EXCEPTION HANDLING

SLIDE 7. Predefined Exception – Example

DECLARE

v\_staffno staff\_masters.staff\_code%type;

v\_name staff\_masters.staff\_name%type;

BEGIN

SELECT staff\_name into v\_name FROM staff\_masters

WHERE staff\_code=&v\_staffno;

dbms\_output.put\_line(v\_name);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Not Found');

END;

/

SLIDE 11 - Numbered Exception – Example

DECLARE

v\_bookno number := 10000008;

child\_rec\_found EXCEPTION;

PRAGMA EXCEPTION\_INIT (child\_rec\_found, -2292);

BEGIN

DELETE from book\_masters

WHERE book\_code = v\_bookno;

EXCEPTION

WHEN child\_rec\_found THEN

INSERT into error\_log

VALUES ('Book entries exist for book:' || v\_bookno);

END;

/

SLIDE 17 - User-defined Exception – Example

DECLARE

dup\_deptno EXCEPTION;

v\_counter binary\_integer;

v\_department number(2) := 50;

BEGIN

SELECT count(\*) into v\_counter FROM department\_masters

WHERE dept\_code=50;

IF v\_counter > 0 THEN

RAISE dup\_deptno ;

END IF;

INSERT into department\_masters

VALUES (v\_department ,'new name');

EXCEPTION

WHEN dup\_deptno THEN

INSERT into error\_log

VALUES ('Dept: '|| v\_department ||' already exists');

END ;

/

SLIDE 19 - Others Exception Handler – Example

CREATE TABLE ERRORS

(ERR\_NUM NUMBER(5)

,ERR\_MSG VARCHAR2(100)

)

/

DECLARE

v\_dummy varchar2(1);

v\_designation number(3) := 109;

Err\_Num number(5);

Err\_Msg varchar2(100);

BEGIN

SELECT 'x' into v\_dummy FROM designation\_masters

WHERE design\_code= v\_designation;

INSERT into error\_log

VALUES ('Designation: ' || v\_designation || 'already exists');

EXCEPTION

WHEN no\_data\_found THEN

insert into designation\_masters values (v\_designation,'newdesig');

WHEN OTHERS THEN

Err\_Num := SQLCODE;

Err\_Msg :=SUBSTR( SQLERRM, 1, 100);

INSERT into errors VALUES( err\_num, err\_msg );

END ;

/

NOTE – TO CHECK THIS CODE USING “WHEN OTHERS HANDLER” REMOVE “WHEN NO\_DATA\_FOUND” HANDLER AND ENTER 110 AS V\_DESIGNATION.

SLIDE 28 - Benefits of using Dynamic SQL

CREATE TABLE MY\_TABLE

(VAL1 NUMBER

,VAL2 VARCHAR2(2)

)

/

INSERT INTO MY\_TABLE

VALUES(&VAL1

,'&VAL2'

)

/

BEGIN

EXECUTE IMMEDIATE 'TRUNCATE TABLE my\_table';

END;

/

SLIDE 29 - Single Row Queries

DECLARE

v\_sql VARCHAR2(100);

v\_date DATE;

BEGIN

v\_sql := 'SELECT Sysdate FROM dual';

EXECUTE IMMEDIATE v\_sql INTO v\_date;

DBMS\_OUTPUT.PUT\_LINE(v\_date);

END;

/

* SLIDE 33 - To retrieve values from a Dynamic statement (INTO clause), refer the following example:

DECLARE

l\_cnt varchar2(20);

BEGIN EXECUTE IMMEDIATE 'SELECT count(1) FROM staff\_masters' INTO l\_cnt; dbms\_output.put\_line(l\_cnt);

END;

/

SLIDE 34 - To pass values to a Dynamic statement (USING clause), refer the following example:

DECLARE

l\_depnam varchar2(20) := 'testing';

l\_loc varchar2(10) := 'Dubai';

BEGIN

EXECUTE IMMEDIATE 'INSERT into dept

VALUES (:1, :2, :3)' using 70, l\_depnam,l\_loc;

commit;

END;

/

Dev Special - I made an example for understanding raise

DECLARE   
    exp\_demo EXCEPTION;   
 BEGIN   
    IF 1!=2   
    THEN   
            RAISE exp\_demo;   
    END IF;   
 EXCEPTION   
    WHEN others THEN   
            dbms\_output.put\_line('one cannot be equal to two');   
 END;