Assignment 2

Qi Guo

1. Create new Exhibits

1a

1. I removed an attribute called exLocation from table Exhibitions

since ExhibitionsLocations table (created in point 3) has already covered the functionality of exLocation(exhibition-location).

1. I dropped old primary key in Exhibitions table called exName and make a new combined primary key with exName and exStartDate since there might be two exhibitions with the same name but on a different start date.
2. I added a new table called ExhibitionsLocations to keep track of all past and new exhibits in different locations. There might be different exhibitions in the same location but in different time period. So this table is used to make sure that all history data regarding locations and exhibitions are stored properly.
3. I added data to ExhibitionLocations table since it is a table that newly created.
4. I rebuilt the WorksLocations table as I have to add two new attributes: wlStartDate and wlEndDate; and set a combined primary key of wlIDNumber, wlIDLetter, wlLocation, wlStartDate. The reason why I didn’t choose wlEndDate as part of the primary key is because end date could be null.
5. I rebuilt a table called Doors (it was called DoorsConnections before to keep track of all doors associated with different locations. Since one door connects two locations, this table is used to make sure that all connections are recorded.
6. I removed the DoorConnections table since the newly added Doors cover the functionality for it.
7. I added start date and end date for ExhibitionsWorks table along with the date data coming from Exhibitions table.
8. I didn’t make changes in Works table because works table has IDLetter and IDNumber, which are enough to distinguish different works.
9. I didn’t make changes in Media table because it was set to refer to the primary keys of works table, which works fine as it links more than one media to each work.
10. Below is the e-r diagram for this part.



1b

1. Added three exhibitions to Exhibitions table with
   1. The fifth day after the end of the previous exhibit as the start date
   2. 4 months after the start date as the end date

2. Evolve the Database

2a

1. I revised the works table to include identification to distinguish three different ownerships for works: owned works or borrowed works and potential works.
2. I added 15 new works to your database and have them be long term borrowed works. Owners for these works are various cartoon character donors. Then I moved them to the storage since they have been borrowed and arrived to the museum.
3. I added 15 new works belonging to other museums as potential works that could be borrowed to the works table and set the owners as some names of museums. But I didn’t move them to one of the locations of the museum (no potential works in WorksLocations table) since they have not actually arrived at the museum.
4. I added ownership and owner for the works that belongs to the museum with ownership being ‘owned works’ and owner being ‘The National Museum of American History’, which is the data source of my database.
5. Thus, I have 100 works instead of 65 works by the end of 2a because I inserted 70 works to the database in Assignment1.
6. Below is a e-r diagram for this part.



2b

1. Build a new large gallery called galleryD with a suggested minimum of 15 works and suggested maximum of 25 works and I set the dimension to 250.
2. Added one door connection to Doors table between the new gallery galleryD and one of the small galleries (the one furthest from the storage facility), which is gallery B in this case.
3. Since the galleryD will be ready for use in two months, I added a boolean attribute, lcIsAvailable to the Locations table to keep track of whether the current location is available or not.
4. I set rules to the ExhibitionsLocations table for the galleryD that can't be used until 2 months later.
5. I added data for lcIsAvailable for all locations based on whether today is between their exhibitions start date and end date. lcIsAvailable is set to be False for galleryD for now.
6. I didn’t revise my data structure to accommodate using data about connecting rooms in scheduling multi-gallery exhibitions. Since I didn’t set any primary keys for this table, exhibitions with same name and location are allowed for this table.

2c

1. I created a view to store the query for lexemes frequency using ts\_stat function.
2. Two more views are based on the above query, one sorted by number of different items and the other one sorted in alphabetic order.
3. I set a new attribute called ‘work characteristic’ to Works table as there are no attributes that could cover most part of the works. And here are the sub-characteristics:
   1. ‘Print’ is the word that appears the most(40) but it is one of the subtypes for the works table, so it can't be chosen as a new attribute.
   2. 'Engraved' and 'woodblock' are two words that appear 32 times and electrotype appears 7 times. It makes sense to put them together to come up with a new attribute of 'printing technique'
   3. 'Photograph' appears 16 times so that it can a new attribute
   4. 'Watercolor' appears 11 times so a new attribute called 'painting method' can be added
   5. 'Bottle' and 'Glass' appear 10 times, 'Jar' appears 8 times so that a new attribute called 'container' can be added as another new attribute
   6. 'Woman' appears 3 times, 'ambassador' appears 2 times, 'bracero' appears 10 times, 'man' appears once, 'men' appears once, 'boy' appears once so they can be combined as 'people'
   7. 'Quilt' appears 6 times so a new attribute called 'ornamentation' can be added
   8. There are 9 works with undefined characteristic so I just let it as Null.
4. I added records for each item to include the selected value from above if description containing this value. I used 6 update queries to set the Characteristic attribute for 91 rows and other 9 rows are left Null.
5. I would suggest the following further changes:

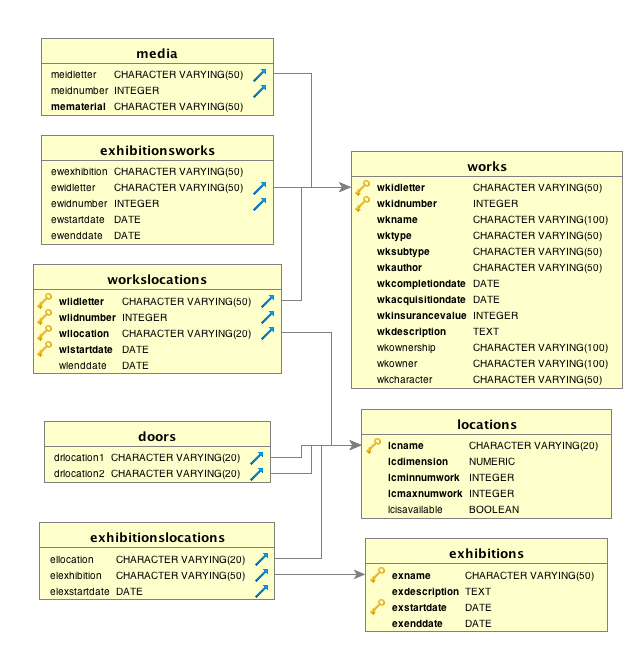
1. It would be efficient if we could filter the query from the description attribute of all the records in the database to identify all the different words that are found in it so that we can get rid of invalid data such as single letters or numbers, before making decision for new attributes.

2. There would be less redundant codes for the update table part if we could use a conditional expression like CASE.

3. If we set a trigger to find the lexemes frequency and update the works table accordingly when new works are added or old work descriptions are updated,

we don't need to worry about manually find new attribute or update works table from time to time.

1. Below is a e-r diagram by the end of this part.



3. Queries to Validate

1a

All tables are updated correctly as shown in the verification file.

1b

Three new exhibitions are inserted correctly with

1. The fifth day after the end of the previous exhibit as the start date for the new exhibition
2. 4/5/3 months after the start date is the end date for the new exhibition if works are in galleryA/ gallery B/ galleryC

2a

I end up with 100 works in database as I had 70 works rather than 65 works in Assignment 1.

15 works are borrowed, 15 works are potential and 70 works are owned, which are expected.

2b

galleryD is correctly added to Locations table with min of 15 works and max of 25 works and is not available now.

Two rules are proved to be working as the tests for inserting new data to WorksLocations table with galleryD as location name failed.

2c

3 queries regarding lexemes frequency are outputted correctly as shown in the verification file.

A new attribute ‘characteristic’ is added, and new data are added to Works table successfully. There are 9 rows with no data for characteristic, so it is left null.