***Rosendo B. Redondo Jr.BSIT 3-A***

***PRE-TEST***

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*MODULE2:***

1. **What are the origins of the object-oriented approach?**

The object-oriented paradigm took its shape from the initial concept of a new programming approach, while the interest in design and analysis methods came much later.

* The first object–oriented language was Simula (Simulation of real systems) that was developed in 1960 by researchers at the Norwegian Computing Center.
* In 1970, Alan Kay and his research group at Xerox PARK created a personal computer named Dynabook and the first pure object-oriented programming language (OOPL) - Smalltalk, for programming the Dynabook.
* In the 1980s, Grady Booch published a paper titled Object Oriented Design that mainly presented a design for the programming language, Ada. In the ensuing editions, he extended his ideas to a complete object–oriented design method.
* In the 1990s, Coad incorporated behavioral ideas to object-oriented methods.

1. **What primary characteristics should an OID possess?**

* OIDs are independent of data contained in the object. The internal data values are not used to generate identification.
* OIDs are generated by the object system. Users or programs have no control over identification.
* OIDs last the lifetime of the object. The OID never changes even when the data contents may change.

1. **Discuss the various type constructors. How are they used to create complex object structures?**

Complex objects within the database must be constructed of other objects found within the database. Type constructors are the mechanism for this purpose. The simplest constructors are base or atomic, tuple, and set. For instance, if we view an object as a three-term definition, we could have the object ID as the first term, the second term would be the constructor type, and the third and last would be the value we are establishing for it. In illustration some of these would be:

object1 = OID1, set, {I1,I2,i3} object2 = OID2, atomic, 5

object3 = OID3, tuple, (DeptName, DeptNumber, DepMgr)

With these types of constructors, one can establish the new object, get its object ID, and give it a value.

1. **Discuss the concept of encapsulation, and tell how it is used to create abstract data types.**

Encapsulation gathers the data and methods of an object and puts them into a package, creating a well-defined boundary around the object.  Encapsulation is often referred to as *information hiding,*and encapsulation can be used to restrict which users and what operations can be performed against the data inside the object.

Classes provide encapsulation or information hiding by access control.  A class will grant or deny access to its objects using the *public* and *private*access specifies.  Public members define an interface between a class and the users of that class.  Public members can be accessed by any function in a program.  Objects can contain both public and private variables, the *public* variables are used with the objects methods or interfaces.

P*rivate* variables are only known to the object, and cannot be accessed by an interface.  For example a private method might be used to compute an internal value.

1. **Explain what the following terms mean in object-oriented database terminology: method, signature, message, collection, extent.**

* Method
* Signature
* Message
* Collection
* Extent

1. **What is the relationship between a type and its subtype in a type hierarchy?**

Types in Theta are grouped into a type hierarchy. Each type can have several supertypes. At the top of the hierarchy is type "any", which has no methods and is the supertype of all types. The type hierarchy is based on the notion of type equality. A type is always a subtype and also a supertype of itself. The subtype (supertype) relation is transitive: if T is a subtype of S and S is a subtype of R, then T is a subtype of R.

1. **What is the constraint that is enforced on extents corresponding to types in the type hierarchy?**

All persistent objects of a particular type can be stored in an extent. Extents corresponding to a type hierarchy have set/subset constraints enforced on their collections of persistent objects.

1. **What is the difference between persistent and transient objects? How persistence handled in typical OO database systems?**

The data manipulated by an object-oriented database can be either transient or persistent. Transient data is only valid inside a program or transaction; it is lost once the program or transaction terminates. Persistent data is stored outside of a transaction context, and so survives transaction updates. Usually the term persistent data is used to indicate the databases that are shared, accessed and updated across transactions. Persistence and class extensions are orthogonal in the sense that one can have an object-oriented system that has class extensions, but uses different mechanism for persistence. However, the object-oriented database languages that support class extensions usually make the extensions persistent.

1. **How do regular inheritance, multiple inheritance, and selective inheritance differ?**

Regular inheritance- a specialized version of the more generic concept that the base class represents). For example, in graphical user interface programming it's a very common pattern to have all visual elements being of type "Widget".

Multiple inheritance- Multiple inheritance is a feature of some object-oriented computer programming languages in which an object or class can inherit characteristics and features from more than one parent object or parent class.

Selective inheritance- occurs when sub type inherits only some of the functions of super type, here except clause is used to specify the functions that are not to be inherited by sub type. This inheritance is not provided by OO database systems.10. Discuss the concept of polymorphism/operator overloading.

1. **Discuss the concept of polymorphism/operator overloading.**

Refers to ability of operation to apply to different types of object in such situation operation name is same but implementation depends on type of objects it is applied. It is also called as operator overloading.

1. **Discuss how each of the following features is realized in SQL 2008: object identifier, type inheritance, encapsulation of operations, and complex object structures.**

An object identifier is used to give the name to the object which is unique and independent of the attribute values.

Type Inheritance: allows the inheritance of the attributes and methods in a database

Encapsulation of Operations: invokes the operations without touching or distributing the external programs. It allows modifying the internal structure of the program.

Complex object structure: can be constructed with the tuples, set, list, and array which are set of basic constructors. The performance of the database will be improved.

1. **In the traditional relational model, creating a table defined both the table type (schema or attributes) and the table itself (extension or set of current tuples). How can these two concepts be separated in SQL 2008?**

Both the table type and the table itself is defined in creation of a table in the relational database. In relational database management system, there are some features which are recognized by vendors.

The newer versions of relational system and the object database are proposed to use these features. The database systems are characterized as object relational DBMS using this features.

A recent version of the SQL standard 2008 for relational database management systems is also known as SQL/ foundation .this includes many features know as SQL/object.

1. **Describe the rules of inheritance in SQL 2008.**

Inheritance of all attributes of the tables. The inheritance hierarchy is determined by the order of the super types in the under clause. The run time of the parameters is always considered with the dynamic linking of the parameters.

1. **What are the differences and similarities between objects and literals in the ODMG object model?**

Objects and literals are basic blocks of the object model. The difference is that objects has both identifier and state, literal has no object identifier. Object state can change overtime by modifying object value, literal is basically a constant value that does not change.

Objects are identified by their OIDs where literals are identified by their value.

Objects will have a lifetime it depends on whether persistent object or transient object lifetime is not applical to literal.

1. **List the basic operations of the following built-in interfaces of the**

Basic operations copy: rebates a new copy of the object, Delete: deletes the objects, same as compares the objects identity to another object. Cardinality: returns the number of elements in the collections, set: Create Union: returns a new object of typeset union of two sets

1. **ODMG object model: Object, Collection, Iterator, Set, List, Bag, Array, and Dictionary.**

Cardinality: returns the number of elements in the collections,

1. **Describe the built-in structured literals of the ODMG object model and the operations of each.**

Create Union: returns a new object of typeset union of two sets

1. **What are the differences and similarities of attribute and relationship properties of a user-defined (atomic) class?**

Operation:

Date

Unsigned short year ();

Unsigned short month ();

Unsigned short day ();

Boolean Is\_ equal (in date other \_date);

Boolean Is\_ greater (in date other \_date);

Time

Unsigned hour ();

Unsigned minute ();

Unsigned second ();

Unsigned millisecond ();

Boolean Is\_ greater (in date other \_time);

Boolean Is\_ equal (in date other \_time);

Time add\_interval (in interval some interval);

Interval subtract (in the other\_time);

Time stamp

Unsigned short year ();

Unsigned short month ();

Unsigned short day ();

Unsigned short hour ();

Unsigned short minute ();

Unsigned short second ();

Unsigned short millisecond ();

Timestamp plus (in interval some\_interval);

Timestamp minus (in interval some\_interval );

Boolean Is\_ greater (in interval some \_interval);

Boolean Is\_ equal (in timestamp other \_timestamp);

Interval

Unsigned short day ();

Unsigned short hour ();

Unsigned short minute ();

1. **What the differences and similarities of class inheritance are via extends and interface inheritance via “:” in the ODMG object model?**

User defined (atomic) class? A user defined atomic object type is defined as a class by specifying its properties and operations. The prooperties define state of the object and are further distinguished into attributes and relationships.

Attribute: a property that defines some aspect of an object. Attributes have values that are stored within the object. However attribute values can be OIDs of other objects. Attribute values can even be specified via methods that are used to calculate that attribute values.

Relationship: is a property that specifies that two objects in the database as related.

Relationships are generally bidirectional.

1. **Why are the concepts of extents and keys important in database applications?**

Interface is a specification of the abstract behavior of an object type, which specifies the operation signatures. Although an interface may have state properties as part of specifications. These cannot be inherited from the interface. An interface is not insatiable. They are mainly used to specify abstract operations that can be inherited by classes or by other interfaces. This is called behavior inheritance.

Extends another inheritance relationship, called Extends and specified by the extends keyword is used to inherit both state and behavior strictly among classes. In an extends Inheritance, both the super type and the subtype must be classes. Multiple inheritance is not permitted via extends

1. **Describe the following OQL concepts: database entry points, path expressions, iterator variables, named queries (views), aggregate functions, grouping, and quantifiers.**

IN ODMG object model, the database designers can declare an extent for any object type that is defined via a class declaration. The extent is given a name, and it will contain all persistent object of class. Hence, the extent behaves as a set of object that holds all persistent objects of class.

1. **What is meant by the type orthogonality of OQL?**

Path expressions are a mechanism for expressing permitted sequences of execution. For example, a path expression like “{read}, write " might specify that either multiple simultaneous executions of read or a single execution of write but not both are allowed at any point in time.

The iterator (loop) variable is the variable which stores a portion of the iterable when the for loop is being executed. Each time the loop iterates, the value of the iterator variable will change to a different portion of the iterable.

A named query is a SQL expression represented as a table. In a named query, you can specify an SQL expression to select rows and columns returned from one or more tables in one or more data sources. A named query is like any other table in a data source view (DSV) with rows and relationships, except that the named query is based on an expression.

An aggregate function is a mathematical computation involving a set of values that results in a single value expressing the significance of the data it is computed from. Aggregate functions are often used in databases, spreadsheets and many other data manipulation software packages now common in the workplace.

In SQL, data grouping is performed using a GROUP BY clause. The SQL GROUP BY clause allows us to group individual data based on defined criteria. You can group individual data by one or more table columns.

1. **What are the main differences between designing a relational database and an object database?**

Type orthogonality of OQL means attributes, relationship and operation name scan is used interchangeably within a path expression, as long as the type system of OQL is not compromised.

1. **Describe the steps of the algorithm for object database design by EER-to OO mapping.**

In relational database managing system data is transferred in a relational way. This means that each access control table that stores data has a key field that identifies a row. In both network and hierarchical database accessing information is performed differently. Relational database connects data tables with rows to transfer information while In object oriented database management we have an entirely different approach where the information is represented in objects. If you are familiar with object-oriented programming, you’ll recognize the pattern. The main difference between object oriented database management system and a relation model is their approach on a digital transcript of information and the programming language. When data is stored in an object-oriented database system, it is in the form of an object. Each object consists of two elements where one is a piece of data (sound, text, video, etc.) and another is instruction for software. The instruction determines how the information will be transferred to another data file and the piece of data regulated where a specific type of this information will be heading to. In this complex system of managing data, it’s not enough to simply know a specific language but to understand commands as well.

1. **Describe the steps of the algorithm for object database design by EER-to OO mapping.**

Create on CDL class for each EER entity type or sub class.

Add relationships properties or reference attributes for each binary relationship into the CDL classes participating in the relationship.

Add appropriate operations for each class.

Specify inheritance relationships via extends clause.

Map weak entity types in the same way as regular entities.