

SQL PROJECT

ONLINE BOOK STORE

This project is a relational database system designed for managing an online bookstore, covering book inventory, customer orders, and payment processing. It ensures smooth operations by organizing data on books, users, and transactions for efficient sales management.



SQL Database Management System for “Online Bookstore”

Agenda for SQL Project on Online Bookstore

- 01 Structure of the database (tables, relationships, and schema)
- 02 Sample queries for retrieving book details, customer orders, and payments
- 03 Integration with the website for real-time data handling



Tip: Use links to go to a different page inside your presentation. Links work best for pages like this one!



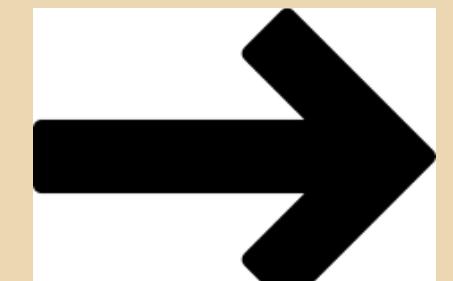
How: Highlight text, click on the link symbol on the toolbar, and select the page in your presentation that you want to connect.

Kindly delete this note after editing this page. Thank you!

**SOLVING ALL
20 QUESTIONS
INCLUDES BASICS
AND ADVANCE
QUESTIONS**



BASICS QUESTIONS



Online book store/postgres@PostgreSQL 13* X

Online book store/postgres@PostgreSQL 13

No limit

Scratch Pad X

Query History

55

56 -- 1) Retrieve all books in the "Fiction" genre:

57

58 **SELECT * FROM BOOKS**

59 **WHERE Genre = 'Fiction'**

60

61

62

63

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 Andre...	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

Total rows: 60 Query complete 00:00:00.113 CRLF In 60.5

Online book store/postgres@PostgreSQL 13* X

Online book store/postgres@PostgreSQL 13

No limit

Query History

Scratch Pad

63

64

65 -- 2) Find books published after the year 1950:

66

67 **SELECT * FROM BOOKS**

68 **WHERE Published_Year>1950;**

69

70

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	1000	25.05	10

Object Explorer Online book store/postgres@PostgreSQL 13*

Query History Scratch Pad

```
70
71 -- 3) List all customers from the Canada:
72
73 SELECT * FROM Customers
74 WHERE Country = 'Canada'
75
76
77
```

Data Output Messages Notifications

Showing rows: 1 to 3 Page No: 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 00:00:00.110 CRLF Ln 77, Col 1

Online book store/postgres@PostgreSQL 13

No limit

Query History

```
-- 4) Show orders placed in November 2023:  
SELECT * FROM ORDERS  
WHERE Order_Date  
Between '2023-11-01' And '2023-11-30';
```

Data Output Messages Notifications

Showing rows: 1 to 25 Page No: 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

Object Explorer

Online book store/postgres@PostgreSQL 13*

Query History

Scratch Pad

```
87  
88 -- 5) Retrieve the total stock of books available:  
89  
90 SELECT SUM(stock) as Total_Stock  
91 From Books;  
92  
93
```

Data Output

	total_stock	bigint
1	25056	

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

This screenshot shows the pgAdmin 4 interface for a PostgreSQL database named 'Online book store'. The left sidebar is the Object Explorer, displaying various database objects like FTS Dictionaries, Functions, and Tables. The main area has tabs for 'Query' and 'Query History', with the 'Query' tab active. The query itself is a simple SQL statement to calculate the total stock of books. Below the query, the 'Data Output' tab shows the result of the query: a single row with a column labeled 'total_stock' containing the value '25056'. The status bar at the bottom indicates 'Showing rows: 1 to 1 | Page No: 1 of 1'.

Window Help

Online book store/postgres@PostgreSQL 13*

Online book store/postgres@PostgreSQL 13

No limit

Query History

Scratch Pad

94

95

96 -- 6) Find the details of the most expensive book:

97

98 `select * from books`

99 `order by price`

100 `DESC limit 1;`

101

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	340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88

Online book store/postgres@PostgreSQL 13*

Online book store/postgres@PostgreSQL 13

No limit

Query History

Scratches

```
102
103 -- 7) Show all customers who ordered more than 1 quantity
104
105 select * from orders
106 where quantity > 1;
107
108
```

Data Output Messages Notifications

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

	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

Window Help

Online book store/postgres@PostgreSQL 13*

tionaries
ers
nplates
Tables
ns
lized View
rs
ires
ces
3)
s
mbers
s
Function:

110 -- 8) Retrieve all orders where the total amount
111 -- exceeds \$20:
112
113 **select * from orders**
114 **where total_amount > 20;**
115
116

Data Output Messages Notifications

Showing rows: 1 to 473 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

Window Help

Online book store/postgres@PostgreSQL 13* X

onaries
ers
plates
ables
s
zed View
s
es
es
)
ners
unction:
124

Query Query History

117
118 -- 9)List all genres available in the Books table:
119
120
121 **select distinct genre from books;**
122
123
124

Data Output Messages Notifications

Showing rows: 1 to 7 | Page No: 1

	genre
1	Romance
2	Biography
3	Mystery
4	Fantasy
5	Fiction
6	Non-Fiction
7	Science Fiction

Window Help

Online book store/postgres@PostgreSQL 13*

Online book store/postgres@PostgreSQL 13

No limit

Query History

Scratch Pad

```
125
126 -- 10) Find the book with the lowest stock:
127
128 select * from books order by stock ;
129
130
131
```

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	44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0
2	163	Object-based eco-centric challenge	Douglas McCarthy	Non-Fiction	1905	19.11	0
3	127	Business-focused real-time benchmark	David Nelson	Science Fiction	1997	11.66	0
4	60	Robust eco-centric capacity	Brian Haney	Biography	1990	35.14	0
5	378	Future-proofed heuristic function	Samantha McClain	Romance	1903	6.01	0
6	137	Networked contextually-based encryption	Leslie Montoya	Biography	1942	26.33	1
7	449	Universal homogeneous adapter	Gina Lopez	Romance	1966	40.17	1
8	259	Pre-emptive incremental secured line	Jason Perez	Science Fiction	1994	14.48	1
9	19	Progressive asymmetric Internet solution	Sean Miller	Science Fiction	1990	11.31	1

Window Help

Online book store/postgres@PostgreSQL 13* X

Online book store/postgres@PostgreSQL 13

No limit

Query Query History

Scratch Pad

```
133 -- 11) Calculate the total revenue  
134 -- generated from all orders:  
135  
136 select sum(total_amount) as Revenue  
137 from orders;
```

```
138  
139
```

Data Output Messages Notifications

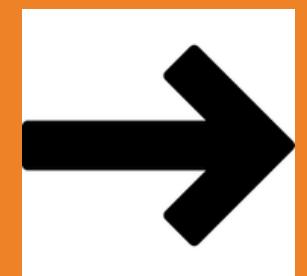
+ SQL

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

	revenue	numeric
1	75628.66	



ADVANCE QUESTIONS



Window Help

Online book store/... X Online book store/postgres@PostgreSQL 13* X

Online book store/postgres@PostgreSQL 13

No limit

Scratch Pad X

Query History

6

7

8 --1) Retrieve the total number of books sold
9 -- for each genre

10

11 **select** b.Genre,**sum**(o.quantity) **as** total_books_sold
12 **from** orders o
13 **join** books b **on** o.book_id = b.book_id
14 **Group by** b.Genre;

Data Output Messages Notifications

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

	genre	total_books_sold
1	Romance	439
2	Biography	285
3	Mystery	504
4	Fantasy	446
5	Fiction	225
6	Non-Fiction	351
7	Science Fiction	447

New Window Help

Online book store/... X Online book store/postgres@PostgreSQL 13* X

Dictionaries
Parsers
Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Triggers
Tables (3)
Books
Customers
Orders
User Functions
Views
Functions

Query History Scratch Pad X

16
17
18 --2) Find the average price of books in the
19 -- "Fantasy" genre
20
21
22 SELECT AVG(price) AS Average_Price
23 FROM Books
24 WHERE Genre = 'Fantasy';

Data Output Messages Notifications

average_price numeric

	average_price
1	25.9816901408450704

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

This screenshot shows a pgAdmin 4 interface connected to a PostgreSQL 13 database. The left sidebar lists various database objects like Dictionaries, Parsers, and Tables. The main window has two tabs: 'Query' (selected) and 'Query History'. The 'Query' tab contains a multi-line SQL statement. The SQL code includes comments explaining the purpose: finding the average price of books in the 'Fantasy' genre. The code uses the AVG function on the 'price' column from the 'Books' table, filtered by the 'Genre' column being 'Fantasy'. The results are displayed in a table titled 'average_price' with one row containing the value 25.9816901408450704. Below the table, status bars show 'Showing rows: 1 to 1' and 'Page No: 1 of 1'.

Online book store/postgres@PostgreSQL 13

No limit

Query History

Scratch Pad

```
-- 3) List customers who have placed
-- at least 2 orders:
SELECT o.customer_id, c.name, COUNT(o.order_id)
AS ORDER_COUNT
FROM orders o
JOIN customers c ON o.customer_id=c.customer_id
GROUP BY o.customer_id, c.name
HAVING COUNT(Order_id) >=2;
```

Data Output Messages Notifications

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

	customer_id	name	order_count
1	225	Christopher McCullough	2
2	418	Kiara Blankenship MD	3
3	322	William Cameron	3
4	325	Emily Vargas	4
5	376	Justin Donaldson	2
6	486	Melanie Kelly	2
7	461	Crystal Pierce	3

Edit View Window Help

Online book store/... × Online book store/postgres@PostgreSQL 13* ×

Online book store/postgres@PostgreSQL 13

No limit

Query History Scratch Pad

38

39

40 -- 4) Find the most frequently ordered book:

41

42 `select o.book_id , b.title , count(order_id)`

43 `as order_count`

44 `from orders o`

45 `join books b on o.book_id=b.book_id`

46 `group by o.book_id,b.title`

47 `order by order_count desc limit 1;`

Data Output Messages Notifications

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

	book_id	title	order_count
1	88	Robust tangible hardware	4

Window Help

Online book store/... Online book store/postgres@PostgreSQL 13*

Online book store/postgres@PostgreSQL 13

No limit

Query History Scratch Pad

48

49

50 -- 5) Show the top 3 most expensive

51 -- books of 'Fantasy' Genre :

52

53

54 **SELECT * FROM books**

55 **WHERE genre = 'Fantasy'**

56 **ORDER BY price DESC LIMIT 3;**

57

Data Output Messages Notifications

SQL

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

Online book store/... X Online book store/postgres@PostgreSQL 13 X

Online book store/postgres@PostgreSQL 13

No limit

Query History

Scratch Pad

```
61 -- 6) Retrieve the total quantity of
62 -- books sold by each author:
63
64 select b.author, sum(o.quantity) as total_books_sold
65 from orders o
66 join books b on o.book_id=b.book_id
67 group by b.author;
68
69
```

Data Output Messages Notifications

author character varying (100) total_books_sold bigint

	author	total_books_sold
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

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

Total rows: 314 Query complete 00:00:00.108 CRLF Ln 67, Col 19

Edit View Window Help



> FTS Dictionaries

> FTS Parsers

> FTS Templates

> Foreign Tables

> Functions

> Materialized Views

> Operators

> Procedures

> Sequences

> Tables (3)

> books

> customers

> orders

> Trigger Functions

> Types

> Views

Subscriptions

Company

Employee

Customer

Order

Cast

Catalogs

Event Triggers

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

> Aggregates

> Collations

> Domains

Total rows: 443

Query complete 00:00:00.118

CRLF Ln 79, Col 1



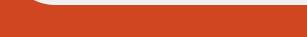
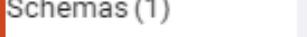
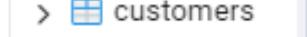
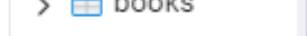
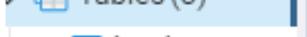
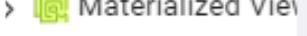
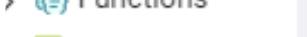
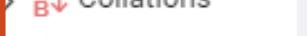
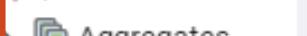
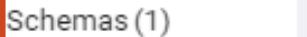
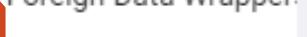
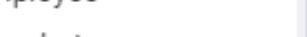
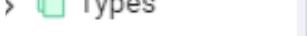
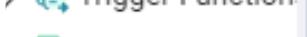
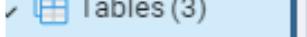
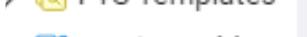
Online book store/postgres@PostgreSQL 13*



Online book store/postgres@PostgreSQL 13



No limit



min 4

Select Tools Edit View Window Help

Browser

> FTS Dictionaries
> FTS Parsers
> FTS Templates
> Foreign Tables
> Functions
> Materialized Views
> Operators
> Procedures
> Sequences
Tables (3)
> books
> customers
> orders
> Trigger Functions
> Types
> Views
> Subscriptions
> company
> employee
pizzahut
> Casts
> Catalogs
> Event Triggers
> Extensions
> Foreign Data Wrappers
> Languages
> Publications
Schemas (1)

Online book store/postgres@PostgreSQL 13* Online book store/postgres@PostgreSQL 13

Online book store/postgres@PostgreSQL 13 No limit

Query Query History

o1

```
82 -- 8) Find the customer who spent  
83 -- the most on orders:
```

84

```
85 select c.customer_id,c.name,sum(o.total_amount)  
86 as total_spent  
87 from orders o  
88 join customers c on o.customer_id =c.customer_id  
89 group by c.customer_id,c.name  
90 order by total_spent desc limit 1;
```

Data Output Messages Notifications

SQL

Showing rows: 1 to 1 Page No: 1 of

	customer_id [PK] integer	name character varying (100)	total_spent numeric
1	457	Kim Turner	1398.90

Online book store/... x Online book store/postgres@PostgreSQL 13*

Online book store/postgres@PostgreSQL 13

No limit

Query History Scratch Pad

```
97 -- 9) Calculate the stock remaining after
98 -- fulfilling all orders:
99
100 select b.book_id, b.title, b.stock, coalesce(sum(quantity),
101      as order_quantity,
102      b.stock - coalesce(sum(quantity), 0) as order_quantity
103 from books b
104 left join orders o on b.book_id=o.book_id
105 group by b.book_id order by b.book_id;
```

Data Output Messages Notifications

book_id [PK] integer title character varying (100) stock integer order_quantity bigint order_quantity bigint

	book_id	[PK] integer	title	character varying (100)	stock	integer	order_quantity	bigint	order_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	

Total rows: 500 Query complete 00:00:00.121 CRLF Ln 104, Col 42

to search

26°C Haze ENG 7:12 PM

**THANKS FOR
WATCHING**

