Southern Leyte State University – Bontoc

San Ramon, Bontoc, Southern Leyte

**IT 301: Advanced Database Systems**

**Name: JEMALINE T. TEMARIO Year & Section: 3-B**

**Instructor: MR. REXAL TOLEDO Date: 12-14-2020**

***MODULE 2:***

***POST-TEST***

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

* Object-Oriented databases try to maintain a direct correspondence between real world and database objects so that objects don’t lose their integrity and can be easily be identified and operated on.

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

* OID should be immutable, and can be used by only one object.
* An Object-Oriented database system provides a unique identity that is system generated.
* Object identifier value is not visible to external users and the value of an object can’t be changed.

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

* Type constructors:

In Object-Oriented databases, the state of a object may be constructed from other objects by using certain type constructors. Basic constructors are atom, tuple, and set. Other commonly use constructors include list, bag and array.

**Atom** – it is used to represent all basic atomic values such as integers, real number, character string, Booleans and other basic data types that system supports directly.

**Tuple** – it is of the form where is an attribute name and each is an OID.

**Set** – set of object identifiers which are OID’s for a set of objects that are typically of same type.

**List** – list is similar to set but OID’s in a list are ordered. List can have arbitrary number.

**Array** – the object is a single dimensional array of object identifiers, number of elements is the maximum size of array.

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

* Both the object structure and the operations can be applied to objects that applied objects are included in the object class definition.
* In traditional database systems, information about complex objects is scattered about many relations or record. The internal structure of an any object in OOPLS includes the specification of instance variables. Which hold the values that define the internal state of the object. The instance variable may be encapsulated within the object and not necessarily visible to external users.

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

* **Method** in object oriented database defines that which type of function has been used like procedures, or functions.
* **Signature** in object oriented database is used for searching anything from database in fast and flexible way.
* **Messages** are the information used to request from other object to execute methods.
* **Collection** is the set of database in object oriented database.
* **Extent** defines the limit for any query and it means it is used to set the protocol in object oriented database.

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

* The relationship between type and subtype is that type is about the process means it will tell that which type of transaction is going to take place and subtype means a more explanative operation and behavior which is going to take place during operation are being committed.

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

* The constraint is that every object in an extent that corresponds to a sub type must also be a member of the extent corresponds to its super type in type hierarchy.

1. What is the difference between persistent and transient objects? How persistence handled in typical Object-Oriented database systems?

* **Persistent Objects** have representation in databases and identifiers. These objects are saved in the databases. Persistent objects are synchronized with that of the database. **Transient Objects** are that has been just instantiated. Transient objects don’t have representation and identifier value in the databases. The **transient objects** will be destroyed by the garbage collector if they don’t have any use and reference in the

database. The persistence in Object Oriented Databases is handled by creating Persistent Object and Transient Objects. As stated the persistent objects have representations in the databases.

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

* **Regular inheritance** – in a type hierarchy occurs when a new type or class inherit much of their structure and or operations from previously defined types or classes.
* **Multiple inheritance** – is a type of hierarchy occurs when a certain type T is a subtype of two types and hence inherits the functions of both super types.
* **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.

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

* Refers to the 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 call 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 inheritance of the attributes and methods in 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 database are proposed to use these features. The database system are characterized as object relational DBMS using these features. A recent version of the SQL standard 2008 for relational database management system is also known as SQL/Foundation. This includes many features known as SQL/O.

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

* Inheritance of all attributes of the table. 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?**

* The difference between object and literals of ODMG object model is, it can be atomic or it can be structured means. The similarities between objects and literal of ODMG object model is both are represented by id if any object is to be represented it will be represented by OIDs and if a literal is to be represented it will be represented by LID’s.

1. List the basic operations of the following built-in interfaces of the
2. ODMG object model: Object, Collection, Iterator, Set, List, Bag, Array, and
3. Dictionary

* **Object** - object in ODMG model can be fixed or variable too based on requirement. It shows the behavior of the model and program. Object can be represented by OId’s.
* **Collection** – it is the set of the data in model and all the data has storage in different locations in system.
* **Iterator** – iterator is used to show and make changes in collection.
* **Set** – ODMG model is has predefined set of input which is used to perform any operations.
* **List** – list in ODMG model helps in performing any operation link wise.
* **Bag** – bag has different set of values and in ODMG model it has main use to set behavior and inheritance property.
* **Array** – there is a function in which set of inputs are performed in one input.
* **Dictionary** – ODMG model has some fixed input and whose meaning can be looked up in dictionary.

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

* Structured literals correspond to values that are constructed using tuple constructer. They include data, interval time and timestamp.

Operations:-

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 \_ equal ( in time other time);

Boolean is \_ greater ( in time other time);

Time add \_ interval (in interval some interval );

Interval subtract ( in time 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 \_ equal ( in interval some \_ interval );

Boolean is \_ greater ( in timestamp other \_ timestamp);

Interval:

Unsigned short day ( ) ;

Unsigned short hour ( ) ;

Unsigned short minute ( ) ;

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

* A user defined atomic object type as a class by specifying its properties and operations. The properties 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. **What the differences and similarities of class inheritance are via extends and interface inheritance via “:” in the ODMG object model?**

* An 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 also non-instantiable- that is, one cannot create object that correspond to interface definition. Because instances are non-instantiable they are mainly used to specify abstract operations that can be inherited by classes or by other interfaces. This is called behavior inheritance and is specified by colon (:) notation.

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

* In ODMG object model, the database designers can declare an extent for any object type that is defied 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 object of class.

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

* **Database entry points** – an entry point to the database is needed for each query, which can be any named persistent object. For many queries, the entry point is the name of the extent of a class.
* **Path expressions** – the concept of path expression can be used to specify a path to related attributes and objects. A path expression typically starts at a persistent objects name, or at the iterator variable that ranges over individual objects in a collection.
* **Iterator variables** – the iterator (loop) variable is the variable which stores a portion of the iterable when 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.
* **Named queries (views)** – named query is a virtual view on your data source that allows you to change the data you are mining in SQL Server without making changes to your original data. As a named calculation is a virtual column on a DSV table, a named query is nothing but a virtual view on your data source.
* **Aggregate functions** – aggregate function or aggregation function is a function where the values of multiple rows are grouped together to form a single summary value. Common aggregate functions include: Average (i.e., arithmetic mean) Count.
* **Grouping** – used with SQL functions to group the result from one or more tables.
* **Quantifiers** – claims that statements within its scope are true for every value of the unique variable.

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

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

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

* When compared to a **relational database** management system, an **object**-oriented **database** stores complex data and relationships **between** data directly, without mapping to **relational** rows and columns whereas a **relational database** stores information in tables with rows and columns.

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

* Steps for mapping EER to OO:
* Create on ODL class for each EER entity or sub class
* Multi-valued attributes are declared by sets, bags or list constructors.
* Composite attributes are mapped into tuple constructors.
* Add relationship properties for each binary relationship
* Relationship cardinality: single – valued for 1:1 and N : 1 directions ; set – valued for 1 : N and M : N directions.
* Relationship attributes create via tuple constructors
* Include appropriate operations for each class
* Operations are not available from the EER schemes; original requirements must be reviewed
* Corresponding constructor and destructor operations must also be added
* ODL class that corresponds to a subclass in the EER schema
* Inherits type and methods of its superclass in ODL schema
* Weak entity types
* Mapped same as regular entity types
* Weak entities that do not participate in any relationships may alternatively be presented as composite multi-value attribute of the owner entity type.
* Categories (union types)
* Difficult to map to ODL
* An n-ary relationship with degree n > 2
* Map into a separate class, with appropriate references to each participating class