CloudCommand

Cyber Security

CloudCommand GmbH chr.schumacher@gmx.tm

CloudCommand

Grundlagen der Programmierung

CloudCommand GmbH chr.schumacher@gmx.tm

Einführung in Datenbanken

Definition

Eine Datenbank ist eine organisierte Sammlung von strukturierten Informationen oder Daten, die typischerweise elektronisch in einem Computersystem gespeichert sind.

Merkmale von Datenbanken

Strukturierte Speicherung großer Datenmengen:

- Datenbanken ermöglichen die effiziente und strukturierte Speicherung großer Mengen an Daten.

Beispiele:

- Die Verwaltung von Kundendaten, Produktinformationen oder Transaktionsdaten geht.

Merkmale von Datenbanken

Datenintegrität und Konsistenz:

- Durch die Nutzung von Datenbankmanagementsystemen (DBMS) wird sichergestellt, dass Daten konsistent sind. Mechanismen wie Transaktionen und Constraints helfen dabei, die Datenintegrität zu gewährleisten.

Merkmale von Datenbanken

Leistungsfähige Abfragen (Queries) und Datenbearbeitung:

- Datenbanken bieten leistungsfähige Abfragesprachen wie SQL, die es ermöglichen, Daten schnell und effizient zu durchsuchen, zu sortieren und zu manipulieren. Dies ist entscheidend für die schnelle Verarbeitung und Analyse von Daten.

Merkmale von Datenbanken

Sicherheit und Zugriffskontrolle:

- Datenbanken bieten Sicherheitsmechanismen, um den Zugriff auf sensible Daten zu kontrollieren.
- Es existieren Administratoren, ’normale’ Benutzer, Benutzer mit erweiterten Rechten, etc..
- Benutzerrechte und Authentifizierungsmethoden sorgen dafür, dass nur autorisierte Personen auf bestimmte Daten zugreifen können.

Merkmale von Datenbanken

Backup und Wiederherstellung:

- Datenbanken können mittels standardisierter Prozesse auf einfache Weise repliziert werden.
- Des Weiteren existieren ebenso standardisierte Prozesse mit Wiederherstellungsoptionen, um Daten im Falle eines Systemausfalls oder einer Beschädigung wiederherstellen zu können.
- Im Falle eines Incident trägt dies dazu bei den BIA möglichst gering zu halten.

Merkmale von Datenbanken

Multi-User-Zugriff:

- In der Regel unterstützen Datenbanken den gleichzeitigen Zugriff mehrerer Benutzer auf die Datenbank, ohne dass es zu Konflikten oder Datenverlust kommt.

Merkmale von Datenbanken

Datenmodellierung und Beziehung:

- Datenbanken ermöglichen die Modellierung komplexer Beziehungen zwischen verschiedenen Dateneinheiten.
- Relationale Datenbanken verwenden Verweise in Tabellen, um Beziehungen zwischen Datensätzen darzustellen, was die Organisation und Verknüpfung von Daten erleichtert.
- Außerdem wird so die Redundanz von Daten vermieden.

Merkmale von Datenbanken

Skalierbarkeit und Performance:

- Datenbanken sind darauf ausgelegt, mit wachsender Datenmenge und zunehmenden Benutzeranfragen zu wachsen (skalieren).
- Durch Optimierungstechniken wie Indizierung und Partitionierung kann die Leistung auch bei großen Datenmengen aufrechterhalten werden.

Merkmale von Datenbanken

Berichterstellung und Datenanalyse:

- Datenbanken erleichtern durch die vielfältigen Abfragemöglichkeiten die Erstellung von Berichten und Datenanalysen jeder Art.
- Sie ermöglichen es Unternehmen, fundierte Entscheidungen auf der Grundlage von Datenanalysen und Berichten zu treffen.
- Schlagwort: **Big Data**

Vereinfachte Unterteilung von Datenbanken

- **Relationale Datenbanken**
- **NoSQL Datenbanken**
- **In-Memory Datenbanken**

(Auch andere Unterscheidung gebräuchlich: hierarchisch, netzwerkartig, relational, objektorientiert und dokumentorientiert)

Relationale Datenbanken

Daten werden als Tabellen (Relationen) gespeichert

- Die Tabellen setzen sich aus Spalten und Zeilen zusammen

Verwenden Structured Query Language (SQL)

- SQL ist eine eigene Programmiersprache zum Abrufen von Daten aus einer Datenbank

Populäre Implementationen

- MySQL, PostgreSQL und Microsoft SQL Server

Relationale Datenbanken

Relationale Datenbanken sind nach wie vor in vielen Bereichen vorherrschend, werden aber zunehmend von anderen Datenbanktechnologien abgelöst.

“Andere Datenbanktechnologien” sollen im Folgenden als NoSQL Datenbanken zusammengefasst und kurz vorgestellt werden.

NoSQL Datenbanken

Sammelbegriff

- Fasst alle Datenbanktechnologien, die nicht SQL verwenden zusammen

Unterteilung (nicht vollständig)

- Dokumentorientiert (MongoDB)
- Objektorientiert (Redis)
- Graph Datenbanken (Neo4j)

Struktur der Datenbank und der Zugriff auf Daten hängt von der Art der Technologie ab (z.B. dokumentorientiert)

In-Memory Datenbanken

Daten im RAM

- Die Daten der Datenbank werden in den Arbeitsspeicher geladen (Im Gegensatz zu von der Festplatte geladen)
- **Dadurch:** bessere Performance allerdings wesentlich höhere Kosten

In-Memory Datenbanken werden hauptsächlich für Echtzeitanwendungen verwendet, bei denen eine niedrige Latenz notwendig ist.

Abschluss

- SQL und NoSQL Datenbanken
- In-Memory Datenbanken

Wir werden uns im Folgenden ausschließlich mit herkömmlichen, relationalen SQL Datenbanken beschäftigen.

SQL – Relationale Datenbanken

Definition:

- SQL ist eine **Datenbanksprache** um Abfragen von relationalen Daten zu ermöglichen.
- Unter dem Acronym SQL versteht man üblicherweise “Structured Query Language”

SQL – Relationale Datenbanken

Begrifflichkeiten:

- **Datenbank**
  Unter Datenbank verstehen wir die Ganzheit aller darin gespeicherten Daten, enthalten in einer oder mehreren Tabellen
- **Tabellen**
  Eine Datenbank kann die Daten in verschiedenen Tabellen speichern
- **Zeilen und Spalten**
  Die Tabellen sind wiederum in Zeilen und Spalten unterteilt
- **Primärindex**
  Jeder Datensatz benötigt eine eindeutige Kennung in einer Datenbank, die der Primärindex definiert

SQL – Relationale Datenbanken

Verschiedene Operationen:

- Die Standardoperationen fasst man unter dem Akronym **CRUD** zusammen
  Create: Neue Daten einfügen
  Read: Existierende Daten lesen
  Update: Existierende Daten verändern
  Delete: Daten entfernen
- Für jede Operation existieren entsprechende Schlüsselwörter um diese Aktionen durchzuführen

SQL – Relationale Datenbanken

Verschiedene Operationen:

- Um sogenannte Abfragen (engl. Queries) an die Datenbank zu erstellen verwendet man Schlüsselwörter
  Create: **INSERT**
  Read: **SELECT/FROM**
  Update: **UPDATE**
  Delete: **DELETE**
- Abfragen werden aus einer Kombination von Schlüsselwörtern und Werten erzeugt und mit einem **Semikolon ; abgeschlossen**.
- Mehrere SQL Queries können mit einem Semikolon in einer Befehlszeile verknüpft werden (SQL Injection !)

SQL – Relationale Datenbanken

Daten lesen: Schlüsselwörter SELECT/FROM

- Mit der Konstellation **SELECT what FROM which_table;** können zielgerichtet Daten aus einer Datenbank extrahiert werden

Erfrage alle Informationen der Tabelle myTable
SELECT * FROM myTable;
Erfrage nur den Inhalt der Spalte column
SELECT column FROM myTable;
Erfrage gleichzeitig den Inhalt der Spalte column1 und der Spalte column2
SELECT column1, column2 from myTable;

SQL – Relationale Datenbanken

Daten lesen - Erweiterung: WHERE

- Wir können unser vorheriges Statement erweitern, um zielgerichteter nach bestimmten Werten zu suchen **SELECT what FROM which_table WHERE condition;**

Erfrage alle Namen der Städte mit einer Population größer 1 Mio.
SELECT name FROM cities WHERE (population > 1000000);
Erfrage alle Benutzernamen, die Administratorrechte haben
SELECT name FROM users WHERE (privileges = “Administrator”);

SQL – Relationale Datenbanken

Daten erstellen: INSERT

- Wir können einer Tabelle neue Werte hinzufügen mit dem INSERT Schlüsselwort

Fügt der Tabelle myTable eine neue Reihe mit den Werten Max Muster hinzu
INSERT INTO myTable (col1, col2) VALUES (“Max”, “Muster”);

Fügt der Tabelle employees eine neue Reihe mit Eckdaten von John Doe hinzu
INSERT INTO employees (id, name, position, salary) VALUES (1, “John Doe”, “Manager”, 70000);

SQL - Relationale Datenbanken

Daten aktualisieren: UPDATE

- Wir können bestehende Werte einer Tabelle aktualisieren mit dem UPDATE Schlüsselwort

Setzt den Wert von column1 in Tabelle myTable zu value1

UPDATE myTable SET column1 = value1;

Setzt den Wert salary des Mitarbeiters mit der ID 1 zu 75000

UPDATE employees SET salary = 75000 WHERE id = 1;

SQL – Relationale Datenbanken

Daten löschen: DELETE

- Wir können bestehende Werte einer Tabelle löschen mit dem DELETE Schlüsselwort

Alle Reihen einer Tabelle löschen (Vorsicht walten lassen!)

DELETE FROM myTable;

Löscht den Mitarbeiter mit der “employee_id” 5

DELETE FROM employees WHERE employee_id = 5;

MariaDB unter Kali

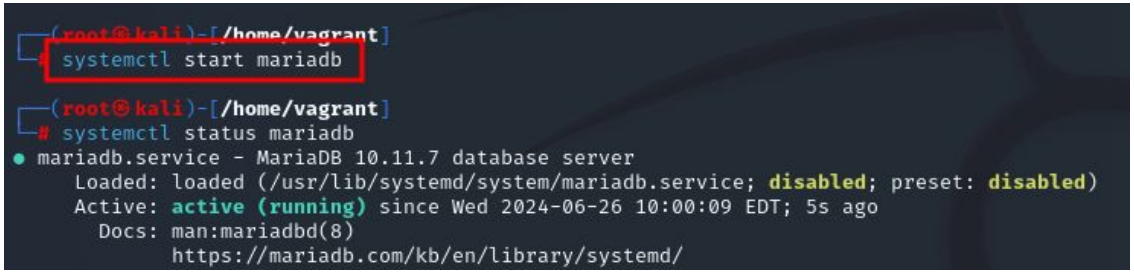
Was genau ist MariaDB?

- MariaDB ist eine Abspaltung der Datenbanksoftware MySQL. Eine relationale SQL Datenbank ist unter Kali Linux standardmäßig installiert.

MariaDB unter Kali

MariaDB Service starten (in (Kali-) Linux)

(root) systemctl start mariadb



MariaDB unter Kali

Datenbank Datei

- **Datenbank.sql Datei** zum automatischen Erstellen einer Datenbank. Es werden die vorher kennengelernten Befehlen verwendet

```
CREATE DATABASE meine_datenbank;
USE meine_datenbank;

CREATE TABLE personen (
    ID INT PRIMARY KEY,
    Vorname VARCHAR(50),
    Nachname VARCHAR(50),
    Email VARCHAR(100)
);

INSERT INTO personen (ID, Vorname, Nachname, Email) VALUES
(1, 'Max', 'Mustermann', 'max.mustermann@example.com'),
(2, 'Anna', 'Müller', 'anna.mueller@example.com'),
(3, 'Tom', 'Schmidt', 'tom.schmidt@example.com'),
(4, 'Sarah', 'Fischer', 'sarah.fischer@example.com'),
(5, 'David', 'Weber', 'david.weber@example.com'),
(6, 'Lena', 'Schneider', 'lena.schneider@example.com'),
(7, 'Markus', 'Schulz', 'markus.schulz@example.com'),
(8, 'Julia', 'Bauer', 'julia.bauer@example.com'),
(9, 'Paul', 'Wagner', 'paul.wagner@example.com'),
(10, 'Lisa', 'Hoffmann', 'lisa.hoffmann@example.com');
```

MariaDB unter Kali

Datenbank Datei

- Eine neue Datenbank meine_datenbank mit den Schlüsselworten **CREATE DATABASE** erstellen und auswählen.

  CREATE DATABASE meine_datenbank;
  USE meine_datenbank;

MariaDB unter Kali

Datenbank Datei

- Eine neue Tabelle personen mit den Schlüsselworten **CREATE TABELE** erstellen und den Spalten ID, Vorname, Nachname und Email erstellen.

```
CREATE TABLE personen (
        ID INT PRIMARY KEY,
        Vorname VARCHAR(50),
        Nachname VARCHAR(50),
        Email VARCHAR(100)
);
```

MariaDB unter Kali

Datenbank Datei

- Der neuen Tabelle mehrere Spalten hinzufügen mit den Schlüsselworten **INSERT INTO**:

```
INSERT INTO personen (ID, Vorname, Nachname, Email) VALUES
(1, 'Max', 'Mustermann', 'max.mustermann@example.com'),
(2, 'Anna', 'Müller', 'anna.mueller@example.com'),
(3, 'Tom', 'Schmidt', 'tom.schmidt@example.com'),
...
```

MariaDB unter Kali

Mit dem MariaDB Server verbinden

- Das startet einen MariaDB Server auf dem MySQL Standard Port 3306
- Verbindung zum Server über Konsolenapplikation: **mariadb -u root**

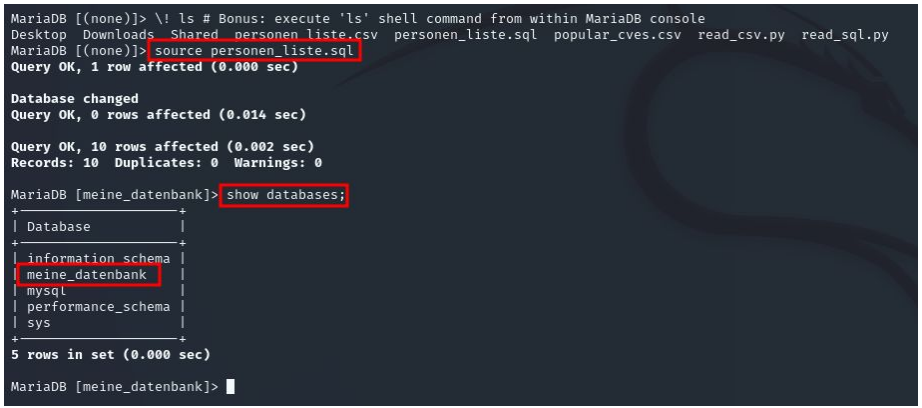
```
┌──(root@kali)-[/home/vagrant]
└─# mariadb -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.7-MariaDB-4 Debian n/a

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

MariaDB unter Kali

Datenbank Datei

- Einlesen der Datenbank.sql Datei mit der MariaDB Konsole (mariadb)

source personen_liste.sql



CloudCommand

CloudCommand GmbH chr.schumacher@gmx.tm