Tony Nguyen

Dr. Shawn Bowers

CPSC 321 01

26 September 2023

Homework 2

1. Create relational schemas.

    a) Album(<u>title</u>, year, <u>group</u>, record_label)

    - Primary key: (title, group)
    - Foreign key:
        - Album.group REFS MusicGroup.name
        - Album.record_label REFS RecordLabel.name

    b) Track(year, <u>track_name</u>)

    - Primary key: track_name

    c) Song(<u>song_title</u>, year, track_id)

    - Primary key: song_title
    - Foreign key: Songs.track_id REFS Track.track_name
    - Candidate key: track_id

    d) MusicGroup(<u>name</u>, year_formed)

    - Primary key: name

    e) Artist(<u>name</u>, birth_year)

    - Primary key: name

    f) RecordLabel(<u>name</u>)

    - Primary key: name

g) Genre(label, description)

- Primary key: label
- Candidate key: description

h) Album&Track(alb_title, track_id, group_name)

- Primary key: (alb_title, track_id)
- Foreign key:
  - Album&Track.alb_title REFS Album.title
  - Album&Track.track_id REFS Track.track_name
  - Album&Track.group_name REFS MusicGroup.name

i) Song&Artist(song_title, artist_name)

- Primary key: (song_title, artist_name)
- Foreign key:
  - Song&Artist.song_title REFS Song.song_title
  - Song&Artist.artist_name REFS Artist.name

j) Group&Genre(group_name, genre_label)

- Primary key: (group_name, genre_label)
- Foreign key:
  - Group&Genre.group_name REFS MusicGroup.name
  - Group&Genre.genre_label REFS Genre.label

k) Group&Influencer(group_name, influenced_by)

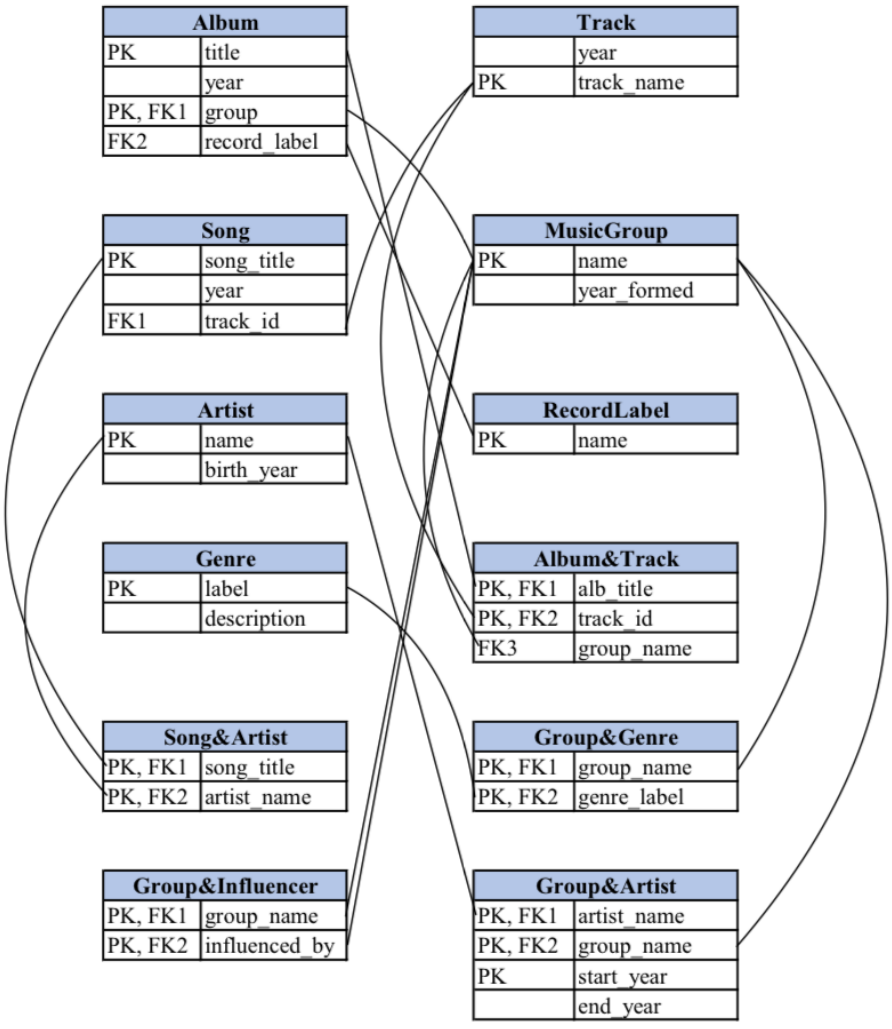
- Primary key: (group_name, influenced_by)
- Foreign key:
  - Group&Influence.group_name REFS MusicGroup.name

- Group&Influence.influenced_by REFS MusicGroup.name

l) Group&Artist(<u>artist_name, group_name, start_year</u>, end_year)

- Primary key: (artist_name, group_name, start_year)
- Foreign key:
  - Group&Artist.artist_name REFS Artist.name
  - Group&Artist.group_name REFS MusicGroup.name

2. Schema Diagram



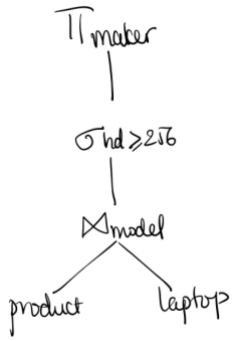
3. Schema Design Reflection

- The current schema design is pretty good. It depicts the schemas, as well as their attributes. Furthermore, “bridging” schemas show the relationship and their constraints.
- For example, consider Taylor Swift’s folklore album:
    - Album(folklore, 2020, Taylor Swift, Republic Records)
        - Republic Records refers to RecordLabel.
        - Taylor Swift, here being considered under MusicGroup, as of the constraint. From here, we can know which genre she sings, thanks to the Group&Genre table, and who she was influenced by using the Group&Influencer table. If this is a band, we will know its members thanks to the Group&Artist schema.
        - The Album&Track schema connects the album folklore to Taylor Swift (under MusicGroup) and the Track table.
        - The Song schema connects the songs within folklore to the track, which is connected to the Artist table (so that, for example, Aaron Dessner can be included in folklore as he is not a member of Taylor Swift’s group – and she is doing solo as well) thanks to the Song&Artist schema.
    - However, there are some possible redundancies. Since we cannot use any surrogate key, there are some possible assumptions that we need to make. And because of that, more “bridging” table is required.

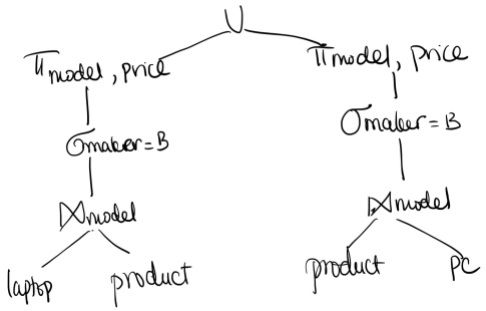
- Track_name is used across multiple tables: Track, Song, Album&Track.
- Group_name is used across Album, Group&Influencer, MusicGroup, Album&Track, Group&Genre, Group&Artist. I imagine if we can use surrogate keys here, we can reduce the amount of group_name, especially since we don't have to include many attributes to create primary key combinations.

4. Relational Algebras

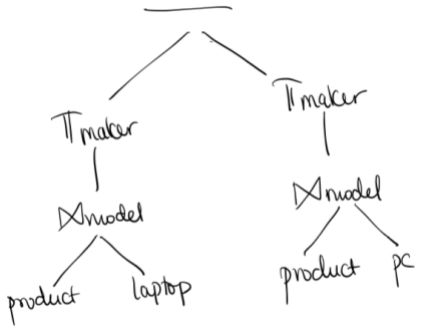
a) $\pi_{maker}(\sigma_{hd\geq 256}(product \bowtie_{model} laptop))$



b) $\pi_{model,price}(\sigma_{maker=B}(product \bowtie_{model} laptop)) \cup \pi_{model,price}(\sigma_{maker=B}(product \bowtie_{model} pc))$

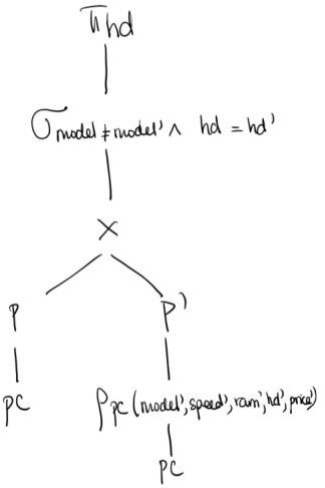


c) $\pi_{maker}(product \bowtie_{model} laptop) \setminus \pi_{maker}(product \bowtie_{model} pc)$



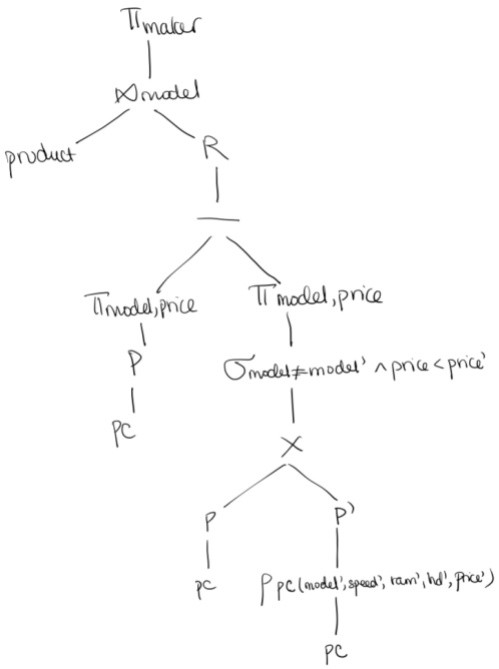
d)

- $P' \coloneqq \rho_{pc(model',speed',ram',hd',price')}(pc)$
- $P \coloneqq pc$
- $\pi_{hd}\left(\sigma_{model \neq model' \wedge hd=hd'}(P \times P')\right)$



e)

- $P' \coloneqq \rho_{pc(model',speed',ram',hd',price')}(pc)$
- $P \coloneqq pc$
- $R \coloneqq \pi_{model,price}(P) \setminus \pi_{model,price}\left(\sigma_{model \neq model' \wedge price<price'}(P \times P')\right)$
- $\pi_{maker}(product \bowtie_{model} R)$


Π maker
⋈ model
product
R
—
Π model, price
Π model, price
P
σ model≠model' ∧ price<price'
PC
×
P
P'
pc
P pc(model', speed', ram', hd', price')
PC