO get\_name;

<<get\_id>>

DBMS\_OUTPUT.PUT\_LINE('Poor');

GOTO the\_end;

<<get\_name>>

DBMS\_OUTPUT.PUT\_LINE('Poor123');

GOTO get\_id;

<<the\_end>>

DBMS\_OUTPUT.PUT\_LINE('Poor123');

END;

# **NULL Statement**

DECLARE

n\_credit\_status VARCHAR2( 50 );

BEGIN

n\_credit\_status := 'BLOCK';

CASE n\_credit\_status

WHEN 'BLOCK' THEN

DBMS\_OUTPUT.PUT\_LINE('Poor123@BLOCK');

WHEN 'WARNING' THEN

DBMS\_OUTPUT.PUT\_LINE('Poor123@WARNING');

ELSE

NULL;

END CASE;

END;

## Iterative processing with loops

## DECLARE

## l\_counter NUMBER := 0;

## BEGIN

## LOOP

## l\_counter := l\_counter + 1;

## IF l\_counter > 3 THEN

## EXIT;

## END IF;

## dbms\_output.put\_line( 'Inside loop: ' || l\_counter ) ;

## END LOOP;

## -- control resumes here after EXIT

## dbms\_output.put\_line( 'After loop: ' || l\_counter );

## END;

## FOR LOOP

DECLARE

l\_step PLS\_INTEGER := 2;

BEGIN

FOR l\_counter IN 1..5 LOOP

dbms\_output.put\_line ('counter current value' || l\_counter);

dbms\_output.put\_line ('total value' || l\_counter\*l\_step);

END LOOP;

END;

BEGIN

FOR l\_counter IN REVERSE 1..3

LOOP

DBMS\_OUTPUT.PUT\_LINE( l\_counter );

END LOOP;

END;

# **WHILE Loop**

DECLARE

n\_counter NUMBER := 1;

BEGIN

WHILE n\_counter <= 5

LOOP

DBMS\_OUTPUT.PUT\_LINE( 'Counter : ' || n\_counter );

n\_counter := n\_counter + 1;

END LOOP;

END;

# **CONTINUE**

BEGIN

FOR n\_index IN 1 .. 10

LOOP

-- skip odd numbers

IF MOD( n\_index, 2 ) = 1 THEN

CONTINUE;

END IF;

DBMS\_OUTPUT.PUT\_LINE( n\_index );

END LOOP;

END;

## Select Into

## PL/SQL SELECT INTO example

-- customers

CREATE TABLE customers

(

customer\_id NUMBER

GENERATED BY DEFAULT AS IDENTITY START WITH 320

PRIMARY KEY,

name VARCHAR2( 255 ) NOT NULL,

address VARCHAR2( 255 ) ,

website VARCHAR2( 255 ) ,

credit\_limit NUMBER( 8, 2 )

);

-- contacts

CREATE TABLE contacts

(

contact\_id NUMBER

GENERATED BY DEFAULT AS IDENTITY START WITH 320

PRIMARY KEY,

first\_name VARCHAR2( 255 ) NOT NULL,

last\_name VARCHAR2( 255 ) NOT NULL,

email VARCHAR2( 255 ) NOT NULL,

phone VARCHAR2( 20 ) ,

customer\_id NUMBER ,

CONSTRAINT fk\_contacts\_customers

FOREIGN KEY( customer\_id )

REFERENCES customers( customer\_id )

ON DELETE CASCADE

);

SELECT \* FROM customers;

SELECT \* FROM contacts;

Insert into CUSTOMERS (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (177,'United Continental Holdings','2904 S Salina St, Syracuse, NY',5000,'http://www.unitedcontinentalholdings.com');

Insert into CUSTOMERS (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (180,'INTL FCStone','5344 Haverford Ave, Philadelphia, PA',5000,'http://www.intlfcstone.com');

Insert into CUSTOMERS (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (184,'Publix Super Markets','1795 Wu Meng, Muang Chonburi, ',1200,'http://www.publix.com');

Insert into CUSTOMERS (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (187,'ConocoPhillips','Walpurgisstr 69, Munich, ',2400,'http://www.conocophillips.com');

Insert into CUSTOMERS (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (190,'3M','Via Frenzy 6903, Roma, ',1200,'http://www.3m.com');

Insert into CUSTOMERS (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (192,'Exelon','Via Luminosa 162, Firenze, ',500,'http://www.exeloncorp.com');

Insert into CONTACTS (CONTACT\_ID,FIRST\_NAME,LAST\_NAME,EMAIL,PHONE,CUSTOMER\_ID) values (1,'Flor','Stone','flor.stone@raytheon.com','+1 317 123 4104',177);

Insert into CONTACTS (CONTACT\_ID,FIRST\_NAME,LAST\_NAME,EMAIL,PHONE,CUSTOMER\_ID) values (2,'Lavera','Emerson','lavera.emerson@plainsallamerican.com','+1 317 123 4111',180);

Insert into CONTACTS (CONTACT\_ID,FIRST\_NAME,LAST\_NAME,EMAIL,PHONE,CUSTOMER\_ID) values (3,'Fern','Head','fern.head@usfoods.com','+1 812 123 4115',184);

Insert into CONTACTS (CONTACT\_ID,FIRST\_NAME,LAST\_NAME,EMAIL,PHONE,CUSTOMER\_ID) values (4,'Shyla','Ortiz','shyla.ortiz@abbvie.com','+1 317 123 4126',187);

Insert into CONTACTS (CONTACT\_ID,FIRST\_NAME,LAST\_NAME,EMAIL,PHONE,CUSTOMER\_ID) values (5,'Jeni','Levy','jeni.levy@centene.com','+1 812 123 4129',190);

Insert into CONTACTS (CONTACT\_ID,FIRST\_NAME,LAST\_NAME,EMAIL,PHONE,CUSTOMER\_ID) values (6,'Matthias','Hannah','matthias.hannah@chs.net','+1 219 123 4136',192);

---------------------------------------------------------------------------------------------------------

DECLARE

l\_name customers.NAME%TYPE;

l\_customer\_id customers.customer\_id%TYPE := &customer\_id;

BEGIN

-- get the customer name by id

SELECT name INTO l\_name

FROM customers

WHERE customer\_id = l\_customer\_id;

-- show the customer name

dbms\_output.put\_line('Customer name is ' || l\_name);

END;

## PL/SQL Exceptions

DECLARE

l\_name customers.NAME%TYPE;

l\_customer\_id customers.customer\_id%TYPE := &customer\_id;

BEGIN

-- get the customer

SELECT NAME INTO l\_name

FROM customers

WHERE customer\_id = l\_customer\_id;

-- show the customer name

dbms\_output.put\_line('customer name is ' || l\_name);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Customer ' || l\_customer\_id || ' does not exist');

END;

# **Raise Exceptions**

DECLARE

l\_customer\_id customers.customer\_id%TYPE := &customer\_id;

BEGIN

-- get the meax credit limit

IF l\_customer\_id < 0 THEN

RAISE invalid\_number;

END IF;

END;

# **PL/SQL Record**

The following example defines a record type whose name is customer\_contacts and a record whose type is customer\_contacts:

DECLARE

-- define a record type

TYPE r\_customer\_contact\_t

IS

RECORD

(

customer\_name customers.name%TYPE,

first\_name contacts.first\_name%TYPE,

last\_name contacts.last\_name%TYPE );

-- declare a record

r\_customer\_contacts r\_customer\_contact\_t;

BEGIN

NULL;

END;

# **PL/SQL Cursor**