Data Analytics for Immersive Environments - Individual Project

Interactive RDBMS data with Quarto & Shiny

Niall McGuinness

2023-12-14

This assignment is designed for an postgraduate program with a focus on Data Analytics for Game & Extended Reality and Computer Animation. It encompasses five distinct parts, each aimed at assessing and enhancing students' proficiency in different aspects of data analytics as applied to game development and animation.

Table of contents

Learning Outcomes

To practice the following:

- Writing and executing complex SQL queries to retrieve and manipulate data.

- Understanding and applying SQL concepts in a database context.
- Analysing and interpreting data from a SQL database effectively.
- Developing and interpreting linear regression models in the context of game development data.
- Understanding the relationship between different variables and how they impact each other.
- Evaluating the performance of a linear regression model and understanding its limitations and applicability.
- Utilising Quarto to create interactive Shiny web applications.
- Designing user-friendly interfaces for data interaction and visualisation.
- Implementing data visualisation techniques to effectively communicate data insights.
- Applying data analytics techniques to real-world scenarios in game development.
- Analysing and interpreting data specific to game development projects or products.
- Demonstrating the ability to solve problems by applying analytical skills in a structured manner.

Functional Requirements

Part A - SQL (20 Marks)

You will be provided with an SQLite database file (via email before **5PM 18th December 2023**) containing information about various game development projects and their associated assets.

This database includes tables for projects, assets, developers, timelines, and customers. The structure of each table is shown.

Projects

Stores information about each game development project. It includes details like the project's unique identifier, name, start and end dates, budget, and the status of the project. It also links to the customer commissioning the project.

Column Name	Data Type	Description
ProjectID	Integer	Unique identifier for each project (Primary Key)
ProjectName	String	Name of the game development project
StartDate	Date	Start date of the project
EndDate	Date	End date of the project
Budget	Decimal	Total budget allocated for the project
Status	String	Current status of the project (e.g., In Progress, Completed, Cancelled)
CustomerID	Integer	Identifier linking to the Customers table (Foreign Key)

Assets

Stores information about the various assets used in game development projects. Each asset has a unique identifier, and the table includes information about the project it is associated with, the name and type of the asset, and its creation date.

Column Name	Data Type	Description
AssetID	Integer	Unique identifier for each asset (Primary Key)
ProjectID	Integer	Identifier linking to the Projects table (Foreign Key)
AssetName	String	Name of the asset
Type	String	Type of the asset (e.g., 3D Model, Animation, Texture)
CreationDate	Date	Date when the asset was created

Developers

Stores information about the developers working on various projects. It includes each developer's unique identifier, name, area of specialization, and the number of years of experience they have in the field.

Column Name	Data Type	Description
DeveloperID	Integer	Unique identifier for each developer (Primary Key)
Name	String	Name of the developer
Specialisation	String	Area of specialisation (e.g., Programmer, Animator, Artist)
ExperienceYears	Integer	Number of years of experience in the field

Timelines

Tracks significant milestones for each project. It includes a unique identifier for each timeline entry, links to the respective project, descriptions of milestones, and both expected and actual completion dates.

Column Name	Data Type	Description
TimelineID	Integer	Unique identifier for each timeline entry (Primary Key)
ProjectID	Integer	Identifier linking to the Projects table (Foreign Key)
Milestone	String	Description of the milestone
ExpectedCompletionDate	Date	Expected date of completion for the milestone
ActualCompletionDate	Date	Actual date of completion for the milestone

Customers

Stores information about the clients commissioning game development projects. It includes a unique identifier for each customer, along with their name, city, and country.

Column Name	Data Type	Description
CustomerID	Integer	Unique identifier for each customer (Primary Key)
CustomerName	String	Name of the customer
CustomerCity	String	City of the customer
CustomerCountry	String	Country of the customer

ProjectDevelopers

Links developers to the projects they are working on and shows which developer worked on which project and their specific role in that project.

Column Name	Data Type	Description
ProjectID	Integer	Identifier linking to the Projects table (Foreign Key)
DeveloperID	Integer	Identifier linking to the Developers table (Foreign Key)
Role	String	Role of the developer in the project (e.g., Lead, Contributor)

AssetsDevelopers

Links which developer was the principal developer on each asset.

Column Name	Data Type	Description
AssetID	Integer	Identifier linking to the Assets table (Foreign Key)
DeveloperID	Integer	Identifier linking to the Developers table (Foreign Key)

First, you are required to write **three** core SQL queries to perform the following tasks:

1. List the total budget allocated for projects in each country, along with the count of projects per country. Display sorted by the total budget in descending order.
2. List the average development time for projects, categorized by the number of assets used.
3. List the top three developers based on the number of successful projects they've been involved in. Display the results.

Next, you are also required to demonstrate the following **three** general SQL concepts using no fewer than **three** distinct SQL statements:

1. SELECT with LIKE and OR

2. SELECT with DISTINCT and ORDER BY
3. Subquery with SELECT

Display the results in formatted table(s) in a Quarto Notebook file inside a **Quarto Website**. You should add pagination to the tables if the number of rows is larger than 8-10 (see Recommended Reading on creating tables)

Part B - Linear Regression (20 Marks)

Utilise the provided dataset, which contains details about project budgets, timelines, team sizes, and success rates and apply linear regression analysis to understand trends or patterns in the game development lifecycle.

You are required to perform the following tasks:

1. **Model**: Perform linear regression to predict the success rate of a project based on its budget **and** team size and present the data in an appropriate plot.
2. **Interpret**: Interpret the model coefficients and discuss what insights they provide about game development.
3. **Discuss**: Comment on the reliability of the the linear regression model and any outliers present in the data.

Output the results in a Quarto Notebook file inside the **Quarto Website** from Part A.

Part C - Interactive Shiny Application using Quarto Dashboard (30 Marks)

Create an interactive Shiny application using Quarto Dashboards to display and interact with data from the database. You are required develop an interactive web application containing **two pages/tabs** which allow the following:

1. **Data View**: The first page/tab should allow the user to view the contents of **any four tables** in the database with no filtering or queries executed.You may need to consider using a more advanced table library (e.g. kable + kableExtra, reactable, DT) to add pagination to larger tables.
2. **Plot View**: The second page/tab should allow users to visualize data through 2-3 appropriate plots and **interact** with **each** plot. Typically, plots could include data related to project timelines, budget, asset types, etc.

The second page/tab must support user interactions via input fields (e.g., textfield, dropdown, slider, checkbox) to allow interactions such as: * Selecting developers to view more detailed data on the projects they participated in. * Specifying a completion date range for projects.

You must include a minimum of **two fields** for interaction for **each** plot.

Output the results in a **Quarto Website**. The web site does **not** need to be published to a remote server and may be run from your machine. This will be a second, and separate project from the Quarto Website produced for Parts A & B.

Part D - Technology Exploration (10 Marks)

Finally, you will be awarded a maximum of 10 marks for creative use of R packages which have **not** been covered in class. For example, this could be a package for rendering more visually appealing tables (i.e. reactable), animating plots (i.e. gganimate), or rendering the geospatial data from the database (i.e. customer city and customer country) using additional R packages (i.e. ggplot2 and ggmap).

Inside the **Quarto Website** from Parts A & B, you should include a short paragraph in a separate page, entitled *Technology Exploration.qmd*, which technologies you have explored in Parts A, B & C.

Part E - Quality & Conclusions (20 Marks)

You are required to present **two** separate project repos:

1. **Quarto Website** which brings together Parts A, B, D.
2. **Quarto Website** which hosts the Quarto Dashboard for Part C.

You will be assessed in terms of adherence to best practices in coding and data visualization and the quality of your conclusions. The assessment criteria for this component are listed below:

Document Formatting (4 Marks)

- Overall structure and organization of the report.
- Effective use of headings, subheadings, and bullet points.
- Consistency in font, spacing, and margins.
- Proper pagination and table of contents (if applicable).

Suitability of Tables in Part A (4 Marks)

- Clarity and readability of tables.
- Appropriate labelling for tables.
- Consistent and legible formatting in tables and figures.

Suitability of Plots in Part B (4 Marks)

- Relevance and effectiveness of chosen plots to represent the data.
- Ability to convey the intended message or findings clearly.
- Innovative and insightful use of data visualization techniques.

Adherence to Best Practices in Coding and Data Visualization (5 Marks)

- Code clarity, including commenting and organization.
- Use of efficient and appropriate coding methods.
- Adherence to data visualization best practices, including color choice, scale, and avoiding misleading representations.

Conclusion and Reflection (4 Marks)

- Inclusion of a 1-2 paragraph Conclusion section.
- Depth of reflection on what was learned from each part of the assignment.
- Insightful discussion on how the skills and knowledge gained apply to game development and computer animation.

Version Control Requirements

You must use a recognised online code repository (e.g., GitHub) and make regular well-named commits to your private repositories. Two links to your repos must be included in a README as part of the final submission and you must add your lecturer as a developer to the repositories. The repository must be named *2023_DAIE_ICA_1_StudentInitials* and *2023_DAIE_ICA_2_StudentInitials*.

Your grade for this component will depend directly upon the **regularity** of your commits. A development project of this size should consist of a minimum of **5+ distinct commit messages** per repository, spread over the lifetime of the development. Committing all your code in one commit, before the deadline, will be interpreted negatively.

Any submission made which does not include a repository link will not be graded.

Submission Requirements

1. In the README of your repositories you must include the **main file name** to be opened for each of the two projects (e.g. *index.qmd* for Part C Application) for each requirement. I will **not** filter through your submission looking for the correct file for each part. I will interpret the absence of this file as a failure to properly communicate your work which will negatively affect your grade for Part E.
2. A link to your R source code repositories (i.e. Quarto Website, Shiny Application). Ensure that **no** changes are made to the repositories following the submission deadline. You should create a separate fork for this submission and leave it unchanged after the deadline. Ask your lecturer for details on fork creation.
3. The assignment must be entirely the work of each student. Students are **not** permitted to share any pseudocode or source code from their solution with any other group in the class.
4. Students may **not** distribute the source code of their solution to any other student, in any format (i.e., electronic, verbal, or hardcopy transmission).
5. Plagiarised assignments will receive a mark of **zero**. This also applies to the individual/group allowing their work to be plagiarised. Any plagiarism will be reported to the Head of Department and a report will be added to your permanent academic record.
6. Late assignments will **only** be accepted if accompanied by the appropriate medical note. This documentation must be received within 10 working days of the project deadline. The Institute standard penalties for late submission will apply.
7. Each student **must** complete and **sign** a single assignment cover sheet. Please submit the signed cover sheet before 5 pm on the Friday of the week of the deadline.

Support Material

- GitHub - Quarto Website Demo
- GitHub - Shiny App with Quarto Dashboard
- Screencast - Setting up a Quarto Dashboard
- GitHub - Connecting to an SQLite DB

Recommended Reading

- Recommended Text - OpenIntro Statistics Videos & Slides
- SQL Tutorial
- A Comprehensive Introduction to Working with Databases using R
- Simple Linear Regression | An Easy Introduction & Examples
- Linear regression using R programming

Quarto Markdown & Shiny Related Resources

- YouTube - Hello, Quarto: A World of Possibilities (for Reproducible Publishing)
- Quarto - Using R
- Quarto Dashboards | Charles Teague | Posit
- Dashboards with Shiny for R

Plot & Table Related Resources

- The R Graph Gallery
- Get started with esquisse
- Analyze Data quickly with Esquisse
- Add Regression Line to ggplot2 Plot in R
- Fitting and visualizing linear regression models with the ggplot2 R package (CC237)
- Visualize your data using ggplot. R programming is the best platform for creating plots and graphs.
- Drag-and-drop ggplot2 graphs with the Esquisse library
- ggplot for plots and graphs. An introduction to data visualization using R programming
- Animate Graphs in R: Make Gorgeous Animated Plots with gganimate
- How to Make Beautiful Tables in R

Additional Resources

The resources below are provided as a mixture of **optional background reading**, technical reference, and useful tutorials.

- YouTube - Beautiful reports and presentations with Quarto
- How to style your Quarto docs without knowing HTML & CSS
- OPML Options