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/

Databases Project – Spring 2018

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 IMDB movies. 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 reports after the first two milestones are optional, while the final (3rd) 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): 26/03/2018

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

- Actors
- Directors
- Producers
- Writers
- Biographies
- Clips
- Clip-Links
- Genres
- Languages
- Countries
- Release Dates
- Running Times
- Ratings

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): 30/04/2018

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 name and length of the 10 longest clips that were released in France.
b) Compute the number of clips released per country in 2001.
c) Compute the numbers of clips per genre released in the USA after 2013.
d) Print the name of actor/actress who has acted in more clips than anyone else.
e) Print the maximum number of clips any director has directed.
f) Print the names of people that had at least 2 different jobs in a single clip. For example, if X has both acted, directed and written movie Y, his/her name should be printed out. On the other hand, if X has acted as 4 different personas in the same clip, but done nothing else, he/she should not be printed.
g) Print the 10 most common clip languages.
h) Print the full name of the actor who has performed in the highest number of clips with a user-specified type.

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. Provide the SQL code as well as the first 5 rows (when applicable) of the result for each query.
    b. 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/

Deliverable 3: Interesting SQL queries

Deadline: 01/06/2018

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

a) Print the names of the top 10 actors ranked by the average rating of their 3 highest-rated clips that where voted by at least 100 people. The actors must have had a role in at least 5 clips (not necessarily rated).
b) Compute the average rating of the top-100 rated clips per decade in decreasing order.
c) For any video game director, print the first year he/she directed a game, his/her name and all his/her game titles from that year.
d) For each year, print the title, year and rank-in-year of top 3 clips, based on their ranking.
e) Print the names of all directors who have also written scripts for clips, in all of which they were additionally actors (but not necessarily directors) and every clip they directed has at least two more points in ranking than any clip they wrote.
f) Print the names of the actors that are not married and have participated in more than 2 clips that they both acted in and co-directed it.
g) Print the names of screenplay story writers who have worked with more than 2 producers.
h) Compute the average rating of an actor's clips (for each actor) when she/he has a leading role (first 3 credits in the clip).
i) Compute the average rating for the clips whose genre is the most popular genre.
j) Print the names of the actors that have participated in more than 100 clips, of which at least 60% where short but not comedies nor dramas, and have played in more comedies than double the dramas. Print also the number of comedies and dramas each of them participated in.
k) Print the number of Dutch movies whose genre is the second most popular one.
l) Print the name of the producer whose role is coordinating producer, and who has produced the highest number of movies with the most popular genre.

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. After the introduced optimizations, 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/

"IMDB Movies 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. The data is stored in CSV (comma separated values) files.

Actors

This file outlines the actors associated with the clips.

1. FullName
   The actor's name.
2. ClipIds
   The list of clips (clips ids) that the actor participated in.
3. Chars
   The list of characters that the actor played in the given clips (e.g., Luke Skywalker).
4. OrdersCredit
   The list of orders in credits for the given clips (e.g., 2 is the second name mentioned in credits).
5. AddInfos
   Any additional information.

Directors

This file outlines the directors associated with the clips.

1. FullName
   The director's name.

6. ClipIds
   The list of clips (clips ids) that the director participated in.

2. Roles
   The list of roles that the director has in the given clips (e.g., co-director, series director, supervising director, segment director etc.).
3. AddInfos
   Any additional information.

Producers

This file outlines the producers associated with the clips.

1. FullName
   The producer's name.
2. ClipIds
   The list of clips (clips ids) that the producer participated in.
3. Roles
   The list of roles that the producer has in the given clips (e.g., executive, associate, co-producer, etc.).
4. AddInfos
   Any additional information.

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/

Writers

This file outlines the writers associated with the clips.

1. FullName
   The writer's name.
2. ClipIds
   The list of clips (clips ids) that the writer participated in.
3. WorkTypes
   The list of work types that the writer has done in the given clips (e.g., story, screenplay, novel, characters, scenario, etc.).
4. Roles
   The list of roles that the writer has in the given clip (e.g., writer, head writer, developer, etc.).
5. AddInfos
   Any additional information.

Biographies

This file outlines the biographies of actors, directors, producers and writers.

1. Name
   The person's artistic name/acronym.
2. RealName
   The person's real name.
3. Nickname
4. DateAndPlaceOfBirth
5. Height
6. Biography
7. Biographer
8. DateAndCauseOfDeath
9. Spouse
   The information about the spouse.
10. Trivia
11. BiographicalBooks
    The list of biographical books.
12. PersonalQuotes
13. Salary
14. Trademark
15. WhereAreTheyNow
    The current information.

Clips

The file that describes the clips (movies, games etc.).

1. ClipId
   The id of the clip.
2. ClipTitle

The title of the clip.

3. ClipYear
   The release year.
4. ClipType
   The type of the clip (TV – TV movie, V – video movie, VG – video game).

Clip-Links

This file lists the links between the clips.

1. ClipFrom
   The id of the first clip in the link/relationship.
2. ClipTo
   The id of the second clip in the link/relationship.
3. LinkType
   The type of a link between clips (e.g., follows/followed by, references/referenced in, etc.). For instance, the link type *referenced in* implies that ClipFrom is referenced in ClipTo.

Genres

This file lists the genres associated with a clip.

1. ClipId
2. Genre – e.g., short, drama, comedy, crime, adventure, action, fantasy etc.

Languages

This file lists the languages associated with a clip.

1. ClipId
2. Language

Countries

This file lists the countries associated with a clip.

1. ClipId
2. CountryName

Release Dates

This file outlines the release dates of a clip per country.

1. ClipId
2. ReleaseCountry
3. ReleaseDate

Running Times

This file outlines the running times of a clip per country.

1. ClipId

2. ReleaseCountry
3. RunningTime

Ratings

This file outlines the ratings of the clips.

1. ClipId
2. Votes
   The number of votes.
3. Rank
   The rank of the clip.

You can find the data here: http://diaswww.epfl.ch/courses/db2018/project/clips.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/

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.

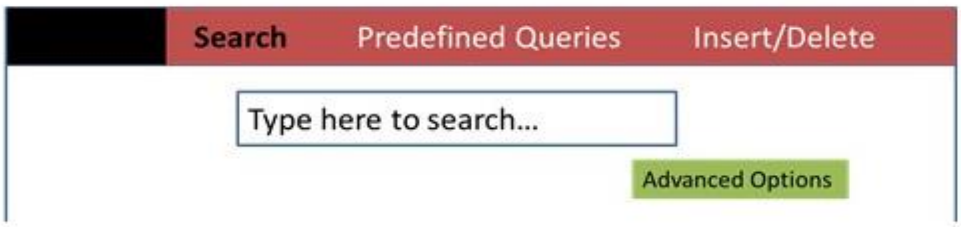
Implementation

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

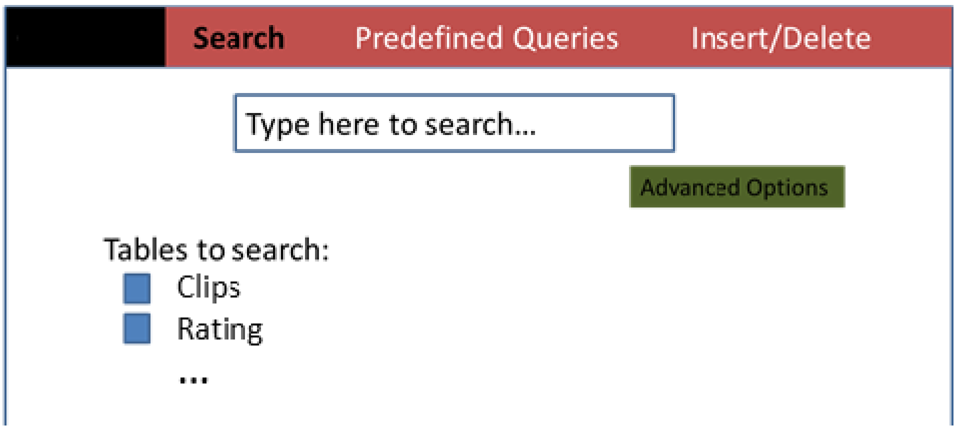
Interface Example

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.

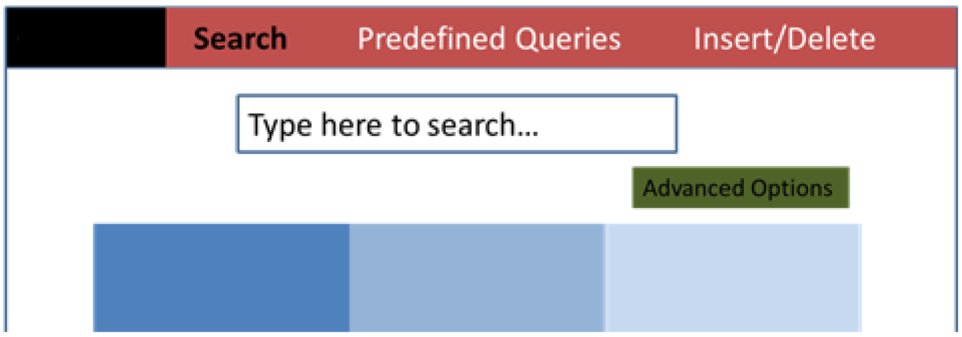
Basic Search Functionality



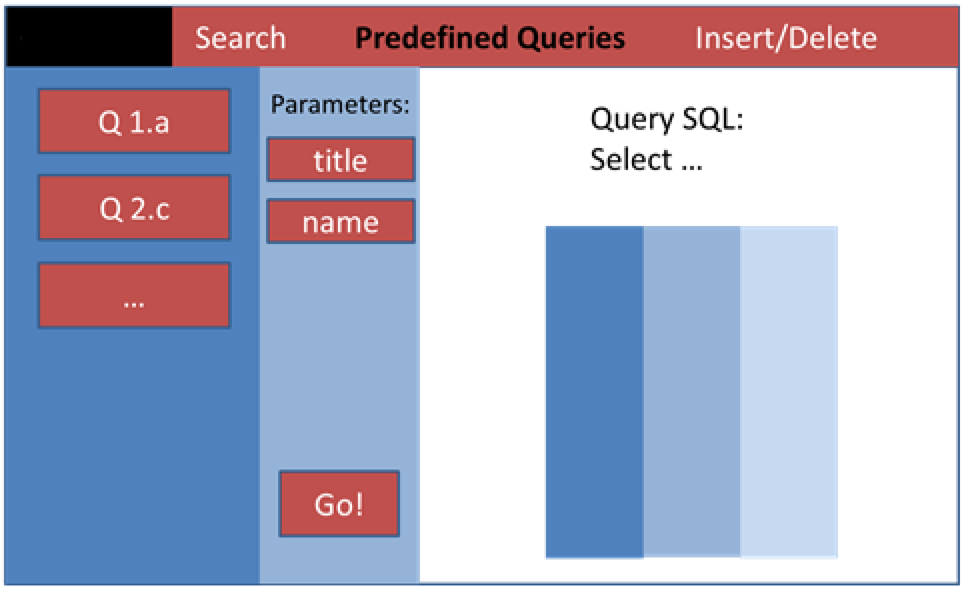
Advanced search options



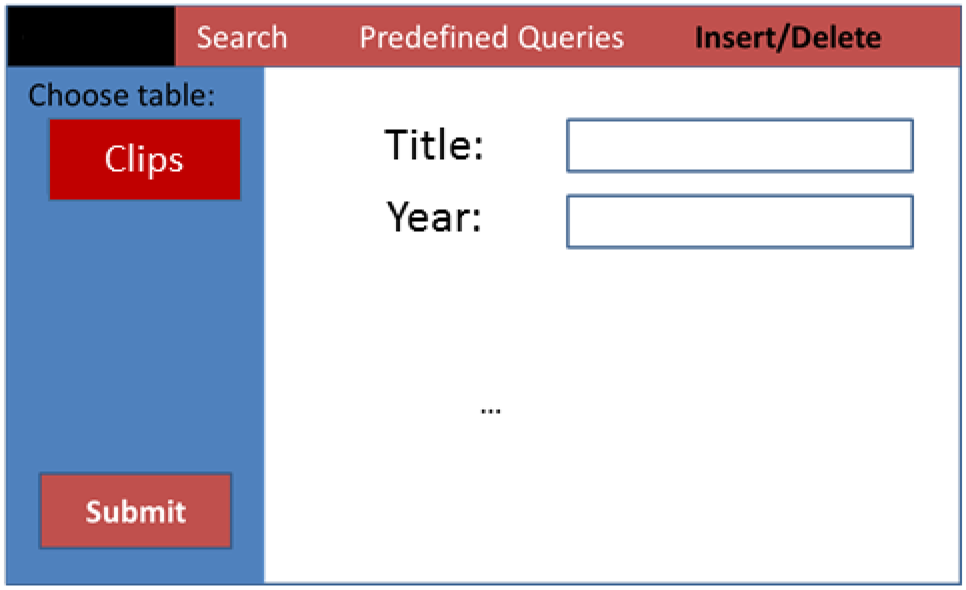
Result printing



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.

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.