Trentoniana Redesign

CSC315 and HON270/LNG371 - Group 2

Team Members:

Jabili Gadde - gaddej1@tcnj.edu

Nicole Kercado - kercadn1@tcnj.edu

Lakaylah Gage - gagel1@tcnj.edu

Anthony Scarpa - scarpas3@tcnj.edu

Claire Segal - segalc1@tcnj.edu

Executive Summary

The objective of this project was to redesign both the database of interview recordings associated with the Trentoniana website as well as the site itself. Both the original Trentoniana site and the database are extremely dated, and not efficient to work with in general. Filtering options on the site are very limited and the overall layout of the UI is very confusing. When our group discussed potential improvements to the site, we immediately turned to the idea of making filtering for recordings easier. This improvement allows users to search for a specific recording through various categories, whether it be the name of the recording, the gender or nationality of the speaker, the topics mentioned in the recording, the decade the recording was recorded, etc. We also needed to include superuser functionality so that administrators of the Trentoniana site would be able to add, insert, and modify tuples in the tables of the database. As such, we provided superusers the ability to insert, delete, and modify recordings, speakers, transcripts, and the topics and categories associated with the recordings. We also provided necessary security to the superuser functionality, as the functionality cannot be accessed without logging in. The usernames and encrypted passwords are stored in the database, and superusers can add and remove superusers as necessary. We implemented our own front end using HTML and CSS and used Python-based Flask to connect our PostgreSQL database to our front end. Our front end UI was implemented to be more easily navigable, more user friendly, more approachable, and cleaner than the original site. Additionally, our PostgreSQL database's tables neatly organize all of the data in a way that makes maintenance and querying simple. The cost of implementing our project would depend on the scale that it would be implemented on. The project was developed to fully replace the old Trentoniana site and database. However, our group developed it on a small scale, with only small amounts of sample data in the database, and the database and site being hosted on our own server on a virtual

machine. The cost of this site would be the cost of maintaining the database (in accordance with how large it needs to be) as well as the cost of hosting the site.

Stage II: Project Proposal & Specifications

*All notes/revisions are in red

Problem Statement: There are a lot of different recordings on the site; however, there aren't many categories for the user to filter these recordings. Also, there is very limited metadata on each recording. The topics category is vague and unhelpful. Additionally, these recordings are inaccessible to the hard of hearing, as there are no written transcripts available.

Objective of the Module: Our objective is to make it easier for the user to navigate through this website and find the recordings that they want to listen to. We will do this by including categories that the user can filter by. We will also include the written transcripts. The categories created will be based off of both the transcripts and the metadata. They both will include information that we can use to make the data for our categories, and these categories will be helpful to the user while searching and navigating through the site.

Description of the desired end product, and the part you will develop for this class: Our desired end product is a web based user interface that organizes a collection of recordings and transcripts. We will redesign the search process by adding categories to filter by, include written transcripts of the recordings in the website, and redesign the UI to look more user friendly and approachable.

Description of the importance and need for the module, and how it addresses the problem: The interface is currently not very user friendly, so the addition of more categories to filter by can make it easier to navigate and find specific recordings and transcripts. The way a website looks can affect how the user perceives it and right now, the website looks dated. To address this issue, we will redesign the UI to look more user friendly and approachable. Currently, there are no written transcripts so the hard of hearing cannot access these recordings; we will include written transcripts in the website.

Plan for how you will research the problem domain and obtain the data needed: Analyze the current website, recordings, and written transcripts transcribed by the LNG class.

Other similar systems / approaches that exist, and how your module is different or will add to the existing system: Currently, the website has some categories that it can filter by, so it does have queries in place; however, these are not particularly effective. The redesigned search of our module will make navigation and search easier and quicker. A similar system is the Schomburg Center, which has a digital platform with podcasts, live streams, online exhibitions, oral history projects, and more. However, their website seems very "crammed" and has an overwhelming amount of content put all in the same place; which makes it harder to navigate and can confuse users. Our site will be easier to navigate and we will make sure not to overwhelm users with content. We want to be clear and thorough but efficient. Their website also does not include filters; however, our module will include filtering options so that users can find what they're looking for faster.

Possible other applications of the system (how it could be modified and reused): The general idea of our system can be applied to other archives in general. Any collection of media, recordings, or even text can be filtered based on relevant categories. Essentially, by changing the categories used, our system could be molded to fit another collection of digitized materials, where the user can filter based on those categories.

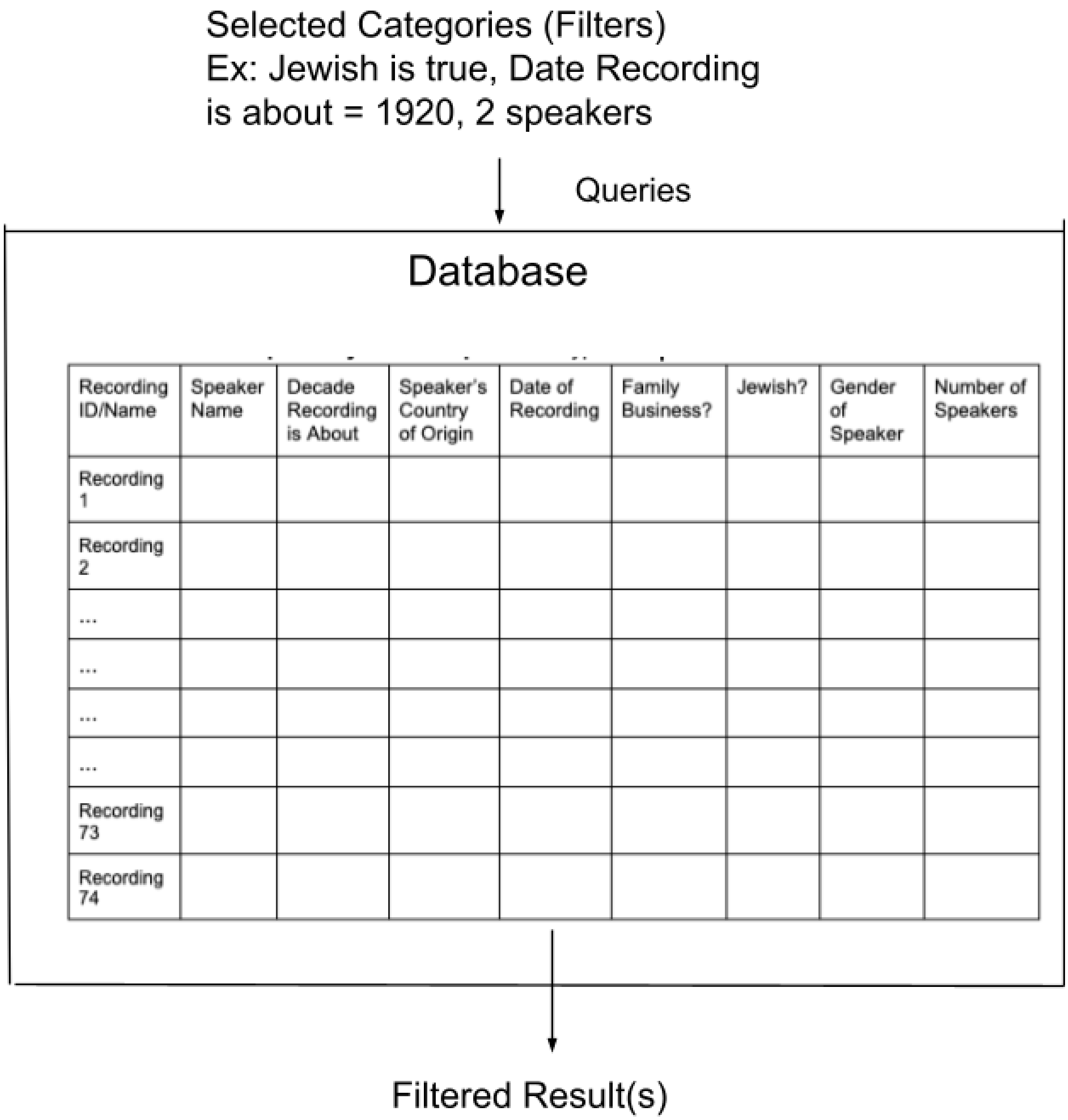
Performance – specify how and to what extent you will address this: The users of the website will be able to filter the recordings by different categories and will also have access to written transcripts of the recordings when they find what recording they want to access. Some examples of categories are speaker name, the decade that the recording is about, the speaker's country of origin, and the date of recording (which is already there, but we will include it for consistency). Some other examples are whether or not the recording refers to a family business, what the gender of the primary speaker is, the number of speakers, and whether or not the speaker is Jewish (which is already under the topics category on the website, but we will include it for consistency and make the UI for it more appealing).

Security – specify how and to what extent you will provide security features:
Different tiers of users (ex. head librarian who has access to the entire database and can make any modifications.

Backup and recovery – specify how and to what extent you will implement this: Every day at midnight, our project will create a backup copy of the database.

Technologies and database concepts the team will need to learn, and a plan for learning these: We need to gain experience with front end development, we need to learn how to create a database as well as how to query, we need to learn how to back up a database (as well as making it backup daily at a specific time), and we need to learn how to best maximize performance.

A diagrammatic representation of the system boundary that specifies what data you will model and which queries you will implement:



Selected Categories (Filters)
Ex: Jewish is true, Date Recording
is about = 1920, 2 speakers

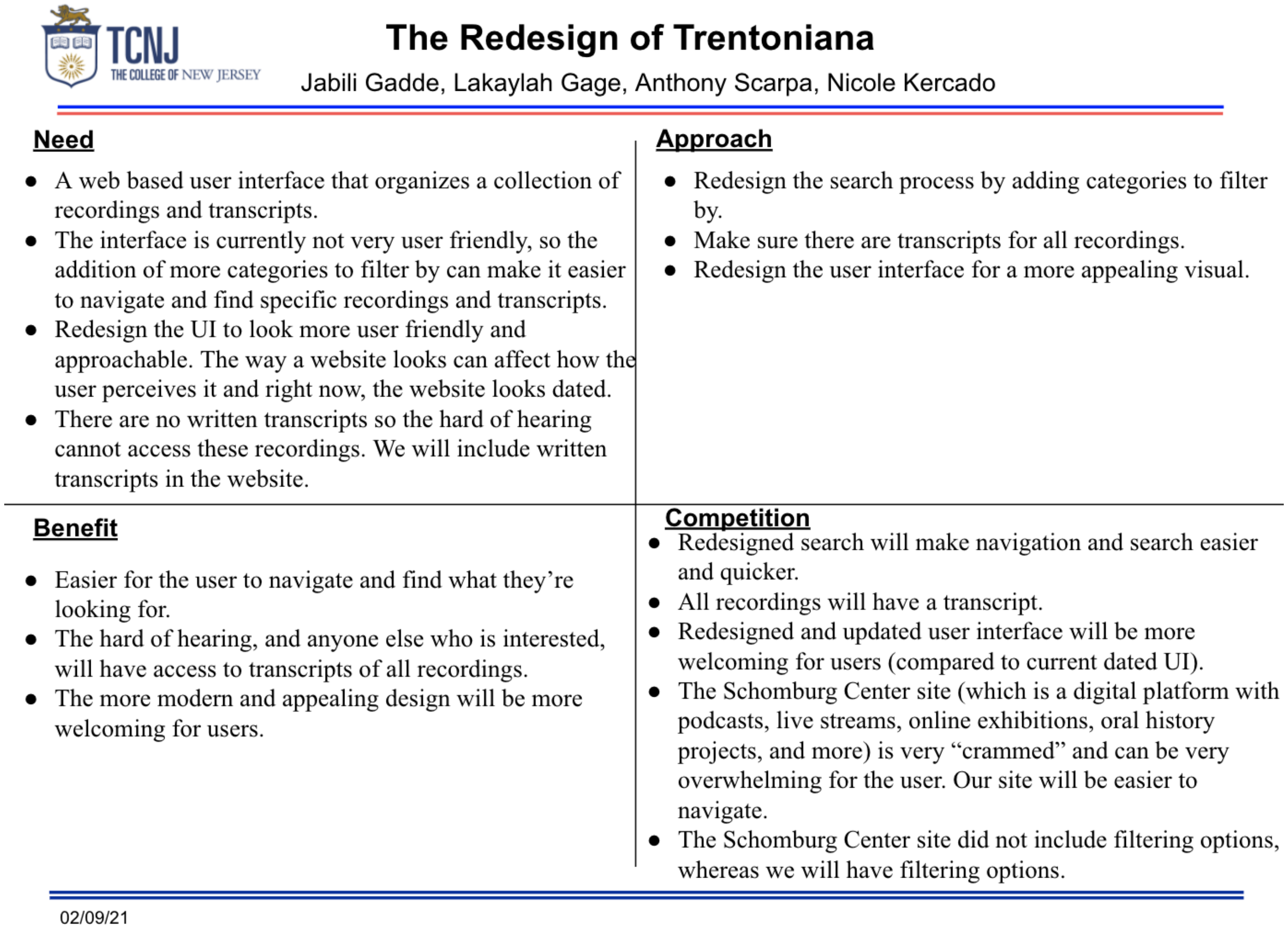
Queries

Database

Recording ID/Name	Speaker Name	Decade Recording is About	Speaker's Country of Origin	Date of Recording	Family Business?	Jewish?	Gender of Speaker	Number of Speakers
Recording 1								
Recording 2								
...								
...								
...								
...								
Recording 73								
Recording 74								

Filtered Result(s)

1-page quad chart:



The Redesign of Trentoniana

Jabili Gadde, Lakaylah Gage, Anthony Scarpa, Nicole Kercado

Need

- A web based user interface that organizes a collection of recordings and transcripts.
- The interface is currently not very user friendly, so the addition of more categories to filter by can make it easier to navigate and find specific recordings and transcripts.
- Redesign the UI to look more user friendly and approachable. The way a website looks can affect how the user perceives it and right now, the website looks dated.
- There are no written transcripts so the hard of hearing cannot access these recordings. We will include written transcripts in the website.

Approach

- Redesign the search process by adding categories to filter by.
- Make sure there are transcripts for all recordings.
- Redesign the user interface for a more appealing visual.

Benefit

- Easier for the user to navigate and find what they're looking for.
- The hard of hearing, and anyone else who is interested, will have access to transcripts of all recordings.
- The more modern and appealing design will be more welcoming for users.

Competition

- Redesigned search will make navigation and search easier and quicker.
- All recordings will have a transcript.
- Redesigned and updated user interface will be more welcoming for users (compared to current dated UI).
- The Schomburg Center site (which is a digital platform with podcasts, live streams, online exhibitions, oral history projects, and more) is very "crammed" and can be very overwhelming for the user. Our site will be easier to navigate.
- The Schomburg Center site did not include filtering options, whereas we will have filtering options.

02/09/21

Note: The diagrammatic representation above is an early sketch and is not an accurate representation of the actual database.

Revisions:

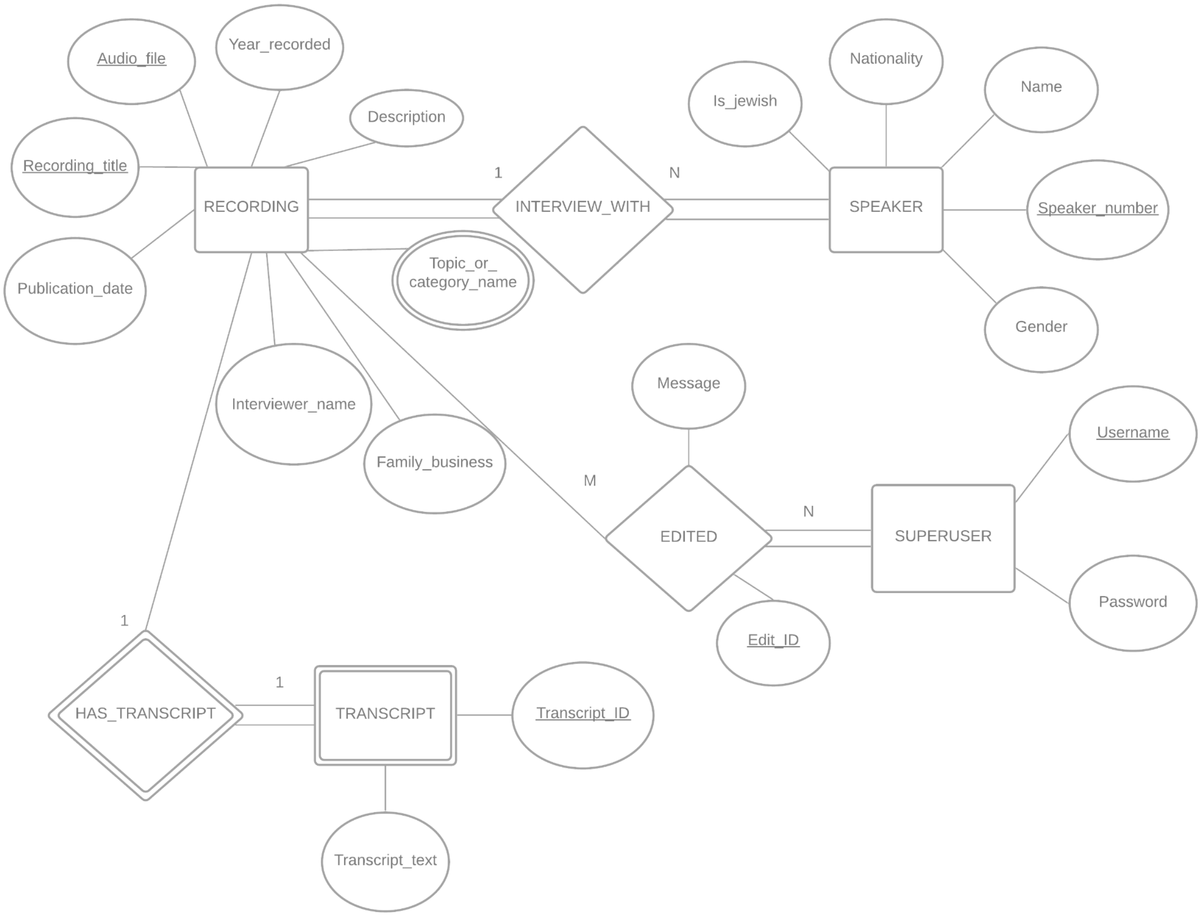
- We could not guarantee that there would be transcripts for all recordings. However, with time and people experienced at transcribing audio, written transcripts could certainly be created. The example the group presented with the database only contained recordings which already had transcripts made by the English department.
- The idea of having "different tiers" of users was somewhat kept, however there were only two tiers of users left by the end, that being a superuser and a normal user.
- The project's database does not back itself up on a regular basis. Instead, it must be done manually.

Stage III and IV: Design

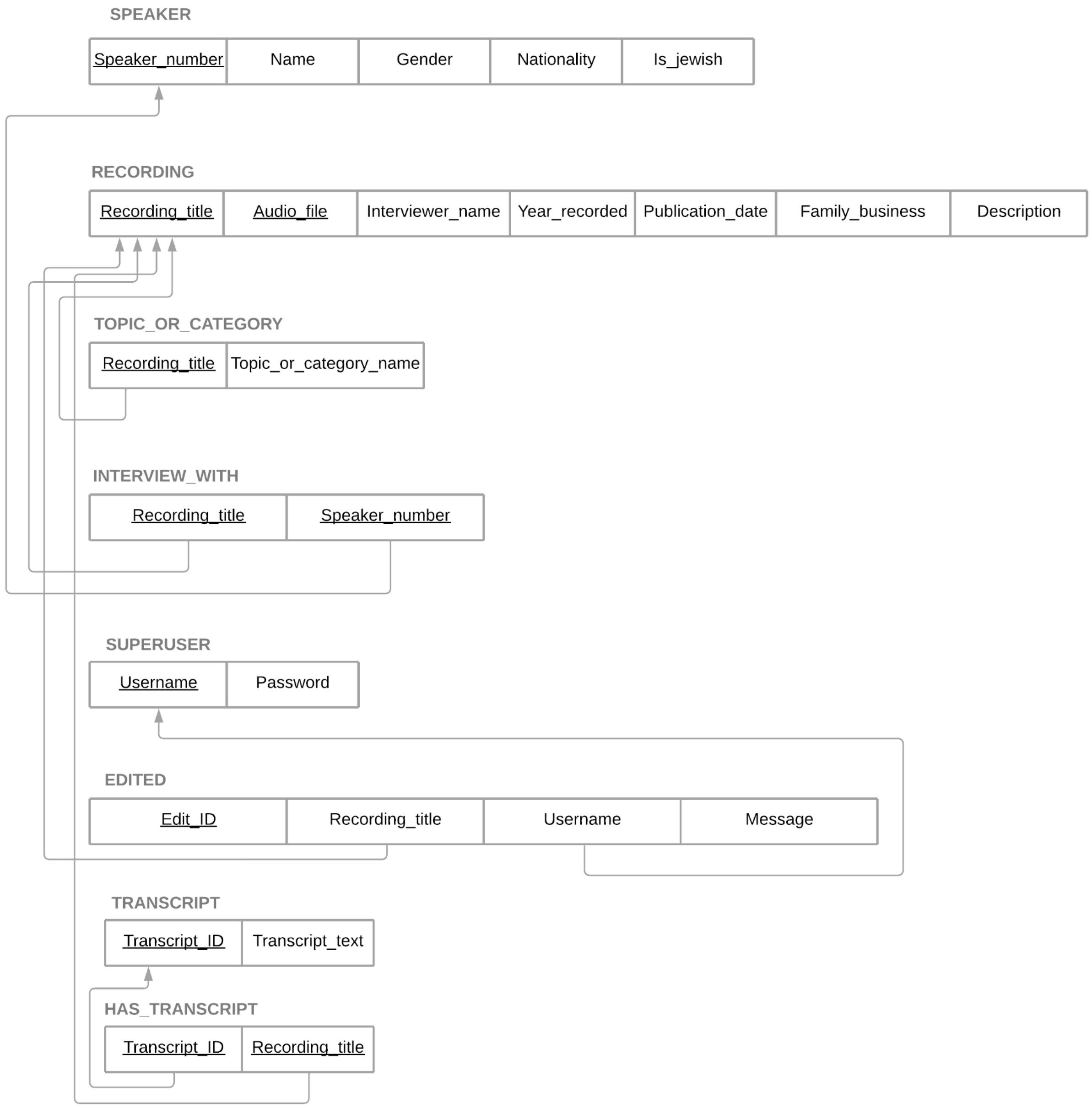
*All notes/revisions are in red

Stage III: Database Model

ER Diagram:



ER Diagram Mapping to Relational Schema:



SPEAKER

Speaker_number	Name	Gender	Nationality	Is_jewish

RECORDING

Recording_title	Audio_file	Interviewer_name	Year_recorded	Publication_date	Family_business	Description

TOPIC_OR_CATEGORY

Recording_title	Topic_or_category_name

INTERVIEW_WITH

Recording_title	Speaker_number

SUPERUSER

Username	Password

EDITED

Edit_ID	Recording_title	Username	Message

TRANSCRIPT

Transcript_ID	Transcript_text

HAS_TRANSCRIPT

Transcript_ID	Recording_title

Calculating the Average Amount of Searches Per Day:

The total number of views for all the recordings combined was 2,215 (from adding up the views of each recording on the Trentoniana site). The collection of the recordings itself had been up for roughly 1,389 days (the site states that the collection of recordings was published on May 23, 2017), meaning that the average amount of searches per day was 2,215 / 1,389, or 1.59 searches per day.

Approximating the Number of Records (Initial Database Size):

There are currently 74 recordings on the site. This means there will be 74 tuples for the RECORDING relation. If we assume that each recording has on average 2 speakers, that means there will be 74 * 2 tuples for the SPEAKER relation (148 tuples). This also means there will be 148 tuples for the INTERVIEW_WITH relationship. If there are 2 topics per recording, there will be 148 tuples for the Topic_or_category relation. If we assume that there are 5-10 super-users who can essentially edit/insert/delete recordings, then we can approximate that there would be 8 tuples for the SUPERUSER relation. We can assume that a super-user would be able to edit/insert/delete every recording, so there would be 74 * 8 tuples for the EDITED relationship. If we assume that there are 24 recordings with transcripts, then that there would be 24 tuples in the TRANSCRIPT relation and 24 tuples for the HAS_TRANSCRIPT relationship.

So to approximate the number of records, 74 + 148 + 148 + 148 + 8 + 592 + 24 + 24 = 1166 records.

Types of Searches:

- Metadata - The data of our data will be searched through.
- Text contents - Any text in the site will be searched through, including any transcripts and descriptions.
- Limit options - On the site, there will be filters available for the user to utilize to narrow down the search. They can filter by topic/category, nationality, gender, and more.

Revisions:

- Added a TRANSCRIPT weak entity, a HAS_TRANSCRIPT identifying relationship, an ADMINISTRATOR entity, and a CAN_EDIT (now removed) entity to our ER diagram
- Added a Description attribute to the RECORDING entity and deleted Exact_date from the RECORDING
- Deleted the CAN_EDIT relation and replaced it with the EDITED relation, which has the attributes Edit_ID (primary key), Recording_title, Username, and Message

Stage IV: Design

*All revisions are in red

Jabili Gadde, Nicole Kercado, Lakaylah Gage, Anthony Scarpa
CSC 315-01
Stage 4: Design

Step 2:

SPEAKER is in BCNF. The table is in 1NF as each of the values are atomic. Each of the non-prime attributes (Name, Gender, Nationality, Is_jewish) are fully functionally dependent on the primary key (Speaker_number), meaning that the table is in 2NF. In addition, none of the non-prime attributes are transitively dependent on the primary key, so the table is also in 3NF. Lastly, since there are no prime attributes (Speaker_number) that are dependent on a non-prime attribute (the second condition for 3NF regarding prime attributes does not apply), the table is in BCNF.

RECORDING is in BCNF. The table is in 1NF as each of the values are atomic. Each of the non-prime attributes (Interviewer_name, Year_recorded, Publication_date, Number_of_speakers, Family_business, Description) are fully functionally dependent on the primary key ({Recording_title, Audio_file}), meaning that the table is in 2NF. In addition, none of the non-prime attributes are transitively dependent on the primary key, so the table is also in 3NF. Lastly, for each functional dependency in the table (Ex. A->B), A is a superkey. Since there are no prime attributes (Recording_title, Audio_file) that are dependent on a non-prime attribute, the table is in BCNF.

TOPIC_OR_CATEGORY is in BCNF because the relation is a binary relation. The table is in 1NF as each of the values are atomic. The non-prime attribute Topic_or_category_name is fully functionally dependent on the primary key Recording_title, so the table is in 2NF. Since the table only has two attributes, a transitive dependency cannot occur, so the table is also in 3NF. Lastly, for the functional dependency in the table (Recording_title->Topic_or_category_name), Recording_title is a superkey, meaning that the table is also in BCNF.

INTERVIEW_WITH is in BCNF because the relation is a binary relation. The table is in 1NF as each of the values are atomic. Since partial dependency is not possible, the table is in 2NF. In addition, the table is also in 3NF since a transitive dependency cannot occur in a relation with only two attributes. Lastly, the attributes in the table (Recording_title and Speaker_number) can be individual superkeys, or they can be combined as a set to be one superkey, meaning that the table is also in BCNF.

SUPERUSER is in BCNF because the relation is a binary relation. The table is in 1NF as each of the values are atomic. The non-prime attribute Password is fully functionally dependent on the primary key Username, so the table is in 2NF. Since the table only has two attributes, a transitive dependency cannot occur, so the table is also in 3NF. Lastly, for the functional dependency in the table (Username->Password), Username is a superkey, meaning that the table is also in BCNF.

EDITED is in BCNF. The table is in 1NF as each of the values are atomic. Each of the non-prime attributes (Recording_title, Username, Message) are fully functionally dependent on the primary key (Edit_id), meaning that the table is in 2NF. In addition, none of the non-prime attributes are transitively dependent on the primary key, so the table is also in 3NF. Lastly, for each functional dependency in the table (Ex. A->B), A (Edit_id) is a superkey. Since there are no prime attributes (Edit_id) that are dependent on a non-prime attribute (the second condition for 3NF regarding prime attributes does not apply), the table is in BCNF.

TRANSCRIPT is in BCNF because the relation is a binary relation. The table is in 1NF as each of the values are atomic. The attribute Transcript_text is fully functionally dependent on the primary key Transcript_id, so the table is in 2NF. Since the table only has two attributes, a transitive dependency cannot occur, so the table is also in 3NF. Lastly, for the functional dependency in the table (Transcript_id->Transcript_text), Transcript_id is a superkey, meaning that the table is also in BCNF.

HAS_TRANSCRIPT is in BCNF because the relation is a binary relation. The table is in 1NF as each of the values are atomic. Since partial dependency is not possible, the table is in 2NF. In addition, the table is also in 3NF since a transitive dependency cannot occur in a relation with only two attributes. Lastly, the attributes in the table (Transcript_id and Recording_title) can be individual superkeys, or they can be combined as a set to be one superkey, meaning that the table is also in BCNF.

Step 3:

Views:

- **User**
  - A normal user that comes to the website to either listen to recordings, view transcripts, or read the descriptions for recordings.
  - Transaction Requirements:
    - Play a recording
      - Display the recording title and an audio file
    - Display a transcript
      - Display the transcript text as well as the recording title
    - Display description
      - Shows:

      - Description of the recording
      - Publication date
      - Year recorded
      - Topics/Categories
    - Filter recordings based on
      - Topics/Categories
      - The decade ~~year~~ the recording was recorded
      - Whether a Family Business is mentioned in the recording
      - The nationality of the speaker(s)
      - Whether or not the speaker(s) is Jewish
      - Gender of the speaker(s)
      - ~~Number of speakers~~
    - Search based on recording title
  - Data Requirements:
    - Play a recording
      - Audio_file (attribute of Recording)
        - To play it
      - Recording_title (attribute of Recording)
        - To find the corresponding audio file
    - Display a transcript
      - Transcript_text (attribute of Transcript)
        - To display the text
      - Recording_title (attribute of Recording)
        - To find the corresponding transcript to a recording
        - Will be used to do a join on the Recording and Has_Transcript relations
      - Transcript_ID (attribute of Transcript)
        - Will be used to do a join on the Transcript relation and the already existing join on the Recording and Has_Transcript relations
    - Display description
      - Description (attribute of Recording)
        - To display it
      - Publication_date (attribute of Recording)
        - To display it
      - Year_recorded (attribute of Recording)
        - To display it
      - Recording_title (attribute of Recording)
        - To find the corresponding description, publication date, and year recorded of a recording
    - Filter recordings based on the filters listed above
      - Topic_or_ category_name (multivalued attribute of Recording; attribute of relation Topic_Or_Category)

      - To filter recordings based on user input (which topics and categories the user wants the recordings displayed to be about)
    - Year_recorded (attribute of Recording)
      - To filter recordings based on user input (to display which recordings were recorded in the years that the user selected)
    - Family_business (attribute of Recording)
      - To filter recordings based on user input (to display recordings that mention a family business or not)
    - ~~Number_of_speakers (attribute of Recording)~~
      - ~~To filter recordings based on user input (to display recordings that have a certain number of speakers)~~
    - Nationality (attribute of Speaker)
      - To filter recordings based on user input (to display only recordings that have at least one speaker of the nationality selected by the user)
    - Is_Jewish (attribute of Speaker)
      - To filter recordings based on user input (to display only recordings that have at least one speaker that is Jewish)
    - Gender (attribute of Speaker)
      - To filter recordings based on user input (to display only recordings that have at least one speaker that is the gender selected by the user)
    - Recording_title (attribute of Recording)
      - To find the corresponding attributes of Year_recorded, Family_business, Number_of_speakers
      - To display the recording titles after the filter has completed
      - To do a join on the relations Recording and Topic_or_category in order to get all the corresponding topics/categories of a recording so as to filter on it
      - To do a join on the relations Recording and Interview_with so as to later know if any of the speakers are Jewish or are of the selected gender or nationality after doing a join on the resulting join and the relation Speaker
    - Speaker_number (attribute of Speaker)
      - To do a join on the relation Speaker and the resulting relation from the join on Recording and Interview_with
  - Search based on recording title
    - Recording_title
      - Needed to find the corresponding recording
- Example Queries:
  - Filter the recordings to find recordings that were recorded in 1989, have a Jewish speaker, and have exactly 1 speaker.

    - Show the description of the recording “Dr. Paul Loser”.
    - Search for the recording that has the recording title of “Izzy Lynn”.
    - Display a transcript for the recording “Mel Kushner”.
- **Superuser**
  - Can see everything and do everything that a normal user can do, but can also change the contents of the database.
  - Transaction Requirements:
    - Can do all the transactions that a user can do
    - Log in and out of the system
      - The user supplies the username and password
    - Display a recording’s attributes and its corresponding speakers and transcript and their attributes as well
      - Upon supplying the title of a recording, the superuser will be able to see this data.
    - Add a recording
      - Add a tuple to the Recording relation which will be able to be accessed by the users after doing so.
    - Add a topic/category for a Recording
      - Add a tuple to the Topic_or_category relation which will be able to be accessed by the users after doing so.
    - Add a transcript
      - Add a tuple to the Transcript relation which will be able to be accessed by the users after doing so.
      - Add a tuple to the Has_transcript relation as well
    - Add a speaker
      - Add a tuple to the Speaker relation (the superuser will have to supply the title of the recording)
      - Add a tuple to the Interview_with relation as well
    - Add a superuser
      - Add a tuple to the Superuser relation
    - Delete a recording
      - Delete a tuple from the Recording relation which will no longer be able to be accessed by the users after doing so.
    - Delete a topic/category for a Recording
      - Delete a tuple from the Topic_or_category relation
    - Delete a transcript
      - Delete a tuple from the Transcript relation which will no longer be able to be accessed by the users after doing so.
      - Delete a tuple from the Has_transcript relation as well
    - Delete a speaker
      - Delete a tuple from the Speaker relation (the superuser will have to supply the speaker number)
      - Delete a tuple from the Interview_with relation
    - Delete a superuser

    - Delete a tuple from the Superuser relation
  - Edit/Update a recording
    - Modify a tuple in the Recording relation which will be updated in the User View after doing so.
  - Edit/Update a transcript
    - Modify a tuple in the Transcript relation which will be updated in the User View after doing so.
  - Edit/Update a speaker
    - Modify a tuple in the Speaker relation (the superuser will have to supply the speaker number)
- Data Requirements:
  - Log in and out of the system
    - Attributes of Superuser (needed to log into the system as a superuser)
      - Username
      - Password
  - Display a recording's attributes and its corresponding speakers and transcript and their attributes as well
    - Attributes of Recording (needed to display to the superuser):
      - Recording_title
        - Superuser input (needed to find the corresponding recording and its attributes)
        - Also needed to do a join on the relations Recording and Interview_with in order to later find the corresponding speakers
        - Also needed to do a join on the relations Recording and Transcript in order to later find the corresponding transcript
      - Audio_file
      - Year_recorded
      - Interviewer_name
      - Publication_date
      - Number_of_speakers
      - Family_business
      - Description
    - Attributes of Speaker (needed to display to the superuser):
      - Speaker_number
        - Needed to do a join on the relation Speaker and the resulting relation from the join on Recording and Interview_with so as to find the corresponding speakers for a recording
      - Name
      - Gender
      - Nationality

        - Is_Jewish
    - Attributes of Transcript (needed to display to the superuser):
        - Transcript_ID
            - Needed to do a join on the relation Transcript and the resulting relation from the join on Recording and Transcript so as to find the corresponding transcript for a recording
        - Transcript_text
- Add a recording
    - Attributes of Recording (to insert a tuple into the Recording relation):
        - Recording_title
        - Audio_file
        - Year_recorded
        - Interviewer_name
        - Publication_date
        - Number_of_speakers
        - Family_business
        - Description
- Add a topic/category
    - Attributes of Topic_or_category (to insert a tuple into the Topic_or_category relation)
        - Recording_title
        - Topic_or_category_name
- Add a transcript
    - Attributes of Transcript (to insert a tuple into the Transcript relation);
        - Transcript_ID
        - Transcript_text
    - Attributes of Has_transcript (to insert a tuple into the Has_transcript relation)
        - Transcript_ID
        - Transcript_text
- Add a speaker
    - Attributes of Speaker (to insert a tuple into the Speaker relation):
        - Speaker_number
        - Name
        - Gender
        - Nationality
        - Is_Jewish
    - Attributes of Interview_with (to insert a tuple into the Interview_with relation)
        - Recording_title
        - Speaker_number

- Add a superuser
    - Attributes of Superuser (to insert a tuple into the Superuser relation):
        - Username
        - Password
- Delete a recording
    - Recording_title
        - Needed to delete a tuple from the Recording relation
- Delete a topic/category
    - Recording_title
        - Needed to delete a tuple from the Topic_or_category relation
    - Topic_or_category_name
        - Needed to delete a tuple from the Topic_or_category relation
- Delete a transcript
    - Transcript_ID
        - Needed to delete a tuple from the Transcript relation and the Has_transcript relation
- Delete a speaker
    - Speaker_number
        - Needed to delete a tuple from the Speaker relation and the Interview_with relation
- Delete a superuser
    - Username
        - Needed to delete a tuple from the Superuser relation
- Edit/Update a recording
    - Attributes of Recording (we don’t know which attributes the superuser will edit, so we may potentially need them all):
        - Recording_title
            - Needed to find the recording to be edited
        - Audio_file
        - Year_recorded
        - Interviewer_name
        - Publication_date
        - Number_of_speakers
        - Family_business
        - Description
- Edit/Update a transcript
    - Attributes of Transcript (we don’t know which attributes the superuser will edit, so we may potentially need them all);
        - Transcript_ID
            - Needed to find the transcript to be edited
        - Transcript_text

- Edit/Update a speaker
  - Attributes of Speaker (we don’t know which attributes the superuser will edit, so we may potentially need them all):
    - Speaker_number
      - Needed to find the recording to be edited
    - Name
    - Gender
    - Nationality
    - Is_Jewish
- Example Queries:
  - Add a recording called “Johnny Appleseed”.
  - Add a transcript to the recording “Dr. Paul Loser”.
  - Delete a recording called “Mel Kushner”.
  - Edit a recording called “Izzy Lynn”.

Step 4:

List of Transactions and their set of SQL queries:

User:

- Play a recording
  - SELECT Audio_file
    FROM RECORDING
    WHERE Recording_title = title;
  - title is representative of the desired recording
- Display a transcript
  - CREATE VIEW SEL AS
    SELECT *
    FROM RECORDING
    WHERE Recording_title = title;
    - title is representative of the desired recording
  - CREATE VIEW REC_HASTRANS AS
    SELECT *
    FROM SEL
    NATURAL JOIN HAS_TRANSCRIPT;
  - SELECT Transcript_text
    FROM REC_HASTRANS
    NATURAL JOIN TRANSCRIPT;
  - DROP REC_HASTRANS;
  - DROP SEL;
- Display description
  - SELECT Description, Publication_date, Year_recorded
    FROM RECORDING
    WHERE Recording_title = title;

    - title is representative of the desired recording
- Filter recordings based on Topics/Categories
    - CREATE VIEW RECORDING_TOPIC AS
      SELECT *
      FROM RECORDING
      NATURAL JOIN TOPIC_OR_CATEGORY;
    - SELECT Recording_title
      FROM RECORDING_TOPIC
      WHERE Topic_or_category_name = tc1 OR Topic_or_category_name = tc2 OR …;
        - tc1, tc2, … are the topics and categories selected by the user
    - DROP VIEW RECORDING_TOPIC;
- Filter recordings based on the decade ~~year~~ the recording was recorded
    - SELECT Recording_title
      FROM RECORDING
      WHERE Year_recorded = year1 OR Year_recorded = year2 OR ...;
    - year1, year2, … are the years selected by the user
- Filter recordings based on whether a Family Business is mentioned in the recording
    - SELECT Recording_title
      FROM RECORDING
      WHERE Family_Business = familyBusiness;
    - familyBusiness is representative of whether or not the user wants the recording to mention a family business
- Filter recordings based on the nationality of the speaker(s)
    - ~~CREATE VIEW RECORDING_SPEAKER AS~~
      ~~SELECT *~~
      ~~FROM RECORDING~~
      ~~NATURAL JOIN SPEAKER;~~
    - ~~SELECT Recording_title~~
      ~~FROM RECORDING_SPEAKER~~
      ~~WHERE Nationality = n1 OR Nationality = n2 OR …;~~
        - ~~n1, n2, … are the nationalities selected by the user~~
    - ~~DROP VIEW RECORDING_SPEAKER;~~
    - CREATE VIEW RECORDING_INTERVIEW AS
      SELECT *
      FROM RECORDING
      NATURAL JOIN INTERVIEW_WITH;
    - CREATE VIEW RECORDING_SPEAKER AS
      SELECT *
      FROM RECORDING_INTERVIEW
      NATURAL JOIN SPEAKER;
    - SELECT Recording_title
      FROM RECORDING_SPEAKER
      WHERE Nationality = n1 OR Nationality = n2 OR …;

    - n1, n2, … are the nationalities selected by the user
  - DROP VIEW RECORDING_SPEAKER;
  - DROP VIEW RECORDING_INTERVIEW;
- Filter recordings based on whether or not the speaker(s) is Jewish
  - ~~CREATE VIEW RECORDING_SPEAKER AS~~
    ~~SELECT *~~
    ~~FROM RECORDING~~
    ~~NATURAL JOIN SPEAKER;~~
  - ~~SELECT Recording_title~~
    ~~FROM RECORDING_SPEAKER~~
    ~~WHERE Is_jewish = sIsJewish;~~
    - ~~sIsJewish is representative of whether or not the user wants the recording to have a Jewish speaker~~
  - ~~DROP VIEW RECORDING_SPEAKER;~~
  - CREATE VIEW RECORDING_INTERVIEW AS
    SELECT *
    FROM RECORDING
    NATURAL JOIN INTERVIEW_WITH;
  - CREATE VIEW RECORDING_SPEAKER AS
    SELECT *
    FROM RECORDING_INTERVIEW
    NATURAL JOIN SPEAKER;
  - SELECT Recording_title
    FROM RECORDING_SPEAKER
    WHERE Is_jewish = sIsJewish;
    - sIsJewish is representative of whether or not the user wants the recording to have a Jewish speaker
  - DROP VIEW RECORDING_SPEAKER;
  - DROP VIEW RECORDING_INTERVIEW;
- Filter recordings based on gender of the speaker(s)
  - ~~CREATE VIEW RECORDING_GENDER AS~~
    ~~SELECT *~~
    ~~FROM RECORDING~~
    ~~NATURAL JOIN SPEAKER;~~
  - ~~SELECT Recording_title~~
    ~~FROM RECORDING_GENDER~~
    ~~WHERE Gender = gender1 OR Gender = gender2 OR …;~~
    - ~~gender1, gender2, … are the nationalities selected by the user~~
  - ~~DROP VIEW RECORDING_GENDER;~~
  - CREATE VIEW RECORDING_INTERVIEW AS
    SELECT *
    FROM RECORDING
    NATURAL JOIN INTERVIEW_WITH;
  - CREATE VIEW RECORDING_SPEAKER AS

    SELECT *
    FROM RECORDING_INTERVIEW
    NATURAL JOIN SPEAKER;

    - SELECT Recording_title
      FROM RECORDING_SPEAKER
      WHERE Gender = gender1 OR Gender = gender2 OR …;
        - gender1, gender2, … are the nationalities selected by the user
    - DROP VIEW RECORDING_SPEAKER;
    - DROP VIEW RECORDING_INTERVIEW;
- ~~Filter recordings based on number of speakers~~
    - ~~SELECT Recording_title,~~
      ~~FROM RECORDING~~
      ~~WHERE Number_of_speakers = numOfSpeakers1 OR Number_of_speakers = numOfSpeakers2 OR …;~~
    - ~~numOfSpeakers1, numOfSpeakers2, … are the number of speakers selected by the user~~
- Search based on recording title
    - CREATE EXTENSION pg_trgm;
    - CREATE INDEX recording_title_trigram ON RECORDING
      USING gist (Recording_title gist_trgm_ops);
    - SELECT Recording_title
      FROM RECORDING
      WHERE Recording_title % ‘title’;

Superuser:

- Log in and out of the system
    - SELECT Username
      FROM SUPERUSER
      WHERE Username = username AND Password = password;
        - username and password are the username and password that the superuser supplies
        - Return the Username if a matching username and password is found
- Display a recording’s attributes and its corresponding speakers and transcript and their attributes as well
    - ~~CREATE VIEW SEL AS~~
      ~~SELECT *~~
      ~~FROM RECORDING~~
      ~~WHERE Recording_title = title;~~
        - ~~title is user input~~
    - ~~CREATE VIEW REC_INTERVIEW AS~~
      ~~SELECT *~~
      ~~FROM SEL~~
      ~~NATURAL JOIN INTERVIEW_WITH;~~

    - ~~SELECT *~~
      ~~FROM REC_INTERVIEW~~
      ~~NATURAL JOIN SPEAKER;~~
        - ~~The recording's attributes and the speaker's attributes are displayed~~
    - ~~CREATE VIEW REC_HASTRANS AS~~
      ~~SELECT *~~
      ~~FROM SEL~~
      ~~NATURAL JOIN HAS_TRANSCRIPT;~~
    - ~~SELECT Transcript_ID, Transcript_text~~
      ~~FROM REC_HASTRANS~~
      ~~NATURAL JOIN TRANSCRIPT;~~
    - ~~DROP REC_HASTRANS;~~
    - ~~DROP REC_INTERVIEW;~~
    - ~~DROP SEL;~~
    - CREATE VIEW SEL AS
      SELECT *
      FROM RECORDING
      WHERE Recording_title = title;
        - title is user input
    - CREATE VIEW REC_INTERVIEW AS
      SELECT *
      FROM SEL
      NATURAL JOIN INTERVIEW_WITH;
    - SELECT *
      FROM SEL;
    - SELECT Speaker_number, Name, Gender, Nationality, Is_jewish
      FROM REC_INTERVIEW
      NATURAL JOIN SPEAKER;
        - The speaker's attributes are displayed
    - CREATE VIEW REC_HASTRANS AS
      SELECT *
      FROM SEL
      NATURAL JOIN HAS_TRANSCRIPT;
    - SELECT Transcript_ID, Transcript_text
      FROM REC_HASTRANS
      NATURAL JOIN TRANSCRIPT;
    - DROP VIEW REC_HASTRANS;
    - DROP VIEW REC_INTERVIEW;
    - DROP VIEW SEL;
- Add a recording
    - INSERT INTO RECORDING(Recording_title, Audio_file, Year_recorded, Interviewer_name, Publication_date, ~~Number_of_speakers~~, Family_business, Description)

VALUES (title, audioFile, year, interviewer, datePublished, ~~numOfSpeakers~~, familyBusiness, descript);
    - The above values are representative of user input.
- Add a topic/category
    - INSERT INTO TOPIC_OR_CATEGORY(Recording_title, Topic_or_category_name) VALUES (title, topic);
    - title and topic are representative of user input
- Add a transcript
    - INSERT INTO TRANSCRIPT(Transcript_ID, Transcript_text)
      VALUES (transcriptID, textCont);
    - INSERT INTO HAS_TRANSCRIPT(Transcript_ID, Recording_title) VALUES (transcriptID, title);
    - The above values are representative of user input.
- Add a speaker
    - INSERT INTO SPEAKER(Speaker_number, Name, Gender, Nationality, Is_jewish)
      VALUES (sNum, sName, sGender, sNationality, sIsJewish);
    - INSERT INTO INTERVIEW_WITH(Recording_title, Speaker_number) VALUES (title, (SELECT Speaker_number FROM SPEAKER WHERE Name = sName));
    - The above values are representative of user input.
- Add a superuser
    - INSERT INTO SUPERUSER(Username, Password)
      VALUES(username, password);
    - username and password are representative of user input
- Delete a recording
    - DELETE FROM RECORDING
      WHERE Recording_title = title;
    - Title is representative of user input.
- Delete a topic/category
    - DELETE FROM TOPIC_OR_CATEGORY
      WHERE Recording_title = title AND Topic_or_category_name = topic;
    - Title and topic are representative of user input
- Delete a transcript
    - DELETE FROM TRANSCRIPT
      WHERE Transcript_ID = trancriptID;
    - DELETE FROM HAS_TRANSCRIPT
      WHERE Transcript_ID = transcriptID;
    - trancriptID is representative of user input.
- Delete a speaker
    - DELETE FROM SPEAKER
      WHERE Speaker_number = sNum;
    - DELETE FROM INTERVIEW_WITH
      WHERE Speaker_number = sNum;
    - sNum is representative of user input.

- Delete a superuser
    - DELETE FROM SUPERUSER
      WHERE Username = username AND Password = password;
    - username and password are representative of user input
- Edit/Update a recording
    - UPDATE RECORDING
      SET Recording_title = title, Audio_file = audioFile, Year_recorded = year , Interviewer_name = interviewer, Publication_date = datePublished, ~~Number_of_speakers = numOfSpeakers~~, Family_business = familyBusiness, Description = descript
      WHERE Recording_title = title;
    - Depending on the user input, only some of these updates would take place. It depends on what the user wants to change in the tuple.
- Edit/Update a transcript
    - UPDATE TRANSCRIPT
      SET Transcript_ID = transcriptID, Transcript_text = textCont
      WHERE Transcript_ID = transcriptID;
    - Depending on the user input, only some of these updates would take place. It depends on what the user wants to change in the tuple.
- Edit/Update a speaker
    - UPDATE SPEAKER
      SET Speaker_number = sNum, Name = sName, Gender = sGender , Nationality = sNationality, Is_jewish = sIsJewish
      WHERE Speaker_number = sNum;
    - Depending on the user input, only some of these updates would take place. It depends on what the user wants to change in the tuple.

Stage V: Tables, Queries, and User Interface

Revisions:

- The populate sql file was edited to include the creation of all the views necessary for the transactions and also to fix the formatting of the transcripts.
- The transaction sql file was edited to remove all of the creation and dropping of views for each transaction. Essentially, before the revision, the views were created and dropped every time the transaction was done, but now, the view is created in the populate file, so the transactions just use those.
- The drop file was edited to include the dropping of all the views for the transactions that were created in the populate file.