Module No.3 :- ORDBMS : Implementation of Abstract Data Type Reference

Practical No.3

Aim: - Implementation of ORDBMS using ADT (Abstract Data Types), References, etc.

Objective: - To learn ORDBMS using ADT (Abstract Data Types), References

Implementation:-

Abstract Data Types:-

Step 1:- Creating user-defined type

Query: -

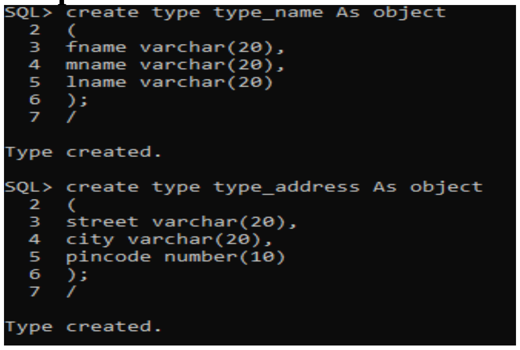
Create type for name:-

```
create type type_name5 As object
(
fname varchar(20),
mname varchar(20),
lname varchar(20)
);
/
```

Create type for address:-

```
create type type_address5 As object
(
street varchar(20),
city varchar(20),
pincode number(10)
);
/
```

Output:-



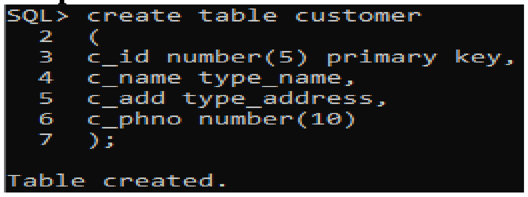
```
SQL> create type type_name As object
  (
  fname varchar(20),
  mname varchar(20),
  lname varchar(20)
  );
  /

Type created.

SQL> create type type_address As object
  (
  street varchar(20),
  city varchar(20),
  pincode number(10)
  );
  /

Type created.
```

Step 2:- Creating table using above type

Query:-

```
create table customer5
(
c_id number(5) primary key,
c_name type_name,
c_add type_address,
c_phno number(10)
);
```

Output:-



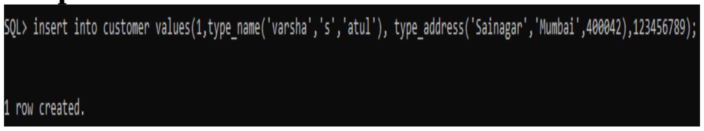
```
SQL> create table customer
  (
  c_id number(5) primary key,
  c_name type_name,
  c_add type_address,
  c_phno number(10)
  );

Table created.
```

Step 3:- Inserting records into the table

Query: -

```
insert into customer5
values(1,type_name('varsha','s','atul'),
type_address('Sainagar','Mumbai',400042),123456789
);
```

Output:-



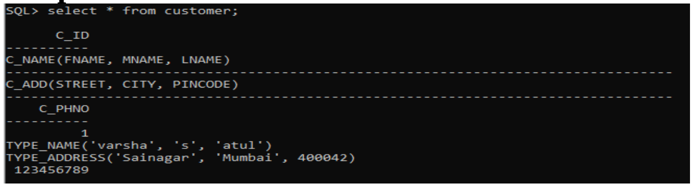
```
SQL> insert into customer values(1,type_name('varsha','s','atul'), type_address('Sainagar','Mumbai',400042),123456789);


1 row created.
```

Step 4:- Display records from the table

Query:-

```
select * from customer;
```

Output:-



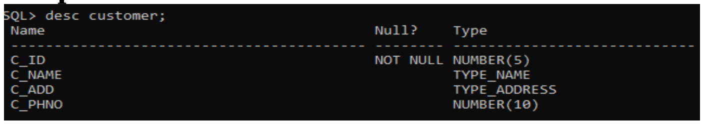
```
SQL> select * from customer;

      C_ID
----------
C_NAME(FNAME, MNAME, LNAME)
--------------------------------------------------------------------------------
C_ADD(STREET, CITY, PINCODE)
--------------------------------------------------------------------------------
    C_PHNO
----------
         1
TYPE_NAME('varsha', 's', 'atul')
TYPE_ADDRESS('Sainagar', 'Mumbai', 400042)
 123456789
```

Step 5:- Describe customer table

Query:-

desc customer;

Output:-

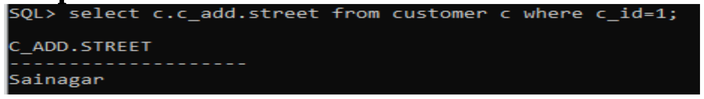


```
SQL> desc customer;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 C_ID                                      NOT NULL NUMBER(5)
 C_NAME                                             TYPE_NAME
 C_ADD                                              TYPE_ADDRESS
 C_PHNO                                             NUMBER(10)
```

Q. Display street name from customer table using abstract data type

Query: -
select c.c_add.street from customer c where c_id=1;

Output:-

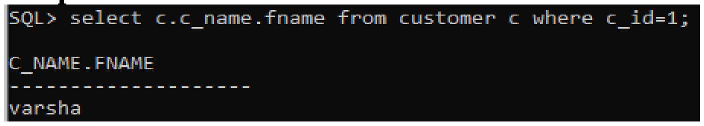


```
SQL> select c.c_add.street from customer c where c_id=1;

C_ADD.STREET
--------------------
Sainagar
```

Q. Display customer's first name from customer table using abstract data type

Query: -
select c.c_name.fname from customer c where c_id=1;

Output:-

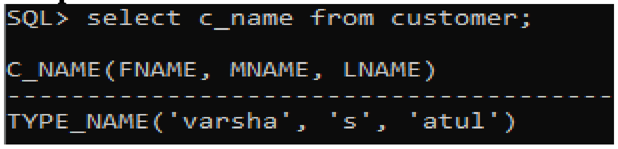


```
SQL> select c.c_name.fname from customer c where c_id=1;

C_NAME.FNAME
--------------------
varsha
```

Q. Display customer name from customer table using abstract data type

Query: -
select c_name from customer;

Output:-

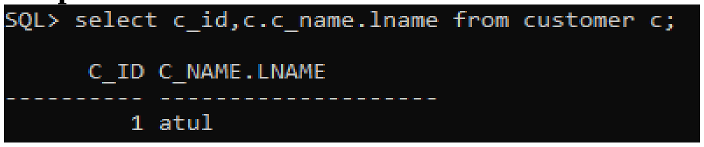


```
SQL> select c_name from customer;

C_NAME(FNAME, MNAME, LNAME)
--------------------------------------------------
TYPE_NAME('varsha', 's', 'atul')
```

Q. Display customer's last name with customerid from customer table using abstract data type

Query: -
select c_id,c.c_name.lname from customer c;

Output:-

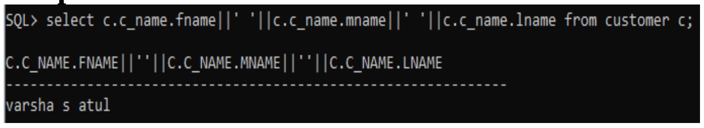


```
SQL> select c_id,c.c_name.lname from customer c;

      C_ID C_NAME.LNAME
---------- --------------------
         1 atul
```

Q. Display customer name in single column from customer table using abstract data type

Query: -
select c_id,c.c_name.lname from customer c;

Output:-



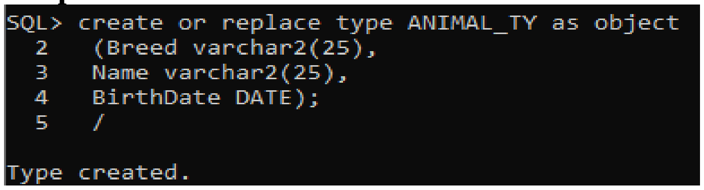
```
SQL> select c.c_name.fname||' '||c.c_name.mname||' '||c.c_name.lname from customer c;

C.C_NAME.FNAME||''||C.C_NAME.MNAME||''||C.C_NAME.LNAME
--------------------------------------------------------------
varsha s atul
```

REF and DREF function:-

Step 1:- Creating Object type

Query:-
create or replace type ANIMAL_TY as object
 (Breed varchar2(25),
 Name varchar2(25),
 BirthDate DATE);
 /

Output:-

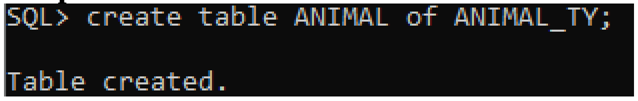


```
SQL> create or replace type ANIMAL_TY as object
  2   (Breed varchar2(25),
  3   Name varchar2(25),
  4   BirthDate DATE);
  5   /

Type created.
```

Step 2 :- Creating table using above type

Query :-
create table ANIMAL of ANIMAL_TY;

Output:-



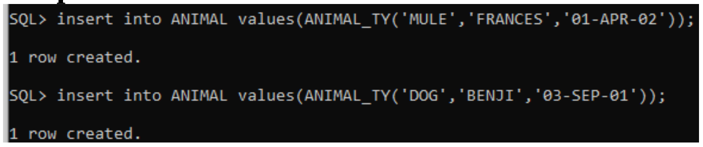
```
SQL> create table ANIMAL of ANIMAL_TY;

Table created.
```

Step 3 :- Inserting records into table

Query :-

insert into ANIMAL values(ANIMAL_TY('MULE','FRANCES','01-APR-02'));
insert into ANIMAL values(ANIMAL_TY('DOG','BENJI','03-SEP-01'));

Output:-

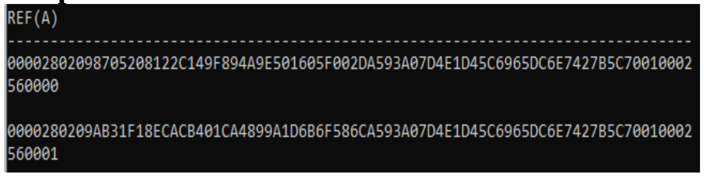


```
SQL> insert into ANIMAL values(ANIMAL_TY('MULE','FRANCES','01-APR-02'));

1 row created.

SQL> insert into ANIMAL values(ANIMAL_TY('DOG','BENJI','03-SEP-01'));

1 row created.
```

The REF Function:-

Query :-

Output:-



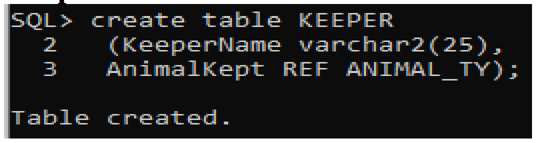
```
REF(A)
--------------------------------------------------------------------------------
000028020987052081 22C149F894A9E501605F002DA593A07D4E1D45C6965DC6E7427B5C70010002
560000

0000280209AB31F18ECACB401CA4899A1D6B6F586CA593A07D4E1D45C6965DC6E7427B5C70010002
560001
```

Using the DEREF Function:-

Q. Creating table using above type

Query:-
create table KEEPER
(KeeperName varchar2(25),
AnimalKept REF ANIMAL_TY);

Output:-

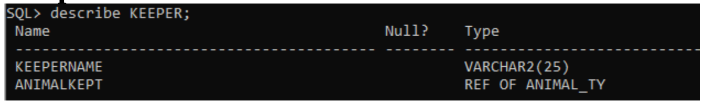


```
SQL> create table KEEPER
  2   (KeeperName varchar2(25),
  3   AnimalKept REF ANIMAL_TY);

Table created.
```

Q. describing table

Query:-
describe KEEPER

Output:-

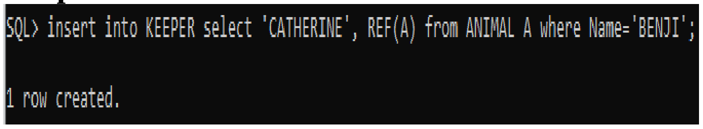


```
SQL> describe KEEPER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 KEEPERNAME                                         VARCHAR2(25)
 ANIMALKEPT                                         REF OF ANIMAL_TY
```

Q. Inserting records into table

Query:-
insert into KEEPER select 'CATHERINE', REF(A) from ANIMAL A where Name='BENJI';

Output:-

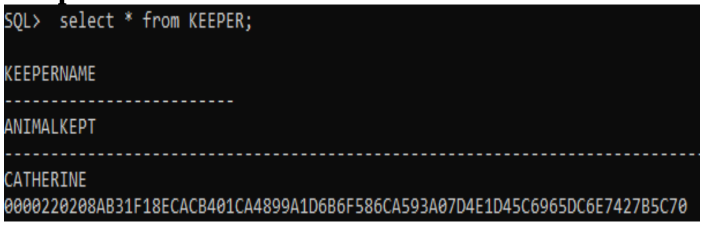


```
SQL> insert into KEEPER select 'CATHERINE', REF(A) from ANIMAL A where Name='BENJI';

1 row created.
```

Q. Display records from table

Query:-
select * from KEEPER;

Output:-

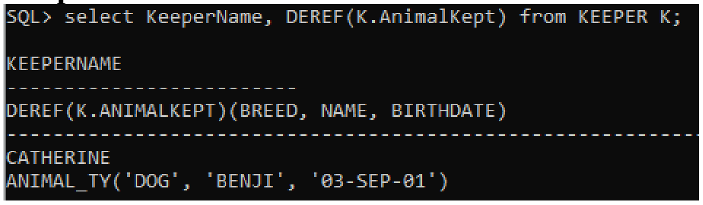


```
SQL>  select * from KEEPER;

KEEPERNAME
-------------------------
ANIMALKEPT
--------------------------------------------------------------------------------
CATHERINE
0000220208AB31F18ECACB401CA4899A1D6B6F586CA593A07D4E1D45C6965DC6E7427B5C70
```

Q. Display records from table using DREF function.

Query:-
select KeeperName, DEREF(K.AnimalKept) from KEEPER K;

Output:-



```
SQL> select KeeperName, DEREF(K.AnimalKept) from KEEPER K;

KEEPERNAME
-------------------------
DEREF(K.ANIMALKEPT)(BREED, NAME, BIRTHDATE)
--------------------------------------------------------------------------------
CATHERINE
ANIMAL_TY('DOG', 'BENJI', '03-SEP-01')
```