

Datenorganisation/Datenbanken - Übungsaufgaben - Blatt 3

Aufgabe 1:

- Welches Laufzeitverhalten O kann ein Sortieralgorithmus im günstigsten Fall erreichen? - Welche Voraussetzung muss die zu sortierende Liste dann erfüllen? Nennen Sie einen Algorithmus, der diesen Fall ausnutzt.
- Mit welchem Laufzeitverhalten (Anzahl an Vergleichen) kann ein Sortierverfahren im schlechtesten Fall auskommen? Wie bezeichnet man ein Verfahren, das dieses Laufzeitverhalten garantiert? Nennen Sie einen Algorithmus, dem man diese Bezeichnung geben kann.
- Betrachten Sie folgendes Array mit 8 Elementen:
6 12 44 18 42 55 67 94

Sortieren Sie das Array mit dem Shaker-Sort-Verfahren. Wieviele Vergleiche waren bis zur fertig sortierten Liste nötig?

Wieviele Vergleiche wären mit dem Verfahren des „direkten Auswählens“ nötig (welches Laufzeitverhalten)? Warum bietet das Shaker-Sort-Verfahren hier die schnellere Methode?

Aufgabe 2:

Warum verwendet man AVL-Bäume nicht zur Datenspeicherung von sortierten Elementen auf externen Speichermedien (z.B. Magnetplatten)? Welche Datenstruktur wird stattdessen häufig verwendet? Betrachten Sie dazu den Fall von 1.000.000.000 Elementen.

Aufgabe 3:

Führen Sie die ER-Modellierung (in Min-Max-Notation) für ein Wartungs-Informationssystem (WIS) eines Herstellers von Standardsoftware durch. Das System soll folgende Aufgaben erfüllen:

Auskünfte im interaktiven Betrieb (z.B. mit SQL-Abfragen), wobei folgende Punkte gefragt sind:

- Information über alle hergestellten Softwareprodukte und Releases dieser Produkte.
- Information über alle Kundenregistrierungen, wobei über das Release und das Produkt Auskunft gegeben werden soll, das der Kunde erworben hat.
- Information darüber, welche Wartungsgruppe ein bestimmtes Produkt betreut, welche Mitarbeiter zu dieser Wartungsgruppe gehören und welche Funktion sie innerhalb der Gruppe wahrnehmen (z.B. Leiter, Systemanalytiker, Programmierer, Projektverwalter, usw.). Die Funktion eines Mitarbeiters kann in den verschiedenen Gruppen, in denen er mitarbeitet, unterschiedlich sein.
- Information über die Problemmeldungen, die von Kunden für die von ihnen erworbenen Produkte eingegangen sind, sowie über die Wartungsmaßnahmen, die ergriffen wurden, um diese Probleme zu beseitigen.

Außerdem sollen folgende **Standardausgaben** (Listen) erzeugt werden:

- Tägliche Problem-Meldungslisten, welche alle Problemmeldungen des Tages für ein bestimmtes Produkt beinhalten, wobei bei jeder Problemmeldung auch das Release, der Name des Kunden, der Ansprechpartner sowie dessen Telefonnummer angegeben sein sollen.

- Monatliche Problem-Erledigungs-Berichte (PEB). Darin werden für eine Gruppe alle Wartungsmaßnahmen aufgeführt, welche in dem betreffenden Monat abgeschlossen wurden, mit Angabe des Produkts, des Abschlußdatums, des Ergebnisses (Kurzform) und des Mitarbeiters oder der Mitarbeiter, welche die Wartungsmaßnahme durchgeführt haben. Die Problemmeldung, welche den Anstoß zur Wartungsmaßnahme gab, wird ebenfalls genannt (mit der Problemmeldungsnr und dem Eingangsdatum).

Leiten Sie aus dem ER-Modell geeignete Relationen ab und formulieren Sie die SQL-Abfragen, die unter dem Punkt „Auskünfte im interaktiven Betrieb“ aufgelistet sind.

Aufgabe 4:

Stellen Sie für die folgende Tabelle *Prüfungsgeschehen* die funktionalen Abhängigkeiten zusammen und überführen Sie die Relation in die BCNF (Boyce-Codd-Normalform).

Prüfungs-nr	Fach	Prüfer	Matrikel-nr	Name	Ge-burtstag	Fachbe-reichsnr	Fachbe-reichsname	Dekan	Note
3	SWE	Meier	35467	Simon	050274	12	Informatik	Bauer	2
3	SWE	Meier	65323	Kunz	091176	12	Informatik	Bauer	3
4	OOP	Schulz	65323	Kunz	091176	12	Informatik	Bauer	1
4	OOP	Schulz	88347	Schöller	221276	12	Informatik	Bauer	3
5	OOP	Seifert	35467	Simon	050274	12	Informatik	Bauer	1
6	BRE	Richter	22243	Biller	300770	08	Wirtschaft	Wild	4
6	BRE	Richter	98564	Kinder	171075	09	Recht	Fiesler	4

Aufgabe 5:

- Fügen Sie in einen zunächst leeren binären Suchbaum der Reihe nach ein:
T, V, N, S, O, M, F, G, U, W, Z
- Welcher maximale Aufwand O kann entstehen, wenn im fertig aufgebauten Baum von a) ein Element gesucht wird?
- Welcher durchschnittliche Aufwand O ergäbe sich für das Suchen in einem ausgeglichenen Baum, wenn die gleiche Anzahl Elemente wie bei a) eingetragen sind?
- Angenommen, die ersten 3 Ebenen (incl. Wurzel) des Baumes von a) beschreiben einen Heap-Baum (die Elemente der weiteren Ebenen sollen nicht betrachtet werden). Zeigen Sie, wie das Sortieren beim Heap-Sort-Verfahren abläuft!

Datenorganisation/Datenbanken - Übungsaufgaben-Lösungen

zu Blatt 1 ~~2~~
① + ②

Blatt 1 - Aufgabe 1:

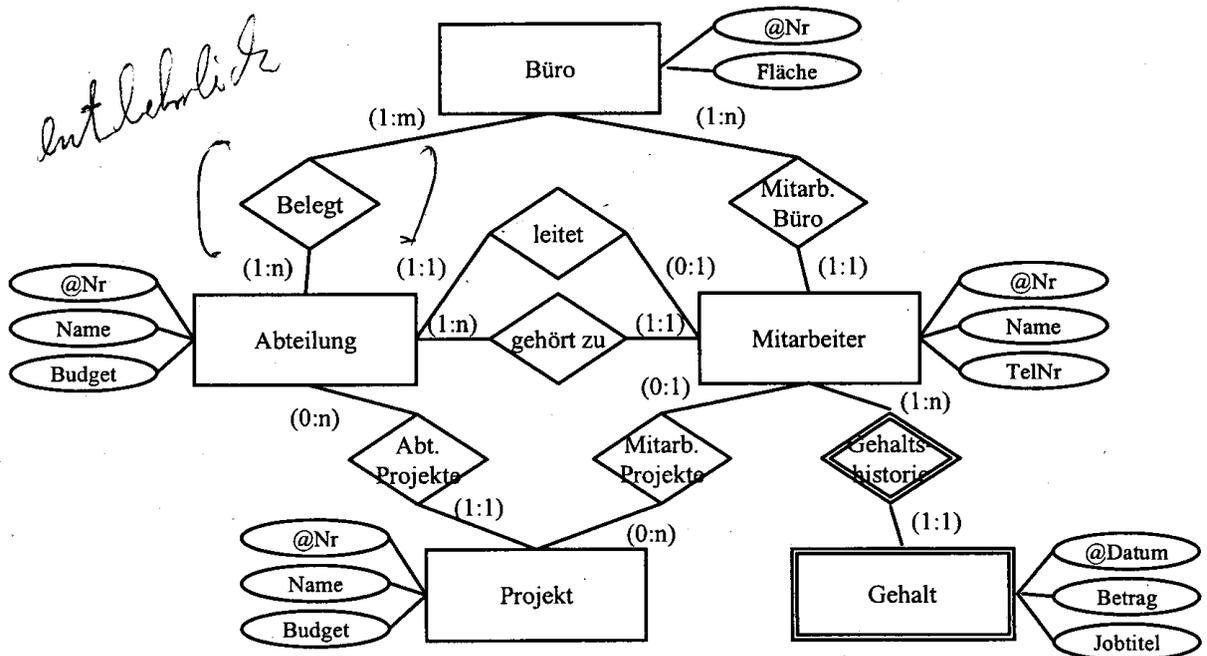
- Hauptvorteile des Shell-Sort-Verfahrens gegenüber dem Bubble-Sort-Verfahren:
 - Sortierung kommt auch im schlechtesten Fall mit $O(N * \log^2(N))$ aus
 Hauptnachteile des Shell-Sort-Verfahrens gegenüber dem Bubble-Sort-Verfahren:
 - Stabilität ist nicht gewährleistet
 - Vorsortierung kann nicht ausgenutzt werden
- Eine Sortiermethode heißt **stabil**, wenn die relative Ordnung der Elemente mit gleichen Schlüsseln beim Sortieren unverändert bleibt, z.B.
 - Direktes Einfügen
 - Direktes Auswählen
 - Bubble-Sort
 - Merge-Sort

Blatt 1 - Aufgabe 2:

In 1. Normalform:

Abteilung = @Abteilungsnr. + Name + Budget + Abteilungsleiter_Mitarbeiternr. + @Mitarbeiternr. + Mitarb._Name + Büronr. + Telefonnr. + Projektnr. + @Jobtitel + @Datum + @Gehalt + @Projektnr. + Projekt_Name + Projekt_Budget + @Raumnr. + Fläche

ERM (bereits in 3NF):



In 3 NF:

Abteilung = @Abteilungsnr. + Name + Budget + ~~<Abteilungsleiter>Nr~~ MA
Fremd (<Abteilungsleiter>Nr) Ref Mitarbeiter

Projekt = @Nr + Name + Budget + ~~<Abteilung>Nr~~ MA
Fremd (<Abteilung>Nr) Ref Abteilung

Mitarbeiter = @Nr + Name + Telefonnr + <Abteilung>Nr + <Büro>Nr + <Projekt>Nr

Fremd (<Abteilung>Nr) Ref Abteilung

Fremd (<Büro>Nr) Ref Büro

Fremd (<Projekt>Nr) Ref Projekt

Gehalt = @Datum + ~~Betrag~~ + Jobtitel + <Mitarbeiter>Nr

Fremd (<Mitarbeiter>Nr) Ref Mitarbeiter

Büro = @Nr + Fläche

Raumbelegung = @<Abteilung>Nr + @<Büro>Nr

Fremd (<Abteilung>Nr) Ref Abteilung

Fremd (<Büro>Nr) Ref Büro

Alternative: (1:n)-Beziehung „Abteilungsprojekte“ abbilden auf eigene Relation:

Projekt = @Nr + Name + Budget + <Mitarbeiter>Nr

Fremd (<Mitarbeiter>Nr) Ref Mitarbeiter

Abteilungsprojekte = @<Abteilung>Nr + @<Projekt>Nr

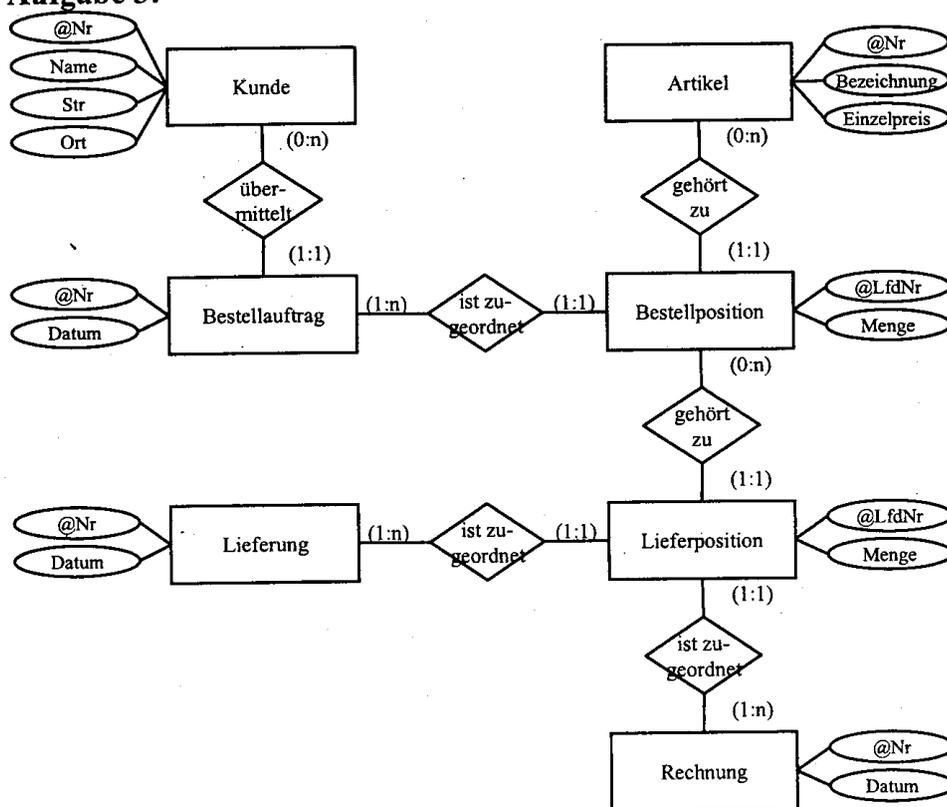
Fremd (<Abteilung>Nr) Ref Abteilung

Fremd (<Projekt>Nr) Ref Projekt

Bewertung:

Beschleunigt unter Umständen den Zugriff von der Abteilung auf die Projekte, verlangt u.U. den Zugriff vom Projekt zur Abteilung.

Blatt 1 - Aufgabe 3:



Blatt 1 - Aufgabe 4:

Kunde = @Nr + Name + Straße + Ort

Artikel = @Nr + Bezeichnung + Einzelpreis
 Bestellauftrag = @Nr + Datum + <Kunde>Nr
 Bestellposition = @LfdNr + @<Bestellauftrag>Nr + <Artikel>Nr + Menge
 Lieferung = @Nr + Datum
 Lieferposition = @LfdNr + @<Lieferung>Nr + <Bestellposition>Nr + <Rechnung>Nr + Menge
 Rechnung = @Nr + Datum

Blatt 2 - Aufgabe 1:

ER-Modell auf Folie (nach Date)

Anforderungen in SQL:

- 1) Alle Angaben zu den in London durchgeführten Projekten

```
SELECT * FROM Project WHERE city = 'London'
```
- 2) Nummer und Name aller Lieferanten, die das Projekt mit der Nummer J1 beliefern.

```
SELECT Sno, Name
FROM Supplier, SPP
WHERE Supplier.SNo = SPP.SNo AND JNo = 'J1'
```
- 3) Alle Lieferungen mit einer Stückzahl zwischen 300 und 750

```
SELECT * FROM SPP WHERE Qty BETWEEN 300 AND 750
```
- 4) Teilenummern von Teilen, die von einem in London beheimateten Lieferanten an ein Projekt in London geliefert werden

```
SELECT Pno FROM Supplier, SPP, Project
WHERE Supplier.city = 'London' AND Project.city = 'London'
AND Supplier.Sno = SPP.Sno AND SPP.Jno = Project.Jno
```
- 5) Für alle Teile, die an ein Projekt geliefert werden, die Teilennr, Projektnr und die Anzahl der gelieferten Teile

```
SELECT Pno, Jno, SUM(Qty) FROM SPP GROUP BY Pno, Jno
```
- 6) Nummern der Teile, von denen an ein Projekt durchschnittlich mehr als 320 Stück geliefert wurden

```
SELECT Pno, Jno FROM SPP
GROUP BY Pno, Jno HAVING AVG(Qty) > 320
```
- 7) Farben der Teile, die von Lieferant mit Nummer S1 geliefert werden

```
SELECT color, Pno FROM Part, SPP
WHERE (Part.Pno = SPP.Pno) AND (SPP.Sno = 'S1')
GROUP BY color, Pno
```
- 8) Nummern von Teilen, die an irgendein Projekt in London geliefert werden

```
SELECT Pno FROM SPP
WHERE Jno IN (SELECT Jno FROM Project WHERE city = 'London')
```

*Select Pno From SPP, Project WHERE SPP.Jno = Project.Jno
and Project.City = "London"*

- 9) Nummern derjenigen Lieferanten, die ein Projekt mit Teil P1 in einer Stückzahl beliefern, die größer ist als die durchschnittliche Stückzahl der Lieferungen von Teil P1
- ```
SELECT Sno FROM SPP
WHERE Pno = 'P1' AND Qty > (SELECT AVG(Qty) FROM SPP
WHERE Pno = 'P1')
```
- 10) Liste aller Städte, die zu wenigstens einem Lieferanten, einem Teil oder einem Projekt gehören
- ```
SELECT city FROM Supplier
UNION SELECT city FROM Part
UNION SELECT city FROM Project
```
- 11) Ändere die Farbe aller roten Teile auf orange
- ```
UPDATE Part SET color = 'orange' WHERE color = 'rot'
```
- 12) Lösche alle Projekte, für die es keine Lieferungen gibt
- ```
DELETE FROM Project WHERE Jno NOT IN
(SELECT Jno FROM SPP)
```
- 13) Erstelle eine Tabelle, die eine Liste der Nummern der Projekte enthält, die entweder in London durchgeführt werden oder durch einen Londoner Lieferanten beliefert werden
- ```
CREATE TABLE Proj_London (Jno CHAR(5));
INSERT INTO Proj_London
(SELECT Jno FROM Project WHERE city = 'London')
UNION (SELECT Jno FROM SPP WHERE Sno IN
(SELECT Sno FROM Supplier WHERE city = 'London'))
```

### Blatt 2 - Aufgabe 2:

- 1) Da in der Regel mehrere Programme auf eine Datenbank zugreifen, ist es sinnvoll, die Verifizierung von Integritätsbedingungen nicht in jeder einzelnen Anwendung zu realisieren, sondern zentral im Datenbanksystem zu implementieren. Dadurch wird die Anwendungsentwicklung entlastet, da sie sich um die Programmierung der Integritätsbedingungen nicht kümmern muß und gleichzeitig die Qualität der Software erhöht, weil die Fehlergefahr wegfällt, die durch mehrfache Implementierung der gleichen Integritätsbedingungen entsteht.
- 2) Das Ansi-Sparc-Modell beschreibt das grundlegende Modell einer Datenbank-Architektur als interne, konzeptionelle und externe Ebene (Ebenen lt. Script beschreiben).  
SQL stellt im wesentlichen Befehle für die konzeptionelle Ebene (z.B. create Table) und für die externe Ebene (z.B. create View) zur Verfügung. Es gibt aber auch Befehle für die interne Ebene (z.B. create Index)