

Output Formatting

January 31, 2016

- Put all the required files of Week 1 in a folder named A1.
- for Exercise 1, create a text file **sol1**. Explain freely your answers.
- for Exercise 2, two files:
 - a text file **sol2-explain**, containing the explanations
 - a text file **sol2-iterations**, with format specified in Appendix A
- for Exercise 3, a latex file **sol3**, representing the relational operators as in Appendix B.
- for Exercise 4, two files per query, as explained in Appendix C.

Zip the A1 folder and submit it on Moodle.

Appendix A

For each step, add a line containing **Step i** where i starts from 1 and increase by 1 at each step, followed by a the list of pairs cause/effect listed in lexicographic order. Each pair is given on a separate line, as shown in the following example:

```
Step 1
x y
y t
Step 2
x t
```

Appendix B

```
\documentclass[11pt]{article}
\usepackage[utf8]{inputenc}
\usepackage{amssymb}
\begin{document}
\begin{itemize}
\item natural join:  $\Join$ 
\item cross product:  $\times$ 
\item intersection:  $\cap$ 
\item difference:  $-$ 
\item renaming:  $\rho(R, \sigma_{\text{color}='red'}(\text{Parts}))$ 
\item selection:  $\sigma_{\text{color}='green' \wedge \text{cost} < 100}(\text{S})$ 
\item projection:  $\pi_{\text{pname}, \text{color}}(\text{Parts})$ 
\end{itemize}
\end{document}
```

If you have not latex installed on your computer, you can use the compiler available at <http://latex.informatik.uni-halle.de/latex-online/latex.php>. The the content of the PDF file generated by the above source should look like in the following page.¹

¹Beware, when copy-pasting the above source, underscores may be erased.

- natural join: \bowtie
- cross product: \times
- intersection: \cap
- difference: $-$
- renaming: $\rho(R, \sigma_{color='red'}(Parts))$
- selection: $\sigma_{color='green' \wedge cost < 100}(S)$
- projection: $\pi_{pname,color}(Parts)$

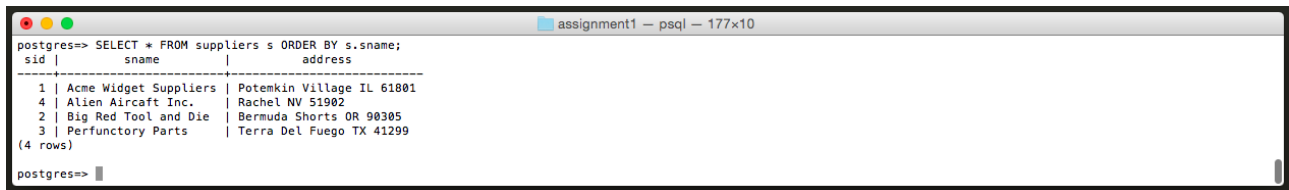
Appendix C

For every query in Exercise 4, create two files in the folder A1:

- One file named *QX* with the extension *sql* where *X* is the number of the query (*Q1.sql*, *Q2.sql*, ..., *Q11.sql*, *Q12.sql*) that will contain the actual query that solves the problem.
- Another file named *QX* with the extension *res* where *X* is the number of the query (*Q1.res*, *Q2.res*, ..., *Q11.res*, *Q12.res*) that will contain the results of the query. Just copy and paste these directly from PostgreSQL.

Example:

Let's say that for a fictitious Query 13 you required to provide the distinct names of all supplies (regardless of whether they currently supply any parts at all or not) and print the output sorted alphabetically.



The screenshot shows a PostgreSQL terminal window titled "assignment1 - psql - 177x10". The prompt is "postgres=>". The query entered is "SELECT * FROM suppliers s ORDER BY s.sname;". The output is a table with three columns: "sid", "sname", and "address". The data is as follows:

sid	sname	address
1	Acme Widget Suppliers	Potemkin Village IL 61801
4	Alien Aircraft Inc.	Rachel NV 51902
2	Big Red Tool and Die	Bermuda Shorts OR 90305
3	Perfunctory Parts	Terra Del Fuego TX 41299

The terminal also shows "(4 rows)" and the prompt "postgres=>" again.

For this query, you would submit two files:

The 1st file (*Q13.sql*) would contain this:

```
SELECT * FROM suppliers s
ORDER BY s.sname;
```

The 2nd file (*Q13.sql*) would contain this:

```
sid |      sname      |      address
-----+-----+-----
1 | Acme Widget Suppliers | Potemkin Village IL 61801
4 | Alien Aircraft Inc.   | Rachel NV 51902
2 | Big Red Tool and Die  | Bermuda Shorts OR 90305
3 | Perfunctory Parts    | Terra Del Fuego TX 41299
```