Databases
Lab

Relational Algebra Exercises

Flavio Bertini

flavio.bertini@smartdata.cs.unibo.it

Exercise 1

Given the following relations:

REGION(<u>Name</u>, Surface)

CITY(<u>Code</u>, Name, Population)

BELONGING(<u>Region</u>, <u>City</u>)

Write a relational algebra expression that returns the name of the regions having cities with more than 1,000,000 inhabitants.

Exercise 1: comments

Let's start by dividing the text in such a way as to identify the operators we need to use:

- name of the regions
- having cities
- with more than 1,000,000 inhabitants

Exercise 1: step 1

Let's start by selecting cities with more than 1,000,000 inhabitants:

$$\sigma_{\text{Population}>1{,}000{,}000}(\text{CITY})$$

Exercise 1: step 2

Then, to provide "the names of regions ..." we must connect the cities with their region, and we need a renaming to allow a natural join:

$$\rho_{\text{Code} \leftarrow \text{City}}(\text{BELONGING})$$
$$\bowtie$$
$$\sigma_{\text{Population} > 1.000.000}(\text{CITY})$$

Exercise 1: final solution

Finally, we can complete the query by using a projection that only selects the name of the regions:

$$\pi_{\text{Region}}(\rho_{\text{Code}\leftarrow\text{City}}(\text{BELONGING}) \bowtie \sigma_{\text{Population}>1.000.000}(\text{CITY}))$$

We do not need to use `REGION`, since `BELONGING` already provides the name of the regions.

Exercise 2

Given the following relation:

```
TRAIN(Code, Start, End, Miles)
```

Write a relational algebra expression that returns all the routes between Boston and Chicago with one interchange.

Exercise 2: comments

- There is only one relation, that is TRAIN.
- Since the exercise requires the routes from Boston to Chicago with only one interchange, we have to join TRAIN with itself.
- Moreover, we also have to rename the attributes before the join to be able to distinguish them after the join.

Exercise 2: step 1

Attributes renaming for departing trains:

$$\rho_{\text{Code1,Switch,Miles1} \leftarrow \text{Code,End,Miles}}(\text{TRAIN})$$

Attributes renaming for arriving trains:

$$\rho_{\text{Code2,Switch,Miles2} \leftarrow \text{Code,Start,Miles}}(\text{TRAIN})$$

Exercise 2: step 2

The first renaming changed `End` to `Switch`, as the second one renamed `Start` to `Switch`.

It is now possible apply a natural join on `Switch` to get all the train routes with only one interchange:

$$\rho_{\text{Code1,Switch,Miles1} \leftarrow \text{Code,End,Miles}}(\text{TRAIN})$$

$$\bowtie$$

$$\rho_{\text{Code2,Switch,Miles2} \leftarrow \text{Code,Start,Miles}}(\text{TRAIN})$$

Exercise 2: final solution

Finally, we can complete the query selecting only the routes from Boston to Chicago:

$$\sigma_{\text{Start='Boston'} \wedge \text{End='Chicago'}}$$

$$(\rho_{\text{Code1,Switch,Miles1} \leftarrow \text{Code,End,Miles}}(\text{TRAIN})$$

$$\bowtie$$

$$\rho_{\text{Code2,Switch,Miles2} \leftarrow \text{Code,Start,Miles}}(\text{TRAIN}))$$

Exercise 2: extended edition

$$\rho_{Code1,Switch,Miles1 \leftarrow Code,End,Miles}(TRAIN)$$

$$\bowtie_{Start='Boston' \wedge End='Chicago'}$$

$$\rho_{Code2,Switch,Miles2 \leftarrow Code,Start,Miles}(TRAIN)$$

$$\rho_{Code1,Switch,Miles1 \leftarrow Code,End,Miles}(\sigma_{Start='Boston'}(TRAIN))$$

$$\bowtie$$

$$\rho_{Code2,Switch,Miles2 \leftarrow Code,Start,Miles}(\sigma_{End='Chicago'}(TRAIN))$$

Exercise 3

Given the following relations:

REGION(<u>Name</u>, Surface)

CITY(<u>Code</u>, Name, Population)

BELONGING(<u>Region</u>, <u>City</u>)

Write a relational algebra expression that returns the distinct names of cities that belong to regions greater than 2,500 square feet. You can only use renaming, selection, projection and natural join operations.

Exercise 3: comments

1. Firstly, we have to select regions with a surface greater than 2,500 square feet.
2. Then, we have to retrieve the name of the cities that belong to those regions.
3. We have to join together all the three relations.

Exercise 3: step 1

Regions with a surface greater than 2,500 square feet:

$$\sigma_{\text{Surface}>2{,}500}(\text{REGION})$$

Exercise 3: step 2

Attributes renaming allows a natural join:

$$\rho_{\text{Region}\leftarrow\text{Name}}(\sigma_{\text{Surface}>2{,}500}(\text{REGION}))$$

$$\bowtie$$

$$\text{BELONGING}$$

$$\bowtie$$

$$\rho_{\text{City}\leftarrow\text{Code}}(\text{CITY})$$

Exercise 3: final solution

We can complete the query retrieving the name of the cities:

$$\pi_{Name}(\rho_{Region \leftarrow Name}(\sigma_{Surface>2,500}(REGION))$$

$$\bowtie$$

$$BELONGING$$

$$\bowtie$$

$$\rho_{City \leftarrow Code}(CITY))$$

After renaming the attribute `Name` refers to `CITY`.

Exercise 3: alternative solution

$$\pi_{Name}(\sigma_{Surface>2{,}500}(\rho_{Region \leftarrow Name}(REGION)$$

$$\bowtie$$

$$BELONGING$$

$$\bowtie$$

$$\rho_{City \leftarrow Code}(CITY)))$$

Exercise 4

Given the following relations:

```
PERSON(Id, Name, Surname, Age)
ENROLLMENT(Person, Course)
COURSE(Name, Price)
```

Write a relational algebra expression that returns the distinct names of people that attend a course that costs more than $10 for each year of their age (i.e., `Price>Age*10`).

Exercise 4: comments

1. Once again, we have to join together all the three relations to compare age and price.
2. Then, we can apply selection and projection to provide the final result.

Exercise 4: step 1

Attributes renaming allows a natural join:

$$\rho_{\text{Person}\leftarrow\text{Id}}(\text{PERSON})$$

$$\bowtie$$

$$\text{ENROLLMENT}$$

$$\bowtie$$

$$\rho_{\text{Course}\leftarrow\text{Name}}(\text{COURSE})$$

Exercise 4: step 2

We can select the right people according with `Price` vs `Age` constraint:

$$\sigma_{Price>Age*10}(\rho_{Person\leftarrow Id}(PERSON)$$

$$\bowtie$$

$$ENROLLMENT$$

$$\bowtie$$

$$\rho_{Course\leftarrow Name}(COURSE))$$

Exercise 4: final solution

We can complete the query retrieving the name of the people:

$$\pi_{Name}(\sigma_{Price>Age*10}(\rho_{Person\leftarrow Id}(PERSON) \bowtie ENROLLMENT \bowtie \rho_{Course\leftarrow Name}(COURSE)))$$

NB: We cannot push down the selection.

Exercise 5

Given the following relations:

```
STUDENT(Id, Name, Surname, School)
EXAM(Student, Subject, Mark, Date)
```

Write a relational algebra expression that returns name and surname of students that scored 30 in at least one exam.

Exercise 5: final solution

We select exams with a score of 30, than we can retrieve students' information using an equi-join:

$$\pi_{Name,Surname}(STUDENT \bowtie_{Id=Student} \sigma_{Mark=30}(EXAM))$$

Exercise 6

Given the following relations:

SUPPLIER(<u>Id</u>, Name, Partners, Headquarter)

PRODUCT(<u>Id</u>, Name, Colour, Size, Store)

SUPPLYING(<u>SupId</u>, <u>ProdId</u>, Quantity)

Write a relational algebra expression that returns the name of suppliers supplying at least one product other than “PX274”.

Exercise 6: final solution

$$\pi_{Name}(\text{SUPPLIER}$$

$$\bowtie_{Id=SupId}$$

$$\sigma_{ProdId \neq \text{'PX274'}}(\text{SUPPLYING}))$$

Exercise 7

Given the following relations:

SUPPLIER(<u>Id</u>, Name, Partners, Headquarter)

PRODUCT(<u>Id</u>, Name, Colour, Size, Store)

SUPPLYING(<u>SupId</u>, <u>ProdId</u>, Quantity)

Write a relational algebra expression that returns the code of suppliers supplying at least two different products.

Exercise 7: final solution

$$\pi_{\text{SupId1}}(\rho_{\text{SupId1, ProdId1, Quantity1}\leftarrow\text{SupId, ProdId, Quantity}}(\text{SUPPLYING})$$

$$\bowtie_{\text{SupId1=SupId2} \wedge \text{ProdId1}\neq\text{ProdId2}}$$

$$\rho_{\text{SupId2, ProdId2, Quantity2}\leftarrow\text{SupId, ProdId, Quantity}}(\text{SUPPLYING}))$$

Exercise 8

Given the following relations:

`STUDENT(Id, Name, Surname, School)`

`EXAM(Student, Subject, Mark, Date)`

Write a relational algebra expression that returns name and surname of students that have **never** scored 30 in an exam.

Exercise 8: final solution

We select exams with a score of 30, than we can retrieve students' information using an equi-join:

$$\pi_{\text{Name,Surname,Id}}(\text{STUDENT}) -$$

$$\pi_{\text{Name,Surname,Id}}(\text{STUDENT}$$

$$\bowtie_{\text{Id=Student}}$$

$$\sigma_{\text{Mark=30}}(\text{EXAM}))$$

Exercise 9

Given the following relations:

```
BOOK(BookId, Title, Author)
USER(UserId, Name, Surname)
BORROWING(UserId, BookId, Date)
```

Write a relational algebra expression that returns the identifier of users that borrowed a book of Fleming.

Exercise 9: final solution

$$\pi_{\text{UserId}}(\sigma_{\text{Author='Fleming'}}(\text{BOOK}) \bowtie \text{BORROWING})$$

Exercise 10

Given the following relations:

BOOK(<u>BookId</u>, Title, Author)

USER(<u>UserId</u>, Name, Surname)

BORROWING(<u>UserId</u>, <u>BookId</u>, <u>Date</u>)

Write a relational algebra expression that returns the identifier of users that only borrowed books of Fleming.

Exercise 10: final solution

We are interested in users that borrowed a Fleming's book, however, we have to remove users that borrowed books written by other authors:

$$\pi_{\text{UserId}}(\sigma_{\text{Author='Fleming'}}(\text{BOOK}) \bowtie \text{BORROWING})$$

$$-$$

$$\pi_{\text{UserId}}(\sigma_{\text{Author}\neq\text{'Fleming'}}(\text{BOOK}) \bowtie \text{BORROWING})$$

Exercise 11

Given the following relations:

```
EXAM(Code, Subject, Teacher)
ROOM(Code, Name, Building, Capacity)
BOOKING(Room, Exam, Date)
```

Write a relational algebra expression that returns the name and the building of the classrooms where a test of the Database course takes place today.

Exercise 11: final solution

$$\pi_{Name,Building}(\pi_{Room}(\sigma_{Subject='Database'}(EXAM)$$

$$\bowtie_{Code=Exam}$$

$$\sigma_{Date='today'}(BOOKING))$$

$$\bowtie_{Room=Code} ROOM)$$

Exercise 12

Given the following relations:

STUDENT(<u>Id</u>, Name, Surname, School)

EXAM(<u>Student</u>, <u>Subject</u>, Mark, Date)

Write a relational algebra expression that returns name and surname of students that scored 30 in at least one exam. Return also the first date of such exams.

Exercise 12: an instance

STUDENT

<u>Id</u>	Name	Surname	School
M1	Sheldon	Cooper	Computer Science
M2	Harvey	Specter	Computer Science

EXAM

<u>Student</u>	<u>Subject</u>	Mark	Date
M1	Database Systems	30	08/05/2012
M1	Data Structures and Algorithms	30	10/05/2012
M1	Software Engineering	30	06/06/2012
M1	Algebra	28	07/01/2011
M2	Operating Systems	26	07/02/2012

Exercise 12: step 1

Let's start by selecting all pairs <student, date> of exams with a score of 30. We will use views to build the final result:

$$R1 := \pi_{Student,Date}(\sigma_{Mark=30}(EXAM))$$

R1

Student	Date
M1	08/05/2012
M1	10/05/2012
M1	06/06/2012

Exercise 12: step 2

Let's clone R1 in order to compare together the dates of exams with a score of 30:

$$R2 := \rho_{\text{Date1} \leftarrow \text{Date}}(R1)$$

R1

Student	Date
M1	08/05/2012
M1	10/05/2012
M1	06/06/2012

R2

Student	Date1
M1	08/05/2012
M1	10/05/2012
M1	06/06/2012

Exercise 12: step 3

We can use a natural join (a cartesian product in this case) to compare the dates:

$$R1 \bowtie R2$$

Date	Student	Date1
08/05/2012	M1	08/05/2012
08/05/2012	M1	10/05/2012
08/05/2012	M1	06/06/2012
10/05/2012	M1	08/05/2012
10/05/2012	M1	10/05/2012
10/05/2012	M1	06/06/2012
06/06/2012	M1	08/05/2012
06/06/2012	M1	10/05/2012
06/06/2012	M1	06/06/2012

Exercise 12: step 4

Now we can compare the dates selecting the lower one:

$$\sigma_{Date<Date1}(R1 \bowtie R2)$$

Date	Student	Date1
08/05/2012	M1	10/05/2012
08/05/2012	M1	06/06/2012
10/05/2012	M1	06/06/2012

Exercise 12: step 5

We are interested in `Student` and `Date1`, so that we can remove them from `R2`:

$$R2 - \pi_{Student,Date1}(\sigma_{Date<Date1}(R1 \ltimes R2))$$

R2

Student	Date1
M1	08/05/2012
M1	10/05/2012
M1	06/06/2012

−

Student	Date1
M1	10/05/2012
M1	06/06/2012

=

Student	Date1
M1	08/05/2012

Exercise 12: final solution

We can complete the query using a renaming and retrieving information from `Student`:

$$R3 := \rho_{Date \leftarrow Date1}(R2 - \pi_{Student,Date1}(\sigma_{Date<Date1}(R1 \bowtie R2)))$$

$$\pi_{Name,Surname,Date}(STUDENT \bowtie_{Id=Student} R3)$$

Exercise 13

Given the following relations:

EXAM(<u>Code</u>, Subject, Teacher)

ROOM(<u>Code</u>, Name, Building, Capacity)

BOOKING(<u>Room</u>, <u>Exam</u>, <u>Date</u>)

Write a relational algebra expression that returns the name of the classrooms with more than or equal to 100 seats that are not booked today.

Exercise 13: final solution

$$BIGROOM := \sigma_{Capacity \geq 100}(ROOM)$$

$$\pi_{Name}(\pi_{Name,Code}(BIGROOM) -$$

$$\pi_{Name,Code}(BIGROOM \bowtie_{Code=Room}$$

$$\sigma_{Date='today'}(BOOKING)))$$

Exercise 14

Given the following relations:

WINESHOP(<u>Code</u>, Name, Address, ZipC)

CATALOG(<u>Wineshop</u>, <u>Wine</u>, Price)

WINE(<u>Code</u>, Name, Color, Region)

Write a relational algebra expression that returns the code of the wine shops that have rosé or red wines.

Exercise 14: final solution

$$\pi_{Wineshop}(CATALOG$$

$$\bowtie_{Wine=Code}$$

$$\sigma_{Color='Rosé' \vee Color='Red'}(WINE))$$

Exercise 15

Given the following relations:

```
WINESHOP(Code, Name, Address, ZipC)
CATALOG(Wineshop, Wine, Price)
WINE(Code, Name, Color, Region)
```

Write a relational algebra expression that returns the name of the wine shops that have rosé and red wines.

Exercise 15: final solution

$$\pi_{Name}($$

$$(\pi_{Wineshop}(CATALOG \bowtie_{Wine=Code} \sigma_{Color='Rosé'}(WINE))$$

$$\cap$$

$$\pi_{Wineshop}(CATALOG \bowtie_{Wine=Code} \sigma_{Color='Red'}(WINE)))$$

$$\bowtie_{Wineshop=Code} WINESHOP)$$

NB: We cannot use `Color='Rosé' ∧ Color='Red'`.

Exercise 16

Given the following relations:

```
PERSON(Name, Age, Gender)
FREQUENTS(Name, Pizzeria)
EATS(Name, Pizza)
SERVES(Pizzeria, Pizza, Price)
```

Write a relational algebra expression that returns the names of all females who eat both mushroom and pepperoni pizza.

Exercise 16: final solution

$$\pi_{Name}(\sigma_{Gender='F' \wedge Pizza='pepperoni'}(PERSON \bowtie EATS))$$

$$\cap$$

$$\pi_{Name}(\sigma_{Gender='F' \wedge Pizza='mushroom'}(PERSON \bowtie EATS))$$

Exercise 17

Given the following relations:

PERSON(<u>Name</u>, Age, Gender)

FREQUENTS(<u>Name</u>, <u>Pizzeria</u>)

EATS(<u>Name</u>, <u>Pizza</u>)

SERVES(<u>Pizzeria</u>, <u>Pizza</u>, Price)

Write a relational algebra expression that returns the names of all pizzerias that are frequented by only females or only males.

Exercise 17: final solution

$$(\pi_{Pizzeria}(\sigma_{Gender='F'}(PERSON) \bowtie FREQUENTS) -$$

$$\pi_{Pizzeria}(\sigma_{Gender='M'}(PERSON) \bowtie FREQUENTS))$$

$$\cup$$

$$(\pi_{Pizzeria}(\sigma_{Gender='M'}(PERSON) \bowtie FREQUENTS) -$$

$$\pi_{Pizzeria}(\sigma_{Gender='F'}(PERSON) \bowtie FREQUENTS))$$

Exercise 18

Given the following relations:

MAGAZINE(<u>Code</u>, Name, Editor)

ARTICLE(<u>Code</u>, Title, Topic, Magazine)

Write a relational algebra expression that returns the code and the name of the magazines that have never published articles about motorcycling.

Exercise 18: final solution

$$\pi_{Code,\ Name}(MAGAZINE$$

$$\bowtie$$

$$(\pi_{Code}(MAGAZINE) -$$

$$\rho_{Code \leftarrow Magazine}(\pi_{Magazine}(\sigma_{Topic='motorcycling'}(ARTICLE))))$$

Exercise 19

Given the following relations:

SAILOR(<u>Id</u>, Name, Level, BirthD)

BOOKING(<u>Sailor</u>, <u>Boat</u>, <u>Date</u>)

BOAT(<u>Id</u>, Name, Colour)

Write a relational algebra expression that returns the codes and names of the sailors who have booked a red and a green boat.

Exercise 19: final solution

$$\pi_{\text{Sid, Sname}}(\rho_{\text{Sid, Sname, Llevel, SbirthD} \leftarrow \text{Id, Name, Level, BirthD}}(\text{SAILOR})$$

$$\bowtie_{\text{Sid=Sailor1}}$$

$$(\rho_{\text{Sailor1} \leftarrow \text{Sailor}}(\text{BOOKING} \bowtie_{\text{Boat=Id}} (\sigma_{\text{Colour='Red'}}(\text{BOAT})))$$

$$\bowtie_{\text{Sailor1=Sailor2}}$$

$$\rho_{\text{Sailor2} \leftarrow \text{Sailor}}(\text{BOOKING} \bowtie_{\text{Boat=Id}} (\sigma_{\text{Colour='Green'}}(\text{BOAT})))))$$

Exercise 20

Given the following relations:

```
PLANE(Id, Name, Range)
CERTIFICATE(Eid, Pid)
EMPLOYEE(Id, Name, Wage)
```

Write a relational algebra expression that returns the codes and names of the employees authorised to fly on at least two aircraft capable of covering distances of more than 5000 miles.

Exercise 20: final solution

$$\pi_{Id,\ Name}(\text{EMPLOYEE}$$

$$\bowtie_{Id=Eid1}$$

$$((\rho_{Id1,\ Eid1 \leftarrow Id,\ Eid}(\sigma_{Range \geq 5000}(\text{PLANE}) \bowtie_{Id=Pid} \text{CERTIFICATE}))$$

$$\bowtie_{Id1 \neq Id2\ \wedge\ Eid1=Eid2}$$

$$(\rho_{Id2,\ Eid2 \leftarrow Id,\ Eid}(\sigma_{Range \geq 5000}(\text{PLANE}) \bowtie_{Id=Pid} \text{CERTIFICATE}))))$$

Appendix: remote exam notation (1)

- Special notation for relational algebra queries to be used on the online platform during the online exam, where:
  - R is a relationship
  - A is an attribute
  - C is a condition

Appendix: remote exam notation (2)

- **Boolean**
  - NOT C
  - C1 OR C2
  - C1 AND C2
- **Union**
  - R1 UNI R2
- **Intersection**
  - R1 INT R2
- **Difference**
  - R1 DIF R2

- **Rename**
  - RID[(<new A1>,<old A1>),(<new A2>,<old A2>),...]R
- **Select**
  - SEL[C]R
  - SEL[C1 AND C2 OR A=”value”]R
- **Projection**
  - PRO[A1, A2, ...]R
- **Join**
  - R1 JOIN[C] R2

Appendix: remote exam notation (3)

A relational algebra query using the standard notation

$$(\pi_{idModello,nome}(\sigma_{categoria='Berlina' \wedge nazionalità='Francia'}(Modello \bowtie \rho_{nomeM \leftarrow nome} Marca)) \quad -$$

$$(\pi_{idModello,nome}(\sigma_{anno='2007'}(Modello \bowtie Valutazione)))) \bowtie_{idModello\ =\ modello} \ Valutazione$$

The same query using the remote exame notation

```
(PRO[idModello,nome](SEL[categoria='Berlina' AND nazionalità='Francia'](Modello JOIN
RID[(nomeM,nome)]Marca)) DIF
(PRO[idModello,nome](SEL[anno='2007'](Modello JOIN Valutazione))))
JOIN[idModello=modello] Valutazione
```