*SQL and Relational Databases, Project Report, 2023*

Project Excelsior

**Florian Kenner - 22208397**

(Florian.Kenner@ucdconnect.ie)

Use a splash page image here [optional]

Use *LaTeX* if you wish but use the general spacing and font/style you find here (1.5 spacing, 12-point font for text, etc.).

Be sure to submit a PDF (not a .DOC file) as your report. Overall, it will be about **25-30 pages**, including diagrams and screenshots. A significant portion of the report should be textual. Do not rely on images to write your report for you.

**Remember**, your project this year concerns a database for a food laboratory. Your database is intended to support the workings of the innovation processes within the laboratory. As such, identify the place of the database in the overall laboratory, and tell us how you would support its operations at the SQL level.

**What to submit**: This report, as a PDF, *and* the necessary SQL files to allow us to examine your database constructs and test your queries.

# Table of Contents

|  |  |  |  |
| --- | --- | --- | --- |
| Section 1: | Section Heading | ……………………………………………………………….. | 1 |
| Section 2: | Section Heading | ……………………………………………………………….. | 2 |
| Section 3: | Section Heading | ……………………………………………………………….. | 3 |
| Section 4: | Section Heading | ……………………………………………………………….. | 4 |
| Section 5: | Section Heading | ……………………………………………………………….. | 5 |
| Section 6: | Section Heading | ……………………………………………………………….. | 6 |
| Section 7: | Section Heading | ……………………………………………………………….. | 7 |

# List of Figures

|  |  |  |  |
| --- | --- | --- | --- |
| Figure 1: | Figure Caption | ……………………………………………………………….. | 1 |
| Figure 2: | Figure Caption | ……………………………………………………………….. | 2 |
| Figure 3: | Figure Caption | ……………………………………………………………….. | 3 |
| Figure 4: | Figure Caption | ……………………………………………………………….. | 4 |
| Figure 5: | Figure Caption | ……………………………………………………………….. | 5 |
| Figure 6: | Figure Caption | ……………………………………………………………….. | 6 |

# Introduction

The following project embodies my vision for the database of the comic retailer excelsior. The comic book industry relies on physical items, which have so far mostly been displayed on physical storefronts and sold in stores. The tradition and history of this domain must be adapted modern demands, in example online retailing. The heart of each online retailer is its underlying database system, which determines how the sold goods are being viewed, which aspects are emphasized and how all parts of the business come together. The main technical part of my vision is being as close to reality as possible when making design and implementation choices. The project should be lean, yet complex enough to consider the peculiarities and subtleties of comic collecting and selling so that every comic enthusiast would be satisfied with this design. Furthermore, the database should meet all technical criteria to be performant, non-redundant and scalable.

Comic books have become far more than simple entertainment products, which get read once and forgotten afterwards. Comic books have turned into collectables which carry both emotional and financial value. There are single very valuable pieces which got produced only a few times many years ago and even fewer which survived until today. At the same time there are many editions produces and sold today in large quantities. The difficult job of a comic retailer is to satisfy all these different target groups with their different needs and demands. Numerous traits are shared by all comics. These can be differentiated on the abstract level of a comic book batch and the level of each physical copy. On the abstract level, each comic book has a title, publisher, release date and so on. Each physical copy has a condition, which is rated by a uniform evaluation system, a buying and selling price, etc.. The customer must be provided with all relevant information so that he can make an educated buying decision and is satisfied in the long term. The intended application of the project is therefore providing a database which stores all information of the comic books the retailer excelsior sells on which a website can be built.

From a technical perspective, the different abstractions of the data have to be considered when designing the database. It can be expected that the stress on the database will be low at the beginning but may grow over time, but it’s not to be expected to enter the domain of big data. The database stores so far only textual information, meaning no pictures or videos. There won’t be an ingoing stream of telemetry data for example which fills up our database. Still, the database must be able to handle increasing traffic through website visitors and deliver results in a short time to provide a satisfying user experience.

The scope of this project is providing a useful solution for dealing with all the data concerning the stock of excelsior. This means keeping track of the current inventory of comics and providing the additional information an enthusiast needs. This additional information to what series a single comic issue belongs, who published this series, which character appears in a comic book and who worked on a comic book. Also all above mentioned relevant information of a single copy must be stored, such as the buying price, the selling price, the condition and the format, meaning the cover size and material. On top of that, the database should keep track of all customer information such as the name, date of birth and address. These two parts must be brought together to keep track of inventory changes to document previous purchases. The basis for two more functions on possible the way of a customer to buying a comic book should be provided with database foundations, a wish list and a shopping cart. Most customers are familiar with such functionalities from other online retailers and may therefore miss those in the online shop of excelsior if absent. Not part of the scope of this project is keeping track of logistic business information such as the physical storage of comics, the buying process of comics before they appear in our inventory and the shipping process. Also, no information regarding the payment except the prices should be stored.

# Database Plan: A Schematic View

For the database design, an intense and deep dive into the topic is very important to be able to understand the context of the data and how everything comes together. Therefore, the first step was collecting as much information as possible about comic book collecting in general. This concerned the releases of new issues and completely new comic book series, how these get numbered. Another important domain is understanding how comic books and graphic novels are related, which properties they share and what distinguishes them. Furthermore, the grading of collectables plays an important role in pricing comic books. The most important key learnings from the research which influenced the data base design were:

Each comic book has a title, a year of publication and an issue number. In a series of comics, e.g. „Iron Man (2002)“ all issues share the same title and only the issue number changes. Additionally, there might be multiple runs of a series with different content, there might be an “Iron Man (1990)” and an “Iron Man (2002)” which have different storylines. The numbering of comics may not be linear, it can be influenced if a character appears in another comic series and the original series may skip several issue numbers, how it comes to the numbering in detail is not important for this project. Graphic novels and comic books are closely related. Graphic novels are usually more extensive than comic books, they contain closed plot. A graphic novel may contain a collection of comic issues, such as “Iron Man (1990) #1-#5”, but graphic novels may also contain unique storylines which never got released as comics. Also, the edition of a graphic novel is very important to consider. There might be several reprints of the same graphic novel and only the first reprint could be of interest for comic enthusiasts. When it comes to the physical specimen of comic books and graphic novels, they share most attributes. There is a grading scale which applies to both which includes a numerical scale from 0.0 to 10.0 and a textual condition. There are numerical bins which result all in the same textual condition, for example comic books with a rating from 9.6 to 9.9 are graded with the condition “Near Mint/Mint” (NM/MT). To get a certain grade, a comic book has to fulfil certain criteria. The grading is done by independent companies and not by our comic book reseller excelsior, therefore we solely have to store this information about our stock.

Diagram

Description automatically generatedThe next step was to collect all requirements for this project. All mentioned features from the project white paper got collected and extended by research on mile high comics. The goal was carrying together, what information an online comic book retailer needed and derive the scope for this project from there. The resulting scope was presented in the first section of this report. The following entity relationship diagram is the core of this project and was designed by following the definitions given in "Database Systems: Design, Implementation, and Management" by Carlos Coronel.

Figure 1: Entity relation diagram

Let’s start by looking at the collectable entity. This entity abstracts the theoretical comic book or graphic novel. This level of abstraction describes all unique comic book issues or graphic novels, not referring to the physical specimen. Each collectable has the attributes title, storyline, year of publication and publisher. The storyline connects the two comic book series “Iron Man (1990)” and “Iron Man (2002)” since both tell the story of Iron man, but besides this share not much more. Each collectable has more traits, which are outsourced and displayed in the entities characters and creators.

The creators entity contains all people who worked on collectables, they posses a first name, a last name and a job type. The job type specifies what exactly their job was in example drawing or writing a story. Each collectable must have at least one creator who worked on it and each creator has worked on n collectables. As a consequence, there can be creators in our database, which have not worked on any collectables we store. This is a conscious decision, so we are able to store information about creators before we have any of their work in our data.

The other property a comic has which is stored in a different table are its character appearances. Each character has a name and a profession. In contrast to the creators, the character name consists only of one token which is implemented since characters might be stored not by referencing their real name, but their nickname such as “superman”. Each collectable has at least one character appearing, and each character can appear in n collectables, behind this lies the same design thought as behind the creators. These two entities are separated from the collectable entity since they can exist without it and are not a fixed part of them, in contrast the title is only part of a collectable and has no self-reliance apart from that. The same thought could be applied to the publisher, but since our scope is only to store his name and he has no further attributes in our database, he does not exist as an independent entity.

Now we move from the theoretical level of the collectable to the concrete physical of the stock. The stock entity represents all physical copies of comic books and graphic novels we sell. As mentioned before, one of their most important attributes is the condition, which consists of a numeric rating and a textual rating. Additionally, each stock item has a format, a buying and a selling price and an edition. The format refers to the size and material of a copy, which might be hard- or softcover, and the cover might be jumbo sized or medium sized and so on. The edition describes in which reprint the stock item was produced. Each stock item must be a collectable, so that all the information which does not belong to the single copy can be presented to the customer. Vice versa, each collectable can be n times in our stock, we might sell multiple copies of a collectable or none at a given time. Besides these traits, a stock item can have a comment in which more specific information is recorded. This information has no further restrictions, it could be about the condition or also tell if it's signed by a celebrity. From the entities collectables and stock items can be concluded, that the database does not store any information about the language of an item we sell. This choice was made based on the assumption, that the title of a collectible already specifies the language. The English comic book version of the amazing spiderman must have a different title than the German one.

All information needed about the goods we sell is can be stored in these entities, the following customer entity which hints at the online domain in which our store exists shows the general interactions our customers have with our web store. Each customer has a first and last name, a date of birth and an email address. Moreover, he has an address which consists of a street and house number, a city, a state, and a zip code. All this information should be self-explanatory, the customer entity is only a compressed abstraction of the real person interacting with our business and exists more for demonstration purposes. This customer may put any number of stock items on his wish list and every stock item can be on any number of wish lists. In the further process, there will be a more distinct separation of a wish list and a shopping cart, but for this model we can treat them both as the same. A customer either wants to buy an item which can be in his virtual shopping cart, his wish list and so on or he has bought an item. This is the second interaction displayed in the diagram. A customer may buy any number of stock items and logically, each stock item can only be bought once by a single customer.

The next step to designing a database concept is transforming these derived entities in a entity relationship diagram with tables, which shows how exactly they are stored in a relational database The following diagram displays this information and was made following the guidelines from the same book as the first diagram:

In this section offer a high-level view of the database and its design. State what you think the principal entities are, as well as their main attributes and the key relations that connect them. Provide an E-R diagram (entities and tables) that illustrates your plan. Motivate your design – state why this way and not another.

# Database Structure: A Normalized View

Describe the main tables in your database and the role played by each. Show that your database meets the definitions of 1NF, 2NF and 3NF normal forms. Is your database also in BCNF normal form? If so, explain how and why.

# Database Views

What views do you provide onto your database? Justify each and define them here. Explain what each view is supposed to provide, and to whom it provides it. Is it wise or necessary to represent a certain relation as an SQL view? Be sure to specify your views (about 4 would be sufficient) and give examples of their use.

# Procedural Elements

Does your design employ procedural extras such as database triggers (in PL/SQL or the MySQL equivalent format)? If so, describe and motivate each. If your design does not contain procedural extras, explain why, and say how you were able to do without these additions. Most projects have some scope for procedural elements (about 4 would be sufficient here).

# Example Queries: Your Database In Action

Your database will provide a structure for the data in an application and a means of accessing and viewing that data. In this section show us the database in action, by providing sample queries and their outputs (please do not provide large data sets as outputs; summarize as appropriate). Provide specific queries to test on your database and tell us what those queries provide to the application. Use your existing database as the basis for your queries. If a query makes reference to any additional tables then provide example rows of this table in section 3.

You may use screenshots here but do not overfill your report with screenshots. Ensure that there is a cohesive argument expressed in the text of the report and that it is not simply a bag of diagrams and queries and screenshots. When you include images, make sure they are readable and actually add to the discussion.

# Conclusions

Provide any concluding thoughts here. How might you build on this work for the future? How might your database support future developments?

# Acknowledgements

Name check any person who helped you with this work. Acknowledge that the work is entirely your own, and that every sentence in this report was written by you and you alone. If you wish to quote another person or piece of work, place the quoted work in quotation marks and cite the author inline. Plagiarism is a very serious infraction that must be dealt with severely. Avoid any ambiguity on this point by citing things carefully!

# References

List any bibliographical citations here [for people and work that you quote/cite in the main text of your report; please do include meaningful citations]