CSCI 3901 Algorithm, Database ERD, Code Design, Data Structure

Objective : The course project is the opportunity to demonstrate all of the concepts from the course in one body of work.

Report by:

- Shivam Bhojani (B00895637) - shivam.bhojani@dal.ca

Algorithm:

PersonIdentity addPerson(String name)

1. Insert new entry in person with the new given name
2. Return true

Boolean recordAttributes(PersonIdentity person, Map<String, String> attributes)

1. Fetch PersonIdentity object
2. Select * from person where PersonIdentityperson.id = p_id
3. If null, return false
4. If the person exists, fetch the p_id.
5. for (Map.Entry<String,String> entry : Map.entrySet())
   a. Validation the key value with the columns in person table
   b. If key==column in person table, store the value in DB

Boolean recordReference(PersonIdentity person, String reference)

1. Select * from person where PersonIdentityperson.id = p_id
2. If null, return false
3. Insert into person_reference values (p_id, reference);
4. Return True

Boolean recordNote(PersonIdentity person, String note)

1. Select * from person where PersonIdentityperson.id = p_id
2. If null, return false
3. Insert into person_notes values (p_id, notes)
4. Return True;

Boolean recordChild(PersonIdentity parent, PersonIdentity child)

1. Check whether parent and child exist in person
2. If null, return false
3. Insert into parentchild_relation (parent_id, child_id)
4. Return false

Boolean recordPartnering(PersonIdentity partner1, PersonIdentity partner2)

1. Check whether partner1 and partner2 exist in person
2. If null, return false
3. Insert into partner_relation (partner 1, partner 2)
4. Return true

Boolean recordDissolution(PersonIdentity partner1, PersonIdentity partner2)

1. Check whether relation exist in partner_relation
2. If null, return false
3. Delete relation from partner_relation

FileIdentifier addMediaFile(String fileLocation)

1. Insert new entry in media archive with the new given name
2. Return true

Boolean recordMediaAttributes(FileIdentifier fileIdentifier, Map<String, String> attributes)

1. Fetch FileIdentifier object
2. Select * from media_archieve where FileIdentifier.id = p_id
3. If null, return false
4. If the file exists, fetch the mediaId.
5. for (Map.Entry<String,String> entry: Map.entrySet())
    a. Validation the key value with the columns in person table
    b. If key==column in file table, store the value in DB

Boolean peopleInMedia(FileIdentifier fileIdentifier, List<PersonIdentity> people)

1. Fetch the media_id from FileIdentifier object.
2. Find p_id from a person.
3. If media_id or person not found, then return false
4. Insert into people_in_media (mediaId, person_Id)
5. Return True

Boolean tagMedia(FileIdentifier, fileIdentifier, String tag)

1. Fetch the media_id from FileIdentifier object.
2. If null, return false
3. Insert into media_tags values (mediaId, tag);
4. Return True

PersonIdentity findPerson(String name)

1. Select query from person table with name in the argument
2. If null, return null;
3. Else, create PersonIdentity object
4. Assign column values from person table to each attributes of PersonIdentity object and return it.

FileIdentifier findMediaFile(String name)

1. Select query from media_archeive table with name in the argument
2. If null, return null;
3. Else, create fileIdentifier object
4. Assign column values from the media_archieve table to each attributes of file identifier object and return it.

BiologicalRelation findRelation(PersonIdentity person1, PersonIdentity person2)

1. Check if any of the person 1 or person2 has no ancestors
   a. If yes, then check whether person2's ancestor is person 1 or person1's ancestor is person2;
   b. If yes, find the generation gap in the same family line
2. Find common ancestors for person1 and person2

Set<PersonIdentity> descendents(PersonIdentity person, Integer generations)

1. Validate person in person table
2. If null, return false or null
3. If person found, create view of parentchild_relation where parentId=person_id
   a. Select from parentchild_relation view where parentId = personId
   b. Delete particular entry
4. Run step in for loop, for number of interactions mentioned in 'generation'
5. OR until select query data becomes null;

Set<PersonIdentity> ancestor(PersonIdentity person, Integer generations)

1. Validate person in person table
2. If null, return false or null
3. If person found, create view of parentchild_relation where childId=person_id
   a. Select from parentchild_relation view where childId = personId
   b. Delete particular entry

4. Run step in for loop, for number of interactions mentioned in 'generation'

5. OR until select query data becomes null;

List<String> notesAndReferences(PersonIdentity person)

1. Select p_id from person where name = person.name
2. Store it in a variable ID.
3. Select notes from person_notes where p_id = ID
4. Select references from person_reference where p_id = ID

Data Structure

1. Map
   a. Record attributes for person data
   b. Record attribute for media data

2. List
   a. List will be used in findrelation
   b. It will compare the ancestors of partner1 and partner2 find relations
   c. List will also be used fetch multi value attributes
      i. Example: Tags, notes, references and Individuals seen in picture.

Code Design

CRC method:

PersonIdentity Class	
Responsibilities: PersonIdentity addPerson(String name) Boolean recordAttributes(PersonIdentity person, Map<String, String> attributes) Boolean recordReference(PersonIdentity person, String reference) Boolean recordNote(PersonIdentity person, String note)	**Collaboration:** Genealogy BiologicalRelation

FileIdentifier	
Responsibilities: FileIdentifier addMediaFile(String fileLocation) Boolean recordMediaAttributes(FileIdentifier fileIdentifier, Map<String, String> attributes) Boolean peopleInMedia(FileIdentifier fileIdentifier, List<PersonIdentity> people) Boolean tagMedia(FileIdentifier, fileIdentifier, String tag)	**Collaboration:** Genealogy

Genealogy

Responsibilities:

PersonIdentity findPerson(String name)

FileIdentifier findMediaFile(String name)

String findName(PersonIdentity id)

String findMediaFile(FileIdentifier fileId)

Collaboration:

PersonIdentity

FileIdentifier

BiologicalRelation

Responsibilities:

Boolean recordChild

Boolean recordPartnering(PersonIdentity partner1, PersonIdentity partner2)

Boolean recordDissolution(PersonIdentity partner1, PersonIdentity partner2)

Collaboration:

PersonIdentity

DataBase Schema

ERD: