Prachi Prakashbhai Raval
B00883324
pr647062@dal.ca

26th November ,2021

CSCI 3901 - FINAL PROJECT

MILETONE 3:
EXTERNAL DOCUMENTATION OF DATA STRUCTURES, CODE DESIGN, KEY ALGORITHMS, AND ANY ADDITIONAL WHITE BOX TESTS

Index

Prachi Prakashbhai Raval
B00883324
pr647062@dal.ca

Data Structure:

1. Family Tree Management and Media Archive management – For storing objects PersonIdentity and File identifier along with their id's, I have used map.

Map fetches the data already present in the Data base, according to the id, and then also stores new objects added.

Two maps used to store objects. Later for testing purposes objects from the map can be passed to other methods in the genealogy class.

```
•  fileIdentifierMap
•  personIdentityMap
```

2. Biological Relationship – For finding the relationship, common ancestor needs to found. For this I have used vector array. This can find the lowest common ancestor and the distance of the person from the ancestor (levels).

```
Vector<Integer>[] tree
```

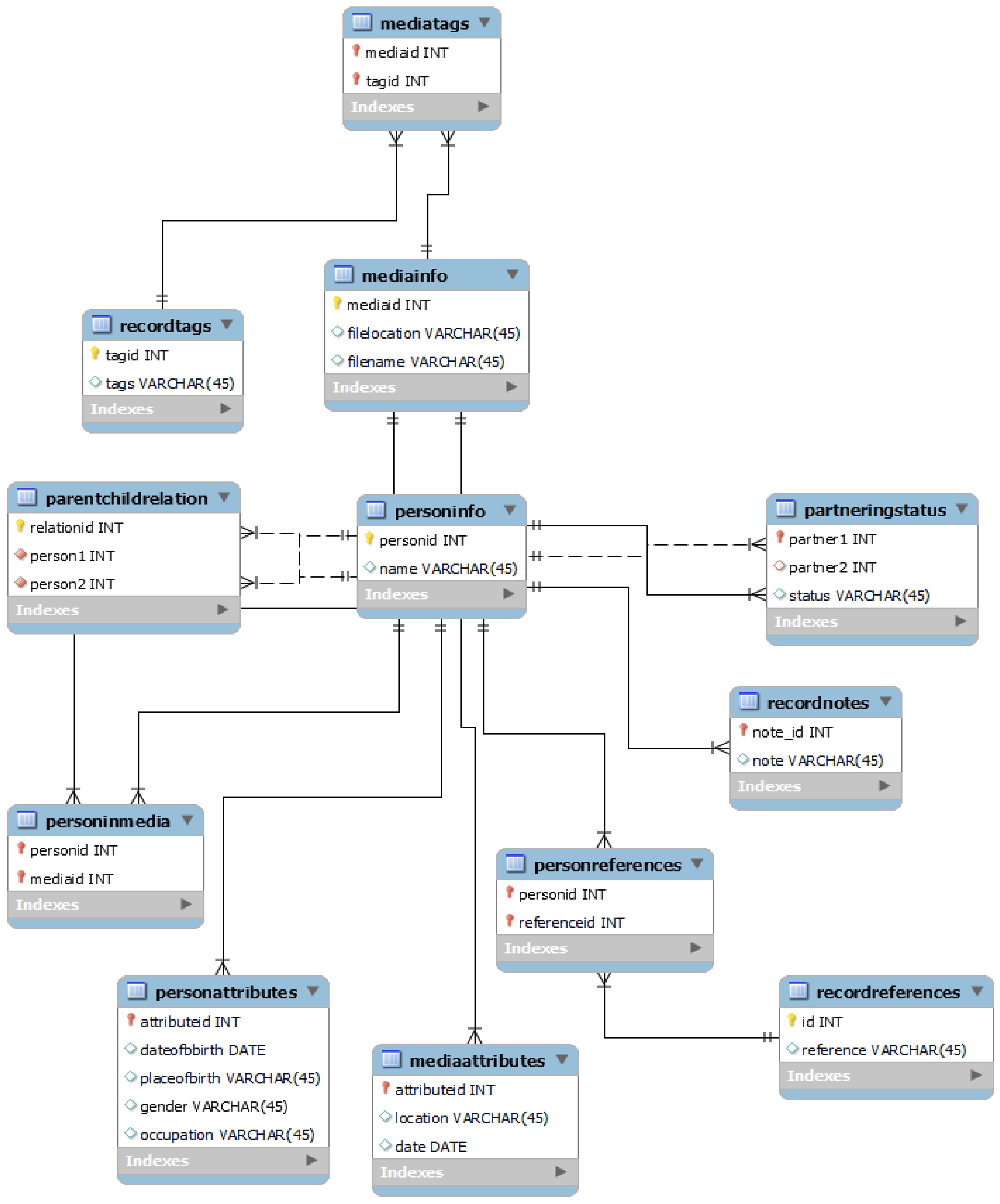
This data structure works fine with single root, roots can be, person in the relation table which does not have any parent, will be updated as root. However, this does not work with multiple roots. If any person in the tree is given more than one parent, this data structure will not work. I am trying to use multiple data structure to store parents and child of a node.

3. For finding ancestors and descendants
   - Two vector arrays can be used to find the ancestors and descendants.
   - Using recursion for each level, ancestors, and descendants according to the generations can be found.

Prachi Prakashbhai Raval
B00883324
pr647062@dal.ca

12th November,2021

ER Diagram Table Schema (As per current table relationships (Might be modified later)



personid in personinfo table – is the primary key which identifies a person uniquely and acts as a foreign key for all other tables, which store information about the person.

mediaid in mediainfo table – is the primary key which uniquely identifies a media file and acts as foreign key for all other tables which store information about the media file.

Code Design and Algorithm:

- *Classes*

  Genealogy – This class will contain all the methods related to family tree management, media archive management, and reporting. This is an overarching class which pulls everything together.

PersonIdentity: This class will consist of two constructors, getters and setters for the name and id.

1. Parameterized - PersonIdentity (String name, int personId) This create new objects and store them in a map.
2. Non parameterized – This will be called every time a new object of PersonIdentity is created in any other class. This is also used to get all the data already from the Database and create objects, after re-running the program.

FileIdentifier: This class is like PersonIdentity, it contains getters and setters and constructors for media management.

BiologicalRelationship: This class contain int degree of removal, int degree of removal., and getter and setter methods, for the same. Relationship between two persons is stored in a unique object of Biological Relationship.

FindRelation – I am making this class to find the degree of removal and cousinship. Every time findRelation method is called, this class calculates the degree of removal and creates and object of relation and using setters sets the value of degree of removal and degree of cousinship. Later the method in Find Relation class returns the object to find Relation method in the genealogy class. Find Relation class also contains methods to find ancestors and descendants of a person within the generations specified.

Prachi Prakashbhai Raval
B00883324
pr647062@dal.ca

White Box test cases:

Input Validation

- Family tree management
    1. Person Name, Location of Birth, Location of Death, Gender, References, Notes
        - String is null
        - String is empty
        - String contains special characters
    2. Date of birth, Date of death (Date format dd/mm/yyyy)
        - String is empty
        - String is null
    3. Map attribute is empty.

- Media Archive management
    1. Media Name, Location of picture taken, Tags
        - String is null
        - String is empty
        - String contains special characters
    2. Location of file, Date of picture taken
        - String is empty
        - String is null
    3. Map attribute is empty.

Boundary Cases

- Family tree management
    1. Person Name, Location of Birth, Location of Death, Gender, References, Notes
        - String is one character
        - String length does not exceed column varchar length
    2. Date of birth, Date of death
        - Day is 01.
        - Date is 31
        - Month is 01
        - Month is 12.

- Media Archive management
    1. Media Name, Location of picture taken, Tags

    - String is one character
    - String length does not exceed column varchar length
2. Location of file, Date of picture taken
    - Day is 01.
    - Date is 31
    - Month is 01
    - Month is 12.