Code-

DECLARE

r\_emp employees%rowtype;

BEGIN

SELECT \* INTO r\_emp

FROM employees

WHERE employee\_id = '9';

dbms\_output.put\_line(r\_emp.first\_name || ' ' ||

r\_emp.last\_name || ' earns ' ||

r\_emp.salary);

END;

Output-

Statement processed.

First9 Last9 earns 75693.76

Code-

DECLARE

--r\_emp employees%rowtype;

type t\_emp IS RECORD (first\_name VARCHAR2(50),

last\_name employees.last\_name%TYPE,

salary employees.salary%TYPE);

r\_emp t\_emp;

BEGIN

SELECT first\_name,last\_name,salary

INTO r\_emp

FROM employees

WHERE employee\_id = '1';

dbms\_output.put\_line(r\_emp.first\_name || ' ' ||

r\_emp.last\_name || ' earns ' ||

r\_emp.salary);

END;

Output-

Statement processed.

First1 Last1 earns 97342.07

Code-

CREATE TABLE retired\_employees

AS SELECT \* FROM employees WHERE 1=2;

SELECT \* FROM retired\_employees;

/

DECLARE

r\_emp employees%rowtype;

BEGIN

SELECT \*

INTO r\_emp

FROM employees

WHERE employee\_id = 1;

r\_emp.salary := 0;

r\_emp.commission\_pct := 0;

INSERT INTO retired\_employees VALUES r\_emp;

END;

Output-

Statement processed.

Code-

DECLARE

TYPE e\_list IS VARRAY(5) OF VARCHAR2(50);

employees e\_list;

BEGIN

employees := e\_list('Alex','Bruce','John','Bob','Richard');

FOR i IN 1..5 LOOP

dbms\_output.put\_line(employees(i));

END LOOP;

END;

Output-

Statement processed.

Alex

Bruce

John

Bob

Richard

Code-

DECLARE

TYPE e\_list IS VARRAY(5) OF VARCHAR2(50);

employees e\_list;

BEGIN

employees := e\_list('Alex','Bruce','John','Bob');

for i IN 1..employees.count() LOOP

dbms\_output.put\_line(employees(i));

END LOOP;

END;

Output-

Statement processed.

Alex

Bruce

John

Bob

Code-

DECLARE

TYPE e\_list IS VARRAY(5) OF VARCHAR2(50);

employees e\_list;

BEGIN

employees := e\_list('Alex','Bruce','John','Bob');

FOR i IN employees.first()..employees.last() LOOP

dbms\_output.put\_line(employees(i));

END LOOP;

END;

Output-

Statement processed.

Alex

Bruce

John

Bob

Code-

DECLARE

TYPE e\_list IS VARRAY(5) OF VARCHAR2(50);

employees e\_list;

BEGIN

employees := e\_list('Alex','Bruce','John','Bob');

FOR i IN 1..5 LOOP

IF employees.exists(i) THEN

dbms\_output.put\_line(employees(i));

END IF;

END LOOP;

END;

Output-

Statement processed.

Alex

Bruce

John

Bob

Code-

DECLARE

TYPE e\_list IS VARRAY(5) OF VARCHAR2(50);

employees e\_list('Alex','Bruce','John','Bob');

BEGIN

--employees := e\_list('Alex','Bruce','John','Bob');

FOR i IN 1..5 LOOP

IF employees.exists(i) THEN

dbms\_output.put\_line(employees(i));

END IF;

END LOOP;

END;

Output-

ORA-06550: line 3, column 13:

PLS-00572: improper constraint form used

Code-

CREATE TYPE e\_list IS VARRAY(15) OF VARCHAR2(50);

/

CREATE OR REPLACE TYPE e\_list AS VARRAY(20) OF VARCHAR2(100);

/

DECLARE

employees e\_list := e\_list();

idx NUMBER := 1;

BEGIN

FOR i IN 100..110 LOOP

employees.extend;

SELECT first\_name

INTO employees(idx)

FROM employees

WHERE employee\_id = i;

idx := idx + 1;

END LOOP;

FOR x IN 1..employees.count() LOOP

dbms\_output.put\_line(employees(x));

END LOOP;

END;

/

DROP TYPE E\_LIST;

Output-

Type created.

Type created.

Type dropped.

Code-

DECLARE

TYPE e\_list IS TABLE OF VARCHAR2(50);

emps e\_list;

BEGIN

emps := e\_list('Alex','Bruce','John');

FOR i IN 1..emps.count() LOOP

dbms\_output.put\_line(emps(i));

END LOOP;

END;

Output-

Statement processed.

Alex

Bruce

John

Code-

DECLARE

TYPE e\_list IS TABLE OF VARCHAR2(50);

emps e\_list;

BEGIN

emps := e\_list('Alex','Bruce','John');

emps.extend;

emps(4) := 'Bob';

FOR i IN 1..emps.count() LOOP

dbms\_output.put\_line(emps(i));

END LOOP;

END;

Output-

Statement processed.

Alex

Bruce

John

Bob

Code-

DECLARE

TYPE e\_list IS TABLE OF employees.first\_name%type;

emps e\_list := e\_list();

idx PLS\_INTEGER:= 1;

BEGIN

FOR x IN 1 .. 10 LOOP

emps.extend;

SELECT first\_name INTO emps(idx)

FROM employees

WHERE employee\_id = x;

idx := idx + 1;

END LOOP;

FOR i IN 1..emps.count() LOOP

dbms\_output.put\_line(emps(i));

END LOOP;

END;

Output-

Statement processed.

First1

First2

First3

First4

First5

First6

First7

First8

First9

First10

Code-

DECLARE

TYPE e\_list IS TABLE OF employees.first\_name%type;

emps e\_list := e\_list();

idx PLS\_INTEGER := 1;

BEGIN

FOR x IN 1 .. 10 LOOP

emps.extend;

SELECT first\_name INTO emps(idx)

FROM employees

WHERE employee\_id = x;

idx := idx + 1;

END LOOP;

emps.delete(3);

FOR i IN 1..emps.count() LOOP

IF emps.exists(i) THEN

dbms\_output.put\_line(emps(i));

END IF;

END LOOP;

END;

Output-

Statement processed.

First1

First2

First4

First5

First6

First7

First8

First9

Code-

DECLARE

TYPE e\_list IS TABLE OF employees.first\_name%TYPE INDEX BY PLS\_INTEGER;

emps e\_list;

BEGIN

FOR x IN 1.. 10 LOOP

SELECT first\_name

INTO emps(x)

FROM employees

WHERE employee\_id = x ;

END LOOP;

FOR i IN emps.first()..emps.last() LOOP

dbms\_output.put\_line(emps(i));

END LOOP;

END;

Output-

Statement processed.

First1

First2

First3

First4

First5

First6

First7

First8

First9

First10

Code-

DECLARE

TYPE e\_list IS TABLE OF employees.first\_name%TYPE INDEX BY PLS\_INTEGER;

emps e\_list;

BEGIN

FOR x IN 1 .. 10 LOOP

SELECT first\_name

INTO emps(x)

FROM employees

WHERE employee\_id = x ;

END LOOP;

FOR i IN emps.first()..emps.last() LOOP

dbms\_output.put\_line(i);

END LOOP;

END;

Output-

Statement processed.

1

2

3

4

5

6

7

8

9

10

Code-

DECLARE

TYPE e\_list IS TABLE OF employees.first\_name%TYPE INDEX BY PLS\_INTEGER;

emps e\_list;

BEGIN

emps(120) := 'Bob';

emps(121) := 'Sue';

FOR i IN emps.first()..emps.last() LOOP

dbms\_output.put\_line(emps(i));

END LOOP;

END;

Output-

Statement processed.

Bob

Sue

Code-

CREATE OR REPLACE TYPE t\_phone\_number AS OBJECT(p\_type VARCHAR2(10),

p\_number VARCHAR2(50)

);

/

CREATE OR REPLACE TYPE v\_phone\_numbers AS VARRAY(3) OF t\_phone\_number;

/

CREATE TABLE emps\_with\_phones(employee\_id NUMBER,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

phone\_number v\_phone\_numbers);

/

SELECT \* FROM emps\_with\_phones;

/

INSERT INTO emps\_with\_phones

VALUES(10,'Alex','Brown',v\_phone\_numbers(t\_phone\_number('HOME','111.111.1111'),

t\_phone\_number('WORK','222.222.2222'),

t\_phone\_number('MOBILE','333.333.3333'))

);

INSERT INTO emps\_with\_phones

VALUES(11,'Bob','Green',v\_phone\_numbers(t\_phone\_number('HOME','000.000.000'),

t\_phone\_number('WORK','444.444.4444'))

);

Output-

Type created.

Type created.

Table created.

no data found

1 row(s) inserted.

1 row(s) inserted.

Code-

SELECT e.first\_name,

last\_name,

p.p\_type,

p.p\_number

FROM emps\_with\_phones e, table(e.phone\_number) p;

/\*\*\*\*\*\* The Code For the Storing Nested Table Example \*\*\*\*\*\*\*/

CREATE OR REPLACE TYPE n\_phone\_numbers AS TABLE OF t\_phone\_number;

/

CREATE TABLE emps\_with\_phones2(employee\_id NUMBER,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

phone\_number n\_phone\_numbers)

NESTED TABLE phone\_number STORE AS phone\_numbers\_table;

/

SELECT \* FROM emps\_with\_phones2;

/

INSERT INTO emps\_with\_phones2

VALUES(10,'Alex','Brown',n\_phone\_numbers(t\_phone\_number('HOME','111.111.1111'),

t\_phone\_number('WORK','222.222.2222'),

t\_phone\_number('MOBILE','333.333.3333'))

);

INSERT INTO emps\_with\_phones2

VALUES(11,'Bob','Green',n\_phone\_numbers(t\_phone\_number('HOME','000.000.000'),

t\_phone\_number('WORK','444.444.4444'))

);

/

SELECT e.first\_name, last\_name, p.p\_type, p.p\_number

FROM emps\_with\_phones2 e, table(e.phone\_number) p;

Output-

| **FIRST\_NAME** | **LAST\_NAME** | **P\_TYPE** | **P\_NUMBER** |
| --- | --- | --- | --- |
| Alex | Brown | HOME | 111.111.1111 |
| Alex | Brown | WORK | 222.222.2222 |
| Alex | Brown | MOBILE | 333.333.3333 |
| Bob | Green | HOME | 000.000.000 |
| Bob | Green | WORK | 444.444.4444 |

[Download CSV](https://livesql.oracle.com/apex/f?p=590:1:14267886078224:::RP::#)

5 rows selected.

Type created.

Table created.

no data found

1 row(s) inserted.

1 row(s) inserted.

## **Result Set 5**

| **FIRST\_NAME** | **LAST\_NAME** | **P\_TYPE** | **P\_NUMBER** |
| --- | --- | --- | --- |
| Alex | Brown | HOME | 111.111.1111 |
| Alex | Brown | WORK | 222.222.2222 |
| Alex | Brown | MOBILE | 333.333.3333 |
| Bob | Green | HOME | 000.000.000 |
| Bob | Green | WORK | 444.444.4444 |

[Download CSV](https://livesql.oracle.com/apex/f?p=590:1:14267886078224:::RP::#)

5 rows selected.

Code-

INSERT INTO emps\_with\_phones2

VALUES(11,'Bob','Green',n\_phone\_numbers(t\_phone\_number('HOME','000.000.000'),

t\_phone\_number('WORK','444.444.4444'),

t\_phone\_number('WORK2','444.444.4444'),

t\_phone\_number('WORK3','444.444.4444'),

t\_phone\_number('WORK4','444.444.4444'),

t\_phone\_number('WORK5','444.444.4444'))

);

SELECT \* FROM emps\_with\_phones2;

UPDATE emps\_with\_phones2

SET phone\_number = n\_phone\_numbers(t\_phone\_number('HOME','000.000.000'),

t\_phone\_number('WORK','444.444.4444'),

t\_phone\_number('WORK2','444.444.4444'),

t\_phone\_number('WORK3','444.444.4444'),

t\_phone\_number('WORK4','444.444.4444'),

t\_phone\_number('WORK5','444.444.4444'))

WHERE employee\_id = 11;

Output-

1 row(s) inserted.

## **Result Set 6**

| **EMPLOYEE\_ID** | **FIRST\_NAME** | **LAST\_NAME** | **PHONE\_NUMBER** |
| --- | --- | --- | --- |
| 11 | Bob | Green | [unsupported data type] |
| 10 | Alex | Brown | [unsupported data type] |
| 11 | Bob | Green | [unsupported data type] |