

BASI DI DATI E SQL

Progettazione Database

1. Che cos'è un DBMS e quali sono le sue funzioni principali? Database Management System: software per gestire basi di dati. Funzioni: gestione dati, controllo accessi, backup, ripristino, transazioni.

2. Spiega il processo di normalizzazione e le prime tre forme normali.

- **1NF:** Eliminare gruppi ripetuti, valori atomici
- **2NF:** 1NF + eliminare dipendenze parziali dalla chiave primaria
- **3NF:** 2NF + eliminare dipendenze transitive tra attributi non-chiave

3. Differenza tra modello concettuale (E-R) e modello logico. Concettuale: Rappresenta entità e relazioni ad alto livello, indipendente dal DBMS **Logico:** Trasforma il concettuale in tabelle con chiavi primarie/esterne, specifico per DBMS relazionale

4. Che cosa sono i vincoli di integrità referenziale? Vincoli che garantiscono coerenza tra tabelle collegate. La foreign key deve sempre riferirsi a una primary key esistente nella tabella padre.

5. Quando usare una relazione molti-a-molti e come implementarla? Quando due entità possono avere più istanze correlate. Si implementa con una tabella di collegamento contenente le chiavi primarie di entrambe le entità.

SQL e Query

6. Differenza tra INNER JOIN, LEFT JOIN e RIGHT JOIN.

- **INNER:** Solo record con corrispondenza in entrambe le tabelle
- **LEFT:** Tutti i record della tabella sinistra + corrispondenze da destra
- **RIGHT:** Tutti i record della tabella destra + corrispondenze da sinistra

7. Quando usare GROUP BY e HAVING? GROUP BY: Raggruppa record per valore comune di un campo **HAVING:** Filtra i gruppi dopo il raggruppamento (come WHERE ma per gruppi)

8. Differenza tra DELETE, DROP e TRUNCATE.

- **DELETE:** Rimuove record da tabella (con WHERE)
- **DROP:** Elimina completamente la tabella
- **TRUNCATE:** Svuota la tabella mantenendo la struttura

9. Che cosa sono le transazioni e le proprietà ACID? **Transazione:** Sequenza di operazioni atomiche **ACID:** Atomicità, Consistenza, Isolamento, Durabilità

10. Come ottimizzare le performance di una query SQL? Usare indici, evitare SELECT *, usare WHERE invece di HAVING quando possibile, limitare i risultati con LIMIT.

PROGRAMMAZIONE WEB

HTML e CSS

11. Differenza tra tag block e inline in HTML. **Block:** Occupano tutta la larghezza (div, p, h1), iniziano nuova riga **Inline:** Occupano solo lo spazio necessario (span, a, strong), sulla stessa riga

12. Che cos'è il CSS e vantaggi nell'usarlo. Cascading Style Sheets: linguaggio per stilizzare HTML. Vantaggi: separazione contenuto/presentazione, riutilizzo, manutenzione facilitata.

13. Differenza tra id e class in CSS. **id:** Identificatore unico, selettore `#nome`, priorità alta **class:** Riutilizzabile, selettore `.nome`, priorità media

PHP e Programmazione Server-Side

14. Differenza tra GET e POST.

- **GET:** Dati nell'URL, visibili, limitati, per recuperare dati
- **POST:** Dati nascosti nel body, illimitati, per inviare dati sensibili

15. Che cosa sono le sessioni PHP e come funzionano? Meccanismo per mantenere dati utente tra pagine diverse. `session_start()` inizia sessione, `$_SESSION[]` memorizza dati.

16. Differenza tra include e require in PHP.

- **include:** Genera warning se file non trovato, script continua
- **require:** Genera errore fatale se file non trovato, script si ferma

17. Come prevenire SQL Injection? Usare prepared statements, validare input, sanitizzare dati, usare parametri bindati.

18. Come gestire gli errori in PHP? try-catch per eccezioni, `error_reporting()` per controllare livello errori, log degli errori.

SICUREZZA INFORMATICA

Crittografia e Autenticazione

19. Differenza tra crittografia simmetrica e asimmetrica. Simmetrica: Stessa chiave per cifrare/decifrare, veloce **Asimmetrica:** Coppia chiavi pubblica/privata, più sicura per scambio chiavi

20. Che cos'è l'hashing e quando si usa? Funzione unidirezionale che produce fingerprint dei dati. Usato per password, integrità dati, firme digitali.

21. Principali minacce alla sicurezza web. SQL Injection, XSS, CSRF, Broken Authentication, Session Hijacking, DDoS.

22. Come funziona HTTPS e perché è importante? HTTP su SSL/TLS, cripta comunicazione client-server, garantisce autenticità e integrità.

23. Che cos'è la firma digitale? Hash del documento cifrato con chiave privata, garantisce autenticità e non ripudio.

RETI E PROTOCOLLI

Modello OSI e TCP/IP

24. Descrivi i 7 livelli del modello OSI.

1. Fisico (bit), 2. Data Link (frame), 3. Network (IP), 4. Transport (TCP/UDP),
5. Session, 6. Presentation (crittografia), 7. Application (HTTP, FTP)

25. Differenza tra TCP e UDP. TCP: Affidabile, connection-oriented, controllo errori, ordinamento **UDP:** Veloce, connectionless, nessuna garanzia, real-time

26. Che cos'è il DNS e come funziona? Domain Name System: traduce nomi domini in indirizzi IP. Gerarchia di server DNS che si interrogano a cascata.

27. Differenza tra IP statico e dinamico. Statico: Indirizzo fisso assegnato manualmente **Dinamico:** Indirizzo assegnato automaticamente da DHCP, può cambiare

28. Che cos'è il NAT e a cosa serve? Network Address Translation: traduce IP privati in pubblici, consente a più dispositivi di condividere un IP pubblico.

Protocolli Applicativi