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/

Project Description

CS-322 Introduction to Database Systems
Spring 2023

Table of 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/

Introduction

In this project you will get a set of files collected by a real-world entity – Yelp!, based on which you need to i) design the database schema and implement the relational schema, ii) discuss the data cleaning, iii) deploy the schema and load the data into a DBMS, iv) write queries, and, finally, v) evaluate and optimize queries with index structures/query plan analysis in order to analyze the performance impact on generated query plans and discuss the query optimizer decisions on querying the given dataset.

Therefore, the goal is to guide you through the design process from getting the unstructured raw data that needs organization, abstract reasoning about the entities and relations that exist, parsing and preparing the data for loading using the programming tools of your choice, to the point where this data is ready to be queried using a relational DBMS. This project simulates a business use case and synthesizes your programming and analysis skills in a practical task with a concrete end goal, along with practicing and implementing the theoretical principles acquired in this class.

IMPORTANT: Read the whole document before starting any work.

Short Description of Project

The dataset contains a subset of data from Yelp, a business directory and crowd-sourced review forum. The project is done in teams of 3 people. The project is separated into 2 parts, which follow the material taught in the lectures. We have synchronized each part of the project with the material of the lectures.

The first part of the project requires you to analyze the dataset and extract the ER (Entity-Relationship) model, translating it to a relational schema, and propose which elements of the dataset need to be cleaned and how it should be modified to follow your relational schema, with reasonable assumptions based on the available data.

The second part of the project starts with an ER model and relational schema that we will provide to you, with the data cleaned and ready to use. **You need to create the database on our Server (credentials will be provided to you) based on our specification,** load the data, and then continue with querying to find certain insights and optimize certain queries to get acquainted with DBMS and the query optimizer. We have thematically split the second part of the projects into Milestones 2 and 3. However, there is no separate deadline, and they will be graded together as Part 2.

For both project parts, you should prepare a document following the provided template describing the completed work. The grading will be done in two separate stages:

1) **First part of the project**: published on **29.02.2023**, report due on **10.04.2023**.
2) **Second part of the project**: published on **10.04.2023**, deadline **29.05.2023**.

We will grade you based on the reports, the state of your database on our server (schema+data), and a presentation/short discussion and Q&A with the TAs. The reports for part 1 and part 2 should contain material about all the work done, where the project deliverable template is is available on Moodle.

We strongly advise you to follow the proposed milestone deadlines, attend project or office hours sessions, and ask your assigned TA for feedback – the goal is to guide you through this process and help break down this project into manageable chunks, and get timely feedback before the project parts are due.

Not delivering the report or any relevant part of the deliverable on time (23:59 on the day due) incurs a penalty of 10% of points per hour. Being 10 hours late counts as 0 points for that part of the project. Delivering corrupt archives or files incurs the same penalty, so we recommend double-checking that your project submission can be downloaded, unpacked, and files opened.

IMPORTANT: You do not need to use the exact template document (.docx or .odt). However your submitted deliverable must include all the elements of the provided report. You are free to add sections or elements if needed.

Part 1: Create the ER model, Design & Create the Schema

Deadline: 10.04.2023 (Submission details will be on Moodle)

The goal of the first part of the project is to design an ER model and a corresponding relational schema. The organization of the data in files and the given description ***DOES NOT IMPLY*** an ER model or a relational schema (e.g., these are the only 4 entities of the ER model). The provided CSV files are the most convenient way to collect the data. However, you need to reason about the entities, relations, and how you can logically organize the data into self-sufficient actors in the business use-case. You must discuss necessary constraints (key, foreign key constraints, nullable values, and others) and understand why certain design choices remove repeated or redundant data points/attributes. This material is covered in the first weeks of the course and will allow you to start on time to analyze and provide the first version of your model.

In the **first part of the project**, you should:

1. Create an ER model for the provided data.
2. Create a relational model from the ER model.
3. Specify the resulting tables, keys, constraints of the relational model (can be DDL or another notation).
4. Explain (do not have to implement!) the data cleaning/transformation necessary to make your ER/relational model possible, and discuss the tradeoffs and why certain (good) practices may not be feasible with the given data.
5. Describe your work in the form of a report which should contain an ER diagram, relational model (tables, keys, 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 document per group**)

Important Note: Before designing the ER model, understand the data and read carefully the notes given in the form of **FAQ** at the end of the project description and the detailed data description. Ask the TAs during the project session or Moodle forum if you need any clarifications.

Tip: Analyze the data and remember that a column in CSV file does not always map to a column in entity/table. Remember that the column values must be atomic (not a list) in relational model (1st Normal Form). Some data columns may become separate tables for this reason. Feel free to group some values/attributes into a separate entity/table if they seem to **repeat** or appear to be logically a separate entity (explain your assumptions over the data and design decision). For example, a frequent design choice is a *star schema* – where attribute groups become entities/tables called *dimension* tables, while *fact* tables refer to them via foreign keys. From the high-level perspective, think first of the entities and relations based on attributes and their meaning.

Points breakdown for the first part of the project: 20 points for the ER model, 10 points for the relational model, 5 points for discussion on data cleaning/transformation. Total: 35% of the project grade.

Part 2, Deliverable 2: Import the Data, the first part of SQL queries

We will provide the starting schema + data on 10.04.2023 (Moodle)

In this phase, we will provide you with a common starting point with the schema and data to import, which you will use in the second part of the project. You have to implement the provided ER model into a relational model via SQL DDL and import the provided CSV data into the Oracle database (we will provide you the group user credentials). You should know how to insert/delete/update data via SQL DML commands, perform the bulk insert via commands or an IDE such as SQLDeveloper or DataGrip, and execute exploratory queries over the data.

You will have to implement and run SQL queries that **we will assign to you on 10.04.2023**.

In summary, in the 2nd deliverable you should:

1. Translate the **provided** ER model into a relational model
2. Implement the relational model via SQL DDL in the **provided database**.
3. Load the provided data (that is already cleaned, parsed, and split into appropriate tables).
4. Implement (using SQL) the assigned queries.
   a. Provide the SQL code and the first 20 rows (when applicable) of the result for each query.
   b. Make sure to output the necessary information, if a format is specified for the query.

Points breakdown for elements of this deliverable: 5 points for implementing the database on the server and loading the data, 15 points for the queries. Total: 20% of the project grade.

IMPORTANT: You must use the database and credentials we will provide your group (Oracle RDBMS). You can use any tool for development or connecting to the database (via JDBC connection).

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/

Part 2, Deliverable 3: Interesting and insightful SQL queries

Deadline (graded): 29.05.2023

A series of more interesting queries that provide more complex insights are to be implemented with SQL. In addition, the performance of **any 3 queries** should be optimized and analyzed in-depth using indexes and evaluated based on the produced query plans and their cost – compare the cost and plans before and after the optimization to justify the difference.

The queries to be implemented **will be assigned to you on 10.04.2023**.

In total, in the 3rd deliverable you should:

1. Implement queries by giving the corresponding SQL code.
    a. Provide the SQL code and the first 20 rows (when applicable) of the result for each query.
    b. Make sure to output the necessary information, if a format is specified for the query.
2. Select 3 queries from Deliverable 3, and accelerate them using indexes. Explain the necessities of indexes based on the queries and the query plans you can find from the system.
3. After the introduced optimizations, report the runtime of all queries in (milli)seconds and explain the distribution of the cost (based again on the plans) for the 3 queries, as well as the discussion based on the cost of the query plan – and how this plan has changed and why.
6. Complete the project report written for the previous deliverable (Deliverable 2) by adding a description of the queries, explanation for the design choices, analysis of the chosen queries, and the changes compared to the work described in the previous deliverables. The report should be submitted as a single PDF file.

Points breakdown for elements of this deliverable: 35 points for the queries, 10 points for optimization. Total: 45% of the project grade.

Yelp! 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.

yelp_academic_dataset_business.csv

This file contains information about the businesses listed on Yelp.

1. Address: the provided business address
2. Attributes: a list of business attributes/specializations. The attributes themselves can be further a list having descriptors of an attribute. **Tip:** there are overall about 6 unique groups in the dataset for attributes, that contain some number/a list of further descriptors. Make sure you follow 1st Normal Form when designing your ER model!
3. Business_id: the unique ID of the business
4. Categories: string representing a list of assigned categories. 1000+ possible unique values overall in dataset. Keep this in mind when parsing and designing your ER model!
5. City: name of the city
6. Hours: list of opening/closing hours per day
7. Is_open: indicates if the listed business is currently open for business
8. Latitude: geographical latitude
9. Longitude: geographical longitude
10. Name: listed name of the business
11. Postal_code: postal code of the business
12. Review_count: the aggregated number of reviews (does not have to match the actual ones in data)
13. Stars: the aggregated number of stars (does not have to match the actual ones in data)
14. State: the name of the state where business is located

yelp_academic_dataset_review.csv

This file contains information about the reviews that users leave on businesses.

1. Business_id: the unique ID of the business this review relates to
2. Cool: the number of users that rated this review as cool
3. Date: the date the review was posted
4. Funny: the number of users that rated this review as funny
5. Review_id: the unique ID of the review
6. Stars: the number of stars user has given to business related to the review
7. Text: the text description of the review (shortened)
8. Useful: the number of users that found this review useful
9. User_id: the unique ID of the user that wrote this review

Tip: As this is a CSV, meaning that the values are split by commas (“,”), and new line (“\n”) separates different rows, make sure to properly handle the **Text** field, as it may contain some of these values between its quotes!

yelp_academic_dataset_tip.csv

This file contains information about the tips (advice) users give about the businesses.

1. Business_id: the unique ID of the business this tip relates to
2. Compliment_count: the number of users that have complimented this tip
3. Date: the date this tip was posted
4. Text: the text description of the tip (shortened)
5. User_id: the unique ID of the user that wrote this tip

Tip: As this is a CSV, meaning that the values are split by commas (","), and new line ("\n") separates different rows, make sure to properly handle the **Text** field, as it may contain some of these values between its quotes!

yelp_academic_dataset_user.csv

This file contains information about the users.

1. Average_stars: average stars this user has received from other users for his reviews
2. Compliment_cool: the number of cool compliments this user received
3. Compliment_cute: the number of cute compliments this user received
4. Compliment_funny: the number of funny compliments this user received
5. Compliment_hot: the number of hot compliments this user received
6. Compliment_list: the number of list compliments this user received
7. Compliment_more: the number of more compliments this user received
8. Compliment_note: the number of note compliments this user received
9. Compliment_photos: the number of photos compliments this user received
10. Compliment_plain: the number of plain compliments this user received
11. Compliment_profile: the number of profile compliments this user received
12. Compliment_writer: the number of write compliments this user received
13. Cool: the number of cool votes sent by this user
14. Elite: list of the years in which this user has an elite status
15. Fans: the number of fans this user has
16. Friends: list of friends, whose elements are the user_id of the friends (**who are also users on Yelp**).
17. Funny: the number of funny votes sent by the user
18. Name: user's first name
19. Review_count: the number of reviews this user has written
20. Useful: the number of useful votes sent by the user
21. User_id: the unique ID of the user
22. Yelping_since: the date when the user joined Yelp

You can find the Project Part 1 data here (457.8MB, .zip) – good luck!

https://drive.switch.ch/index.php/s/43IygARM8SvyGMt

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, and sometimes even for spreadsheet document viewers.

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

A useful and recommended method is to browse the data using scripting languages such as Python, where you can use Pandas library to load the CSV as a DataFrame, and explore parts of data via the functions of the library. This way it is also useful to explore the data for future data cleaning, transformation, and loading to DBMS, and the library also provides a method to explore basic statistics and features of the data.

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 part of your project that you need to consider and think about how the data may be realistically transformed in part 1 of the project – but you will not need to clean the data yourself (you need to make realistic and reasonable assumptions). For the needs of this project, we will provide you with sufficiently cleaned data for part 2 (when you will need to load the data to DBMS)

Which database system should I use?

We will also grant you access to an Oracle installation located on a server at EPFL, **which you must use to submit the data, schema, and queries of part 2 of the project**. To access the Oracle database with the group accounts we provided to you (we will communicate this separately), you need to connect via EPFL network, therefore you need to use EPFL VPN. You can use any system/frontend that allows JDBC connection and file upload to the database backend (Oracle SQLDeveloper, JetBrains DataGrip, …).

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. **Therefore, we propose that you first create simple tables with only primary key properties, or without any specified constraints**. 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 VPN, it may take a long time to upload the data files, thus leading to longer loading times.

What should I pay attention to?

1. **There is no intermediate grading for the two parts of the project**
    a. We still urge you to complete the milestones (Part 2) on time, so that you will not be overwhelmed at the end of the semester.
    b. Discuss with your team and your assigned TA supervisor if you have any doubts or issues.
    c. The project's parts are created so that you will use the things you learn in the course and the exercise session and have hands-on experience.
2. **Collaboration**
    a. We want you to collaborate
    b. We DO NOT GO INTO 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 (and help the optimizer)
3. The only important deadlines are the ones on which you are graded. If you want feedback for intermediate work, ask us in the project sessions!

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 data analyst to focus on the data cleaning and loading and quickly finish their assigned task, they will then be disadvantaged in the course midterm and final, because their 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 a high load due to a deadline. We cannot be more lenient to such teams as a whole for fairness. 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. Please inform us in case there is a conflict or if your teammate decides to withdraw from the course, then we can try to address this issue.

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. Finally, every TA has specified office hours that you can use for further clarifications.