

Datenbanken – (etwas) anders gesehen

Peter K. Antonitsch

Institut für Informatiksysteme/Informatik Fachdidaktik
Alpen-Adria Universität Klagenfurt
Universitätsstraße 65-67
A 9020 Klagenfurt
Peter.Antonitsch@uni-klu.ac.at

Abstract: Im Informatikunterricht werden Datenbanken entweder vom Ansatz des Anwendens oder des Modellierens behandelt. Diese beiden Ansätze sind als „Verwenden von Strukturen“ bzw. „Erzeugen von Strukturen“ zueinander komplementär, vernachlässigen aber häufig die Rückkoppelung des durch die Datenbank repräsentierten Modells mit dem modellierten Realweltausschnitt. Dieser Artikel beschreibt erste Erfahrungen mit einem alternativen Ansatz, der ausgehend von Datenbankabfragen die beiden oben genannten Struktur Aspekte zu vereinen sucht und auch dem Erlernen der bei relationalen Datenbanken verwendeten Strukturierungsmethoden sowie dem Hinterfragen der Strukturen Raum gibt. Zentraler Gedanke ist dabei das Erforschen vorgegebener Datenbankstrukturen durch die Lernenden.

1 Datenbanken im Informatikunterricht

Relationale Datenbanken werden traditionell als Kernstoff im Informatikunterricht angesehen (vgl. [BM04] zur Situation in Österreich). Zwar wird die didaktische Diskussion über Methoden zur schülergerechten Aufbereitung des Themas „Datenbanken“ bei weitem nicht so intensiv geführt wie dies z.B. bei der Einführung in das Programmieren der Fall ist, dennoch lassen sich in der fachdidaktischen Literatur zumindest zwei gängige Vermittlungskonzepte ausmachen. Diese unterscheiden sich voneinander dadurch, dass entweder der „Anwendungsaspekt“ oder der „Modellierungsaspekt“ von Datenbanksoftware besonders betont wird:

Motiviert durch die zunehmende Popularität von Zertifikaten als Nachweis von „Computerwissen“ findet das Konzept der Anwenderschulung auch in den Schulen Beachtung. Dies gilt (zumindest in Österreich) weitgehend für den Informatikunterricht der Sekundarstufe 1, wo grundlegende Fertigkeiten des Arbeitens mit dem Computer vermittelt werden [Ro06]. Dies setzt sich aber auch in der Sekundarstufe II bei Datenbanken als „anwendungsnahe“ Software fort [Ne04] [Ha05a]. Charakteristisch für das Arbeiten auf dieser „Anwendungsebene“ ist die Konzentration auf den Werkzeugaspekt der jeweiligen Datenbanksoftware, während der konzeptionelle Hintergrund bzw. das Design von Datenbanken allenfalls als Weiterführung bzw. Vertiefung angesprochen werden [Ko04] [Ha05b].

Nahezu komplementär zur Anwenderschulung betont das zweite gängige Vermittlungskonzept das Modellieren. Mit der häufig anzutreffenden Themenfolge „konzeptueller Entwurf – logischer Entwurf – Normalisierung – Abfragen“ (z.B. [Sc05] oder [Ja03]) orientiert sich dieser Ansatz offenbar an universitären Datenbankkursen. Der Schwerpunkt der Schülerinnen- und Schülertätigkeit liegt dabei auf dem Entwurf von Datenbanken im Sinne von regelbasiertem Strukturieren eines vorgegebenen Realweltausschnittes. Das Benutzen der Datenbank als Quelle von Informationen tritt auf dieser „Modellierungsebene“ etwas in den Hintergrund.

2 Ein alternativer Ansatz...

Die Lernenden werden bei der Auseinandersetzung mit Datenbanksoftware nach obigen Vermittlungskonzepten auf der Anwendungs- oder der Modellierungsebene tätig. Beide Ebenen sind über das Datenmodell der relationalen Tabellenstruktur miteinander gekoppelt und überschneiden sich im Bereich der Datenbankabfragen (vgl. Abb. 1).

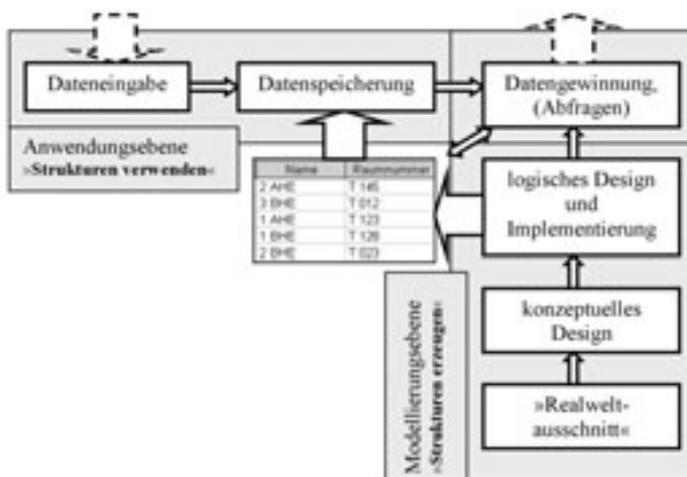


Abbildung 1: Tätigkeitsorientierte Sicht auf Anwendungs- und Modellierungsebene von Datenbanksoftware

Die Lernenden verwenden dabei die zugrunde liegende Struktur in unterschiedlicher Weise: Auf der Anwendungsebene meint „verwenden“ tatsächlich das Benutzen bereits vorhandener Strukturen, auf der Modellierungsebene das Erzeugen der Strukturen, die für die Darstellung des Realweltausschnittes passend sind. Beides lässt sich grob in „Strukturieren“ zusammenfassen.

Strukturieren kennzeichnet sowohl den informatischen als auch den allgemeinbildenden Kern des Unterrichtsthemas „Datenbanken“. Einerseits kennt der fachdidaktische Diskurs das fundamentale informatische Prinzip der strukturierten Zerlegung ([Sc93], S. 23), das im Informatikunterricht vermittelt werden soll. Die „Beherrschung von Komplexität durch Strukturierung“ wird als Kompetenz zur Bewältigung von informatischen

Alltagsanforderungen angesehen ([SS04], S. 40). Diese Kompetenz gilt es, im Rahmen der informatischen Bildung zu entwickeln. Dies weist darauf hin, dass Strukturieren andererseits grundlegende Bedeutung über die Informatik hinaus hat: Durch das Strukturieren von Erfahrungen erschließt sich der Mensch die Welt, Lernen selbst kann als Wahrnehmen, Erkennen und Anwenden von Strukturen verstanden werden [An06]. Als basales Ziel von Allgemeinbildung, das durch Datenbankunterricht erreicht werden kann, ist allerdings das Strukturieren noch zu präzisieren – eine Beschränkung bloß auf das Erzeugen und Benutzen von Strukturen greift zu kurz. „Vollständiges Strukturieren“ ist in diesem Zusammenhang vielmehr als Prozess mit den Phasen „Erlernen von Regeln zur Strukturbildung“ – „Anwenden von Regeln zur Strukturbildung“ – „Datengewinnung aus erzeugten Strukturen“ – „Hinterfragen erzeugter Strukturen“ zu verstehen (vgl. Abb. 2).

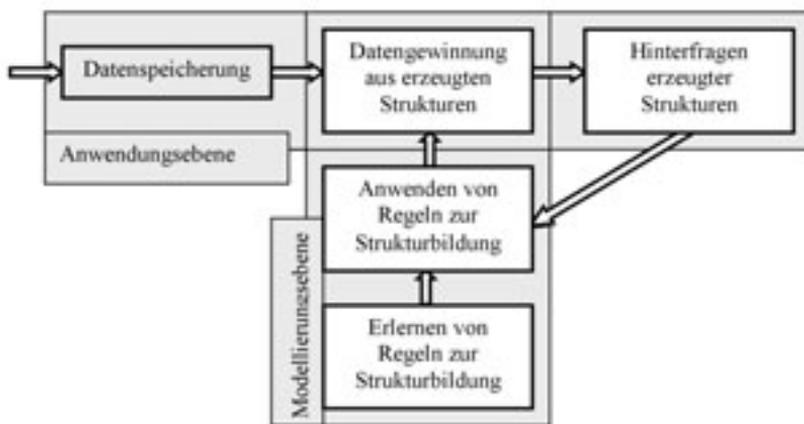


Abbildung 2: Phasenstruktur von „vollständigem Strukturieren“, integriert in die Anwendungs- und Modellierungsebene nach Abb. 1

2.1 Konzept „Erforschen und Anwenden“

Abb. 1 zeigt das Abfragen von Daten als zentrale Tätigkeit bei der Auseinandersetzung mit Datenbanken. Dies motiviert ein Vermittlungskonzept, das ausgehend von Datenbankabfragen alle vier oben identifizierten Strukturierungsaspekte integriert und so nicht nur die innerinformatische Bedeutung von Datenbanken betont, sondern auch dem allgemeinbildenden Aspekt des Strukturierens genügen kann:

Der typische Anwendungsfall für Datenbanken ist das Verknüpfen von Daten, die auf verschiedene Tabellen verteilt sind, mit dem Ziel, dadurch subjektiv neue Informationen zu gewinnen. Das Verknüpfen von Tabellen erfordert allerdings die Kenntnis der relationalen Datenstruktur. Dadurch ergibt sich die Möglichkeit, elementare Bausteine bzw. Regeln des Strukturierens/Modellierens wie „Entität“, „Attribut“, „Schlüsselattribut“ oder (Abbilden einer) „Beziehung“ mit Hilfe von Beispieldatenbanken lernbar zu machen, indem die Struktur von den Lernenden anhand geeignet formulierter Lernaufgaben

erforscht und erschlossen wird. Ausgangspunkt in dieser Phase des Erforschens ist dabei die allgemeine Fragestellung: „Welche Tabellen müssen verknüpft werden, um eine bestimmte Information zu erhalten, und welcher Mechanismus ermöglicht das Verknüpfen dieser Tabellen?“ Dieser Zugang zum „Erlernen von Regeln zur Strukturbildung“ folgt den Ergebnissen neurobiologischer Forschung, nach denen Regeln von unserem Hirn selbst produziert werden, wenn es (viele) strukturell ähnliche Beispiele verarbeitet ([Sp02], S. 68ff).

Die Erfahrungen der Lernenden aus der ersten Phase führen auf drei Anwendungskontexte: Die Verknüpfungen, die zum Extrahieren gewünschter Daten benötigt und beim Erforschen gefunden wurden, können direkt als Datenbankabfragen formuliert werden („klassischer“ Anwendungskontext). Die durch die Anfragen gewonnenen Informationen bilden die Grundlage für die Rückkoppelung des Modells mit der Realität, für das „Hinterfragen erzeugter Strukturen“ („reflexiver“ Anwendungskontext¹). Die erschlossenen Strukturierungsregeln schließlich erlauben das eigenständige Erstellen weiterer Datenbankmodelle von Realweltausschnitten durch die Lernenden („gestaltender“ Anwendungskontext).

Kern des Ansatzes ist also die didaktische Transformation von Anwendungs- und Modellierungsebene unter der Leitidee des vollständigen Strukturierens und dem Ziel, die aktive Beteiligung der Lernenden am Lernprozess zu erhöhen. Diese didaktische Transformation äußert sich auf Inhaltsebene durch die geänderte Akzentuierung und Abfolge von Komponenten gängiger Datenbank-Vermittlungskonzepte (vgl. Abb. 3).

2.2 Zum Problem des „guten“ Einstiegsbeispiels

Das „gute“ Einstiegsbeispiel ist der Angelpunkt des skizzierten Vermittlungskonzepts. Primär muss es Abfragen ermöglichen, die zur Auseinandersetzung mit der vorbereiteten Datenbank und damit zum Erforschen der vorhandenen Strukturen motivieren. Interessante Abfragen erfordern aber, dass die Lernenden mit dem modellierten Realweltausschnitt vertraut sind. Dies ist auch wichtig, damit auf Basis der Vertrautheit mit dem realen System die Abstraktionen des relationalen Modells erfasst werden können und auch eine ernsthafte Rückkoppelung des Modells mit der Realität möglich wird. Letztlich muss das Einstiegsbeispiel einerseits groß genug sein, damit die wesentlichen Strukturmerkmale des relationalen Modells repräsentiert sind, andererseits sollte es aber nicht zu komplex sein, damit die wesentlichen Strukturmerkmale erkennbar bleiben.

Zur Erläuterung des letzten Aspekts dient eine datenbankmäßige Abbildung von „Schule“ (vgl. Abb. 4), die die genannten Kriterien erfüllt und auch für die erste Erprobung des beschriebenen Vermittlungskonzepts im schulischen Informatikunterricht als Einstiegsbeispiel herangezogen wurde (s. Abschnitt 3).

An dem aus Sicht der Lernenden tatsächlich „großen“ Beispiel wurden zur Reduktion der Komplexität folgende strukturellen Vereinfachungen vorgenommen:

- Ausklammern von Aggregationen und Generalisationen als „strukturelle Besonderheiten“ (die auch anders modelliert werden können);

¹ Dieser reflexive Anwendungskontext kann (und sollte) auch zum Thematisieren von Aspekten des Datenschutzes und des Rechts auf informationelle Selbstbestimmung genutzt werden.

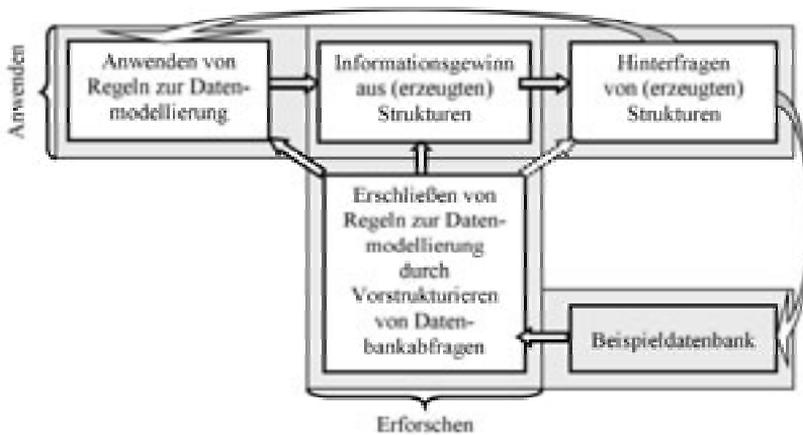


Abbildung 3: Didaktische Transformation der Phasentruktur von „vollständigem Strukturieren“ zur Vermittlung im Datenbankunterricht

- Darstellen jeder Assoziation als eigene Tabelle, unabhängig von den Kardinalitäten der beteiligten Entitäten, und damit im Zusammenhang
- Unterscheiden der Tabellen, die Entitäten darstellen, von denen, die Assoziationen darstellen, z.B. durch unterschiedliche Benennung;
- ausschließliches Modellieren von Entitäten, deren Primärschlüssel aus einem einzigen Schlüsselattribut besteht
- Visualisieren der Datenbankstruktur.

Diese Vereinfachungen und auch die in Abb. 4 dargestellte Beispieldatenbank weisen darauf hin, dass mit dem Einstiegsbeispiel keine „perfekte“ Datenbanklösung angestrebt werden soll. Dies scheint auch unter dem Aspekt des Hinterfragen-Könnens wenig sinnvoll. Wesentlich ist jedoch, dass das Einstiegsbeispiel der Erfahrungswelt der Lernenden entstammt (s.o.) und das strukturelle Grundkonzept des relationalen Datenmodells korrekt repräsentiert wird. „Fehler“ im Modell sind daher durchaus didaktisch intendiert und sollen als Möglichkeit für neue Lernerfahrungen genutzt werden, ganz im Sinne einer konstruktivistischen Fehlerkultur (siehe z.B. [Le01]).

3 ...und erste Erfahrungen

Das oben skizzierte Konzept wurde im Sommersemester des Schuljahres 2005/06 mit Schülern² im Alter von 16 bis 17 Jahren erstmals erprobt, als DBMS wurde Microsoft

² Konkret handelte es sich um eine Gruppe von 13 Schülern und einer Schülerin. Als Tribut an die Lesbarkeit wird im Artikel bei Pluralformen jedoch nur das männliche grammatikalische Geschlecht verwendet, das biologische weibliche ist aber selbstverständlich stets mitgemeint.

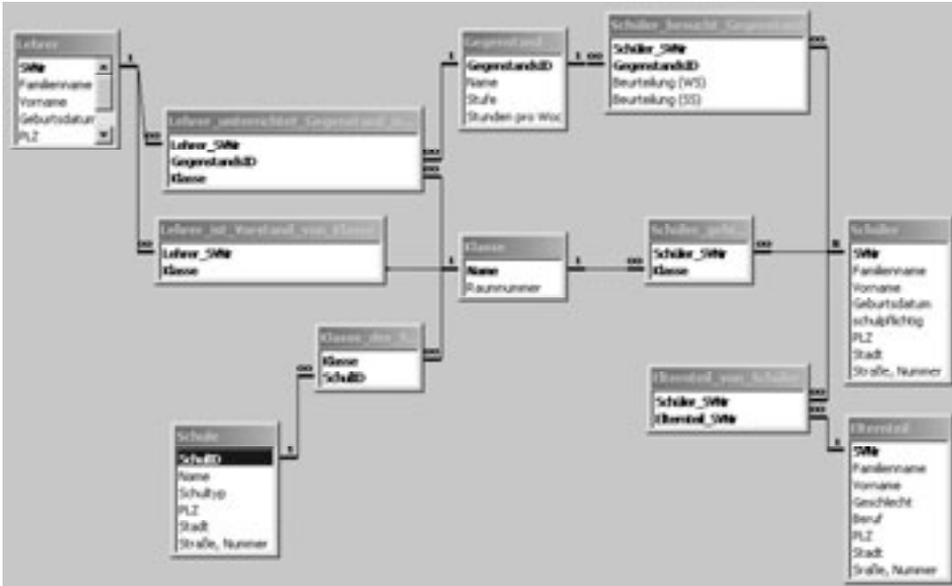


Abbildung 4: Beziehungsdiagramm eines Einstiegsbeispiels zum Erforschen und Hinterfragen der gegebenen Datenbankstruktur gemäß dem beschriebenen Ansatz

Access verwendet³. Geplant war die eine Einführung in das Thema „Datenbanken“ im Ausmaß von 12 Doppelheiten (à 100 Minuten), infolge von nicht vorhergesehenen Stundenausfällen stand jedoch tatsächlich nur etwas mehr als die Hälfte dieser Zeit zur Verfügung. Die informatischen Vorkenntnisse der Lernenden bestanden im Wesentlichen aus Programmierkenntnissen in C, wobei vor dem Datenbankkapitel der Verbund-Datentyp zum Verwalten von Datensätzen behandelt wurde.

3.1 Erforschen: Semantik des Beziehungsdiagramms

Die erste Phase (ca. eine Einheit) diente der Klärung grundlegender Begrifflichkeiten. Die Lehrperson erläuterte den Sinn einer Datenbank als „Beschreibung einer statischen Situation mit dem Ziel, zusammengehörige Daten zu verknüpfen“, und führte „Attribut“ als Synonym für „Komponente“ eines Verbunddatentyps ein. Mit diesem sprachlichen Grundgerüst ausgestattet erhielten die Lernenden die Aufgabe, anhand des Beziehungsdiagramms (vgl. Abb. 4) die Bedeutung der dargestellten „Rechtecke“ und „Linien“, insbesondere aber der unterschiedlich ausgezeichneten Attribute („normal“ bzw. „fett“) zu klären. Dazu erhielten sie einen zusätzlichen Impuls der Form: „Im Beziehungsdiagramm sind bestimmte Rechtecke durch Linien verbunden. Sie möchten wissen, in wel-

³ Die Verwendung von Microsoft Access erklärt sich aus dem Umstand, dass diese Datenbanksoftware an österreichischen Schulen am verbreitetsten ist und auch die für die benötigten Visualisierungen der Datenbankstruktu geeignete Werkzeuge bereitstellt. Selbstverständlich ist aber jedes andere DBMS mit vergleichbarer Funktionalität ebenso geeignet.

chem Raum die Klasse eines bestimmten Schülers (z.B. des Schülers ‚Huber‘) ist. Helfen Ihnen die Verbindungslinien beim Lösen dieser Aufgabe?“

Die Schüler äußerten⁴ sich zunächst derart, dass die Rechtecke das darstellen, „was es gibt, und auch anderes“ und führten als Unterscheidungskriterium die unterschiedlich gebauten „Namen“ der Rechtecke an. Nach der Analyse der „Linien“ wurde die Vermutung geäußert, dass dieses „Andere“ beschreibt, wie „das, was es gibt, zusammenhängt.“ Hinsichtlich der fett ausgezeichneten Attribute fanden die Schüler daraufhin selbst, dass diese „Anknüpfungspunkte“ zwischen den Rechtecksverbindungen darstellen, einige wenige wandten auch ein, dass diese die „Rechtecke“ identifizieren, weil ja z.B. die Sozialversicherungsnummer eindeutig sein müsse.

Die gegensätzliche Interpretation dieser Attribute konnte erst durch die Erläuterung der Lehrperson, dass es sich dabei sowohl um identifizierende als auch um verbindende Attribute handelt, aufgelöst werden. „Identifizierendes Attribut“ bzw. „verbindendes Attribut“ wurde auch in weiterer Folge anstelle des abstrakteren „Schlüsselattribut“ verwendet. Diese Begriffsbildung führte bei den Schülern zu der weiteren Beobachtung, dass diejenigen Rechtecke, die „Dinge“ bezeichnen, nur ein identifizierendes Attribut besitzen, die anderen aber, die Verbindungen darstellen, so viele, „wie durch sie Rechtecke verknüpft werden“. Schließlich fragten die Schüler auch noch nach den „Ziffern“ 1 bzw. ∞ , die bei den Linien dabeistehen, was durch die Lehrperson damit beantwortet wurde, dass jedes Ding (1) beliebig oft (∞) an den entsprechenden Verbindungen teilnehmen könne⁵.

3.2 Erforschen: Makro- und Mikrostruktur der Datenbank

In der zweiten Phase (ca. eine Einheit) wurden die „Rechtecke“ des Beziehungsdigramms als Tabellen identifiziert. Dabei wurden Tabellenentwurf und Datenblattansicht unter dem Aspekt der Datenverknüpfung einander gegenübergestellt (vgl. Abb. 5). Die Schüler erhielten sodann die Aufgabe, den Aufbau und den Zusammenhang der Tabellen „Lehrer“, „Lehrer_ist_Vorstand_von_Klasse“ und „Klasse“ (ohne Verwendung des Computers) ebenfalls in dieser Form darzustellen.

Mit der neuen Sichtweise „alles ist eine Tabelle“ wurden die Schüler durch Aufgaben ähnlich der nachfolgenden angeleitet, im Beziehungsdigramm „Verknüpfungspfade“ über mehrere Tabellen zu identifizieren: „Die Schülerin Huber möchte eine Liste aller Lehrer haben, die sie unterrichten. Welche Tabellen werden benötigt, um diese Information aus der Datenbank zu erhalten? Wie werden die Verknüpfungen zwischen den einzelnen Tabellen hergestellt?“ Dieser Wechsel von der Mikrostruktur der Tabellen zurück zur Makrostruktur der Beziehungen diente einerseits dem Zusammenführen dieser beiden Sichtweisen, andererseits aber auch zur Vorbereitung auf die Formulierung von Abfragen.

⁴ Bei der Darstellung von Schüleräußerungen wird teilweise der Originalwortlaut verwendet, weil sich in diesem auch die Annäherung der Schülervorstellungen an die Begrifflichkeiten der Datenbankwelt dokumentiert.

⁵ Bei dieser Frage verstellt die Modellierung jeder Assoziation als eigene Tabelle natürlich den Blick auf die im konzeptuellen Entwurf auftretenden Kardinalitäten/Konnektivitäten. Offenbar war aber die Erklärung der Lehrperson zu diesem Zeitpunkt mit den Vorstellungen der Schüler vereinbar und wurde akzeptiert.

Angesichts der aus Sicht der Schüler teilweise recht hohen Zahl von bis zu fünf zu verknüpfenden Tabellen wurde die Frage aufgeworfen, weshalb nicht die gesamte Information in einer einzigen Tabelle abgelegt werden könne. Klarerweise konnte dies entsprechend dem Kenntnisstand der Lernenden an dieser Stelle nicht mit „Anomalie“ und „Redundanz“ erklärt werden. Die Lehrperson machte aber anhand eines einfachen Beispiels (Zusammenführen der Tabellen „Schüler“, „Schüler_besucht_Gegenstand“ und „Gegenstand“) bewusst, dass die Speicherung in einer Tabelle zunächst einen Mehraufwand bei der Eingabe der Daten bedeutet, weil in diesem Fall gleiche Daten mehrfach in die Tabelle eingetragen werden müssen. Dabei konnte der Begriff „Redundanz“ eingeführt (!) werden.

3.3 Anwenden: Erstellen von Abfragen mit QBE und SQL

In der dritten Phase (2 Einheiten) wurden zunächst mit Hilfe von „Query by Example“ erste Datenbankabfragen formuliert. Die Wahl von QBE zum Einstieg in Datenbankabfragen erwies sich in mehrfacher Hinsicht von Vorteil:

- Erstens konnten die zuvor gefundenen „Verknüpfungspfade“ eins zu eins in QBE übertragen werden, was schnelle Erfolgserlebnisse bei den Lernenden ermöglicht.
- Die Lernenden hatten – entsprechend dem erworbenen „mental Modell“ der Datenbank – individuelle Erwartungshaltungen, wie das Ergebnis einer Abfrage aussehen sollte. Die Möglichkeit des schnellen Wechsels von der Entwurfsansicht zur Datenblattansicht bot die Gelegenheit, das erwartete mit dem tatsächlichen Ergebnis zu vergleichen, und so Fehler in der Formulierung der Abfrage oder im eigenen Verständnis von der Struktur der Datenbank (!) zu entdecken.
- Abfragen können in QBE vergleichsweise schnell „codiert“ werden. Dadurch erwarben die Lernenden bald einen „Erfahrungsschatz“, vor allem aber gelangten sie von selbst zur Erkenntnis, dass das Ergebnis von Abfragen auf Tabellen stets wieder eine Tabelle ist, und so Grundlage für eine weitere Abfrage sein kann.

Der Übergang von QBE zu SQL erfolgte durch den Wechsel zum „im Hintergrund“ erzeugten SQL-Code, der die wesentliche Struktur einfacher Abfragen (SELECT – FROM – WHERE) zeigt. Für zusammengesetzte Abfragen (d.h. Queries mit Subqueries) wurde der Aspekt wieder aufgegriffen, dass Abfragen stets wieder Tabellen liefern, die Grundlage für weitere Abfragen sein können. Dies sei am Beispiel der folgenden Abfrage kurz erläutert: „Gesucht sind die Beurteilungen aller Schüler in allen Gegenständen. Die Ergebnistabelle soll auch die Namen der Schüler und die Namen der Unterrichtsgegenstände enthalten.“ Der „Verknüpfungspfad“ besteht aus den Tabellen „Schüler“, „Gegenstand“ und „Schüler_besucht_Gegenstand“ (vgl. Abb. 4). Als Zwischenergebnis wurde zunächst eine Tabelle „Schüler_und_Beurteilungen“ mit den relevanten Attributen erstellt:

```
SELECT ... FROM Schüler INNER JOIN Schüler_besucht_Gegenstand
ON Schüler.SVNr = Schüler_besucht_Gegenstand.Schüler_SVNr;
```

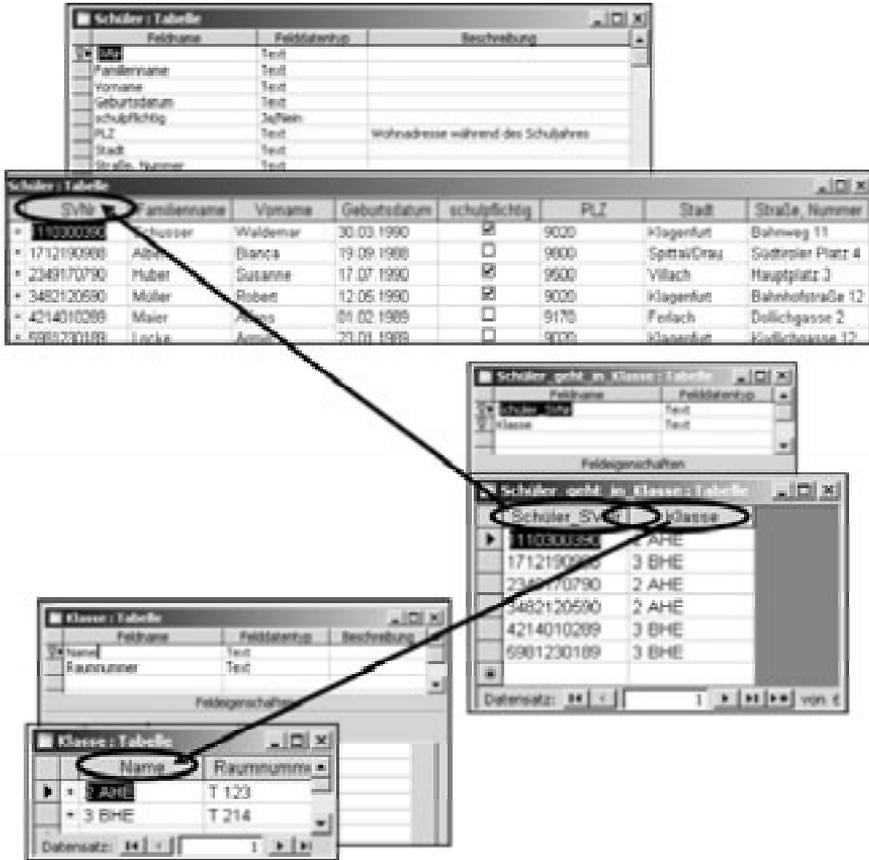


Abbildung 5: Darstellung der „Rechtecke“ aus der Beziehungssicht als Tabellen unter Betonung der „verbindenden Attribute“ – deren Funktion wird durch die einander überlappenden Ellipsen in der Tabelle „Schüler_geht_in_Klasse“ hervorgehoben

Auf Basis der Tabelle zu dieser Subquery lässt sich dann die eigentliche Abfrage recht einfach formulieren:

```
SELECT ... FROM Gegenstand INNER JOIN Schüler_und_Beurteilungen
ON Gegenstand.GegenstandsID = Schüler_und_Beurteilungen.GegenstandsID,
```

verglichen mit:

```
SELECT ... FROM Gegenstand INNER JOIN
(Schüler INNER JOIN Schüler_besucht_Gegenstand
ON Schüler.SVNr=Schüler_besucht_Gegenstand.Schüler_SVNr)
ON Gegenstand.GegenstandsID=Schüler_besucht_Gegenstand.GegenstandsID
```

Zusätzlich wurde dieser Prozess der Modularisierung mit Hilfe der QBE-Entwurfsansicht visualisiert:

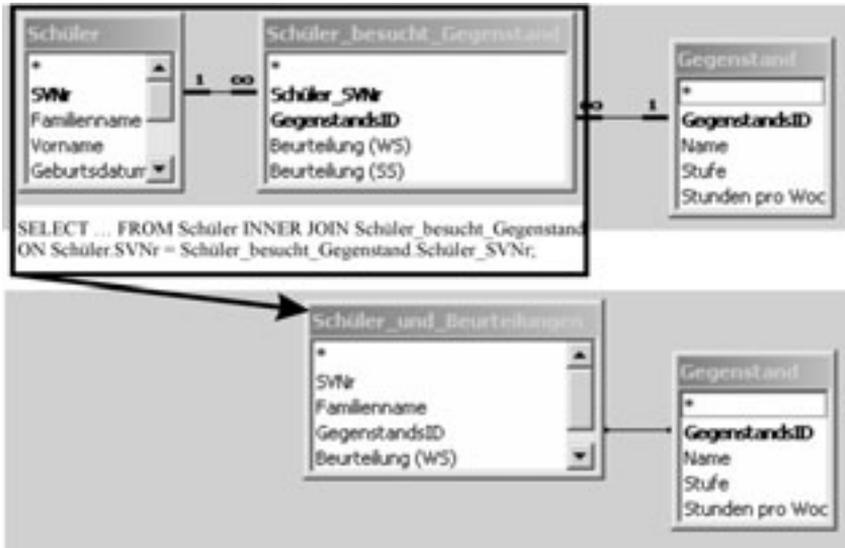


Abbildung 6: Visualisierung von Subqueries mit Hilfe der Ansicht „Abfrageentwurf“

Die Schüler erwarben durch derartige Sequenzen ein „Vorgehensmodell“ zum Formulieren zusammengesetzter Abfragen.

3.4 Anwenden: Hinterfragen und Modellieren

In der letzten Phase (etwa 4 Einheiten) der Unterrichtssequenz wurden die Lernenden angeleitet, die vorgegebene Datenbankstruktur zu hinterfragen. Zusätzlich sollte auch ein einfaches „Tabellenschema“ (entsprechend dem bisher diskutierten) für einen anderen Realweltausschnitt eigenständig entworfen werden.

Zum Einüben des Formulierens von Abfragen wurden den Schülern zunächst weitere, „interessante“ Aufgaben vorgelegt, wie z.B.: „Erzielen Mädchen in Schule bessere Ergebnisse als Burschen?“ (vgl. [Mo92] für weitere Beispiele von Abfragen, die auch „sensiblere“ Daten erschließen). Abgesehen davon, dass derartige Fragestellungen vielfältige Assoziationen der Lernenden zur eigenen Schulsituation hervorriefen (als „Rückkoppelung“ zur „Realität“), ist speziell diese Frage in der gegebenen Beispieldatenbank schlechterdings nicht zu beantworten, da in der Tabelle „Schüler“ das Geschlecht nicht „gemerkt“ wird.

Auf der Suche nach weiteren „Schwächen“ der Datenbank bemerkten die Schüler, dass ja jede Klasse nur einen Klassenvorstand haben kann. Diese Eindeutigkeit kann aber durch die Assoziationstabelle „Lehrer_ist_Vorstand_von_Klasse“ nicht abgebildet werden. Die Schüler äußerten aber recht schnell eigenständig (!) die Vermutung, dass dieses Problem doch durch „Verschieben“ des identifizierenden Attributs von „Lehrer“ zu „Klasse“ gelöst werden könne. Es gäbe offenbar verschiedene Formen von Assoziatio-

nen („eindeutige und nicht-eindeutige“), die im Datenbankmodell auch unterschiedlich abgebildet werden müssten. Besonders bemerkenswert ist die auf dieser Erkenntnis aufbauende Schlussfolgerung eines Schülers, der meinte, es sei dann wohl nicht optimal, beim „Entwerfen“ der Datenbank von dem Tabellenschema auszugehen. Sinnvollerweise sollte es ein „vorgelagertes Schema“ geben.

Mit diesem Impuls und dem bereits vorhandenen Strukturwissen der Schüler konnte die konzeptuelle Modellierung (außer Plan!) rasch besprochen werden. Dabei wurde die UML-Notation verwendet, da diese der Darstellung im Access-Beziehungsdiagramm sehr ähnlich ist. Auch beim daraufhin eigenständig zu erstellenden Datenbankentwurf verwendeten die Schüler diese Form der Notation. Die Überführung in ein konsistentes Tabellenschema (inklusive „besserer“ Auflösung von (1:n)-Beziehungen) bereitete aufgrund der Vorerfahrung mit Datenbank-Tabellen keine Schwierigkeiten.

4 Resümee und Ausblick

Informatikunterricht erhebt den Anspruch, allgemeinbildend zu sein. Damit sind bei der Auswahl und der Aufbereitung der zu unterrichtenden Themen unter anderem die Fragen zu beantworten, inwieweit das jeweilige Thema ein über sich hinausweisendes Merkmal aufweist, und ob dieses Allgemeine von den Lernenden in der arrangierten Lernsituation auch tatsächlich erfasst werden kann (nach [Hu00], S.57).

Strukturieren kann als ein allgemein bildendes Merkmal der Beschäftigung mit Datenbanken angesehen werden. Die Beobachtung, dass bei gängigen Vermittlungskonzepten zu Datenbanken das Strukturieren aber häufig auf das bloße Anwenden mehr oder weniger verstandener Regeln durch die Lernenden reduziert wird, motivierte den alternativen Ansatz, nach dem die Lernenden durch Anwenden von Datenbanken zunächst Verständnis für die zugrunde liegenden Strukturen gewinnen. Dabei ist das vorgestellte Konzept nicht das einzige, das mit diesem Ziel den von Anwenderschulungen her bekannten Ansatz des Arbeitens mit „fertigen“ Beispieltabellen aufgreift. Diese Idee findet sich auch in [FKN06] (S. 5 – 78), jedoch werden dort vergleichsweise „kleine“ Datenbanken verwendet und die Strukturen schrittweise vom Lehrer/von den Autoren erklärt. Zum Unterschied davon betont das oben erläuterte Konzept das „angeleitete Erforschen und Hinterfragen“ der Strukturen einer „großen Datenbank durch die Lernenden selbst.

Die beschriebenen ersten Erfahrungen weisen darauf hin, dass der vorgeschlagene Ansatz – auch bei knapper Zeitressource – zum intendierten Lernerfolg führen kann. Für den Einsatz des Konzepts im Informatikunterricht scheint aber noch zweierlei notwendig. Zum einen sind weitere geeignete Beispiele zu finden. Dies meint Einstiegsbeispiele, die der Erfahrungswelt der Schülerinnen und Schüler entstammen (vorstellbar wäre z.B. das Modell eines Pizza-Service oder einer Online-Tauschbörse), aber auch Beispiele, an denen das erworbene Strukturverständnis angewandt werden kann. Insbesondere bei der zweiten Gruppe kann auf vorhandene und kommentierte gute Beispieldatenbanken, wie z.B. „Flotter Flitzer“ von J. Penon/S. Spolwig zurückgegriffen werden⁶. Zum anderen erscheint die Erweiterung des Ansatzes in Richtung „Erforschen von Redundanz

⁶ Die angeführte Beispieldatenbank findet sich – neben anderen – unter <http://www.be.schule.de/bics/inf2/datenbanken/unterrichtsbeispiele.html>. (24. 04. 2007)

und Anomalien“ ebenso sinnvoll und notwendig, wie eine Vertiefung des reflexiven Anwendungskontextes und das Entwickeln von Konzepten zur Überprüfung der von den Lernenden mit diesem Ansatz erworbenen Kompetenzen.

Literaturverzeichnis

- [An06] Antonitsch P.K.: Databases as a Tool of General Education. In (Mittermeir R.T. Ed.): Informatics Education – The Bridge between Using and Understanding Computers. Proc. 2nd Int. Conf. On Informatics in Secondary Schools – Evolution and Perspectives (IS-SEP), Vilnius 2006. Springer, Berlin Heidelberg, 2006; S. 59-70
- [BM04] Bundesministerium für Bildung, Wissenschaft und Kultur: Lehrplan der AHS-Oberstufe für den Pflichtgegenstand Informatik (Version 2004); verfügbar unter: http://www.bmukk.gv.at/medienpool/11876/lp_neu_ahs_14. (24.04.2007) bzw. Lehrplan für den Wahlpflichtgegenstand Informatik (Version 2004); verfügbar unter http://www.bmukk.gv.at/medienpool/11876/lp_neu_ahs_21. (24.04.2007)
- [FKN06] Fischer H., Knapp T., Neupert H.: Grundlagen der Informatik II. Oldenbourg, München et al. 2006
- [Ha05a] Habenicht K.: Grundlagen der Datenverarbeitung mit MS Access. ADIM Schriftenreihe Wissen in Beispielen, Band 205, Wien 2005; s.a.: <http://www.adim.at/wib/> (17.01.2007)
- [Ha05b] Habenicht K.: Fortgeschrittene Datenverarbeitung mit MS Access. ADIM Schriftenreihe Wissen in Beispielen, Band 206, Wien 2005; s.a.: <http://www.adim.at/wib/> (17.01.2007)
- [Hu00] Hubwieser P.: Didaktik der Informatik. Springer, Berlin – Heidelberg 2000
- [Ja03] Jaros H.: Grundkurs Datenbankentwurf, 3. Aufl. Vieweg, Wiesbaden 2003
- [Ko04] Konopasek K.: Access 2003 Advanced ECDL Modul AM 5, 3. Aufl. ikon, Brunn am Gebirge 2004
- [Le01] Leuders T.: Qualität im Mathematikunterricht. Cornelsen, Berlin 2001
- [Mo92] Modrow, W.: Zur Didaktik des Informatik-Unterrichts, Band2. Dümmler, Bonn 1992
- [Ne04] Neubauer G.: Access 2003 Grundlagen ECDL Modul 5, 5. Aufl.. ikon, Brunn am Gebirge 2004
- [Ro06] Rohrer M.: Evaluation des Informatikunterrichts in den 1./2. Klassen der AHS in Kärnten. Veröffentlichung im Rahmen des IMST³-Projektes, Villach 2006; verfügbar unter: http://imst.uni-klu.ac.at/materialien/2006/1145_337_Langfassung_Rohrer.pdf (17.01.07)
- [Sc05] Schöndorfer C (Betr.): Materialien zu Datenbanken im Rahmen des Lehrganges „Angewandte Informatik“ der Initiative „e-teaching austria“. Onlineversion 2005; verfügbar unter: <http://www.e-teaching-austria.at/AINF/> (17.01.2007)
- [Sc93] Schwill A.: Fundamentale Ideen der Informatik. In: Zentralblatt für Didaktik der Mathematik 1, 1993; S. 20-31
- [Sp02] Spitzer M.: Lernen. Gehirnforschung und die Schule des Lebens. Spektrum, Heidelberg – Berlin 2002
- [SS04] Schubert S., Schwill A.: Didaktik der Informatik. Spektrum, Heidelberg – Berlin 2004