ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Data Modelling and Databases (DMDB)
Spring Semester 2017
Lecturer(s): Gustavo Alonso, Ce Zhang
Assistant(s): Claude Barthels, Eleftherios Sidirourgos, Eliza Wszola, Ingo Müller, Kaan Kara, Renato Marroquín, Zsolt István

ETH Zurich
Systems Group
Date: March 13/March 17, 2017
Last update: August 16, 2017

Exercise 3: Relational Algebra

The exercises marked with * will be discussed in the exercise session. You can solve the other exercises as practice, ask questions about them in the session, and hand them in for feedback. **All exercises may be relevant for the exam.**

Ask **Claude (claude.barthels@inf.ethz.ch)** for feedback on this week's exercise sheet or give it to the TA of your session (preferably stapled and with your e-mail address).

1 Library Database *

Consider the following relational schema:

`Reader` (<u>RDNR</u>, Surname, Firstname, City, Birthdate)
`Book` (<u>ISBN</u>, Title, Author, NoPages, PubYear, PublisherName)
`Publisher` (<u>PublisherName</u>, PublisherCity)
`Category` (<u>CategoryName</u>, BelongsTo)
`Copy` (<u>ISBN</u>, <u>CopyNumber</u>, Shelf, Position)
`Loan` (<u>ReaderNr</u>, <u>ISBN</u>, <u>Copy</u>, ReturnDate)
`BookCategory` (<u>ISBN</u>, <u>CategoryName</u>)

Formulate the following queries in relational algebra:

1. Which are the last names of the readers in Zurich?

2. Which books (Author, Title) are from publishers in Zurich, Bern, or New York?

3. Which books (Author, Title) has the reader Lemmi Schmöker borrowed?

4. Which books in the category 'Alps' do not belong to the category 'Switzerland'? Do not take into account subcategories!

5. Which readers (Surname, Firstname) have borrowed books that were published in their home town?

6. Which readers (Surname, Firstname) have borrowed at least a book that has been borrowed also by the reader Lemmi Schmöker (the reader Lemmi Schmöker should not be included in the results)?

2 Olympic Games

Consider the following relational schema:

`Runner` (<u>Name</u>, Birthday, Country)
`Run` (<u>Name, Distance</u>, Time)

A runner can run in several runs over different race distances. Thanks to high-speed cameras, two runners cannot have the exact same time in the same run.

1. For every description find a matching relational algebra query. For some descriptions there is no matching query.

   Descriptions:

   1. All 100m race distance runs in which only runners from Switzerland (*CH*) participated.
   2. All runs with a distance greater than 100m in which only runners from Switzerland participated.
   3. All runs in which only runners from Switzerland participated.
   4. All 100m race distance runs in which the runners were not from Switzerland.
   5. All runs in which the runners were not from Switzerland.

Relation Algebra Queries:

1. $$\Pi_{\texttt{Name,Distance,Time}}((\texttt{Runner} - \sigma_{\texttt{Country} != '\texttt{CH}'}(\texttt{Runner})) \bowtie (\texttt{Run} \cup \sigma_{\texttt{Distance} < 100}(\texttt{Run})))$$

2. $$\Pi_{\texttt{Name,Distance,Time}}((\sigma_{\texttt{Country} = '\texttt{CH}'}(\texttt{Runner}) - \sigma_{\texttt{Country} != '\texttt{CH}'}(\texttt{Runner})) \bowtie \sigma_{\texttt{Distance} != 100}(\texttt{Run} - \sigma_{\texttt{Distance} < 100}(\texttt{Run})))$$

3. $$\Pi_{\texttt{Name,Distance,Time}}((\sigma_{\texttt{Country} != '\texttt{CH}'}(\texttt{Runner}) - \sigma_{\texttt{Country} = '\texttt{CH}'}(\texttt{Runner})) \bowtie \sigma_{\texttt{Distance}=100}(\texttt{Run} - \sigma_{\texttt{Distance} > 100}(\texttt{Run})))$$

4. $$\Pi_{\texttt{Name,Distance,Time}}((\sigma_{\texttt{Country} = '\texttt{CH}'}(\texttt{Runner}) - \sigma_{\texttt{Country} != '\texttt{CH}'}(\texttt{Runner})) \bowtie \sigma_{\texttt{Distance}=100}(\texttt{Run} - \sigma_{\texttt{Distance} < 100}(\texttt{Run})))$$

5. $$\Pi_{\texttt{Name,Distance,Time}}(\sigma_{\texttt{Country} != '\texttt{CH}'}(\texttt{Runner})) \bowtie \sigma_{\texttt{Distance}>100}(\texttt{Run} - \sigma_{\texttt{Distance} < 100}(\texttt{Run})))$$

Fill in the table below by writing to every description on the left the right query letter on the right. If there is no matching query for a description, put a cross.

Description	Query
1	
2	
3	
4	
5	

2. Which of the following relational algebra expressions finds all runners which **only** participated in 100m race distance runs.
   - ◯ $\Pi_{\texttt{Name}}(\sigma_{\texttt{Distance}=100}\texttt{Run})$
   - ◯ $\Pi_{\texttt{Name}}(\texttt{Runner}) - \Pi_{\texttt{Name}}(\sigma_{\texttt{Distance}=100}\texttt{Run})$
   - ◯ $\Pi_{\texttt{Name}}(\texttt{Run}) - \Pi_{\texttt{Name}}(\sigma_{\texttt{Distance}=100}\texttt{Run})$
   - ◯ $\Pi_{\texttt{Name}}(\texttt{Run}) - \Pi_{\texttt{Name}}(\sigma_{\texttt{Distance!}=100}\texttt{Run})$
   - ◯ $\Pi_{\texttt{Name}}(\texttt{Runner}) - \Pi_{\texttt{Name}}(\sigma_{\texttt{Distance!}=100}\texttt{Run})$

3. We want to find winners for every distance. A winner has the shortest time for a given distance.

   1. $$\Pi_{\texttt{Name,Country,Distance,Time}}(\texttt{Runner} \bowtie (\texttt{Run} - \Pi_{\texttt{Run1.Name,Run1.Distance,Run1.Time}}(\sigma_{\texttt{Run1.time}>\texttt{Run2.time}}(\rho_{\texttt{Run1}}(\texttt{Run}) \bowtie_{\texttt{Name}} \rho_{\texttt{Run2}}(\texttt{Run})))))$$

2. $\Pi_{\texttt{Name,Country,Distance,Time}}(\texttt{Runner} \bowtie (\texttt{Run} - \Pi_{\texttt{Run1.Name,Run1.Distance,Run1.Time}}(\sigma_{\texttt{Run1.time>Run2.time}}(\rho_{\texttt{Run1}}(\texttt{Run}) \bowtie_{\texttt{Distance}} \rho_{\texttt{Run2}}(\texttt{Run})))))$

3. $\Pi_{\texttt{Name,Country,Distance,Time}}(\texttt{Runner} \bowtie (\texttt{Run} - \Pi_{\texttt{Run1.Name,Run1.Distance,Run1.Time}}(\sigma_{\texttt{Run1.time<Run2.time}}(\rho_{\texttt{Run1}}(\texttt{Run}) \bowtie_{\texttt{Name}} (\rho_{\texttt{Run2}}(\texttt{Run}))))))$

4. $\Pi_{\texttt{Name,Country,Distance,Time}}(\texttt{Runner} \bowtie (\texttt{Run} - \Pi_{\texttt{Run1.Name,Run1.Distance,Run1.Time}}(\sigma_{\texttt{Run1.time<Run2.time}}(\rho_{\texttt{Run1}}(\texttt{Run}) \bowtie_{\texttt{Distance}} \rho_{\texttt{Run2}}(\texttt{Run})))))$

Mark all the queries that find the winners with a checkmark (✓) in the table below.

1	2	3	4

3 Result Cardinality *

Consider the following two relations:

R =

A	B
1	x
2	y
2	z
3	x
9	a

S =

B	C	D
x	0	3
y	2	1
y	3	3
w	3	0
y	2	0

Fill out for the following relational algebra expressions how many tuples each of them returns, based on the data given above.

Expression	**Size of result (number of tuples)**
R × S	
R ⋈ S	
R ⟕ S	
R ⟖ S	
R $\bowtie_{A=D}$ S	
$\rho_{C\leftarrow A}$(R) ⋈ S	
Π_B(R) - $\Pi_B(\sigma_{C<3}$(S))	
Π_A(R)$\cap\rho_{A\leftarrow D}(\Pi_D$(S))	
Π_D(S)⋈S	

4 Join Operators

1. Consider the following two relations:

P =

A	B
1	1
2	2
1	3
2	4
3	1
1	2

Q =

B	C	D
1	4	0
2	5	2
1	7	2
3	2	2

For each of the following expressions circle all the tuples that are *not* in its result set (the tuples contain all four columns: [A, B, C, D]).

A. $P \bowtie Q$:

(a) [1, 1, 7, 2]

(b) [1, 2, 5, 2]

(c) [3, 2, 5, 0]

(d) [3, 1, 4, 0]

(e) [2, 4, 2, 2]

B. $P \ltimes\!\!\!\!\bowtie Q$:

(a) [3, 1, 7, 2]

(b) [4, 1, 4, 0]

(c) [2, 4, - , -]

(d) [3, 1, 4, 0]

(e) [1, 3, 2, 2]

2. Which of the following relational algebra expressions represents ① a left outer join (⟕), ② a right outer join (⟖), and ③ a full outer join (⟗).

○ $\Pi_{R \cup S}(S - \Pi_S(R \bowtie S)) \cup (R \bowtie S)$

○ $\Pi_{R \cup S}(R - \Pi_R(R \bowtie S)) \cup (R \bowtie S)$

○ $(R \bowtie S) \cup (\Pi_{R \cup S}(R - \Pi_R(R \bowtie S))) \cup (\Pi_{R \cup S}(S - \Pi_S(R \bowtie S)))$

5 Train Connections *

Consider the following relational schema:

`Cities` (<u>Name</u>, <u>State</u>)
`Stations` (<u>Name</u>, NoPlatforms, CityName, State)
`Itinerary` (<u>ItNr</u>, Length, StartStation, DestinationStation)
`Connections` (<u>FromStation</u>, <u>ToStation</u>, <u>ItNr</u>, Departure, Arrival)

Suppose that the relation `Connections` already contains the transitive closure for each given train, e.g., if there is a *direct* train from Zurich to Geneva with a stop in Bern, then there exists a relation tuple for Zurich → Bern, Bern → Geneva, and Zurich → Geneva. Formulate the following queries in relational algebra:

1. Find all the direct connections from Zurich (any station) to Geneva (any station)

2. Find all the single-transfer connections from Zurich to Locarno. The transfer station can be any of the stations but the connecting trains should run on the same day. (You can use a function DAY() on the attributes Departure and Arrival in order to determine the day.)

3. Is it possible to find all possible connections between two stations independent on the number of transfers?