Informatik Abitur 2018

Netzwerke

P2P: symmetrische Kommunikation: jeder Teilnehmer ist gleichberechtigt und kann mit jedem kommunizieren

Client-Server: Server (Programm) bietet Dienst an, Client nutzt den Dienst (z.B. Web-Server)

Protokoll: notwendige Regeln um einen kontrollieren und eindeutigen Verbindungsaufbau, Datenaustausch und Verbindungsabbau zwischen zwei Rechnern zu gewährleisten

TCP / IP - Schichtenmodell: Ein einzelnes Programm zur Übertragung von Daten über das Netz wäre zu komplex. Deshalb wurden die Aufgaben in Teilbereiche unterteilt, die einzeln programmiert und aufeinander gesetzt werden. (OSI bzw. TCP / IP)

4. Anwendungsschicht	enthält Anwendungen und Prozesse, die auf das Netzwerk zugreifen; benutzt Protokolle wie z.B. FTP, SMTP oder HTTP
3. Transportschicht	- übernimmt die Adressierung des Prozesses auf einem Rechner - vergibt die Port-Nummern (TCP-Protokoll / UDP-Protokoll) → Datenpakete gelangen zur richtigen Anwendung
2. Internetschicht	IP-Protokoll: verwaltet u.a. die Adressen der Rechner und ermöglicht so die Übertragung eines Datenpakets durch das Netz zu einem bestimmten Zielrechner
1. Netzzugangsschicht	Protokolle wissen wie die spezielle Hardware angesprochen werden muss und können Datenpakete zwischen 2 Rechnern austauschen (z.B. Ethernet)

TCP-Protokoll:
Sender-Seite: Übergibt die Daten eines Anwendungsprogramms in Datenpaketen an das IP-Protokoll.
Empfänger-Seite: Setzt die Daten aus den Paketen wieder richtig zusammen und sorgt dafür, dass verloren gegangene Pakete erneut gesendet werden

UDP-Protokoll: Hat weniger Funktionalität als TCP. Anwendungsprogramm muss Daten selber in Datenpakete aufteilen und sich um verloren gegangene Pakete kümmern.

Wenn das Anwendungsprogramm Daten übertragen möchte, werden nacheinander die einzelnen Schichten aufgerufen und jede Schicht packt zu dem Datenpaket eigene Daten hinzu, die sie für ihre Verwaltungsarbeit benötigt.

IP-Adresse: Adresse eines Computers im Internet. Besteht aus einer einmaligen 32-Bit Zahl

und ist in vier Parts durch Punkte unterteilt.
Die ersten Bits kennzeichnen die Größe des Netzes (A, B oder C), die weiteren identifizieren das Netz und die restlichen den entsprechenden Computer. (Bsp.: 193.246.253.248)

Host- und Domainnamen: Host: Name des Rechners; Domain: der Rest; zsm.: Fully Qualified Domain Name (FQDN)
Um FQDN's und IP-Adressen hin und her zu übersetzen braucht man einen Domain Name Service (DNS). Rechner der diesen Dienst anbietet: DNS-Server. Jedoch ist die Zuordnung nicht eindeutig, da auf einem physikalischen Server mehrere Dienste, oder ein Dienst auf mehreren physikalischen Servern angeboten werden kann (Ausfallsicherheit etc.).

Port-Nummern: IP-Adresse des Rechners reicht nicht aus, da häufig mehrere Server auf einem Rechner laufen. Das TCP-Protokoll vergibt Port-Nummern (0 - 65535) um Anwendungen auf einem Rechner zu adressieren. Well Known Ports: reservierte Ports für bekannte Internet-Dienste

Loopback-Adresse: localhost bzw. 127.0.0.1: Datenpakete treffen ohne ins Netzwerk zu gelangen beim sendenden Computer wieder ein. Wird für Programmtests bzw. zur Erreichung lokaler Programme über TCP/IP genutzt (z.B. lokale Datenbank).

Datenbanken

Relationale Datenbanksysteme: besitzen die Fähigkeit Tabellen zu verknüpfen und auf diese Weise komplexere Informationen zu generieren.
Nahezu alle bekannten Datenbanksysteme arbeiten heute nach dem relationalen Modell.

Die Software innerhalb eines Datenbanksystems, welche die Eingabe, Verwaltung und Ausgabe von Daten ermöglicht, nennt man Datenbankmanagementsystem (**DBMS**). Dieses DBMS, das sich unter Umständen auf einem ganz anderen Rechner befindet, wird als *Back End* bezeichnet.

Die Software, die der Benutzer zur Abfrage und Veränderung der Datenbank benutzt, nennt man *Front End* und besitzt auch die Bedienungsoberfläche, die der Benutzer sieht.

Um mit einem DBMS arbeiten zu können, muss ein Front End immer eine Verbindung zu „seinem" Back End aufbauen.

Alle gängigen Datenbank-Front-Ends benutzen die Sprache **SQL** für die Abfrage und Manipulation ihres Back Ends. Das Front End sendet an das Back End SQL-Befehle zur Steuerung des DBMS. SQL ist eine sogenannte deskriptive Sprache: Mit SQL legt man nur fest, was das DBMS tun soll. Wie es das tut, bleibt dem DBMS überlassen.

Referentielle Integrität

Die Referentielle Integrität - auch Beziehungsintegrität genannt - besagt, dass Attributwerte von einem Fremdschlüssel auch als Attributwert eines Primärschlüssels vorhanden sein müssen

- Durch die referentielle Integrität wird Dateninkonsistenz vermieden
- Es muss erst ein Datensatz in der Tabelle mit dem Primärschlüssel hinzugefügt werden, bevor dieser als Attributwert als Fremdschlüssel in einer anderen Tabelle eingetragen werden kann
- Um einen Datensatz aus einer Tabelle mit einem Primärschlüssel zu löschen muss erst alle Datensätze aus allen Tabellen mit dem entsprechenden Fremdschlüssel entfernt werden

Normalisierung

1. Normalenform: Atomar & keine Wiederholungsgruppen

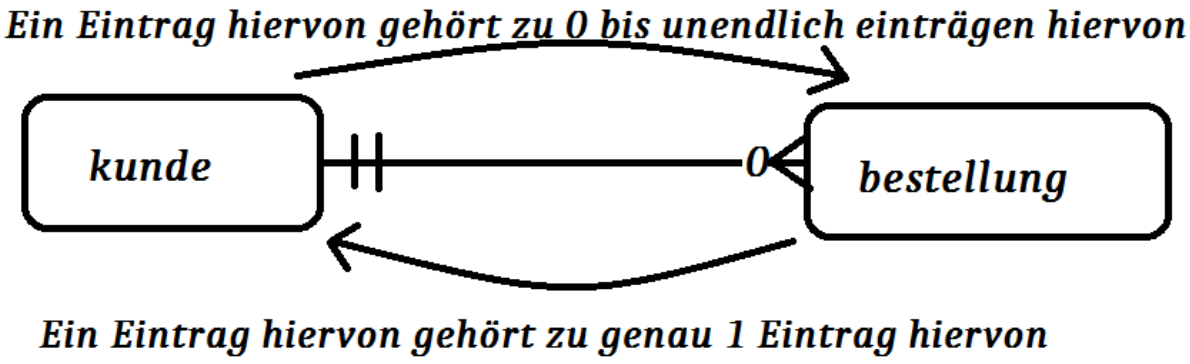
2. Normalenform: Alle Nicht-Schlüssel-Attribute sind voll funktional vom (gesamten) Primärschlüssel abhängig

3. Normalenform: Alle Nicht-Schlüssel-Attribute sind voneinander unabhängig

ER-Diagramme

Kardinalitäten

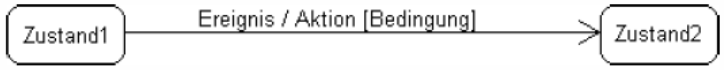
Merkspruch: Ein Eintrag hiervon (ausgehende Tabelle) kann zu … Einträgen hiervon



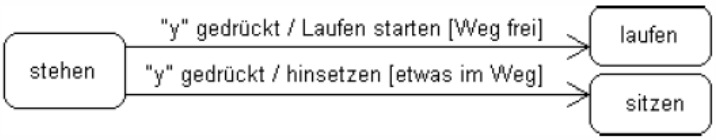
(Tabelle auf welcher Seite das entsprechende Ergebnis des Spruches eingetragen wird) gehören

UML-Zustandsdiagramme

Schema:



Beispiel:



Beschriftung der Zustandsübergänge

Ereignis	Das System wird von außen angestoßen seinen Zustand zu ändern (in der Regel durch den Benutzer). Beispiele: Tastendruck, Klick auf einen Button, Signal von der Systemuhr
/Aktion	Tätigkeit, die das System während des Zustandsübergangs ausführt (z.B. Aufruf einer Methode).
[Bedingung]	Das System entscheidet aufgrund von eigenen Informationen (z.B. Variablenwerten), in welchen Zustand es übergeht. Eine Bedingung ist nur dann sinnvoll, wenn verschiedene Zustände zur Auswahl stehen.

- Alles was als Programmierer einen größeren Aufwand darstellt wird entsprechend als Zustand dargestellt
    - das vereinfacht die Umsetzung des UML-Diagramms in Code