Advanced Database

**Lab Sheet**

# OBJECT TYPES & Collection Types

OBJECTS

1. Create an Object type called O\_ADDRESS that has attributes address\_line1, address\_line2, address\_line3, address\_line4, city, country.

CREATE OR REPLACE TYPE o\_address AS OBJECT(

address\_line1 VARCHAR2 (20),

address\_line2 VARCHAR2 (20),

address\_line3 VARCHAR2 (20),

address\_line4 VARCHAR2 (20),

city VARCHAR2 (10),

country VARCHAR2 (10));

1. Create an Object type called O\_SUBJECT that has attributes module\_code and module\_name.

create type o\_subject AS OBJECT(

subject\_code number,

subject\_name varchar2(20));

***Creating Collections VArrays and Nested Tables***

1. Create 2 varying arrays
   * One to hold up to 6 phone numbers. Call it phones\_varray.
   * One to hold up to 16 College Class object. Call it classes\_varray. A college class object is made up of College Class Name (e.g. ADB, OS), Class Type (e.g. Lab, Lecture), Day, Time). For our purposes treat Day and Time as strings.

CREATE OR REPLACE TYPE **phones\_varray** AS VARRAY(6) OF CHAR(12);

create type o\_college\_class AS OBJECT(

class\_name VARCHAR2(20),

type VARCHAR2(20),

day VARCHAR2(20),

time VARCHAR2(20));

CREATE OR REPLACE TYPE **classes\_varray** AS VARRAY(16) OF o\_college\_class;

1. Create a nested table called SUBJECT\_NTABLE that holds an array of subjects

CREATE OR REPLACE TYPE subject\_ntable AS TABLE OF

o\_subject;

1. Create a table STUDENT that contains a
   * student\_id INTEGER
   * full\_name VARCHAR2
   * phone\_nos Varying Array
   * classes Varying Array
   * subjects\_registered SUBJECTS\_NTABLE
   * home\_address O\_ADDRESS
   * next\_of\_kin VARCHAR2
   * nok\_address O\_ADDRESS

CREATE TABLE student(

student\_id INTEGER,

full\_name VARCHAR2(30),

phone\_no phones\_varray,

subjects\_registered subject\_ntable ,

classes classes\_varray,

home\_address o\_address ,

next\_of\_kin VARCHAR2 (30),

nok\_address o\_address )

NESTED TABLE subjects\_registered STORE AS ModuleTable;

1. Populate the STUDENT with data. For Example

INSERT INTO student VALUES(

1001,

'john smith',

phones\_varray('0877407695','013232322'),

subject\_ntable (

o\_subject(1112,'adb'),

o\_subject(1110,'software development'),

o\_subject(1123,'oosd')),

classes\_varray(o\_college\_class('adb','lab','wed','1pm'),o\_college\_class('oosd','lecture','tues','3pm')),

o\_address('24, the glen','belgard Road ','tallaght',null,'dublin','ireland'),

'Mary Smith',

o\_address('24 the glen','belgard road ','tallaght',null,'dublin','ireland')

);

insert into student values(

1003,

'mary jones',

phones\_varray('08774898895','013343434-'),

subject\_ntable (

o\_subject(1112,'adb'),

o\_subject(1110,'sdev'),

o\_subject(1123,'oosd')),

classes\_varray(o\_college\_class('adb','lab','fri','1pm'),o\_college\_class('oosd','lecture','tues','3pm')),

o\_address('2 the rise','marys road','tallaght',null,'dublin','ireland'),

'Mary oshea',

o\_address('123 elm road','terenure ','dublin 5',null,'dublin','ireland')

);

insert into student values(

1005,

'pat murphy',

phones\_varray('0877778895','012345433'),

subject\_ntable (

o\_subject(1112,'adb'),

o\_subject(1110,'os'),

o\_subject(1123,'oosd')),

classes\_varray(o\_college\_class('adb','lab','fri','1pm'),o\_college\_class('oosd','lecture','tues','3pm')),

o\_address('2 the rise','marys road','malahide',null,'dublin','ireland'),

'patricia murphy',

o\_address('12a captains road','terenure ','dublin 5',null,'dublin','ireland')

);

COMMIT;

1. Query the phone number varray in the student table as follows

SELECT phone\_no FROM student

WHERE student\_id =1001;

1. Try using the same query except using TABLE keyword and note the output

SELECT p.\* FROM student s, TABLE(s.phone\_no) p

WHERE student\_id =1001;

Another way

SELECT p.COLUMN\_VALUE FROM student s, TABLE(s.phone\_no) p

WHERE student\_id =1001;

A WORD ON HOW TO QUERY VARRAY AND NESTED TABLES

For demonstration purpose, below is a table containing a VARRAY of But I don't know how to find which divisions have an employee called John working in them. How could I look each value of the VARRAY: if the first value is not John, look at the next and so on until I get the value desired, or it realizes that the value doesn't exist?

CREATE OR REPLACE TYPE emp\_obj

AS OBJECT

(

last\_name VARCHAR (50),

first\_name VARCHAR (50),

birth\_dt DATE

);

/

CREATE OR REPLACE TYPE emp\_va

AS VARRAY (999)

OF emp\_obj;

/

CREATE TABLE division

(

division\_id NUMBER (9) PRIMARY KEY,

division\_name VARCHAR2 (100),

emps emp\_va

);

We know how to get all the items in my VARRAY with a query like:

SELECT e.first\_name

FROM division d

, TABLE (d.emps) e

WHERE d.division\_name = 'Sales';

You can use single elements from the VARRAY in the WHERE clause in

SELECT DISTINCT

d.division\_name

FROM division d

, TABLE (d.emps) e

**WHERE e.first\_name = 'John'**;

Or another way

SELECT division\_name

FROM division d

WHERE EXISTS

( SELECT 'X' -- Why 'X'? See note below.

FROM TABLE (d.emps)

WHERE first\_name = 'John'

);

**Why 'X'?** You could SELECT anything in the subquery: a literal (like I did) a column, SYSDATE, USER or a list of any kind(s) of values. The only crucial thing is whether or not the subquery returns any rows. SELECTing a literal is faster than SELECTing a real column or SYSDATE, so that's what I did.

9. Now display the student id and fullname from students that attend **adb** class on **Friday** (change the values to suit your data)

SELECT s.student\_id,s.full\_name

FROM student s,

TABLE (s.classes) c

WHERE c.class\_name='adb' AND c.day='fri';

SELECT s.student\_id, s.full\_name

FROM student s

WHERE EXISTS

( SELECT 'X'

FROM TABLE (s.classes) c

WHERE c.class\_name='adb' AND c.day='fri'

);

9. What students are registered for subjects sdev or adb (change the values to suit your data and use the nested table column). Only one row should be displayed for each student registered.

SELECT distinct s.student\_id, s.full\_name

FROM student s,

TABLE (s.subjects\_registered) sr

WHERE sr.subject\_name='adb' OR sr.subject\_name='sdev';

SELECT s.student\_id, s.full\_name

FROM student s

WHERE EXISTS

( SELECT 'X'

FROM TABLE (s.subjects\_registered) sr

WHERE sr.subject\_name='adb' OR

sr.subject\_name='sdev'

);