# **🌐 Books Data SQL Analysis**

## 📅 Project Overview

### Books to Scrape - Advanced SQL Analysis

* Data Source: [http://books.toscrape.com](http://books.toscrape.com/)
* Data Format: Title, Price, Rating, Availability
* Tools Used: MySQL Workbench, Python, CSV
* Goal: Apply advanced SQL techniques on real-world data

## 📅 1: Top Rated Books

### Query

SELECT title, price, rating

FROM books

WHERE rating = 5

ORDER BY price DESC;

### Insight

Most 5-star rated books are priced over £45, indicating a potential premium pricing trend.

## 📅 2: Price Range Distribution

### Query

SELECT

CASE

WHEN price < 20 THEN 'Under 20'

WHEN price BETWEEN 20 AND 40 THEN '20–40'

WHEN price BETWEEN 40 AND 60 THEN '40–60'

ELSE '60+'

END AS price\_range,

COUNT(\*) AS book\_count

FROM books

GROUP BY price\_range;

### Insight

Most books (50) fall in the 40–60 price range, followed by 47 books in the 20–40 range, and 29 books under 20.

## 📅 3: Average Price per Rating

### Query

SELECT rating, ROUND(AVG(price), 2) AS avg\_price

FROM books

GROUP BY rating

ORDER BY rating DESC;

### Insight

Interestingly, lower-rated books tend to be more expensive on average than higher-rated ones.

## 📅 4: Cheapest Book per Rating

### Query

SELECT b1.title, b1.price, b1.rating

FROM books b1

WHERE b1.price = (

SELECT MIN(b2.price)

FROM books b2

WHERE b2.rating = b1.rating

);

### Insight

Found the most affordable books for each rating level; "Patience" is the cheapest among all at £10.16 (Rating 3).

## 📅 5: Books Out of Stock

### Query

SELECT title, price

FROM books

WHERE availability NOT LIKE '%In stock%';

### Insight

All books in the dataset were found to be in stock.

## 📅 6: Rating vs Price (Summary)

### Query

SELECT

rating,

COUNT(\*) AS books\_in\_rating,

ROUND(AVG(price), 2) AS avg\_price

FROM books

GROUP BY rating

ORDER BY avg\_price DESC;

### Insight

Rating 3 has the highest average price and most books, while Rating 5 has the lowest average price and fewest books.

## 📅 7: Price Rank within Rating

### Query

SELECT

title, rating, price,

RANK() OVER (PARTITION BY rating ORDER BY price DESC) AS price\_rank

FROM books;

### Insight

Ranks books by price within their rating. Useful to find top or bottom-priced books within similar quality.

## 📅 8: Price Variability

### Query

SELECT

STDDEV(price) AS price\_stddev,

VARIANCE(price) AS price\_variance

FROM books;

### Insight

Standard deviation is 14.36, showing moderate variability in book pricing.

## 📅 9: Duplicate Book Titles

### Query

SELECT title, COUNT(\*) AS occurrences

FROM books

GROUP BY title

HAVING COUNT(\*) > 1;

### Insight

No duplicate titles were found in the dataset.

## 📅 10: Price Above Rating Average

### Query

SELECT \*

FROM books b

WHERE price > (

SELECT AVG(price)

FROM books

WHERE rating = b.rating

);

### Insight

Found 11 books that are priced higher than the average for their rating, including titles like "Sapiens" and "A Light in the Attic".

## 📅 Summary & Learnings

* Applied grouping, filtering, window functions, and CASE logic
* Derived real-world insights (pricing, rating, availability)
* Built queries ready for business dashboards
* Great project to showcase SQL proficiency in interviews