



Tomislav Kušanić

Back to the basics:

FOR LOOP revisited

Who are we?

Leading IT service
provider for public and
private sector in Croatia.
We are part of the
Constellation Software
Incorporated group.

30+
years of
experience

500+
employees

350+
clients

650+
projects

Constellation Software Inc. among the top 10 most valuable software in the world

Posted on 30/08/2022

TOP 10 MOST VALUABLE SOFTWARE BRANDS 2022

1 

 Microsoft

\$184.2bn

2 

 ORACLE

\$29.1bn

3 

 SAP

\$18.3bn

4 

 salesforce

\$17.9bn

5 

 Adobe

\$14.1bn

6 

 vmware

\$7.0bn

7 

 servicenow

\$5.2bn

8 

 CONSTELLATION
SOFTWARE
INC.

\$4.0bn

9 

 quickbooks

\$3.8bn

10 

 Red Hat

\$3.7bn

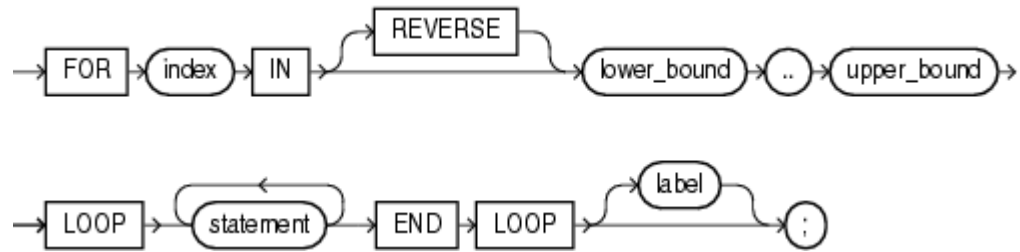
About me

- Tomislav Kušanić
 - tomislav.kusanic@in2.hr
- 15 years of experience with Oracle (E-Business Suite, Apex, BI)

Today's topic

- FOR LOOPS prior to 21c and what new features awaits us in 21c

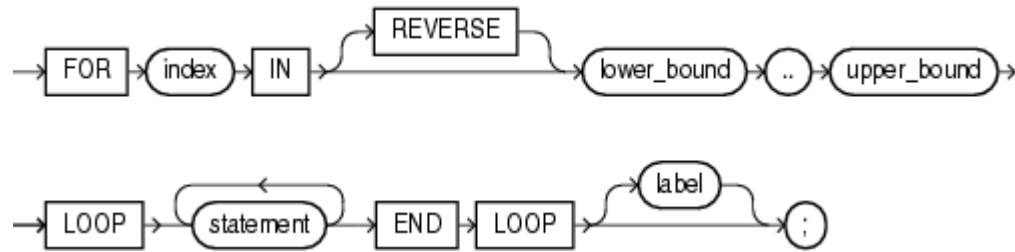
FOR LOOP before 21c



index

- implicitly declared integer variable
- local to FOR LOOP statement
- can be read but cannot be changed within FOR LOOP
- increments automatically by 1

FOR LOOP before 21c



lower/upper bound

- must evaluate to NUMBER
- can be:
 - numeric literals
 - numeric variables
 - numeric expressions
- evaluated once at the beginning of FOR LOOP
- stored as temporary PLS_INTEGER (rounded to nearest integer if necessary)
- if lower_bound = upper_bound statements run only once

FOR LOOP before 21c

declare

```
type t_collection is table of varchar2(25) index by pls_integer;
```

```
l_full          t_collection;
```

begin

```
l_full(1) := 'Absolute Zero';
```

```
l_full(2) := 'Acid green';
```

```
l_full(3) := 'Aero';
```

```
l_full(4) := 'Aero blue';
```

```
l_full(5) := 'African violet';
```

<<FOR LOOP>>

end;

FOR LOOP before 21c

<<FOR LOOP>>

```
for i in 1..5
loop
  dbms_output.put_line(lpad(to_char(i),2,'0') || '-' || l_full(i));
end loop;
```

- 01 - Absolute Zero
- 02 - Acid green
- 03 - Aero
- 04 - Aero blue
- 05 - African violet

FOR LOOP before 21c

<<FOR LOOP>>

```
for i in 1..l_full.count
loop
  dbms_output.put_line(lpad(to_char(i),2,'0') || '-' || l_full(i));
end loop;
```

- 01 - Absolute Zero
- 02 - Acid green
- 03 - Aero
- 04 - Aero blue
- 05 - African violet

FOR LOOP before 21c

```
function upper  
return number  
is  
begin  
    return 5;  
end upper;
```

01 - Absolute Zero
02 - Acid green
03 - Aero
04 - Aero blue
05 - African violet

<<FOR LOOP>>

```
for i in 1..upper  
loop  
    dbms_output.put_line(lpad(to_char(i),2,'0') || ' -  
' || l_full(i));  
end loop;
```

FOR LOOP before 21c

```
l_lower number := 1;  
l_upper number := 5;
```

<<FOR LOOP>>

```
for i in l_lower..l_upper  
loop  
  dbms_output.put_line(lpad(to_char(i),2,'0') || ' -'  
' || l_full(i));  
end loop;
```

01 - Absolute Zero
02 - Acid green
03 - Aero
04 - Aero blue
05 - African violet

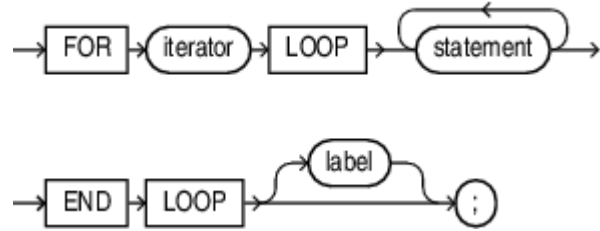
FOR LOOP before 21c

<<FOR LOOP>>

```
for i in reverse 1..5
loop
  dbms_output.put_line(lpad(to_char(i),2,'0') || '-' || l_full(i));
end loop;
```

- 05 - African violet
- 04 - Aero blue
- 03 - Aero
- 02 - Acid green
- 01 - Absolute Zero

FOR LOOP in 21c - iterator

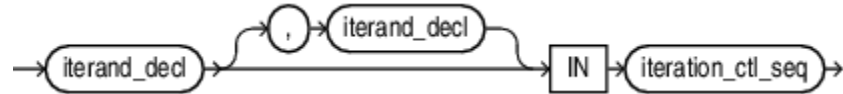


iterator

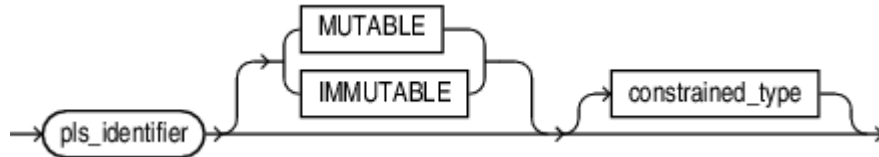
- specifies iterand and iteration controls
- local to FOR LOOP statement
- can be read but cannot be changed within FOR LOOP*

FOR LOOP in 21c – iterand declaration

iterator



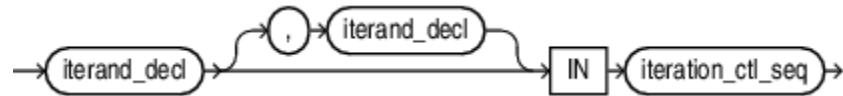
iterand declaration



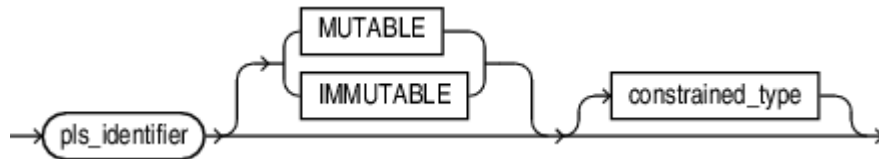
- iterand type can be implicitly or explicitly declared
- cannot be explicitly initialized
- implicitly declared if no type declaration follows iterand in LOOP header.
Implicit type is determined by first iteration control

FOR LOOP in 21c – implicit iterand types

iterator



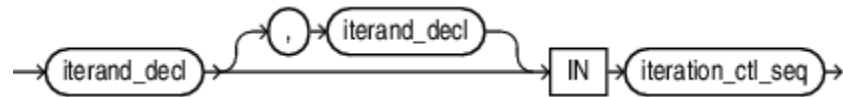
iterand declaration



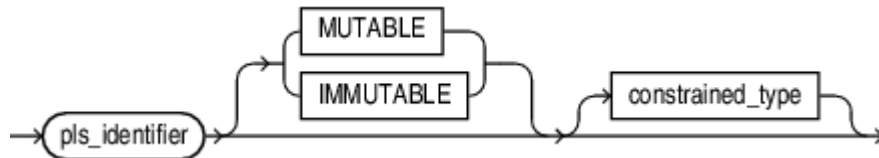
| Iteration Control | Implicit Iterand Type |
|--------------------|---|
| Stepped control | PLS_INTEGER |
| Single expression | PLS_INTEGER |
| Cursor control | CURSOR%ROWTYPE |
| VALUES OF control | Collection element type |
| INDICES OF control | Collection index type |
| PAIRS OF control | First iterand denotes index type of collection Second iterand denotes element type of collection |

FOR LOOP in 21c – iterand mutability

iterator



iterand declaration



MUTABLE | IMMUTABLE

- mutability property determines whether iterand value can be changed in the loop body
- if all iteration controls are cursor controls iterand is mutable by default
- Otherwise immutable
- Default mutability can be changed with MUTABLE | IMMUTABLE
- any modification to iterand for stepped range iteration control or repeated single expression iteration control will likely affect the behaviour of that control and the sequence of the values it produces

FOR LOOP in 21c – iterand mutability

```
begin
  for i in 1..5
  loop
    i := i + 1;
    dbms_output.put_line(i);
  end loop;
end;
```

ORA-06550: line 4, column 9:

PLS-00363: expression 'i' cannot be used as an assignment target

ORA-06550: line 4, column 9:

PL/SQL: Statement ignored

FOR LOOP in 21c – iterand mutability

```
begin
  for i mutable in 1..5
  loop
    i := i + 1;
    dbms_output.put_line(i);
  end loop;
end;
```

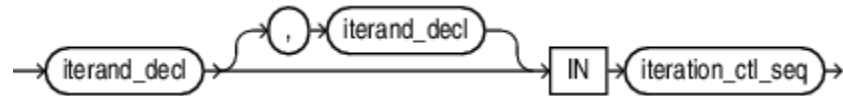
2

4

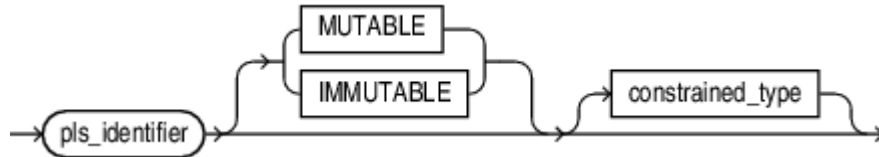
6

FOR LOOP in 21c – constrained type

iterator



iterand declaration



constrained_type

- iterand type is explicitly declared when iterand type is specified in the loop header
- Constrain defined for a type is considered when assigning values to the iterand.
- Values generated by the iteration controls must be assignment compatible with the iterand type

FOR LOOP in 21c – constrained type

```
begin
  for i in 1.25..5.25
  loop
    dbms_output.put_line(i);
  end loop;
end;
```

1
2
3
4
5

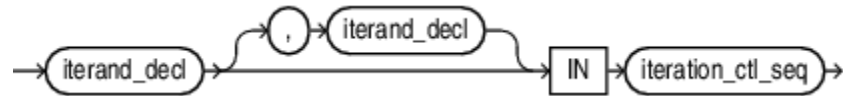
FOR LOOP in 21c – constrained type

```
begin
  for i number(5,2) in 1.25..5.25
  loop
    dbms_output.put_line(i);
  end loop;
end;
```

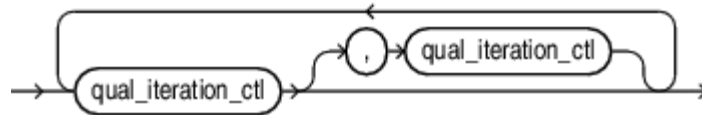
```
1.25
2.25
3.25
4.25
5.25
```

FOR LOOP in 21c – iteration control sequence

iterator



iteration control sequence



- multiple iteration controls may be chained together by separating them with commas
- **pairs of** iteration controls may not be mixed with other kinds of iteration controls because two iterands are required

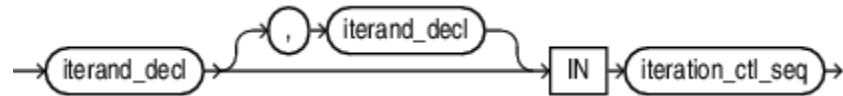
FOR LOOP in 21c – iteration control sequence

```
begin
  for i in 1..3,11..13,101..103
  loop
    dbms_output.put_line(i);
  end loop;
end;
```

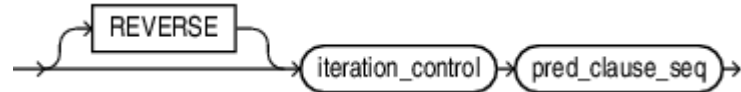
```
1
2
3
11
12
13
101
102
103
```


FOR LOOP in 21c – iteration control - reverse

iterator



qualified iteration control



REVERSE

- reverses order of values in the sequence
- Cannot be used:
 - when pipelined function is specified in iteration control
 - with single expression iteration since it generates single value
 - when iteration control specifies a SQL statement. Instead use ORDER BY
 - when collection is a cursor, cursor variable, dynamic SQL or is expression that calls pipelined table function

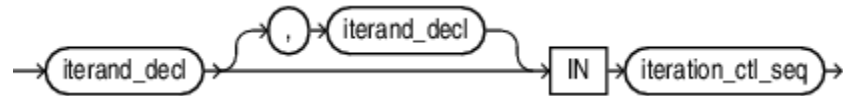
FOR LOOP in 21c – iteration control - reverse

```
begin
  for i in 1..3,reverse 11..13,101..103
  loop
    dbms_output.put_line(i);
  end loop;
end;
```

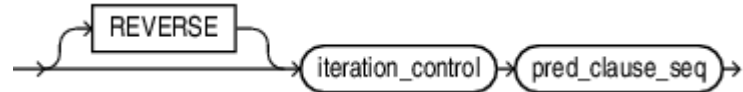
```
1
2
3
13
12
11
101
102
103
```

FOR LOOP in 21c – iteration control

iterator

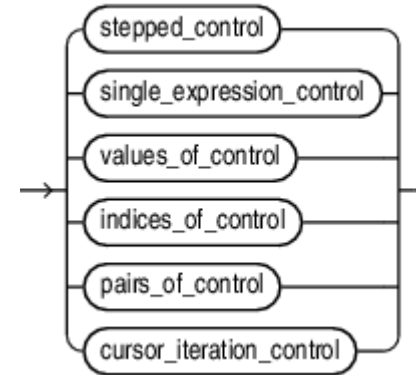


qualified iteration control



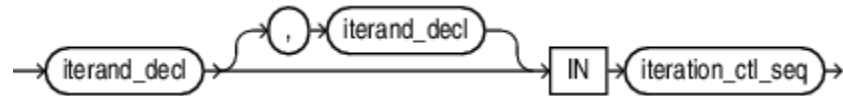
iteration control

- provides sequence of values to iterand

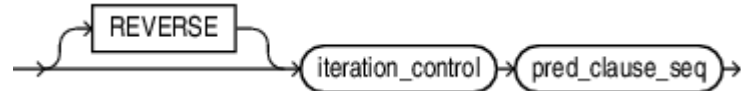


FOR LOOP in 21c – stepped control

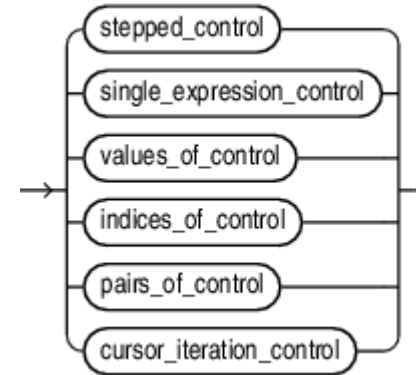
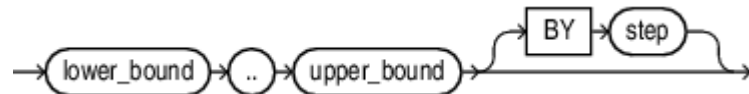
iterator



qualified iteration control



stepped_control



FOR LOOP in 21c – stepped control

```
begin
  for i in 1..10 by 3
    loop
      dbms_output.put_line(i);
    end loop;
end;
```

```
1
4
7
10
```

FOR LOOP in 21c – stepped control

```
begin
  for i number(5,2) in 1.25..2.25 by 0.25
  loop
    dbms_output.put_line(i);
  end loop;
end;
```

1.25

1.5

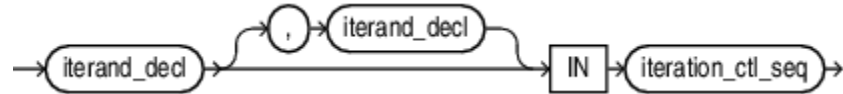
1.75

2

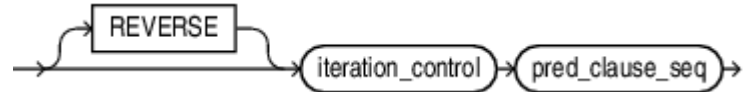
2.25

FOR LOOP in 21c – single expression control

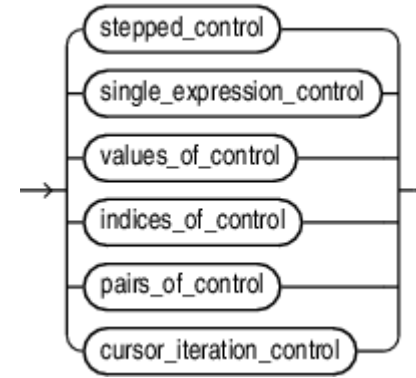
iterator



qualified iteration control



single expression control



FOR LOOP in 21c – single expression control

```
begin  
  for i in 5  
  loop  
    dbms_output.put_line(i);  
  end loop;  
end;
```

5

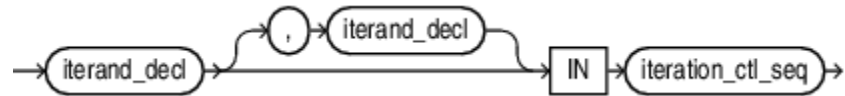
FOR LOOP in 21c – single expression control

```
BEGIN
  for i in 5, repeat i*2 while i<100
  loop
    dbms_output.put_line(i);
  end loop;
end;
```

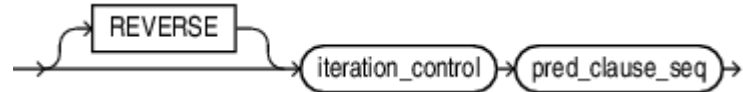
```
5
10
20
40
80
```

FOR LOOP in 21c – values of control

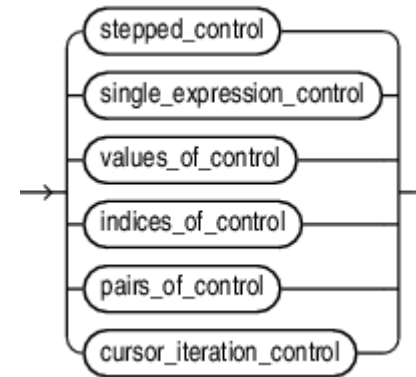
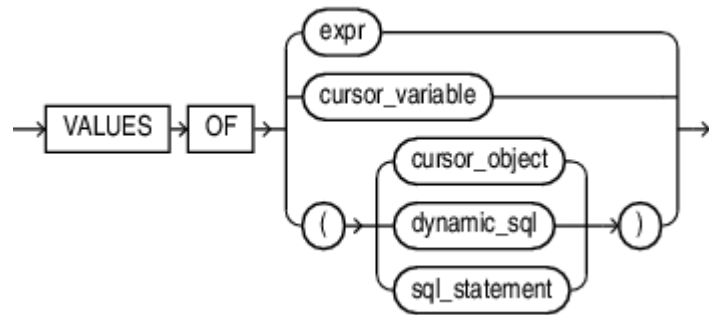
iterator



qualified iteration control



values of control



FOR LOOP in 21c – values of control

declare

```
type t_delivery_address is record(city varchar2(25)
                                ,country varchar2(25)
                                );
```

```
type t_destinations is table of t_delivery_address index by pls_integer;
```

```
l_countries t_destinations;
```

begin

```
l_countries(1).city := 'Zagreb';
l_countries(1).country := 'HR';
```

```
l_countries(2).city := 'Rovinj';
l_countries(2).country := 'HR';
```

```
l_countries(3).city := 'Split';
l_countries(3).country := 'HR';
```

```
l_countries(4).city := 'Arnhem';
l_countries(4).country := 'NL';
```

```
l_countries(5).city := 'Middelburg';
l_countries(5).country := 'NL';
```

```
l_countries(6).city := 'Utrecht';
l_countries(6).country := 'NL';
```

```
l_countries(7).city := 'Maastricht';
l_countries(7).country := 'NL';
```

FOR LOOP in 21c – values of control

```
for r in values of l_countries  
loop  
    dbms_output.put_line(r.city);  
end loop;
```

Zagreb

Rovinj

Split

Arnhem

Middelburg

Utrecht

Maastricht

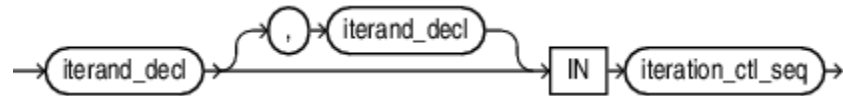
FOR LOOP in 21c – values of control

```
for r in values of l_countries when r.country = 'NL'  
  loop  
    dbms_output.put_line(r.city);  
  end loop;
```

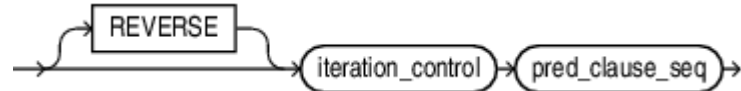
Arnhem
Middelburg
Utrecht
Maastricht

FOR LOOP in 21c – indices of control

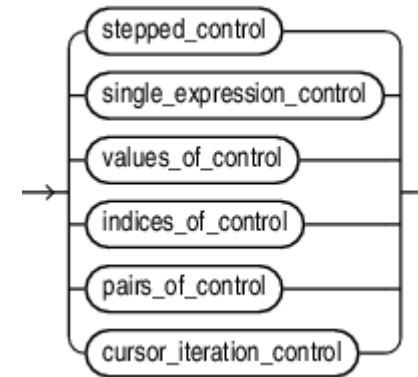
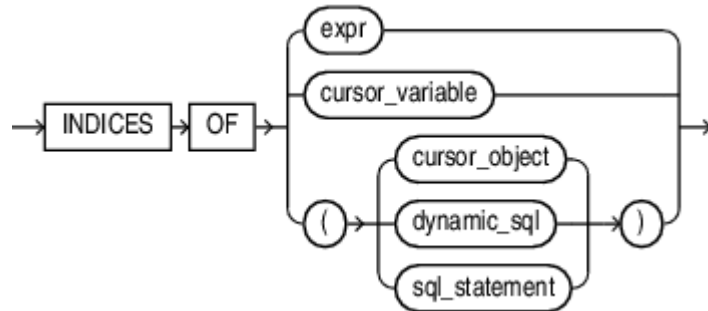
iterator



qualified iteration control



indices of control



FOR LOOP in 21c – indices of control

```
for i in indices of l_countries  
loop  
    dbms_output.put_line(i);  
end loop;
```

```
1  
2  
3  
4  
5  
6  
7
```

FOR LOOP in 21c – indices of control

```
for i in indices of I_countries when i.country = 'NL'  
loop  
  dbms_output.put_line(i);  
end loop;
```

ORA-06550: line 29, column 43:

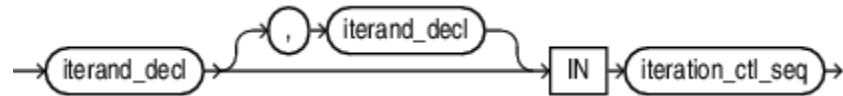
PLS-00487: Invalid reference to variable 'I'

ORA-06550: line 29, column 4:

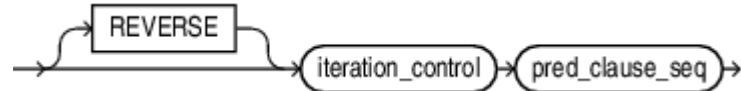
PL/SQL: Statement ignored

FOR LOOP in 21c – pairs of control

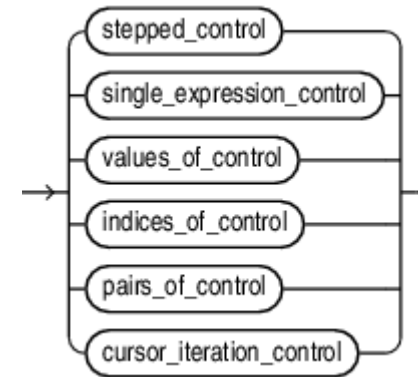
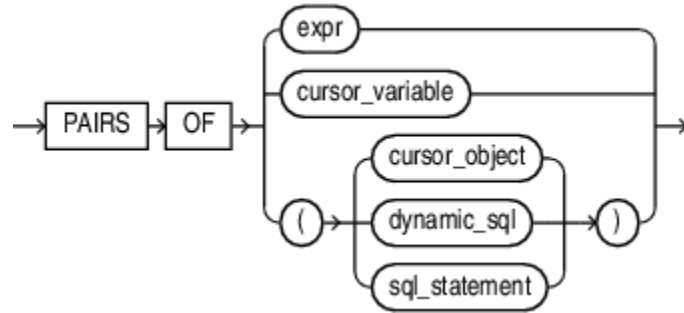
iterator



qualified iteration control



pairs of control



FOR LOOP in 21c – pairs of control

```
for i,city in pairs of l_countries
loop
  dbms_output.put_line(i || ':' || city.country || '-' || city.city);
end loop;
```

- 1: HR-Zagreb
- 2: HR-Rovinj
- 3: HR-Split
- 4: NL-Arnhem
- 5: NL-Middelburg
- 6: NL-Utrecht
- 7: NL-Maastricht

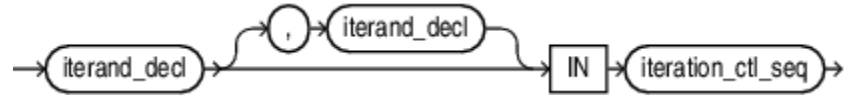
FOR LOOP in 21c – pairs of control

```
for i,city in pairs of l_countries when city.country = 'NL'  
  loop  
    dbms_output.put_line(i || ':' || city.country || '-' || city.city);  
  end loop;
```

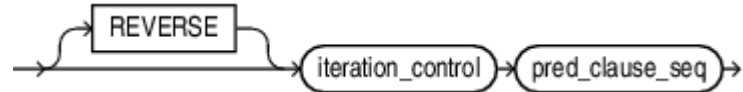
4: NL-Arnhem
5: NL-Middelburg
6: NL-Utrecht
7: NL-Maastricht

FOR LOOP in 21c - iterator

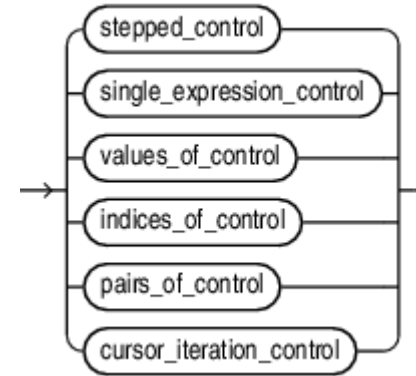
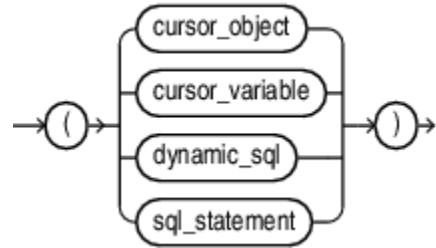
iterator



qualified iteration control



cursor iteration control



FOR LOOP in 21c – cursor_iteration_control

```
begin
  for city in (select distinct cust_city
              from   sh.customers
              where  cust_state_province = 'England - Norfolk')
  loop
    dbms_output.put_line(city.cust_city);
  end loop;
end;
```

Diss
Downham Market
Wymondham
Norwich
Great Yarmouth
King's Lynn

FOR LOOP in 21c – cursor_iteration_control

```
create or replace function get_customers(p_cust_state_province in sh.customers.cust_state_province%type)
return sys_refcursor
is
c_customers sys_refcursor;
begin
open c_customers for
SELECT distinct cust_first_name
, cust_last_name
, cust_gender
, extract(year from sysdate) - cust_year_of_birth age
, cust_marital_status
FROM sh.customers
WHERE cust_state_province = p_cust_state_province;

return c_customers;
end;
```

FOR LOOP in 21c – cursor_iteration_control

declare

```
type t_cust_rec is record(first_name sh.customers.cust_first_name%type
    ,last_name sh.customers.cust_last_name%type
    ,gender sh.customers.cust_gender%type
    ,age number
    ,marital_status sh.customers.cust_marital_status%type);
```

```
l_customers sys_refcursor;
l_customers2 sys_refcursor;
```

begin

```
l_customers := get_customers(p_cust_state_province => 'England - Norfolk');
l_customers2 := get_customers(p_cust_state_province => 'England - Norfolk');
for rc t_cust_rec in values of l_customers when rc.age = 48, l_customers2 when rc.age = 50
loop
    dbms_output.put_line(rpad(rc.first_name || ' ' || rc.last_name ,25) || rc.age || rc.gender);
end loop;
close l_customers;
end;
```

FOR LOOP in 21c – cursor_iteration_control

| | |
|-------------------|-----|
| Henry Lincoln | 48M |
| Orilla Lauderdale | 48F |
| Royal Underhill | 48M |
| Titania Feathers | 48F |
| Wilmet Lickey | 48M |
| ... | |
| Zel Livesay | 50M |
| Rosalie Kuehen | 50F |
| Yola Joseph | 50F |
| Dinah Grimshaw | 50F |
| Trent Roman | 50M |
| Val Littlefield | 50F |
| Rollo Rider | 50M |
| Theodora Links | 50F |

FOR LOOP in 21c – sparse collections

```
DECLARE
```

```
  type t_cities is table of varchar2(25) index by pls_integer;
```

```
  l_cities t_cities;
```

```
begin
```

```
  l_cities(1) := 'Dubrovnik';
```

```
  l_cities(5) := 'Rovinj';
```

```
  l_cities(55) := 'Osijek';
```

```
  for i,c in pairs of l_cities
```

```
  loop
```

```
    dbms_output.put_line(lpad(i,2) || '- ' || c);
```

```
  end loop;
```

```
end;
```

FOR LOOP in 21c – sparse collection

1 - Dubrovnik

5 - Rovinj

55 - Osijek

FOR LOOP in 21c – character index

```
declare
```

```
  type t_vendor_info is record(name  varchar2(50)  
                               ,address varchar2(150)  
                               );
```

```
  type t_vendors is table of t_vendor_info index by varchar2(15);
```

```
  l_vendors t_vendors;
```

```
BEGIN
```

```
  l_vendors('HR69401829750').name  := 'Plinacro';
```

```
  l_vendors('HR69401829750').address := 'Savska cesta 88a, 10000 Zagreb, Republika Hrvatska';
```

```
  l_vendors('HR68195665956').name  := 'IN2 d.o.o.';
```

```
  l_vendors('HR68195665956').address := 'Marohničeva 1/1, 10000 Zagreb, Republika Hrvatska';
```

```
  l_vendors('US77-0467272').name  := 'Netflix Inc';
```

```
  l_vendors('US77-0467272').address := '100 WINCHESTER CIRCLE,LOS GATOS,CA-95032, USA';
```

```
  for trn,vendor in pairs of l_vendors
```

```
  loop
```

```
    dbms_output.put_line(trn || ' - ' || vendor.name);
```

```
  end loop;
```

```
end;
```

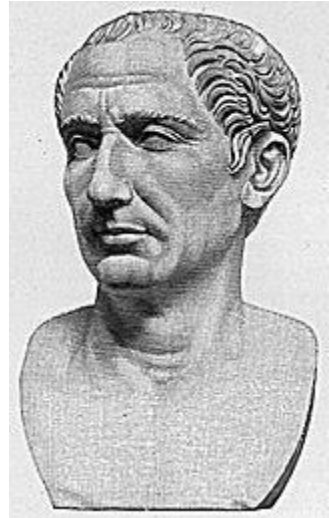
FOR LOOP in 21c – character index

HR68195665956 - IN2 d.o.o.

HR69401829750 - Plinacro

US77-0467272 - Netflix Inc

FOR LOOP in 21c – looping through dates?



Gaius Julius Caesar 12 July 100 BC – 15 March 44 BC
JULIAN CALENDAR 8AD

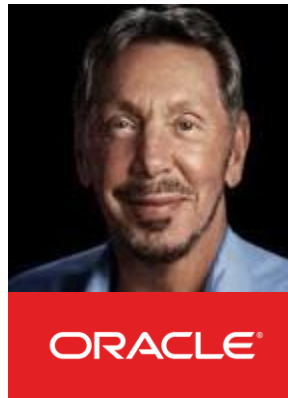
FOR LOOP in 21c – looping through dates?



Joseph Justus Scaliger 5 August 1540 – 21 January 1609

Julian Period > Julian Date - number of days since 01 January 4712 BC

FOR LOOP in 21c – looping through dates?



Larry Ellison 17 August 1944

ORACLE Date format Julian: `to_char(sysdate,'j')`

```
select to_char(to_date('17.08.1944','dd.mm.yyyy'),'j')  
from dual
```

Borned 2.431.320 days after 01 January 4712 BC

FOR LOOP in 21c – looping through dates!

```
declare
```

```
l_date date;  
l_julian_date number;  
l_new_date date;
```

```
function to_julian(p_date in date)
```

```
return number
```

```
is
```

```
begin
```

```
    return to_number(to_char(p_date,'j'));
```

```
end to_julian;
```

```
function from_julian(p_number in number)
```

```
return date
```

```
is
```

```
begin
```

```
    return to_date(to_char(p_number),'j');
```

```
end from_julian;
```


FOR LOOP in 21c – looping through dates!

```
begin
```

```
for datum number in to_julian(to_date('01.01.2022','DD.MM.YYYY'))..to_julian(to_date('01.02.2022','DD.MM.YYYY')) by 5  
loop  
    dbms_output.put_line(to_char(from_julian(datum),'DD.MM.YYYY'));  
end loop;
```

```
end;
```

```
01.01.2022  
06.01.2022  
11.01.2022  
16.01.2022  
21.01.2022  
26.01.2022  
31.01.2022
```

- <https://docs.oracle.com/en/database/oracle/oracle-database/21/Inpls/iterator.html#GUID-BD211E6F-8B4A-494A-AECF-AC26A241FF98>

