



www.itsci.mju.ac.th/sayan

LEC 09: EXCEPTION

SAYAN UNANKARD
1/2558

EXAMPLE OF AN EXCEPTION

```
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname
  FROM employees
  WHERE first_name='John';
  DBMS_OUTPUT.PUT_LINE ('John''s last name is :'
                        || v_lname);
END;
```

EXAMPLE OF AN EXCEPTION

```
DECLARE
  v_lname VARCHAR2(15);
BEGIN
  SELECT last_name INTO v_lname
  FROM employees
  WHERE first_name='John';
  DBMS_OUTPUT.PUT_LINE ('John''s last name is :'
                        || v_lname);

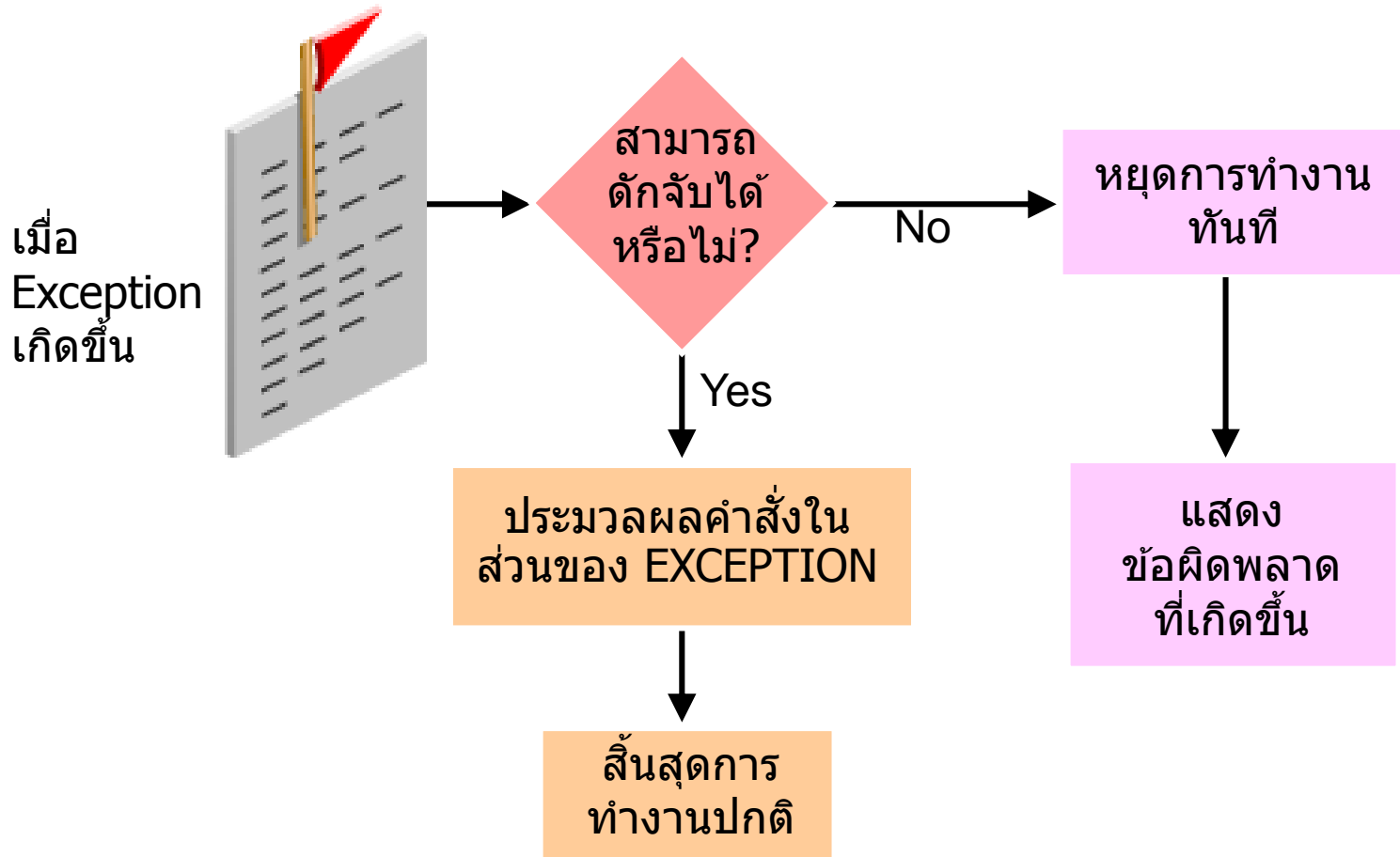
EXCEPTION
  WHEN TOO_MANY_ROWS THEN
    DBMS_OUTPUT.PUT_LINE('Your select statement
    retrieved multiple rows. Consider using a cursor.');
```

END;
/

HANDLING EXCEPTIONS WITH PL/SQL

- การจัดการข้อผิดพลาดที่เกิดจากการประมวลผลโปรแกรมในส่วนของ PL/SQL error
- Exception สามารถเกิดขึ้นจาก
 - เกิดขึ้นจาก Oracle server
 - เกิดขึ้นจากโปรแกรม
- Exception สามารถควบคุมได้โดย
 - ดักจับโดยตัวจัดการ (handler)
 - โดยการส่งข้อผิดพลาดไปยังสภาพแวดล้อมภายนอก

HANDLING EXCEPTIONS



EXCEPTION TYPES

ประเภทของ Exception

- Predefined Oracle server
 - Non-predefined Oracle server
- } Implicitly raised
- User-defined
- Explicitly raised

TRAPPING EXCEPTIONS

Syntax:

EXCEPTION

```
WHEN exception1 [OR exception2 . . .] THEN
    statement1;
    statement2;
    . . .
[WHEN exception3 [OR exception4 . . .] THEN
    statement1;
    statement2;
    . . .]
[WHEN OTHERS THEN
    statement1;
    statement2;
    . . .]
```

GUIDELINES FOR TRAPPING EXCEPTIONS

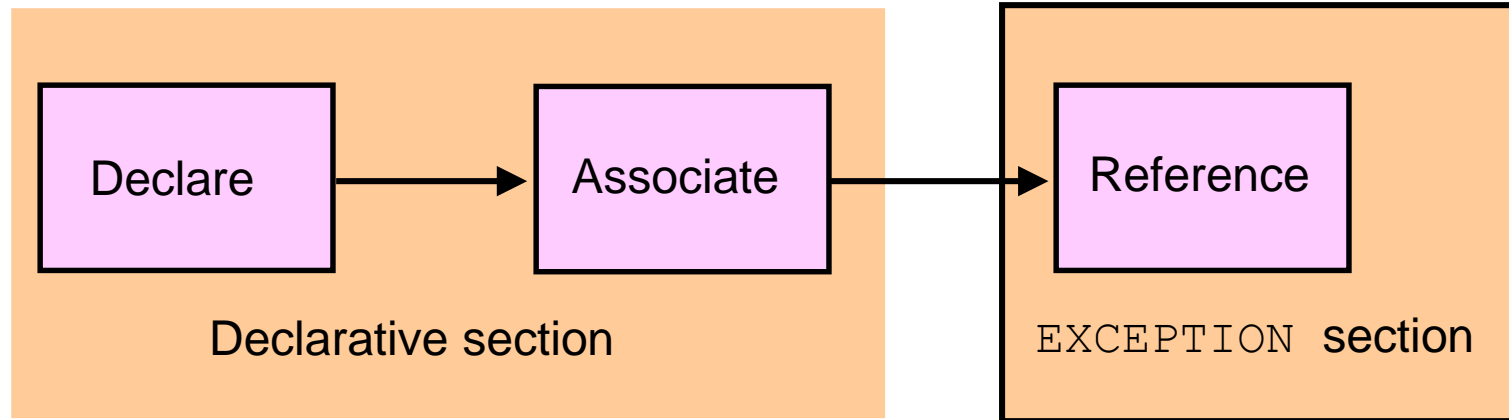
- นำหน้าส่วนของการจัดการข้อผิดพลาดด้วยคีย์เวิร์ดว่า EXCEPTION
- สามารถจัดการข้อผิดพลาดที่เกิดขึ้นได้หลายตัว
- จะมีการทำงานเพียง 1 เงื่อนไขก่อนที่จะสิ้นสุดการทำงาน
- WHEN OTHERS เป็นเงื่อนไขสุดท้ายในการตรวจสอบ หมายถึงกรณีอื่น ๆ นอกเหนือจากที่ระบุไว้

TRAPPING PREDEFINED ORACLE SERVER ERRORS

- การอ้างอิงชื่อ Exception ที่ถูกกำหนดไว้แล้วโดย Oracle Server
- ตัวอย่างของ Predefined Exception
 - NO_DATA_FOUND -ORA-01403
 - TOO_MANY_ROWS - ORA-01422
 - INVALID_CURSOR - ORA-01001
 - ZERO_DIVIDE - ORA-01476
 - DUP_VAL_ON_INDEX - ORA-00001

(attempts to store a duplicate value or values in a database column that is constrained by a unique index.)

TRAPPING NON-PREDEFINED ORACLE SERVER ERRORS



กำหนดชื่อของ
exception

ใช้คำสั่ง PRAGMA
EXCEPTION_INIT
เพื่อกำหนดว่าชื่อของ
exception ที่สร้างขึ้น
จัดการ error รหัสใด

จัดการเมื่อเกิด
เงื่อนไขที่กำหนดไว้

NON-PREDEFINED ERROR

To trap Oracle server error number -01400 (“cannot insert NULL”):

```
DECLARE
  e_insert_excep EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_insert_excep, -01400);
BEGIN
  INSERT INTO departments
    (department_id, department_name) VALUES (280, NULL);
EXCEPTION
  WHEN e_insert_excep THEN
    DBMS_OUTPUT.PUT_LINE('INSERT OPERATION FAILED');
    DBMS_OUTPUT.PUT_LINE(SQLERRM);
END;
/
```

The diagram illustrates the PL/SQL code with three numbered annotations:

- 1**: Points to the exception declaration `e_insert_excep EXCEPTION;`.
- 2**: Points to the PRAGMA statement `PRAGMA EXCEPTION_INIT(e_insert_excep, -01400);`.
- 3**: Points to the exception handler `WHEN e_insert_excep THEN`.

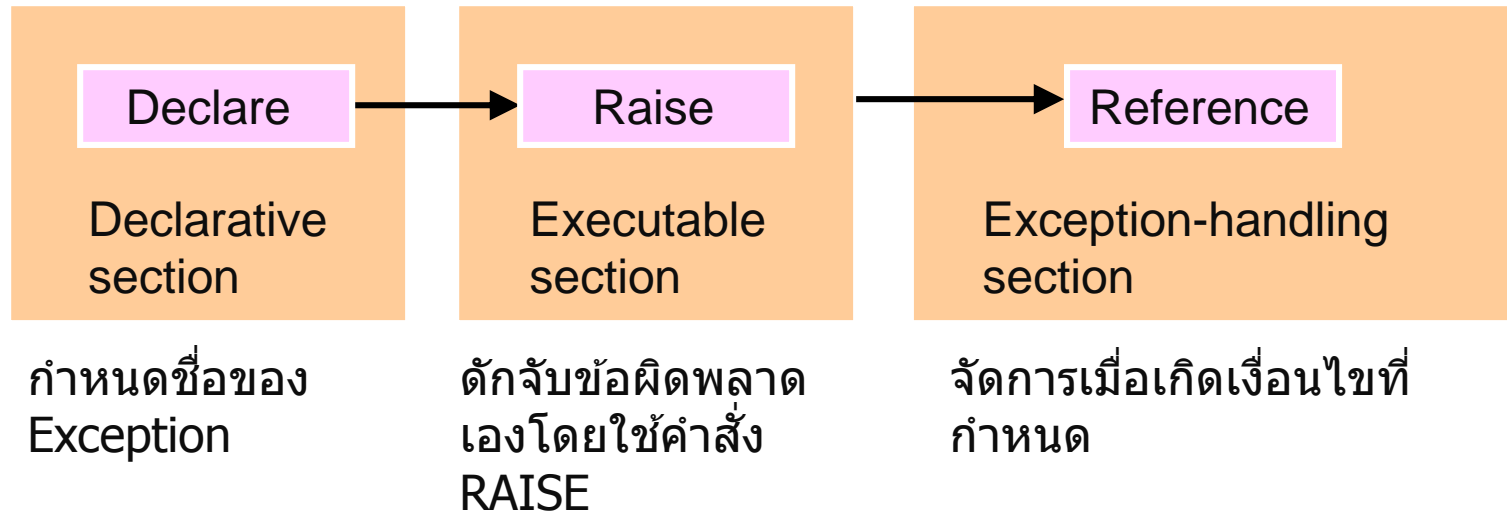
FUNCTIONS FOR TRAPPING EXCEPTIONS

Example

```
DECLARE
  error_code      NUMBER;
  error_message   VARCHAR2(255);
BEGIN
  ...
EXCEPTION
  ...
  WHEN OTHERS THEN
    ROLLBACK;
    error_code := SQLCODE;
    error_message := SQLERRM;
    INSERT INTO errors (e_user, e_date, err_code, err_message)
      VALUES (USER, SYSDATE, error_code, error_message);
END;
/
```

- ฟังก์ชันในการดักจับข้อผิดพลาด
 - SQLCODE: คื้ค่าหมายเลขของ error code
 - SQLERRM: คื้ค่าข้อความพร้อมที่อธิบายถึงหมายเลข error number

TRAPPING USER-DEFINED EXCEPTIONS



TRAPPING USER-DEFINED EXCEPTIONS

```
DECLARE
  v_deptno NUMBER := 500;
  v_name VARCHAR2(20) := 'Testing';
  e_invalid_department EXCEPTION;
BEGIN
  UPDATE departments
  SET department_name = v_name
  WHERE department_id = v_deptno;
  IF SQL%NOTFOUND THEN
    RAISE e_invalid_department;
  END IF;
  COMMIT;
EXCEPTION
  WHEN e_invalid_department THEN
    DBMS_OUTPUT.PUT_LINE('No such department id.');
```

1

2

3

/

PROPAGATING EXCEPTIONS IN A SUBBLOCK

โปรแกรมย่อยสามารถดัก
จับข้อผิดพลาดในส่วนของ
โปรแกรมหลักได้

```
DECLARE
    . . .
    e_no_rows    exception;
    e_integrity  exception;
    PRAGMA EXCEPTION_INIT (e_integrity, -2292);
BEGIN
    FOR c_record IN emp_cursor LOOP
        BEGIN
            SELECT ...
            UPDATE ...
            IF SQL%NOTFOUND THEN
                RAISE e_no_rows;
            END IF;
        END;
    END LOOP;
EXCEPTION
    WHEN e_integrity THEN ...
    WHEN e_no_rows THEN ...
END;
/
```