Provide a concise overview of SQLi and DDoS, including risk, indicators, and key mitigations (defensive focus only).

1) SQL Injection (SQLi) — Summary

Definition:

SQL Injection is a web application vulnerability where untrusted input is handled in a way that can change the meaning of a database query. This can allow unauthorized access to data or unintended database actions.

Business Impact:

- Data breaches (PII, credentials, financial records)
- Account takeover and unauthorized privilege changes
- Data manipulation or deletion
- Compliance, legal, and reputational damage

Common Causes:

- Building SQL queries using string concatenation with user input
- Weak server-side input validation
- Over-privileged database accounts
- Exposing detailed database errors to users

Detection Indicators (Defensive):

- Spike in database or application query errors
- Repeated unusual inputs in URL parameters or form fields
- Increased failed login/search requests with abnormal patterns
- WAF/IDS alerts related to SQLi attempts
- Unexpected query performance degradation

Top Mitigations:

- Use parameterized queries / prepared statements everywhere

- Enforce server-side input validation (allow-lists where possible)

- Apply least privilege to database users and roles

- Implement safe error handling (generic user messages, detailed internal logs)

- Add automated security checks (SAST/DAST) and regular code reviews

2) Distributed Denial of Service (DDoS) — Summary

Definition:

A DDoS attack aims to make a website or service unavailable by overwhelming it with high volumes of traffic or requests from many sources.

Business Impact:

- Downtime and service disruption

- Revenue loss, SLA breaches, and customer churn

- Increased operational costs (incident response, scaling, bandwidth)

- Damage to trust and brand reliability

Common Types (High-Level):

- **Volumetric:** floods bandwidth with huge traffic volumes

- **Protocol-based:** exhausts network or infrastructure resources

- **Application-layer:** overloads specific endpoints (e.g., login/search)

Detection Indicators (Defensive):

- Sudden traffic spikes (RPS/PPS/bandwidth)

- Higher latency, timeouts, 5xx errors

- Unusual traffic sources (new geographies/ASNs) or endpoint concentration

- Resource saturation (CPU/memory) at proxy/load balancer/app servers

- Increased WAF/CDN blocks/challenges

Top Mitigations:

- Use CDN + DDoS protection to absorb/filter traffic upstream

- Implement rate limiting and endpoint-specific throttling

- Enable and tune WAF/bot management rules

- Improve resilience with caching, autoscaling, and redundancy

- Maintain an incident response runbook (triage → mitigation → comms → review)

3) Conclusion

SQLi threatens confidentiality and integrity through unsafe database interactions, while DDoS threatens availability by overwhelming systems. Strong prevention combines secure coding, least privilege, monitoring, and layered network protections (WAF/CDN/rate limits), supported by a clear response plan.