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.

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 the emails of the people who have attended a show in January 2020 and February 2020.

   ShowsInJan2020 ←
   $\pi_{email}(\sigma_{year=2020 \wedge month=January} Show \bowtie_{Show.id=Purchase.showID} Purchase)$

   ShowsInFeb2020 ←
   $\pi_{email}(\sigma_{year=2020 \wedge month=February} Show \bowtie_{Show.id=Purchase.showID} Purchase)$

   $$ShowsInJan2020 \cap ShowsInFeb2020$$

   You can't use
   $\pi_{email}(\sigma_{year=2020 \wedge (month=\text{January} \vee month=\text{February})} Show \bowtie_{Show.id=Purchase.showID} Purchase)$
   as that will find everyone who has attended a show in January **and/or** February and won't be able to remove people who have only attended a show in one of those months.

2. What songs were performed in the shows with the most attendance? In the event of a tie, list all the songs that were performed in the shows.

$$\text{EverythingButMostAttendance} \leftarrow \pi_{s1.id}\left(\sigma_{6<12}(\rho(s1, Show) \times Show)\right)$$

$$\text{HighestAttendance} \leftarrow \pi_{id}(Show) - EverythingButMostAttendance$$

$$SongsPerformed \bowtie_{SongsPerformed.showID=HighestAttendance.id} HighestAttendance$$

Wondering what the selection condition in EverythingButMostAttendance is trying to say? The numbers refer to the column index (where the first column has an index of 1). Whenever we do a cross product, columns are renamed by position if there is a name class. In this case, all the columns in the two Show instances have a name clash so all the columns were renamed to their column index instead.

(See the "More About Question 2" section below for a more detailed explanation.)

3. Find the email addresses of people who have purchased a ticket for every performance that includes a piece composed by Tchaikovsky.

$$\pi_{email,showID}(Purchase) / \pi_{showID}\left(\sigma_{composer=Tchaikovsky}(SongsPerformed)\right)$$

More About Question 2

If you are wondering why we need to create the EverythingButMostAttendance relation, it is because we want to find all the shows that are NOT the show(s) with the most attendance. Consider the following relational instances:

Show(id, attendance) [There are more attributes in Show but for brevity we only show the important ones]

Id	Attendance
1	10
2	5
3	10
4	8

What does $\rho(s1, Show) \times \rho(s2, Show)$ give us? It gives us the following:

S1.Id	S1.Attendance	S2.id	S2.Attendance
1	10	1	10
2	5	1	10
3	10	1	10
4	8	1	10
1	10	2	5
2	5	2	5
3	10	2	5
4	8	2	5
1	10	3	10
2	5	3	10
3	10	3	10
4	8	3	10
1	10	4	8
2	5	4	8
3	10	4	8
4	8	4	8

Let's process $\sigma_{s1.attendance<s2.attendance}\big(\rho(s1, Show) \times \rho(s2, Show)\big)$. Why do we need to do this? It's because we want to find all the shows where there is AT LEAST one show with a larger attendance number. I.e., we want to find all the shows that do not have the largest attendance.

S1.Id	S1.Attendance	S2.id	S2.Attendance
~~1~~	~~10~~	~~1~~	~~10~~
2	5	1	10
~~3~~	~~10~~	~~1~~	~~10~~
4	8	1	10
~~1~~	~~10~~	~~2~~	~~5~~
~~2~~	~~5~~	~~2~~	~~5~~
~~3~~	~~10~~	~~2~~	~~5~~
~~4~~	~~8~~	~~2~~	~~5~~
~~1~~	~~10~~	~~3~~	~~10~~
2	5	3	10
~~3~~	~~10~~	~~3~~	~~10~~
4	8	3	10
~~1~~	~~10~~	~~4~~	~~8~~
2	5	4	8
~~3~~	~~10~~	~~4~~	~~8~~
~~4~~	~~8~~	~~4~~	~~8~~

Now, we do a projection on the result to only save s1.id. Since projection is a set operation, duplicates are automatically removed.

EverythingButMostAttendance

S1.Id
2
4

These are the show IDs of shows that did NOT have the maximum number of attendees.

$\pi_{id}(Show)$ will return:

Id
1
2
3
4

$\pi_{id}(Show) - EverythingButMostAttendance$

Id
1
3

These are the show IDs of the most attended shows!