ABITUR-TRAINING

Gymnasiale Oberstufe

Informatik

STARK

Inhalt

Autor: Patrick Gerth

Vorwort

Liebe Schülerin, lieber Schüler,

das vorliegende Buch unterstützt Sie dabei, sich erfolgreich auf die **schriftliche Abiturprüfung im Fach Informatik** vorzubereiten. Das besondere Konzept des Bands hilft Ihnen, Ihr im Unterricht erworbenes **Wissen abzusichern**, und ermöglicht Ihnen die **Anwendung** Ihrer Kenntnisse an ausgewählten Prüfungsaufgaben.

- Alle **abiturrelevanten Prüfungsinhalte** finden Sie systematisch und lehrplanorientiert gegliedert und in kompakter, aber anschaulicher Form zusammengefasst. Insbesondere anhand zahlreicher anschaulicher **Beispiele** können Sie sich den Prüfungsstoff noch einmal vergegenwärtigen und einprägen.
- Zur besseren Verständlichkeit sind im Buch zahlreiche **Beispiel-Programmcodes** in Java, SQL und Pseudocode abgedruckt (siehe Hinweise auf der nächsten Seite). Sie zeigen Ihnen, wie sich die jeweiligen Konzepte in echten Programmen umsetzen lassen, und unterstützen Sie so beim Transfer von den Inhalten zur praktischen Anwendung.
- Zur sofortigen, aufgabenbezogenen Anwendung Ihres Wissens finden Sie nach jedem Kapitel eine Reihe von **Übungsaufgaben**, die sich direkt auf die Inhalte des Theorieteils beziehen. So fällt es leicht, sich systematisch mit einzelnen Themengebieten zu befassen.
- Anhand der **komplexen Prüfungsaufgaben**, in denen mehrere zumeist anwendungsbezogene Aufgaben zu Aufgabenblöcken zusammengefasst sind, können Sie Ihre Fähigkeiten testen. Da es sich dabei um repräsentativ ausgewählte Teile **originaler Abituraufgaben** aus verschiedenen Bundesländern handelt, bekommen Sie ein Gefühl für die Anforderungen und die herrschende Aufgabenkultur. Verweise auf die thematisch passenden Kapitel am Anfang jedes Aufgabenblocks ermöglichen Ihnen, Aufgaben passend zu Ihrer momentanen Lernphase auszuwählen.
- Zu vielen Inhaltsabschnitten, Aufgabenstellungen und Lösungsvorschlägen stehen Ihnen **Tipps** zur Verfügung. Sie sind vor allem darauf ausgelegt, Sie bei einer formal korrekten und strukturierten Bearbeitung der Aufgaben bzw. Auseinandersetzung mit Themen zu unterstützen sowie Ihre Aufmerksamkeit auf das Wesentliche zu lenken. Sie liefern aber auch inhaltliche Zusatzinformationen oder zeigen alternative Antwortmöglichkeiten auf.

- **Ausführliche Lösungsvorschläge** zu allen Aufgaben dienen der Selbstkontrolle und machen Sie mit den Anforderungen an eine adäquate Lösung vertraut.

Viel Erfolg bei der Arbeit mit diesem Buch und bei der Abiturprüfung!

Ihr STARK Verlag

6 Kryptologie

6.1 Grundlagen

Die Kryptologie ist die Lehre der Verschlüsselungen und Entschlüsselungen von Informationen. Sie beschäftigt sich also sowohl mit **kryptografischen Verfahren**, mit deren Hilfe Daten verschlüsselt und so vor Dritten geheim gehalten werden, als auch mit der **Kryptoanalyse**, der Informationsgewinnung aus verschlüsselten Daten.

Folgende Grundbegriffe sind wichtig:

- **Klartext**:
  originale Nachricht bzw. Originaldaten
- **Kryptotext** (oder **Geheimtext**, **Chiffretext**):
  verschlüsselte Nachricht bzw. Daten
- **Verschlüsseln/Entschlüsseln** (oder **Chiffrieren/Dechiffrieren**):
  Überführen des Klartextes in den Kryptotext und andersherum
- **Schlüssel**:
  signifikante Größe beim Verfahren des Verschlüsselns/Entschlüsselns

TIPP

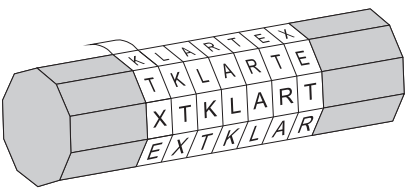
Schlüssel können aus einzelnen Ziffern, Wörtern, langen Zeichenketten etc. bestehen.

6.2 Historische Verschlüsselungsverfahren

Skytale

Die Skytale ist ein militärisches Verschlüsselungsverfahren, das bereits vor mehr als 2 500 von den Spartanern eingesetzt wurde.

Es besteht aus einem Stab, um den ein Band gewickelt wird. Dieses wird der Stablänge nach von oben nach unten (mit dem Klartext) beschrieben. Danach wird das Band abgewickelt. In abgewickeltem Zustand ist eine scheinbar willkürliche Buchstabenfolge zu sehen. Um die Nachricht zu entschlüsseln, muss das Band auf einen Stab identischen Durchmessers gewickelt werden. Dieser Durchmesser ist somit der Schlüssel. Analog zum Aufwickeln des Bandes kann man die Buchstabenfolge in mehreren Spalten einer Tabelle schreiben. Die Anzahl der Buchstaben, die pro Spalte untereinander in eine Spalte geschrieben werden, entspricht dem Stabdurchmesser. Liegt diese Information nicht vor, kann die Spaltenlänge durch Ausprobieren (Bruteforce-Verfahren) herausgefunden werden.



K	L	A	R	T	E	X
T	K	L	A	R	T	E
X	T	K	L	A	R	T
E	X	T	K	L	A	R

Cäsar-Verschlüsselung

Die Cäsar-Verschlüsselung (benannt nach einem Anhänger dieser Technik – Gaius Julius Cäsar) ist ein Verfahren, bei dem jeder Buchstabe im Alphabet auf einen anderen Buchstaben verschoben (transponiert) wird (z. B. B auf E). Die Weite der Verschiebung ist bei allen Buchstaben gleich, wird im Vorfeld festgelegt und bildet den Schlüssel.
Historisch wurde zur Verschlüsselung und Entschlüsselung eine Konstruktion aus zwei Scheiben benutzt, die gegeneinander verschoben wurden (siehe Abbildung).

BEISPIEL

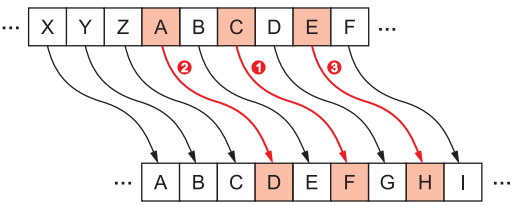
Aus dem Satz

CAESAR KENNT DEN TRICK

wird mit einer Verschiebung von 3:

FDHVDU NHQQW GHQ WULFN

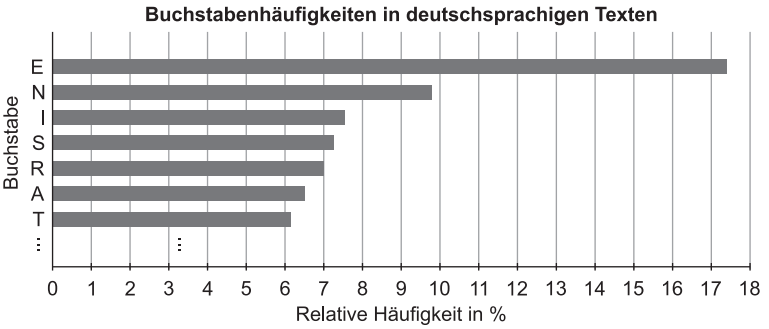
Die ersten drei Verschlüsselungsschritte lassen sich auch wie folgt visualisieren:



Die Cäsar-Verschlüsselung gilt als eines der bekanntesten Beispiele für **monoalphabetische Substitutionen**, einer Gruppe von Verfahren, bei denen die Zeichen des Klartextes nur anhand eines einzigen Verschlüsselungsalphabets durch andere Zeichen ersetzt werden.
Die entscheidende Schwäche dieser Verfahren ist, dass Häufigkeitsverteilungen und andere strukturelle Grundmerkmale des Klartextes gewahrt (bzw. nur verschoben) werden. Dieser Umstand liefert einen Ansatzpunkt für Entschlüsselungsverfahren.

BEISPIEL

In deutschsprachigen Texten kommen die Buchstaben des Alphabets in bestimmten Häufigkeiten vor: das E ist der mit Abstand häufigste Buchstabe, gefolgt von den Buchstaben N und I. In der Tabelle sind die relativen Häufigkeiten der sieben häufigsten Buchstaben gezeigt.



TIPP

Eine heuristische Analyse funktioniert umso besser, je länger der untersuchte Kryptotext ist.

Unterzieht man nun einen durch monoalphabetische Substitution verschlüsselten Kryptotext einer **heuristischen Analyse**, zählt man also, wie oft jeder Buchstabe vorkommt, so lässt sich durch Vergleich mit der bekannten Häufigkeitsverteilung in deutschen Texten der Schlüssel des Verfahrens ermitteln – im Fall der Cäsar-Verschlüsselung die Anzahl der Stellen, um die verschoben wurde.

Vigenère-Verfahren

Beim Vigenère-Verfahren, das im 16. Jahrhundert entwickelt wurde, wird zu Beginn ein Schlüsselwort definiert. Dieses wird Buchstabe für Buchstabe wiederholt unter die Buchstaben des Klartextes geschrieben. Nun gibt der jeweilige Buchstabe des Schlüssels an, um wie viel der Buchstabe des Klartextes verschoben werden soll. Der dadurch entstehende Geheimtextbuchstabe lässt sich über ein **Vigenère-Quadrat** (siehe nächste Seite) ablesen. Da hier nicht nur ein Verschlüsselungsalphabet verwendet wird, sondern mehrere, handelt es sich um eine **polyalphabetische Substitution**.

BEISPIEL

Es soll das Schlüsselwort POLY verwendet werden, um den folgenden Satz zu verschlüsseln. Dabei wird es wiederholt stellenweise unter den Klartext geschrieben:

Z U V I E L F U E R C A E S A R
P O L Y P O L Y P O L Y P O L Y

Es ergibt sich folgender Kryptotext:

O I G G T Z Q S T F N Y T G L P

Übungsaufgaben

29 Fassen Sie das Kerckhoffs'sche Prinzip kurz zusammen. Geben Sie je ein Beispiel für ein Verfahren, bei dem das Prinzip erfüllt ist, und für eines, bei dem es nicht erfüllt ist.

30.1 Gegeben ist folgender Text, der mit dem Cäsar-Verschlüsselungsverfahren (Schlüssel: 6) verschlüsselt wurde:

BUXHKXKOZATM

Geben Sie den Klartext an.

TIPP

Beim Entschlüsseln muss die Verschiebung von 6 rückgängig gemacht werden.

30.2 Erläutern Sie, was der größte sinnvolle Schlüssel ist.

TIPP

Gehen Sie vom deutschen Alphabet ohne Umlaute aus.

30.3 Begründen Sie, weshalb ein mehrfaches Verschlüsseln nacheinander mit der Cäsar-Verschlüsselung keine Auswirkungen auf die Sicherheit des Verfahrens hat.

30.4 Erläutern Sie anhand des folgenden Kryptotextes, was beim systematischen Probieren von Schlüsseln ein Ansatz sein könnte, um die Auswahl möglicher Schlüssel einzugrenzen bzw. schnell zum richtigen Schlüssel zu gelangen.

IU IJMVL QAB MA LCVSMT

TIPP

Gehen Sie davon aus, dass die Positionen der Leerzeichen bei der Verschlüsselung gewahrt wurden.

31.1 Erläutern Sie, weshalb das One-Time-Pad-Verfahren explizit die Wiederverwendung von Schlüsseln untersagt.

31.2 Geben Sie einen Nachteil bei der praktischen Umsetzung des One-Time-Pad-Verfahrens an.

32 Erläutern Sie, wie ein Angriff eines Außenstehenden (Man-in-the-Middle-Angriff) beim Public-Key-Verschlüsselungsverfahren aussehen könnte.

TIPP

Starten Sie damit, dass sich der Außenstehende beim Schlüsselaustausch dazwischenschaltet.

Aufgabenblock 2

THEMENBEREICHE

- 3 Praktische Programmierung (ab S. 25)
- 5 Datenbanken (ab S. 81)

Der Informatikprojektkurs möchte ausgewählte Bahnlinien in einer Datenbank abbilden. Dazu sollen in einer ersten Version u. a. die Bahnhöfe und die Streckenabschnitte gespeichert werden.

- Ein Streckenabschnitt ist eine direkte Verbindung *von* dem einen Bahnhof *zu* dem anderen Bahnhof ohne Zwischenhalt. Aus Sicherheitsgründen kann ein Streckenabschnitt nur in einer Richtung und somit nicht für die Gegenrichtung genutzt werden.
- Eine Bahnlinie besteht aus Streckenabschnitten, die die Folge der Bahnhöfe auf dieser Bahnlinie beschreiben. Dabei wird für jeden zugehörigen Streckenabschnitt die Fahrzeit in Minuten angegeben.

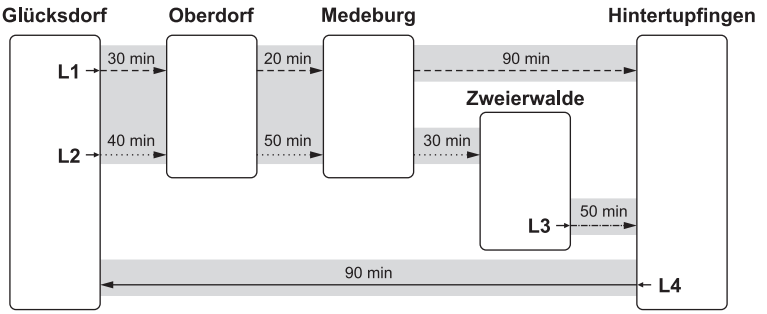


Abbildung 1: Schematische Darstellung der Bahnlinien (L1, L2, L3 und L4) und der Fahrzeiten auf den zugehörigen Streckenabschnitten zwischen den Bahnhöfen (Glücksdorf, Oberdorf, Medeburg, Zweierwalde und Hintertupfingen)

Zur Verwaltung der Daten möchte der Projektkurs eine relationale Datenbank aufbauen, die der folgenden Teilmodellierung entspricht:

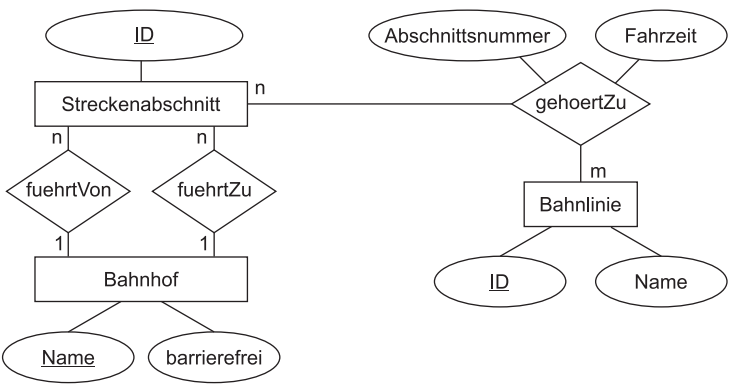


Abbildung 2: Teilmodellierung der Datenbank in Form eines Entity-Relationship-Modells

Im Folgenden soll mit dem in Abbildung 3 gegebenen Datenbankschema gearbeitet werden. Es stellt einen Ausschnitt der Gesamtdatenbank dar. Beispieldaten zu diesem Datenbankschema sind im Anhang zu finden.

```
Bahnhof(Name, barrierefrei)
Streckenabschnitt(ID, ↑vonBahnhofName,
                  ↑zuBahnhofName)
Bahnlinie(ID, Name)
gehoertZu(↑StreckenabschnittID, ↑BahnlinieID,
          Abschnittsnummer, Fahrzeit)
```

Abbildung 3: Datenbankschema eines Ausschnittes der Datenbank

1 Erläutern Sie, wie die Bahnlinie L1 mit den zugehörigen Streckenabschnitten in der Datenbank (vgl. Beispieldaten) gespeichert ist.

Beschreiben Sie die in Abbildung 2 gegebene Teilmodellierung des Entity-Relationship-Modells.

Erläutern Sie jeweils unter Berücksichtigung der Kardinalitäten, wie die im Entity-Relationship-Modell abgebildeten Beziehungstypen im Datenbankschema umgesetzt wurden.

2 Aus der Datenbank sollen folgende Informationen abgefragt werden:

2.1 Gesucht sind für jede Bahnlinie der Name und die jeweilige Gesamtfahrzeit.

2.2 Gesucht sind alle Streckenabschnitte mit der maximalen Fahrzeit. Für diese Streckenabschnitte sollen jeweils die ID der Bahnlinie, der Name des „von-Bahnhofs“ sowie der Name des „zu-Bahnhofs“ und die Fahrzeit des Streckenabschnitts ausgegeben werden.

Entwerfen Sie für die obigen Anfragen 2.1 und 2.2 jeweils eine SQL-Anweisung.

3 Folgende SQL-Anweisung ist gegeben:

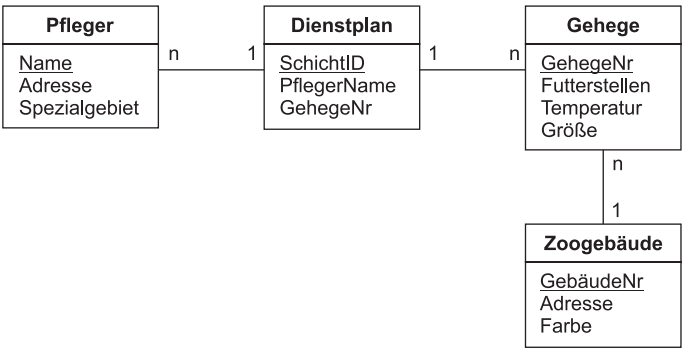
```
 1 SELECT G1.BahnlinieID, S.zuBahnhofName
 2 FROM (
 3     SELECT BahnlinieID,
 4            MAX(Abschnittsnummer) AS max
 5     FROM gehoertZu
 6     GROUP BY BahnlinieID
 7   ) AS G1
 8   INNER JOIN gehoertZu AS G2
 9     ON G1.max = G2.Abschnittsnummer AND
10        G1.BahnlinieID = G2.BahnlinieID
11   INNER JOIN Streckenabschnitt AS S
12     ON G2.StreckenabschnittID = S.ID
```

Analysieren und erläutern Sie die Unterabfrage (Zeilen 3–6).

Relation Gehege – Zoogebäude:
Da ein Zoogebäude mehrere Gehege beherbergen kann, aber ein Gehege nur in einem Zoogebäude zu finden ist, handelt es sich um eine 1 : n-Kardinalität.

28.2 Es lässt sich die Zwischentabelle **Dienstplan** verwenden, in der jede Zeile einer Schicht in einem bestimmten Gehege entspricht. Sie enthält die Primärschlüssel der beiden Tabellen **Pfleger** und **Gehege** als Fremdschlüssel.

Datenbankschema:



Kryptologie

29 Das Kerckhoffs'sche Prinzip postuliert, dass die Sicherheit eines guten Verschlüsselungsverfahrens nicht davon abhängen darf, ob es bekannt ist oder nicht.

Die Cäsar-Verschlüsselung erfüllt das Prinzip nicht. Denn wenn das Verfahren bekannt ist, müssen lediglich alle (z. B. 25) möglichen Schlüssel ausprobiert oder der Schlüssel mit anderen Methoden ermittelt werden.

Beim RSA-Verfahren ist das Prinzip erfüllt. Der Algorithmus ist vollständig veröffentlicht (und sogar der öffentliche Schlüssel), aber die Sicherheit beruht allein auf dem geheimen privaten Schlüssel.

30.1 Klartext:

VORBEREITUNG

Anmerkung:
Die Entschlüsselung kann anhand der nebenstehenden Scheiben nachvollzogen werden.

30.2 Der größte sinnvolle Schlüssel ist 25, weil das deutsche Alphabet 26 Buchstaben enthält (ohne Umlaute). Ein größerer Schlüssel ergibt daher keinen Sinn, da eine Verschiebung um 26 einer Verschiebung um 0 entspricht. Durch die Periodizität des Verschlüsselungsalgorithmus kann man festlegen, dass alle Schlüssel, die Modulo 26 den gleichen Wert ergeben, äquivalent sind.

30.3 Verschlüsselt man einen Text mit der Cäsar-Verschlüsselung mehrfach nacheinander, so ist der Gesamtschlüssel äquivalent zur Summe aller Schlüssel der Verschlüsselungskette. Die Größe des Schlüssels hat aber keine Auswirkung auf die Sicherheit des Verfahrens (und ab Schlüssel 26 wiederholen sich die Schlüssel).

30.4 Man kann damit beginnen, sich die Wörter des Kryptotextes mit zwei Buchstaben anzusehen. In der deutschen Sprache gibt es nur eine begrenzte Anzahl solch kurzer Wörter, die häufigsten sind:
an, im, am, es, du, so, ob, da, in, zu

Man könnte also z. B. diejenigen Schlüssel in Betracht ziehen, die zu diesen Wörtern führen (wobei für das erste Wort selbst „in“ und „im“ ausgeschlossen werden kann, weil der erste Buchstabe im Kryptotext I lautet).

Auch weiß man, dass es keine 2- oder 3-Buchstaben-Wörter gibt, die mit dem Buchstaben X beginnen. Folglich lassen sich die Schlüssel 11, 19 und 15 ausschließen, da diese die Schlüssel wären, die X auf I, Q bzw. M verschieben würden.

31.1 Da dem One-Time-Pad-Verfahren das Vigenère-Verfahren zugrunde liegt, besteht eine Möglichkeit für einen Angriff darin, nach sich wiederholenden Buchstabenfolgen zu suchen (Kasiski-Test). Durch die Wiederverwendung eines Schlüssels ergeben sich ebenfalls Muster, die die Ermittlung der Schlüssellänge und letztendlich des Schlüssels ermöglichen.

31.2 *Mögliche Antworten:*

- Weil der Schlüssel mindestens so lang sein muss wie der Klartext, verdoppelt sich der benötigte Speicherplatz (was insbesondere bei großen Datenmengen relevant sein kann).
- Der Schlüssel muss auf sicherem Wege ausgetauscht werden, was einen erheblichen logistischen Aufwand bedeutet.
- Um die absolute Zufälligkeit des Schlüssels und damit dessen Sicherheit zu gewährleisten, muss ein erheblicher Aufwand betrieben werden. (Echte Zufallszahlen können z. B. durch physikalische Prozesse erzeugt werden.)

Aufgabenblock 2

1 Alle Bahnlinien sind mit `ID` und `Name` in `Bahnlinie` aufgelistet, alle Streckenabschnitte sind mit `ID`, `vonBahnhofName` und `zuBahnhofName` in `Streckenabschnitt` aufgelistet. Diese beiden Tabellen sind über die Tabelle `gehoertZu` verknüpft, in der darüber hinaus für den jeweiligen Streckenabschnitt eine `Abschnittsnummer` bezüglich der jeweiligen Bahnlinie sowie die `Fahrzeit` angegeben ist.
Konkret ist die Bahnlinie mit `ID` L1 und `Name` Bergexpress in `gehoertZu` drei Streckenabschnitten (`ID` 1, 2 bzw. 3) zugeordnet. Zu jedem dieser Streckenabschnitte lässt sich der Start- und Endbahnhof `Streckenabschnitt` entnehmen.

Beispiel eines Streckenabschnitts der Bahnlinie Bergexpress:
Der Streckenabschnitt mit der `ID` 1 hat die `Abschnittsnummer` 1 der Bahnlinie Bergexpress. Die `Fahrzeit` beträgt 30 Minuten, der Startbahnhof ist Glücksdorf und der Endbahnhof Oberdorf.

TIPP

Die 1 : n-Kardinalität von Bahnhof – Streckenabschnitt bedeutet:
Ein Bahnhof kann Teil mehrerer Abschnitte sein, jeder Abschnitt besteht aber nur aus je einem Start- und Endbahnhof.

Die n : m-Kardinalität von Streckenabschnitt – Bahnlinie bedeutet:
Ein Streckenabschnitt kann zu mehreren Bahnlinien gehören, eine Bahnlinie kann aus mehreren Streckenabschnitten bestehen.

Die in Abbildung 2 abgebildete Modellierung beschreibt den Zusammenhang zwischen den Entitäten Bahnhof, Streckenabschnitt und Bahnlinie. Dabei werden die Bahnhöfe mit Namen als Primärschlüssel und mit der Barrierefreiheit als zusätzliches Attribut beschrieben. Jedem Bahnhof werden im Anschluss Streckenabschnitte in zwei separaten 1 : n-Relationen zugewiesen. Die Streckenabschnitte sind wiederum mit dem Primärschlüssel ID deklariert. Den Streckenabschnitten werden über eine n : m-Relation verschiedene Bahnlinien zugeteilt und es wird angegeben, wie lange die jeweilige Fahrzeit ist und welche Abschnittsnummer im Kontext der jeweiligen Bahnlinie sie haben. Bahnlinien werden eindeutig über ihre ID (Primärschlüssel) beschrieben und tragen zusätzlich einen Namen.

Das in Abbildung 3 gezeigte Datenbankschema löst die beiden 1 : n-Relationen `fuehrtVon` und `fuehrtZu` über die beiden Schlüssel `vonBahnhofName` und `zuBahnhofName` in der Tabelle `Streckenabschnitt`. Die n : m-Relation `gehoertZu` wird über eine eigene, gleichnamige Tabelle realisiert. Der Primärschlüssel dieser Tabelle setzt sich aus den beiden Fremdschlüsseln `StreckenabschnittID` und `BahnlinieID` zusammen und bildet außerdem die beiden Attribute `Abschnittsnummer` und `Fahrzeit` ab.

TIPP

Das explizite Nennen der Präfixe, also beispielsweise `Bahnlinie.` und `gehoertZu.`, dient der besseren Lesbarkeit, auch wenn es wegen der Eindeutigkeit der Attribute hier nicht unbedingt nötig ist.

2.1

```
SELECT Bahnlinie.Name,SUM(gehoertZu.Fahrzeit)
FROM gehoertZu
INNER JOIN Bahnlinie
        ON gehoertZu.BahnlinieID=Bahnlinie.ID
GROUP BY Bahnlinie.Name
```

2.2

```
SELECT gehoertZu.BahnlinieID,
       Streckenabschnitt.vonBahnhofName,
       Streckenabschnitt.zuBahnhofName,
       gehoertZu.Fahrzeit
FROM gehoertZu
INNER JOIN Streckenabschnitt
        ON Streckenabschnitt.ID =
           gehoertZu.StreckenabschnittID
WHERE Fahrzeit = (
              SELECT MAX(gehoertZu.Fahrzeit)
              FROM gehoertZu
             )
```

3

Bei der Abfrage der Zeilen 3–6 wird eine Liste aller Bahnlinien-IDs aus `gehoertZu` erstellt und diesen die jeweilige höchste dazugehörige Abschnittsnummer als `max`-Attribut zugewiesen. (Durch die Gruppering nach IDs wird erreicht, dass pro Bahnlinie die maximale Abschnittsnummer ermittelt wird.) Die Liste dient fortan als Unterselektion für die umliegende Abfrage.

Die Maximalwert-Liste wird mit einer zweiten Instanz von `gehoertZu` und mit `Streckenabschnitt` (über die entsprechenden IDs) vereinigt (Zeilen 8–10 und 11–12) . Aus der gesamten Vereinigungstabelle werden die Spalten `BahnlinieID` und `zuBahnhofName` ausgegeben (Zeile 1).

Aus der Unterselektion der maximalen Abschnittsnummer pro der Bahnlinien-ID erhält man die jeweils letzten Streckenabschnitte. Insgesamt ergibt sich also eine Liste aller Bahnlinien-IDs mit dem jeweiligen Endbahnhof (der gesamten Linie).

TIPP

Erklärung der Methoden:

`toFirst()`: Das erste Objekt der Liste wird aktuelles Objekt.

`hasAccess()`: Gibt `true` zurück, wenn es ein aktuelles Objekt gibt.

`getContent()`: Gibt das aktuelle Objekt zurück, falls es eines gibt.

`remove()`: Löscht das aktuelle Objekt.

4

Die Methode `ermittleEtwas` soll alle Bahnhöfe in einer bestimmten „Entfernung“ (Anzahl `pHops` der „Schritte“) von einem Startbahnhof `pBName` bestimmen.

- Zeile 3: `ermittleNachbarbahnhoefe(pBName)` liefert eine Liste der zu `pBName` direkt benachbarten Bahnhöfe.
- Zeile 4: Die Schleife läuft, solange `pHops > 0`, d. h. noch „Schritte“ übrig sind.
- Zeilen 5–15: In jeder Iteration:
  - Für jeden Bahnhof in `nachbarn`
    1. wird der Bahnhof zu `sammlung` hinzugefügt (gespeichert).
    2. werden mit `ermittleNachbarbahnhoefe()` wiederum alle Nachbarn ermittelt und zu `temp` hinzugefügt.
  - Danach wird `nachbarn` durch `temp` ersetzt (d. h., man geht eine Stufe weiter im Netzwerk).
  - `pHops` wird dekrementiert.

www.stark-verlag.de
info@stark-verlag.de

STARK