DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Databases Project – Spring 2017

Contents

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Introduction

In this project the students will get a set of data files. Based on that data, they will i) design a database schema, ii) load the data into a DBMS, iii) write and optimize queries, and, finally, iv) implement an interface that will access the database and offer an interactive experience querying a given dataset.

IMPORTANT: Read the whole document before starting doing any work.

Short Description of Project

The dataset contains data about comic books. The project is done in teams of 3 people.
The project is separated into 3 milestones, which follow the material taught in the lectures. We have synchronized each milestone with the material of the lectures for your convenience.

The first milestone requires you to analyze the dataset and extract the E-R (Entity-Relationship) schema as well as getting acquainted with a DBMS.

The second milestone requires you to express a simple set of queries on top of the loaded database. The goal of this part is to familiarize with data loading and the challenging task of data cleaning. You will also get to apply your SQL skills, and get a first intuition about how query performance is directly dependent on i) the way you formulate a query and ii) the logical and physical design of your database. Simultaneously, during this milestone you have to design a first version of the interface to query the dataset.

Finally, in the third part of the project you will express a set of more sophisticated SQL queries, which you will also analyze to come up with a detailed description of the execution. In addition, during this milestone you will have to **fully implement** the interface.

For each of these milestones the students should prepare a document following the provided template which describes the completed work. The grading will be done based on the final report as well on a presentation and short discussion with the TAs. The final report should contain material about all the work done for the 3 milestones combined into one document**. The intermediate reports after each milestone are optional, while only the final deliverable is mandatory and gradable**. **However, only the teams that submit the intermediate reports will get feedback on their progress.**

IMPORTANT: Only the teams that deliver the intermediate milestone deliverables within deadline will receive feedback!

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 1: Create ER model, Design & Create Schema

Deadline (to get feedback): 27/03/2017

The students will use the data from the following data files:

- Story
- Issue
- Series
- Indicia_Publisher
- Publisher
- Brand
- Issue_Reprint
- Story_Reprint
- Story_Type
- Series_Publication_Type
- Language
- Country

The goal of this deliverable is to design an ER model and a corresponding relational schema, and create the database tables in a database system. The organization of the data in files and the given description ***DOES NOT IMPLY*** an ER model or a relational schema. It is given to help the student understand the format of the data faster. Finally, a discussion about constraints and removing redundant information should be included in the project report.

In the 1st deliverable the students should:

1. Create an ER model for the provided data.
2. Design the database and the constraints needed to maintain the database consistent.
3. Provide the SQL commands to create the tables in a relational database system.
4. Describe their work in the form of a report which should contain an ER diagram, SQL DDL code for table creation, description of the data constraints, and justification of the design choices (in a few paragraphs). The report should be submitted as a single pdf file (**one pdf per group**).

Important Note: Before designing an E-R schema, understand the data and read carefully the notes given in the form of **FAQ** at the end of the project description. If you need any clarifications, ask the TAs during the project session or office hours.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 2: Import Data. Basic SQL queries

Deadline (to get feedback): 01/05/2017

In this phase, students have to import the provided raw data into their database. Besides this initial loading phase, the students should accommodate the insertion of new data into any table using a user interface. Through this interface, users should also be able to perform simple queries over the data, e.g., search for a keyword in any table.

Besides this insert and search functionality, students have to implement the following queries in SQL:

a) Print the brand group names with the highest number of Belgian indicia publishers.
b) Print the ids and names of publishers of Danish book series.
c) Print the names of all Swiss series that have been published in magazines.
d) Starting from 1990, print the number of issues published each year.
e) Print the number of series for each indicia publisher whose name resembles 'DC comics'.
f) Print the titles of the 10 most reprinted stories
g) Print the artists that have scripted, drawn, and colored at least one of the stories they were involved in.
h) Print all non-reprinted stories involving Batman as a non-featured character.

In summary, in the 2nd deliverable the students should:

1. Parse the given data and import them in the created database as described in your 1st deliverable.
2. Implement (using SQL) the queries described above.
    a. Note: Consider the use of indexes to accelerate long-running queries.
3. Start working on the interface to access and visualize the data. A website or a java application are good choices, but students are free to choose any technology they want.
    a. More information can be found on the "Interface" section of this document
    b. For this deliverable, some mockup screenshots of an interface and a first version of the search /insert queries it triggers in the backend suffice.
4. Extend the project report from the first deliverable with the description of the work done for the second deliverable and an explanation for the design choices. Include any changes to the design covered in the first deliverable, with justification of the changes. Include the screenshot of the interface and the description of the way search functionality is implemented in the application. The report should be submitted as a single pdf file.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 3: Interesting SQL queries

Deadline: 02/06/2017

A series of more interesting queries should be implemented with SQL.

a) Print the series names that have the highest number of issues which contain a story whose type (e.g., cartoon) is not the one occurring most frequently in the database (e.g, illustration).
b) Print the names of publishers who have series with all series types.
c) Print the 10 most-reprinted characters from Alan Moore's stories.
d) Print the writers of nature-related stories that have also done the pencilwork in all their nature-related stories.
e) For each of the top-10 publishers in terms of published series, print the 3 most popular languages of their series.
f) Print the languages that have more than 10000 original stories published in magazines, along with the number of those stories.
g) Print all story types that have not been published as a part of Italian magazine series.
h) Print the writers of cartoon stories who have worked as writers for more than one indicia publisher.
i) Print the 10 brand groups with the highest number of indicia publishers.
j) Print the average series length (in terms of years) per indicia publisher.
k) Print the top 10 indicia publishers that have published the most single-issue series.
l) Print the 10 indicia publishers with the highest number of script writers in a single story.
m) Print all Marvel heroes that appear in Marvel-DC story crossovers.
n) Print the top 5 series with most issues
o) Given an issue, print its most reprinted story.

In total, in the 3rd deliverable the students should:

1. Accommodate all above queries by giving the corresponding SQL code.
    a. Note: Consider the use of indexes to accelerate your long-running queries.
2. Explain the necessities of indexes based on the queries and the query plans that you can find from the system (you are free to select any 3 queries you like from the queries of the 3rd deliverable).
3. Report the runtime of all queries in milliseconds and explain the distribution of the cost (based again on the plans) for the 3 queries selected in part 2.
4. Present the results of the queries.
5. Build an interface to run queries/insert data/delete data giving as parameters the details of the queries. **Read the "Interface" section of this document for more details.**
6. Complete the project report written for the previous deliverables by adding description of the queries and the interfaces, explanation for the design choices, analysis of the chosen queries, as well as the changes compared to the work described in the previous deliverables. The report should be submitted as a single pdf file.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

"Grand Comics Database" data description

In this section, we present the data on which the project is based. Read carefully the data description, the FAQ and if in doubt ask the TAs for clarification.
You can also consult docs.comics.org, which describes the original data source from which we derived the dataset. Do note that i) we have removed some of the fields described from the files we provided you, and that ii) the schema described in this website CANNOT BE ASSUMED TO BE CORRECT.
The data is stored in CSV (comma separated values) files.

Story

This file describes stories that have been featured in comic books.

1. id
   The unique identifier of the story record.
2. title
   The title under which the story was published.
3. feature
   The name(s) of the feature(s), if any—usually the name of the primary character(s).
4. issue_id
   The issue in which the story was published (connection to the Issue file).
5. script
   The story author(s).
6. pencils
   The artist(s) who did the drawings.
7. inks
   The artist(s) who did the inking.
8. colors
   The artist(s) who added color to non-colored artwork.
9. letters
   The creator(s) or studio(s) that did the lettering/typesetting.
10. editing
    Editing details that are specific for the story.
11. genre
12. characters
    The character(s) appearing in the story.
13. synopsis
14. reprint_notes
    Textual reprint information not modelled elsewhere in the database.
15. notes
16. type_id
    The story type (connection to the Story_Type file).

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Issue

This file describes the actual "physical" comic book issues in which stories are published.

1. id
   The unique identifier of the issue record.
2. number
   The issue number or other identifier from the indicia, the cover, or both.
3. series_id
   The series in which the issue was published (connection to the Series file).
4. indicia_publisher_id
   The indicia publisher (connection to the Indicia_Publisher file).
5. publication_date
6. price
7. page_count
8. indicia_frequency
   The publication frequency.
9. editing
   The issue-level editor and other credits.
10. notes
    Arbitrary notes about the entire issue.
11. isbn
12. valid_isbn
    The ISBN of the issue in a valid form.
13. barcode
14. title
    The title of the issue.
15. on_sale_date
    The date the issue went on sale.
16. rating

Series

This file describes the series of comic book stories that may exist (e.g., V for Vendetta).

1. id
   The unique identifier of the series.
2. name
   The name of the series.
3. format
   The description of the physical format of the issues in the series.
4. year_began
5. year_ended
   The first and last (if any) years of publication
6. publication_dates
   First and last (if any) full cover publication dates of the series, separated by a hyphen (i.e. '-').

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

7. first_issue_id
8. last_issue_id
   The ids of the first and the last (if any) issues in the series (connection to the Issue file).
9. publisher_id
   The publisher of the series (connection to the Publisher file).
10. country_id
    The country in which the series was published (connection to the Country file)
11. language_id
    The language in which the series was published (connection to the Language file)
12. notes
13. color
14. dimensions
15. paper_stock
16. binding
    Color, dimensions, paper stock and binding information about the issues in the series.
17. publishing_format
18. publication_type_id
    The type of publication (connection to the Publication_Type file).

Indicia Publisher

This file describes the actual official company or person who published the book, as opposed to an informal (**but commonly used name**) for a publisher. For example, while Marvel is a well-known ***publisher***, it actually corresponds to ***multiple indicia publishers***, such as "Marvel Comics Group", "Marvel Publishing, Inc.", "Zenith Books, Inc.", etc.

1. id
   The unique identifier of the indicia publisher.
2. name
   The name of the indicia publisher.
3. publisher_id
   The corresponding master publisher (connection to the Publisher file).
4. country_id
   The country of the indicia publisher (connection to the Country file).
5. year_began
6. year_ended
   The years of the first and last (if any) publication.
7. is_surrogate
   A Boolean indicating whether the indicia publisher is a company related to the master publisher or an unrelated company who published on behalf of the master publisher.
8. notes
9. url
   The URL for the company's website, if and only if it is distinct from the master publisher website.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Publisher

This file holds information about publishers. These "master publishers" are the common names of publishers as typically grouped by comic book researchers (e.g., Marvel, DC, Dark Horse). Each one of them may correspond to multiple indicia publishers (i.e., official companies).

1. id
   The unique identifier of the master publisher.
2. name
   The name of the master publisher.
3. country_id
   The country of the master publisher (connection to the Country file).
4. year_began
5. year_ended
   The years of the first and last (if any) publication.
6. notes
7. url
   The URL for the publisher's website.

Brand Group

This file describes brand groups. A publisher holds multiple distinct brands, each identified as a brand group. For example, some of the brand groups under the ownership of Marvel are "Disney Comics" and "Marvel Universe Fantastic Four Group".

1. id
   The unique identifier of the brand group.
2. name
   The name of the brand group.
3. year_began
4. year_ended
   The years of the first and last (if any) publication.
5. notes
6. url
   The URL for the group's website, if and only if distinct from the master publisher website.
7. publisher_id
   The master publisher of the brand group (connection to the Publisher file).

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Story_Reprint
This file lists stories that have been reprinted. The origin story has been reprinted as the target story.

1. id
2. origin_id
3. target_id

Issue_Reprint
This file lists issues that have been reprinted.

1. id
2. origin_issue_id
3. target_issue_id

Story_Type
This file describes the types of stories in the dataset (e.g., photo story, comic story, text article).

1. id
2. name

Series_Publication_Type
This file describes the types of series publications.

1. id
2. name

Language
This file describes the languages in which comic book series have been written.

1. id
2. code
3. name

Country
This file describes the countries associated with a publisher or a comic book series.

1. id
2. code
3. name

You can find the data here: http://diaswww.epfl.ch/courses/db2017/project/comics.zip

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Interface

Here we describe the requirements for the interface. The goal of implementing this interface is to get hands-on experience with a technology used to connect and query a database from an application.

Functionality

In the context of this project we require the following functionality:

- Implement a query submission interface. This should not be a text box where SQL is typed. It should be a set of boxes and drop-down menus.
- The Query submission interface should be able to accommodate all the queries requested in the context of this project. Make sure to accommodate a way to input parameters without displaying SQL code to the user.
- The results of the queries should be printed in a user-friendly manner -- not just a console printout of the results. Some options would be either visualizing results in a page-based format (e.g., like the biography box in a wiki page) or clean column printouts (clear column separation and easily legible text).
- The search box. Add a word-based search box, which you can use to perform keyword-based search without necessarily knowing the schema of the data (e.g., search for 'Frank Miller'). The follow-up search functionality should allow the user to choose one of the results (e.g., by clicking on it or selecting a box next to it) and get more information about it.

Design

The design of the interface is left entirely up to you. As long as it covers the full functionality described above, the colors/size of windows, etc. does not play any role.

Implementation

Students are free to choose any technology they want. In the past, your colleagues usually opted for Java-based applications or a website.

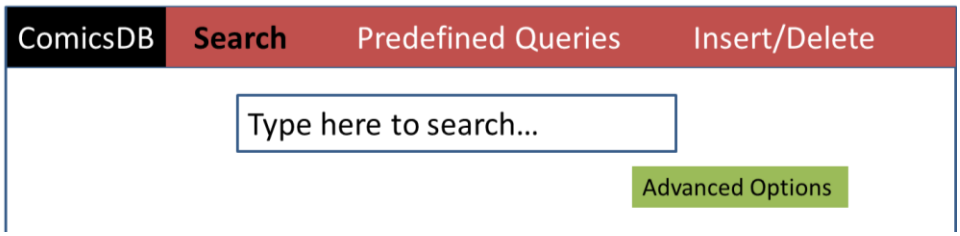
DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

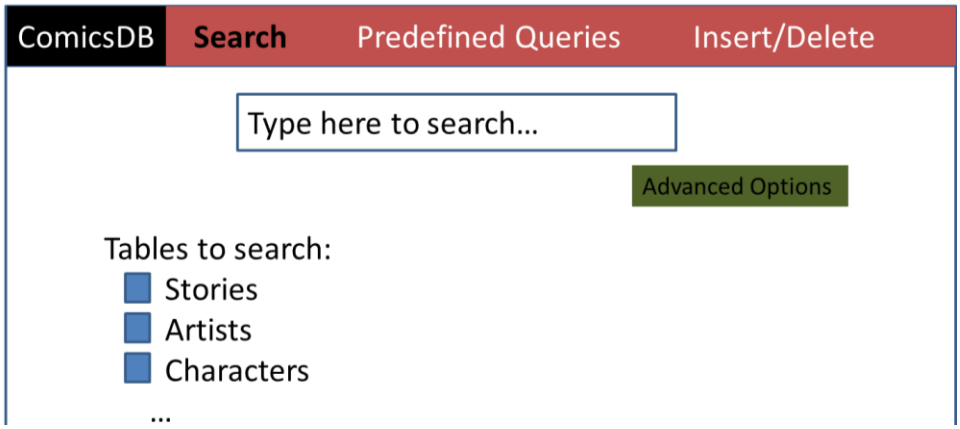
Interface Example

We now provide templates of the functionality expected from your interface. This template is meant as a mock-up of the three basic user interactions expected: i) search, ii) deliverable queries, iii) insertion/deletion. You are free to implement this functionality in any way you want.

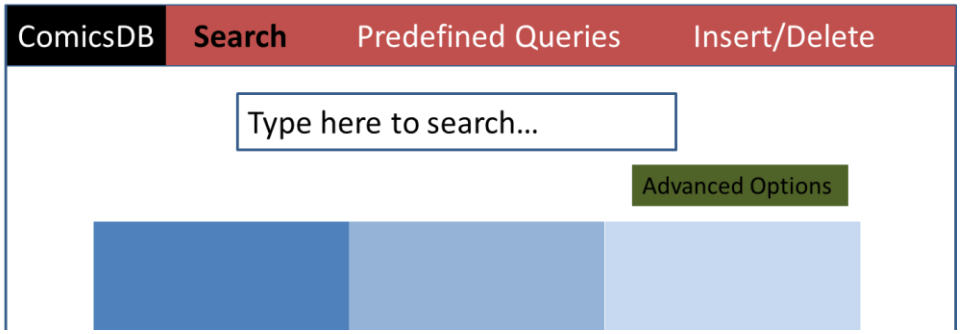
Basic Search Functionality



Advanced search options



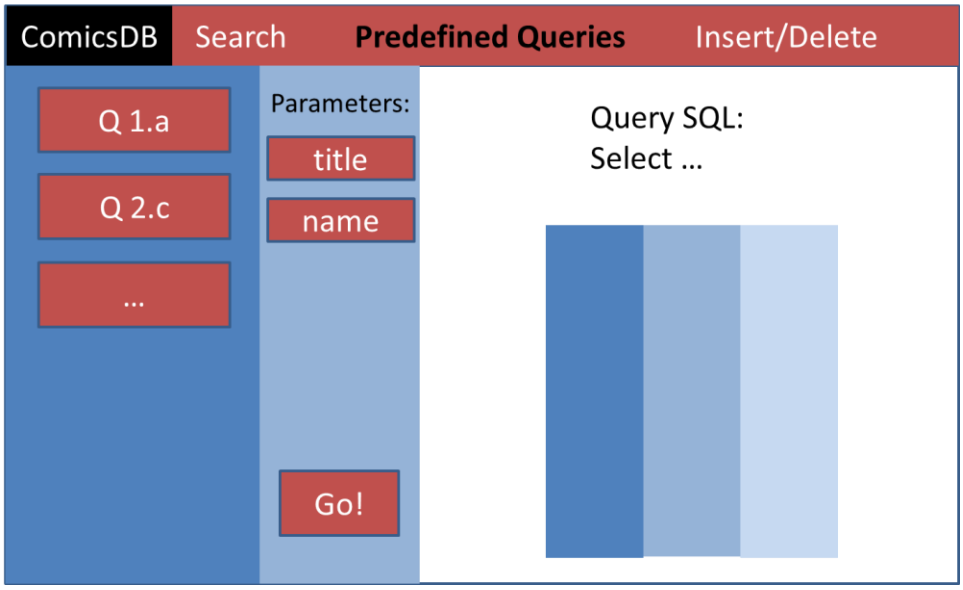
Result printing



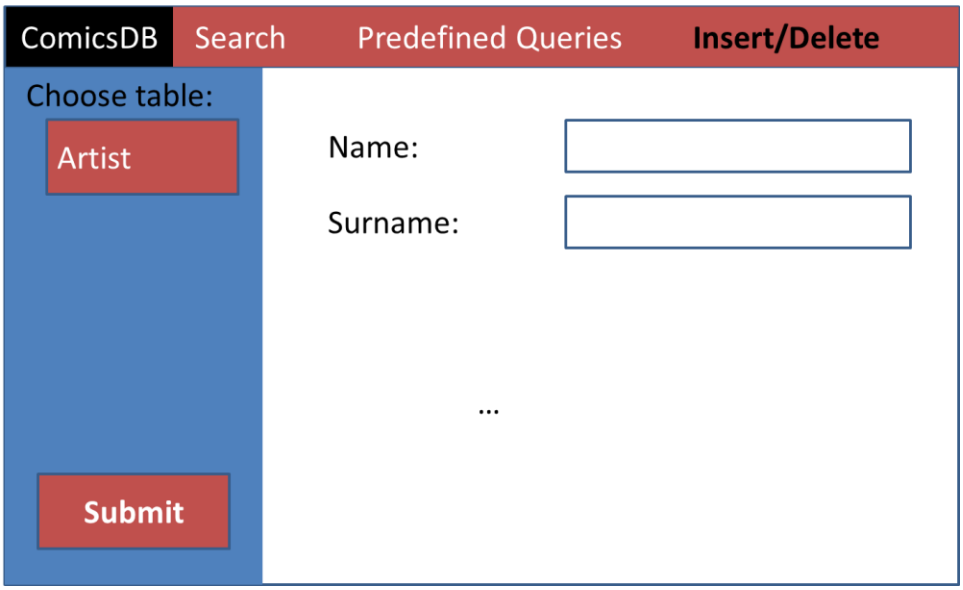
DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Predefined Queries



Insert Functionality



DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Frequently Asked Questions

How does one browse the data?

The dataset size is substantial, so it is hard to open most files using a notepad or text editor.
Applications such as Notepad++ and Sublime Text do a better job, but may still have issues with bigger files.
We thus also propose using Unix commands such as:

1. head: prints the first 50 lines of the file
2. less: allows backward movement in the file as well as forward movement
3. vi text editor: this editor does not open the whole file but only the part that is displayed

Which is the format of the given data?

The given data is CSV files (Comma Separated Values) which are values separated with comma (,). Each column represents a specific attribute. Usually in CSV files the name of the attribute is given in the first line of the file.

Why are the datasets "dirty"?

Real-world data is almost always dirty; missing values are commonplace; users abuse DBMS datatypes and store values based on their arbitrary, ad-hoc rules. We consider data cleaning to be a major part of your project. Regarding how to perform data cleaning, there is more than one correct solution. Some possible ways are the following:

- Use Unix commands such as sed, grep, awk.
- Use your favorite scripting language, or a typical program that handles data inconsistencies. For example, Python, Java, and Scala all feature CSV parsers which you can use to read and transform the data.
- Load the data in a DBMS and then use DBMS functions to transform them based on your requirements.

Which database system should I use?

You are free to use any DBMS you want. Typical open-source examples are MySQL and PostgreSQL. We will also grant you access to an Oracle installation located on a server of the DIAS lab, which you can use. We will do our best to troubleshoot any issues with the Oracle server and help with issues related to your own installations.

Which character encoding should I set?

All files use utf-8 encoding. Take care of initializing your database using the correct encoding before creating tables or loading the data.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

What should I do if it takes too long to load the data?

The two most common reasons for a slow data loading process are the following:

1. Defining too many indexes/foreign key relations in your tables can delay loading significantly. **We therefore propose that you first create simple tables with only primary key properties, or without any constraints specified at all**. Once data is loaded, add the more complex table relations and indexes.
2. If you are using the database system provided by us, make sure that you are connected to the epfl network via cable (i.e., use a machine from the laboratory). If you connect via wi-fi or from home via vpn, it takes a long time to upload the data files, thus leading to large loading times.

An additional scenario is that you have set up your own database server (e.g., PostgreSQL or MySQL), and that the default resources allocated to it are very few. In that case, some useful links are the following:

- https://dev.mysql.com/doc/refman/5.5/en/innodb-buffer-pool.html
- http://www.rathishkumar.in/2017/01/how-to-allocate-innodb-buffer-pool-size-in-mysql.html
- https://wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server

What should I pay attention to?

1. **There is no intermediate grading**
   a. We still urge you to complete the milestones on time, so that you will not be overwhelmed at the end of the semester.
   b. The parts of the project are created in a way so that you will use the things you learn in the course and the exercise session and have hands-on experience.
   c. Every one of your deliverables should include the text from the previous ones too, explicitly updated to reflect the changes you made to address the feedback we gave you.
2. **Collaboration**
   a. We want you to collaborate
   b. We DO NOT CARE how you will split the work -> As long as you do equal parts of the work
   c. Writing the queries can (and should!) be done by everyone!
      i. You can solve the queries in multiple ways to find the optimal one!
3. The only important deadline on which you are graded is the last one **BUT** if you want feedback make sure to send us the milestone deliverables!

What is more important? The user interface or the actual "database work"?

The course concentrates on data management, not on HCI or web-based development. Therefore, it is by far more important to us that you concentrate on writing efficient queries, tuning your system, and deciding on which indexing structures are the appropriate ones for your needs. We will give a full grade to any user interface that covers the basic functionality we request.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Can I discard some data?

Dropping some erroneous values is acceptable. Under no circumstances, however, should you drop a significant chunk of the data. Whenever you drop some data, you should include the description of the dropped data and the reason for doing so.

How long should the deliverables be?

There is no strict page limit, as long as the deliverables report on the points we requested and are informative.

How should I choose my team?

Putting teams together is entirely up to you. Our advice is that every team member should be exposed equally to every task of the project. While, for example, it might appear tempting to a good frontend developer to focus on the user interface and quickly finish her assigned task, she will then be disadvantaged in the course midterm and final, because her SQL and query optimization experience will be limited.

What should I do if one of my teammates does not work?

We advise that you address the issue early on, before you encounter high load due to a deadline. We cannot be more lenient to such teams as a whole for fairness reasons. During the final project presentation, however, it becomes obvious whether a team member did not place equal effort; this student will get a lower grade.

When can I ask questions about the project?

The weekly project session is the intended place for questions. Otherwise, please use the moodle forum for questions that are of interest to your colleagues too. Finally, every TA has specified office hours that you can use for further clarifications.