Laptop Market Dynamics: A SQL-Driven Exploration

- Sujai Adithya Muralidharan

DB Creation, Population and Business Questions

Discussion of how I converted the dataset into tables:

- Project Data Acquisition:
  - Acquired dataset from Kaggle consisting of a primary CSV file named "Raw Laptop Data" and three supplementary CSV files.

- Data Examination and Assessment:
  - Conducted a comprehensive analysis of the dataset.
  - Identified prevalent errors, including entries placed in incorrect columns.

- Data Cleaning Process:
  - Undertook meticulous data cleansing to rectify errors.
  - Ensured accurate values were correctly positioned within their respective columns.

- Enhancement of Dataset Structure:
  - Assigned unique identifiers to specific metrics for improved organizational structure.
  - Defined appropriate data types for each column to enhance dataset integrity.

- Table Design and Organization:
  - Divided the dataset into nine distinct tables for a more nuanced analysis.
  - Structured tables to reflect refined and organized data subsets.

- CSV File Generation:
  - Created individual CSV files for each new table, capturing cleaned and structured data.

- MySQL Integration:
  - Leveraged the "Table Data Import Wizard" for seamless integration of tables into MySQL.
  - Ensured a smooth transition from cleaned CSV files to a well-organized MySQL database.

- Summary of Transformation Journey:
  - The project involved a multi-step process, including meticulous data cleaning, thoughtful schema design, and the implementation of a robust import process.
  - The result is a refined and organized dataset, optimized for sophisticated analysis within the MySQL database environment.

Challenges faced during importing data and how did I overcome these data importation challenges:

ASCII Encoding Error:

One of the initial challenges encountered during the data import was an ASCII encoding error. This issue arose when the data being imported contained characters that were not compatible with the ASCII encoding format. To address this challenge, a Python script was employed to modify the encoding of the problematic data. The script converted the data to a compatible encoding format, ensuring that special characters were handled appropriately. This step was crucial in preventing data corruption and maintaining data integrity throughout the import process.

VLOOKUP for Data Manipulation:

Another significant challenge emerged when it became necessary to perform complex data manipulations during the import. Certain relationships between datasets required the use of VLOOKUP to merge and align data appropriately. To overcome this challenge, VLOOKUP functions were implemented strategically within the data import process. These functions facilitated the matching and merging of data from different sources, ensuring that the imported datasets were cohesive and accurately represented the desired relationships. The use of VLOOKUP not only resolved data alignment issues but also contributed to the overall data quality and coherence.

Data Consistency:

Ensuring data consistency and integrity posed an ongoing challenge, especially when dealing with large datasets. Inconsistent or incomplete data could potentially lead to errors and inaccuracies in subsequent analyses. To address this challenge, thorough data validation checks and cleansing processes were integrated into the import workflow. This included identifying and handling missing or duplicate data, validating data types, and implementing data quality checks. By enforcing strict data validation measures, the import process was optimized to maintain high levels of data consistency and integrity.

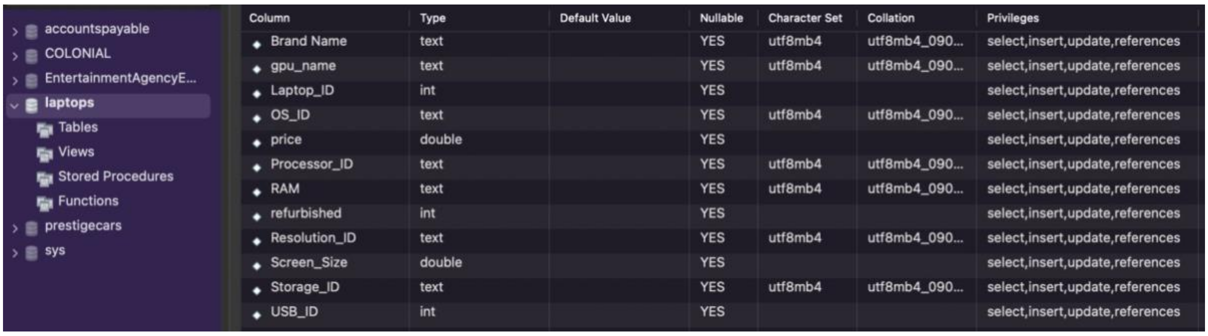
Missing Values:

The presence of missing values in the imported data posed a challenge to data completeness and accuracy. Addressing these missing values was crucial for meaningful analysis. Missing value imputation techniques were employed to handle the absence of data. This involved using statistical methods, such as mean or median imputation, or leveraging domain knowledge to fill in missing values where appropriate. By systematically addressing missing values, the imported datasets were made more robust for subsequent analyses.

Data dictionary for every table in the database:

1. laptops Table:

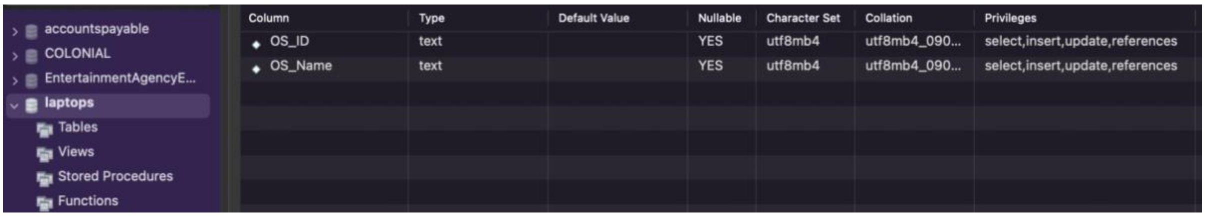
- Laptop_ID (Primary Key): This is a unique identifier assigned to each laptop in the table. It serves as the primary key to distinguish between different laptops.
- Brand_Name (Text): Represents the brand name of the laptop, indicating the manufacturer or brand associated with the device.
- Processor_ID (Foreign Key): Links to the `Processor_ID` in the `cpu` table, establishing a relationship between the laptop and the specific processor it uses.
- Screen_Size (Double): Denotes the size of the laptop screen in inches, providing information about the physical dimensions of the display.
- Resolution_ID (Foreign Key): Refers to the `Resolution_ID` in the `resolution` table, indicating the display resolution of the laptop.
- OS_ID (Foreign Key): Connects to the `OS_ID` in the `os` table, specifying the operating system installed on the laptop.
- RAM (Text): Represents the amount of Random Access Memory (RAM) in the laptop, indicating the device's memory capacity.
- Storage_ID (Foreign Key): Points to the `Storage_ID` in the `storage` table, indicating the storage capacity of the laptop.
- gpu_name (Text): Specifies the name of the dedicated Graphics Processing Unit (GPU) in the laptop, if applicable.
- USB_ID (Foreign Key): References the `USB_ID` in the `usb` table, specifying the type of USB ports on the laptop.
- Price (Double): Indicates the price of the laptop, providing information about its cost.
- Refurbished (Binary): A binary indicator (Yes/No) stating whether the laptop is refurbished.



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
Brand Name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
gpu_name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
Laptop_ID	int		YES			select,insert,update,references
OS_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
price	double		YES			select,insert,update,references
Processor_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
RAM	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
refurbished	int		YES			select,insert,update,references
Resolution_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
Screen_Size	double		YES			select,insert,update,references
Storage_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
USB_ID	int		YES			select,insert,update,references

2. os Table:

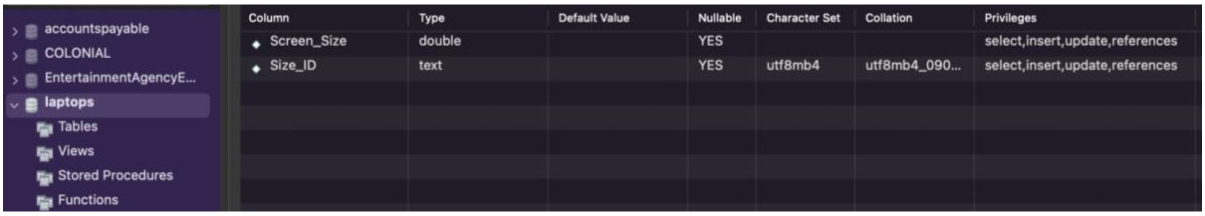
- OS_ID (Primary Key): Serves as a unique identifier for each operating system listed in the table.
- OS_Name (Text): Represents the name of the operating system associated with the `OS_ID`.



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
OS_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
OS_Name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

3. size Table:

- Size_ID (Primary Key): Uniquely identifies each screen size category listed in the table.
- Screen_Size (Double): Denotes the screen size in inches, providing standardized categories for different screen sizes.

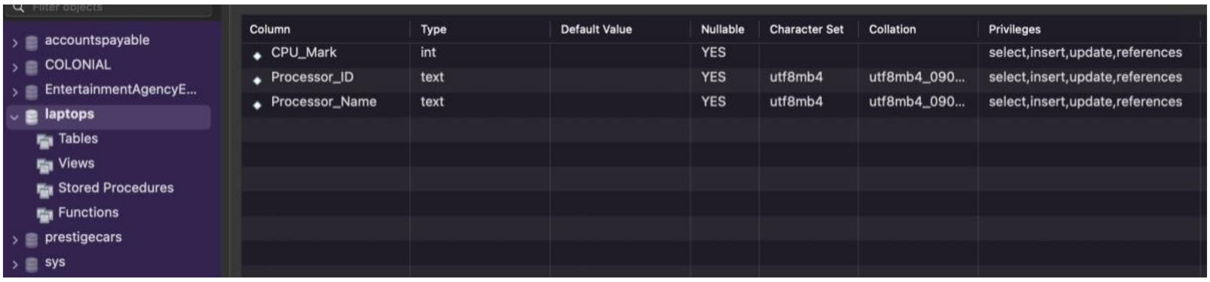


- accountspayable
- COLONIAL
- EntertainmentAgencyE...
- laptops
  - Tables
  - Views
  - Stored Procedures
  - Functions

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
Screen_Size	double		YES			select,insert,update,references
Size_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

4. usb Table:
   - USB_ID (Primary Key): Uniquely identifies each USB type listed in the table.
   - USB_Type (Text): Specifies the type of USB port associated with the `USB_ID`.

5. cpu Table:
   - Processor_Name (Text): Indicates the name of the processor, providing details about the type or model.
   - Processor_ID (Primary Key): A unique identifier assigned to each processor in the table.
   - CPU_Mark (Integer): Represents the benchmark score for the CPU, providing information about its performance.

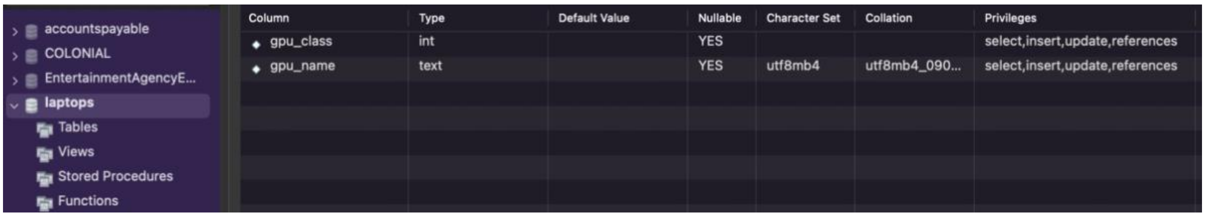


Filter objects

- accountspayable
- COLONIAL
- EntertainmentAgencyE...
- laptops
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prestigecars
- sys

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
CPU_Mark	int		YES			select,insert,update,references
Processor_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
Processor_Name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

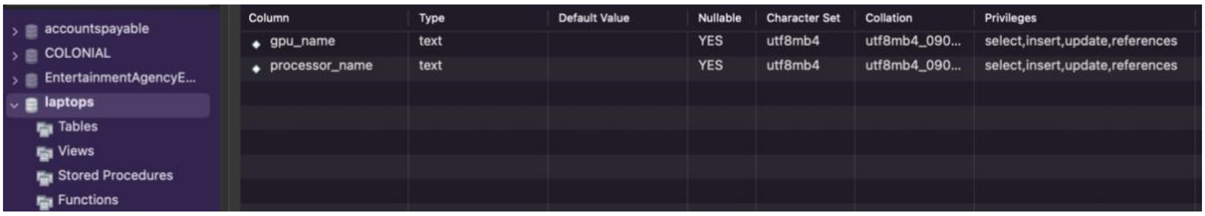
6. gpuclass Table:
   - gpu_name (Primary Key): Uniquely identifies each GPU listed in the table.
   - gpu_class (Integer): Represents the class of the GPU, with values ranging from 1 (highest class) to 4 (lowest class).



- accountspayable
- COLONIAL
- EntertainmentAgencyE...
- laptops
  - Tables
  - Views
  - Stored Procedures
  - Functions

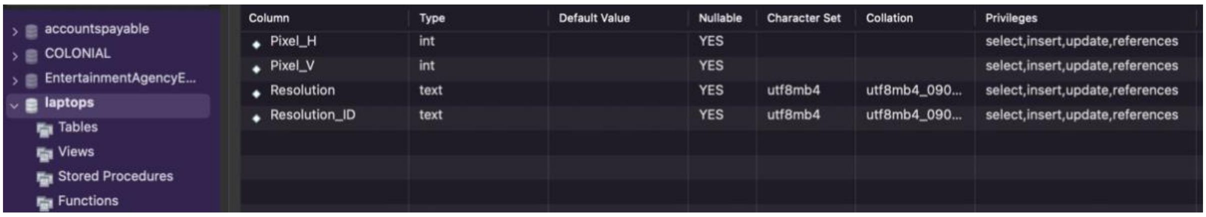
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
gpu_class	int		YES			select,insert,update,references
gpu_name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

7. intgpu Table:
   - processor_name (Text): Indicates the name of the processor with an integrated GPU.
   - gpu_name (Text): Specifies the name of the integrated GPU associated with the processor.



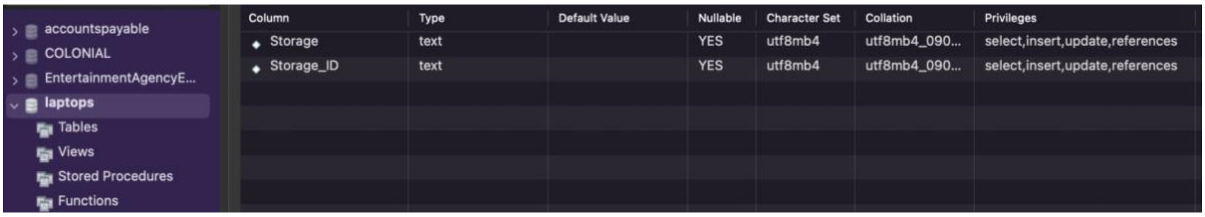
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
gpu_name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
processor_name	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

8. resolution Table:
   - Resolution_ID (Primary Key): Uniquely identifies each resolution type listed in the table.
   - Resolution (Text): Describes the resolution type (e.g., FHD, UHD), providing details about the display quality.
   - Pixel_H (Integer): Represents the horizontal pixel count for the resolution.
   - Pixel_V (Integer): Represents the vertical pixel count for the resolution.



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
Pixel_H	int		YES			select,insert,update,references
Pixel_V	int		YES			select,insert,update,references
Resolution	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
Resolution_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

9. storage Table:
   - Storage_ID (Primary Key): Uniquely identifies each storage capacity listed in the table.
   - Storage (Text): Describes the storage capacity (e.g., 128GB, 256GB), providing details about the laptop's storage size.



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
Storage	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references
Storage_ID	text		YES	utf8mb4	utf8mb4_090...	select,insert,update,references

The list of business questions:

1. Rank the top 5 laptops with the highest CPU benchmark scores, including information about the processor, GPU, and screen resolution.
2. Determine the percentage distribution of laptops across different USB types for each brand.
3. What is the average screen size of laptops for each GPU class, and how does it compare to the overall average screen size?
4. Find the top 3 USB types with the highest number of laptops and the percentage they contribute to the total, considering only laptops with UHD and FHDPLUS resolution.
5. Identify the top 3 resolutions with the highest average RAM capacity for laptops and provide the percentage of total RAM each resolution represents.
6. Identify the brand with the highest average storage size for laptops with a screen size larger than the overall average. Also, provide the cumulative percentage of total storage for all laptops.

7. Among refurbished laptops, what is the average price difference between those with storage capacity greater than 256GB and those with storage capacity 256GB or less? Additionally, provide the percentage distribution of these two storage categories among refurbished laptops.
8. Determine the top 3 screen sizes with the highest average laptop prices, considering only laptops with a screen resolution of 2K and FHD.