Sql 1 answers - in-class exercise

Introduction To Relational Databases (The University of British Columbia)

Here are some relations that exist in a database for an orchestra.

email address

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, day, showing, attendanceNumber)

- Showing describes whether a show was during morning, afternoon, or evening
- {year, month, day, showing} is a candidate key for Show

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 SQL statements to answer the following questions:

1. Find the email addresses of people who attended a show in January 2019. If you have multiple conditions in your WHERE clause, connect them with an AND (e.g., WHERE year > 1939 AND movieID > 2).

   Do you need DISTINCT? Why or why not?

   Solution:

   Yes, we do need DISTINCT to prevent from selecting someone who attended more than one show in January 2019.

```
SELECT DISTINCT pur.email
FROM Purchase AS pur, Show AS S
WHERE S.id = pur.showID AND S.year = 2019 AND S.month = 'January'
```

2. Find the year, month, day, and attendance numbers of all shows that had a Canadian musician perform in it.

3. Find the names of all musicians who play the flute.
   Do you need DISTINCT? Why or why not?

Question 2

Solution:

```
SELECT DISTINCT S.year, S.month, S.day, S.attendenceNumber
FROM Show AS S, Musician AS M, PerformedIN AS P
WHERE M.nationality = 'Canadian' AND S.id = P.showID AND P.id = M.id
```

Question 3

Solution:

No, there is no need for DISTINCT here, because there does exist a situation where there are 2 people whose names are identical playing the flute, which is possible and reasonable.

```
SELECT M.name
FROM Musician AS M
WHERE M.instrument = 'flute'
```

Sql 2 answers - in-class exercise

Introduction To Relational Databases (The University of British Columbia)

Here are some relations that exist in a database for an orchestra.

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, day, showing, attendanceNumber)
- Showing describes whether a show was during morning, afternoon, or evening
- {year, month, day, showing} is a candidate key for Show

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 SQL statements to answer the following questions:

1. Use the INTERSECT operator for the following question.

   Find the showIDs of shows where the symphony performed songs by Mozart and Beethoven.
   Another way to think about this question: Find the shows where at least one song composed by Mozart and at least one song composed by Beethoven were performed.

Solution:

```
SELECT SP1.ShowID
FROM Songsperformed AS SP1
WHERE SP1.composer = 'Mozart'
INTERSECT
SELECT SP2.ShowID
FROM Songsperformed AS SP2
WHERE SP2.composer = 'Beethoven'
```

2. Write a query to solve question 1 but this time, do not use the INTERSECT query. If this is not possible, explain why.

Solution:

3. Write a query to solve question 1 with the EXISTS/NOT EXISTS operator. If this is not possible, explain why.

4. Write a query to solve question 1 with the IN/NOT IN operator. If this is not possible, explain why.

Sql 4 in-class exercise

Introduction To Relational Databases (The University of British Columbia)

Here are some relations that exist in a database for an orchestra.

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, day, showing, attendanceNumber)

- Showing describes whether a show was during morning, afternoon, or evening
- {year, month, day, showing} is a candidate key for Show

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 SQL statements to answer the following questions:

1. Find the total number of attendees for each day there was a show. Don't forget that a single day can have multiple shows!

2. Find the total number of Canadian musicians per instrument.

3. For all the shows where the audience's average age is greater than the average age of our mailing list, what songs were performed?

)