الجزء الأول

SQL/PL

جامعة الإمام محمد بن سعود الإسلامية

إعداد :: طلال سعود العصيمي

1432 / 2011

للتواصل ::

imam.Course@gmail.com

الفصل الأول

تعريف و إسناد المتغيرات

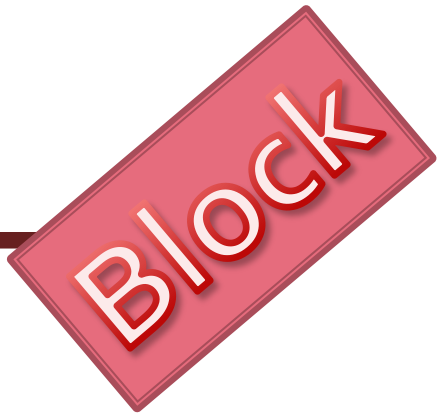
الشكل العام

Declare
هنا تعريف المتغيرات و المؤشرات

Begin
حق البرنامج Body

Exception
معالجة الأخطاء

end;



ملاحظة هامة :

قسم declare وقسم exception
هما اختيارين أي لا يشترط وجودهما

أي إذا كان لا يوجد لديك تعريف متغيرات لا تستخدم Declare

وإذا كنت لا تتعامل مع الأخطاء لا تستخدم Exception

إسـناد و تعريف المتغيرات

```
name :='ahmed khaled';
v:=15;
```

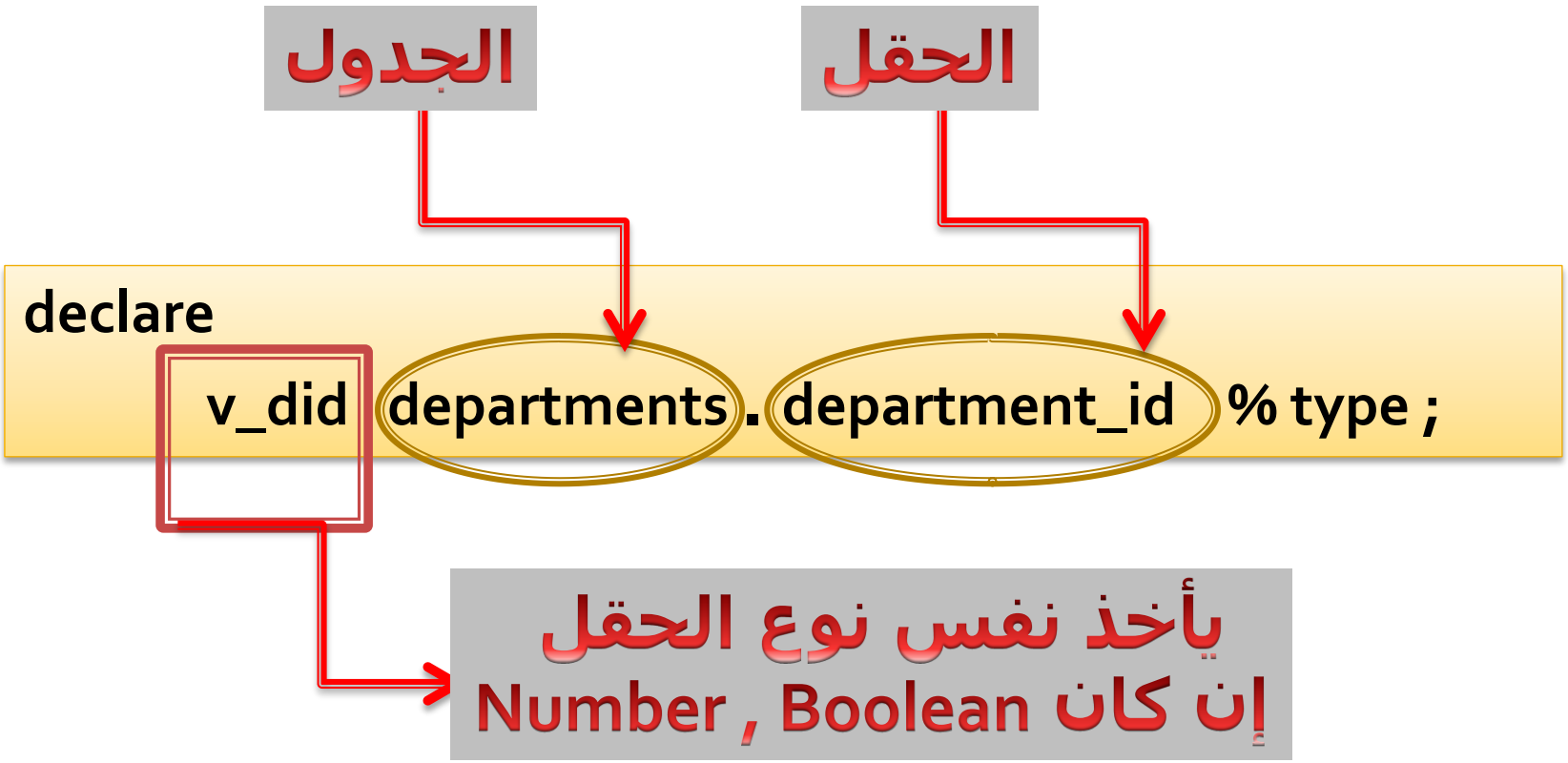
```
v_x number;
v_y number(10);
v_z number(8,2) := 55.22;
name varchar2(50);
v_lname varchar2(20) :='Hilal';
v_bdate date;
is_married boolean :=false;
```

يجب وضع النقطتين

قبل =

إسناد و تعريف المتغيرات

خاصية نوع الحقل : type%



إسناد و تعريف المتغيرات

% rowtype :

مجموعة متغيرات، وتستخدم لتأسيس مصفوفة من المتغيرات مبنية على الأعمدة الموجودة في مؤشر ما أو جدول ما

الطباعة (الإخراج)

```
dbms_output.put_line(    );
```

```
dbms_output.put_line(v_y);
dbms_output.put_line('##########');
dbms_output.put_line('The name is : '|| name);
```

فائدة || الموجدة ضمن عملة الطباعة هي للوصل بين التعبيرين

Example (1) ::

```
declare
        v_did number;
        v_dname varchar2(50);
        v_mid number;
        v_lid number;
begin
        select department_id,department_name,manager_id,location_id
        into v_did,v_dname,v_mid,v_lid
        from departments
        where department_id=10;

        dbms_output.put_line('The recors is : ');
        dbms_output.put_line('ID : '|| v_did);
        dbms_output.put_line('Name : '|| v_dname);
        dbms_output.put_line('Manager: '||v_mid);
        dbms_output.put_line('Location : '|| v_lid);
end ;
```

شـــرح الـكود

Example (1) ::

select تستخدم لاختيار الأعمدة والبيانات التي نحتاجها

```
begin
        select department_id , department_name , manager_id , location_id
```

into تستخدم لاختيار الأعمدة **ولكن** من خلال ماتم تعريفه في Declare

```
        into v_did , v_dname,v_mid , v_lid
```

تستخدم لتحديد الجدول الذي سنسترجع منه البيانات

```
        from departments
        where department_id=10;
```

Example (2) ::

```
declare
        v_did  number;
        v_dname  varchar2(50);
        v_mid  number;
        v_lid  number;
```

هنا قمنا بإنشاء متغيرات جديدة

```
begin
        v_did :=666;
        v_dname :='Imam UN';
        v_mid :=105;
        v_lid :=1700;
```

ثم تم إسناد قيم جديدة للمتغيرات

```
        insert  into departments
( department_id,department_name, manager_id , location_id)
        values(v_did  , v_dname , v_mid  , v_lid);
end;
```

هنا نضيف القيم الجديدة للجدول

Example (3) ::

```
declare
        v_did departments.department_id%type;
        v_dname departments.department_name%type;
        v_mid departments.manager_id%type;
begin
        select department_id,department_name,manager_id
        into v_did,v_dname,v_mid
        from departments
        where department_id=10;

        dbms_output.put_line('The recors is : ');
        dbms_output.put_line('ID : '|| v_did);
        dbms_output.put_line('Name : '|| v_dname);
        dbms_output.put_line('Manager: '||v_mid);
end ;
```

شـــرح الـكود

Example (3) ::

```
declare
        v_did  departments.department_id %type;
        v_dname  departments.department_name %type;
        v_mid  departments.manager_id %type;
```

هنا قمنا بإنشاء متغيرات جديدة باستخدام (%type)

الذي ينشئ قيم من نفس نوع الحقل

شـــرح الـكود

Example (3) ::

```
begin
        select department_id , department_name , manager_id
        into v_did , v_dname , v_mid
        from departments
        where department_id=10;
```

تستخدم لاختيار الأعمدة والبيانات التي نحتاجها من خلال

ما تم تعريفه من متغيرات

شـــرح الـكود

Example (3) ::

```
        dbms_output.put_line('The recors is : ');
        dbms_output.put_line('ID : '|| v_did);
        dbms_output.put_line('Name : '|| v_dname);
        dbms_output.put_line('Manager: ' || v_mid);
end ;
```

طباعة الحقول التي نريدها

اسم الحقل و ما يقابله من متغيرات

Example (4) ::

```
declare
        v_rec departments % rowtype;
begin
        select *
        into v_rec
        from departments
        where department_id=10;

        dbms_output.put_line('The recors is : ');
        dbms_output.put_line('ID : '|| v_rec.department_id);
        dbms_output.put_line('Name : '|| v_rec.department_name);
        dbms_output.put_line('Manager: '||v_rec.manager_id);
        dbms_output.put_line('Location : '|| v_rec.location_id);
end ;
```

شـــرح الـكود

Example (4) ::

```
declare
        v_rec departments % rowtype;
```

عرفنا (v_rec) و هي مصفوفة من المتغيرات مبنية على الأعمدة الموجودة في الجدول

شـــرح الـكود

Example (4) ::

```
begin
        select  *
        into v_rec
        from departments
        where department_id=10;
```

هنا اخترنا جميع الحقول الموجودة في الجدول

شـــرح الـكود

طباعة الحقول التي نريدها

Example (4) ::

```
        dbms_output.put_line('The recors is : ');
        dbms_output.put_line('ID : '|| v_rec . department_id);
        dbms_output.put_line('Name : '|| v_rec . department_name);
        dbms_output.put_line('Manager: '||v_rec . manager_id);
        dbms_output.put_line('Location : '|| v_rec . location_id );
end ;
```

المتغير الجديد

و ما يقابله من حقل

الفصل الثاني

أوامر اللغة

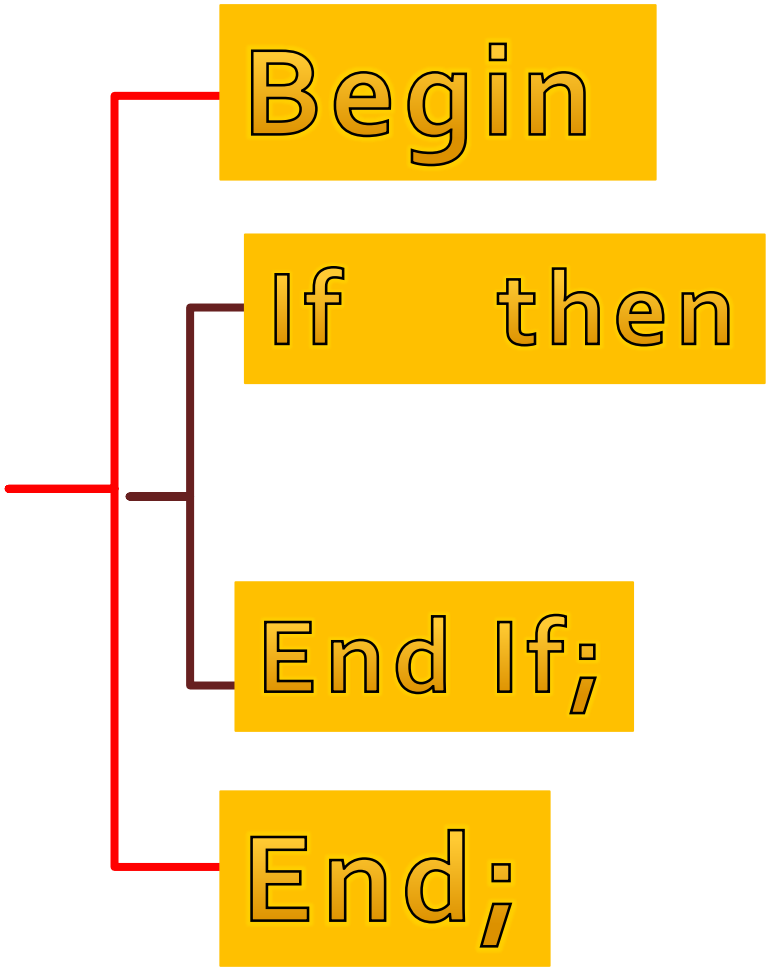
أوامر اللغة

١.جملة الشرط

IF then

أوامر اللغة

الشكل العام



Example (5) ::

أوامر اللغة

```
declare
        v_salary  employees.salary %type;
begin
        select salary  into  v_salary
        from employees
        where  employee_id=150;
        if  v_salary > 10000 then
                dbms_output.put_line('VIP');
        end if ;
end;
```

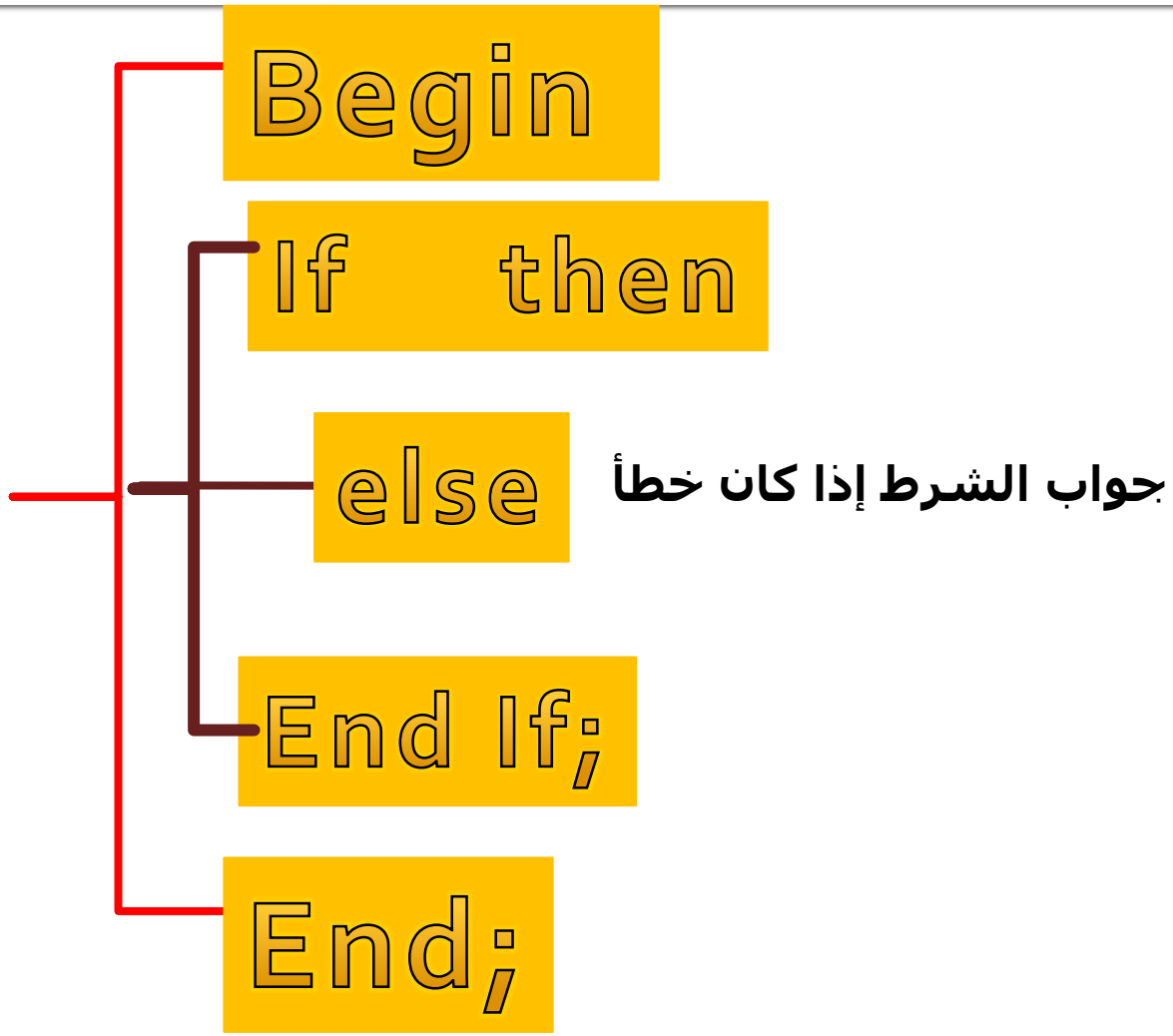
نعرف متغير جديد (v_salary) نختار راتب الموظف عندما

يكون (employee_id) = ١٥٠ إذا كان الراتب أكبر من ١٠٠٠

نطبع رسالة (VIP)

أوامر اللغة

الشكل العام



Example (6) :: جملة الشرط IF

```
declare
        v_salary  employees.salary %type;
begin
        select salary into v_salary
        from employees
        where  employee_id=102;
        if  v_salary > 10000 then
                dbms_output.put_line('VIP');
        else
                dbms_output.put_line('normal');
        end if ;
end;
```

نعرف متغير جديد (v_salary) نختار راتب الموظف عندما

يكون (employee_id) = ١٠٢إذا كان الراتب أكبر من ١٠٠٠

نطبع رسالة (VIP) و إذا كان أقل من ١٠٠٠ نطبع رسالة (normal)

أوامر اللغة

الشكل العام

Example (7) :: IF & Else

```
declare
        v_salary employees.salary%type;
begin
        select salary into v_salary
        from employees
        where employee_id=150;
        if v_salary > 10000 then
                dbms_output.put_line('VIP');
        elsif (v_salary >=3000 and v_salary <=10000) then
                dbms_output.put_line('normal');
        else
                dbms_output.put_line('low');
        end if;
end;
```

Example (7) :: IF & ElsIF

```
declare
        v_salary employees.salary%type;
begin
        select salary into v_salary
        from employees
        where employee_id=150;
        if v_salary > 10000 then
                dbms_output.put_line('VIP');
        elsif (v_salary >=3000 and v_salary <=10000) then
                dbms_output.put_line('normal');
        else
                dbms_output.put_line('low');
        end if;
end;
```

شـــرح الـكود

Example (7) ::

نعرف متغير جديد (v_salary) نختار راتب الموظف

عندما يكون (employee_id) = ١٠٢

إذا كان الراتب أكبر من ١٠٠٠ نطبع رسالة (VIP)

و إذا كان الراتب أكبر من ٣٠٠٠ و أقل من ١٠٠٠

نطبع رسالة (normal)

و إذا كان غير ذلك نطبع رسالة (Low)

Example (8) :: IF & ElsIF

```
declare
        v_salary employees.salary%type;
begin
        select salary into v_salary
        from employees
        where employee_id=150;
        if v_salary > 10000 then
                dbms_output.put_line('VIP');
        elsif (v_salary  between  3000 and 10000) then
                dbms_output.put_line('normal');
        else
                dbms_output.put_line('low');
        end if;
end;
```

شـــرح الـكود

Example (8) ::

نعرف متغير جديد (v_salary) نختار راتب الموظف

عندما يكون (employee_id) = ١٠٢

إذا كان الراتب أكبر من ١٠٠٠ نطبع رسالة (VIP)

و إذا كان الراتب بين ٣٠٠٠ و ١٠٠٠ نطبع رسالة (normal)

و إذا كان غير ذلك نطبع رسالة (Low)

ملاحظة ::

لكن لاحظ هنا كتابة elsif

ولا تخطي فيها فهي ليست elseif

بإمكاننا استخدام (And , Between) داخل

جملة الشرط (IF & elsIf)

أوامر اللغة

٢. التكرار

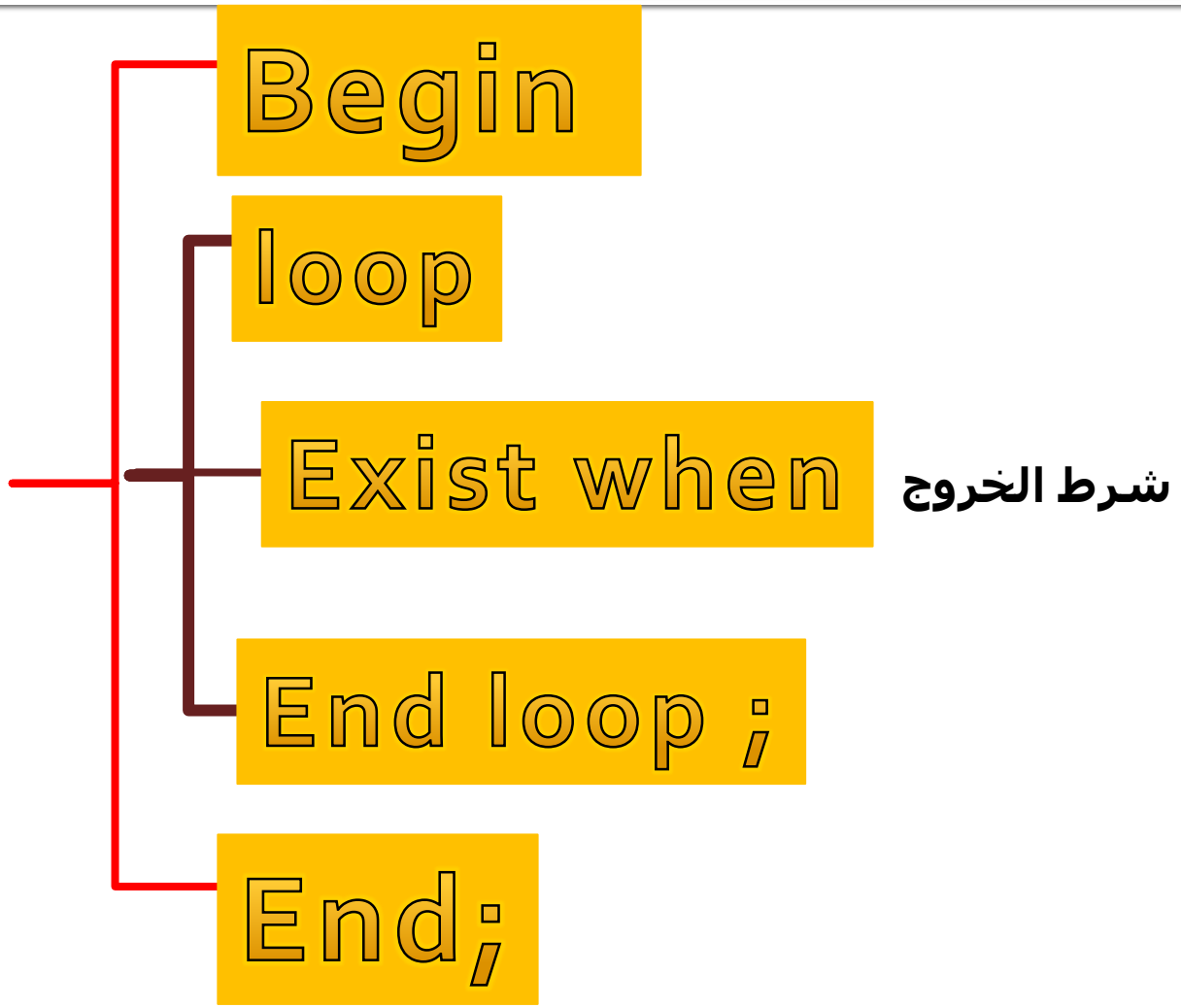
ويوجد عدة أوامر للتكرار وهي:

أوامر اللغة

(i) loop-exit-end

أوامر اللغة

الشكل العام



Example (9) ::

Loop

```
declare
  v_counter number := 1;
begin

loop
        dbms_output.put_line(v_counter);
        v_counter:=v_counter+1;
        exit when v_counter>10;
end loop;
end ;
```

العداد

نعرف عداد (v_counter)

و نسند له قيمة (١)

العداد كل مرة يزود (١)

(يخرج خارج Loop) عندما يكون أكبر من ١٠

Example (10) ::

Loop

```
declare
v_counter  number :=10;
begin

loop
        dbms_output.put_line(v_counter);
        v_counter:=v_counter-1;
        exit when v_counter=0;
end loop;
end ;
```

العداد

نعرف عداد (v_counter)

و نسند له قيمة (١)

العداد كل مرة ينقص(١)

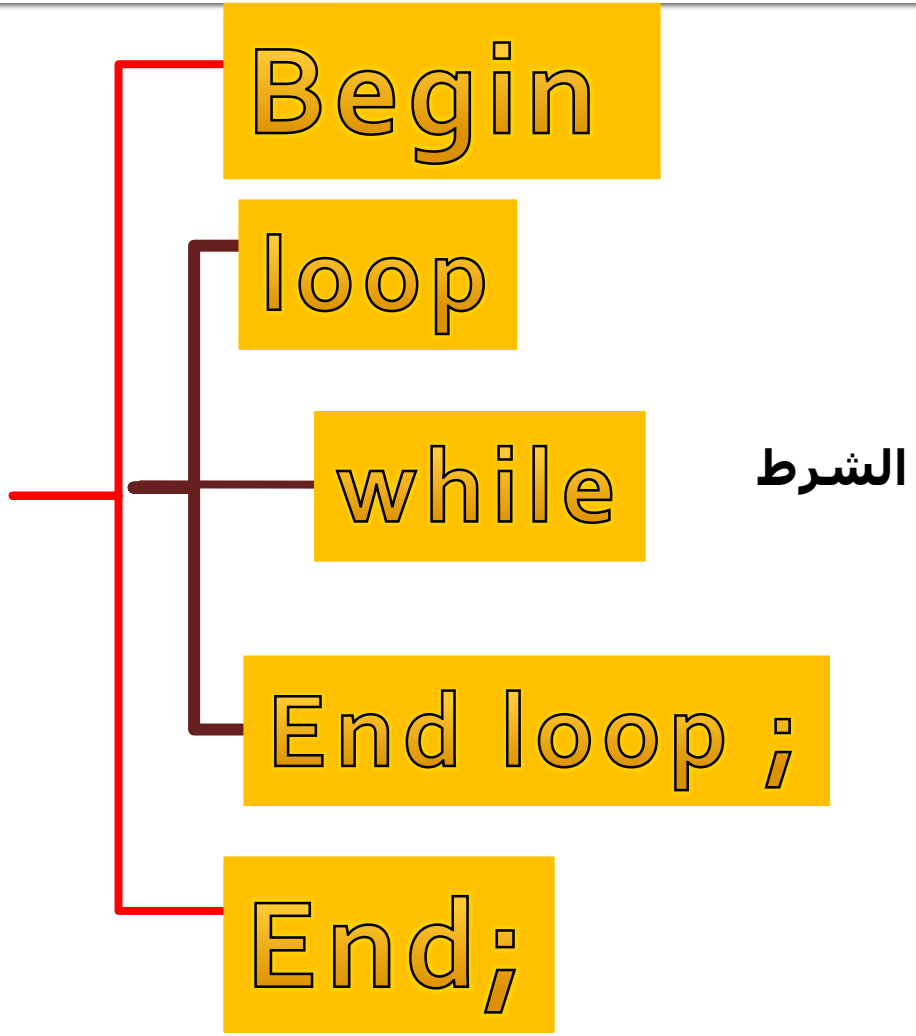
(يخرج خارج Loop) عندما يكون العداد = صفر

أوامر اللغة

(ii) WHILE - LOOP - END

أوامر اللغة

الشكل العام



Example (11) ::

While

```
declare
v_counter number :=1;
begin
while v_counter <=10 loop
                dbms_output.put_line(v_counter);
                v_counter:=v_counter+1;
end loop;
end ;
```

العداد

نعرف عداد (v_counter)

و نسند له قيمة (١)

إذا كان العداد أصغر من ١٠

يطبع العدد و يزود (١)

Example (12) ::

While

```
declare
v_counter number :=10;
begin
while v_counter >=1 loop
        dbms_output.put_line(v_counter);
        v_counter:=v_counter-1;
end loop;

end ;
```

العداد

نعرف عداد (v_counter)
و نسند له قيمة (١)

إذا كان العداد أكبر من ١
يطبع العدد و ينقص (١)

Example (13) ::

While

```
declare
        v_sal  employees.salary %type;
        v_counter number :=0;
begin
        select salary into v_sal
        from employees
        where employee_id=106;
        while v_sal <10000 loop
                v_sal :=v_sal*1.2;
                v_counter :=v_counter+1;
        end loop;
        dbms_output.put_line('The number of monthes is : '||v_counter);
end;
```

Example (13) :: While

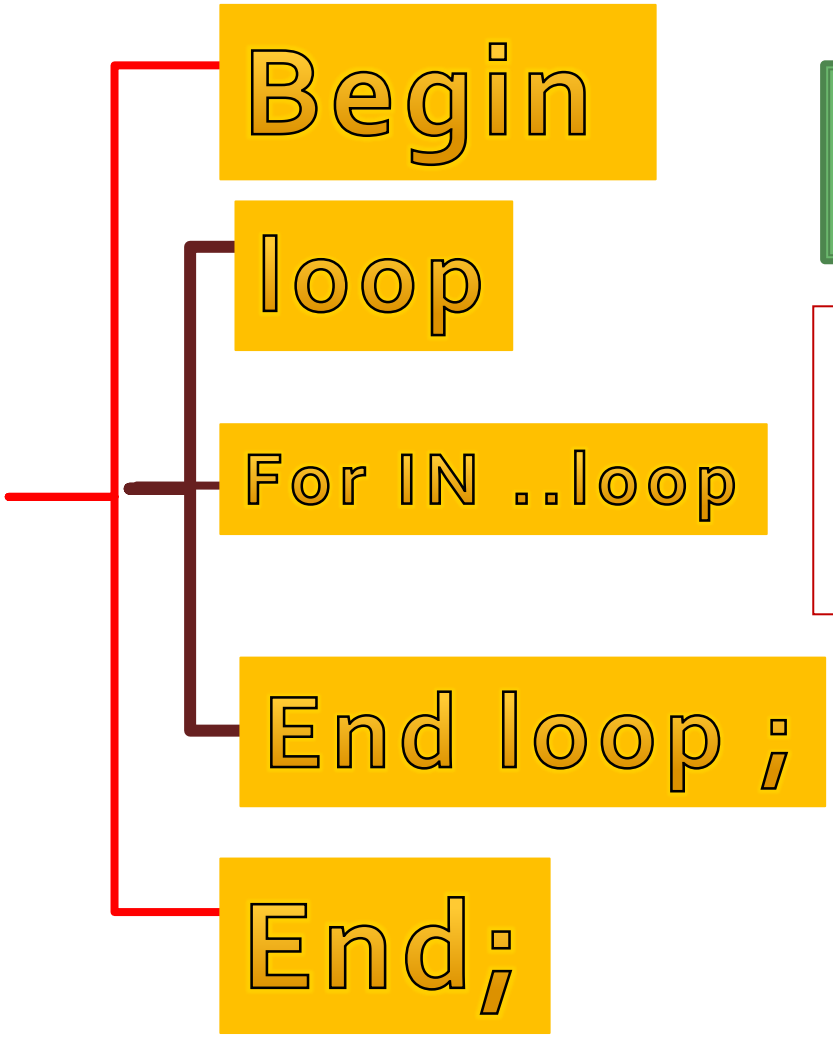
عرفنا متغير اسمه (v_sal) و عرفنا عداد اسمه (v_counter) و أسندنا له قيمة (صفر) ثم يختار (salary) للموظف رقم ١٠٦ إذا كان أقل من (١٠٠٠٠) ويزود له ١٠ % (١,٢) و يطبع الراتب بعد الزيادة

أوامر اللغة

(iii) FOR - IN - LOOP - END

أوامر اللغة

الشكل العام



الصيغة العامة

FOR i IN البداية.. النهاية LOOP
الجمل المراد تكرارها
LOOP END

Example (13) ::

For

العداد ما يحتاج Declare

```
begin
For v_counter in 1 ..10 loop
        dbms_output.put_line(v_counter);
end loop;
end ;
```

ناتج البرنامج ::

v_counter = 1
v_counter = 2
v_counter = 3
v_counter = 4
سوف يطبع إلى الرقم ١٠

Example (14) ::

For

```
begin
For v_counter  in reverse 1..10 loop
        dbms_output.put_line(v_counter);
end loop;
end ;
```

Declare العداد ما يحتاج

العداد يبدأ بالعكس

ملاحظة ::

في جملة For

ما نحتاج Declare لأن نقدر نعرف متغير

داخل الشرط

مثل:: مثال رقم ١٢ و ١٣

الفصل الثالث

CURSORS

المؤشرات

CURSORS الشكل العام

تستخدم المؤشرات لإدارة عبارات التحديد Select في لغة sql وكما لاحظنا الأوامر السابقة مثل IFوالتكرار لم نستخدمها مع بيانات الجداول المخزنة ولعمل ذلك لابد من استخدام هذه المؤشرات.

وهناك نوعين من المؤشرات هي الضمنية والصريحة وسوف نتطرق لك واحد بالتفصيل والأمثلة اللازمة.

المؤشرات الصريحة :

يتم تعريف هذا النوع من المؤشرات كجزء من الإعلان Declareويجب ان تشتمل عبارة SQLالمعرفه على عبارة التحديد Selectفقط

CURSORS الشكل العام

وعند استخدام المؤشرات الصريحة دائما ما ستكتب أربعة مكونات كما يلي:

١- يتم تعريف المؤشر في الجزء Declare

٢- يتم فتح المؤشر بعد عبارة Begin

الصيغة العامة لتعريف المؤشر الصريح كما يلي:

DECLARE

CURSOR اسم المؤشر IS

الاستعلام

CURSORS الشكل العام

تقوم باستبدال اسم المؤشر باسم مؤشر حقيقي

وتقوم بوضع جملة الاستعلام select في مكان الاستعلام

ولكي تقوم بفتح هذا المؤشر وتستخدمه نقوم بفتحه باستخدام الامر open كمايلي:

OPEN اسم المؤشر

وبعد فتح المؤشر تقوم باسترجاع او تحميل البيانات سطر(سجل) واحد من المؤشر الذي تم تعريفه باستخدام الامر FETCH كمايلي:

FETCH اسم المؤشر INTO متغير١،متغير٢،.....

CURSORS الشكل العام

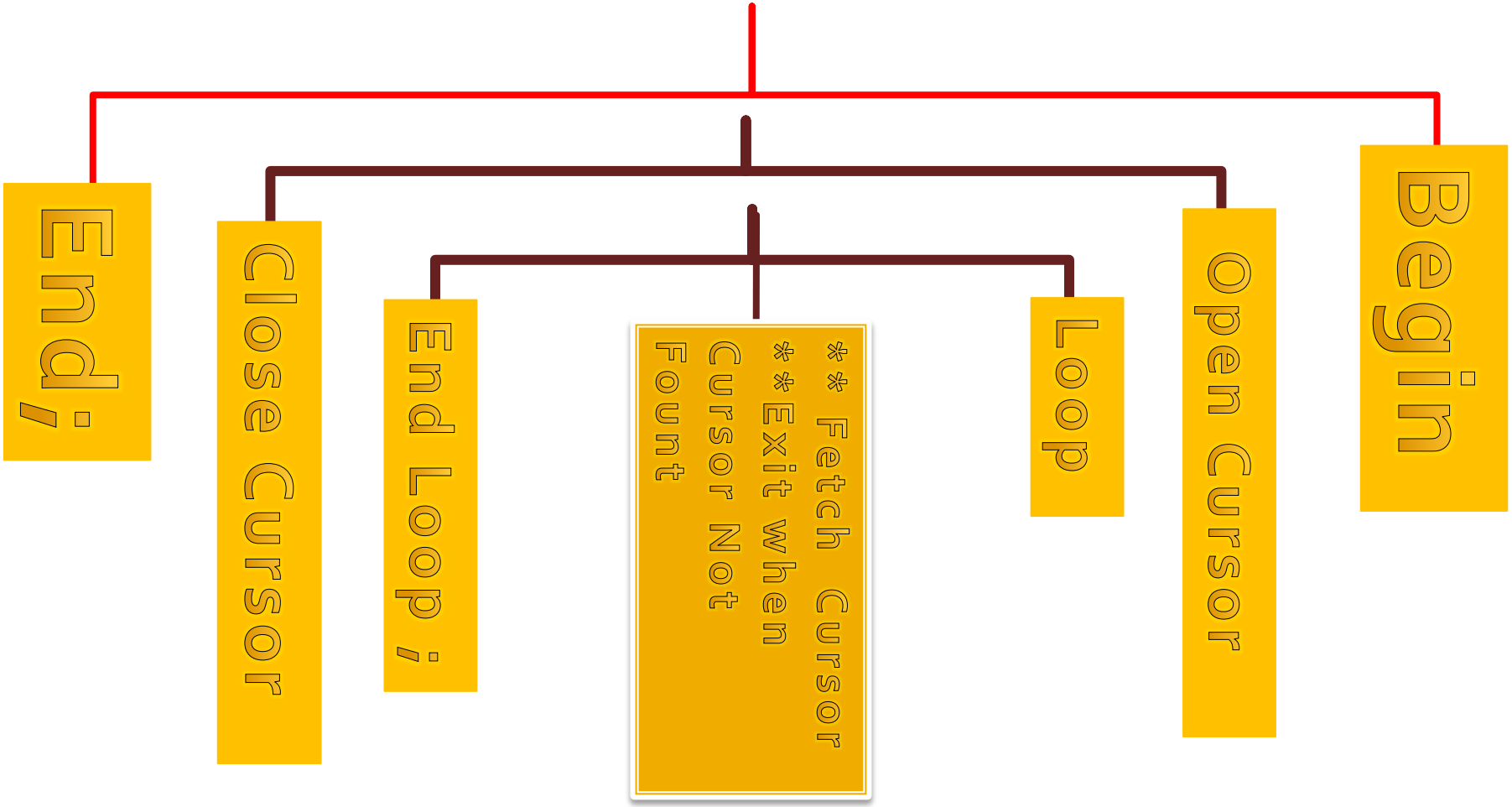
ومعنى هذا اي قم باسترجاع البيانات من المؤشر المعطى اسمه وحملها into الى المتغيرات كع ملاحظة ان عدد المتغيرات يساوي عدد الحقول الموجودة في استعلام المؤشر.

وبعد الانتهاء من اجراء العمليات على المؤشر يجب عليك اغلاقه ويتم اغلاقه كمايلي:

```
close cursor_name
```

الشكل العام

CURSORS _ Loop



Example (15) :: Cursors _ Loop

```
declare
        v_id employees.employee_id%type;
        v_fname employees.first_name%type;
        v_lname employees.last_name%type;
        v_sal employees.salary%type;
        cursor c is select employee_id,first_name,last_name,salary
        from employees where department_id=50;
begin
        open c;
        loop
                fetch c into v_id,v_fname,v_lname,v_sal;
                exit when  c % notfound;
        dbms_output.put_line(v_id||'  '||v_fname||'  '||v_lname||'  '||v_sal);
        end loop;
        close c;
end ;
```

شـــرح الـكود

Example (15) ::

```
declare
        v_id employees.employee_id%type;
        v_fname employees.first_name%type;
        v_lname employees.last_name%type;
        v_sal employees.salary%type;
```

نعرف المتغيرات التي نريد

```
        cursor c is select employee_id,first_name,last_name,salary
        from employees
        where department_id=50;
```

نعرف مؤشر (c) و نختار رقم الموظف و الاسم الأول و الاسم الأخير للموظف و الراتب

Example (15) :: شـــرح الـكود

```
begin
        open c;
        loop
fetch   c  into v_id,v_fname,v_lname,v_sal;

exit when  c % notfound;
dbms_output.put_line(v_id||'  '||v_fname||'  '||v_lname||'  '||v_sal);
        end loop;
        close c;
end ;
```

جلب المؤشر

إذا لم نجد قيمة المؤشر نخرج

ملاحظات

لاحظنا في المثال السابق أن الاستعلام في Cursor
سوف يعود بسجل واحد لكن ماذا يحدث لو أعاد المؤشر أكثر من سجل
وأردنا المرور على كافة السجلات ؟

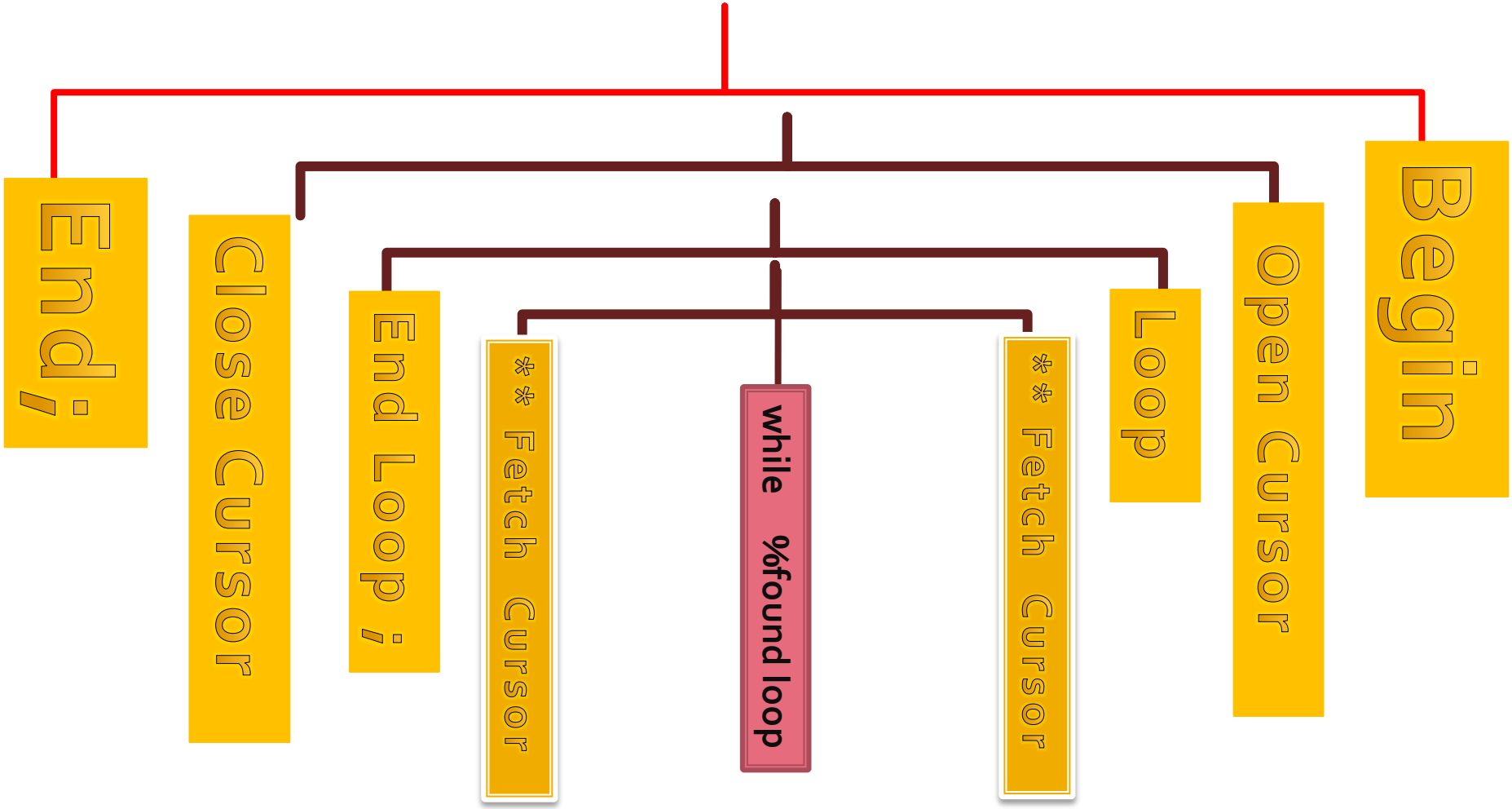
لذا نستخدم الأمر Found OR NotFound

نستخدم الأمر NotFound مع Loop

هل سجلات المؤشر انتهت أم لا ونعرف ذلك من خلال خاصية Found

الشكل العام

CURSORS _ While



Example (16) :: Cursors _ While

```
declare
        v_id employees.employee_id%type;
        v_fname employees.first_name%type;
        v_lname employees.last_name%type;
        v_sal employees.salary%type;
        cursor c is select employee_id,first_name,last_name,salary
        from employees where department_id=50;
begin
        open c;
        fetch c into v_id,v_fname,v_lname,v_sal;
        while c %found loop
        dbms_output.put_line(v_id||'  '||v_fname||'  '||v_lname||'  '||v_sal);
        fetch c into v_id,v_fname,v_lname,v_sal;
        end loop;
        close c;
end ;
```

شـــرح الـكود

Example (16) ::

```
begin
        open c;
        fetch c into v_id,v_fname,v_lname,v_sal;
        while c %found loop
        dbms_output.put_line(v_id||'  '||v_fname||'  '||v_lname||'  '||v_sal);
        fetch c into v_id,v_fname,v_lname,v_sal;
        end loop;
        close c;
end ;
```

هنا نستخدم Cursor مع While

نستخدم ::

two Fetch && %Found

Example (17) :: Cursors _ While

```
declare
        cursor c is select employee_id,first_name,last_name,salary
        from employees where department_id=50;
        v_rec c %rowtype;
begin
        open c;
        fetch c into v_rec;
        while c %found loop
        dbms_output.put_line(v_rec.employee_id||'  '||v_rec.first_name||'
'||v_rec.last_name||'  '||v_rec.salary);
                fetch c into v_rec;
        end loop;
        close c;
end ;
```

شـــرح الـكود

Example (17) ::

هنا نفس المثال السابق و لكن الفرق تعريف المتغيرات و خصوصا الأمر %rowtype

نعرف المؤشر (c) ثم نعرف مجموعة المتغيرات (v_rec)

Example (18) :: Cursors _ For

```
declare
        cursor c is select employee_id,first_name,last_name,salary
        from employees
        where department_id=50;
begin
        for v_rec in c loop
        dbms_output.put_line(v_rec.employee_id||'  '||v_rec.first_name||'
        '||v_rec.last_name||'  '||v_rec.salary);
        end loop;
end;
```

شـــرح الـكود

Example (18) ::

عند استخدام جملة التكرار For مع المؤشر

لا نحتاج إلى ::

Fetch & Open & Close

Example (19) :: Cursors _ IF

```
declare
        v_job employees.job_id%type;
        v_id employees.employee_id%type;
        cursor c is select employee_id,job_id from employees;
begin
open c;
loop
        fetch c into v_id,v_job;
        exit when c%notfound;
        if v_job='ST_CLERK' then
                delete from employees where employee_id=v_id;
        elsif v_job='IT_PROG' then
                update employees set department_id=50 where employee_id=v_id;
        else
                dbms_output.put_line('No Changes');
        end if;
end loop;
close c;
end ;
```

شـــرح الـكود

Example (19) ::

نعرف متغير (v_sal) و (v_id) ونعرف مؤشر (C)

إذا لم نجد رقم الموظف (v_id) و الراتب (v_sal) إذا لم نجده نخرج

* إذا كان الراتب بين ٣٠٠٠ و ١٠٠٠٠ نعمل updateللراتب و نزوده ١٠ %

(١,٢)

* إذا كان الراتب أقل من ٣٠٠٠ نعمل updateللراتب و نزوده ١٥ %

(١,١٥)

* إذا كان الراتب غير ذلك نعمل updateللراتب و نزوده ٢٠% (١,١)

ملاحظة

لم نستخدم مع Loop الأمر Fetch كما في

المثال رقم (١٥)

لكن استخدمنا جملة التكرار (IF & IFels)

Example (20) :: Cursors _ IF

```
declare
        v_sal employees.salary%type;
        v_id employees.employee_id%type;
        cursor c is select employee_id,salary from employees;
begin
        open c;
        loop
                fetch c into v_id,v_sal;
                exit when c%notfound;
                if v_sal < 3000 then
        update employees set salary=salary*1.2 where employee_id=v_id;
                elsif v_sal between 3000 and 10000 then
        update employees set salary=salary*1.15 where employee_id=v_id;
                else
        update employees set salary=salary*1.1 where employee_id=v_id;
                end if;
        end loop;
        close c;
end ;
```

الفصل الرابع

Exception

الاستثناء

Exception

Exception handlers :

هو جزء اختياري ، و هذا الجزء يبدأ بكلمة Exception

و في هذا الجزء يتم تحديد الإجراء الذي سيتم اتخاذه في حالة حدوث خطأ

Exception

تنقسم الأخطاء إلى ::

1. Too many rows
2. No data found
3. Zero divide
4. Others

Example (21) :: Exception

1. Too many rows

```
declare
        v_lname employees.last_name%type;
begin
      dbms_output.put_line('Before Exception');
            select last_name into v_lname
             from employees;
            dbms_output.put_line(v_lname);
exception
            when too_many_rows then
                  dbms_output.put_line('Pleae set Comndition in sql statement');
end;
```

Example (22) :: Exception

2. No data found

```
declare
        v_lname employees.last_name%type;
begin
        select last_name into v_lname
        from employees
        where employee_id=101;
        dbms_output.put_line(v_lname);
exception
        when no_data_found then
        dbms_output.put_line('No data returned from sql statement');
end;
```

Example (23) :: Exception

3. Zero divide

```
declare
        v_lname employees.last_name%type;
begin
        select last_name into v_lname
         from employees
        where employee_id=101;
        dbms_output.put_line(50/0);
exception
        when zero_divide then
                dbms_output.put_line('Cant division on zero');
end;
```

Example (25) :: Exception

عرفنا جدول خاص بالأخطاء يتكون من :

```
create table error
(
        title varchar2(25),

        err_date date
)
```

Title
خاص بعنوان الخطأ

err_date
خاص بتاريخ الخطأ

Example (25) :: Exception

```
declare
        v_lname employees.last_name%type;
begin
        select last_name into v_lname from employees ;
        dbms_output.put_line(v_lname);
        dbms_output.put_line(50/0);
exception
        when too_many_rows then
                dbms_output.put_line('Pleae set Comndition in sql statement');
                insert into error values('Too many rows',sysdate);
        when no_data_found then
                dbms_output.put_line('No data returned from sql statement');
                insert into error values('no data found',sysdate);
        when zero_divide then
                dbms_output.put_line('Cant division on zero');
                insert into error values('Zero Divide',sysdate);
end;
```

شـــرح الـكود

Example (25) ::

```
exception
        when too_many_rows then
                dbms_output.put_line('Pleae set Comndition in sql statement');
                insert into error values('Too many rows',sysdate);
        when no_data_found then
                dbms_output.put_line('No data returned from sql statement');
                insert into error values('no data found', sysdate);
        when zero_divide then
                dbms_output.put_line('Cant division on zero');
                insert into error values('Zero Divide',sysdate);
end;
```

خاص بالتاريخ

إذا وجدنا خطا نعمل له (insert) حسب نوعه
في جدول الأخطاء الذي قمنا بإنشائه

Example (26) :: Exception

4. Others

```
declare
        v_lname employees.last_name%type;
begin
        select last_name into v_lname from employees ;
        dbms_output.put_line(v_lname);
        dbms_output.put_line(50/0);
exception
        when too_many_rows then
                dbms_output.put_line('Pleae set Comndition in sql statement');
                insert into error values('Too many rows',sysdate);
        when no_data_found then
                dbms_output.put_line('No data returned from sql statement');
                insert into error values('no data found',sysdate);
        when zero_divide then
                dbms_output.put_line('Cant division on zero');
                insert into error values('Zero Divide',sysdate);
        when others then
                dbms_output.put_line('Error');
end;
```

Example (26) :: شـــرح الـكود

```
exception
        when too_many_rows then
                dbms_output.put_line('Pleae set Comndition in sql statement');
                insert into error values('Too many rows',sysdate);
        when no_data_found then
                dbms_output.put_line('No data returned from sql statement');
                insert into error values('no data found',sysdate);
        when zero_divide then
                dbms_output.put_line('Cant division on zero');
                insert into error values('Zero Divide',sysdate);
        when others then
                dbms_output.put_line('Error');
```

هذا النوع :: يشمل جميع الأنواع الثلاثة السابقة

Exception

**تعريف الخطأ كمتغير

طريقة استخدام الأوامر التالية مع داخل (Exception)

** Raise
** Commit
** Rollback

Example (27) :: Exception

```
declare
        v_sal employees.salary%type;
        my_exp exception;
begin
        update employees set salary=salary*1.7 where employee_id=200;
        select salary into v_sal from employees where employee_id=200;
        if v_sal >5500 then
                raise my_exp;
        else
                dbms_output.put_line('The salary is updated');
                commit;
        end if;
exception
        when my_exp then
                dbms_output.put_line('The salary is more than 7500');
                rollback;
end;
```

شـــرح الـكود

Example (27) ::

**تعريف الخطأ كمتغير ::

يمكن تعريف متغير (خاص بالأخطاء) ضمن Declare

```
declare
        v_sal employees.salary%type;
        my_exp exception;
```

شـــرح الـكود

Example (27) ::

** Raise

```
if v_sal >5500 then
    raise my_exp;
```

إذا طلع الشرط (IF) خطأ يعمل إزاحة له(اعتراض)
إلى مكان الخاص بالأخطاء

شـــرح الـكود

Example (27) ::

** Commit

```
else
        dbms_output.put_line('The salary is updated');
        commit;
end if;
```

وظيفة (**commit**) إذا خرجنا من data base

يـحـتـفظ بجميع التحديثات و القيم

Example (27) :: شـــرح الـكود

** Rollback

```
exception
        when my_exp then
                dbms_output.put_line('The salary is more than 7500');
                rollback;
end;
```

وظيفة (**rollback**) إذا خرجنا من data base

يتراجع عن جميع التحديثات و القيم

الفصل الخامس

Procedure & Function

Procedure

الشكل العام

```
Create or replace procedure( Name of procedure )
(parameter )
```

تستطيع أن تختار أي وحدة ⟵ IS / AS

يكون Declare هنا بعد (IS / AS)

```
Begin
Exception
End (My procedure) ;
```

Procedure

التعريف

شاهدنا في الدروس الماضية ان اي اجراء نقوم بكتابة اني اذا اردت استخدامة اكثر من مرة فاني اقوم بكتابة كل مرة في * sql plus لكي احصل على النتائج لكن ماهو رأيك لو نقوم بتخزين هذا الاجراء في قاعدة البيانات ونعطية اسم وحينما نحتاجه نستدعية باسمه وهذا يوفر علينا الشيء الكثير لذلك درسنا هذا اليوم هو الاجرائيات المخزنة.

ولكي نقوم بانشاء اجراء مخزن نقوم بمايلي :

CREATE [OR REPLACE] PROCEDURE procedure_name(متغيرات الادخال الممررة ومتغيرات الاخراج)

حيث تمثل procedure_name اسم الاجراء المستخدم.

اما OR REPLACE فهي توضع حينما تعلم ان الاجراء موجود من السابق.

اما عن المغيرات التي بين القوسين فهي اما متغيرات مدخله مثل اذا كان لديك اجراء حساب معدل طالب وتريد تمرير رقم الطالب الذي تريد حساب معدله فهذه هي تعتبر كمدخلات ولتعريف متغير بهذا الشكل يكون كمايلي :

```
student_id  in number(9)
```

لاحظ اسم المتغير هو student_id ثم بعده وضعنا الكلمة in ومعنى ان هذا المتغير يعتبر كمدخل

Procedure

التعريف

اما لتعريف متغير يعود بقيمة من الاجراء مثلا لو اردنا تعرف متغبر يرجع بمعدل الطالب يتم التعريف كمايلي :

```
ave out number(5,2)
```

بعد تنفيذ الاجراء يكون هذا المتغير يحتوي على معدل الطالب الذي تم تمرير رقمه مثلا.

مع العلم انه يمكن تعريف متغير للمدخلات والمخرجات حيث تمرر به القيمه اولا وبعد تنفيذ الاجراء يتم وضع القيمه في نفس المتغير وتتم كمايلي :

```
ave in out number(5,2)
```

ومعنى هذا اي مدخل ومخرج في نفس الوقت .

Function

الشكل العام

```
Create or replace  Function( Name of Function)
(parameter )
Return ( data type of function )
```

تستطيع أن تختار أي وحدة ⟵ IS / AS

Declare يكون هنا بعد (IS / AS)

```
Begin
Exception
End (My Function ) ;
```

Function

التعريف

تحدثنا في الشريحة السابقة عن الإجراءات المخزنة و الآن نتحدث عن الوظائف المخزنة

لكن الفرق أن الوظائف المخزنة ترجع قيمة

الصيغة العامة كما يلي :

```
CREATE [OR REPLACE] FUNCTION function_name(الاخراج متغيرات الادخال الممررة ومتغيرات)
RETURN datatype
```

حيث تمثل function_name اسم الوظيفة المستخدمه.

اما REPLACE OR فهي توضع حينما تعلم ان الاجراء موجود من السابق.

اما عن المغيرات التي بين القوسين فهي اما متغيرات مدخله مثل اذا كان لديك اجراء حساب معدل طالب وتريد تمرير رقم الطالب الذي تريد حساب معدله فهذه هي تعتبر كمدخلات ، وهي بنفس الطريقة التي تعاملنا بها مع الاجراءت المخزنة لاتغير على المتغيرات وطرق تعريفها.اما RETURN datatype فهي تدل على نوع القيمة المعادة من الوظيفة .

Example (28) :: Procedure

```
create or replace procedure print_all_emp(   p_did
employees.department_id%type)
as
        cursor c is select first_name,last_name,salary
        from employees
        where department_id=p_did;
begin
        for v_rec in c loop
                dbms_output.put_line(v_rec.first_name||'
                        '||v_rec.last_name||'   '||v_rec.salary);
        end loop;
end print_all_emp;
```

Example (28) :: شـــرح الـكود

```
create or replace procedure print_all_emp
( p_did employees.department_id%type );
```

نوع متغير (Procedure)

اسم (Procedure)

Example (29) ::

Procedure

```
create table circle
( id number,
r number);
```

عرفنا جدول جديد للدائرة

```
create or replace procedure circle_proc (p_id number)
 is
        v_r number;
begin
        select r into v_r
        from circle where id=p_id;
        dbms_output.put_line('Muhit : '||(2*3.14*v_r));
        dbms_output.put_line('Area : '||(v_r*v_r*3.14));
exception
        when no_data_found      then
                dbms_output.put_line('The circle you selected
no exists');
end circle_proc;
```

Example (29) :: شـــرح الـكود

```
create or replace procedure circle_proc (p_id number)
 is
        v_r number;
begin
        select r into v_r
        from circle where id=p_id;
        dbms_output.put_line(' Muhit : '||(2*3.14*v_r));
        dbms_output.put_line('Area : '||(v_r*v_r*3.14));
exception
        when no_data_found      then
        dbms_output.put_line('The circle you selected no exists');
end circle_proc;
```

محيط الدائرة

Example (30) ::

Function

```
create or replace function getR(p_id number)
return number
is
        v_r number;
begin
        select r into v_r
        from circle
        where id=p_id;
        return v_r;
exception
        when no_data_found      then
                return 0;
end getR;
```

Example (31) :: Procedure

```
create or replace procedure circle_proc( p_id number)
 is
        v_r number;
 begin
        v_r := getR( p_id );

        dbms_output.put_line('Muhit : '||(2*3.14*v_r));
        dbms_output.put_line('Area : '||(v_r*v_r*3.14));
end ;
```

قمنا بإسناد قيمة المتغير الجديد (v_r) إلى (Function)

السابقة التي قمنا بتعريفها في **(Example 30)**

Example (32) ::

Function

```
create or replace function getDid (p_eid number)
return number
is
        v_did number;
begin
        select department_id into v_did
        from employees
        where employee_id=p_eid;
        return v_did;
exception
        when no_data_found then
                return 0;
end getDid;
```

Example (33) :: Procedure

```
create or replace procedure updateEmpSal(p_eid number)
is
        v_did number;
begin
        v_did :=getDid(p_eid);
        if v_did!=0 then
                if(v_did=50) then
update employees set salary=salary*1.5 where employee_id=p_eid;
                else
update employees set salary=salary*1.2 where employee_id=p_eid;
                end if;
        else
                dbms_output.put_line('Employee not found');
        end if;
end     updateEmpSal;
```

Example (34) :: Procedure

NO_STU	COURSE_CODE	MARK	POINT
111	216CS	88	13.5
222	225CS	75	10.5
333	225CS	40	3
111	225CS	90	14.25
222	216CS	78	10.5
333	216CS	85	13.5

لو اردنا تصميم وظيفة ترجع بمعدل الطالب الفصل اي يتم تمرير رقم الطالب الى الوظيفة ثم يتم حساب المعدل الفصلي للطالب

ويتم حساب المعدل الفصل للطالب كمايلي =مجموع النقاط ÷ مجموع عدد الساعات لمقررات

Example (34) :: Procedure

```
1  create or replace function stu_avea(stnum in studys.NO_STU%type)
2  return real
3  as
4  hour courses.hours%type;
5  avrage number(4,2);
6  sum_hours courses.hours%type:=0;
7  point studys.POINT%type;
8  total_Point studys.POINT%type:=0;
9  codem courses.CODE%type;
10 cursor sumpoint
11 is
12 select COURSE_CODE,POINT
13 from studys
14 where NO_STU=stnum;
15 begin
16 open sumpoint;
17 loop
18 fetch sumpoint into codem,point;
```

Example (34) ::

Procedure

```
19 exit when sumpoint%notfound;
20 select hours
21 into hour
22 from courses
23 where code=codem;
24 total_Point:=total_Point+point;
25 sum_hours:=sum_hours+hour;
26 end loop;
27 close sumpoint;
28 avrage:=total_Point/sum_hours;
29 return avrage;
30 end;
```

شـــرح الـكود

Example (34) ::

السطر رقم ١: لتعريف الوظيفة
السطر رقم ٢: نوع القيمة التي سوف ترجع بها الوظيفة
السطر رقم ٤: تعريف متغير عدد الساعات وهو نفس حقل عدد ساعات المقرر الموجودة في جدول courses
السطر رقم ٥: تعريف متغير الذي سوف نضع به المعدل
السطر رقم ٦: تعريف متغير لكي يوضع به مجموعات الساعات للطلاب في كل المواد
السطر رقم ٧: تعريف متغير لكي يوضع به عدد نقاط الطلاب في اي مقرر
السطر رقم ٨: تعريف متغير لكي يوضع به مجموع عدد نقاط الطالب في كل المقرر
السطر رقم ٩: تعريف متغير لكود المادة
السطر رقم ١٠: تعريف مؤشر صريح للحصول على كود المادة لكي نستفيد منه في الحصول على عدد الساعات وعدد النقاط في ذلك المقرر لكي نضيفها الى مجموع النقاط
السطر رقم ١٦: فتح هذا المؤشر لكي نتعامل معه
السطر رقم ١٧: الدخول على حلقة لكي نمر على جميع الجدول
السطر رقم ١٨: تحديث قيم المؤشر للسجل الحالي في المتغيرات codem,point
السطر رقم ١٩: شرط انهاء الحلقة وهو اذا لم يجد اي سجل في المؤشر
السطر رقم ٢٠: مؤشر ضمني لكي يقوم بالحصول على عدد ساعات الطلاب في المقرر الموجود حاليا في المؤشر الصريح ويضع عدد الساعات فى المتغير hour

شـــرح الـكود

Example (34) ::

السطر رقم ٢٤: اضافة عدد النقاط للمقرر الحالي الى مجموع النقاط السابق
السطر رقم ٢٥: اضافة عدد الساعات للمقرر الحالي الى مجموع الساعات السابق
السطر رقم ٢٦: الخروج من الحلقة
السطر رقم ٢٧: انهاء المؤشر الضمني
السطر رقم ٢٨: حساب المعدل وهو مجموع النقاط تقسيم مجموع عدد الساعات
السطر رقم ٢٩: الرجوع بقيمة المعدل

تم بحمد الله