Kafka Day 6 – Revision Notes (Failure & Recovery)

---

1. Kafka Broker Failure Recovery

Handled well. You explained how ISR helps with leader election and availability.
✅ Missed: controlled shutdown vs unclean leader election trade-off.

---

2. Unclean Leader Election

Risk = data loss if out-of-sync replica is elected.
Teams enable it under pressure for availability.
✅ Config: `unclean.leader.election.enable=false`

---

3. Producer Retries & Ordering

Retries can cause duplication unless `enable.idempotence=true`.

Missed: `max.in.flight.requests.per.connection` affects ordering.

---

4. High Availability Configs

Replication factor = 3, `min.insync.replicas` = 2, `acks=all`.
✅ Missed: validate ISR shrinkage, broker config to prevent unclean election.

---

5. Log Compaction

Good recall of compacted topics retaining latest key.
✅ Strengthen: compare with log retention in long outages.

---

6. Lag Debugging Strategy

Mentioned: increase consumers, reduce batch size, improve downstream.
✅ Missed: Kafka-specific configs like `fetch.min.bytes`, `max.poll.records`.

---

7. Data Durability after Restart

Flush-to-disk was mentioned.
✅ Missed: exact config `log.flush.interval.ms`, role of segment flush.

---

8. `min.insync.replicas` Significance
Explained well — how it protects write safety.
✅ Could add: what happens when ISR < min (producer errors).

---

9. Batch Size Trade-offs
Larger = better throughput, but recovery delay risk.
Smaller = fast retry, lower throughput. You covered this well.

---

10. Rebalancing Impact
Impact: latency spikes, consumer lag.
✅ Covered: CooperativeStickyAssignor, rebalance tuning.
Keep reinforcing: **cause** → **effect** → **mitigation**.

---

11. Kafka Producer Retries (Missed in Live Review)
Retries can cause duplication unless idempotence is on.
Use `retries`, `retry.backoff.ms`, `max.in.flight.requests.per.connection`.
Always combine with `acks=all` and `enable.idempotence=true`.