Relational Algebra and SQL - Solutions

1 Relational Algebra - Task 1

Formulate the following queries in relational algebra:

a) Which are the last names of the readers in Zurich?
   \[\Pi_{\text{Surname}}(\sigma_{\text{City}=\text{Zurich}}(\text{READER}))\]

b) Which books (Author, Title) are from publishers in Zurich, Bern or New York?
   \[\Pi_{\text{Author,Title}}(\text{BOOK} \bowtie (\sigma_{\text{City}=\text{Zurich OR City=Bern OR City=New York}}(\text{PUBLISHER})))\]

c) Which books (Author, Title) has the reader Lemmi Schm"oker borrowed?
   \[\Pi_{\text{Author,Title}}(\text{BOOK} \bowtie \text{LOAN} \bowtie \text{ReaderNr=RDNR} (\sigma_{\text{Surname=Schmoker AND Firstname=Lemmi}}(\text{READER})))\]

d) Which books in the category "Alps" do not belong to the category "Switzerland"? Do not take into account subcategories!
   \[\Pi_{\text{ISBN}}(\sigma_{\text{CategoryName=Alps}}(\text{BOOKS})) - \Pi_{\text{ISBN}}(\sigma_{\text{CategoryName=Switzerland}}(\text{BOOKS}))\]

e) Which readers (Surname, Firstname) have borrowed books that were published in their home town?
   \[\Pi_{\text{Firstname,Surname}}(\sigma_{\text{City=PublisherCity}}(\text{PUBLISHER} \bowtie \text{BOOK} \bowtie \text{LOAN} \bowtie \text{ReaderNr=RDNR} \text{READER}))\]

f) Which readers (Surname, Firstname) have borrowed at least a book that has been borrowed also by the reader Lemmi Schm"oker (the reader Lemmi Schm"oker should not be included in the results)?
   \[\Pi_{R_1.\text{Firstname},R_1.\text{Surname}}(\rho_{R_1}(\text{READER}) \bowtie \text{ReaderNr=RDNR} \rho_{L_1}(\text{LOAN}))
   \bowtie R_1.\text{RDNR}<>R_2.\text{RDNR} R_1.\text{ISBN}=L_2.\text{ISBN}
   \rho_{L_2}(\sigma_{\text{Surname=Schmoker AND Surname=Lemmi}}\text{READER}) \bowtie \text{ReaderNr=RDNR} \rho_{L_2}(\text{LOAN}))\]

2 Relational Algebra - Task 2

Formulate the following queries in relational algebra:

a) Find all the direct connections from Zurich to Geneva
b) 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 \( \text{DAY()} \) on the attributes Departure and Arrival in order to determine the day)

\[
\begin{align*}
(\rho_{\text{FromName}←\text{Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName}=\text{Zurich}}(\text{STATIONS})))) & \bowtie_{\text{FromName}=\text{FromStation}} \text{CONNECTIONS} \\
(\rho_{\text{ToName}←\text{Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName}=\text{Geneva}}(\text{STATIONS})))) & \bowtie_{\text{ToName}=\text{ToStation}} \\
(\rho_{\text{FromName}←\text{Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName}=\text{Locarno}}(\text{STATIONS}))))
\end{align*}
\]

\[
(\rho_{\text{FromName}←\text{Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName}=\text{Zurich}}(\text{STATIONS})))) \bowtie_{\text{FromName}=c1.\text{FromStation}} \rho_{c1}(\text{CONNECTIONS}) \\
\bowtie_{c1.\text{ToStation}=c2.\text{FromStation} \land c1.\text{Arrival}<c2.\text{Departure} \land \text{DAY}(c1.\text{Arrival})=\text{DAY}(c2.\text{Departure}) \land c1.\text{ItNr}<c2.\text{ItNr}} \ ho_{c2}(\text{CONNECTIONS}) \\
(\rho_{\text{ToName}←\text{Name}}(\Pi_{\text{Name}}(\sigma_{\text{CityName}=\text{Locarno}}(\text{STATIONS}))))
\]

\[
(\sigma_{\text{FromStation}=\text{ZurichHB}} \bowtie_{\text{ToStation}=\text{GenevaHB}} (\text{CONNECTIONS})) \cup \\
(\sigma_{c1.\text{FromStation}=\text{ZurichHB}} \bowtie_{c2.\text{ToStation}=\text{GenevaHB}} (\rho_{c1}(\text{CONNECTIONS}))) \cup \\
(\sigma_{c1.\text{FromStation}=\text{ZurichHB}} \bowtie_{c2.\text{ToStation}=\text{GenevaHB}} (\rho_{c1}(\text{CONNECTIONS}))) \cup \\
(\sigma_{c1.\text{FromStation}=\text{ZurichHB}} \bowtie_{c3.\text{ToStation}=\text{GenevaHB}} (\rho_{c1}(\text{CONNECTIONS}))) \cup \\
(\sigma_{c1.\text{FromStation}=\text{ZurichHB}} \bowtie_{c3.\text{ToStation}=\text{GenevaHB}} (\rho_{c1}(\text{CONNECTIONS}))) \cup \\
(\sigma_{c2.\text{ToStation}=\text{c3.\text{FromStation} \land c2.\text{ItNr}=c3.\text{ItNr}}} \rho_{c2}(\text{CONNECTIONS})) \cup \\
(\sigma_{c2.\text{ToStation}=\text{c3.\text{FromStation} \land c2.\text{ItNr}=c3.\text{ItNr}}} \rho_{c2}(\text{CONNECTIONS}))
\]

This is a union of direct connections with the connections with one intermediate station and connections with two intermediate stations. The general case of an indeterminate number of intermediate stations is not covered.

3 Relational Algebra - Task 3

\[
\begin{align*}
R \bowtie S & = \Pi_{R \cup S}(R - \Pi_{R}(R \times S)) \cup (R \times S) \\
R \bowtie S & = \Pi_{R \cup S}(S - \Pi_{S}(R \times S)) \cup (R \times S) \\
R \bowtimes S & = \sigma_{\text{p}}(R \times S) \\
R \bowtimes S & = (R \times S) \cup (\Pi_{R \cup S}(R - \Pi_{R}(R \times S))) \cup (\Pi_{R \cup S}(S - \Pi_{S}(R \times S)))
\end{align*}
\]

4 SQL - Task 1

a) Which are the last names of the readers in Zurich?

SELECT DISTINCT Surname FROM Reader WHERE City = 'Zurich' ORDER BY Surname DESC
b) Which books (Author, Title) are from publishers in Zurich, Bern or New York?

\[
\text{SELECT Author, Title}
\text{FROM Book B, Publisher P}
\text{WHERE B.PublisherName = P.PublisherName}
\text{AND (P.PublisherCity = 'Zurich')}
\text{OR P.PublisherCity = 'Bern'}
\text{OR P.PublisherCity = 'New York')}
\]

c) Which books (Author, Title) has the reader Lemmi Schm"oker borrowed?

\[
\text{SELECT B.Author, B.Title}
\text{FROM Reader R, Loan L, Book B}
\text{WHERE R.Surname = 'Schm"oker'}
\text{AND R.Firstname = 'Lemmi'}
\text{AND R.RDNR = L.ReaderNr}
\text{AND L.ISBN = B.ISBN}
\]

d) Which books in the category "Alps" do not belong to the category "Switzerland" at the same time? Do not take into account subcategories!

\[
\text{(SELECT ISBN}
\text{FROM BookCategory}
\text{WHERE CategoryName = 'Alps')} \text{ EXCEPT}
\text{(SELECT ISBN}
\text{FROM BookCategory}
\text{WHERE CategoryName = 'Switzerland')}
\]

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

\[
\text{SELECT R.Firstname, R.Surname}
\text{FROM Reader R, Loans L, Book B, Publisher P}
\text{WHERE R.RDNR = L.ReaderNr}
\text{AND L.ISBN = B.ISBN}
\text{AND B.PublisherName = P.PublisherName}
\text{AND R.City = P.PublisherCity}
\]

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

\[
\text{SELECT R1.Firstname, R1.Surname}
\text{FROM Reader R1, Loan L1, Loan L2, Reader R2}
\text{WHERE R2.Firstname='Lemmi'}
\text{AND R2.Name = 'Schm"oker'}
\text{AND L2.ReaderNr = R2.RDNR}
\]
5 SQL - Task 2

Given the relational schema in the first exercise express the following questions in SQL

a) List all the publishers and their respective books.

```
SELECT Publishername, Title
FROM Book B RIGHT OUTER JOIN Publisher P ON B.PublisherName = P.PublisherName
(ORDER BY B.PublisherName ASC, Title ASC);
```

b) Which book has the maximum number of pages?

```
SELECT Title
FROM Book
WHERE NoPages IN
(SELECT MAX(NoPages) FROM Book);
```

This query can output several records in the case when there are many books which have the maximum number of pages.

c) Which authors have written more than 5 books?

```
SELECT Author, COUNT(Title) AS number_books FROM Book
GROUP BY Author
HAVING number_books > 5
```

The number of books per author can be determined only after the grouping. Thus, HAVING is necessary in this case.

d) Which book has more pages than twice the average of the number of pages of all books?

```
SELECT Title
FROM Book
WHERE NoPages >= 2* (SELECT AVG(NoPages) FROM Book);
```

e) Which categories do not have any subcategories?

```
SELECT C1.CategoryName FROM Category C1 WHERE NOT EXISTS
(SELECT CategoryName
FROM Category C2
WHERE C2.BelongsTo = C1.CategoryName);
```

Alternative solution:

```
SELECT C1.CategoryName
```
FROM Category C1 LEFT OUTER JOIN Category C2 ON C1.CategoryName = C2.BelongsTo
WHERE C2.CategoryName IS NULL

f) Which author has written more books? (*) First we build a table A with the number of books for each author and then we find the maximum in this table.

SELECT A.Author FROM
(SELECT Author, COUNT(Title) AS number_books FROM Book
GROUP BY Author) A
WHERE A.number_books = (SELECT MAX(number_books) FROM A);

Alternative solution with HAVING:

SELECT Author, COUNT(Title) AS number_books FROM Book
GROUP BY Author
HAVING number_books >= ALL(
SELECT COUNT(Title) FROM Book
GROUP BY Author)

This solution requires that no reader has borrowed two copies of the same book. If we cannot assume this, the problem can be solved by COUNT (DISTINCT ISBN) or with a subquery that for every reader finds he different ISBNs of the books he has borrowed.

g) Which reader has borrowed all the books (by ISBN, not copies) from the author "Ephraim Kishon"?(*)

SELECT Firstname, Lastname
FROM Loan, Reader, Book
WHERE Reader.RDNR= Loan.ReaderNr AND
Author = 'Ephraim Kishon'
GROUP BY Firstname, Lastname
HAVING COUNT(Loan.ISBN) =
(SELECT COUNT(ISBN) FROM Book
WHERE Author = 'Ephraim Kishon')

This solution requires that no reader has borrowed two copies of the same book. If we cannot assume this, the problem can be solved by COUNT (DISTINCT ISBN) or with a subquery that for every reader finds he different ISBNs of the books he has borrowed.

h) For which of the books there is at least one copy available? (*) It is assumed that in the Loan table there are only the loans. Otherwise, it would be necessary to filter the loans by the return date.

SELECT Title FROM Book WHERE ISBN IN
(SELECT ISBN FROM
(SELECT CopyNumber, ISBN FROM Copy)
EXCEPT
(SELECT Copy, ISBN FROM Loan));

From the existing copies we select only those that are not in the loans.
Alternative solution:

```sql
SELECT ISBN
FROM Copy C GROUP BY C.ISBN HAVING COUNT(*) >
(SELECT COUNT(*)
FROM Loan L

The copies of books are grouped by ISBN and the number of copies for each ISBN is compared with
the number of copies that are borrowed for the same ISBN. The query can be rewritten differently
so that the copies and the loans are grouped separately and then joined by the predicate L.number < C.number.

i) Which are the ten oldest books? (**)  

```sql
SELECT ISBN, Author, Title FROM Book
ORDER BY PubYear TOP 10 -- (SQL Server)
LIMIT 10 -- (MySQL, PostGres)
LIMIT 0,10 -- (MySQL, PostGres)
FETCH FIRST 10 ROWS -- (DB2)

```sql
SELECT ISBN, Author, Title FROM
(SELECT ISBN, Author, Title FROM Book
ORDER BY PubYear ) WHERE ROWNUM < 10 -- (Oracle)

Unfortunately, there is no single syntax.

j) Which are all the subcategories of the "Sport" category (direct or non-direct categories)? (**)  

```sql
(SELECT CategoryName
FROM Category
CONNECT BY PRIOR CategoryName = BelongsTo START WITH BelongsTo = 'Sport')
EXCEPT
(SELECT CategoryName
FROM Category
WHERE CategoryName = 'Sport');

This solution is specific for Oracle. In DB2 or SQL Server there are other approaches available. At
a known fixed depth of the hierarchy the problem can be solved similarly as in the lectures.

6 SQL - Task 3

Formulate in SQL the following modifications to the database of the first exercise:

a) The reader Max Muster borrows the copy with CopyNumber 4 of the book with ISBN 123456.

```sql
INSERT INTO Loan (ReaderNr, ISBN, Copy) SELECT RDNR, 123456, 4
FROM Reader
WHERE Firstname = 'Max' AND Surname = 'Muster'
```
b) Delete the books that are published after 2013.

    DELETE FROM Book WHERE PubYear > 2013

c) Change the return date of all the books in the category "Databases" that should be returned before 15.03.2013 so that they can be kept for 30 days longer (Assume that you can add days to dates in SQL).

    UPDATE Loan SET ReturnDate = ReturnDate + 30 WHERE ReturnDate < '2013-03-15'
    AND ISBN IN
    (SELECT ISBN
    FROM BookCategory
    WHERE CategoryName = 'Databases')