More Object Relational!

1. Create an Object type called MO_PERSON that has attributes **name**, and **email address**. (all varchars!)

2. Create an Object type called MO_CONTACT that has attributes **MO_PERSON** , and a **phone_number**.

3. Create an Object Table called MyContacts based on MO_CONTACT that contains contacts with the Primary Key of email address and insert an object. Note, the person Object's name attribute is mandatory and the phone_number is optional. INSERT 5 objects and make the permanent. List the email address and phone number for one of the contacts.

4. Create a table called MO_CUSTOMER_TAB that contains a CUST_ID integer (Primary Key), mandatory columns, CONTACT_DETAILS (OF TYPE MO_CONTACT AND IS MANDATORY), COMPANY NAME, CREDIT_LIMIT must be between €1600 and €10,000 and the constraint name to be given is CUSTOMER_CREDIT_CHECK . A MARKET column must be either national or international and the constraint must be named CUSTOMER_MARKET_CHECK.

5. Populate the MO_CUSTOMER_TAB TABLE with a record. Then undo your INSERT by removing it from your database. (NOTE: Oracle APEX auto-commits so just write the command down!)

Go to next page

Part B Object Relation Databases

1. The database designer has been looking at Oracle's Object relational features and has decided to convert the relational structure below with the following requirements **[Please note what are primary keys, mandatory and optional when creating the relevant tables]:**

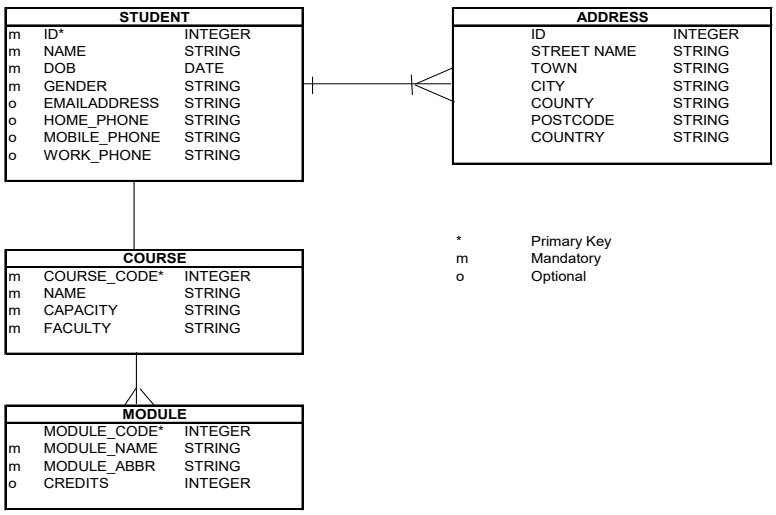
   Please read through the whole set of requirements before starting.

- Student Table called STUDENT_UnderGrad will reference a MO_Course Object(see table below for attributes) in an Object table for courses using **REF**.
-
- Student Table called STUDENT_UnderGrad with an appropriate VARRAY collection type for phone objects. Create appropriate Phone object type. It will have a string attribute called PHONE_CONTACT_TYPE that will contain a value to represent the type of phone e.g. home, work etc. and another attribute that contains the phone number.

- Addresses for student will be implemented as a nested table in the student table. An address will have its own object type call MO_Address

- An object type MO_Module(containing the attributes in the Module table) with an appropriate object table will reference the course objects table using **REF**. Note how the 1:m relationship is now implemented.

-



2. Populate the 3 tables you have created with data. Note where the REFs are. **Hint** populate in the order of the Course object Table, the Module Object table and then the Student relational table.

3. Write a query that returns the the student name, the course they are on and the faculty they are in.

4. Write a query that provides the phone numbers and street names for a particular student