

Here are some relations that exist in a database for a symphony.

Person(email, name, age)

- This relation stores anyone who has signed up for our mailing list. Tuples in this relation may not be listed in Purchase. There may be tuples in this relation that are not listed in Purchase (i.e., there may be some people who do not make purchases).

Show(id, year, month, date, showing, attendanceNumber)

- Showing describes whether a show was during morning, afternoon, or evening

Song(composer, title)

SongsPerformed(showID, composer, title)

- showID is a foreign key referring to Show
- composer and title are foreign keys referring to attributes of the same name in Song

Purchase(email, showID, price)

- email is a foreign key referring to the email attribute in Person
- showID is a foreign key referring to Show

Musician(id, name, instrument, position, nationality)

PerformedIn(id, showID)

- id refers to the attribute of the same name in Musician
- showID is a foreign key referring to Show

Write relational algebra statements to answer the following questions:

1. Find all the musicians who play the piano or the violin.

$$\sigma_{instrument=piano \vee instrument=violin}(Musician)$$

2. Find the email addresses of all the people who have signed up for our mailing list who did not purchase a ticket to a show.

$$\pi_{email}(Person) - \pi_{email}(Purchase)$$

3. Find the songs which have been performed exactly once.

$\text{RepeatedSongs} \leftarrow \pi_{sp1.showID, sp1.composer, sp1.title} \left(\sigma_{\text{Note 1}} \left(\rho(sp1, \text{SongsPerformed}) \times \rho(sp2, \text{SongsPerformed}) \right) \right)$

$\text{SongsPerformed} - \text{RepeatedSongs}$

Note 1: $sp1.showID \neq sp2.showID \wedge sp1.composer = sp2.composer \wedge sp1.title = sp2.title$

More About Question 3

Confused about why RepeatedSongs contains all the songs that have been played more than once? Let's look at an example.

Songs(composer, title)

Composer	Title
Michelle	A song
Rachel	A different song
Jessica	Some other song

SongsPerformed(showID, composer, title)

ShowID	Composer	Title
1	Michelle	A song
2	Michelle	A song
2	Rachel	A different song

SongsPerformed x SongsPerformed

1	2	3	4	5	6
1	Michelle	A song	1	Michelle	A song
2	Michelle	A song	1	Michelle	A song
2	Rachel	A different song	1	Michelle	A song
1	Michelle	A song	2	Michelle	A song
2	Michelle	A song	2	Michelle	A song
2	Rachel	A different song	2	Michelle	A song
1	Michelle	A song	2	Rachel	A different song
2	Michelle	A song	2	Rachel	A different song
2	Rachel	A different song	2	Rachel	A different song

Only keep the tuples where all of the following conditions are met:

- The value in column 2 == the value in column 5
- The value in column 3 == the value in column 6
- The value in column 1 IS NOT EQUAL to the value in column 4

1	2	3	4	5	6
1	Michelle	A song	1	Michelle	A song
2	Michelle	A song	1	Michelle	A song
2	Rachel	A different song	1	Michelle	A song
1	Michelle	A song	2	Michelle	A song
2	Michelle	A song	2	Michelle	A song
2	Rachel	A different song	2	Michelle	A song
1	Michelle	A song	2	Rachel	A different song
2	Michelle	A song	2	Rachel	A different song
2	Rachel	A different song	2	Rachel	A different song

At this point, we have all the songs that have been played two or more times.

The projection tells us to only keep columns 1, 2, and 3. Keeping all three of these columns makes it so that we can easily do the set difference with SongsPerformed since the schemas will match.

RepeatedSongs(showID, composer, title)

ShowID	Composer	Title
1	Michelle	A song
2	Michelle	A song

SongsPerformed – RepeatedSongs

ShowID	Composer	Title
2	Rachel	A different song

One final note: because we have made it so that we are using \neq in our predicate that is checking on the IDs of the shows, we are okay to compare the showIDs – we are guaranteed that every duplicate will appear in RepeatedSongs, not just one. However, if instead you used a different predicate, like $<$, then you would not be guaranteed that, and you would need to project it out before you did the comparison. So be careful!