**Technical Report**

For Project 2, we worked as a team to complete the ETL process on a dataset.

**Extract**

**Transform**

**&**

**Load**

After considering multiple topic areas and datasets our group decided on the topic of “gender representation within modern day video games across multiple platforms”.

We found and used three datasets that focused on gender representation within modern day video games, sourced from Kaggle.com & diamondlobby.com\*. These datasets were games data, character data & sexualisation info.

* We have three datasets (638 character data across 65 modern day popular games)
* We joined the data set based on Characters.ID = Sexulixation.ID & Game.Game ID = Character.Game.
* The types of functions we used as part of our transform element to format the data to be able to load into our database (SQL) of the process is we cleaned select columns to ensure the dataframe in our jupyter notebook is similar to our tables within SQL over the 3 datasets and data tables. We cleaned up column names to either simplify names or renaming columns to shorter or more relevant names.
* We had to make sure that the numeric columns had no null values and fill them with zeros and string columns with nan so that the data could conform to our SQL table parameters.
* We had to change the numeric data column to date to fit sequel table requirements
* Make sure the data type in pandas agreed to the sequel datatypes form the jupyter dataframe and then connection strings
* We also sat and cleaned and tested our individual work area’s together as we split the data sets between the group and once it worked we aggregated the codes into a master file.
* The final jupyter code is Final\_jupytercode.ipynb and the sequel file : final\_sequeldatabase

The Schemata that is used in the final production of our database is saved into our group Git Hub Repository at along with the ERD;

[Database Schema](https://github.com/kass173/Project-2/blob/main/Gen_Rep_%20Games.sql)

[Entity Relationship Diagram](https://github.com/kass173/Project-2/blob/main/ERD_Project_2.PNG)

**Group Communication**

To be able to complete this project effectively we ensure we had communication within our group. We put in place a few mediums firstly being opening a slack channel just with our group members in it.

Graphical user interface, text, application, chat or text message

Description automatically generated

Secondly we also created a WhatsApp group again just with our group members in it which we discuss the project during working hours.

Graphical user interface, text, application, chat or text message

Description automatically generated

We also booked in extra session where we work together and on our respective parts using Team as the platform.

Graphical user interface, text, application, chat or text message

Description automatically generated

**The Project**

We created workload and separated then between the group and set deadline for review so that we had a realistic timeline to complete the project.

* + Project ideation
  + Data fetching
  + Data analysis
  + Cleaning & Transforming
  + Data loading into final database
  + Creating documentation (Technical report and Project Visuals)

After gaining all the information needed about the project we managed to break it down into manageable work chunks but also having the availability of the team to work on our work targets which we identified and is show in the process map on our git repository.

* Process Map - [Project 2 Process Map](https://github.com/kass173/Project-2/blob/main/Process%20Map/Colorful%20Process%20Prjt%202.png)

**Final Database**

Table connections :

* We had to ensure the primary keys used for the tables were unique.
* Table 1 Games and Table 2 Character can be joined on 1 column, ie game.game\_id = character.game .
* Game\_id is the primary key in table Game . Game column in Character table hence functions as a foreign key.
* Character Table and Sexualization have 2 columns on the basis of which they can be linked ie the sexualization\_total column and the id column.

-- game.game\_id = character.game

-- character.Id = sexualization.id

-- character.sexualization\_total = sexualization.sexualization\_total

--Table1 games; Primary Key : game\_id

--Table2 characters; Primary Key : game ; Foreign Key: Game

--Table 3 sexualization ; Foreign Key: id

[Final SQL Database](https://github.com/kass173/Project-2#:~:text=yesterday-,Gen_Rep_%20Games.sql,-uploading%20final%20sequel)

explains the final database, tables/collections, and why the topic was chosen