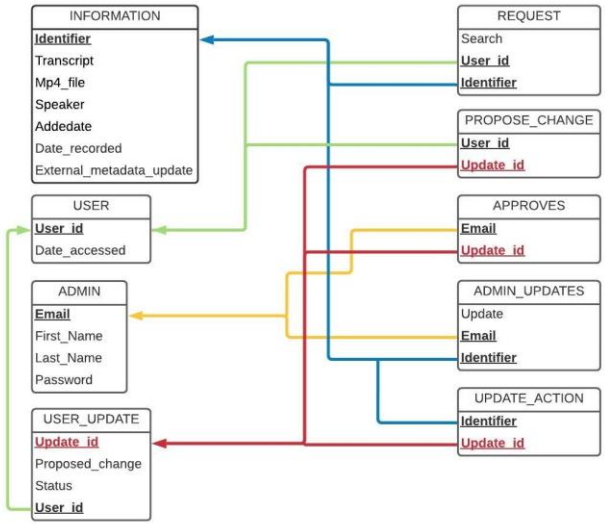
Stage 4

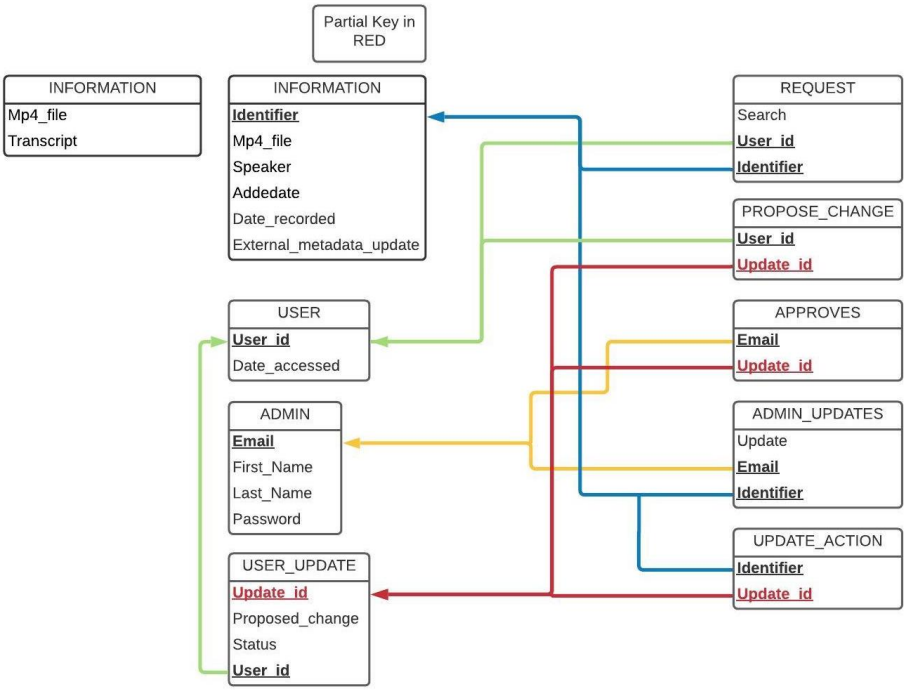
Group 13

Normalization

Original



Boyce Codd Normal Form



Transcript is dependent on Mp4_file which is dependent on Identifier so it does not meet the requirement of no transitive dependencies. The other tables all elements are dependent on their primary key or are foreign keys.

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,
      External_metadata_update DATE,
      Addeddate DATE,
      Transcript text,
      Mp4_file,
      Speaker text,
      Date_recorded DATE
);
```

Admin can alter data in INFORMATION:

Add new entry:

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

Update existing entry:

```
UPDATE INFORMATION
SET Transcript = value1, External_metadata_update = 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:

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

Searching based on date:

```
SELECT Speaker, Date_recorded
FROM Information
WHERE Date_recorded = “Search”;
```

Searching based on contents:

```
SELECT Speaker, Date_recorded
FROM Information
WHERE Transcript ILIKE Search%;
```

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);
```