

TRENTONIANA DATABASE

Matt Hannum, Kyle Novoa, Alexis Kirk, Chris McKeever, Zach Mataya

GROUP 13

The Trenton library data on Trentoniana is in audio forms on archive.org and is not organized. The data is not user friendly with no transcripts for the files and it is difficult to search for a specific audio file. Additionally, some of the audio files are lengthy and a user is unable to search for a specific part of the file. These files should be accessible in an organized manner and open to the public to easily gain the material for simple and efficient use. There is also no way to manage the files so there also needs to be a way for admins to manage the database.

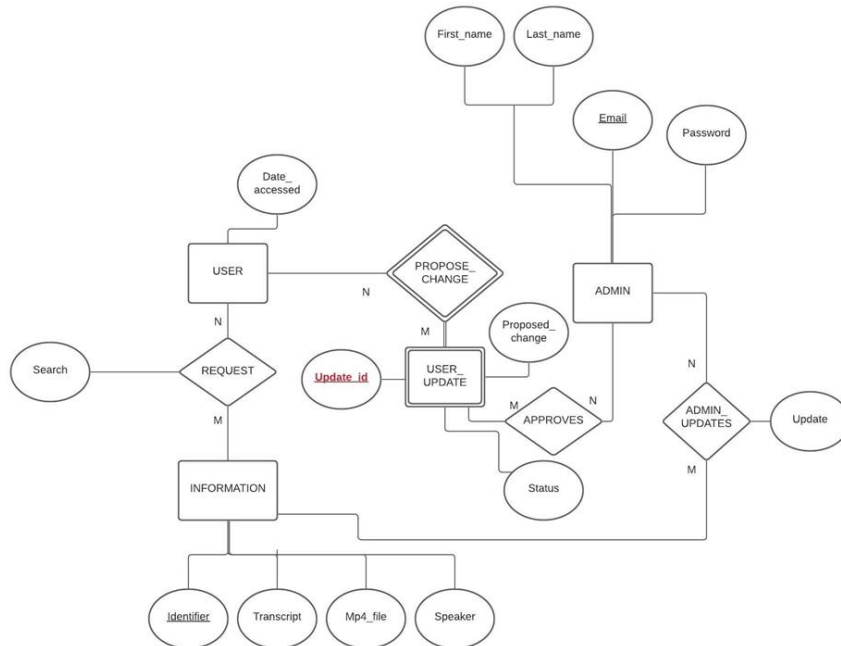
Our goal is to create a database that makes the information from Trentoniana easily accessible to the public. We will use the transcripts and metadata provided to create the information in the database. The database will make it quick and easy for users to find what they are looking for. We will use PostgreSQL for the database, Python and JavaScript for the backend, Flask to make the webserver, and HTML for the frontend.

Our new database will make it easier for users to find the information they are looking for on the Trentoniana website through a better search system. The information will be preserved in the database for people to continue to learn from it in the future and preserve its history. Admins will be able to manage the information stored in the database and update it and add more to it. We will also provide a suggestion area for users to propose changes if they have entries that can be added to the database or notice mistakes in the current entries. Our system will provide a user interface for people to access the data easily and admins to keep it up to date. Our method is better than the current method to access this information. The current way they are stored is unorganized and does not supply transcripts of the audio recordings. Our method makes it easier to find the file people are looking for and provides a transcript to follow along with the recording.

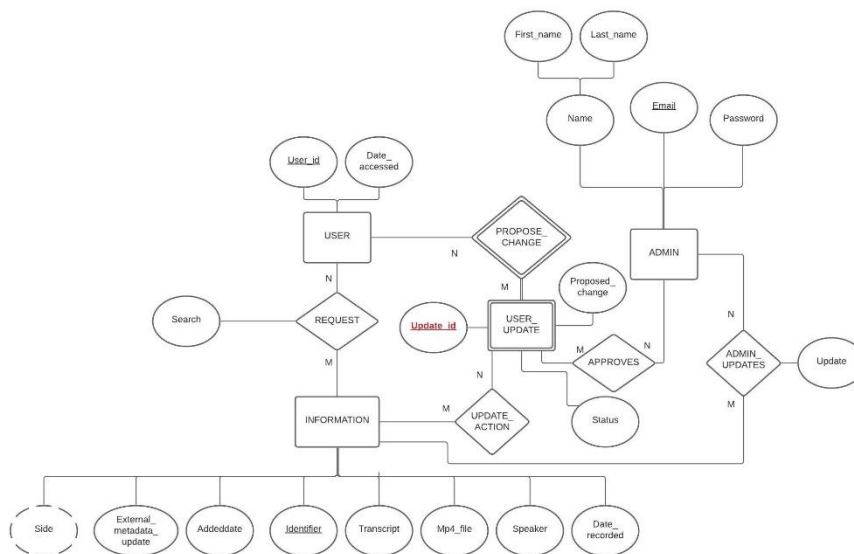
Our database will be low cost to implement. The user interface makes it so anyone can manage the database. There is no computer science required for database management so any of the librarians can manage it. It can be implemented as a separate website that the current library website can link to.

ER Diagram

Final:



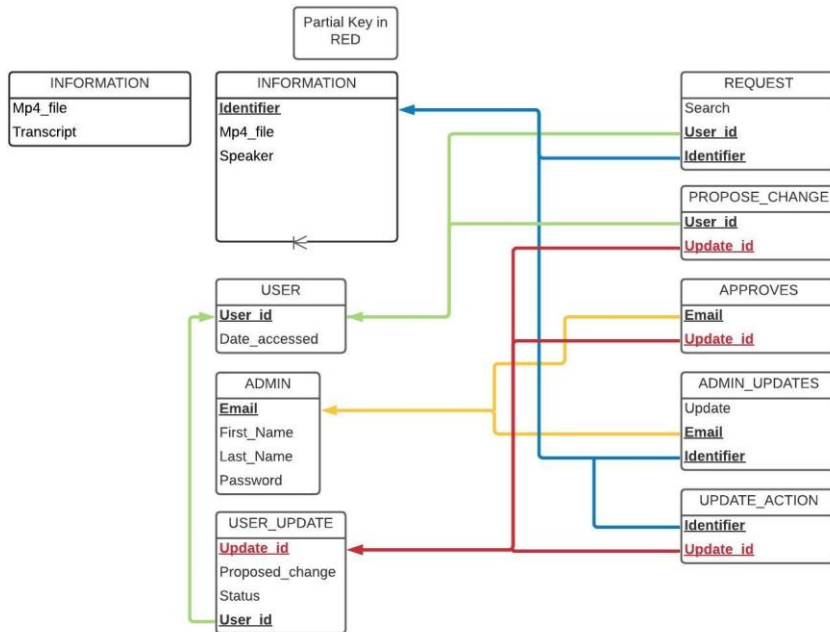
Original:



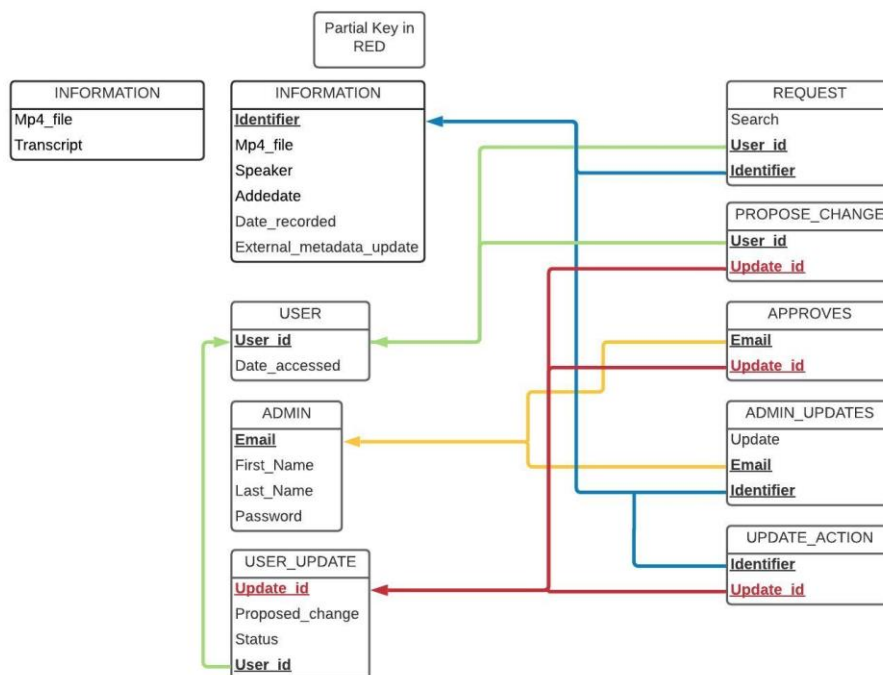
When making the ER diagram into the database, we found some things would be tricky and some of the data was not given that we originally thought was. That is why parts of the database changed. We made updating the user update status not make the change as well and removed some columns from some of the entities.

Relational Schema

Final:



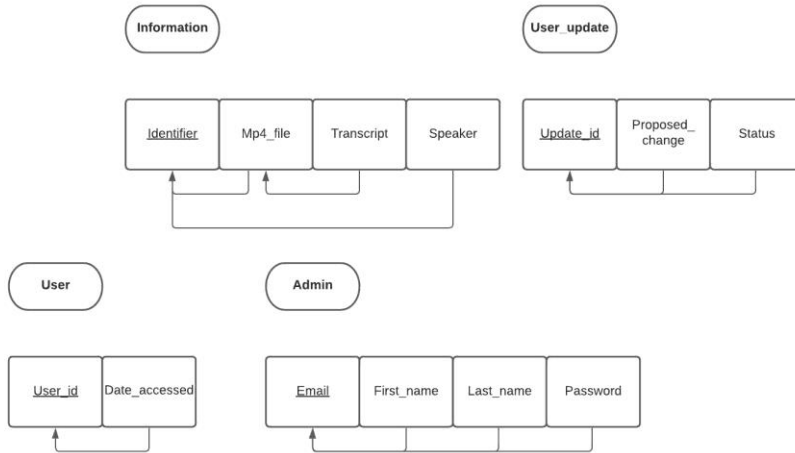
Original:



The relational schema was also updated to match the ER diagram. Both images are in normal form.

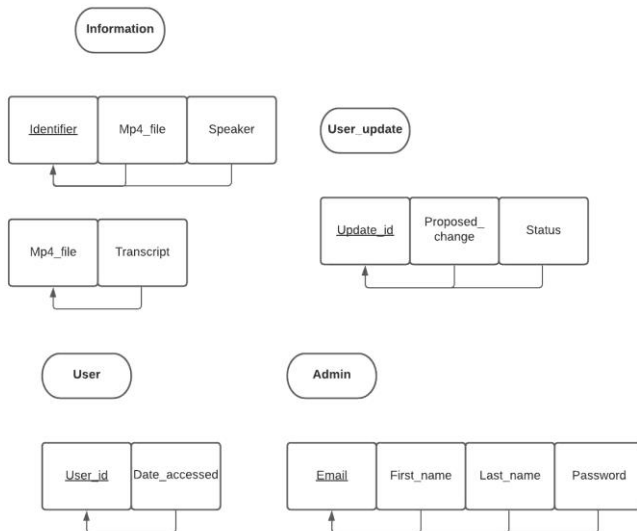
Normalization

Before Normalization:



Information is not in BCNF because of the transitive functional dependency of Transcript to Mp4_file to Identifier. The remaining are all in BCNF in their original state. Whenever FD $X \rightarrow A$ holds in R , then X is a superkey of R .

After Normalization:



Information is split into two places, each with a single functional dependency.

Different Views

User:

1. Search for info using different searching methods
2. Propose changes to the database

Admin:

1. Can change the database by updating current info or add new entries
2. Can view user suggestions and approve or deny them

Transactions

User:

1. Search based on speaker
2. Search based on date
3. Search based on contents
4. Propose a change

Admin:

1. Add a new entry
2. Update an existing entry
3. Delete an entry
4. Approve or deny a change

SQL Queries

```
CREATE TABLE INFORMATION (  
    Identifier PRIMARY KEY,  
    Transcript text,  
    Mp4_file,  
    Speaker text  
);
```

Admin can alter data in *INFORMATION*:

Add new entry:

```
INSERT INTO INFORMATION (Identifier, Transcript, Mp4_file, Speaker)  
VALUES (value1, value2, value3, ...);
```

Update existing entry:

```
UPDATE INFORMATION  
SET Transcript = value1, Mp4_file = value2, ...  
WHERE Identifier = "...";
```

Delete entry:

```
DELETE FROM INFORMATION  
WHERE Identifier = "...";
```

```
CREATE TABLE USER_UPDATE (  
    Update_id SERIAL PRIMARY KEY,  
    Status text,  
    Proposed_change text  
);
```

User can propose a change:

```
INSERT INTO USER_UPDATE (Proposed_change)  
VALUES (value1);
```

Admin can then approve or deny the change:

```
UPDATE USER_UPDATE  
SET Status = value1  
WHERE Update_id = "...";
```

When given a Search under a category:**Searching for speaker:**

This search method for searching for speaker was NOT implemented into the final product

```
SELECT Speaker, Date_recorded  
FROM Information  
WHERE metaphone (Speaker, 6) = metaphone('Search', 6);
```

This search method for searching for speaker was implemented into the final product

```
SELECT *  
FROM Information  
WHERE sound_name ILIKE 'Garfing'
```

Creating and updating the user and admin:

```
CREATE TABLE ADMIN (  
    Email PRIMARY KEY,  
    Name text,  
    Password text  
);
```

```
INSERT INTO ADMIN (Email, Name, Password)  
VALUES (value1, value2, value3);
```

```
UPDATE ADMIN
SET Email = value1, Name = value2, Password = value3,
WHERE Email = "...";
```

```
DELETE FROM ADMIN
WHERE Email = "...";
```

```
CREATE TABLE USER (
    User_id SERIAL PRIMARY KEY,
    Date_accessed DATE
);
```

```
INSERT INTO USER (Date_accessed)
VALUES (value1);
```


User Interface

Home menu.

Trentoniana Database Search Tool

Enter a search term and then select the catagory you would like to search in!

Search By JHS Name (Ex: "JHS 12") ☐

Search By Name (Ex: "Barker") ☐

View All Entries ☐

Propose Change

If there are any mistakes in the data or entries you wish to be entered, please propose your ideas here.

Admin Login

User Name

Password

Results when view all entries selected.

JHS 12	Barker, Maynard (JHS12)	https://archive.org/details/JHS05SideA	https://trentonlib.org/wp-content/uploads/2021/02/JHS-12-Barker.pdf
JHS 17	Axelrod, Ron (JHS17)	https://archive.org/details/JHS17SideA	https://trentonlib.org/wp-content/uploads/2021/02/JHS-16-Garfing.pdf
JHS 16	Garfing, Arthur (JHS16)	https://archive.org/details/JHS16SideA	https://trentonlib.org/wp-content/uploads/2021/02/JHS-16-Garfing.pdf

Result from submitting as suggested change.

12	waiting	Mistake in JHS 24
----	---------	-------------------

New Search

Admin Page: Admins can update, insert, and delete entries from the dataset. They can also view the users, user changes, and other admins. They can update the status on suggested updates and insert and delete admins. In the python code, it checks that the admin inserts entries that do not already exist, provide a valid primary key for the entry they wish to update, and a valid primary key for the entry they wish to delete. This prevents users from breaking the database.

Update

Fill out the following information to update a database entry. Name is a required field. Only fill in parts that you wish to update.

JHS Name *REQUIRED*

Sound File Name

Link

Transcript

Insert

Fill out the following information to insert a new database entry. Name is a required field. Only fill in parts that you wish to insert.

JHS Name *REQUIRED*

Sound File Name

Link

Transcript

Delete

Enter the name of the entry you would like to delete.

JHS Name *REQUIRED*

View Users

Enter the date you want to see the users for

Date format "yyyy-mm-dd"

View all users ☐

View User Suggestions

Search based on suggestion ID

View all entries ☐

View only verified ☐

View only waiting ☐

View only denied ☐

Update Suggestions Status

Suggestion ID

Set status to verified ☐

Set status to waiting ☐

Set status to denied ☐

View admins

Search by email

View all admins ☐

Add admins

Email

First Name

Last Name

Password

Admin viewing suggstions by status.

3	waiting	Broken link on JHS 17
5	waiting	Mistake in JHS 12
6	waiting	Mistake in JHS 13
7	waiting	Mistake in JHS 12
8	waiting	Mistake in JHS 14
9	waiting	Mistake in JHS 15
11	waiting	Mistake in JHS 23
12	waiting	Mistake in JHS 24

[Back to Admin Page](#)