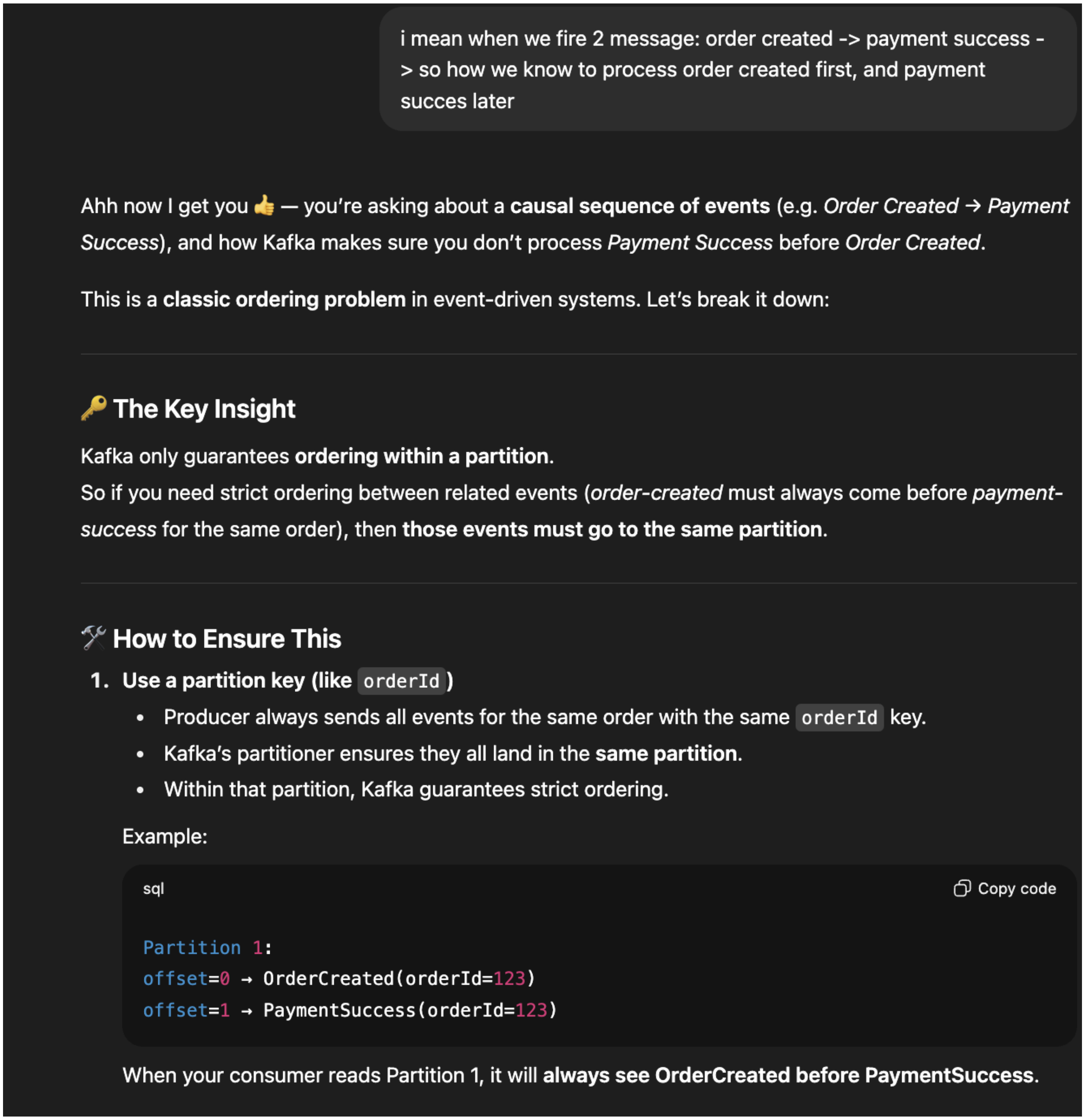
1. How does Kafka ensure the order of 2 messages: Order Created -> Payment Success?

- Fire it into the same topic.
- Offset to show the order of the message in the same partition.



i mean when we fire 2 message: order created -> payment success -> so how we know to process order created first, and payment succes later

Ahh now I get you 👍 — you're asking about a **causal sequence of events** (e.g. *Order Created → Payment Success*), and how Kafka makes sure you don't process *Payment Success* before *Order Created*.

This is a **classic ordering problem** in event-driven systems. Let's break it down:

🔑 The Key Insight

Kafka only guarantees **ordering within a partition**.
So if you need strict ordering between related events (*order-created* must always come before *payment-success* for the same order), then **those events must go to the same partition**.

🛠️ How to Ensure This

1. **Use a partition key (like `orderId`)**
   - Producer always sends all events for the same order with the same `orderId` key.
   - Kafka's partitioner ensures they all land in the **same partition**.
   - Within that partition, Kafka guarantees strict ordering.

   Example:

```sql
Partition 1:
offset=0 → OrderCreated(orderId=123)
offset=1 → PaymentSuccess(orderId=123)
```

   When your consumer reads Partition 1, it will **always see OrderCreated before PaymentSuccess**.