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 2022

Table of Contents

Introduction

In this project you will get a set of files collected by a real-world entity, based on which you need to i) design the database schema and implement the relational schema, ii) parse and load the data into a DBMS, iii) write queries, and, finally, iv) 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.

Project summary

The dataset contains data about comic books. With so many cinematic universes, different storylines, and potential clashes with the consistency of the stories and having appropriate rights for the use: this dataset can contain information valuable for decision-makers for what the next comic or movie should be. For example, a data analyst may want to know in which languages some comics were published or the most reprinted characters from the comics to analyze the popularity over time. While the reports have been digitized and provided in CSV files, these files are not ready to be queried. Some databases may support using the CSV files directly or using spreadsheet software, but this process is error-prone and cumbersome. Further, it does not scale to data that may be subsequently loaded for more fine-grained analysis, resulting in gigabytes of data. You will be given a subset of the data to test and implement the proof of concept database.

You have been hired to take these CSV files and produce a database that will enable analysts to get answers rapidly. The analysts have also provided you with a list of queries that they want to execute quickly. The TAs are your quality control managers and are there to provide feedback, ensuring that the final deliverable is up to the client's expectations: they want a database that can quickly answer their queries and is future-proof and scalable because it uses a relational DBMS.

The project is done in teams of 3 people. The project is separated into 2 graded parts with 3 milestones, which follow the material taught in 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 first part of the project corresponds to the Milestone/Deliverable 1, and is graded at the noted deadline.

The second part of the project starts with an ER model that we will provide to you, with the data cleaned for your convenience. **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, as well as optimize certain queries to get acquainted with DBMS and the query optimizer. We have split the second part of the projects into Milestones 2 and 3.

The second milestone requires you to express a set of queries on top of the loaded database. The goal of this milestone is to familiarize yourself with data loading and DDL. 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.

Finally, in the third milestone, 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 and propose ways to reduce query execution time. You will analyze the queries and their respective query plans in order to optimize the execution, either with building appropriate index structures or rewriting the queries to make the execution more efficient (or both), and discuss the decisions that the query optimizer took, such as if it even considered the newly created index structure.

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

1) **First part of the project**, Deliverable 1: published on **23.02.2022**, report due on **25.03.2022**.
2) **Second part of the project**, Deliverable 2 + Deliverable 3: published on **26.03.2022**, deadline **23.05.2022**.

We will grade you based on the reports, the state of your database on our server (schema+data), as well on 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 and 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 deliverables on time counts as 0 points for that part.

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.

First Part (Deliverable 1): Create the ER model, Design & Create the Schema

Deadline (graded): 25.03.2022 (Submission Via Moodle)

The goal of this deliverable 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. that these are the only 3 entities of the ER model). The provided CSV files are simply the way it is the most convenient to collect the data, however, you need to reason about what are the entities, relations, and how can you logically organize the data into self-sufficient actors in the business use-case. You need to 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 1st deliverable, **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 that would be necessary to make possible your ER/relational model, 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, as well as the detailed data description. If you need any clarifications, ask the TAs during the project session or office hours, or ask on Moodle forum.

Tip: Analyze the data and keep in mind that a column in CSV file does not always map to a column in entity/table. Remember that the column values have to 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. Think first of the entities and relations based on attributes and their meaning, from the high-level perspective.

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.

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

We will provide the starting schema + data on 26.03.2022 (Moodle)

Suggested deadline (not graded): 30.04.2022

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, or perform the bulk insert via commands or an IDE such as SQLDeveloper or DataGrip, as well as to execute exploratory queries over the data.

You will have to implement and run SQL queries that **we will assign to you on 26.03.2022**.

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 (Oracle).
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 as well as 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: 10 points for implementing the database on the server and loading the data, 10 points for the queries. Total: 20% of the project grade.

IMPORTANT: You need to use the database and credentials that we will provide your group (Oracle RDBMS). You can use any tool for development or for connecting to the database (via JDBC connection usually) – you will not need to install or set up a database yourself or use licensed software on your side.

Second part, Deliverable 3: Interesting and insightful SQL queries

Deadline (graded): 23.05.2022

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 by 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 26.03.2022**.

In total, in the 3rd deliverable you should:

1. Implement queries by giving the corresponding SQL code.
    a. Provide the SQL code as well as 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 by using indexes. Explain the necessities of indexes based on the queries and the query plans that 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, as well as 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.

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**
- **ii) the schema described in this website CAN NOT 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 to the story.
11. genre
12. characters
    The character(s) appearing in the story.
13. synopsis
14. reprint_notes
    Textual reprint information not modeled elsewhere in the database.
15. notes
16. type_id
    The story type (connection to the Story_Type file).

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. '-').

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 value which indicates whether the indicia publisher is a company related to the master publisher or an unrelated company that 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.

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).

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

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!

Tip: When there are more than 2 values in the possible value set (for example, something that is natural to be represented as a Boolean or an atomic value such as an integer or double), due to 1st Normal Form and value constraints it is common for some of these attributes to become dimensions that are separate entities. For example, this enables you to subsequently add new categories of attributes, or for example have a localization (display the text in a different language, simply by knowing which key corresponds to the text in a given language).

You can find the dataset here (.zip file, ~120MB): https://drive.switch.ch/index.php/s/I8fQKctZc6POjoc

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. **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 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 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 in case you have any doubts or issues.
   c. 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.
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, make sure to ask us reasonable questions in the project sessions or office hours!

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.