

# Data Analytics Agent (smol)

“For when people bother you for adhoc requests”

# You have many tables, that links to each other

Invoice Table								Product Table			Address		
invoice_id	timestamp	subtotal	gst	total	origin_address_id	origin_address_type		id	product_code	name	id	street	postal code
1	2025-02-06 13:45:43.679803+00	364	36	400	1	1		1	D0001	Helicopter Drone	1	outlet 1 street	123
2	2025-02-03 13:45:43.679803+00	227.5	22.5	250	4	2		2	R0001	Robot Dolly	2	outlet 2 street	1234
3	2025-01-26 13:45:43.679803+00	500.5	49.5	550	2	1		3	T0001	Motorized Tripod	3	outlet 3 street	12345
4	2025-01-22 13:45:43.679803+00	364	36	400	4	2					4	outlet 4 street	123456
5	2025-01-17 13:45:43.679803+00	910	90	1000	2	1					5	outlet 5 street	1234567
6	2025-01-10 13:45:43.679803+00	728	72	800	3	1							
7	2024-12-26 13:45:43.679803+00	910	90	1000	3	1							
8	2024-12-21 13:45:43.679803+00	364	36	400	5	2							
9	2024-12-11 13:45:43.679803+00	637	63	700	5	2							
10	2025-12-06 13:45:43.679803+00	728	72	800	5	2							
Invoice Item								Outlet Table			Warehouse Table		
id	invoice_id	product_id	quantity	price	line_total			id	name	address_id	id	name	address_id
1	1	1	1	200	200			1	Outlet 1	1	1	Warehouse 1	4
2	1	2	2	100	200			2	Outlet 2	2	2	Warehouse 2	5
3	2	3	5	50	250			3	Outlet 3	3			
4	3	1	2	200	400								
5	3	3	7	50	350								
6	4	1	2	200	400								
7	5	2	10	100	1000								
8	6	1	1	200	200								
9	6	2	6	100	600								
10	7	1	5	200	1000								
11	8	1	2	200	400								
12	9	2	7	100	700								
13	10	3	20	40	800								
								Total Sales By Location					
								Outlet 1		400			
								Outlet 2		1550			
								Outlet 3		1800			
								Warehouse 1		650			
								Warehouse 2		1900			

# And when stakeholders make an adhoc request,

Please track all invoices by location and create a bar chart where x axis is outlet or warehouse name and y axis is sales amount

# automate away !

## Text2SQL & Code-Generated Visualisation

Chatbot

Factor R0001: Overall, D0001 has a higher total sales figure, driven by its dominance in Outlet 3. These findings suggest outlet-specific strategies may be necessary to optimize sales for each product.

Please track all invoices by location and create a bar chart where x axis is outlet or warehouse name and y axis is sales amount

Our analysis reveals a significant variation in sales performance across different locations, with Outlet 3 and Warehouse 2 being the top-performing outlets. The data suggests that warehouses tend to generate higher sales compared to outlets, with an average sales amount of \$1225 for warehouses versus \$978 for outlets. This insight can inform strategic decisions on resource allocation and optimization of sales channels.



Select a Default Prompt

I want to find out how D0001 and R0001 perform across outlets 1 to 3. Please output the results in a table where column headers are outlet names and row headers are product names


I want to find out what percentage of my total sales come from warehouse sales

●

 Please track all invoices by location and create a bar chart where x axis is outlet or warehouse name and y axis is sales amount



Enter message or upload file...



Data retrieved

location_name	sales_amount
Outlet 1	400
Outlet 2	1550
Outlet 3	1800
Warehouse 1	650
Warehouse 2	1900

Text2SQL result

```
SELECT COALESCE(o.name, w.name) AS location_name,
SUM(i.total) AS sales_amount
FROM datachat_invoice i
LEFT JOIN datachat_outlet o ON i.origin_address_id = o.address_id
AND i.origin_address_type = 1
LEFT JOIN datachat_warehouse w ON i.origin_address_id = w.address_id
AND i.origin_address_type = 2
GROUP BY location_name;
```

Visualisation Output

Sales Amount by Location



Code Output

```
1 import matplotlib.pyplot as plt
2
3 def plot_sales_by_location(location_name, sales_amount):
4     # Create a bar chart
5     plt.figure(figsize=(10, 6))
6     plt.bar(location_name, sales_amount, color='skyblue')
7
8     # Add titles and labels
9     plt.title('Sales Amount by Location')
10    plt.xlabel('Location Name')
11    plt.ylabel('Sales Amount')
12
13    # Rotate x-axis labels for better readability if needed
14    plt.xticks(rotation=45)
15
16    # Display the plot
17    plt.tight_layout()
18    plt.show()
19
20 # Data
21 location_name = ['Outlet 1', 'Outlet 2', 'Outlet 3', 'Warehouse 1', 'Warehouse 2']
22 sales_amount = [400, 1550, 1800, 650, 1900]
23
24 # Call the function to plot the data
25 plot_sales_by_location(location_name, sales_amount)
```