

Übung 1

Aufgabe 1.1

Schauen Sie sich die Darstellung aus der Vorlesung an, die den Unterschied zwischen einer Desktop-Datenbank und einer Client-Server-Datenbank erläutert. Geben Sie für beide Systeme an:

- a) Notwendige Parameter für den Verbindungsaufbau
C/S: IP-Adresse, Port, Benutzername, log. DB, Passwort
Desktop-DB: Pfadname der Datenbank-Datei, Passwort
- b) Vorteile und Nachteile
CS: Zugriff von „überall“ mit Netzwerkverbindung zwischen Client und Server
performanter Mehrbenutzerzugriff, Transaktionssicherheit (ACID), Nachteil:
komplex für Otto-Normal-User, Datenbank-Admin erforderlich
Desktop-DB: Zugriff nur, wenn File-Access gegeben ist (lokal, Cloud, File-Server)
Mehrbenutzerzugriff möglich, aber konkurrierender Schreibzugriff problematisch,
(Schreib-)Sperrung, so dass immer nur ein Nutzer schreibend auf der DB-Datei
arbeiten kann
- c) Typische Einsatzgebiete
CS: Unternehmensanwendungen (Enterprise Resource Planning), Online-
Shops/Web-Anwendungen
Desktop-DB: SOHO, Small Office/Home Office (MS Access/Excel),
Android/Firefox/Safari (sqlite)

MS Access ist ein Beispiel für eine Desktop-Datenbank. Trotzdem gibt es manche IT-Laien, die von einem „Access-Server“ sprechen. Was könnte damit gemeint sein?

Access-„Server“ ist der File-Server, auf dem die Access-DB-Datei liegt. Die eigentliche DB-Funktionalität ist aber immer noch Client-seitig in der Desktop-Datenbank realisiert..

Aufgabe 1.2

Ein Unternehmen möchte ein Customer Relationship Management System (CRM) zur Verwaltung von Kundendaten einführen.

Innendienstler der zentralen Verwaltung greifen von ihren normalen Arbeitsplatzrechnern auf die Daten zu und haben Zugriff auf alle Kundendaten. Außendienstler werden mit Laptops ausgestattet, die auch offline direkt beim Kunden eingesetzt werden sollen, und sie haben nur Zugriff auf die Kundendaten ihres Bezirks.

- a) Skizzieren Sie ein grobes Schaubild der Systemarchitektur. Symbole können Sie frei wählen, z.B. Strichmännchen für User, beschriftete Kästchen für Komponenten, Wolken für Netzwerkgrenzen usw. Machen Sie dabei kenntlich, an welchen Stellen Datenbanksysteme zum Einsatz kommen sollen.
- b) In welchen Bereichen des CRM-Systems würden Sie ein Desktop DBS und in welchen Bereichen ein Client-Server DBS einsetzen?

Auf den Laptops der Außendienstler würde man typischerweise eine Desktop-Datenbank einsetzen, weil diese ohne Konfiguration und Administration betrieben werden kann. Die Innendienstler greifen direkt auf eine zentrale Client-Server

Übung 1

Datenbank zu. Die Außendienstler müssen regelmäßig die Daten in die zentrale C/S-Datenbank überspielen (Achtung: nicht trivial wegen Konflikten!).

Aufgabe 1.3

Konsistenzhaltung der Daten beim unkontrollierten, parallelen Datenzugriff ist ein zentrales Problem in Mehrbenutzersystemen. Ein klassisches Beispiel hierfür ist eine gegenseitige Bank-Überweisung zwischen zwei Konten A und B. Wenn A einen Betrag x zu B überweist und B einen Betrag y zu A, sollte immer gelten, dass $\text{Kontostand}(A) + \text{Kontostand}(B)$ konstant ist, da ansonsten Geld verschwunden oder neu entstanden ist. Konstruieren Sie einen Ablauf zweier gegenseitiger Überweisungen, bei dem die Eigenschaft, dass die Kontostandssumme konstant sein soll, nach dem Abschluss der zwei Überweisungen verletzt ist.

- A liest eigenen Kontostand in Variable a ein.
- A dekrementiert a um x .
- B liest eigenen Kontostand in Variable b ein.
- B dekrementiert b um y
- B liest A's Kontostand in Variable a' ein.
- B inkrementiert a' um y .
- B schreibt a' in A's Kontostand zurück.
- A schreibt a in A's Kontostand zurück \rightarrow B's Änderung an A's Kontostand ist verloren gegangen.

Aufgabe 1.4

Sie entwerfen eine Webanwendung zur Hochschulverwaltung. Mit Ihrem Kollegen diskutieren Sie das Pro und Contra eines Datenbanksystems als Backend für Ihre Anwendung. Ihr Kollege ist skeptisch und würde die Datenverwaltung am liebsten selbst implementieren. Dazu bringt er unter anderem folgende Argumente vor:

- a) Die Installation und Wartung eines Datenbanksystems ist aufwendig, die Erstellung eines eigenen Datenformats dagegen straight-forward und flexibler. Einige Datenformate sind inhärent unflexibel, da es keinen standardisierten Pfad zur Erweiterung, Verteilung, Recovery etc. gibt. Man bedenke beispielsweise, dass allein viele Dateisysteme keine Dateien über einer fixen Größe, bei FAT32 beispielsweise traditionell 2GB erlauben. Hinzu kommt, dass durch die manuelle Erstellung von Dateiformaten keine standardisierten Datentypen verwendet werden und das gesamte "Schema" der Datenspeicherung leicht uneinheitlich wird. Im Vergleich dazu ist der Aufwand zu Erstinstallation einer Datenbank vernachlässigbar, insbesondere, da heutzutage nicht zwangsläufig ein komplexes Produkt wie IBM DB2 o.ä. verwendet werden muss, sondern man für kleine Eigenentwicklungen auch durchaus eine eingebettete Datenbank wie etwa sqlite verwendet werden kann.
- b) Mehrbenutzersynchronisation würde für die Anwendung sowieso nicht benötigt und lässt sich im Zweifel einfach nachrüsten. Mehrbenutzersynchronisation ist inhärent notwendig, wenn sie ein System entwickeln, auf das mehrere Personen zugreifen. Insbesondere ist diese eine der

Übung 1

Eigenschaften, die nicht einfach "nachgepatched" werden kann, sondern sehr tief in eine Datenverwaltungsschicht integriert werden muss. Datenbanksysteme erlauben es, ohne über Nebenläufigkeit nachdenken zu müssen auf Daten zuzugreifen und das "erwartete" Ergebnis zu erhalten. Siehe dazu das ACID Paradigma, was i.A. von DBMS erfüllt wird.

- c) Es ist unsinnig, dass jeder Entwickler zuerst eine eigene Anfragesprache (SQL) lernen muss, nur um Daten aus der Datenbank zu extrahieren.
Ein eigenes Datenformat und dessen API (wenn es denn zumindest eine API für den Zugriff gibt) muss auch gelernt werden, dafür ist SQL standardisiert und kann auch beim Wechsel des DBMS weiterverwendet werden.
- d) Redundanz ist hilfreich, wieso sollte man auf sie verzichten?
Redundanz sorgt auch für Anomalien, etwa beim Updaten von Daten.

Finden Sie stichhaltige Antworten auf die von Ihrem Kollegen in den Raum gestellten Äußerungen.

Aufgabe 1.5

Bei der Arbeit mit Datenbanken/Datenbanksystemen gibt es eine Reihe von unterschiedlichen Berufsbildern und Rollen. Identifizieren Sie diese Berufsbilder/Rollen und beschreiben Sie deren typische Aufgaben und Verantwortlichkeiten.

Siehe Folien DBS-WS2223-Kap1-Addendum1-Berufsbilder-m.pdf

Aufgabe 1.6 (Hausaufgabe)

Finden Sie über eine Internetsuche heraus, welches Datenmodell den folgenden Datenbanksystemen zugrunde liegt: IMS, OrientDB, LDAP, Oracle, Neo4j, SQLite.

- Oracle, SQLite: Relational
- OrientDB, Neo4j: Netzwerk/Graph
- IMS, LDAP: Hierarchisch