UMBC

**IS 722: Systems & Information Integration**

**Project Deliverable 2: Metadata layer on top of databases**

**DESCRIPTION**

A major step in information integration is to provide access and retrieve information from different databases. You have already created a database. Let’s assume that your group has already different databases called **local DB1,** **local DB2** and **local DB3**. If your group has only one database, then that one is the **local DB1** and for the second one use the script that is provided to generate **local DB2**. Let us assume that **local DB1** has merged with **local DB2** and **local DB3.** The local DBs are actually a set of tables and records that you need to put together in a single database (the group database) which contains several tables:

* Tables that make up **local DB1**
* Tables that make up **local DB2**
* Tables that make up **local DB3**

The purpose of this homework is to create an **integration/metadata/ontology layer** on top of these databases, local DB1,2,3. Think of the integration layer as the metadata layer which contains information describing the 3 databases. Your job is to create this metadata layer.

***This is a group deliverable, you need to work with all the members of the group.***

**Creation of a metadata layer**. You need to create a metadata layer to be used for integration of information from the participating databases. This metadata layer is a table which contains information about the local database schemas (local DB1,2,3) which reside in your Oracle account. The information in the metadata layer contains the following:

1. **Canonical representation**. It refers to the name of an object (table name, field name) at the metadata layer. For example, the name “Client” in local DB1, and “Patron” in local DB2. However, these are “local names” representing customers at each local DB. At the metadata layer one may use the “Customer” name to identify either Clients or Patrons (this is just an example; there could be different field names in the actual databases). The canonical representation of an object is the “global” name of that object at the metadata layer. It can be used to describe the corresponding names of the same object at each local DB. For example, Customer (canonical name); Client (local DB1); Patron (local DB2); Patient (local DB3).
2. **Data Types and other semantic differences**. This is a description of the local data types used for each column name in each DB. For example, varchar2(20), number, date, etc. One may also store a function that converts data from the canonical representation to the local DB.
3. **Additional fields**. One may use additional information about the mappings or translations between canonical to local.

It is recommended that for each concept (entity/field/etc.) that is present in each local db, there must be three representations:

1. a **canonical representation** (global level) for the concept. This is a representation that one uses to identify the concept globally (see first column in the example table below).
2. a local representation for local DB1: it is the name of the column in local DB1 that represents that same concept. Also the data type of that column in local DB1 (see columns 2 and 3 in example table below)
3. a local representation for local DB2: it is the name of the column in local DB2 that represents that same concept. Also the data type of that column in local DB1 (see columns 4 and 5 in example table below)
4. a local representation for local DB3: it is the name of the column in local DB3 that represents that same concept. Also the data type of that column in local DB1 (similar to columns 4 and 5 in example table below)

See an example table below for a very simple representation of two concepts: a customer (represented in DB1 and DB2) and a product ID (also in both DBs). Note that all data entered in the table below are strings (varchar2):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Canonical Representation** | **Column name in local DB1** | **Data Type in local DB1** | **Column name in local DB2** | **Data Type in local DB2** |
| ‘Customer’ | ‘Client’ | ‘Varchar2(20)’ | ‘Patron’ | ‘Char(50)’ |
| ‘Product\_ID’ | ‘CD\_ID’ | ‘Varchar2(10)’ | ‘CD’ | ‘number’ |
| … | … | … | … | … |

One may need to add more fields to the metadata table representing the function(s) that map a canonical name to a local DB name, and/or to capture semantic and syntactic differences that are present between the canonical and the local names. You must not create views on top of the existing tables of local DB1 and local DB2. You need to create an actual table with metadata about DB1 tables and fields, and DB2 tables and fields. All the above tables are in your own account (basically, they are just tables in your Oracle account).

How to create a metadata/integration layer?  
Look at the schema of local DB1.   
Look at the schema of local DB2.

Look at the schema of local DB3.  
Identify semantically similar fields and for each field create a column in the metadata table:

1. Provide a name (canonical representation) of that field.
2. Identify the corresponding name (local DB1 name) of the same field in DB1
3. Identify the data type of that field in DB1
4. Identify a function to map the canonical name to the DB1 name (e.g. CONCAT if applicable)
5. Identify the corresponding name (local DB2 name) of the same field in DB2
6. Identify the data type of that field in DB2
7. Identify a function to map the canonical name to the DB2 name (if applicable)
8. Identify the corresponding name (local DB3 name) of the same field in DB3
9. Identify the data type of that field in DB3
10. Identify a function to map the canonical name to the DB3 name (if applicable)

A row with values of columns 1-10 is going to be inserted as a single record in the metadata table, representing a single field (e.g “Name”).

* Fields that are not present in a local database have NULL values in some of the columns in the metadata table.
* For aggregate fields type the function that puts them together (e.g. CONCAT, or SUM, etc.)
* Table names must also be inserted in the metadata table (just as you have added fields). The data type for tables should be "TABLE"

Let me reiterate that this assignment is about metadata. The final output of this assignment is to make sure that you have enough information in your metadata (integration) layer.

**DELIVERABLES**

**Create a file in MS Word or pdf and upload it on Blackboard. This file should contain the following. Creation of the integration (metadata) layer**

1. **[30 points]** E-R diagram of your metadata layer (this is very simple)
2. **[30 points]** Create table commands for the metadata layer
3. **[40 points]** Screen-shots with queries showing “select \* from …” for the table you created in the metadata layer.