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

**ORDBMS stands for Object-Relational Database Management System**

It provides all the facilities of RDBMS with the additional support of object oriented concepts. The concept of classes, objects and inheritance are supported in this database. It is present in the ground level between the RDBMS and OODBMS. In this data can be manipulated by using any query language. It is complex because it has to take care of both Relational database concepts and as well as Object Oriented concepts

Examples of ORDBMSs include:

• Oracle Database by Oracle Corporation.

• Informix by IBM

• SQL Server by Microsoft

• Greenplum Database by Pivotal Software

**Advantages of ORDBMS :-**

**Reuse and Sharing:-** The main advantages of extending the Relational data model come from reuse and sharing. Reuse comes from the ability to extend the DBMS server to perform standard functionality centrally, rather than have it coded in each application.

**Increased Productivity:-** ORDBMS provides increased productivity both for the developer and for the, end user

**Use of experience in developing RDBMS:-** Another obvious advantage is that .the extended relational approach preserves the significant body of knowledge and experience that has gone into developing relational applications. This is a significant advantage, as many organizations would find it prohibitively expensive to change. If the new functionality is designed appropriately, this approach should allow organizations to take advantage of the new extensions in an evolutionary way without losing the benefits of current database features and functions.

**Disadvantages of ORDBMS:-** The ORDBMS approach has the obvious disadvantages of complexity and associated increased costs. Further, there are the proponents of the relational approach that believe the· essential simplicity’ and purity of the .relational model are lost with these types of extension.

**Abstract Data Types (ADT):-** By using abstract data types, which are user-defined types, together with various routines, you can uniquely define and use data with complex structures and perform operations on such data. When you define a column as having an abstract data type, you can conceptualize and model its data based on object-oriented concepts. In addition, by applying object-oriented software development techniques, you can reduce the workload for database design, UAP development, and maintenance

**REF Function:-**

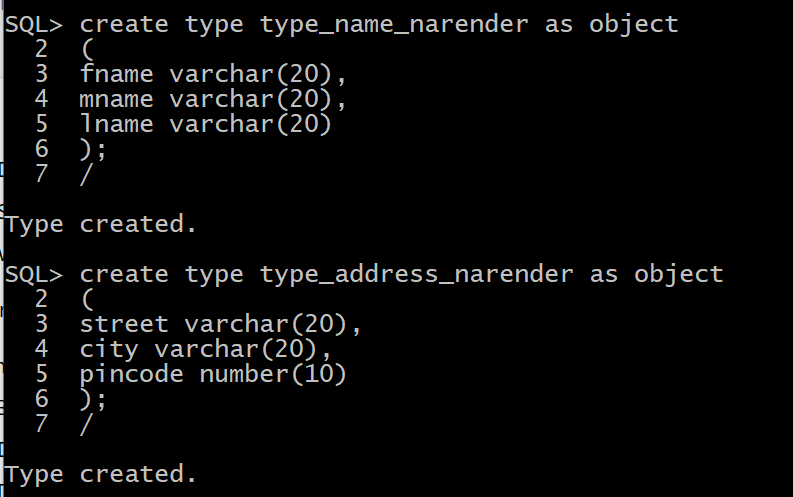
In Oracle PL/SQL, REF data types are pointers that uniquely identify a piece of data as an object. A reference can be established between an existent valid object and a table or type attribute using the REF pointer data type. An attribute referring to a nonexistent object leads to "dangling" situation. Note that a NULL object reference is different from a Dangling Reference. To insert data into a ref column, the REF function is used to get an object instance reference

**DEREF Function:-**

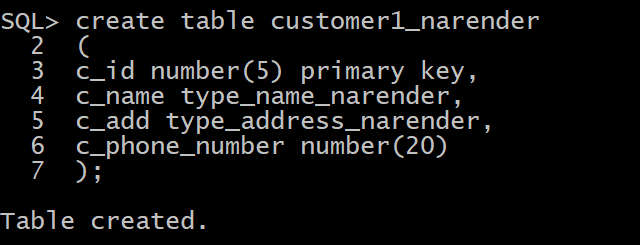
DEREF returns the object reference of argument expr, where expr must return a REF to an object. If you do not use this function in a query, then Oracle Database returns the object ID of the REF instead.

1. **ABSTRACT DATA TYPE:**

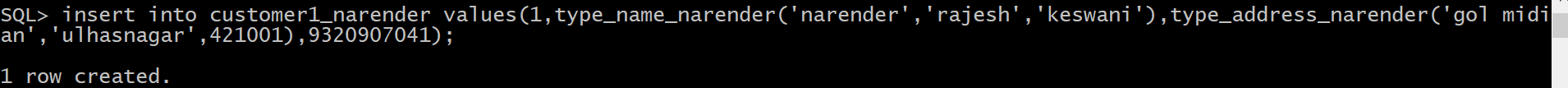
**Creating type : type\_name\_narender and type\_address\_narender**



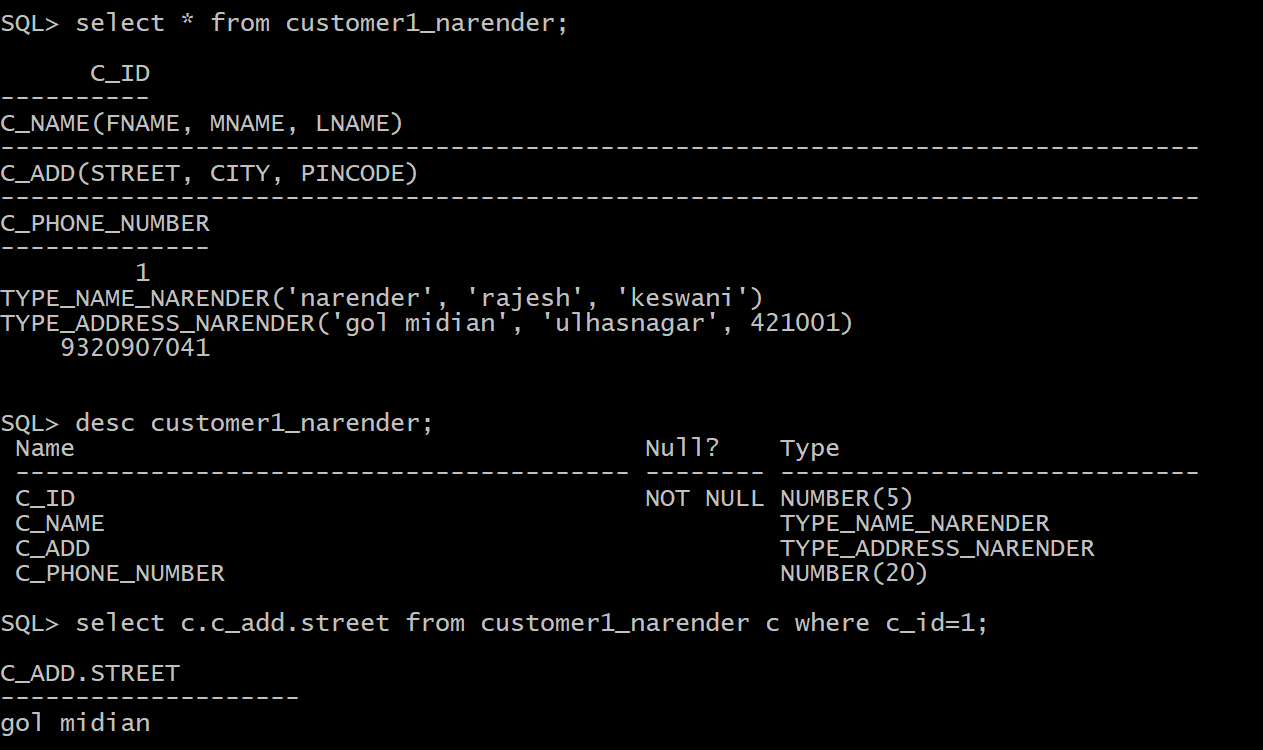
**Creating table customer1\_narender:**



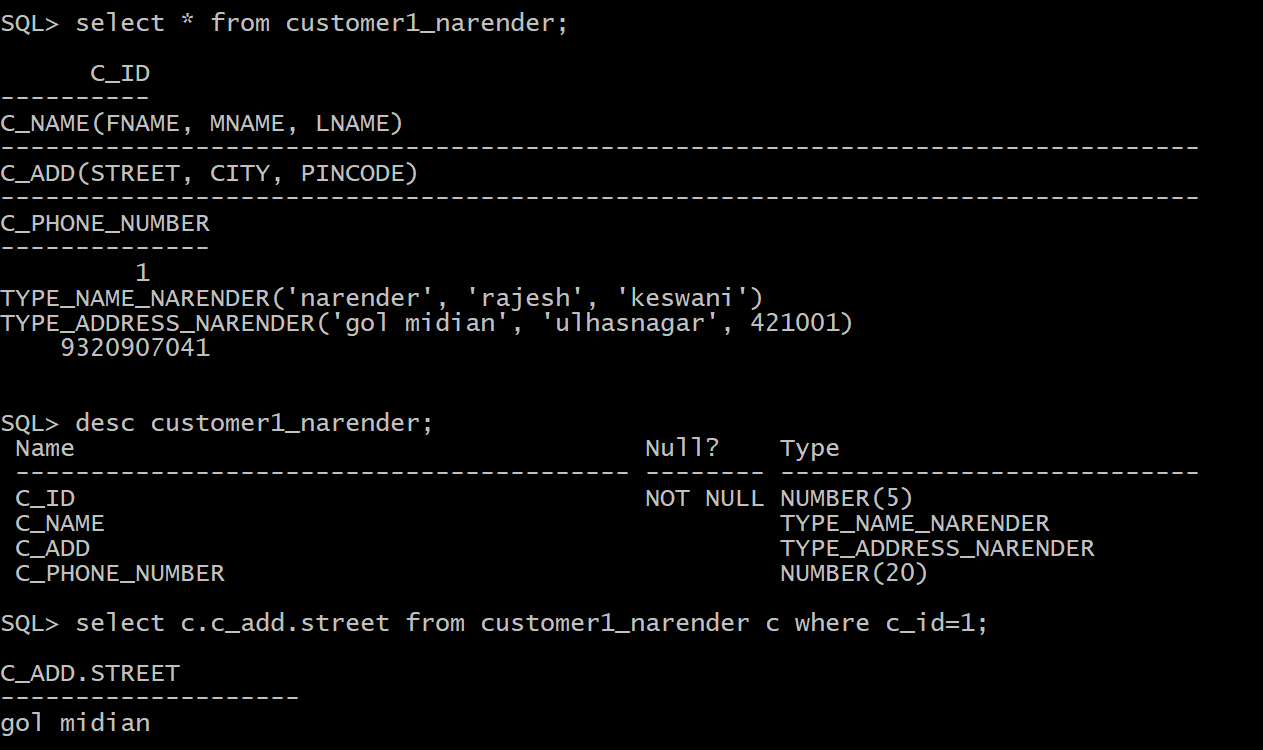
**Inserting the records in the customer1\_narender table:**



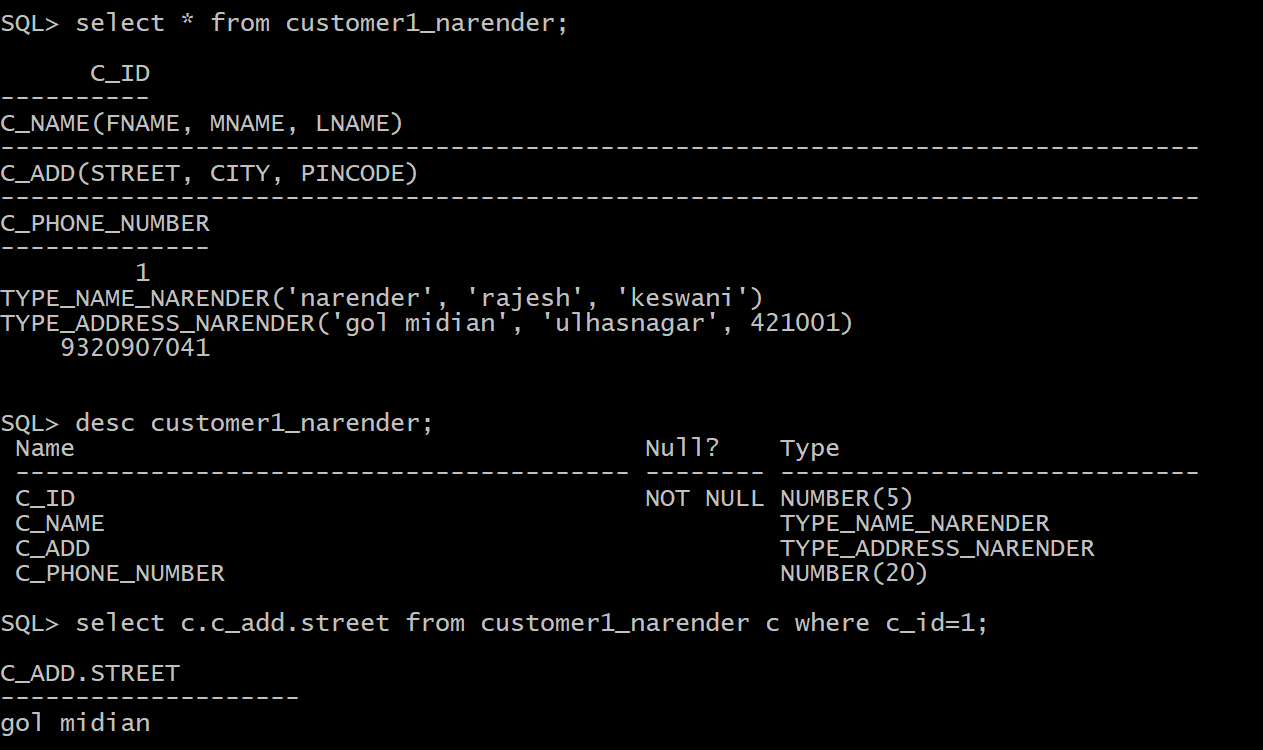
**Displaying all records:**



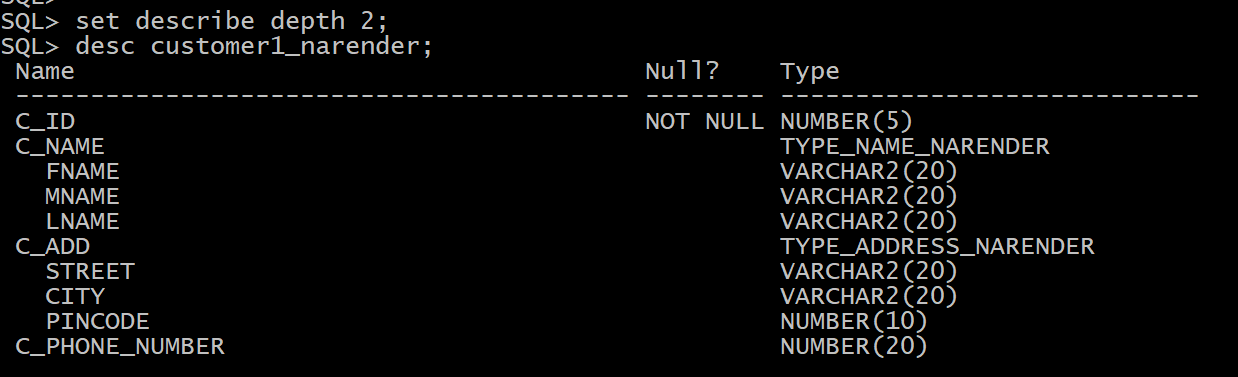
**Checking the structure of the table:**



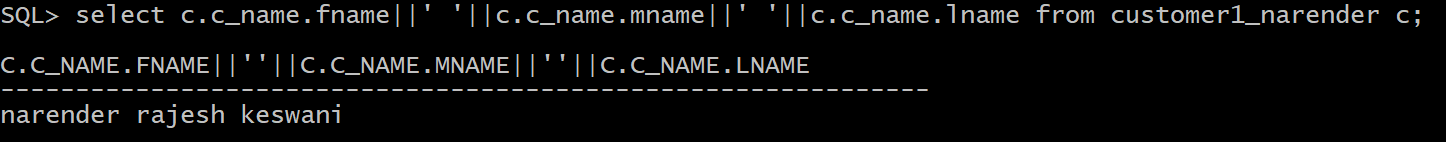
**Displaying only street of the customer of c\_id=1:**



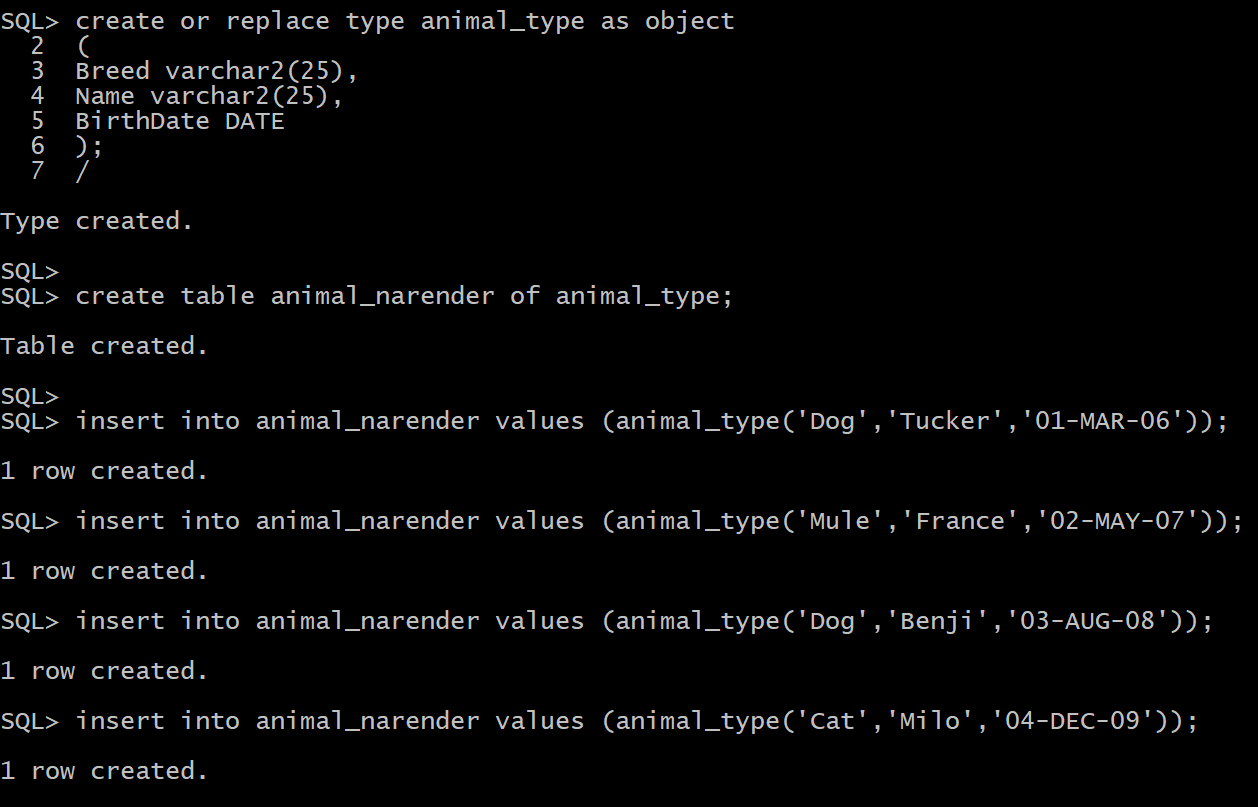
**Viewing in depth structure of the table:**

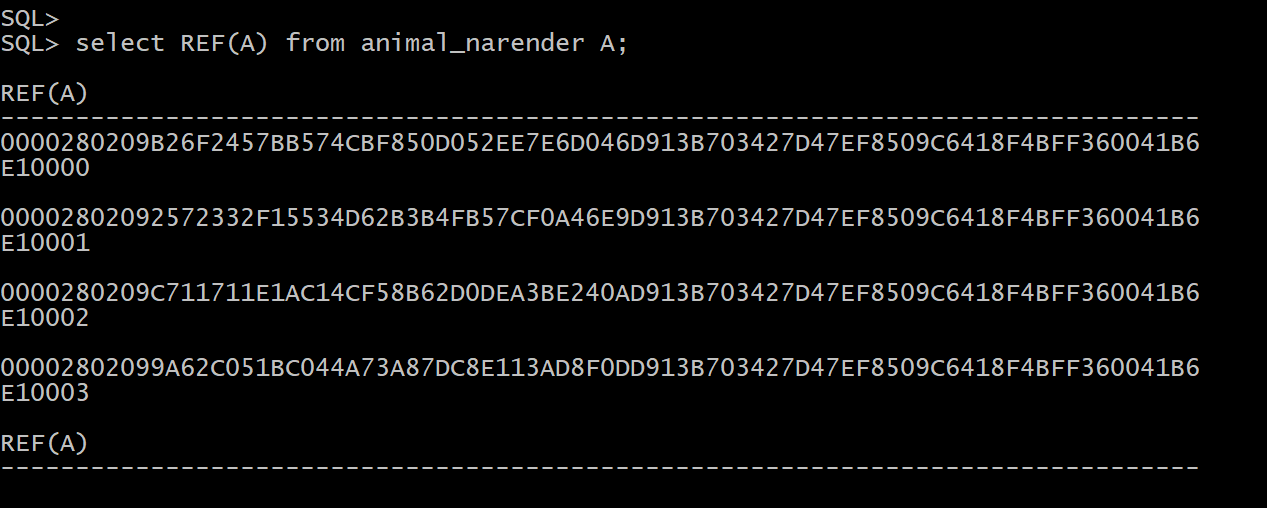


**Displaying first name , middle name , last name of the customer1\_narender table:**



1. **REF:**





1. **DEREF:**

