CSCI 3901 - Project

Milestone 3

Name: Benny Daniel Tharigopala

Banner ID: B00899629

Data structures

Classes:

1. **PersonIdentity** – Manages data relevant to individuals in the system.

   Data Structures:

   i. Set<PersonIdentity> Children – To store the information of children for each parent.

   ii. Map<Integer, List<String>> Notes – To store notes on an individual

   iii. Map< Integer, List<String>> References – To store references on an individual

   iv. Map<String, String> Attributes – To store attributes of an individual

2. **FileIdentifier**– Manages data relevant to media files in a archive.

   Data Structures:

   i. Map< Integer,List<String>> Tags – To store Tags relevant to a media file.

   ii. Set<PersonIdentity> PeopleInMedia – To store the identifiers for people who appear in a media file.

   iii. Map<String, String> Attributes – To store attributes of a media file.

3. **BiologicalRelation** – Manages Data relevant to relationships between individuals in the system.

   Data Structures:

   i. Set<PersonIdentity> Descendants.

   ii. Set<PersonIdentity> Ancestors.

   iii. Map< List<Integer>, String> Relationships – To store the relationship between any two individuals.

4. **Genealogy** – Parent Class which encompasses all other classes.

   Data Structures:

   i. Set<PersonIdentity> People – To add a person to the family tree.

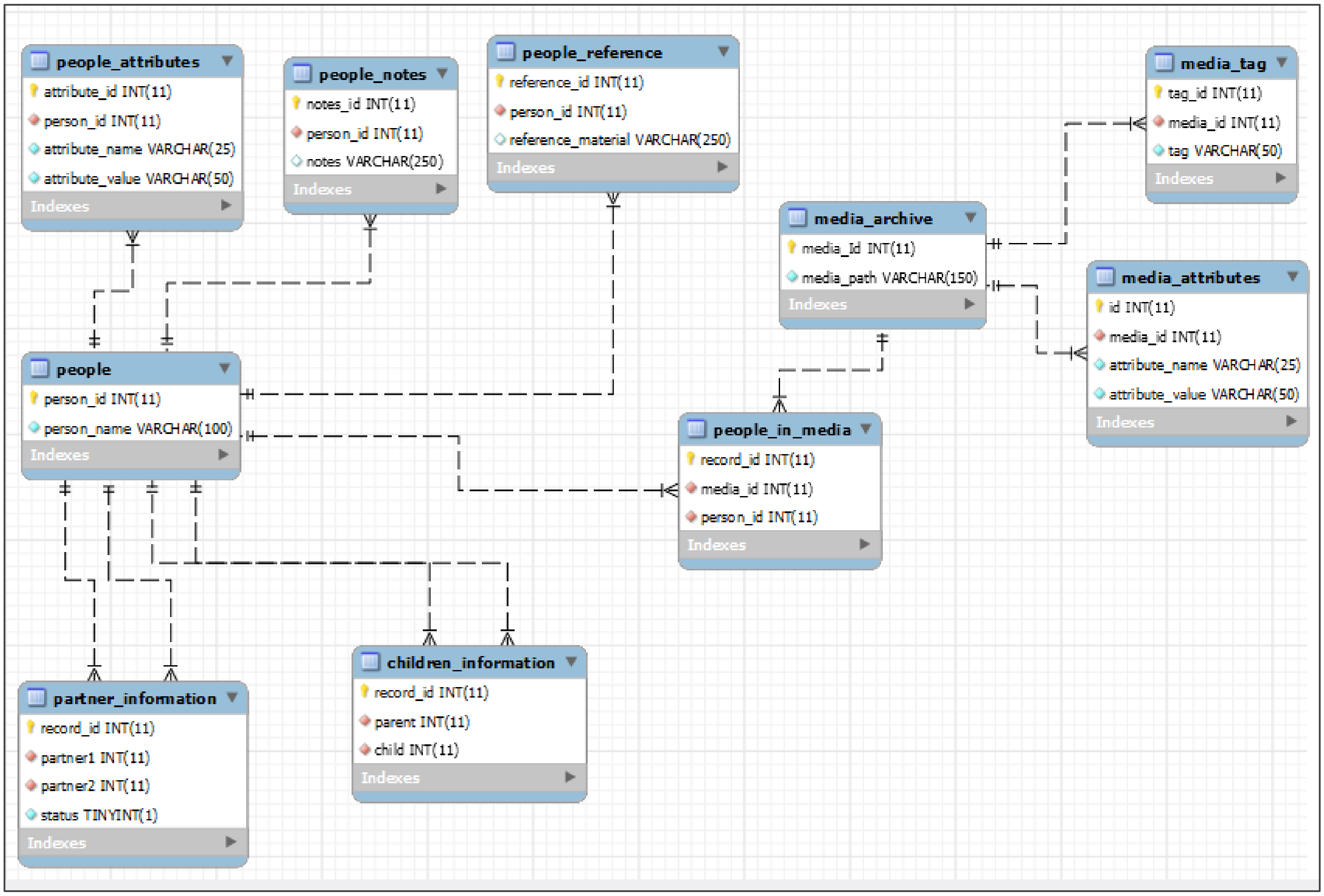
   ii. Set<FileIdentifier>Media – To add Media files to an archive.

   iii. Set<FileIdentifier> MediaByTag – To store Media files for a given tag or location.

   iv. List<FileIdentifier> MediaByPeople – To store Media files for a given set of people or a person's children.

Code Design

1. ER Diagram:



2. Store Information

A major part of code design is to obtain information from the user and store it in the Database.

The following information will be stored and managed in a Database:

1. Name of a person and a unique identifier for the person.
2. Attributes of a person.
3. Notes and source reference materials for a person.
4. Partnering information between 2 people.

5. Records which describe a biological "Parent-Child" relationship between two individuals.
6. Media file paths (absolute paths) a unique identifier for the files.
7. Attributes of a media file.
8. Tags for a media file.
9. Relationship between individuals and media files. That is, records which store the identifiers of individuals who appear in a media file.

3. Report Information

Once the information from users is stored. The next step is to manipulate the data and render details which satisfy requirements.

1. The relationship between two individuals can be determined through Algorithm 1 as shown in the next section.
2. The set of ancestors and descendants within a particular generation gap can be obtained by traversing the Family Tree.
3. Notes & references for individuals can be obtained by a "Select" query in the database.
4. Media files can be filtered and returned on the basis of "Tags", "Location" and Individuals who appear in the media file.

Key Algorithms

One of the key algorithms in the project determines the relationship between two individuals in the system. The relationship is expressed in terms of fields known as "**Cousinship**" & "**Degree of Removal**".

1. The algorithm to determine the relationship between any two individuals is as follows:
2. Obtain the identifiers for the two individuals Person A and Person B.
3. Determine the nearest common ancestor of Person A and Person B. Let this be Person Z.
4. Determine the generation gap between Person A and Person Z. Let this be gAZ.

5. Determine the generation gap between Person B and Person Z. Let this be gBZ.
6. Assign the value [Minimum(GAZ,GBZ)-1] as the Cousinship between the individuals.
7. Assign the value [GAZ-GBZ] as the Degree of Removal between the individuals.
8. Return the Cousinship and Degree of Removal values.

Figure 1 illustrates the concept of Cousinship and Degree of Removal in a typical Family Tree.

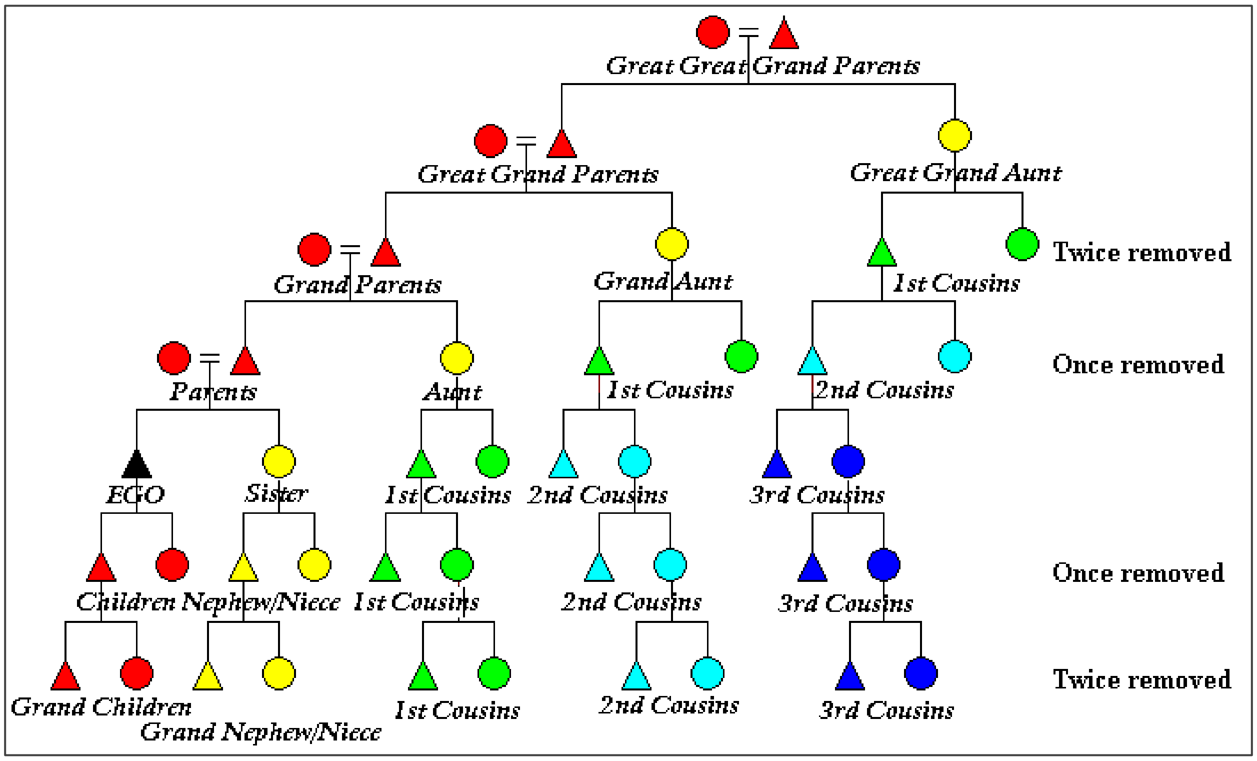


Figure 1: Degree of Cousinship and Removal [1]

Another key algorithm is one, which **retrieves a set of media files** relevant to a set of people or tags or location.

The algorithm is as follows:

1. Obtain the information required to filter the files in the Media Archive. This might be a set of identifiers for people or media files, or, one or more tags, or a location.
2. Retrieve the identifiers of files from the database by filtering files that are relevant to the objects obtained in Step 1. This can be achieved by passing the values in the 'Where" clause of a "Select" statement.
3. Return the identifiers.

To obtain the Media files relevant to the immediate children of a person:

1. Obtain the list of children for the person in context
2. Retrieve the identifiers of files from the database by filtering files that are relevant to the list of identifiers obtained in Step 1. This can be achieved by passing the values in the 'Where" clause of a "Select" statement.
3. Return the identifiers.

Additional White Box Tests

1. Add Persons to the Database:

- Add a person to the database when no other individuals exist in the database.
- Add a person to the database when 1 individual exists in the database.
- Add a person to the database when many individuals exist in the database.
- Add a person with an invalid name (Null / Empty String / Special characters).
- Add a person with a name that's identical to an existing name in the database.

2. Record Attributes of Persons:

- Record valid attributes of a person who exists in the database.
- Record valid attributes of a person who doesn't exist in the database.
- Update the attributes of a person who exists in the database.
- Update the attributes of a person who doesn't exist in the database.

- Record the birth date of a person where the date is greater than the current date.
- Record the death date of a person where the date is greater than the current date.
- Record attributes of a person without birth date, that is, all attributes **except** birth date are provided.
- Record attributes of a person without gender.
- Record attributes of a person without occupation.

3. Record a Reference Material for Persons:

- Record a material for a person who exists in the database.
- Record a material for a person who doesn't exist in the database.
- Record multiple materials for a person who exists in the database.

4. Record Notes for Persons:

- Record notes for a person who exists in the database.
- Record notes for a person who doesn't exist in the database.
- Record multiple notes for a person who exists in the database.

5. Record a Child for Persons:

- Record a valid (existent) child for a person who exists in the database.
- Record a child for a person who doesn't exist in the database.
- Record a relationship between the same child and parent multiple times.
- Record multiple children for a person who exists in the database.
- Record a child for a person who exists in the database when the child does not exist.

6. Record a Partnering Relation between Persons:

- Record a partnering relation between people who exist in the database.
- Record a partnering relation between two people who already have a symmetric partnering relation defined.
- Record a partnering relation between people when one person doesn't exist in the database.
- Record a partnering relation between people when both individuals do not exist in the database.
- Record a partnering relation between two people who have a symmetric dissolution relation defined **between them.**
- Record a partnering relation between two people who have a symmetric dissolution relation defined **with other people.**

7. Record a dissolution of a Partnering Relation between Persons:

- Record a dissolution between people who exist in the database and had a partnering relation defined between them.
- Record a dissolution between people who exist in the database but do not have a partnering relation defined between them.
- Record a dissolution between two people who already had a dissolution of relation defined between them.
- Record a dissolution between people when one person doesn't exist in the database.
- Record a dissolution between people when both individuals do not exist in the database.
- Record a dissolution between people who exist in the database and have a partnering relation defined **NOT** between them but with other Persons.

8. Add Media Files to an Archive:

- Add a media file to the archive.
- Add a media file to the archive when the location of the file already exists in the archive.
- Add a media file with an invalid file path.

9. Record Attributes of a Media file in the Archive:

- Record valid attributes of a media file which exists in the archive.
- Record valid attributes of a media file which doesn't exist in the archive.
- Update the existing attributes of a media file with new ones.
- Record date for a media file when the date is greater than the current date.

10. Record People in Media Files:

- Record People in a file when both the file and people exist.
- Record People in a file when any one of the file and people don't exist.
- Record People in a file when both the file and people don't exist.
- Record redundant information. That is, record that a set of people exist in a media file, multiple times.

11. Record tags for Media Files:

- Record tags for a file which exists in the archive.
- Record tags for a file which doesn't exist in the archive.
- Record empty tags for a file.
- Record multiple tags for a same media file.

12. Report:

- Locate an individual when the individual exists in the database.
- Locate an individual when the individual doesn't exist in the database.
- Location an individual when the individual's name is empty. That is, pass an empty string while attempting to locate a person.
- Report a person's name when the person exists.
- Report a person's name when the person does not exist.
- Report notes and source reference materials for a person who exists in the database.
- Report notes and source reference materials for a person who does not exist in the database.

- Report relations when two people exist.
- Report relations when any one person does not exist.
- Report relations when two people do not exist.
- Report relation between people who have direct ancestor descendant relationship.
- Report relation between people who do not have direct ancestor descendant relationship (Pibling and Nibling included).
- Report ancestors / descendants of a person within a given generation.
- Report the ancestors of a person when the information isn't available in the database.
- Report the descendants of a person when the person doesn't have any descendants.
- Report people who are linked a media file.
- Report people who are linked a media file but the file path doesn't exist in the archive.
- Locate a media file when the file exists in the archive.
- Locate a media file when the file doesn't exist in the archive.
- Location a media file when the file path is empty. That is, pass an empty string while attempting to locate a media file.
- Report files which fall within a given date range.
- Report media files when any one of start date or end date is not passed.
- Report media files when both start date or end date are not passed.
- Report files which are supposedly related to a given set of people who exist in the database.
- Report files which are supposedly related to a given set of people, but the people do not exist.
- Report media files linked to a given tag and timeframe.
- Report media files linked to a given tag and timeframe but the tag doesn't exist.
- Report media files linked to a given location and timeframe.
- Report media files linked to a given location and timeframe but the location doesn't exist.

- Report media files (in chronological order) linked to the immediate children of a person, within a timeframe.
- Report media files (in chronological order) linked to the immediate children of a person, within a timeframe when the person has no immediate children.
- Report media files (in chronological order) linked to the immediate children of a person, within a timeframe when there are no media files.

References

1. https://www.umanitoba.ca/faculties/arts/anthropology/tutor/descent/cognatic/collateral.html#top