

2016-07-04 Exam Exercises

The following exercises and their solutions were originally authored by *Professor Maria de Marsico*, for the "Basi di Dati" exam session on July, the 4th, 2016.

Their translation is ongoing.

1. Relational Algebra

Consider a database with the following schema, describing theatrical performances:

```
THEATRE (TheatreID, Name, City)
PLAY (Title, Author)
PERFORMANCE (TheatreID, Date, Title)
```



Multiple theatres, sharing the same name, can be present in different cities. Bologna and Florence both have a "Teatro Comunale", for example.

Write **relational algebra expressions** for the following queries:

1. Find the *names* of the theatres, in **Rome**, where plays by "**Pirandello**" were performed in **2020**.
2. Get the *names* of the plays that were performed **exclusively** in Rome's theatres.

1.1. Answer

We must identify the relations containing the needed data. We require all of them, as:

- **THEATRE** provides both the cities and the theatre names we're after
- **PLAY** holds data concerning the play's author
- **PERFORMANCE** ties theatres and plays together

The easiest query consists of **joining** the three relations and **selecting** those tuples whose city is "Rome" *and* whose actor is "Pirandello" which *also* are within the appropriate date range.



Pay attention to the **joins'** order, so that any two operands share a compatible attribute. In this case **natural joins** will suffice.

Define ***r*** as the relation stemming from:

$$\sigma_{City=Roma \wedge Author=Pirandello \wedge Date \geq 01-01-2020 \wedge Date \leq 31-12-2020} (THEATRE \bowtie PERFORMANCE \bowtie PLAY)$$

Conclude by **projecting** on **Name**, a unique attribute within ***r***:

$$\pi_{Name} r$$

Or retrieve ***r*** more *efficiently* by **selecting** the relevant tuples *before joining* them:

$$\sigma_{City=Roma} THEATRE \bowtie \sigma_{Date \geq 01-01-2020 \wedge Date \leq 31-12-2020} PERFORMANCE \bowtie \sigma_{Author=Pirandello} PLAY$$

1.2. Answer

The attribute values within **PERFORMANCE** and **THEATRE** are sufficient to obtain the plays' titles. **PLAY** is unnecessary, as it contains authors' names.

While performing selections, tuples are either included *or* discarded sequentially. Consider a join between **THEATRE** and **PERFORMANCE**, then a selection based on the resulting tuples' cities; were we to include a tuple whose city is "Rome" and whose title is "King Lear", it wouldn't be possible to remove it when we encounter a subsequent tuple with an equal title but a different city.



This class of relational algebra problems is best handled with **subtractions**. The tuples that *don't* meet the selection criteria are first collected and then removed from the set of all the candidate tuples.

Let us define **r** which gathers the titles of all the plays that **weren't** performed in Rome:

- **join** **THEATRE** and **PERFORMANCE**, whose data are essential
- **select** the tuples whose **City** attribute is **different** from *Rome*
- **project** on **Title**

$$\pi_{Title} \sigma_{City \neq Rome} (THEATRE \bowtie PERFORMANCE)$$

Alternatively, a more *efficient* query would narrow down the tuples in **THEATRE**:

$$\pi_{Title} (\sigma_{City \neq Rome} THEATRE \bowtie PERFORMANCE)$$

It's now sufficient to remove **r** from the projection of **PERFORMANCE** on **Title**:

$$\pi_{Title} (PERFORMANCE) - r$$



Since **PLAY's key is `Title`**, which therefore uniquely identifies each play, it could be used in lieu of **PERFORMANCE**, in the query above. It might even be more efficient on account of its lower tuple count.



The following query is wrong, as it includes plays that were performed in Rome **and** in other cities, therefore **not exclusively** in Rome:

$$\pi_{Title} \sigma_{City = Rome} (THEATRE \bowtie PERFORMANCE)$$