Benjamin Tate
8/7/2016
CS340 -- Section 400

Final Project -- Diagrams and Written Info

Outlines:

Outline:

My database is about something that has become known as "The 27 Club," which is the large number of popular and/or influential musicians who died at age twenty-seven. On the scale of this project, it might not be the most telling of databases, but with the somewhat eerie scale of the 27 club in its entirety, it would be interesting to see what other connections might link its members. It is relatively easy to go to Wikipedia's entry on the 27 club and skim over the list of members for things like cause of death or exact age, so I decided to have my database delve into dots that are a little harder to connect, such as place of birth, genre of music, instrument(s) of choice, and who influenced whom. Again, due to time constraints, the starting scale of this database is very small compared to what it could be, so a lot of connections will likely not be apparent.

Database Outline in Words:

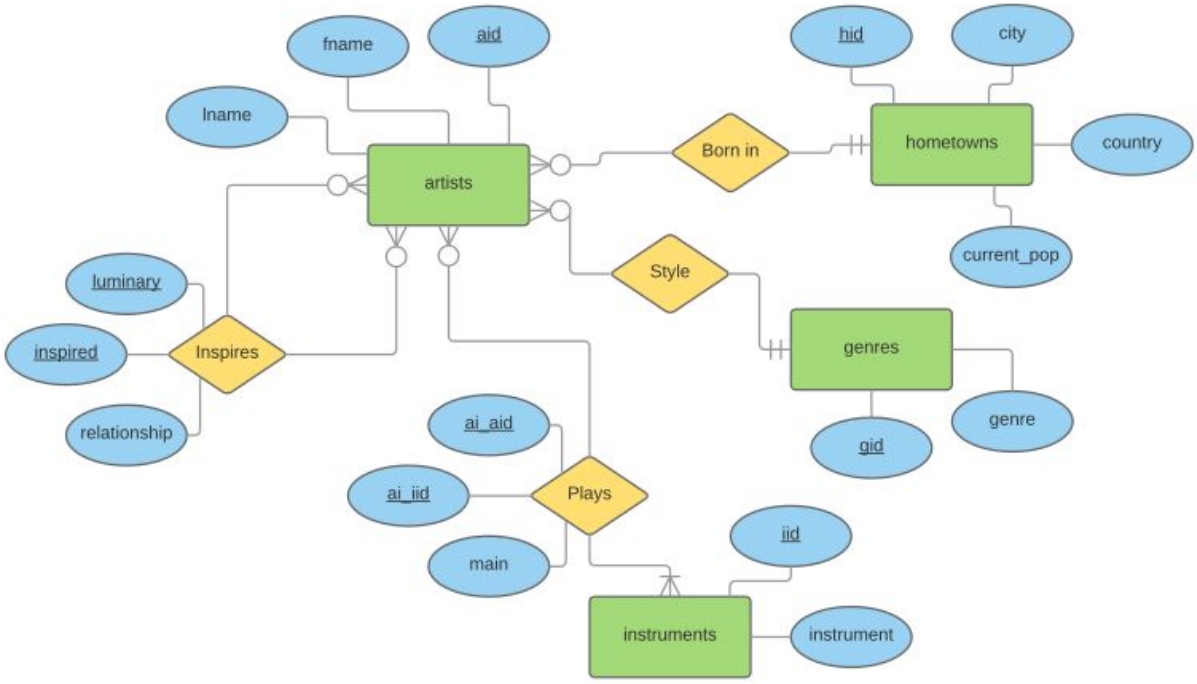
This database has four entities. The main one is the artists, so naturally they are sort of the central hub of the database structure. Each artist has a unique first and last name combination as well as references to their hometown, which includes city (or city, state), country, and current population (as finding the populations when the artists lived there would have yielded unbalanced views of the scales), and primary genre. Since so many artists dabbled in multiple genres, and may have had a style that was hard to fit into an established genre, I apted to simplify things by using only what I felt was the main, "umbrella" genre for each artist (e.g. all subsets of rock = rock). Because of this, artists' relationships to both hometowns and genres are many-to-one. The instruments entity is very simple, with just the name of the instrument as an attribute, but its relationship to artists is more complex. Since most artists on the list seem to play at least two instruments, I felt that all of them should be represented, so this is a many-to-many relationship. Included in the art_inst table is a boolean representing whether or not the given instrument is a "main" of the given artist. One artist can have multiple main instruments (i.e. guitar and vocals). The last relationship is between artists and other artists. This is meant to represent when one artist, the 'luminary,' inspired another artist, the 'inspired.' In this table is an optional relationship value to represent significant relationships such as

family ties or direct mentorship. The mere fact that this inspiration relationship exists is enough to represent musical influence, so if that is as deep as the connection goes, this field can be left null. The inspirations relationship is also many-to-many, as an artist can inspire many, and can be inspired by many.

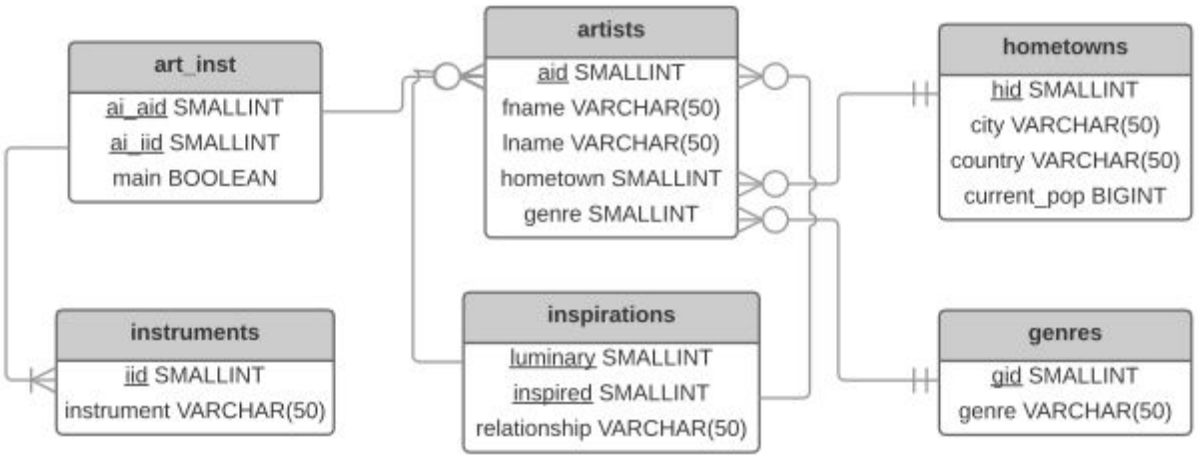
Constraints include the fact that an artist's first/last name combination must be unique, genre and instrument names must be unique, and city/country combinations must be unique for a given hometown (in addition to the unique ids that each table automatically creates). Because of this all of these fields must not be left blank. A row in the artists table must also not fail to reference a member of the hometowns and genres tables, as this would defeat the purpose of the entry. Lastly, each artist/artist combination in the inspirations table and each artist/instrument combination in the instruments table must exist only once to avoid conflicting data. Because of this, these combinations are the primary keys of their respective tables.

Diagrams:

ER Diagram of Database:



Database Schema:



Queries:

Table Creation Queries:

```
DROP TABLE IF EXISTS inspirations;
DROP TABLE IF EXISTS art_inst;
DROP TABLE IF EXISTS artists;
DROP TABLE IF EXISTS instruments;
DROP TABLE IF EXISTS genres;
DROP TABLE IF EXISTS hometowns;

CREATE TABLE hometowns (
    hid SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    city VARCHAR(50) NOT NULL,
    country VARCHAR(50) NOT NULL,
    current_pop BIGINT UNSIGNED,
    PRIMARY KEY (hid),
    UNIQUE KEY (city, country)
)ENGINE=InnoDb DEFAULT CHARSET=utf8;

CREATE TABLE genres (
    gid SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    genre VARCHAR(50) NOT NULL,
    PRIMARY KEY (gid)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE instruments (
    iid SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    instrument VARCHAR(50) NOT NULL,
    PRIMARY KEY (iid),
    UNIQUE KEY (instrument)
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE artists (
    aid SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    fname VARCHAR(50) NOT NULL,
    lname VARCHAR(50) NOT NULL,
    hometown SMALLINT UNSIGNED NOT NULL,
    genre SMALLINT UNSIGNED NOT NULL,
    PRIMARY KEY (aid),
    UNIQUE KEY (fname, lname),
    KEY idx_fk_hometown (hometown),
    KEY idx_fk_genre (genre),
    CONSTRAINT `fk_hometown` FOREIGN KEY (hometown) REFERENCES hometowns (hid)
ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT `fk_genre` FOREIGN KEY (genre) REFERENCES genres (gid) ON DELETE
RESTRICT ON UPDATE CASCADE
```

```
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE art_inst (
    ai_aid SMALLINT UNSIGNED NOT NULL,
    ai_iid SMALLINT UNSIGNED NOT NULL,
    main BOOLEAN NOT NULL DEFAULT FALSE,
    PRIMARY KEY (ai_aid, ai_iid),
    CONSTRAINT `fk_art_inst_ai_aid` FOREIGN KEY (ai_aid) REFERENCES artists
(aid) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT `fk_art_inst_ai_iid` FOREIGN KEY (ai_iid) REFERENCES instruments
(iid) ON DELETE RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE inspirations (
    luminary SMALLINT UNSIGNED NOT NULL,
    inspired SMALLINT UNSIGNED NOT NULL,
    relationship VARCHAR(50),
    PRIMARY KEY (luminary, inspired),
    CONSTRAINT `fk_inspirations_art1` FOREIGN KEY (luminary) REFERENCES artists
(aid) ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT `fk_inspirations_art2` FOREIGN KEY (inspired) REFERENCES artists
(aid) ON DELETE RESTRICT ON UPDATE CASCADE
)ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

General Use Queries:

Adding:

```
--ARTISTS INSERT
    INSERT INTO artists (fname,lname,hometown,genre)
    VALUES ([artFNameInput],[artLNameInput],[artHomeInput],[artGenInput])

--GENRE INSERT
    INSERT INTO genres (genre) VALUES ([genNameInput])

--HOMETOWN INSERT
    INSERT INTO hometowns (city,country,current_pop)
    VALUES ([homeCityInput],[homeCountryInput],[currentPopInput])

--INSTRUMENT INSERT
    INSERT INTO instruments (instrument) VALUES ([instNameInput])

--ART/INST INSERT
    INSERT INTO art_inst (ai_aid,ai_iid,main)
    VALUES ([artInstArtInput],[artInstInstInput],[artInstMainInput])

--INSPIRATION INSERT
    INSERT INTO inspirations (luminary,inspired,relationship)
    VALUES ([inspLumInput],[inspInspInput],[inspRelInput])
```

Searching:

```
--NAME, HOMETOWN, & GENRE
    SELECT fname,lname,city,country,genre
    FROM (
        SELECT *
        FROM genres g
        INNER JOIN (
            SELECT aid, fname, lname, hometown, genre AS genreId
            FROM artists a
            WHERE a.aid = [searchArtInput]
        ) AS tbl1 ON tbl1.genreId = g.gid
        INNER JOIN (
            SELECT *
            FROM hometowns
        ) AS h ON h.hid = tbl1.hometown
    ) AS tbl2

--INSTRUMENT COUNT
    SELECT COUNT(instrument) AS instCount
    FROM (
        SELECT * FROM artists a
        INNER JOIN (
            SELECT *
            FROM art_inst
        ) AS ai ON ai.ai_aid = a.aid
        INNER JOIN (
            SELECT *
            FROM instruments
        ) AS i ON ai.ai_iid = i.iid
    ) AS tbl
    WHERE tbl.aid = [searchArtInput]
    GROUP BY fname

--MAIN INSTRUMENTS
    SELECT instrument,main
    FROM (
        SELECT * FROM artists a
        INNER JOIN (
            SELECT *
            FROM art_inst
        ) AS ai ON ai.ai_aid = a.aid
        INNER JOIN (
            SELECT *
            FROM instruments
```

```
        ) AS i ON ai.ai_iid = i.iid
    ) AS tbl
    WHERE tbl.aid = [searchArtInput]
    AND tbl.main = '1'

--SECONDARY INSTRUMENTS
    SELECT instrument,main
    FROM (
        SELECT * FROM artists a
        INNER JOIN (
            SELECT *
            FROM art_inst
        ) AS ai ON ai.ai_aid = a.aid
        INNER JOIN (
            SELECT *
            FROM instruments
        ) AS i ON ai.ai_iid = i.iid
    ) AS tbl
    WHERE tbl.aid = [searchArtInput]
    AND tbl.main = '0'

--INSPIRATION TO COUNT
    SELECT COUNT(iaid) AS lumCount
    FROM (
        SELECT aid AS laid, fname AS lfname, lname AS llname, relationship,
iaid, ifname, ilname
        FROM artists l
        INNER JOIN (
            SELECT * FROM inspirations insp
            INNER JOIN (
                SELECT aid AS iaid, fname AS ifname, lname AS ilname
                FROM artists
            ) AS i ON insp.inspired = i.iaid
        ) AS tbl1 ON tbl1.luminary = l.aid
        WHERE tbl1.luminary = [searchArtInput]
    ) AS tbl2 GROUP BY lfname

--INSPIRATION TO
    SELECT ifname,ilname
    FROM (
        SELECT aid AS laid, fname AS lfname, lname AS llname, relationship,
iaid, ifname, ilname
        FROM artists l
        INNER JOIN (
            SELECT * FROM inspirations insp
            INNER JOIN (
                SELECT aid AS iaid, fname AS ifname, lname AS ilname
                FROM artists
```

```
            ) AS i ON insp.inspired = i.iaid
        ) AS tbl1 ON tbl1.luminary = l.aid
        WHERE tbl1.luminary = [searchArtInput]
    ) AS tbl2 GROUP BY lfname

--INSPIRED BY COUNT
    SELECT COUNT(laid) AS inspCount
    FROM (
        SELECT aid AS laid, fname AS lfname, lname AS llname, relationship,
iaid, ifname, ilname
        FROM artists l
        INNER JOIN (
            SELECT * FROM inspirations insp
            INNER JOIN (
                SELECT aid AS iaid, fname AS ifname, lname AS ilname
                FROM artists
            ) AS i ON insp.inspired = i.iaid
        ) AS tbl1 ON tbl1.luminary = l.aid
        WHERE tbl1.inspired = [searchArtInput]
    ) AS tbl2 GROUP BY ifname

--INSPIRED BY
    SELECT COUNT(laid) AS inspCount
    FROM (
        SELECT aid AS laid, fname AS lfname, lname AS llname, relationship,
iaid, ifname, ilname
        FROM artists l
        INNER JOIN (
            SELECT * FROM inspirations insp
            INNER JOIN (
                SELECT aid AS iaid, fname AS ifname, lname AS ilname
                FROM artists
            ) AS i ON insp.inspired = i.iaid
        ) AS tbl1 ON tbl1.luminary = l.aid
        WHERE tbl1.inspired = [searchArtInput]
    ) AS tbl2 GROUP BY ifname
```