COURSE PROJECT

External documentation

Overview:

Genealogy is a class where we have a family relation between individuals and associated media files and tags and help answer question about how a person is related to another person and get more data on pictures and tags a person is present in the archive.

Genealogy encapsulates all the other classes File Identifier, Media Tag, PersonIdentity, contains all the interaction and logic for the reporting functions, helps answer all the answers with its reporting functions.

Brief about each class:

Geanology

Genealogy is the main class that hold everything to gather it contains both person identity, media tags and File identifier it builds relation and performs data operations through various functions and reports them back to the user as well.

It holds array list of People Identifier objects, Array list of all the media attributes and Array list of all Media tags related to media files, supporting functions and logic to connect and validate a family tree to find ancestors, descendants and answer question on relations, tags and media of particular people.

Person Identity

Person Identity is a class that represents a person in a family tree, it stores a person's relation, parents, children and attributes of a person like name, gender, occupation, birthdate, birth location, death date, death location.
Also stores media files a person belongs to and his relation.

File Identifier

File identifier is a class that represents a media file in the system it stores the database id of the file, filename/location attributes of the file like date, place, city, province, country, also stores array list of tags and array list of people in media.

Media Tag

Media tag consists of tag and its data base id and the files related to media, it has a many to many relationships with File Identifier.

Biological Relation

Biological Relation is a class that stores degree of cousin ship and degree of removal between 2 people.

Approach:

Genology is a class that encapsulates all the other classes, placed all the attributes related to a particular entity in its class and the logic of accessing and maintaining them is done in geonology.

All data is maintained in array list of objects inside genology and also inserted into the data base, as soon as the function loads up the constructor and initializes the database connection.

Any data or manipulation is maintained in the object in the family tree as well as the changes are made in the data base, the data is loaded into the data base as soon as the object is created and the data is loaded into the system.

All updating and insertions are maintained on a case-to-case basis and with majority being handled in java and rest fetching from the SQL whenever required.

Data structures and their relations to each other:

Data structures

1) Genealogy:

- Arraylist of <PersonIdentity>
- Arraylist of < FileIdentifier>
- ArrayList of < MediaTag>
  to maintain relation between all the other main entities with main class and handles and implements logics and returns, inserts, updates data base.

2) PersonIdentity:

Person identity represents a person instance.

- ArrayList of <String> for refences and notes
- ArrayList of < FileIdentifier> for media files associated with the person
- Set of < PersonIdentity> for children and media
  To store all the details belonging to a person and stores data base id to validate the incoming object more efficiently rather than querying the data base each time.

3) FileIdentifier:

Fileidentifier is a class that represents Media file, store all attributes.

- ArrayList<MediaTag> tags
- ArrayList<PersonIdentity> peopleInMedia
  Stores and maintains all the data and updates and maintains consistency with it, and also maintains relationship with tags and people belonging to media.

4) MediaTag:

Media Tag class represents a Tag, tag can belong to multiple media files.

- ArrayList<FileIdentifier> taggedFiles

A tag stores array list of tagged files to maintain relation between tag and its associated files.

Key algorithms and design elements:

1) Finding ancestors and descendants:

The data structure tree has connections to both parents and children for each person traversing was possible, solved by getting recursive function.

Step1: Start with the person assign it with current generation 0.

Step2: traverse with set of parents or children based on if ancestors or descendants.

Step3: Recursively set the person with next level either parents or children based on if its ancestors or descendants.

Step4: If the generation is reached add to the set of descendants or ancestors set and return the it back to the calling function

2) Finding Degree of Removal and Degree of Cousinship

Setp1: Finding the common ancestor reclusively similar to ancestor function approach above.

Step2: Finding the offset from the common ancestor

Step3: Calculate the degree of removal and degree of cousinship using the formula, (min {nX, nY} -1) cousins |nX – nY| removed).

DATABASE DESIGN:

Below are the table details and date base connection details.

Tables: File name: ***Table creation.sql***

Contains all the SQL commands for table creation.

1) mediaattribute: Stores details of Media attributes

2) media file: Stores media file location

3) mediapeople: Media people stores relation between person and media

4) mediatag: Media tag stores tag name**.**

5) personattributes: Stores attributes of all the media file.

6) personchild: Stores person and child mapping

7) personidentity: Stores person identity of a person.

8) personnotereference: stores notes and reference of a person.

9) personrelationship: Relationship between 2 people.

DataBase connection:

To connect to data base changes the Properties file attributes in file

File name: ***Config_database_Connection.properties.***

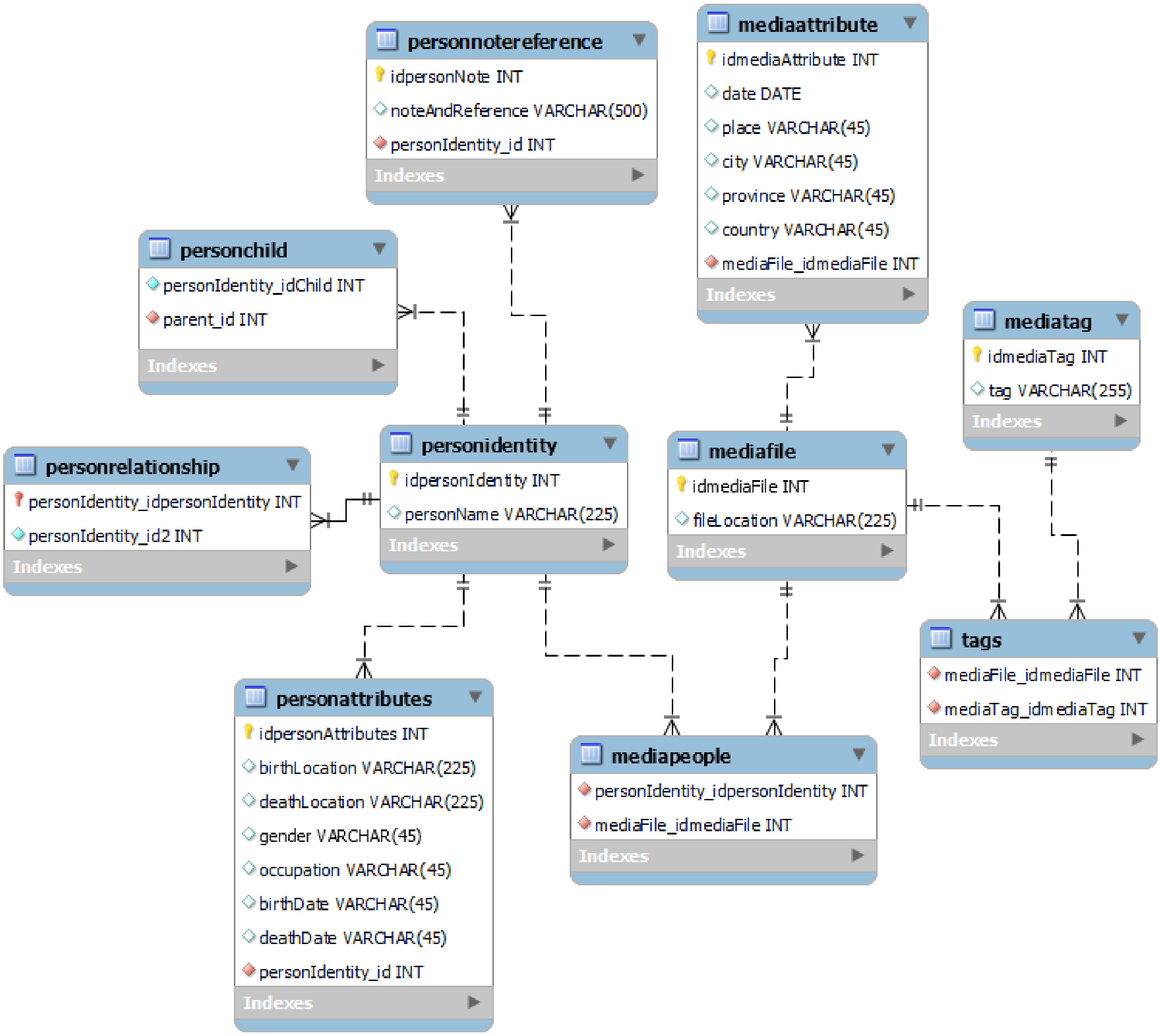
connectionURL: jdbc: mysql://127.0.0.1:3306/mydb

user: root

password: root

Database ER Diagram:

Over view of ER diagram and relation between tables



TestCases:

1) Adding a person.

- Add 1 person into system
- Add 3 people into the system
- Add n people when there are many existing in the system
- Add people with same name
- Add a person with no name

2) Recording details of the person.

- Add details of a person that does not exists
- Add details of a person which is same as details of another person
- Details that are not practically possible ex: DOB 31/2/2023
- Details of a person with name containing single letter
- Details of a person with no occupation
- Details of a person with no gender
- Detail of a person with no DOB
- Details of a person with empty details for all the attributes

3) Record References

- Add references to a person that does not exists in the system
- Add reference to a person whose who already have references
- Add multiple references to a single person
- Add single word references
- Add single letter references
- Add website link references
- Add location references

4) Record Notes

- Add notes to a person that does not exists in the system
- Add notes to a person whose who already have multiple notes
- Add multiple notes belonging to the same person
- Add single word notes
- Add single letter notes

5) Record Parent child relations

- Record parent child relation between 2 individuals
- Record when parent has already 1 child relation defined
- Record when child has already 1 parent relation defined
- Record when parent does not exist in the system
- Record when child does not exist in the system
- Record when the relation is already defined

6) Record Partnering relation

- Record partner relation between 2 individuals
- Record when partner has already 1 relation defined
- Record when both partners does not exist in the system
- Record when one partner exits in the system
- Record when the relation is already defined

7) Record dissolution

- Record dissolution relation between 2 people
- Record dissolution when there was no predefined partnering relation
- Record dissolution when one partner does not exist
- Record dissolution when both partners does not exist
- Record when there has already been dissolution that exits

8) Add Media files

- Add Media file to the location which has 2 letters
- File location when there are no previous files
- File location when there are previous files that exits

9) Record media details

- Record when media does not exist
- Record when media exits

- Record when media already contains details
- Record when details are empty
- Record when empty attributes are passed

10) Record people in media details

- Record media details for one person in the list
- Record media details for many people in the list
- Try to record when the file is not present
- Try to record details when people are not in the list
- Try to record when there already exits details.
- Try to record details when their people in the list have details associated to them.

11) Record Tag on Media

- Record Tags for media files in the system
- Try recording Tags when media files don't exist
- Try recording when media files exist and not tags
- Try recording when media files exits and multiple tags exists.

12) Reporting person back

- Find person based on name
- Try to find person whose name does not exits
- Try to find a person whose name has just one letter

13) Reporting files back

- Find files based on names
- Finding a media file when no name is passed
- When files don't exist in the system.
- When file name doesn't exist in the system.
- Reporting people name based on personal identity

14) Reporting relations.

- Try reporting people name when no person identity exists.
- Try reporting files when fide identifier is passed
- Reporting relation between people with biological relation
- Reporting relation between people with no direct relation between people
- Reporting all decedents of a person
- Reporting all decedents of a person up to a particular generation
- Reporting all decedents of a person when there are no descendants
- Reporting all ancestors of a person
- Reporting all ancestors of a person up to a creatin level o Return the set of media files based on location, start date, end date.
- Return media files when only location is passed
- Return media files when start date is passed
- Return media files when end date is passed
- Find media files belonging to an individual based on start date end date
- Finding media files within some ranges of dates
- Media files with in some range of dates when no files exist
- Media files when both date ranges are same date
- Try fetching media files when no file exits between a range
- Try fetching files when dates are not valid
- Try finding media when the person does not exist
- Try finding media of a person's biological children
- Try to find media of children of a person when a person has no children.

15) Reporting tags and media

- Return the set of media files based on location, start date, end date.
- Return media files when only location is passed
- Return media files when start date is passed
- Return media files when end date is passed
- Find media files belonging to an individual based on start date end date
- Finding media files within some ranges of dates
- Media files with in some range of dates when no files exist
- Media files when both date ranges are same date
- Try fetching media files when no file exits between a range

- Try fetching files when dates are not valid
- Try finding media when the person does not exist
- Try finding media of a person’s biological children
- Try to find media of children of a person when a person has no children.

PersonIdentity findPerson (String name)

- Pass the same name as that of a person that exists in the data base
- Add when there is no data in the data base
- Add when there is lots of people in the data base
- Try and add if there is no connection to the data base

FileIdentifier findMediaFile (String name)

- Add media when no file exists in the data base
- Add media when there are lots of data in the data base.
- Add when there is no connection in the data base.
- Add where there is a connection with the media file already present

String find Name (Person Identity id)

- When there is no person present in the data base
- When the person exists in the data base
- When a lot of people exits in the data base
- When null is passed

String findMediaFile (File Identifier field)

- When there is no file present in the data base
- When the file exists in the data base
- When a lot of files that exits in the data base
- When null is passed into the column.

BiologicalRelation findRelation(PersonIdentity person1, PesonIdentity person2)

- When there is no relation between people
- When there is relation between 2 people

- When either one of then does not exist
- When there is a dissolution before

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

- If there are no decedents to a person
- If there is more than their level descendant's
- If there are many levels of descendants.
- If 0 is passed as an integer
- If null is passed to the person

Set<PersonIdentity> ancestores(Person Identity person, Integer generations)

- If the person doesn't have any ancestors
- If the person has more than 3 levels of ancestors
- If the person is not present in the data base
- If the 0 is passed inside the integers

List<String> notesAndReferences(PersonIdentity person)

- Get notes and reference when a person doesn't have any
- When a person is not present in data base
- If a person is present but doesn't have any reference and only notes
- If a person had only reference.

Set<FileIdentifier> findMediaByTag(String tag , String startDate, String endDate)

- When tag does not exist in the data base
- When tag exists and contains no data
- When tag is associated with lot of media files

Set<FileIdentifier> findMediaByLocation(String location, String startDate, String endDate)

- − When location is not associated with any media files
- − When null is passed for dates
- − When there are n number of files that exists with a particular location

List<FileIdentifier> findIndividualsMedia(Set<PersonIdentity> people, String startDate, String endDate)

- − Where people do no exists in the data base
- − Where people don’t have any media files associated with them
- − When there are a lot of media files associated with them
- − When null is passed as start date and end date
- − When null is passed as end date and a start date is given

- • List<FileIdentifier> findBiologicalFamilyMedia(PersonIdentity person)
- − Where people don’t have any relations defined
- − When no media exists for them
- − When a person doesn’t exist in the data base
- − When a person exists and biological family media does not exist
- − When may media are associated with biological family

Design consideration:

1) Date has to be in YYYY, YYYY-MM, YYYY-MM-DD format.

2) Attributes for a person is limited to gender, occupation, birthdate, birthLocation, deathDate, deathLocation.

3) Attributes for a file is limited to date, place, city, province and country.

4) Null returned if no common ancestor is found

5) Null returned when media file is not present