Oracle PL/SQL – Object-oriented Programming

**ORACLE OBJECT-ORIENTED FEATURES**

Object technology can potentially eliminate database complexity and increase performance. In the previous chapter you saw that databases could be very complex. The attributes are contained in multiple tables.

Object technology can potentially eliminate database complexity and increase performance.

A second reason for using objects is a possible increase in performance. Oracle’s object technology is based upon Rowids. A Rowid identifies the location of a record in the database. Sets of attributes and tables that would normally be in related but independent tables can be embedded directly into a table. This causes related attribute records to be associated to the parent record by Rowid rather than a primary key.

**User-defined data types (ADT’s)**

ADT’s are user-defined data types. ADT’s allow the database designer to employ object-oriented techniques in database design. There are two categories of ADT’s. These are “object types” and “collection types”. Object types enable a database designer to use the same database components for different entities. This reusability is an important component of object-oriented design called inheritance. Collection types allow the developer to embed repeating sets of attributes into a table, forming an object.

**Object types**

Address Object type used in the Employees and Customer tables

**Creating an object type and tables that use the object type**

```
Create type Address as object
  street_number  varchar2(10),
  street         varchar2(30),
  city           varchar2(20),
  state          varchar2(2),
  zip            varchar2(7));

Create table employees
  (payroll_number          number,
   classification          varchar2(25),
   wages                    number(7,2),
   employment_date         date,
   department              varchar2(3),
   address_t               Address);  

Create table customers
  (account_number          number,
   meter_type              varchar2(15),
   billing_style           varchar2(2),
   account_rep             varchar2(20),
   address_t               Address);  
```

**Collection types**

**Varrays**

Defining and using a varray with a table

```
Create type readings_v as varray(24) as number;

Create table customers
  (account_number          number,
   meter_type              varchar2(15),
   billing_style           varchar2(2),
   account_rep             varchar2(20),
   readings                readings_v);  
```

End listing
Object tables

There are two ways that object tables differ from varrays. These are:

1. An object table record can contain more than one element or field per row.
2. The number of records in an object table does not have to be defined in advance. Records can be continually added to the object table.

Defining a nested object table

CREATE TYPE READINGS_O AS OBJECT
  (READING_DATE DATE,
   READING NUMBER);

CREATE TYPE READINGS_TABLE AS TABLE OF READINGS_O;

CREATE TABLE CUSTOMERS
  (ACCOUNT_NUMBER NUMBER,
   METER_TYPE VARCHAR2(15),
   BILLING_STYLE CHAR(2),
   ACCOUNT_REP VARCHAR2(20),
   READINGS READINGS_TABLE);

END LISTING