**1. Was ist eine Relationale Datenbank?**

Eine **Relationale Datenbank** speichert Daten in **Tabellen** (Relationen), die über gemeinsame Spalten miteinander **verknüpft** sind. Jede Tabelle hat Zeilen (Tupel) und Spalten (Attribute).

**2. Nennen Sie drei verschiedene relationale Datenbank Management Systeme DBMS?**

* **MySQL**
* **PostgreSQL**
* **Oracle Database**

**3. Welche Aufgaben erfüllt ein DBMS für Sie als Entwickler?**

Ein DBMS **verwaltet Daten persistent, sichert deren Integrität und Konsistenz**, ermöglicht **effizienten Datenzugriff** (Lesen, Schreiben), kontrolliert **Berechtigungen** und **parallele Zugriffe**, optimiert Abfragen und bietet Mechanismen für **Datensicherung und -wiederherstellung**.

**4. Wie greifen Benutzer (keine Entwickler) auf eine Datenbank zu?**

Benutzer greifen indirekt über **Anwendungssoftware** (z.B. Web-Apps, Desktop-Programme) oder **Berichtstools** zu, die eine benutzerfreundliche Oberfläche bieten und im Hintergrund mit der Datenbank kommunizieren.

**5. Handelt es sich bei SQL um eine Programmiersprache?**

**Nein, SQL ist keine vollwertige Programmiersprache**, sondern eine **deklarative Sprache** zur Definition und Manipulation von Datenbanken. Sie beschreibt *was* getan werden soll, nicht *wie*. Erweiterungen wie PL/SQL sind prozedural.

**6. Ist die Sprache SQL auf allen Datenbanken gleich?**

**Nein.** Es gibt einen **ANSI/ISO-Standard**, aber Datenbankhersteller implementieren oft **spezifische Dialekte und Erweiterungen** (z.B. bei Datentypen, Funktionen, Syntax). Grundlegende Befehle sind jedoch meist gleich.

**7. Was versteht man unter NoSQL Datenbank?**

**NoSQL** ist ein Oberbegriff für Datenbanksysteme, die **nicht dem relationalen Modell folgen** und oft keine SQL verwenden. Sie sind für **Skalierbarkeit, Flexibilität** (schemalos) und verteilte Umgebungen optimiert und nutzen verschiedene Datenmodelle (Dokumente, Schlüssel-Wert, Graphen).

**8. Nennen Sie drei NoSQL Datenbanksysteme?**

* **MongoDB** (Dokumentenorientiert)
* **Apache Cassandra** (Spaltenorientiert)
* **Redis** (Schlüssel-Wert)

**9. Was versteht man unter Datenbank Schema?**

Das **Datenbank Schema** ist der **logische Bauplan** einer Datenbank. Es definiert Tabellen, Spalten, Datentypen, Beziehungen, Constraints und Indizes – also die gesamte Struktur. Z.b. Visio

**10. Wozu dient ein Entity Relationship Model (ERM)?**

Ein **ERM** ist ein **grafisches Werkzeug zur Modellierung** einer Datenbankstruktur. Es visualisiert **Entitäten** (Objekte) und ihre **Beziehungen** zueinander, um das Design zu planen und zu kommunizieren.

**11. Wie sieht ein Enhanced Entity Relationship (EER) Model aus?**

Ein **EER-Modell** erweitert das ERM um komplexere Modellierungskonzepte wie **Generalisierung/Spezialisierung** (Vererbung) und **Kategorien**, um detailliertere Beziehungen abzubilden. Bsp. DBeaver

**12. Was bedeuten die Begriffe: Entität, Attribut und Tupel?**

* **Entität:** Ein Objekt oder Konzept (entspricht einer **Tabelle**).
* **Attribut:** Eine Eigenschaft der Entität (entspricht einer **Spalte**).
* **Tupel:** Ein einzelner Datensatz der Entität (entspricht einer **Zeile**).

**13. Was versteht man unter Datenkonsistenz?**

**Datenkonsistenz** bedeutet, dass die Daten in einer Datenbank **logisch korrekt, widerspruchsfrei und gültig** sind und alle definierten Regeln (Constraints) einhalten.

**14. Was bedeutet Redundanz im Zusammenhang mit einer Datenbank?**

**Redundanz** bedeutet die **mehrfache, unnötige Speicherung derselben Information** an verschiedenen Stellen in der Datenbank, was zu Inkonsistenzen führen kann.

**15. Was versteht man unter Normalform?**

**Normalformen** sind Regeln im Datenbankdesign zur **Minimierung von Redundanz** und zur **Verbesserung der Datenintegrität**, um Anomalien zu vermeiden.

**16. Wie unterscheiden sich die erste, zweite und dritte Normalform?**

* **1NF:** Alle Attribute sind **atomar**, und jede Zeile hat einen eindeutigen Primärschlüssel.
* **2NF:** Ist in 1NF und **keine Nicht-Schlüsselattribute sind partiell von einem Teil des Primärschlüssels abhängig**.
* **3NF:** Ist in 2NF und **keine Nicht-Schlüsselattribute sind transitiv von einem anderen Nicht-Schlüsselattribut abhängig**.

**17. Was versteht man unter Primär- und Fremdschlüssel?**

* **Primärschlüssel:** **Identifiziert jede Zeile einer Tabelle eindeutig** (muss eindeutig und NOT NULL sein).
* **Fremdschlüssel:** Ein Attribut, das auf den **Primärschlüssel einer anderen Tabelle verweist**, um eine Beziehung herzustellen und die referentielle Integrität zu sichern.

**18. Was ist referentielle Integrität?**

**Referentielle Integrität** ist eine Regel, die sicherstellt, dass **Beziehungen zwischen Tabellen gültig und konsistent** sind. Fremdschlüsselwerte müssen entweder auf existierende Primärschlüssel verweisen oder NULL sein.

**19. Was bringt Object-Relational Mapping (ORM)?**

**ORM** überbrückt objektorientierte Programmiersprachen und relationale Datenbanken. Es ermöglicht Entwicklern, **Datenbankdaten als Objekte zu behandeln**, was die Produktivität erhöht und die Datenbankabstraktion fördert, da weniger manuelles SQL nötig ist.

**20. Wie unterscheiden sich die Datentypen Char und Varchar?**

* **CHAR:** **Feste Länge**, füllt mit Leerzeichen auf, belegt immer maximalen Speicherplatz.
* **VARCHAR:** **Variable Länge**, belegt nur benötigten Speicherplatz (plus Längeninfo), keine Auffüllung. VARCHAR ist meist speichereffizienter.

**21. Was ist ein Character Set und wie unterscheiden sich ASCII, LATIN1 und UTF-8?**

Ein **Character Set** ist eine Sammlung von Zeichen mit zugehörigen numerischen Codes.

* **ASCII:** 7-Bit, 128 Zeichen (Englisch).
* **LATIN1 (ISO-8859-1):** 8-Bit, 256 Zeichen (Westeuropa, inkl. Umlaute).
* **UTF-8:** Variable Bytes, **kodiert Unicode** (alle Zeichen der Welt), ist abwärtskompatibel zu ASCII. **Moderner Standard.**

**22. Wie unterscheiden sich die Datentypen Integer und Float?**

* **INTEGER:** Speichert **ganze Zahlen** **exakt**, ohne Nachkommastellen.
* **FLOAT:** Speichert **reelle Zahlen** mit Nachkommastellen als **Annäherung**, was zu Rundungsfehlern führen kann.

**23. Wozu verwendet man den Datentyp Decimal?**

**DECIMAL** wird verwendet, um **exakte Dezimalzahlen** zu speichern, insbesondere für **Finanz- und Währungswerte**, wo Präzision und die Vermeidung von Rundungsfehlern entscheidend sind.

**24. Wie unterscheiden sich die Datentypen Timestamp und Datetime?**

* **TIMESTAMP:** **Zeitzonenbewusst**, wird in UTC gespeichert und bei Abruf konvertiert. Kleinerer Wertebereich. Ideal für created\_at/updated\_at.
* **DATETIME:** **Zeitzonenunabhängig**, speichert den Wert genau wie eingegeben. Größerer Wertebereich. Ideal für fixe, lokale Zeitpunkte (z.B. Geburtstag).

**25. Wie unterscheiden sich 0 und NULL?**

* **0:** Ist ein **definierter numerischer Wert** (die Zahl Null).
* **NULL:** Ist ein **Marker für das Fehlen eines Wertes** ("unbekannt", "nicht vorhanden"). NULL ist kein Wert und Vergleiche benötigen IS NULL / IS NOT NULL.