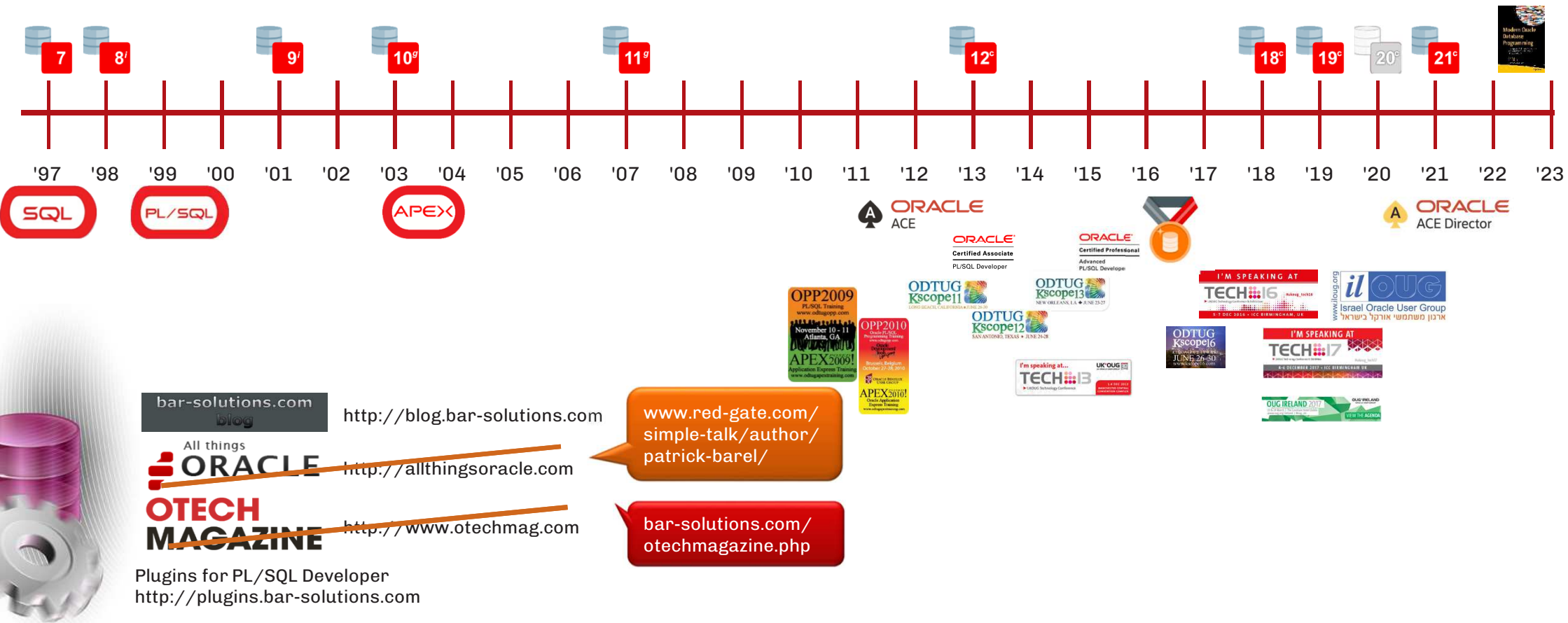
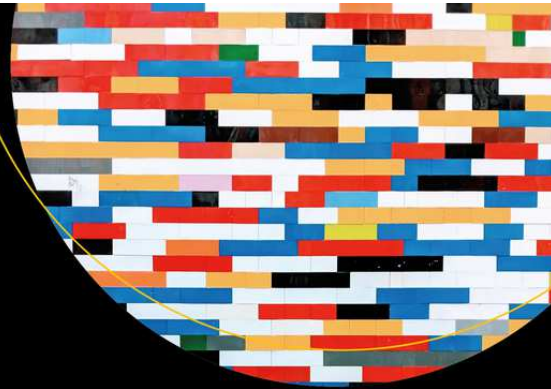
 PBarel@Qualogy.com

<http://blog.bar-solutions.com>



About me...





Modern Oracle Database Programming

Level Up Your Skill Set to Oracle's Latest
and Most Powerful Features in SQL,
PL/SQL, and JSON

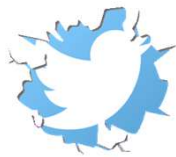
Alex Nuijten
Patrick Barel

Foreword by Chris Saxon

Apress®



Contact me...



@patch72



PBarel@Qualogy.com

Patrick.Barel@GMail.com

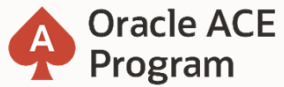
patrick@bar-solutions.com



Patrick.Barel@GMail.com



Patrick Barel



500+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community



3 membership tiers



For more details on Oracle ACE Program:
ace.oracle.com



Nominate

~~yourself~~ or someone you know:

ace.oracle.com/nominate

Connect:  aceprogram_ww@oracle.com

 Facebook.com/OracleACEs

 [@oracleace](https://twitter.com/oracleace)



Oracle Cloud Infrastructure

New Free Tier

oracle.com/cloud/free

Always Free

Services you can use for unlimited time

+

30-Day Free Trial

Free credits you can use for more services





Mentor and Speaker Hub

Our goal is to *connect* speakers with mentors to assist in *preparing* technical sessions and *improving* presentation skills

Interested? Read more and get in touch

<https://mashprogram.wordpress.com>

SYMPOSIUM 42

Created by the community, to support the community

Sharing of reliable knowledge

Supporting the various user groups and individuals



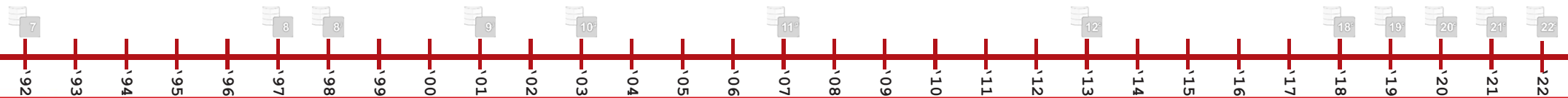
@sym_42



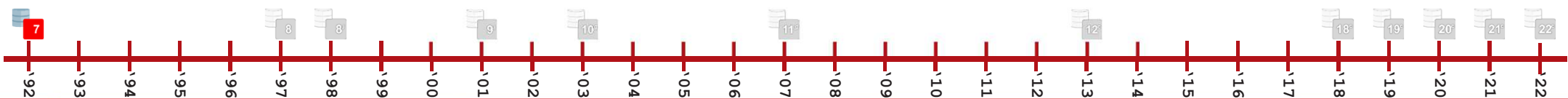
<https://sym42.org/>

A collection's a collection
No matter how small





Oracle 7



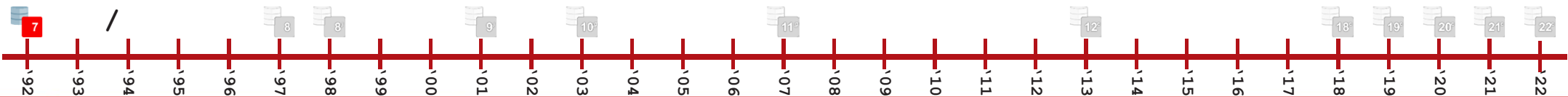




Oracle 7

declare

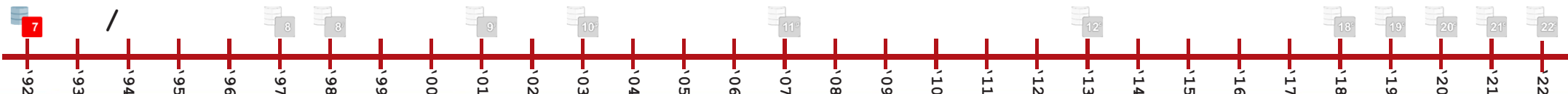
```
type names_pt is table of varchar2(30) index by binary_integer;  
l_names names_pt;  
l_indx binary_integer;  
begin  
  -- fill up the collection  
  l_names(1) := 'Hooly';  
  l_names(42) := 'Heddy';  
  l_names(10) := 'Hilder';  
  l_names(l_names.count + 1) := 'Holy';  
  l_names(-1) := 'Haley';  
  -- display the contents of the collection  
  l_indx := l_names.first;  
  while l_indx is not null loop  
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));  
    l_indx := l_names.next(l_indx);  
  end loop;  
end;
```



Oracle 7

declare

```
type names_pt is table of varchar2(30) index by binary_integer;
l_names names_pt;
l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```



Oracle 7

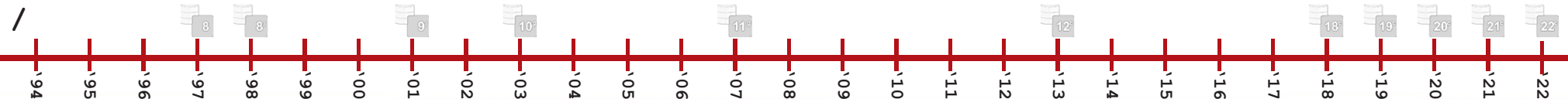
```
declare
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```

1 Hooly



Oracle 7

```
declare
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```



1 Hooly

42 Heddy

Oracle 7

```
declare
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```

10	Hilder
----	--------

1	Hooly
42	Heddy



Oracle 7

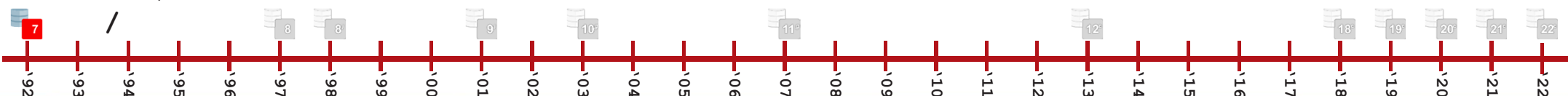
```

declare
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;

```

4 Holy

1	Hooly
10	Hilder
42	Heddy



Oracle 7

```
declare
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```

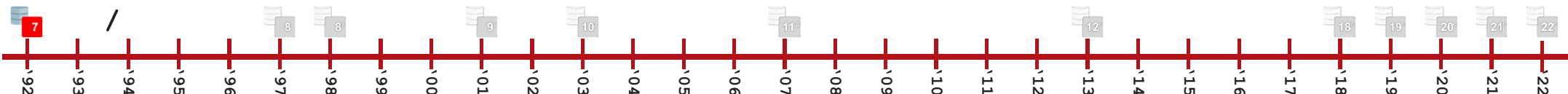
-1	Haley	1	Hooly
		4	Holy
		10	Hilder
		42	Heddy



Oracle 7

```
declare
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```

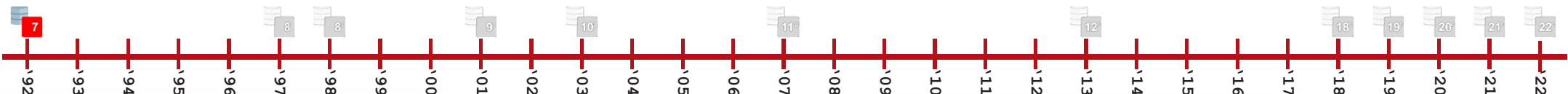
-1	Haley
1	Hooly
4	Holy
10	Hilder
42	Heddy

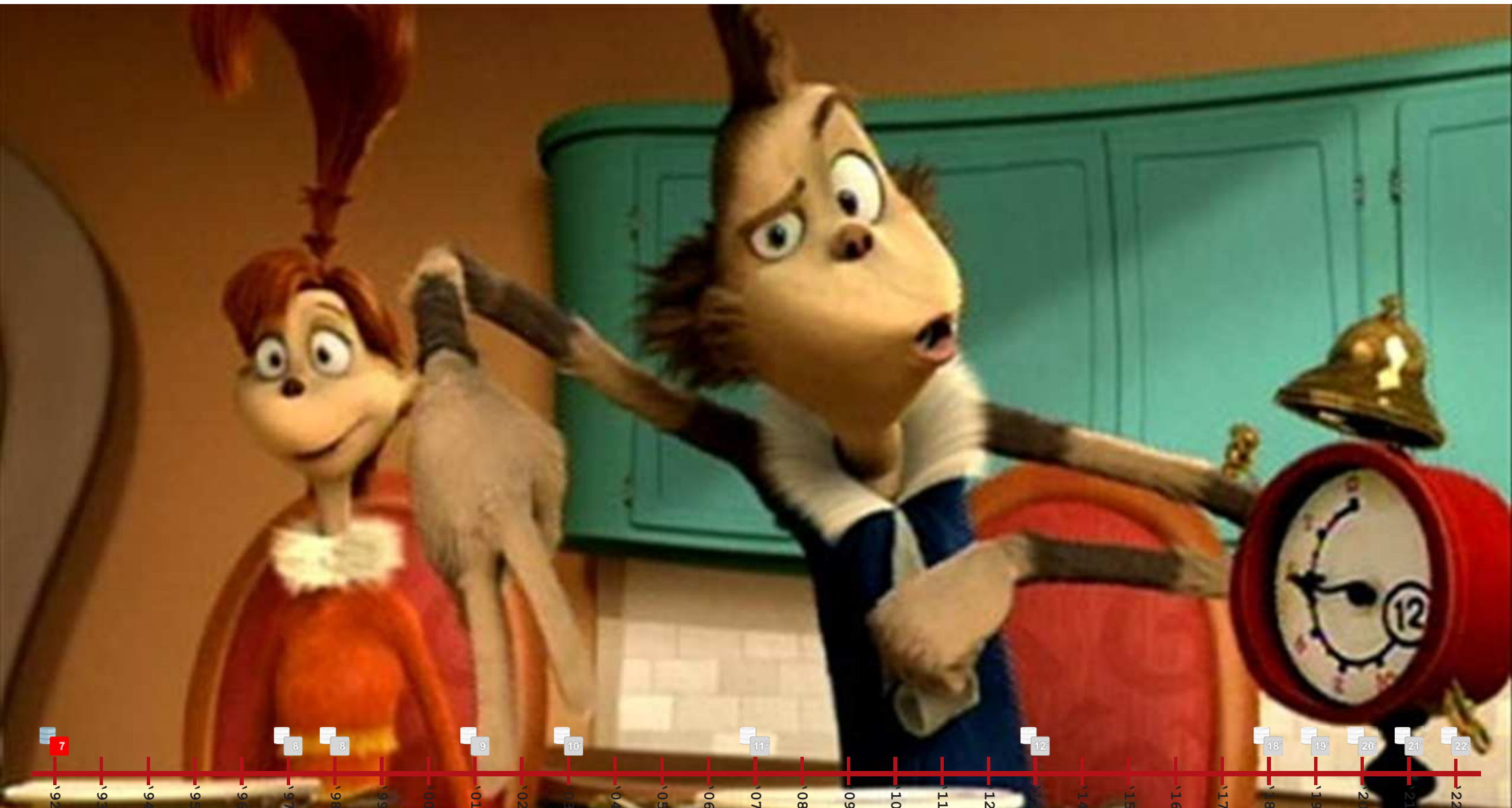


Oracle 7

```
C:\ORANT\BIN\PLUS33.exe
SQL*Plus: Release 3.3.2.0.2 - Production on Fri Jan 31 08:57:05 2020
Copyright (c) Oracle Corporation 1979, 1994. All rights reserved.

Connected to:
Oracle7 Workgroup Server Release 7.3.2.2.1 - Production Release
PL/SQL Release 2.3.2.2.0 - Production
SQL> _
```





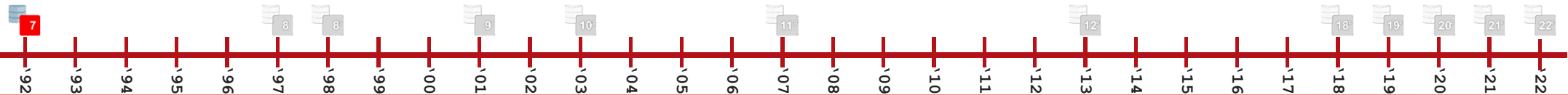
Oracle 7

prompt create the RelationTypes table

```
create table RelationTypes(  
  Type number(2)  
  , Name varchar2(16)  
)  
/
```

prompt create the NedMcDodd table

```
create table NedMcDodd(  
  ID          number(4)  
  , Firstname varchar2(32)  
  , Lastname  varchar2(32)  
  , Type      number(2)  
)  
/
```



prompt add the Relation Types to the table

begin

```
insert into RelationTypes(Type, Name) values (10, 'Relative');
insert into RelationTypes(Type, Name) values (12, 'Wife');
insert into RelationTypes(Type, Name) values (15, 'Son');
insert into RelationTypes(Type, Name) values (18, 'Daughter');
insert into RelationTypes(Type, Name) values (20, 'Friend');
insert into RelationTypes(Type, Name) values (30, 'Enemy');
commit;
```

end;

/

prompt add the characters to the NedMcDodd table

begin

-- Relative

```
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1010, 'Sally', 'O'Malley', 12);
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1020, 'JoJo', 'McDodd', 15);
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1030, 'Hooly', 'McDodd', 18);
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1040, 'Hedy', 'McDodd', 18);
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1050, 'Helen', 'McDodd', 18);
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1060, 'Heather', 'McDodd', 18);
insert into NedMcDodd(ID, Firstname, Lastname, Type) values (1070, 'Hildy', 'McDodd', 18);
```



Oracle 7

declare

```
cursor c_names is
  select n.firstname from NedMcDodd n where n.type = 18;
type names_pt is table of varchar2(30) index by binary_integer;
l_names names_pt;
begin
  -- fill up the collection by selecting from the table
  open c_names;
  loop
    fetch c_names into l_names(l_names.count + 1);
    exit when c_names%notfound;
  end loop;
  close c_names;
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
end;
```



Oracle 7

declare

```
cursor c_names is
  select n.firstname from NedMcDodd n where n.type = 18;
type names_pt is table of varchar2(30) index by binary_integer;
l_names names_pt;
```

begin

```
-- fill up the collection by selecting from the table
```

```
open c_names;
loop
  fetch c_names into l_names(l_names.count + 1);
  exit when c_names%notfound;
end loop;
close c_names;
```

```
-- display the contents of the collection
for l_indx in l_names.first .. l_names.last loop
  dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
end loop;
```

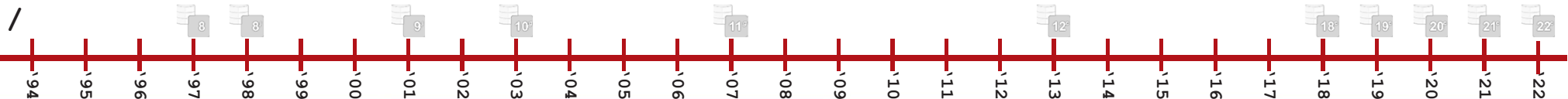
end;

/

'92 '93 '94 '95 '96 '97 '98 '99 '00 '01 '02 '03 '04 '05 '06 '07 '08 '09 '10 '11 '12 '13 '14 '15 '16 '17 '18 '19 '20 '21 '22

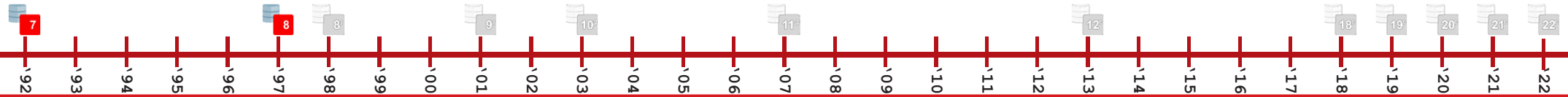
Oracle 7

```
declare
  cursor c_names is
    select n.firstname from NedMcDodd n where n.type = 18;
  type names_pt is table of varchar2(30) index by binary_integer;
  l_names names_pt;
begin
  -- fill up the collection by selecting from the table
  open c_names;
  loop
    fetch c_names into l_names(l_names.count + 1);
    exit when c_names%notfound;
  end loop;
  close c_names;
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
end;
```





Oracle 8

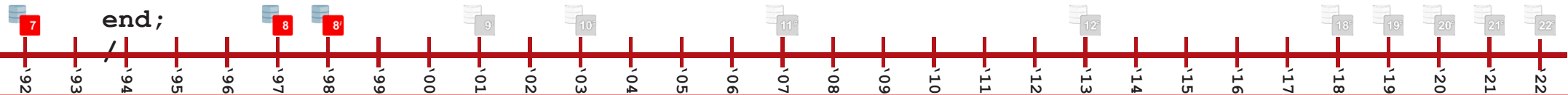


Oracle 8i

declare

```
type names_nt is table of varchar2(30);
l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

end;



Oracle 8i

declare

```
type names_nt is table of varchar2(30);
```

```
l_names names_nt;
```

begin

```
-- initialize the collection
```

```
l_names := names_nt();
```

```
-- extend the collection
```

```
l_names.extend(5); -- make room for 5 items right away
```

```
-- fill up the collection
```

```
l_names(1) := 'Haley';
```

```
l_names(5) := 'Heddy';
```

```
l_names(4) := 'Hilder';
```

```
l_names(2) := 'Hooly';
```

```
l_names(3) := 'Holy';
```

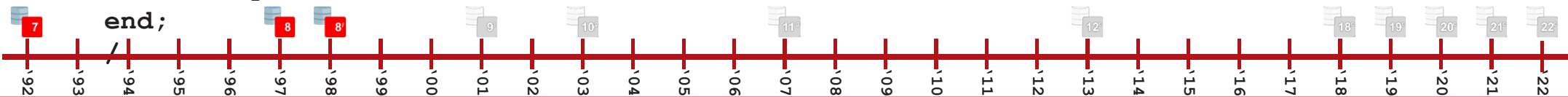
```
-- display the contents of the collection
```

```
for l_indx in l_names.first .. l_names.last loop
```

```
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
```

```
end loop;
```

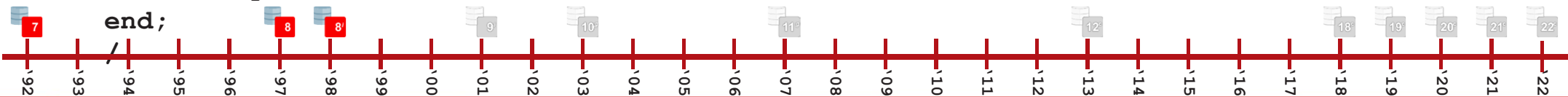
```
end;
```



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

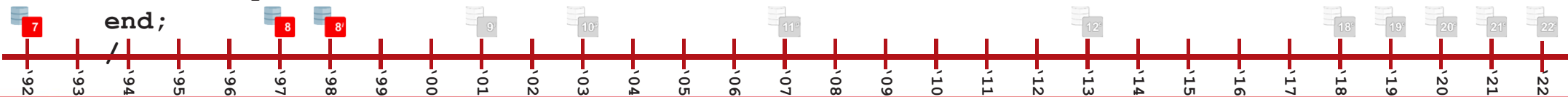
1	
2	
3	
4	
5	



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

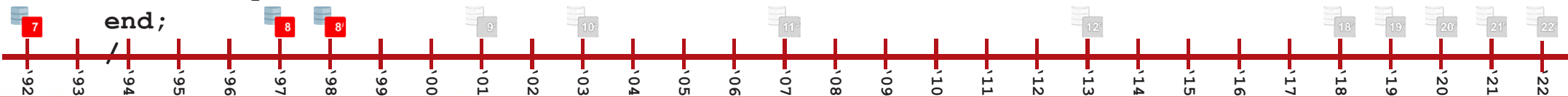
1	Haley
2	
3	
4	
5	



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

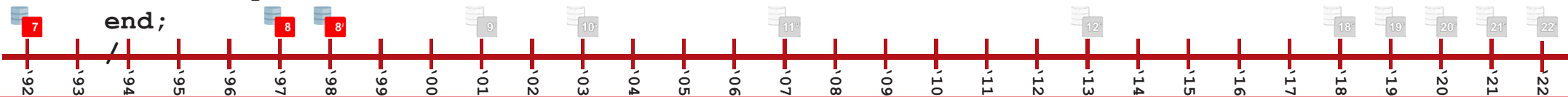
1	Haley
2	
3	
4	
5	Heddy



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

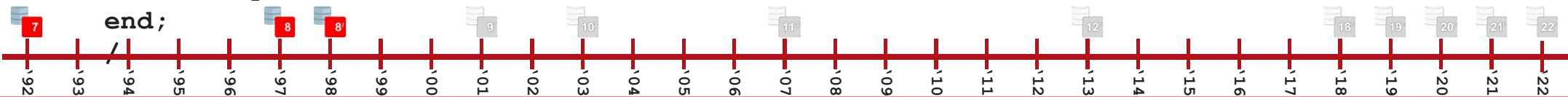
1	Haley
2	
3	
4	Hilder
5	Heddy



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

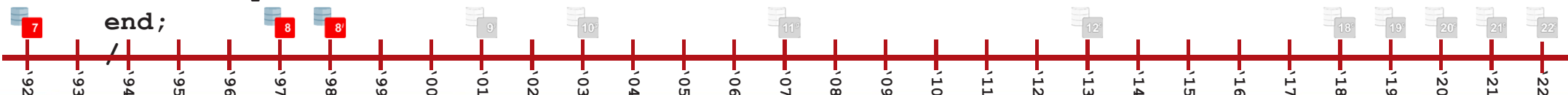
1	Haley
2	Hooly
3	
4	Hilder
5	Heddy



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

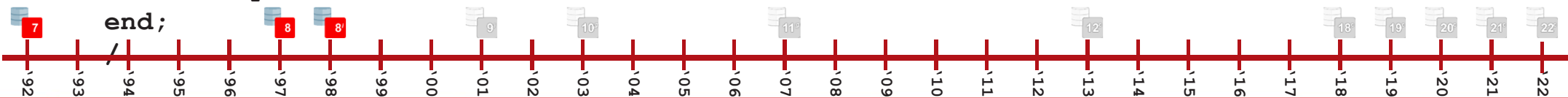
1	Haley
2	Hooly
3	Holy
4	Hilder
5	Heddy



Oracle 8i

```
declare
  type names_nt is table of varchar2(30);
  l_names names_nt;
begin
  -- initialize the collection
  l_names := names_nt();
  -- extend the collection
  l_names.extend(5); -- make room for 5 items right away
  -- fill up the collection
  l_names(1) := 'Haley';
  l_names(5) := 'Heddy';
  l_names(4) := 'Hilder';
  l_names(2) := 'Hooly';
  l_names(3) := 'Holy';
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
```

1	Haley
2	Hooly
3	Holy
4	Hilder
5	Heddy



```

declare
  cursor c_names is
    select n.firstname from NedMcDodd n where n.type = 18;
  type names_ibi is table of varchar2(30) index by binary_integer;
  l_names names_ibi;
begin
  -- fill up the collection by selecting from the table
  open c_names;
  fetch c_names bulk collect into l_names;
  close c_names;
  -- display the contents of the collection
  for l_indx in l_names.first .. l_names.last loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
end;
/

```



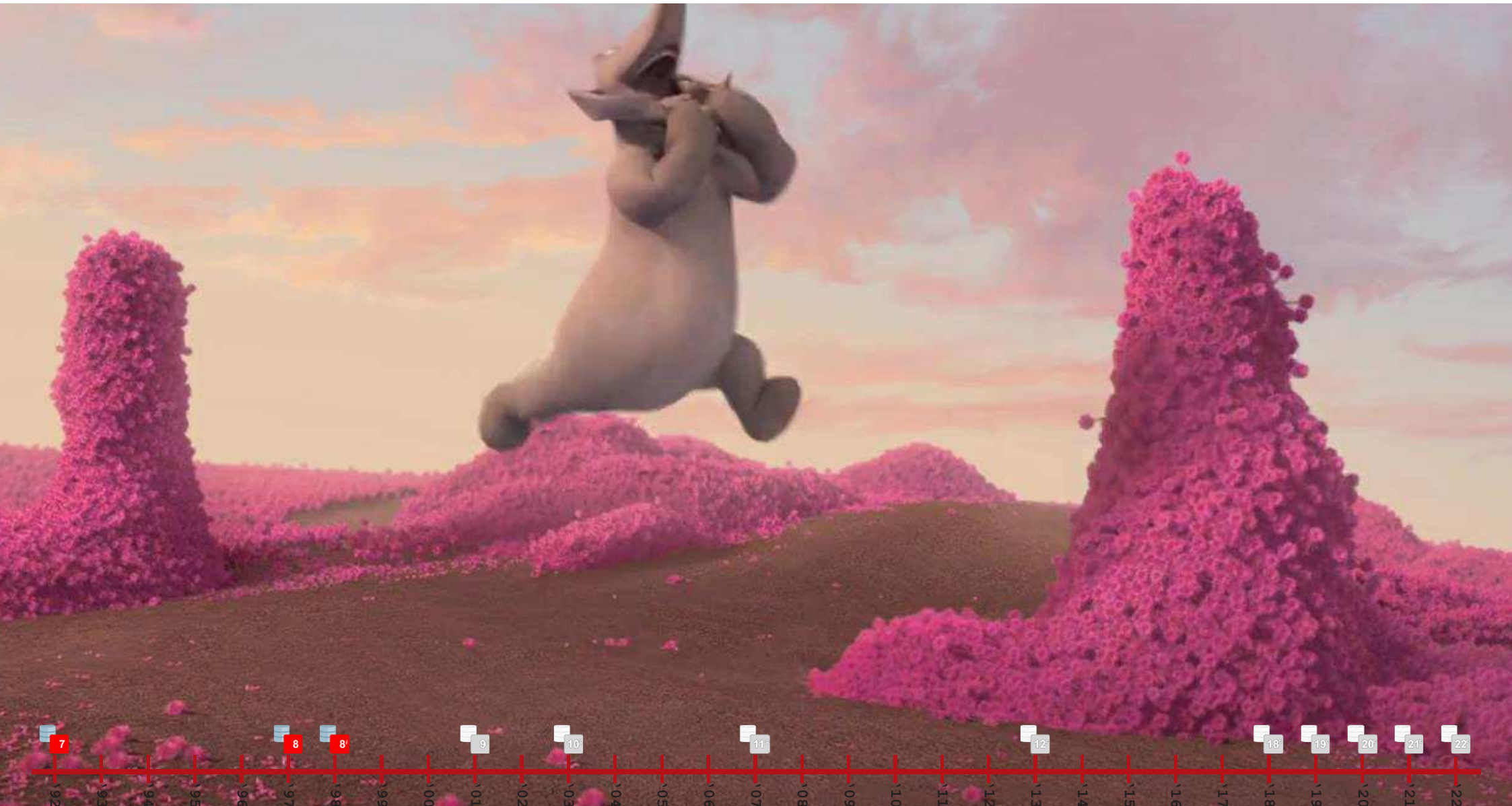

```

declare
  cursor c_names is
    select n.firstname from NedMcDodd n where n.type = 18;
  type names_ibi is table of varchar2(30) index by binary_integer;
  l_names names_ibi;
begin
  -- fill up the collection by selecting from the table
  open c_names;
  fetch c_names bulk collect into l_names;
  close c_names;
  -- display the contents of the collection
  for l_indx in 1 .. l_names.count loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
  end loop;
end;
/

```

Densely filled,
starting at 1





declare

```
c_limit constant number := 4;  
cursor c_names is select n.firstname from NedMcDodd n where n.type = 18;  
type names_ibi is table of varchar2(30) index by binary_integer;  
l_names names_ibi;
```

begin

```
open c_names;
```

loop

```
fetch c_names bulk collect into l_names limit c_limit;
```

```
if l_names.count > 0 then -- display the contents of the collection
```

```
for l_indx in l_names.first .. l_names.last loop
```

```
dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
```

```
end loop;
```

```
end if;
```

```
exit when l_names.count < c_limit;
```

```
end loop;
```

```
close c_names;
```

```
end;
```



Oracle 8i

declare

```
type id_ibi          is table of NedMcDodd.id%type          index by binary_integer;  
type firstname_ibi  is table of NedMcDodd.firstname%type  index by binary_integer;  
type lastname_ibi   is table of NedMcDodd.lastname%type   index by binary_integer;  
type type_ibi       is table of NedMcDodd.type%type        index by binary_integer;
```

```
cursor c_NedMcDodd is  
select n.id, n.firstname, n.lastname, n.type  
   from NedMcDodd n where n.type in (12, 15, 18);
```

```
l_ids          id_ibi;  
l_firstnames  firstname_ibi;  
l_lastnames   lastname_ibi;  
l_types       type_ibi;
```

begin

```
open c_NedMcDodd;  
fetch c_NedMcDodd bulk collect into l_ids  
                                     , l_firstnames
```



Oracle 8i

declare

```
type id_ibi          is table of NedMcDodd.id%type          index by binary_integer;  
type firstname_ibi  is table of NedMcDodd.firstname%type  index by binary_integer;  
type lastname_ibi   is table of NedMcDodd.lastname%type   index by binary_integer;  
type type_ibi       is table of NedMcDodd.type%type        index by binary_integer;
```

```
cursor c_NedMcDodd is  
select n.id, n.firstname, n.lastname, n.type  
       from NedMcDodd n where n.type in (12, 15, 18);
```

```
l_ids          id_ibi;  
l_firstnames  firstname_ibi;  
l_lastnames   lastname_ibi;  
l_types       type_ibi;
```

begin

```
open c_NedMcDodd;  
fetch c_NedMcDodd bulk collect into l_ids  
                                     , l_firstnames
```



Oracle 8i

```
declare
```

```
type id_ibi          is table of NedMcDodd.id%type          index by binary_integer;  
type firstname_ibi  is table of NedMcDodd.firstname%type  index by binary_integer;  
type lastname_ibi   is table of NedMcDodd.lastname%type   index by binary_integer;  
type type_ibi       is table of NedMcDodd.type%type        index by binary_integer;
```

```
cursor c_NedMcDodd is  
select n.id, n.firstname, n.lastname, n.type  
from NedMcDodd n where n.type in (12, 15, 18);
```

```
l_ids          id_ibi;  
l_firstnames  firstname_ibi;  
l_lastnames   lastname_ibi;  
l_types       type_ibi;  
begin  
open c_NedMcDodd;  
fetch c_NedMcDodd bulk collect into l_ids  
                                     , l_firstnames
```



Oracle 8i

```
declare
```

```
type id_ibi          is table of NedMcDodd.id%type          index by binary_integer;  
type firstname_ibi  is table of NedMcDodd.firstname%type  index by binary_integer;  
type lastname_ibi   is table of NedMcDodd.lastname%type   index by binary_integer;  
type type_ibi       is table of NedMcDodd.type%type       index by binary_integer;
```

```
cursor c_NedMcDodd is  
select n.id, n.firstname, n.lastname, n.type  
from NedMcDodd n where n.type in (12, 15, 18);
```

```
l_ids          id_ibi;  
l_firstnames  firstname_ibi;  
l_lastnames   lastname_ibi;  
l_types       type_ibi;
```

```
begin
```

```
open c_NedMcDodd;  
fetch c_NedMcDodd bulk collect into l_ids  
                                     , l_firstnames
```



Oracle 8i

```
l_ids      id_ibi;
l_firstnames  firstname_ibi;
l_lastnames  lastname_ibi;
l_types     type_ibi;
begin
  open c_NedMcDodd;
  fetch c_NedMcDodd bulk collect into l_ids
                                     , l_firstnames
                                     , l_lastnames
                                     , l_types;

  close c_NedMcDodd;
  dbms_output.put_line('Number of relatives :'||l_ids.count);
  for indx in l_ids.first .. l_ids.last loop
    dbms_output.put_line('Relative('||l_ids(indx)||') : '||l_firstnames(indx));
  end loop;
end;
/
```







Oracle 8i

```
create or replace type Names_NT as table of varchar2(32)
/
```

```
create or replace package McDodd as
  function Daughters return Names_nt;
  function Relatives(type_in in number) return Names_nt;
end;
/
```

```
create or replace package body McDodd as
  function Daughters return Names_nt
  is
    l_Daughters Names_NT;
  begin
    l_Daughters := Names_NT();
    l_Daughters.extend(2);
    l_Daughters(1) := 'Hooly';
    l_Daughters(2) := 'Holy';
```



Oracle 8i

```
create or replace type Names_NT as table of varchar2(32)
/
```

```
create or replace package McDodd as
  function Daughters return Names_nt;
  function Relatives(type_in in number) return Names_nt;
end;
/
```

```
create or replace package body McDodd as
  function Daughters return Names_nt
  is
    l_Daughters Names_NT;
  begin
    l_Daughters := Names_NT();
    l_Daughters.extend(2);
    l_Daughters(1) := 'Hooly';
    l_Daughters(2) := 'Holy';
```



Oracle 8i

```
create or replace type Names_NT as table of varchar2(32)
```

```
/
```

```
create or replace package McDodd as
```

```
    function Daughters return Names_nt;
```

```
    function Relatives(type_in in number) return Names_nt;
```

```
end;
```

```
/
```

```
create or replace package body McDodd as
```

```
    function Daughters return Names_nt
```

```
    is
```

```
        l_Daughters Names_NT;
```

```
begin
```

```
    l_Daughters := Names_NT();
```

```
    l_Daughters.extend(2);
```

```
    l_Daughters(1) := 'Hooly';
```

```
    l_Daughters(2) := 'Holy';
```



Oracle 8i

```
create or replace package body McDodd as
function Daughters return Names_nt
is
  l_Daughters Names_NT;
begin
  l_Daughters := Names_NT();
  l_Daughters.extend(2);
  l_Daughters(1) := 'Hooly';
  l_Daughters(2) := 'Holy';
  return l_Daughters;
end;

function Relatives(type_in in number) return Names_nt
is
  l_Names Names_NT;
begin
  select n.firstname
  bulk collect into l_Names
  from NedMcDodd n
```



Oracle 8i

```
function Relatives(type_in in number) return Names_nt
is
  l_Names Names_NT;
begin
  select n.firstname
  bulk collect into l_Names
  from NedMcDodd n
  where n.type = type_in;
  return l_Names;
end;
```

```
end;
/
select column_value from table(cast (McDodd.Daughters() as Names_nt))
/
select column_value from table(cast (McDodd.Relatives(20) as Names_nt))
/
```



Oracle 9i

declare

```
type names_aa is table of varchar2(30) index by varchar2(10);
```

```
l_names names_aa;  
l_indx varchar2(10);  
begin  
  -- fill up the collection  
  l_names('Hooly') := 'One';  
  l_names('Heddy') := 'Two';  
  l_names('Hilder') := 'Three';  
  l_names('Holy') := 'Four';  
  l_names('Haley') := 'Five';  
  -- display the contents of the collection  
  l_indx := l_names.first;  
  while l_indx is not null loop  
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));  
    l_indx := l_names.next(l_indx);  
  end loop;
```



Oracle 9i

declare

```
type names_aa is table of varchar2(30) index by varchar2(10);
```

```
l_names names_aa;  
l_indx varchar2(10);  
begin  
  -- fill up the collection  
  l_names('Hooly') := 'One';  
  l_names('Heddy') := 'Two';  
  l_names('Hilder') := 'Three';  
  l_names('Holy') := 'Four';  
  l_names('Haley') := 'Five';  
  -- display the contents of the collection  
  l_indx := l_names.first;  
  while l_indx is not null loop  
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));  
    l_indx := l_names.next(l_indx);  
  end loop;
```

Hooly	One
-------	-----



Oracle 9i

```
declare
  type names_aa is table of varchar2(30) index by varchar2(10);

  l_names names_aa;
  l_indx  varchar2(10);
begin
  -- fill up the collection
  l_names('Hooly') := 'One';
  l_names('Heddy') := 'Two';
  l_names('Hilder') := 'Three';
  l_names('Holy') := 'Four';
  l_names('Haley') := 'Five';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
```

Heddy	Two	Hooly	One
-------	-----	-------	-----



Oracle 9i

```
declare
  type names_aa is table of varchar2(30) index by varchar2(10);

  l_names names_aa;
  l_indx  varchar2(10);
begin
  -- fill up the collection
  l_names('Hooly') := 'One';
  l_names('Heddy') := 'Two';
  l_names('Hilder') := 'Three';
  l_names('Holy') := 'Four';
  l_names('Haley') := 'Five';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
```

Hilder	Three
--------	-------

Heddy	Two
Hooly	One



Oracle 9i

```
declare
  type names_aa is table of varchar2(30) index by varchar2(10);

  l_names names_aa;
  l_indx  varchar2(10);
begin
  -- fill up the collection
  l_names('Hooly') := 'One';
  l_names('Heddy') := 'Two';
  l_names('Hilder') := 'Three';
  l_names('Holy') := 'Four';
  l_names('Haley') := 'Five';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
```

Heddy	Two
Hilder	Three
Hooly	One

Holy	Four
------	------



Oracle 9i

```
declare
  type names_aa is table of varchar2(30) index by varchar2(10);

  l_names names_aa;
  l_indx  varchar2(10);
begin
  -- fill up the collection
  l_names('Hooly') := 'One';
  l_names('Heddy') := 'Two';
  l_names('Hilder') := 'Three';
  l_names('Holy') := 'Four';
  l_names('Haley') := 'Five';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
```

Haley	Five	Heddy	Two
		Hilder	Three
		Holy	Four
		Hooly	One



Oracle 9i

```
declare
  type names_aa is table of varchar2(30) index by varchar2(10);

  l_names names_aa;
  l_indx  varchar2(10);
begin
  -- fill up the collection
  l_names('Hooly') := 'One';
  l_names('Heddy') := 'Two';
  l_names('Hilder') := 'Three';
  l_names('Holy') := 'Four';
  l_names('Haley') := 'Five';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
```

Haley	Five
Heddy	Two
Hilder	Three
Holy	Four
Hooly	One



Oracle 9i

```
l_names names_aa;  
l_indx varchar2(10);  
begin  
  -- fill up the collection  
  l_names('Hooly') := 'One';  
  l_names('Heddy') := 'Two';  
  l_names('Hilder') := 'Three';  
  l_names('Holy') := 'Four';  
  l_names('Haley') := 'Five';  
  -- display the contents of the collection  
  l_indx := l_names.first;  
  while l_indx is not null loop  
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));  
    l_indx := l_names.next(l_indx);  
  end loop;  
end;  
/
```

Haley	Five
Heddy	Two
Hilder	Three
Holy	Four
Hooly	One

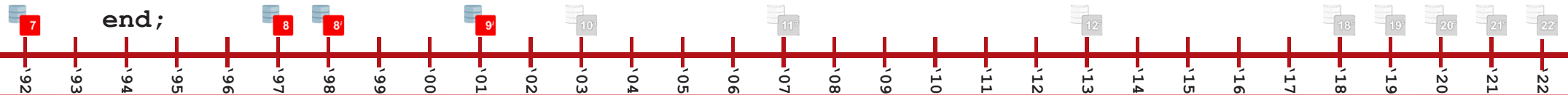


Oracle 9i

declare

```
type NedMcDodd_aa is table of NedMcDodd%rowtype index by binary_integer;
```

```
cursor c_NedMcDodd
is
select n.id, n.firstname, n.lastname, n.type
  from NedMcDodd n
  where n.type in (12, 15, 18);
l_NedMcDodds NedMcDodd_aa;
begin
  open c_NedMcDodd;
  fetch c_NedMcDodd bulk collect into l_NedMcDodds;
  close c_NedMcDodd;
  dbms_output.put_line('Number of relatives : '||l_NedMcDodds.count);
  for indx in l_NedMcDodds.first .. l_NedMcDodds.last loop
    dbms_output.put_line('Relative('||l_NedMcDodds(indx).id||') : ' ||
                        l_NedMcDodds(indx).firstname);
  end loop;
end;
```



Oracle 9i

declare

```
type NedMcDodd_aa is table of NedMcDodd%rowtype index by binary_integer;
```

```
cursor c_NedMcDodd  
is
```

```
select n.id, n.firstname, n.lastname, n.type  
from NedMcDodd n  
where n.type in (12, 15, 18);
```

```
l_NedMcDodds NedMcDodd_aa;
```

begin

```
open c_NedMcDodd;
```

```
fetch c_NedMcDodd bulk collect into l_NedMcDodds;
```

```
close c_NedMcDodd;
```

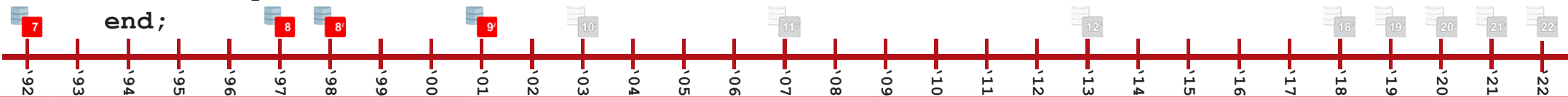
```
dbms_output.put_line('Number of relatives : '||l_NedMcDodds.count);
```

```
for indx in l_NedMcDodds.first .. l_NedMcDodds.last loop
```

```
dbms_output.put_line('Relative('||l_NedMcDodds(indx).id||') : ' ||  
l_NedMcDodds(indx).firstname);
```

```
end loop;
```

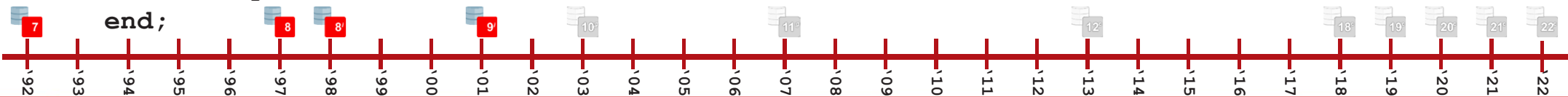
```
end;
```



Oracle 9i

```
declare
  type NedMcDodd_aa is table of NedMcDodd%rowtype index by binary_integer;

  cursor c_NedMcDodd
  is
  select n.id, n.firstname, n.lastname, n.type
     from NedMcDodd n
    where n.type in (12, 15, 18);
  l_NedMcDodds NedMcDodd_aa;
begin
  open c_NedMcDodd;
  fetch c_NedMcDodd bulk collect into l_NedMcDodds;
  close c_NedMcDodd;
  dbms_output.put_line('Number of relatives : '||l_NedMcDodds.count);
  for indx in l_NedMcDodds.first .. l_NedMcDodds.last loop
    dbms_output.put_line('Relative('||l_NedMcDodds(indx).id||') : ' ||
                          l_NedMcDodds(indx).firstname);
  end loop;
end;
```



Oracle 9i

```
create or replace type Names_NT as table of varchar2(32)
```

```
/
```

```
create or replace package McDodd as
```

```
    function Daughters return Names_NT pipelined;
```

```
    function Relatives(type_in in number) return Names_NT pipelined;
```

```
end;
```

```
/
```

```
create or replace package body McDodd as
```

```
    function Daughters return Names_NT pipelined
```

```
    is
```

```
        l_row NedMcDodd.Firstname%type;
```

```
    begin
```

```
        l_row := 'Hooly';
```

```
        pipe row(l_row);
```

```
        l_row := 'Holy';
```

```
        pipe row(l_row);
```



Oracle 9i

```
create or replace type Names_NT as table of varchar2(32)
/
```

```
create or replace package McDodd as
  function Daughters return Names_NT pipelined;
  function Relatives(type_in in number) return Names_NT pipelined;
end;
/
```

```
create or replace package body McDodd as
  function Daughters return Names_NT pipelined
  is
    l_row NedMcDodd.Firstname%type;
  begin
    l_row := 'Hooly';
    pipe row(l_row);
    l_row := 'Holy';
    pipe row(l_row);
    ...
  end;
```



Oracle 9i

```
create or replace type Names_NT as table of varchar2(32)
```

```
/
```

```
create or replace package McDodd as
```

```
    function Daughters return Names_NT pipelined;
```

```
    function Relatives(type_in in number) return Names_NT pipelined;
```

```
end;
```

```
/
```

```
create or replace package body McDodd as
```

```
    function Daughters return Names_NT pipelined
```

```
    is
```

```
        l_row NedMcDodd.Firstname%type;
```

```
    begin
```

```
        l_row := 'Hooly';
```

```
        pipe row(l_row);
```

```
        l_row := 'Holy';
```

```
        pipe row(l_row);
```



Oracle 9i

```
create or replace package body McDodd as
function Daughters return Names_NT pipelined
is
  l_row NedMcDodd.Firstname%type;
begin
  l_row := 'Hooly';
  pipe row(l_row);
  l_row := 'Holy';
  pipe row(l_row);
  return;
end;
```

```
function Relatives(type_in in number) return Names_NT pipelined
```

```
is
```

```
begin
```

```
  for l_row in (select n.firstname from NedMcDodd n where n.type = type_in) loop
```

```
    pipe row(l_row.firstname);
```

```
  end loop;
```

```
  return;
```



Oracle 9i

```
function Relatives(type_in in number) return Names_NT pipelined
is
begin
  for l_row in (select n.firstname from NedMcDodd n where n.type = type_in) loop
    pipe row(l_row.firstname);
  end loop;
  return;
end;
end;
/
```

```
select * from table(McDodd.Daughters)
/
select * from table(McDodd.Relatives(20))
/
```



Oracle 9i

```
clear screen
set serveroutput on size unlimited

drop table FatMcDodd
/

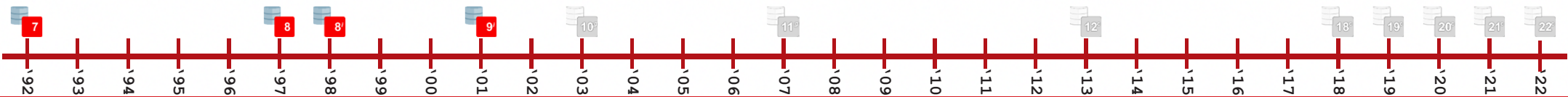
prompt create a big fat table
create table FatMcDodd as select * from NedMcDodd where 1=2
/
begin
  for indx in 1 .. 10000 loop
    insert into FatMcDodd select * from NedMcDodd;
  end loop;
  commit;
end;
/
select count(*) from FatMcDodd
/
```



Oracle 9i

```
end loop;
commit;
end;
/
select count(*) from FatMcDodd
/

set timing on
declare
  l_id FatMcDodd.id%type;
begin
  for rec in (select * from FatMcDodd) loop
    l_id := rec.id;
  end loop;
end;
/
set timing off
```



PL/SQL procedure successfully completed
Executed in 4.453 seconds

Oracle 9i

```
declare
  l_id FatMcDodd.id%type;
begin
  for rec in (select * from FatMcDodd) loop
    l_id := rec.id;
  end loop;
end;
/
set timing off
```



PL/SQL procedure successfully completed
Executed in 1.235 seconds

Oracle 9i

```
declare
  cursor c_FatMcDodd
  is
  select * from FatMcDodd;
  type t_FatMcDodd is table of FatMcDodd%rowtype index by pls_integer;
  lFatMcDodd t_FatMcDodd;
  l_id FatMcDodd.id%type;
begin
  open c_FatMcDodd;
  loop
    fetch c_FatMcDodd bulk collect into lFatMcDodd limit 10000;
    exit when lFatMcDodd.count = 0;
    for indx in lFatMcDodd.first .. lFatMcDodd.last loop
      l_id := lFatMcDodd(indx).id;
    end loop;
  end loop;
end;
```



Oracle 10g

```
clear screen
```

```
set serveroutput on size unlimited
```

```
drop table FatMcDodd
```

```
/
```

```
prompt create a big fat table
```

```
create table FatMcDodd as select * from NedMcDodd where 1=2
```

```
/
```

```
begin
```

```
  for indx in 1 .. 10000 loop
```

```
    insert into FatMcDodd select * from NedMcDodd;
```

```
  end loop;
```

```
  commit;
```

```
end;
```

```
/
```

```
select count(*) from FatMcDodd
```

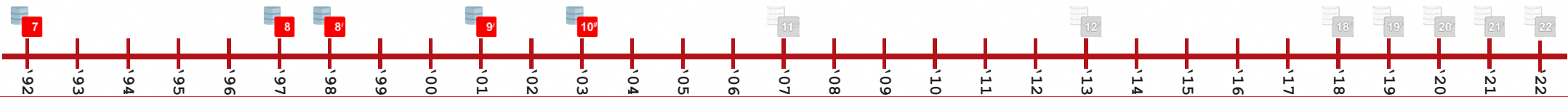
```
/
```



Oracle 10g

```
end loop;
commit;
end;
/
select count(*) from FatMcDodd
/

set timing on
declare
  l_id FatMcDodd.id%type;
begin
  for rec in (select * from FatMcDodd) loop
    l_id := rec.id;
  end loop;
end;
/
set timing off
```



PL/SQL procedure successfully completed
Executed in 0.701 seconds

Oracle 10g

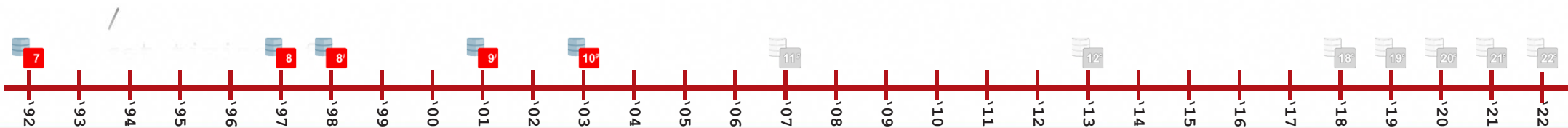
```
declare
  l_id FatMcDodd.id%type;
begin
  for rec in (select * from FatMcDodd) loop
    l_id := rec.id;
  end loop;
end;
/
set timing off
```

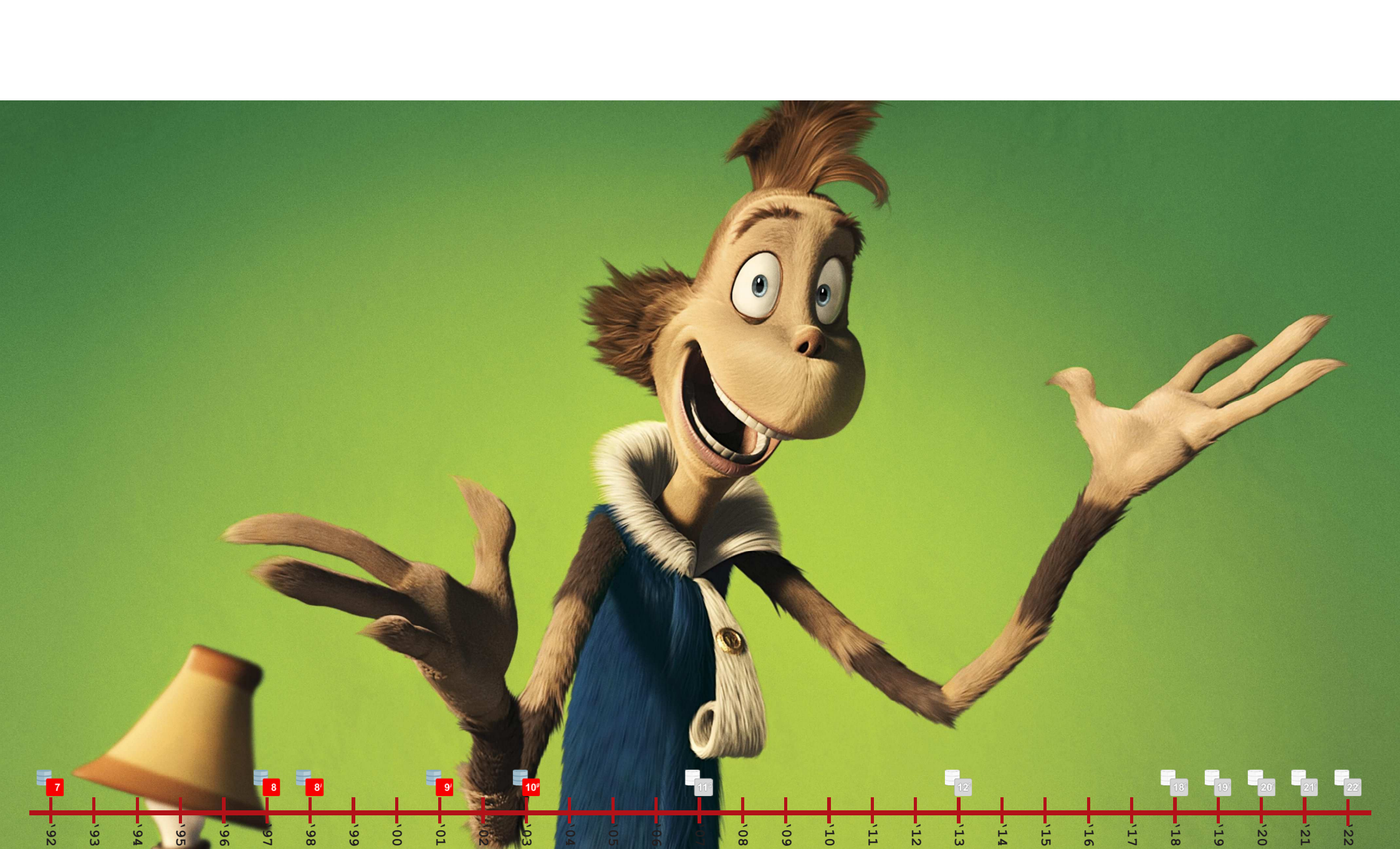


PL/SQL procedure successfully completed
Executed in 0.581 seconds

Oracle 10g

```
declare
  cursor c_FatMcDodd
  is
  select * from FatMcDodd;
  type t_FatMcDodd is table of FatMcDodd%rowtype index by pls_integer;
  lFatMcDodd t_FatMcDodd;
  l_id FatMcDodd.id%type;
begin
  open c_FatMcDodd;
  loop
    fetch c_FatMcDodd bulk collect into lFatMcDodd limit 10000;
    exit when lFatMcDodd.count = 0;
    for indx in lFatMcDodd.first .. lFatMcDodd.last loop
      l_id := lFatMcDodd(indx).id;
    end loop;
  end loop;
end;
```





Oracle 10g

create or replace package McDodd as

```
type Names_NT is table of NedMcDodd.firstname%type;  
function Daughters return McDodd.Names_NT pipelined;  
function Relatives(type_in in number) return McDodd.Names_NT pipelined;  
end;  
/
```

create or replace package body McDodd as

```
function Daughters return McDodd.Names_NT pipelined  
is  
l_row NedMcDodd.firstname%type;  
begin  
l_row := 'Hooly';  
pipe row(l_row);  
l_row := 'Holy';  
pipe row(l_row);  
return;  
end;
```



Oracle 10g

```
create or replace package McDodd as
  type Names_NT is table of NedMcDodd.firstname%type;
  function Daughters return McDodd.Names_NT pipelined;
  function Relatives(type_in in number) return McDodd.Names_NT pipelined;
end;
/
```

```
create or replace package body McDodd as
  function Daughters return McDodd.Names_NT pipelined
  is
    l_row NedMcDodd.firstname%type;
  begin
    l_row := 'Hooly';
    pipe row(l_row);
    l_row := 'Holy';
    pipe row(l_row);
    return;
  end;
```



Oracle 10g

```
create or replace package McDodd as
  type Names_NT is table of NedMcDodd.firstname%type;
  function Daughters return McDodd.Names_NT pipelined;
  function Relatives(type_in in number) return McDodd.Names_NT pipelined;
end;
/
```

```
create or replace package body McDodd as
  function Daughters return McDodd.Names_NT pipelined
  is
    l_row NedMcDodd.firstname%type;
  begin
    l_row := 'Hooly';
    pipe row(l_row);
    l_row := 'Holy';
    pipe row(l_row);
    return;
  end;
```



Oracle 10g

```
create or replace package body McDodd as
function Daughters return McDodd.Names_NT pipelined
is
  l_row NedMcDodd.firstname%type;
begin
  l_row := 'Hooly';
  pipe row(l_row);
  l_row := 'Holy';
  pipe row(l_row);
  return;
end;
function Relatives(type_in in number) return McDodd.Names_NT pipelined
is
begin
  for l_row in (select n.firstname from NedMcDodd n where n.type = type_in) loop
    pipe row(l_row.firstname);
  end loop;
  return;
end;
```



Oracle 10g

```
function Relatives(type_in in number) return McDodd.Names_NT pipelined
is
begin
  for l_row in (select n.firstname from NedMcDodd n where n.type = type_in) loop
    pipe row(l_row.firstname);
  end loop;
  return;
end;
end;
/
```

```
select * from table(McDodd.Daughters)
/
select * from table(McDodd.Relatives(20))
/
```

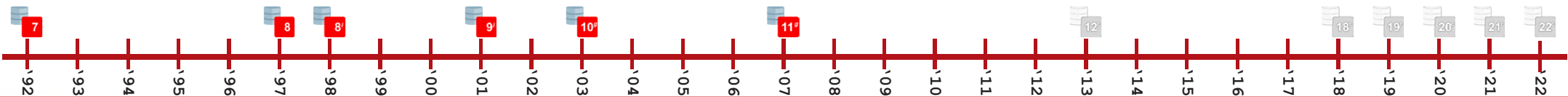




Oracle 11g

```
create or replace package McDodd as
  type Daughters_NT is table of NedMcDodd.firstname%type;
end;
/

declare
  Daughters McDodd.Daughters_NT;
begin
  Daughters := McDodd.Daughters_NT();
  Daughters.extend(2);
  Daughters(1) := 'Hooly';
  Daughters(2) := 'Holy';
  for i in (select n.firstname from NedMcDodd n where n.type = 18
           and n.firstname not in (select * from table(Daughters))) loop
    dbms_output.put_line(i.firstname);
  end loop;
end;
/
```



Oracle 11g

create or replace

```
type Daughters_NT as table of varchar2(32)
/
```

declare

```
Daughters Daughters_NT;
```

begin

```
Daughters := Daughters_NT();
```

```
Daughters.extend(2);
```

```
Daughters(1) := 'Hooly';
```

```
Daughters(2) := 'Holy';
```

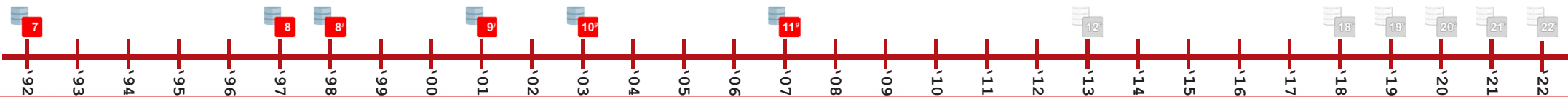
```
for i in (select n.firstname from NedMcDodd n where n.type = 18
         and n.firstname not in (select * from table(Daughters))) loop
```

```
dbms_output.put_line(i.firstname);
```

```
end loop;
```

```
end;
```

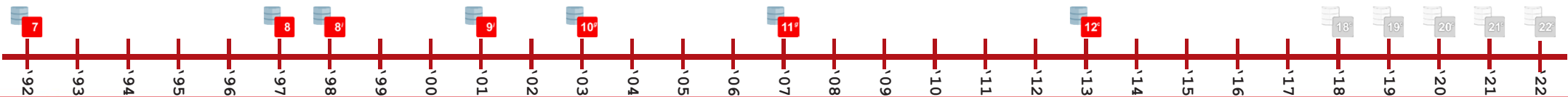
```
/
```



Oracle 12c

```
create or replace package McDodd as
  type Daughters_NT is table of NedMcDodd.firstname%type;
end;
/

declare
  Daughters McDodd.Daughters_NT;
begin
  Daughters := McDodd.Daughters_NT();
  Daughters.extend(2);
  Daughters(1) := 'Hooly';
  Daughters(2) := 'Holy';
  for i in (select n.firstname from NedMcDodd n where n.type = 18
           and n.firstname not in (select * from table(Daughters))) loop
    dbms_output.put_line(i.firstname);
  end loop;
end;
/
```



Oracle 12c

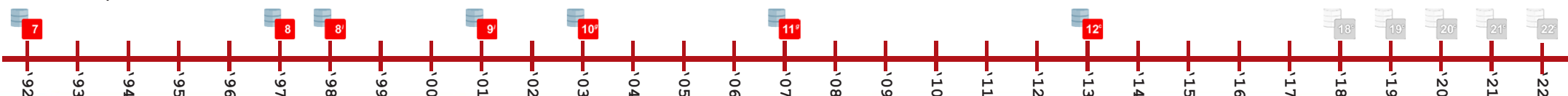
```
create or replace package McDodd as
  type Daughters_AA is table of NedMcDodd.firstname%type index by pls_integer;
end;
/
```

declare

```
  Daughters McDodd.Daughters_AA;
```

begin

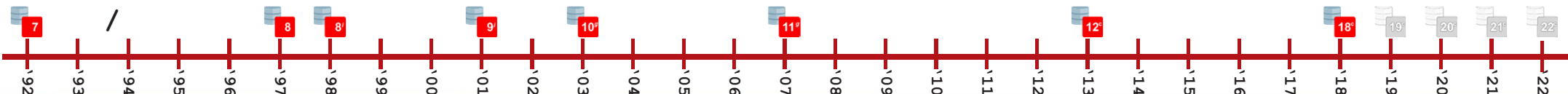
```
  Daughters(1) := 'Hooly';
  Daughters(2) := 'Holy';
  for i in (select n.firstname from NedMcDodd n where n.type = 18
            and n.firstname not in (select * from table(Daughters))) loop
    dbms_output.put_line(i.firstname);
  end loop;
end;
/
```





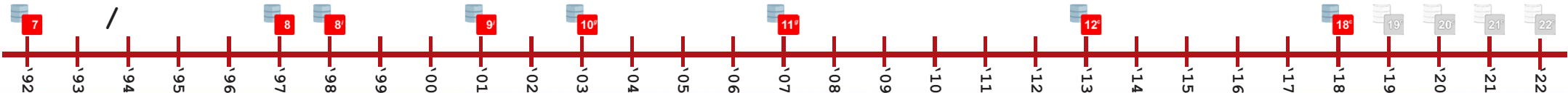
Oracle 18c

```
declare
  type names_aa is table of varchar2(30) index by binary_integer;
  l_names names_aa;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```



Oracle 18c

```
declare
  type names_aa is table of varchar2(30) index by binary_integer;
  l_names names_aa;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names := names_aa(1 => 'Hooly'
    ,42 => 'Heddy'
    ,10 => 'Hilder'
    ,(l_names.count + 1) => 'Holy'
    ,-1 => 'Haley');
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```



Oracle 18c

```
declare
  type McDodd_rec is record
    ( firstname varchar2(32)
    , type      number(2)
    );
  type McDodd_AA is table of McDodd_rec index by pls_integer;
  l_McDodd McDodd_AA;
  l_indx   pls_integer;
begin
  -- fill up the collection
  l_McDodd := McDodd_AA(1 => McDodd_rec(firstname => 'Hooly', type => 18)
    ,42 => McDodd_rec(firstname => 'Heddy', type => 18)
    ,10 => McDodd_rec(firstname => 'Hilder', type => 18)
    ,4  => McDodd_rec(firstname => 'Holy', type => 18)
    ,-1 => McDodd_rec(firstname => 'Haley', type => 18));

  -- display the contents of the collection
  l_indx := l_McDodd.first;
  while l_indx is not null loop
```



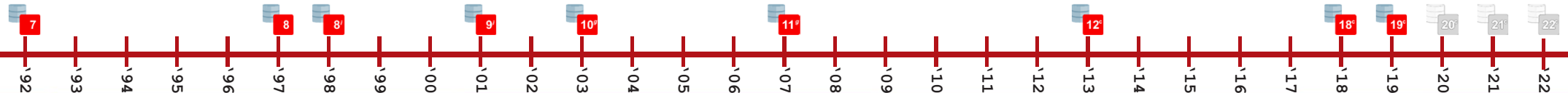
Oracle 18c

```
declare
  type McDodd_rec is record
    ( firstname varchar2(32)
    , type      number(2)
    );
  type McDodd_AA is table of McDodd_rec index by pls_integer;
  l_McDodd McDodd_AA;
  l_indx   pls_integer;
begin
  -- fill up the collection
  l_McDodd := McDodd_AA(1 => McDodd_rec(firstname => 'Hooly', type => 18)
    ,42 => McDodd_rec(firstname => 'Heddy', type => 18)
    ,10 => McDodd_rec(firstname => 'Hilder', type => 18)
    ,4  => McDodd_rec(firstname => 'Holy', type => 18)
    ,-1 => McDodd_rec(firstname => 'Haley', type => 18));

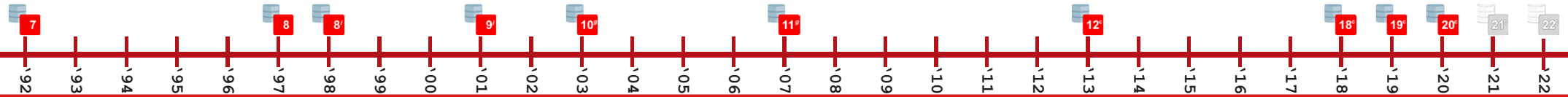
  -- display the contents of the collection
  l_indx := l_McDodd.first;
  while l_indx is not null loop
```



Oracle 19c

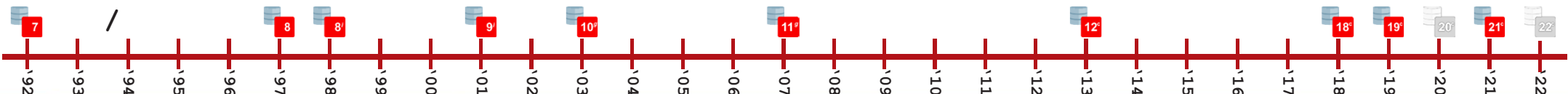


Oracle 20c



Oracle 21c

```
declare
  type names_aa is table of varchar2(30) index by binary_integer;
  l_names names_aa;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection
  l_indx := l_names.first;
  while l_indx is not null loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));
    l_indx := l_names.next(l_indx);
  end loop;
end;
```



Oracle 21c

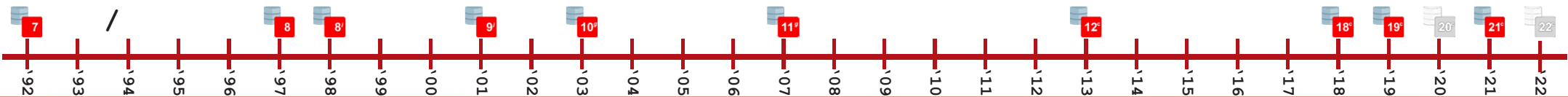
```
declare
  type names_aa is table of varchar2(30) index by binary_integer;
  l_names names_aa;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection

  for l_indx in indices of l_names loop
    dbms_output.put_line(l_indx || ' ' || l_names(l_indx));

  end loop;
end;
```

```
-1) Haley
1) Hooly
4) Holy
10) Hilder
42) Heddy

PL/SQL procedure successfully completed
```



Oracle 21c

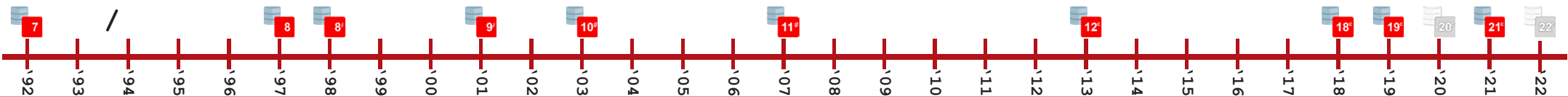
```
declare
  type names_aa is table of varchar2(30) index by binary_integer;
  l_names names_aa;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection

  for l_indx in values of l_names loop
    dbms_output.put_line(l_indx
```

```
Haley
Hooly
Holy
Hilder
Heddy

PL/SQL procedure successfully completed
```

```
end loop;
end;
```



Oracle 21c

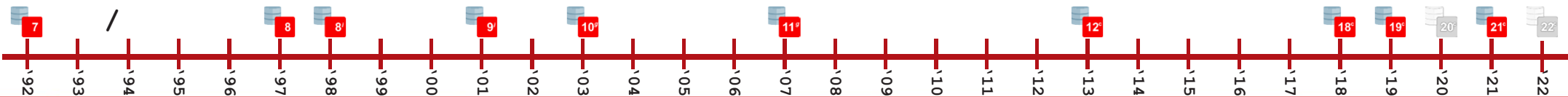
```
declare
  type names_aa is table of varchar2(30) index by binary_integer;
  l_names names_aa;
  l_indx binary_integer;
begin
  -- fill up the collection
  l_names(1) := 'Hooly';
  l_names(42) := 'Heddy';
  l_names(10) := 'Hilder';
  l_names(l_names.count + 1) := 'Holy';
  l_names(-1) := 'Haley';
  -- display the contents of the collection

  for l_indx, l_value in pairs of l_names loop
    dbms_output.put_line(l_indx || ' ' || l_value );

  end loop;
end;
```

```
-1) Haley
1) Hooly
4) Holy
10) Hilder
42) Heddy

PL/SQL procedure successfully completed
```





This presentation was inspired by the movie

Dr. Seuss'
HORTON
HEARS A WHO!™

An animation done by:

Blue Sky™





More info:

Collections in Oracle Part 1

https://patch72.com/Collections_Part1

Collections in Oracle Part 2

https://patch72.com/Collections_Part2

Bulk Processing in Oracle Part 1

https://patch72.com/BulkProcessing_Part1

Bulk Processing in Oracle Part 2

https://patch72.com/BulkProcessing_Part2



Easy Initializing for Records and Arrays

<https://blogs.oracle.com/oraclemagazine/easy-initializing-for-records-and-arrays>

Oracle-Base

<https://oracle-base.com/>

LiveSQL (Search for 'Collections')

<https://livesql.oracle.com/>

Contact details:

Blog:

<https://blog.bar-solutions.com>

Twitter:



@patch72

LinkedIn:



<https://www.linkedin.com/in/patrickbarel/>

Email:



Patrick.Barel@GMail.com



Q&A