

Database 3

مراجعة عامة
PL/SQL

المحاضرة الأولى

Dr. Ramez Alkhatib

PL/SQL

- المتحولات Variables
- الشروط Conditions
- الحلقات Loops
- الاستثناءات Exceptions
- المؤشرات Cursors
- أنماط البيانات المجردة Abstract Data Types
- المصفوفات المتغيرة (varying arrays)
- الجداول المتداخلة Nested Tables
- الإجراءات و التوابع Procedures and Functions
- الحزم البرمجية Packages
- القوالب Triggers

أجزاء البرنامج

DECLARE

<variable declarations>

مقطع التصريحات

BEGIN

<executable statements>

المقطع التنفيذي

EXCEPTION

<exception handling>

معالجة الاستثناءات

END ;

نهاية البرنامج

مثال

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    radius INTEGER;
    area    NUMBER(14,2);
BEGIN
    radius := 3;
    area := Pi * Power(radius,2);
    INSERT INTO areas VALUES (radius, area);
END;
/
```

```
SQL> start d:/plsql/3.sql;
```

```
PL/SQL procedure successfully completed.
```

```
SQL> select * from areas;
```

RADIUS	AREA
3	28.27

المتحولات %TYPE , %ROWTYPE

```
lname employee.last_name%TYPE;
```

```
Dept_var Department%ROWTYPE;
```

```
Dept_var.department_name
```

الشروط

```
IF <some condition> THEN
    <some command>
ELSIF <some condition> THEN
    <some command>
ELSE
    <some command>
END IF;
```

```
IF <some condition> THEN
    IF <some condition> THEN
        <some command>
    END IF;
ELSE
    <some command>
END IF;
```

الشروط

```
DECLARE
    I INTEGER := 3;
BEGIN
    CASE I
        WHEN 1 THEN DBMS_OUTPUT.PUT_LINE('ONE');
        WHEN 2 THEN DBMS_OUTPUT.PUT_LINE('TOW');
        WHEN 3 THEN DBMS_OUTPUT.PUT_LINE('THREE');
        WHEN 4 THEN DBMS_OUTPUT.PUT_LINE('FOUR');
        WHEN 5 THEN DBMS_OUTPUT.PUT_LINE('FIVE');
        ELSE      DBMS_OUTPUT.PUT_LINE('?');
    END CASE;
END;
/
```

الحلقات

- **LOOP**
- **FOR**
- **WHILE**

مثال Loop

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    radius INTEGER;
    area    NUMBER(14,2);
BEGIN
    radius := 3;
    LOOP
        area := Pi * Power (radius, 2);
        INSERT INTO areas VALUES (radius, area);
        radius := radius + 1;
        EXIT WHEN area > 100;
    END LOOP;
END;
/
```

```
SQL> select * from areas;
```

RADIUS	AREA
3	28.27
4	50.27
5	78.54
6	113.1

مثال FOR

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    radius INTEGER;
    area    NUMBER(14,2);
BEGIN
    FOR radius IN 1..7 LOOP
        area := Pi * Power (radius, 2);
        INSERT INTO areas VALUES (radius, area);
    END LOOP;
END;
/
```

مثال WHILE

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    radius INTEGER;
    area    NUMBER(14,2);
BEGIN
    radius := 3;
    WHILE radius <= 7 LOOP
        area := Pi * Power (radius, 2);
        INSERT INTO areas VALUES (radius, area);
        radius := radius + 1;
    END LOOP;
END;
/
```

الاستثناءات

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    radius INTEGER;
    area    NUMBER(14,2);
    some_variable NUMBER(14,2);
BEGIN
    radius := 3;
    LOOP
        some_variable := 1/(radius-4);
        area := Pi * Power (radius, 2);
        INSERT INTO areas VALUES (radius, area);
        radius := radius + 1;
        EXIT WHEN area > 100;
    END LOOP;
END;
/
```

```
DECLARE
*
ERROR at line 1:
ORA-01476: divisor is equal to zero
ORA-06512: at line 9
```

الاستثناءات

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    radius INTEGER;
    area    NUMBER(14,2);
    some_variable NUMBER(14,2);
BEGIN
    radius := 3;
    LOOP
        some_variable := 1/(radius-4);
        area := Pi * Power (radius, 2);
        INSERT INTO areas VALUES (radius, area);
        radius := radius + 1;
        EXIT WHEN area > 100;
    END LOOP;
EXCEPTION
    WHEN ZERO_DIVIDE THEN
        INSERT INTO areas VALUES (0, 0);
END;
/
```

المؤشرات (الضمني) SELECT / INTO

```
DECLARE
    min_salary NUMBER;
    max_salary NUMBER;
BEGIN
    SELECT MIN(salary), MAX(salary)
    INTO min_salary, max_salary
    FROM employee;
    DBMS_OUTPUT.PUT_LINE('Smallest salary is: '||min_salary);
    DBMS_OUTPUT.PUT_LINE('Biggest salary is: '||max_salary);
END;
/
```

المؤشرات الصريحة

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    area    NUMBER(14,2);
    CURSOR rad_cursor IS
        SELECT * FROM radius_vals;
    rad_val rad_cursor%ROWTYPE;
BEGIN
    OPEN rad_cursor;
    FETCH rad_cursor INTO rad_val;
    area := Pi * Power(rad_val.radius,2);
    INSERT INTO areas VALUES (rad_val.radius, area);
    CLOSE rad_cursor;
END;
/
```

المؤشرات مع الحلقات

```
DECLARE
    Pi CONSTANT NUMBER(9,7) := 3.1415926;
    area    NUMBER(14,2);
    CURSOR rad_cursor IS
        SELECT * FROM radius_vals;
    rad_val rad_cursor%ROWTYPE;
BEGIN
    OPEN rad_cursor;
    LOOP
        FETCH rad_cursor INTO rad_val;
        EXIT WHEN rad_cursor%NOTFOUND;
        area := Pi * Power(rad_val.radius,2);
        INSERT INTO areas VALUES (rad_val.radius, area);
    END LOOP;
    CLOSE rad_cursor;
END;
/
```

أنماط البيانات المجردة

```
CREATE TYPE ADDRESS_TY AS OBJECT
(
    street VARCHAR2(50) ,
    city    VARCHAR2(25) ,
    state   CHAR(2) ,
    zip     NUMBER
)
/
```

```
INSERT INTO CUSTOMER VALUES
(1,
    PERSON_TY('John Smith',
        ADDRESS_TY('57 PLEASANT ST', 'FINN', 'NH', 11111)));
```

المصفوفات المتغيرة

BORROWER	
<u>NAME</u>	TOOL
Jed Hopkins	{Shovel, Axe, Hammer}
John Smith	{Shovel, Saw}

```
CREATE OR REPLACE TYPE TOOLS_VA  
    AS VARRAY(5) OF VARCHAR2(25);
```

```
CREATE TABLE Borrower (  
    name VARCHAR2(25) PRIMARY KEY ,  
    tools TOOLS_VA  
);
```

مسألة

يرغب أحد معارض الألبسة بتخزين بيانات السلع والزبائن وطلبات الشراء المتوفرة لديه بحيث يتم انشاء التالي :

- جدول السلع ويتضمن الحقول التالية: رقم السلعة - اسم السلعة - تاريخ الانتاج (يوم- شهر -سنة) - الكمية المتوفرة - الشركة المنتجة.
- جدول طلبات الشراء ويتضمن الحقول التالية : رقم الطلب- اسم الزبون (اسم أول- اسم الأب - الكنية) - البريد الالكتروني (أكثر من بريد الكتروني) - بطاقة اعتماد (رقم البطاقة - تاريخ الانتهاء - رمز الأمان) - الكمية المطلوبة - الرقم الوطني - رقم الزبون رقم السلعة .

المطلوب :

- (١) انشاء الجداول السابقة مستخدما ما تراه مناسباً من الأنماط والمصفوفات اذا استدعى الأمر ذلك ومضيفاً ما تراه مناسباً من القيود.
- (٢) انشاء كتلة برمجية تقوم بإيجاد أعلى سلعة متوفرة في المعرض باستخدام مفهوم المؤشرات .
- (٣) انشاء كتلة برمجية تقوم بإعطاء معلومات عن السلع وكل طلبات الشراء التي تخصها باستخدام مفهوم المؤشرات.