Online Book Store

SQL Project

3 CSV Files

Books.csv

Customers.csv

Orders.csv

Common Columns:

• Book_ID (Books & Orders)

• Customer_ID (Customers & Orders)

Basic Queries

1) Retrieve all books in the 'Fiction' genre:

```
SELECT *
FROM books
WHERE genre IN ('Fiction');
```

Output:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	39.23	78
28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	37.51	79
29	Quality-focused multi-tasking challenge	Katrina Underwood	Fiction	1905	31.12	100
31	Implemented encompassing conglomeration	Melissa Taylor	Fiction	2010	21.23	44
39	Optimized national process improvement	Megan Goodwin	Fiction	1978	10.99	42

2) Find books published after the year 1950:

```
SELECT *
FROM books
WHERE published_year > 1950;
```

Output:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
9	Optimized interactive challenge	Colin Buckley	Fantasy	1987	14.33	70

3) List all customers from Canada:

```
SELECT *
FROM customers
WHERE country IN ('Canada');
```

Output:

Customer_ID	Name	Email	Phone	City	Country
38	Nicholas Harris	christine93@perkins.com	1234567928	Davistown	Canada
415	James Ramirez	robert54@hall.com	1234568305	Maxwelltown	Canada
468	David Hart	stokesrebecca@gmail.com	1234568358	Thompsonfurt	Canada
NULL	NULL	NULL	NULL	NULL	NULL

4) Show orders placed in November 2023:

```
SELECT *
FROM orders
WHERE Order_Date BETWEEN '2023-11-01' AND '2023-11-30';
```

Output:

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amour
30	438	248	2023-09-23	6	281.94
42	415	70	2023-09-19	3	70.35
66	180	323	2023-09-08	8	286.64
119	328	148	2023-09-24	3	36.75
166	272	486	2023-09-17	6	114.78
176	86	479	2023-09-19	6	229.62

5) Retrieve total stock of books available:

```
SELECT SUM(stock)
FROM books;
```

Output:

SUM(stock)
25056

6) Find details of the most expensive book:

```
SELECT *
FROM books
ORDER BY price DESC
LIMIT 1;
```

Output:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88

7) Show customers who ordered more than 1 quantity:

```
SELECT *
FROM orders
WHERE quantity > 1;
```

Output:

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amou
1	84	169	2023-05-26	8	188.56
2	137	301	2023-01-23	10	216.60
3	216	261	2024-05-27	6	85.50
4	433	343	2023-11-25	7	301.21
5	14	431	2023-07-26	7	136.36
6	439	119	2024-10-11	5	249.40

8) Retrieve orders where total amount exceeds $20:

```
SELECT *
FROM orders
WHERE total_amount > 20;
```

Output:

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
1	84	169	2023-05-26	8	188.56
2	137	301	2023-01-23	10	216.60
3	216	261	2024-05-27	6	85.50
4	433	343	2023-11-25	7	301.21
5	14	431	2023-07-26	7	136.36
6	439	119	2024-10-11	5	249.40

9) List all genres available in Books table:

```
SELECT DISTINCT genre
FROM books
ORDER BY genre;
```

Output:

genre
Biography
Fantasy
Fiction
Mystery
Non-Fiction
Romance
Science Fiction

10) Find book with lowest stock:

```
SELECT *
FROM books
ORDER BY stock
LIMIT 1;
```

Output:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
378	Future-proofed heuristic function	Samantha Mcclain	Romance	1903	6.01	0
44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0
60	Robust eco-centric capacity	Brian Haney	Biography	1990	35.14	0
127	Business-focused real-time benchmark	David Nelson	Science Fiction	1997	11.66	0
163	Object-based eco-centric challenge	Douglas Mccarthy	Non-Fiction	1905	19.11	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

11) Calculate total revenue from all orders:

```
SELECT SUM(total_amount) AS total_revenue
FROM orders;
```

Output:

total_revenue
75628.66

Advance Queries

1) Total number of books sold for each genre:

```
SELECT b.genre, SUM(o.quantity) AS sold_books
FROM books b
JOIN orders o ON b.Book_ID = o.Book_ID
GROUP BY b.genre;
```

Output:

genre	sold_books
Biography	285
Fantasy	446
Science Fiction	447
Mystery	504
Romance	439
Non-Fiction	351
Fiction	225

2) Average price of books in 'Fantasy' genre:

```
SELECT genre, ROUND(AVG(price), 2)
FROM books
WHERE genre IN ('Fantasy');
```

Output:

genre	avg_price
Fantasy	25.98

3) Customers who placed at least 2 orders:

```
SELECT c.customer_id, c.name, COUNT(o.order_id) AS total_orders
FROM customers c
JOIN orders o ON c.Customer_ID = o.Customer_ID
GROUP BY c.customer_id, c.name
HAVING COUNT(o.order_id) >= 2;
```

Output:

Customer_ID	name	COUNT(order_id)
84	Gary Blair	2
137	Steven Miller	2
216	Phillip Allen	2
14	John Wood	2
195	Dominique Turner	3
109	Jacob Kelley	2
94	Mr. David Cox	3
131	Peter Smith	2
454	April Anderson	2
420	Andrew Murray	3
462	James Brewer	3
377	Darrell Khan	2

4) Most frequently ordered book:

```
SELECT b.title, b.author, SUM(o.quantity) AS qty
FROM orders o
JOIN books b ON o.Book_ID = b.Book_ID
GROUP BY b.title, b.author
ORDER BY qty DESC
LIMIT 1;
```

Output:

title	author	qty
Realigned multi-tasking installation	Patrick Contreras	28

5) Top 3 most expensive Fantasy books:

```
SELECT *
FROM books
WHERE genre IN ('Fantasy')
ORDER BY price DESC
LIMIT 3;
```

Output:

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
462	Innovative 3rdgeneration database	Allison Contreras	Fantasy	1988	49.23	62
238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72
NULL	NULL	NULL	NULL	NULL	NULL	NULL

6) Total quantity sold by each author:

```
SELECT b.author, SUM(o.quantity) AS sold_books
FROM orders o
JOIN books b ON o.Book_ID = b.Book_ID
GROUP BY b.author;
```

Output:

author	title	sold_books
Margaret Moore	Networked tertiary approach	8
John Dav	ized high-level installation	13
Christopher Fuentes	Intuitive content-based toolset	6
Marissa Smith	De-engineered grid-enabled secured line	16
Christopher Dixon	Synergized fresh-thinking monitoring	15
Tonya Saunders	Switchable modular moratorium	21
Larry Hunt	Function-based dedicated frame	6
Brandon Foster	Proactive 5thgeneration middleware	4
Michelle Bell	Mandatory executive groupware	11
Mary French	Profound leadingedge capability	14
Lisa Lopez	Digitized executive flexibility	1
Derrick Howard	Streamlined coherent initiative	5

John Davidson

7) Cities where customers spent over $30:

```
SELECT c.city, o.total_amount AS spent_amount
FROM customers c
JOIN orders o ON c.Customer_ID = o.Customer_ID
WHERE o.total_amount > 30
ORDER BY c.city;
```

Output:

city	spent_amount
Adkinsview	125.51
Adrianafort	79.29
Aguilarside	148.68
Aguilarside	253.75
Aguilarside	148.77
Alanton	170.75
Alanton	38.64
Amandamouth	49.72
Angelaside	42.19
Angelastad	426.10
Angelatown	87.01
Annhaven	156.87

8) Customer who spent the most:

```
SELECT c.customer_id, c.name, SUM(o.total_amount) AS sp_amount
FROM customers c
JOIN orders o ON c.Customer_ID = o.Customer_ID
GROUP BY c.Customer_ID, c.name
ORDER BY sp_amount DESC
LIMIT 1;
```

Output:

customer_id	name	sp_amount
457	Kim Turner	1398.90
174	Jonathon Strickland	1080.95
364	Carrie Perez	1052.27
405	Julie Smith	991.00
386	Pamela Gordon	986.30
425	Ashley Perez	942.62
474	Anthony Young	929.19
163	Robert Clark	746.65
167	Justin Spencer	719.93
214	Alexander Scott	682.15
437	Cynthia Cooper	667.27
98	Robert Blair	633.90

9) Stock remaining after fulfilling all orders:

```
SELECT b.book_id, b.title,
IFNULL((b.stock - SUM(o.quantity)), b.stock) AS remaining_stk
FROM books b
LEFT JOIN orders o ON b.Book_ID = o.Book_ID
GROUP BY b.book_id, b.title;
```

Output:

book_id	title	remaining_stk
1	Configurable modular throughput	97
2	Persevering reciprocal knowledge user	19
3	Streamlined coherent initiative	22
4	Customizable 24hour product	8
5	Adaptive 5thgeneration encoding	8
6	Advanced encompassing implementation	2
7	Open-architected exuding structure	90
8	Persistent local encoding	81
9	Optimized interactive challenge	70
10	Ergonomic national hub	24
11	Secured zero tolerance time-frame	5
12	Polarized optimal array	63