CSC3 Capstone Project: ViKER

Team Members: DPLJER001, STNGAB001, GRMSTJ001

Test Plan

Since the implementation of ER to ARM is something that, to our knowledge, has not been done before we have chosen test cases from text book examples of ER to RM, and we will alter the text book output to include the additional features in the ARM. The test cases will all be text based for now, as we will focus on building the back end first and the from end at a later time. We do expect to have some version of the front end ready for the prototype demonstration to the client, but we will not develop a specific test plan for it yet.

| Test Case | Input Description | Output (Expected) Description |
| --- | --- | --- |
| 1 | An ER represented in XML:    Strong entity 1:   * 1 identifier attribute * 3 regular attributes | A textual (probably XML) representation of the equivalent ARM:  Table (relation) 1:   * Table name: same as the entity * Attributes: *self* OID attribute (indicating an object), and the 4 remaining attributes with datatype *anyType* * Primary key: *self* * *pathfd:* argument composed of the identifier attribute, which will point towards the *self* OID in memory. * Constraints: none * Disjoint constraints: none |
| 2 | An ER represented in XML:  Strong entity 1:   * 1 identifier attribute * 2 regular attributes * 1 composite attribute (e.g. Address = {Street number, Street name, Suburb, City, Country, Postal Code}. | A textual (probably XML) representation of the equivalent ARM:  Table (relation) 1:   * Table name: same as the entity * Attributes: *self* OID attribute (indicating an object), 3 regular attributes with datatype *anyType, as well as* the subcomponents of the composite attribute each as individual attributes with datatype *anyType.* * Primary key: *self* * *pathfd:* argument composed of the identifier attribute, which will point towards the *self* OID in memory. * Constraints: none * Disjoint constraints: none |
| 3 | An ER representing in XML of:  A strong entity:   * 1 identifier attribute * 1 regular attribute.   A weak entity:   * a composite identifier composed of 3 attributes * 2 regular attributes.   The Strong entity to weak entity relationship will be a one-to-many | A textual (probably XML) representation of the equivalent ARM made up of two table declarations:  Table 1: Strong Entity   * Table name same as entity name * Attributes: self OID and the 2 attributes with datatype anyType. * Primary key declaration: self * Pathfd: the argument will be made up of the identifier attribute of the strong entity * Constraints (none) * Disjointness constraint: disjoint with table (relation) of weak entity   Table 2: Weak (dependant entity)   * Table name: same as weak entity name * Attributes: self OID, and all 5 remaining attributes with datatype anyType * Primary key: self * Pathfd: this argument will be made up of the composite unique identifier of the weak entity **as well as** the unique identifier of the strong entity upon which the weak entity depends. * Constraint: a single foreign key which is the identifier attribute of the strong entity, which is also the argument for the pathfd of the strong entity which points to the self OID belonging to the strong entity relation (table) in memory. |
| 4 | An ER representing in XML of:  Strong entity 1:   * 1 identifier attribute * 3 regular attributes   Strong Entity 2:   * 1 identifier attribute * 1 regular attribute   A one (and only one) to zero-to-many relationship from Strong entity 1 to Strong entity 2. | A textual (probably XML) representation of the equivalent ARM made up of two table declarations:  Table 1: Strong Entity 1   * Table name same as entity name * Attributes: self OID and the 4 attributes with datatype anyType. * Primary key declaration: self * Pathfd: the argument will be made up of the identifier attribute * Constraints (none) * Disjointness constraint: disjoint with table (relation) of Strong entity 2   Table 2: Strong Entity 2   * Table name same as entity name * Attributes: self OID and the 2 attributes with datatype anyType. * Primary key declaration: self * Pathfd: the argument will be made up of the identifier attribute * Constraints: the identifier attribute of String entity 1 is a foreign key * Disjointness constraint: disjoint with table (relation) of Strong entity 1 |
| 5 | An ER representing in XML of:  Strong entity 1:   * 1 identifier attribute * 3 regular attributes   Strong Entity 2:   * 1 identifier attribute * 1 regular attribute   A zero-to-many to zero-to-many relationship from Strong entity 1 to Strong entity 2:   * This relationship has its own entity | A textual (probably XML) representation of the equivalent ARM made up of two table declarations:  Table 1: Strong Entity 1   * Table name same as entity name * Attributes: self OID and the 4 attributes with datatype anyType. * Primary key declaration: self * Pathfd: the argument will be made up of the identifier attribute * Constraints (none) * Disjointness constraint: disjoint with table (relation) of Strong entity 2   Table 2: Strong Entity 2   * Table name same as entity name * Attributes: self OID and the 2 attributes with datatype anyType. * Primary key declaration: self * Pathfd: the argument will be made up of the identifier attribute * Constraints: the identifier attribute of String entity 1 is a foreign key * Disjointness constraint: disjoint with table (relation) of Strong entity 1   Table 3: An intersection relation (table)   * Table name: intersection relation 1 * Attributes: self OID and 3 attributes with datatype anyType:   + Primary key from Table 1   + Primary Key from table 2   + Regular attribute describing the raltionship * Primary key declaration: self * Pathfd: the argument will be a composition of the two primary keys (from table 1 and table 2) * Constraints:   + Foreign key: primary key of table 1   + Foreign key: primary key of table 2 * Disjointness constraint: disjoint with table 1 |
|  |  |  |