

SQL Project on Books, Customers & Orders

INTRODUCTION

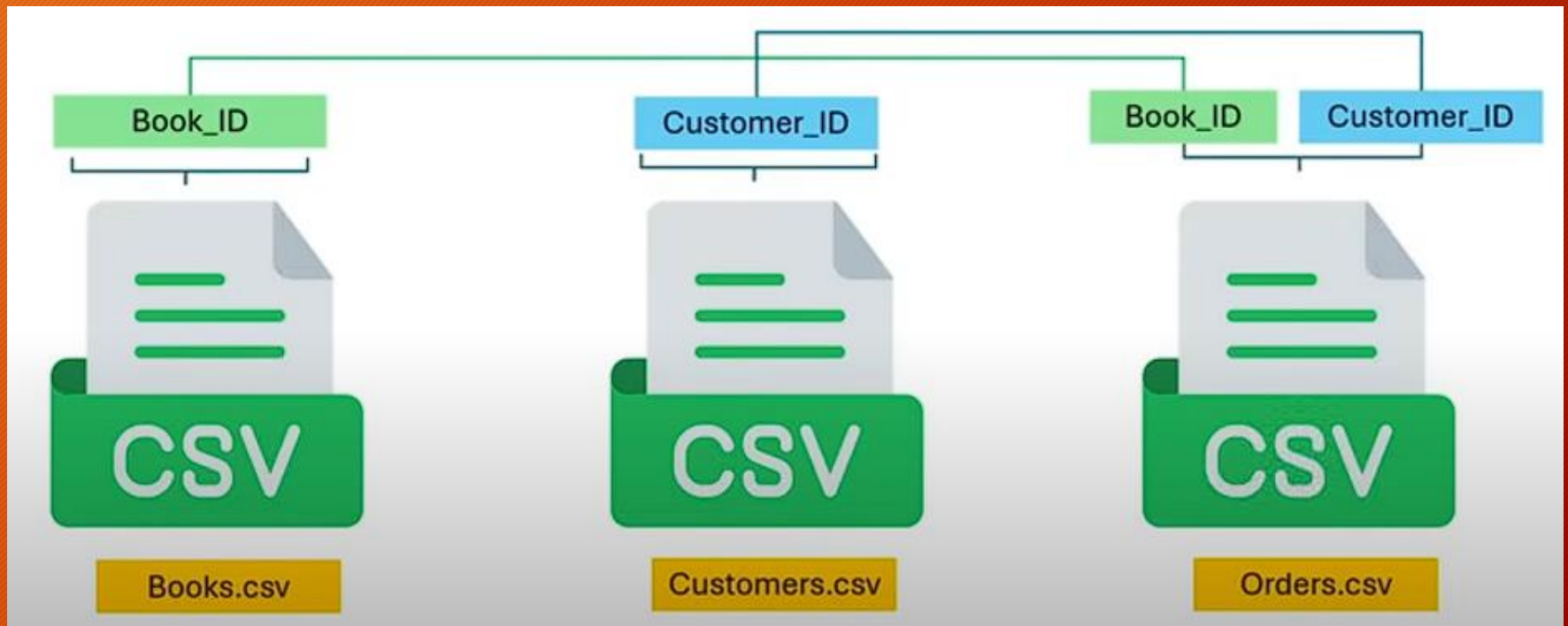
- ◆ This SQL project is based on a hospital management dataset containing information about hospitals, departments, patients, and medical records.
- ◆ The goal was to analyze healthcare data using SQL queries and extract meaningful insights for better decision-making.
- ◆ Key operations included calculating averages, identifying trends, and summarizing data across hospitals, cities, and departments.
- ◆ The project highlights the real-world application of SQL through grouping, filtering, date calculations, and performance analysis.

OBJECTIVES

- ◆ To analyze hospital-related data using SQL and derive meaningful insights.
- ◆ To perform data aggregation and filtering based on departments, cities, and time periods.
- ◆ To identify key metrics such as patient count, doctor availability, and medical expenses.
- ◆ To enhance data analysis skills by solving real-world business queries using SQL.

For this project, I used these three CSV files.

Tables must have at least one common column with same column name and same data type.



Books

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
138
139 SELECT * FROM Books;
```

Data Output Messages Notifications

Showing rows: 1 to 500 Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	1	Configurable modular throughput	Joseph Crane	Biography	1949	21.34	100
2	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
3	3	Streamlined coherent initiative	Derrick Howard	Non-Fiction	1913	15.75	27
4	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
5	5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
6	6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
7	7	Open-architected exuding structure	Jacqueline Young	Romance	1927	43.63	95
8	8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
9	9	Optimized interactive challenge	Colin Buckley	Fantasy	1987	14.33	70
10	10	Ergonomic national hub	Samantha Ruiz	Mystery	2015	24.63	25
11	11	Secured zero tolerance time-frame	Denise Barnes	Fantasy	1998	35.95	10
12	12	Polarized optimal array	Destiny Scott	Non-Fiction	1989	27.43	63
13	13	Adaptive 5thgeneration orchestration	Jaclyn Miller	Romance	1913	14.04	99
14	14	Re-engineered demand-driven parallelism	Jeremy Hayes	Science Fiction	1933	6.04	95
15	15	User-friendly motivating strategy	Keith Smith	Non-Fiction	1997	23.83	58
16	16	Vision-oriented tangible project	Christopher Price	Mystery	1941	10.07	8
17	17	Reduced secondary core	Benjamin Peters	Fantasy	1966	5.37	45
18	18	Adaptive 4thgeneration concept	Hector Palmer	Non-Fiction	2021	39.47	32

Total rows: 500 Query complete 00:00:00.215

CRLF Ln 138, Col 1

Customers

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

140 SELECT * FROM Customers;

Data Output Messages Notifications

Showing rows: 1 to 500 Page No: 1 of 1

	customer_id [PK] Integer	name character varying (100)	email character varying (100)	phone character varying (15)	city character varying (50)	country character varying (150)
1	1	Deborah Griffith	balljoseph@wright-keith.net	1234567891	South Craigfort	Denmark
2	2	Crystal Clements	kimberlybennett@curtis.com	1234567892	East Derekberg	Nicaragua
3	3	Susan Fuller	beanmichael@burnett-stewart.net	1234567893	Austinbury	Equatorial Guinea
4	4	Jamie Ramirez	amandahood@warren.com	1234567894	Dianamouth	Slovenia
5	5	Marcus Murphy	connerjohn@yahoo.com	1234567895	Smithbury	Guinea-Bissau
6	6	Stephen Vasquez	ricemiguel@yahoo.com	1234567896	Hamiltonstad	Rwanda
7	7	Susan Hicks	jeffrey91@yahoo.com	1234567897	East Rebecca	Montenegro
8	8	Matthew Johnson	austinkenneth@manning.net	1234567898	Kirstenborough	Israel
9	9	Matthew Williams	jeffrey41@diaz.com	1234567899	Rebeccafurt	Somalia
10	10	Ronald Osborn	staciekelley@heath.com	1234567900	Lake Benjamin	Cameroon
11	11	Thomas Garcia	rmiller@gmail.com	1234567901	West Monicabury	Argentina
12	12	Jennifer Murray	wilsonbrittany@hotmail.com	1234567902	South Ashleychester	Zimbabwe
13	13	Kristine Kim	sarahwilliams@hotmail.com	1234567903	Lake Robert	Nigeria
14	14	John Wood	johnsonalexander@gmail.com	1234567904	Richardsonville	New Caledonia
15	15	Vanessa Gaines	tbullock@gmail.com	1234567905	Rodriguezmouth	Wallis and Futuna
16	16	Stacey Flores	jackjackson@hotmail.com	1234567906	East Michaelfurt	Saint Helena
17	17	Christine Maldonado	ztaylor@yahoo.com	1234567907	Smithborough	Chile
18	18	David Watson	laurenlawson@alvarez-mcfarland.com	1234567908	New Josephtown	Equatorial Guinea
19	19	Marc Nash	joshua59@gmail.com	1234567909	Davidton	French Guiana

Total rows: 500 Query complete 00:00:00.140

CRLF Ln 138, Co

Orders

pgAdmin 4

File Object Tools Edit View Window Help

Welcome | Last Assignment.sql* X

company/postgres@PostgreSQL 17

Navigation icons: folder, save, dropdown, edit, filter, No limit, play, stop, E, bar chart, dropdown, checkmark, refresh, list, help.

Query Query History

Data Output	Messages	Notifications
-------------	----------	---------------

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️

📈

SQL

Showing rows: 1 to 500

Page No: 1

≡

📄

▼

📋

▼

🗑️

🗄️

⬇️


📈

SQL

Showing rows: 1 to 500

Page No: 1

	order_id [PK] integer ↗	customer_id integer ↗	book_id integer ↗	order_date date ↗	quantity integer ↗	total_amount numeric (10,2) ↗	
1	1	84	169	2023-05-26	8	188.56	
2	2	137	301	2023-01-23	10	216.60	
3	3	216	261	2024-05-27	6	85.50	
4	4	433	343	2023-11-25	7	301.21	
5	5	14	431	2023-07-26	7	136.36	
6	6	439	119	2024-10-11	5	249.40	
7	7	195	467	2023-10-23	6	82.92	
8	8	32	159	2024-05-07	4	144.84	
9	9	109	407	2024-01-04	9	379.71	
10	10	94	122	2024-07-09	4	123.00	
11	11	131	206	2023-10-16	1	38.01	
12	12	454	3	2024-06-17	2	31.50	
13	13	420	180	2023-06-08	5	125.45	
14	14	454	319	2023-08-24	2	85.22	
15	15	127	479	2023-01-10	6	229.62	
16	16	412	196	2023-10-06	8	53.52	
17	17	462	481	2023-03-20	5	52.75	
18	18	377	101	2024-08-07	4	193.96	
19	19	496	60	2023-11-17	9	316.26	


 Total rows: 500 Query complete 00:00:00.241

Retrieve all books in the “Fiction”

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
32
33 -- 1) Retrieve all books in the "Fiction" genre:
34 SELECT * FROM BOOKS
35 WHERE genre = 'Fiction';
36
```

Data Output Messages Notifications

Showing rows: 1 to 60 Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
2	22	Multi-layered optimizing migration	Wesley Escobar	Fiction	1908	39.23	78
3	28	Expanded analyzing portal	Lisa Coffey	Fiction	1941	37.51	79
4	29	Quality-focused multi-tasking challenge	Katrina Underwood	Fiction	1905	31.12	100
5	31	Implemented encompassing conglomerati...	Melissa Taylor	Fiction	2010	21.23	44
6	39	Optimized national process improvement	Megan Goodwin	Fiction	1978	10.99	42
7	40	Adaptive didactic interface	Natalie Gonzalez	Fiction	1923	25.97	94
8	47	Reverse-engineered directional conglomer...	John Christian	Fiction	2006	20.37	90
9	62	Re-contextualized real-time strategy	Nicole Lynch	Fiction	1953	26.34	23
10	63	Polarized heuristic database	Franklin Mack	Fiction	1989	22.38	56
11	100	Synchronized client-server service-desk	James Alvarado	Fiction	1906	49.89	29
12	116	Multi-tiered foreground contingency	Jamie Gates	Fiction	1938	41.82	50

Find books published after the year 1950:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
36
37 -- 2) Find books published after the year 1950:
38 SELECT * FROM BOOKS
39 WHERE published_year > '1950';
40
```

Data Output Messages Notifications

Showing rows: 1 to 292 Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	2	Persevering reciprocal knowledge user	Mario Moore	Fantasy	1971	35.80	19
2	4	Customizable 24hour product	Christopher Andrews	Fiction	2020	43.52	8
3	5	Adaptive 5thgeneration encoding	Juan Miller	Fantasy	1956	10.95	16
4	6	Advanced encompassing implementation	Bryan Morgan	Biography	1985	6.56	2
5	8	Persistent local encoding	Troy Cox	Science Fiction	2019	48.99	84
6	9	Optimized interactive challenge	Colin Buckley	Fantasy	1987	14.33	70
7	10	Ergonomic national hub	Samantha Ruiz	Mystery	2015	24.63	25
8	11	Secured zero tolerance time-frame	Denise Barnes	Fantasy	1998	35.95	10
9	12	Polarized optimal array	Destiny Scott	Non-Fiction	1989	27.43	63
10	15	User-friendly motivating strategy	Keith Smith	Non-Fiction	1997	23.83	58
11	17	Reduced secondary core	Benjamin Peters	Fantasy	1966	5.37	45
12	18	Adaptive 4thgeneration concept	Hector Palmer	Non-Fiction	2021	39.47	32

Total rows: 292 Query complete 00:00:00.279

CRLF Ln 40, Col 1

List all customers from the Canada:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
40
41 -- 3) List all customers from the Canada:
42 SELECT * FROM Customers
43 WHERE country = 'Canada';
44
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1 of 1

	customer_id [PK] integer	name character varying (100)	email character varying (100)	phone character varying (15)	city character varying (50)	country character varying (150)
1	38	Nicholas Harris	christine93@perkins.com	1234567928	Davistown	Canada
2	415	James Ramirez	robert54@hall.com	1234568305	Maxwelltown	Canada
3	468	David Hart	stokesrebecca@gmail.com	1234568358	Thompsonfurt	Canada

Total rows: 3 Query complete 00:00:00.212

CRLF Ln 44, Col 1

Show orders placed in November 2023:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
44
45 -- 4) Show orders placed in November 2023:
46 SELECT * FROM Orders
47 WHERE order_date BETWEEN '2023-11-01' AND '2023-11-30';
48
```

Data Output Messages Notifications

Showing rows: 1 to 25 Page No: 1 of 1

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_Amount numeric (10,2)
1	4	433	343	2023-11-25	7	301.21
2	19	496	60	2023-11-17	9	316.26
3	75	291	375	2023-11-30	5	170.75
4	132	469	333	2023-11-22	7	194.32
5	137	474	471	2023-11-25	8	363.04
6	163	207	384	2023-11-23	3	101.76
7	182	129	293	2023-11-01	7	125.51
8	200	313	303	2023-11-23	1	6.57
9	213	325	447	2023-11-17	7	253.75
10	231	22	384	2023-11-11	1	33.92
11	245	386	97	2023-11-01	9	411.66
12	252	405	387	2023-11-15	5	237.10

Total rows: 25 Query complete 00:00:00.098

CRLF Ln 48, 0

Retrieve the total stock of books available:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
48
49 -- 5) Retrieve the total stock of books available:
50 SELECT SUM(stock) AS total_stock FROM Books;
51
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	total_stock bigint
1	25056

Find the details of the most expensive book:

The screenshot shows the pgAdmin 4 web interface. The top menu bar includes File, Object, Tools, Edit, View, Window, and Help. The main toolbar contains icons for connecting, refreshing, saving, and executing queries. The query editor shows the following SQL code:

```
-- 6) Find the details of the most expensive book:
SELECT * FROM Books
ORDER BY price DESC
LIMIT 1;
```

Below the query editor, the 'Data Output' tab is active, displaying a table with the results of the query. The table has 7 columns: book_id, title, author, genre, published_year, price, and stock. The results show one row with the following data:

book_id	title	author	genre	published_year	price	stock
340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88

The interface also includes a 'Messages' tab and a 'Notifications' tab. The bottom status bar shows 'Showing rows: 1 to 1' and 'Page No: 1 of 1'.

Show all Customers who ordered more than 1 quantity of a book:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
-- 7) Show all customers who ordered more than 1 quantity of a book:
SELECT * FROM Orders
WHERE quantity > 1;
```

Data Output Messages Notifications

Showing rows: 1 to 438 Page No: 1 of 1

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	1	84	169	2023-05-26	8	188.56
2	2	137	301	2023-01-23	10	216.60
3	3	216	261	2024-05-27	6	85.50
4	4	433	343	2023-11-25	7	301.21
5	5	14	431	2023-07-26	7	136.36
6	6	439	119	2024-10-11	5	249.40
7	7	195	467	2023-10-23	6	82.92
8	8	32	159	2024-05-07	4	144.84
9	9	109	407	2024-01-04	9	379.71
10	10	94	122	2024-07-09	4	123.00
11	12	454	3	2024-06-17	2	31.50
12	13	420	180	2023-06-08	5	125.45

Retrieve all orders where the total amount exceeds \$20

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
60
61 -- 8) Retrieve all orders where the total amount exceeds $20:
62 SELECT * FROM Orders
63 WHERE total_amount > 20;
64
```

Data Output Messages Notifications

Showing rows: 1 to 473 Page No: 1 of 1

	order_id [PK] integer	customer_id integer	book_id integer	order_date date	quantity integer	total_amount numeric (10,2)
1	1	84	169	2023-05-26	8	188.56
2	2	137	301	2023-01-23	10	216.60
3	3	216	261	2024-05-27	6	85.50
4	4	433	343	2023-11-25	7	301.21
5	5	14	431	2023-07-26	7	136.36
6	6	439	119	2024-10-11	5	249.40
7	7	195	467	2023-10-23	6	82.92
8	8	32	159	2024-05-07	4	144.84
9	9	109	407	2024-01-04	9	379.71
10	10	94	122	2024-07-09	4	123.00
11	11	131	206	2023-10-16	1	38.01
12	12	454	3	2024-06-17	2	31.50

List all genres available in the Books table:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
64
65 -- 9) List all genres available in the Books table:
66 SELECT DISTINCT(genre) AS all_genre FROM Books;
67
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1 of 1

	all_genre character varying (50)
1	Romance
2	Biography
3	Mystery
4	Fantasy
5	Fiction
6	Non-Fiction
7	Science Fiction

Find the book with the lowest stock:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
69 -- 10) Find the book with the lowest stock:
70 SELECT * FROM Books
71 ORDER BY stock ASC
72 LIMIT 1 ;
73
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0

Calculate the total revenue generated from all orders:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
73
74 -- 11) Calculate the total revenue generated from all orders:
75 SELECT SUM(total_amount) AS total_revenu FROM Orders;
76
77
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	total_revenu numeric
1	75628.66

Retrieve the total number of books sold for each genre:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
81 -- Advance Questions :
82
83 -- 1) Retrieve the total number of books sold for each genre:
84 SELECT b.genre ,SUM(o.quantity) AS each_genre
85 FROM Books b
86 JOIN Orders o
87 ON b.Book_id = o.Book_id
88 GROUP BY genre;
```

Data Output Messages Notifications

Showing rows: 1 to 7 Page No: 1 of 1

	genre character varying (50)	each_genre bigint
1	Romance	439
2	Biography	285
3	Mystery	504
4	Fantasy	446
5	Fiction	225
6	Non-Fiction	351
7	Science Fiction	447

Find the average price of books in the “Fantasy” genre:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
-- 2) Find the average price of books in the "Fantasy" genre:
SELECT AVG(price) AS avg_price FROM Books
WHERE genre = 'Fantasy';
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	avg_price numeric
1	25.9816901408450704

List customers who have placed at least 2 orders:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
95 -- 3) List customers who have placed at least 2 orders:
96 SELECT o.customer_id, c.name, COUNT(o.order_id) AS Order_Count FROM orders o
97 JOIN Customers c
98 ON c.customer_id = o.customer_id
99 GROUP BY o.customer_id, c.name
100 HAVING COUNT(o.order_id) >= 2
101 ORDER BY COUNT(o.order_id) ASC;
```

Data Output Messages Notifications

Showing rows: 1 to 139 Page No: 1 of 1

	customer_id integer	name character varying (100)	order_count bigint
1	225	Christopher Mccullough	2
2	81	Matthew Hall	2
3	84	Gary Blair	2
4	207	Sarah Williams	2
5	458	Yvonne Martin	2
6	109	Jacob Kelley	2
7	232	Kelli Bishop	2
8	313	David Nelson	2

Find the most frequently ordered book:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
103
104 -- 4) Find the most frequently ordered book:
105 SELECT o.book_id,b.title,COUNT(o.order_id) AS Order_Count
106 FROM Orders o
107 JOIN books b ON o.book_id = b.book_id
108 GROUP BY o.book_id,b.title
109 ORDER BY COUNT(o.order_id) DESC LIMIT 1;
110
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	book_id integer	title character varying (100)	order_count bigint
1	88	Robust tangible hardware	4

Show the top 3 most expensive books of 'Fantasy' Genre:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* x

company/postgres@PostgreSQL 17

No limit

Query Query History

```
111 -- 5) Show the top 3 most expensive books of 'Fantasy' Genre :
112 SELECT * FROM Books
113 WHERE genre = 'Fantasy'
114 ORDER BY price DESC LIMIT 3;
115
116
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	author character varying (100)	genre character varying (50)	published_year integer	price numeric (10,2)	stock integer
1	240	Stand-alone content-based hub	Lisa Ellis	Fantasy	1957	49.90	41
2	462	Innovative 3rdgeneration database	Allison Contreras	Fantasy	1988	49.23	62
3	238	Optimized even-keeled analyzer	Sherri Griffith	Fantasy	1975	48.97	72

Retrieve the total quantity of books sold by each author:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
117 -- 6) Retrieve the total quantity of books sold by each author:
118 SELECT b.author,SUM(o.quantity ) AS Total_Books_Sold
119 FROM Books b
120 JOIN Orders o ON b.book_id = o.book_id
121 GROUP BY b.author;
122
```

Data Output Messages Notifications

Showing rows: 1 to 314 Page No: 1 of 1

	author character varying (100)	total_books_sold bigint
1	Jared Cortez	10
2	Tracy Parker	11
3	Taylor Wang	9
4	Cathy Knight	6
5	Bianca Matthews	3
6	Douglas Malone	6
7	James Alvarado	9
8	Betty Cross	6
9	Michael Hill	20
10	Steven McDonald	15
11	Paul Miles	19

List the cities where customers who spent over \$30 are located:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
123 -- 7) List the cities where customers who spent over $30 are located:
124 SELECT DISTINCT c.city,o.total_amount
125 FROM Customers c
126 JOIN Orders o ON c.customer_id = o.customer_id
127 WHERE o.total_amount > 30;
```

Data Output Messages Notifications

Showing rows: 1 to 443 Page No: 1 of 1

	city character varying (50)	total_amount numeric (10,2)
1	Taylorfurt	189.45
2	Leeport	141.39
3	Port Jasonview	149.12
4	Port Aaronstad	145.44
5	Matthewfurt	328.50
6	Angelaside	42.19
7	Lindaburgh	325.92
8	Stephanieberg	156.60
9	Freemanland	198.75
10	Natashaville	399.04
11	North Joseph	125.45

Total rows: 443 Query complete 00:00:00.098

CRLF Ln 127, Col 27

Find the customer who spent the most on orders:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
128
129 -- 8) Find the customer who spent the most on orders:
130 SELECT c.name , SUM(o.total_amount) AS Total_Spent
131 FROM Customers c
132 JOIN Orders o ON c.customer_id = o.customer_id
133 GROUP BY c.name
134 ORDER BY Total_Spent DESC LIMIT 1;
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1 of 1

	name character varying (100)	total_spent numeric
1	Kim Turner	1398.90

Calculate the stock remaining after fulfilling all orders:

pgAdmin 4

File Object Tools Edit View Window Help

Welcome Last Assignment.sql* X

company/postgres@PostgreSQL 17

No limit

Query Query History

```
135 --9) Calculate the stock remaining after fulfilling all orders:
136 SELECT b.Book_id,b.title,b.stock,COALESCE(SUM(o.quantity),0) AS Order_quantity,
137        b.stock - COALESCE(SUM(o.quantity),0) AS Remaing_Quantity
138 FROM Books b
139 LEFT JOIN Orders o ON b.book_id = o.book_id
140 GROUP BY b.book_id ORDER BY b.book_id;
141
```

Data Output Messages Notifications

Showing rows: 1 to 500 Page No: 1 of 1

	book_id [PK] integer	title character varying (100)	stock integer	order_quantity bigint	remaing_quantity bigint
1	1	Configurable modular throughput	100	3	97
2	2	Persevering reciprocal knowledge user	19	0	19
3	3	Streamlined coherent initiative	27	5	22
4	4	Customizable 24hour product	8	0	8
5	5	Adaptive 5thgeneration encoding	16	8	8
6	6	Advanced encompassing implementation	2	0	2
7	7	Open-architected exuding structure	95	5	90
8	8	Persistent local encoding	84	3	81
9	9	Optimized interactive challenge	70	0	70

Total rows: 500 Query complete 00:00:00.121

CRLF Ln 140, Col 39

ADVANTAGE

- ◆ It improves practical SQL skills by applying queries to real-world hospital data.
- ◆ Helps in understanding how data-driven decisions can be made in healthcare.
- ◆ Strengthens knowledge of joins, grouping, filtering, and date functions.
- ◆ Builds confidence in solving business problems through structured queries.



THANK YOU