Big Data Technologies

Lab Sheet

OBJECT TYPES & Collection Types

OBJECTS

1. Create an Object type called O_ADD that has attributes address_line1, address_line2, address_line3, address_line4, city, country.

```
CREATE OR REPLACE TYPE O_ADD AS OBJECT(
address_line1 VARCHAR2 (20),
address_line2 VARCHAR2 (20),
address_line3 VARCHAR2 (20),
address_line4 VARCHAR2 (20),
city VARCHAR2 (10),
country VARCHAR2 (10));
```

2. 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),
credits INTEGER);
```

Creating Collections VArrays and Nested Tables

3. Create 2 varying arrays
    - One to hold up to 6 phone numbers. Call it phones_varray.
    - One to hold up to 16 College Class objects
        - A Class Object is made up of College
        - ClassName a Varchar (e.g. of values it will hold BDT, OS, Networking),
        - ClassType a Varchaer (e.g. of values it will hold. Lab, Lecture, Tutorial),
        - Day – a Varchar
        - Time – a Varchar
        - [for our purposes treat day and time as strings]

```
CREATE OR REPLACE TYPE phones_varray AS VARRAY(6) OF
VARCHAR2(12);

CREATE TYPE o_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_class;
```

4. Create a nested table called SUBJECT_NTABLE that holds an array of subjects

```
CREATE OR REPLACE TYPE subject_ntable  AS TABLE OF
              o_subject;
```

5. Create a table STUDENT that contains a

student_id	INTEGER
full_name	VARCHAR2
phone_nos	Varying Array you created above
classes	Varying Array you created above
subjects_registered	Nested Table you created earlier
home_address	O_ADD
next_of_kin	VARCHAR2(30)
nok_address	O_ADD

```
CREATE TABLE tud_student(
student_id           INTEGER,
full_name            VARCHAR2(30),
phone_no             phones_varray,
subjects_registered  subject_ntable,
classes              classes_varray,
home_address         o_add,
next_of_kin          VARCHAR2 (30),
nok_address          o_add )
NESTED TABLE subjects_registered STORE AS
SubjectExternalTable;
```

6. Populate the STUDENT with data. For Example

```
INSERT INTO tud_student VALUES(
1001,
'john smith',
phones_varray('0877407695','013232322'),
subject_ntable (
                o_subject(1112,'bdt',5),
                o_subject(1110,'sdev',5),
                o_subject(1123,'oosd',5)),
classes_varray(o_class('adb','lab','wed','1pm'),
               o_class('oosd','lecture','tues','3pm')),
o_add('24, the glen','belgard Road
','tallaght',null,'dublin','ireland'),
'Mary Smith',
o_add('24 the glen','belgard road
','tallaght',null,'dublin','ireland')
);


insert into tud_student values(
1003,
'mary jones',
phones_varray('08774898895','013343434-'),
subject_ntable (
                o_subject(1112,'bdt',5),
                o_subject(1110,'sdev',5),
                o_subject(1123,'oosd',5)),
classes_varray(o_class('adb','lab','fri','1pm'),
               o_class('oosd','lecture','tues','3pm')),
o_add('2 the rise','marys
road','tallaght',null,'dublin','ireland'),
'Mary oshea',
o_add('123 elm road','terenure ','dublin
5',null,'dublin','ireland')
);

insert into tud_student values(
1005,
'pat murphy',
phones_varray('0877778895','012345433'),
subject_ntable (
                o_subject(1112,'adb',5),
                o_subject(1110,'os',5),
                o_subject(1123,'oosd',5)),
classes_varray(o_class('adb','lab','fri','1pm'),
               o_class('oosd','lecture','tues','3pm')),
o_add('2 the rise','marys
road','malahide',null,'dublin','ireland'),
'patricia murphy',
o_add('12a captains road','terenure ','dublin
5',null,'dublin','ireland')
);
COMMIT;
```

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

```
        SELECT phone_no FROM student
        WHERE  student_id =1001;

        The data is listed as one coleumn
```

8. 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;
```

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

```
SELECT s.student_id,s.full_name
FROM tud_student s,
TABLE (s.classes) c
WHERE c.class_name='bdt' AND c.day='fri';

Or

SELECT s.student_id, s.full_name
FROM tud_student s
WHERE   EXISTS
  ( SELECT  'X'
            FROM    TABLE (s.classes) c
            WHERE c.class_name='bdt' AND c.day='fri'
    );
```

10. What students are registered for subjects sdev or bdt (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 tud_student s,
TABLE (s.subjects_registered) sr
WHERE sr.subject_name='bdt' OR sr.subject_name='sdev';

OR

SELECT s.student_id, s.full_name
FROM tud_student s
WHERE   EXISTS
  ( SELECT  'X'
            FROM    TABLE (s.subjects_registered) sr
            WHERE sr.subject_name='bdt' OR
            sr.subject_name='sdev'
    );
```

11. List out the class timetable for a particular student. Order it by the type of class e.g.tutorial, lab.

```
SELECT  cv.class_name, cv.type, cv.day,cv.time
FROM tud_student s,
TABLE (s.classes) cv
WHERE full_name='mary jones'
ORDER BY cv.type;
```

12. Show all the class details only for a class type e.g. lecture for a particular student.

```
SELECT  cv.class_name, cv.type, cv.day,cv.time
FROM tud_student s,
TABLE (s.classes) cv
WHERE full_name='mary jones'
AND cv.type = 'lecture';
```