Apache Kafka: Topics, Partitions & Offsets

1. Kafka Topics (The Basics)

Kafka mein **Topic** sabse important concept hai.

- **Definition:** Topic ek particular **data stream** hai.
- **Analogy:** Ye Database ke **Table** jaisa hota hai, lekin bina kisi constraints ke.
- **Naming:** Har topic ka ek naam hota hai (e.g., `logs`, `purchases`, `twitter_tweets`, `trucks_gps`).
- **Storage:** Aap topic mein kisi bhi format ka data bhej sakte hain (JSON, Avro, Text, Binary).
- **No Querying:** Aap topic ko database ki tarah query (`SELECT * FROM...`) nahi kar sakte.
    - Data add karne ke liye **Producers** chahiye.
    - Data read karne ke liye **Consumers** chahiye.

---

2. Partitions & Offsets (Internal Structure)

Ek Topic bahut bada ho sakta hai, isliye usse chhotay parts mein divide kiya jata hai jinhe **Partitions** kehte hain.

- **Partitions:** Ek topic mein kitne bhi partitions ho sakte hain (e.g., 100 partitions). Humare example mein hum 3 partitions lenge (0, 1, aur 2).
- **Ordering:** Har partition ke andar messages ordered hote hain.
- **Offsets:** Har message ko ek unique ID milti hai jise **Offset** kehte hain.
    - Offset 0 se start hota hai aur increase hota rehta hai (0, 1, 2, 3...).
    - **Note:** Offset sirf uss specific partition ke liye meaningful hai. Partition 0 ka Offset 3 aur Partition 1 ka Offset 3 alag-alag data hain.

Text Diagram: Topic with 3 Partitions

```
Topic: trucks_gps

Partition 0: [Msg 0] -> [Msg 1] -> [Msg 2] -> [Msg 3] ... (Writes continue)
             (Oldest)                          (Newest)

Partition 1: [Msg 0] -> [Msg 1] -> [Msg 2] ...

Partition 2: [Msg 0] -> [Msg 1] -> [Msg 2] -> [Msg 3] -> [Msg 4] ...
```

Note: Data hamesha end mein append hota hai.

---

3. Critical Concepts (Yaad rakhne wali baatein)

1. **Immutability (Change nahi ho sakta):**
    - Ek baar data partition mein write ho gaya, toh wo **change ya update nahi ho sakta**.
    - Aap Kafka mein data delete bhi nahi kar sakte (Data sirf append hota hai).
2. **Data Retention:**
    - Data hamesha ke liye store nahi hota.
    - **Default Retention:** 1 Week (Uske baad data gayab ho jayega).
3. **Offset Reuse:**
    - Agar purana data delete bhi ho jaye, tab bhi purane Offsets reuse nahi hote. Sequence hamesha aage badhta hai.
4. **Order Guarantee:**
    - **Very Important:** Message ka order sirf **Partition ke andar** guaranteed hota hai.
    - Pure Topic ke across (Partition 0 vs Partition 1) koi order guarantee nahi hoti.

---

4. Real World Example: Trucks GPS

Samajhne ke liye ek **Truck Fleet** ka example lete hain.

- **Scenario:** Aapke paas bahut saare trucks hain. Har truck ke paas GPS hai jo har 20 seconds mein apni location bhejta hai.
- **Topic Name:** `trucks_gps`
- **Producer:** Trucks (jo data bhej rahe hain: Truck ID, Latitude, Longitude).
- **Partitioning:** Maan lijiye humne topic mein 10 Partitions banaye hain.
  - Agar hum koi "Key" specify nahi karte, toh data random partitions (0 se 9) mein jayega.

Benefit of Kafka here: Ek hi data stream (`trucks_gps`) ko multiple log use kar sakte hain:

1. **Service A (Dashboard):** Real-time location dikhane ke liye.
2. **Service B (Notifications):** Customer ko SMS bhejne ke liye jab truck paas pahunche.

---

5. Summary Checklist

- [x] **Topic:** Data ka stream (Table jaisa).
- [x] **Partition:** Topic ka sub-part jahan data actual mein store hota hai.
- [x] **Offset:** Message ki ID (0, 1, 2...) jo sirf partition ke andar unique hai.
- [x] **Immutability:** Data edit/delete nahi kar sakte.
- [x] **Order:** Sirf partition level par maintain hota hai, global level par nahi.

Next Step: Would you like to create a visual diagram of **Producers and Consumers** interacting with these Topics?