USER - DEFINE DATATYPES:

USER DEFINE DATATYPES ARE INTRODUCED IN ORACLE 8.0 VERSION.WHEN PRE-DEFINE DATATYPES ARE NOT REACHING TO OUR REQUIREMENTS THEN WE CREATE OUR OWN DATATYPES ARE CALLED AS USER DEFINE DATATYPES.

THE ADVANTAGE OF USER DEFINE DATATYPES ARE REUSABILITY THAT MEANS WE CAN CREATE DATATYPE AND REUSE IN MULTIPLE TABLES.ORACLE SUPPORTS THE FOLLOWING THREE TYPES OF USERS DEFINE DATATYPES.

1. OBJECT TYPE (OR) COMPOSITE TYPE
2. VARRAY
3. NESTED TABLE.

1. OBJECT TYPE (OR) COMPOSITE TYPE:

IT ALLOWS GROUP OF VALUES /ELEMENTS OF DIFFERENT DATATYPES.

SYNATAX:

```
CREATE TYPE <TYPE NAME> AS OBJECT(<COL1>
DATATYPE[SIZE], <COL2> DATATYPE[SIZE], ........); /
```

EX:

```
CREATE TYPE COURSE_TYPE AS OBJECT (CID NUMBER (4), CNAME
VARCHAR2(10), FEE NUMBER (10));

/
```

TESTING:

EX:

```
CREATE TABLE STUDENTS (SID NUMBER (4), SNAME
VARCHAR2(10), COURSE COURSE_TYPE);

TABLE CREATED.


SQL> INSERT INTO STUDENTS VALUES (101,'SAI’, COURSE_TYPE
(1021,'ORACLE',1200));

SQL> INSERT INTO STUDENTS VALUES (102,'WARD’,
COURSE_TYPE (1022,'C',500));
```

TO SELECT:

SQL> SELECT S.SID, S. SNAME, S.COURSE.CID, S. COURSE.CNAME, S.COURSE.FEE FROM STUDENTS S;

(OR)

SQL> SELECT S.SID, S. SNAME, S.COURSE.CID CID, S. COURSE.CNAME CNAME, S.COURSE.FEE FEE FROM STUDENTS S;

TO UPDATE:

SQL> UPDATE STUDENTS S SET S.COURSE.FEE=2000 WHERE S.SID=101;

TO DELETE:

SQL> DELETE FROM STUDENTS S WHERE S.COURSE.CID=1022;

2. VARRAY:

IT ALLOWS GROUP OF VALUES /ELEMENTS OF SAME DATATYPES.VARRAY SHOULD DECLARE WITH SIZE.

SYNTAX:

CREATE TYPE <TYPE NAME> IS VARRAY(SIZE) OF DATATYPE[SIZE];

/

EX:

CREATE TYPE MBNO_ARRAY1 IS VARRAY (3) OF NUMBER (10); /

TESTING:

SQL> CREATE TABLE EMPLOYEE (EMPNO NUMBER (4), MBNO MBNO_ARRAY1);

TABLE CREATED.
SQL> INSERT INTO EMPLOYEE VALUES (1021, MBNO_ARRAY1(9703542749,8502045789));

SQL> INSERT INTO EMPLOYEE VALUES (1022, MBNO_ARRAY1(9632587412,8523691478,7412356896));

3.NESTED TABLE:

> A TABLE WITHIN ANOTHER TABLE IS CALLED AS NESTED TABLE.

> NESTED TABLE ALSO ALLOW GROUP OF VALUES /ELEMENTS OF DIFF. DATATYPES.

> NESTED TABLE IS NOT DECLARE WITH SIZE.

STEPS TO CREATE NESTED TABLE:

STEP1: CREATE AN OBJECT TYPE:

SYNATAX:

CREATE TYPE <TYPE NAME> AS OBJECT(<COL1> DATATYPE[SIZE], <COL2> DATATYPE[SIZE], ...); /

STEP2: CREATE NESTED TABLE TYPE:

SYNTAX:

CREATE TYPE <TYPE NAME> AS TABLE OF <OBJECT TYPE NAME>; /

STEP3: CREATE A TABLE:

SYNTAX:

CREATE TABLE <TN>(<COL1> <DATATYPE>[SIZE], ..., <COL N> <NESTED TABLE TYPE NAME>)

NESTED TABLE <COL N NAME> STORE AS <ANY NAME>;
EX:

STEP1: CREATE TYPE ADDR_TYPE AS OBJECT (HNO NUMBER (4), STREET VARCHAR2(10), CITY VARCHAR2(10));

/

STEP2:

```
CREATE TYPE ADDR_ARRAY AS TABLE OF ADDR_TYPE; /
```

STEP3:

```
CREATE TABLE CUSTOMER (CID NUMBER (4), CNAME
VARCHAR2(10), CADDRESS ADDR_ARRAY) NESTED TABLE
CADDRESS STORE AS CUST_ADDR;
```

TESTING:

```
SQL> INSERT INTO CUSTOMER VALUES (1,'SAI’, ADDR_ARRAY
(ADDR_TYPE (1122,'GANDHI','HYD')));

SQL> INSERT INTO CUSTOMER VALUES (2,'WARD’, ADDR_ARRAY
(ADDR_TYPE (1123,'ASHOK','CHE'), ADDR_TYPE
(1124,'VASATI','MUM')));
```

NOTE: WE CAN ALSO SELECT, UPDATE, DELETE, INSERT DATA WITHIN NESTED TABLE BY USING THE FOLLOWING SYNTAX,

SYNTAX:

```
SELECT / UPDATE / DELETE / INSERT (SELECT <NESTED TABLE  TYPE
COLUMN NAME> FROM <TN>);
```

EX:

```
SQL> SELECT * FROM TABLE (SELECT CADDRESS FROM CUSTOMER
WHERE CID=1);

SQL> UPDATE TABLE (SELECT CADDRESS FROM CUSTOMER
WHERE CID=2) SET HNO=1024 WHERE HNO=1124;
SQL> DELETE FROM TABLE (SELECT CADDRESS FROM CUSTOMER
WHERE CID=2) WHERE CITY='MUM';

SQL> INSERT INTO TABLE (SELECT CADDRESS FROM CUSTOMER
WHERE CID=1) VALUES (1124,'YUVIN','HYD');
```

NOTE: IN ORACLE WE WANT TO VIEW USER TYPES THEN FOLLOW THE

FOLLWING DATADICTIONARY IS "USER_TYPES".

EX:

```
SQL> DESC USER_TYPES;
SQL> SELECT TYPE_NAME FROM USER_TYPES;
```

SYNTAX TO DROP TYPE:

```
SQL> DROP TYPE <TYPE NAME> FORCE;
```

EX:

```
DROP TYPE MBNO_ARRAY1 FORCE;
```