BEGIN

dbms\_output.put\_line('HEY GUYS');

END;

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

DECLARE

c VARCHAR2(3 CHAR);

a number:=10;

b number:=20;

d NUMBER;

BEGIN

c := 'ab';

d := a+b;

dbms\_output.put\_line('Addition' || d);

END;

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

DECLARE --notes

a integer := 30;

b integer := 40; ------geetha

c integer;

f real;

BEGIN

c := a + b;

dbms\_output.put\_line('Value of c: ' || c);

f := 100.0/3.0;

dbms\_output.put\_line('Value of f: ' || f);

END;

/\*

gsdfsrgthgh

\*/

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

DECLARE

a CONSTANT INTEGER :=100;

BEGIN

IF a < 500 THEN

DBMS\_OUTPUT.PUT('condition executed ');

END IF;

dbms\_output.put\_line('Value of a: ' || a);

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;

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

DECLARE

c\_grade CHAR( 1 );

c\_rank VARCHAR2( 20 );

BEGIN

c\_grade := 'B';

CASE c\_grade

WHEN 'A' THEN

c\_rank := 'Excellent' ;

WHEN 'B' THEN

c\_rank := 'Very Good' ;

WHEN 'C' THEN

c\_rank := 'Good' ;

WHEN 'D' THEN

c\_rank := 'Fair' ;

WHEN 'F' THEN

c\_rank := 'Poor' ;

ELSE

c\_rank := 'No such grade' ;

END CASE;

DBMS\_OUTPUT.PUT\_LINE( c\_rank );

END;

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

BEGIN

GOTO get\_name;

<<get\_id>>

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

GOTO the\_end;

<<get\_name>>

DBMS\_OUTPUT.PUT\_LINE('Poor123NAME');

GOTO get\_id;

<<the\_end>>

DBMS\_OUTPUT.PUT\_LINE('Poor123END');

END;

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

DECLARE

n NUMBER;

BEGIN

NULL;

END;

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

DECLARE

n\_credit\_status VARCHAR2( 50 );

BEGIN

n\_credit\_status := 'BLOCK123';

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;

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

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;

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

BEGIN

FOR l\_counter IN 1..5

LOOP

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

END LOOP;

END;

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

BEGIN

FOR l\_counter IN REVERSE 1..3

LOOP

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

END LOOP;

END;

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

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;

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

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;

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

-- customers

CREATE TABLE customers1

(

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 )

);

SELECT \* FROM customers1;

Insert into CUSTOMERS1 (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 CUSTOMERS1 (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (180,'INTL FCStone','5344 Haverford Ave, Philadelphia, PA',5000,'http://www.intlfcstone.com');

Insert into CUSTOMERS1 (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (184,'Publix Super Markets','1795 Wu Meng, Muang Chonburi, ',1200,'http://www.publix.com');

Insert into CUSTOMERS1 (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (187,'ConocoPhillips','Walpurgisstr 69, Munich, ',2400,'http://www.conocophillips.com');

Insert into CUSTOMERS1 (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (190,'3M','Via Frenzy 6903, Roma, ',1200,'http://www.3m.com');

Insert into CUSTOMERS1 (CUSTOMER\_ID,NAME,ADDRESS,CREDIT\_LIMIT,WEBSITE) values (192,'Exelon','Via Luminosa 162, Firenze, ',500,'http://www.exeloncorp.com');

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;

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

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;

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

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;

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

CREATE OR REPLACE PROCEDURE print\_contact(

p\_customer\_id NUMBER

)

IS

r\_contact contacts%ROWTYPE;

BEGIN

-- get contact based on customer id

SELECT \*

INTO r\_contact

FROM contacts

WHERE customer\_id = p\_customer\_id;

-- print out contact's information

dbms\_output.put\_line( r\_contact.first\_name || ' ' ||

r\_contact.last\_name || '<' || r\_contact.email ||'>' );

EXCEPTION

WHEN OTHERS THEN

dbms\_output.put\_line( SQLERRM );

END;

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

CREATE TABLE EVALUATIONS\_LOG ( log\_date DATE

, action VARCHAR2(50));

CREATE OR REPLACE TRIGGER EVAL\_CHANGE\_TRIGGER

AFTER INSERT OR UPDATE OR DELETE

ON EVALUATIONS

DECLARE

log\_action EVALUATIONS\_LOG.action%TYPE;

BEGIN

IF INSERTING THEN

log\_action := 'Insert';

ELSIF UPDATING THEN

log\_action := 'Update';

ELSIF DELETING THEN

log\_action := 'Delete';

ELSE

DBMS\_OUTPUT.PUT\_LINE('This code is not reachable.');

END IF;

INSERT INTO EVALUATIONS\_LOG (log\_date, action)

VALUES (SYSDATE, log\_action);

END;

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

begin

execute immediate 'drop table item\_type';

exception

when others then null;

end;

/

CREATE TABLE item\_type(

id number(19) NOT NULL,

name varchar2(255) NOT NULL,

deposit BINARY\_DOUBLE NOT NULL,

cost\_per\_day BINARY\_DOUBLE NOT NULL,

PRIMARY KEY (id));

INSERT INTO item\_type (id, name, deposit, cost\_per\_day) VALUES (1,'Food', '50000', '10000');

INSERT INTO item\_type (id, name, deposit, cost\_per\_day) VALUES (2,'Books', '20000', '5500');

select \*from employees123;

create table emp(id number(10),name varchar2(40),age number(3));

INSERT INTO emp (id, name,age) VALUES (1,'Geetha',30);

select \*from emp;

select \*from Student;

insert into Student values(101,'adam');

insert into Student values(102,'abhi');

insert into Student values(108,'Geetha');

create table student1(id number(10),name varchar2(40),age number(3));

create table book123(id number(10), name varchar(50),price number(10));

insert into book123(id,name,price) values (111,'java core',999);

insert into book123(id,name,price) values (112,'python core',999);

insert into book123(id,name,price) values (113,'spring core',1999);

select \*from book123;