Proyecto SQL — Servicio de libros (TripleTen)

Autor: José Enrique Piñango Bustamante

Objetivo general: Analizar la base de datos de libros, autores, editoriales, calificaciones y reseñas para obtener hallazgos que sustenten una propuesta de valor del producto.

Tablas:

- `books(book_id, author_id, title, num_pages, publication_date, publisher_id)`
- `authors(author_id, author)`
- `publishers(publisher_id, publisher)`
- `ratings(rating_id, book_id, username, rating)`
- `reviews(review_id, book_id, username, text)`

Objetivos del estudio

1. Cuantificar la producción editorial y su evolución (libros publicados en fechas recientes).
2. Medir popularidad y satisfacción: volumen de reseñas y calificaciones promedio por libro.
3. Identificar editoriales relevantes excluyendo material muy corto (libros > 50 páginas).
4. Estimar la calidad promedio de autores con suficiente evidencia (≥ 50 calificaciones por libro).
5. Evaluar el comportamiento de usuarios muy activos (quienes califican > 50 libros): promedio de reseñas de texto que realizan.

Antes de los ejercicios, validaremos la conexión y exploraremos una muestra de cada tabla.

In [7]:
```
# importar librerías
import pandas as pd
from sqlalchemy import create_engine

# --- Conexión TripleTen ---
db_config = {
    'user': 'practicum_student',
    'pwd': 's65BlTKV3faNIGhmvJVzOqhs',
    'host': 'rc1b-wcoijxj3yxfsf3fs.mdb.yandexcloud.net',
    'port': 6432,
    'db': 'data-analyst-final-project-db'
}

connection_string = 'postgresql://{}:{}@{}:{}/{}'.format(
    db_config['user'],
```

```
    db_config['pwd'],
    db_config['host'],
    db_config['port'],
    db_config['db']
)

# Nota: según instrucciones, basta con sslmode=require
engine = create_engine(connection_string, connect_args={'sslmode': 'require'})

# Si en tu entorno necesitas validar con CA.pem, descomenta y ajusta la ruta:
# engine = create_engine(connection_string, connect_args={
#     'sslmode': 'verify-ca',
#     'sslrootcert': 'CA.pem'  # asegúrate de que CA.pem esté junto al .ipynb
# })
```

In [8]:
```
def run_sql(query: str, store_as: str = None):
    """
    Ejecuta una consulta SQL en el engine global y muestra el DataFrame.
    Opcionalmente guarda el resultado como CSV si store_as tiene un nombre de archi
    """
    df = pd.read_sql(query, con=engine)
    display(df)
    if store_as:
        df.to_csv(store_as, index=False)
        print(f"Archivo guardado: {store_as}")
    return df
```

Exploración inicial de tablas

Validamos la conexión y revisamos las primeras filas de cada tabla para confirmar estructura y tipos.

In [9]:
```
query = """
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'public'
ORDER BY table_name;
"""
run_sql(query)
```

	table_name
0	advertisment_costs
1	authors
2	books
3	check_avg
4	orders
5	publishers
6	ratings
7	reviews
8	visits

Out[9]:

	table_name
0	advertisment_costs
1	authors
2	books
3	check_avg
4	orders
5	publishers
6	ratings
7	reviews
8	visits

In [10]:
```
# books
run_sql("""
SELECT *
FROM books
ORDER BY book_id
LIMIT 5;
""")

# authors
run_sql("""
SELECT *
FROM authors
ORDER BY author_id
LIMIT 5;
""")

# publishers
run_sql("""
```

```
SELECT *
FROM publishers
ORDER BY publisher_id
LIMIT 5;
""")

# ratings
run_sql("""
SELECT *
FROM ratings
ORDER BY rating_id
LIMIT 5;
""")

# reviews
run_sql("""
SELECT *
FROM reviews
ORDER BY review_id
LIMIT 5;
""")
```

	book_id	author_id	title	num_pages	publication_date	publisher_id
0	1	546	'Salem's Lot	594	2005-11-01	93
1	2	465	1 000 Places to See Before You Die	992	2003-05-22	336
2	3	407	13 Little Blue Envelopes (Little Blue Envelope...	322	2010-12-21	135
3	4	82	1491: New Revelations of the Americas Before C...	541	2006-10-10	309
4	5	125	1776	386	2006-07-04	268

	author_id	author
0	1	A.S. Byatt
1	2	Aesop/Laura Harris/Laura Gibbs
2	3	Agatha Christie
3	4	Alan Brennert
4	5	Alan Moore/David Lloyd

	publisher_id	publisher
0	1	Ace
1	2	Ace Book
2	3	Ace Books
3	4	Ace Hardcover
4	5	Addison Wesley Publishing Company

	rating_id	book_id	username	rating
0	1	1	ryanfranco	4
1	2	1	grantpatricia	2
2	3	1	brandtandrea	5
3	4	2	lorichen	3
4	5	2	mariokeller	2

	review_id	book_id	username	text
0	1	1	brandtandrea	Mention society tell send professor analysis. ...
1	2	1	ryanfranco	Foot glass pretty audience hit themselves. Amo...
2	3	2	lorichen	Listen treat keep worry. Miss husband tax but ...
3	4	3	johnsonamanda	Finally month interesting blue could nature cu...
4	5	3	scotttamara	Nation purpose heavy give wait song will. List...

Out[10]:

	review_id	book_id	username	text
0	1	1	brandtandrea	Mention society tell send professor analysis. ...
1	2	1	ryanfranco	Foot glass pretty audience hit themselves. Amo...
2	3	2	lorichen	Listen treat keep worry. Miss husband tax but ...
3	4	3	johnsonamanda	Finally month interesting blue could nature cu...
4	5	3	scotttamara	Nation purpose heavy give wait song will. List...

Chequeos rápidos de consistencia

- `books.book_id` debe existir en `ratings.book_id` y `reviews.book_id` (no necesariamente todos los libros tienen ratings/reviews).
- `author_id` en `books` debe existir en `authors`.
- `publisher_id` en `books` debe existir en `publishers`.

- `publication_date` debe parsearse como fecha (no haremos conversión en pandas; lo manejaremos desde SQL cuando sea necesario).
- `num_pages` debe ser numérico y > 0 para libros estándar.

In [12]:
```
query = """
SELECT COUNT(*) AS books_after_2000
FROM books
WHERE publication_date > '2000-01-01';
"""
run_sql(query);
```

	books_after_2000
0	819

Libros publicados después del 1 de enero de 2000

El resultado muestra que **819 libros** en la base de datos fueron publicados después del **1 de enero del 2000**.

Conclusión:

- La base de datos incluye una cantidad significativa de títulos contemporáneos (819).
- Esto confirma que la mayor parte del catálogo es relativamente reciente, lo que es importante para orientar la propuesta de valor hacia las preferencias modernas de los lectores.

In [13]:
```
query = """
WITH reviews_ct AS (
    SELECT book_id, COUNT(*) AS reviews_count
    FROM reviews
    GROUP BY book_id
),
ratings_avg AS (
    SELECT book_id, AVG(rating) AS avg_rating
    FROM ratings
    GROUP BY book_id
)
SELECT
    b.book_id,
    b.title,
    COALESCE(rv.reviews_count, 0)                AS reviews_count,
    ROUND(rt.avg_rating::numeric, 2)             AS avg_rating
FROM books b
LEFT JOIN reviews_ct  rv USING (book_id)
LEFT JOIN ratings_avg rt USING (book_id)
ORDER BY reviews_count DESC,
         avg_rating DESC NULLS LAST,
         b.book_id;
"""
df_ex2 = run_sql(query)
```

	book_id	title	reviews_count	avg_rating
0	948	Twilight (Twilight #1)	7	3.66
1	302	Harry Potter and the Prisoner of Azkaban (Harr...	6	4.41
2	299	Harry Potter and the Chamber of Secrets (Harry...	6	4.29
3	656	The Book Thief	6	4.26
4	734	The Glass Castle	6	4.21
...	...	...	...	...
995	191	Disney's Beauty and the Beast (A Little Golden...	0	4.00
996	221	Essential Tales and Poems	0	4.00
997	387	Leonardo's Notebooks	0	4.00
998	83	Anne Rice's The Vampire Lestat: A Graphic Novel	0	3.67
999	808	The Natural Way to Draw	0	3.00

1000 rows × 4 columns

Número de reseñas de usuarios y calificación promedio por libro

Resultados:

- Libros populares en reseñas: *Twilight (Twilight #1)* (7 reseñas), *Harry Potter and the Prisoner of Azkaban* (6 reseñas), *The Book Thief* (5 reseñas).
- Calificaciones promedio: oscilan entre ~3.6 y ~4.4 para los títulos más reseñados.
- Existen libros con **0 reseñas de texto** pero sí con calificaciones (`avg_rating`), lo que muestra que no todos los usuarios dejan comentarios.

Conclusión:

- El número de reseñas de texto es relativamente bajo comparado con el total de calificaciones, lo que sugiere que la mayoría de los usuarios prefiere calificar rápidamente en lugar de escribir comentarios.
- Los títulos con más reseñas tienden a coincidir con libros muy conocidos y exitosos (ejemplo: *Twilight*, *Harry Potter*), lo que valida que el sistema captura bien la popularidad.
- La diferencia entre `reviews_count` y `avg_rating` indica que conviene analizar **ambos indicadores juntos** para evaluar popularidad y satisfacción.

In [15]:
```
query = """
SELECT
    p.publisher_id,
    p.publisher,
```

```
    COUNT(*) AS books_over_50_pages
FROM books b
JOIN publishers p ON p.publisher_id = b.publisher_id
WHERE b.num_pages > 50
GROUP BY p.publisher_id, p.publisher
ORDER BY books_over_50_pages DESC, p.publisher_id
LIMIT 1;
"""
run_sql(query);
```

	publisher_id	publisher	books_over_50_pages
0	212	Penguin Books	42

Editorial con más libros (> 50 páginas)

Resultado:

- La editorial con más libros sustantivos es **Penguin Books**, con **42 títulos** de más de 50 páginas.

Conclusión:

- Penguin Books lidera claramente el catálogo de publicaciones significativas en la base de datos.
- Esto refuerza su rol como una editorial dominante, con gran presencia en el mercado y un aporte fuerte a la colección moderna de libros.
- Para una propuesta de valor, vale la pena considerar acuerdos o recomendaciones de catálogo enfocados en esta editorial.

In [16]:
```
query = """
WITH book_stats AS (
    SELECT
        b.book_id,
        b.author_id,
        AVG(r.rating)  AS avg_rating,
        COUNT(r.rating) AS n_ratings
    FROM ratings r
    JOIN books b USING (book_id)
    GROUP BY b.book_id, b.author_id
),
author_stats AS (
    SELECT
        a.author_id,
        a.author,
        ROUND(AVG(bs.avg_rating)::numeric, 3) AS author_avg_rating,
        COUNT(*)                              AS books_considered
    FROM book_stats bs
    JOIN authors a ON a.author_id = bs.author_id
    WHERE bs.n_ratings >= 50
    GROUP BY a.author_id, a.author
)
```

```
SELECT *
FROM author_stats
ORDER BY author_avg_rating DESC, books_considered DESC, author
LIMIT 1;
"""
run_sql(query);
```

	author_id	author	author_avg_rating	books_considered
0	236	J.K. Rowling/Mary GrandPré	4.284	4

Autor con mayor calificación promedio (libros con ≥ 50 calificaciones)

Resultado:

- Autor: **J.K. Rowling/Mary GrandPré**
- Calificación promedio: **4.284**
- Libros considerados: **4** (todos con ≥ 50 calificaciones)

Conclusión:

- J.K. Rowling destaca como la autora mejor valorada en la base de datos, con una calificación promedio superior a 4.2.
- El hecho de que varios de sus libros superen ampliamente el umbral de 50 calificaciones muestra que su popularidad y consistencia son sólidas.
- Esto confirma que los títulos de *Harry Potter* no solo son muy leídos, sino también bien recibidos por la comunidad lectora.

In [17]:
```
query = """
WITH heavy_raters AS (
    SELECT
        username,
        COUNT(*) AS ratings_count
    FROM ratings
    GROUP BY username
    HAVING COUNT(*) > 50
),
reviews_per_heavy_user AS (
    SELECT
        hr.username,
        COALESCE(COUNT(rv.review_id), 0) AS text_reviews_count
    FROM heavy_raters hr
    LEFT JOIN reviews rv
      ON rv.username = hr.username
    GROUP BY hr.username
)
SELECT
    ROUND(AVG(text_reviews_count)::numeric, 3) AS avg_text_reviews_among_heavy_rate
    COUNT(*)                                   AS users_considered
FROM reviews_per_heavy_user;
```

```
"""
run_sql(query);
```

	avg_text_reviews_among_heavy_raters	users_considered
0	24.333	6

Promedio de reseñas de texto entre usuarios con > 50 calificaciones

Resultado:

- Usuarios considerados: **6**
- Promedio de reseñas de texto: **24.33** reseñas por usuario

Conclusión:

- Los usuarios más activos (quienes califican más de 50 libros) no solo califican, sino que también escriben un volumen considerable de reseñas de texto.
- En promedio, estos heavy raters publican más de 24 reseñas cada uno, lo que los convierte en una fuente clave de feedback cualitativo.
- Este hallazgo sugiere que estos usuarios son altamente comprometidos con la plataforma y representan un segmento importante para estrategias de fidelización.

Conclusiones generales del estudio

Tras analizar la base de datos de libros, autores, editoriales, calificaciones y reseñas, se obtuvieron los siguientes hallazgos:

1. **Producción editorial reciente**
   - Se identificaron **819 libros** publicados después del 1 de enero del 2000.
   - Esto refleja un catálogo con una fuerte presencia de títulos contemporáneos, lo que favorece el diseño de una propuesta de valor centrada en obras modernas.
2. **Popularidad y calificaciones por libro**
   - Libros como *Twilight* y *Harry Potter* destacan por concentrar el mayor número de reseñas de texto (6–7) y calificaciones promedio superiores a 3.6–4.4.
   - Muchos libros reciben calificaciones pero no reseñas, lo que confirma que la mayoría de usuarios opta por dejar puntuaciones rápidas en lugar de comentarios largos.
3. **Editorial dominante**
   - **Penguin Books** es la editorial con mayor número de títulos sustantivos (> 50 páginas), con **42 libros**.

- Esto la posiciona como un actor clave en el catálogo de la plataforma.

4. **Autor mejor valorado**

   - **J.K. Rowling/Mary GrandPré** es la autora con la mejor calificación promedio (**4.284**) entre los libros con al menos 50 calificaciones, con **4 títulos considerados**.
   - Esto confirma la relevancia y calidad percibida de la saga *Harry Potter* en la base.

5. **Usuarios altamente activos (heavy raters)**

   - Se identificaron **6 usuarios** que calificaron más de 50 libros.
   - En promedio, estos heavy raters escribieron **24.3 reseñas de texto** cada uno, mostrando un nivel de compromiso alto y aportando feedback cualitativo valioso.

Recomendaciones

- **Fomentar la participación cualitativa**: incentivar a más usuarios a dejar reseñas de texto, no solo calificaciones, para enriquecer el feedback.
- **Estrategias con editoriales clave**: Penguin Books podría ser una aliada importante para acuerdos de catálogo y promociones.
- **Autores estrella como ganchos**: destacar títulos de J.K. Rowling y otros autores de alto rating como contenido atractivo para atraer y retener usuarios.
- **Fidelización de heavy raters**: diseñar beneficios especiales (gamificación, insignias, acceso anticipado) para los usuarios que más reseñan y califican, pues aportan valor estratégico a la plataforma.