

Die 9 Klasse hatte die grundlegenden Mechanismen des Anlegens von Datenbanken und deren Benutzung zum Thema. Ein wichtiger Aspekt wurde bisher noch vernachlässigt, die **Modellierung** einer Datenbank, d.h. die Umsetzung einer gegebenen Situation in die Struktur einer Datenbank.

Werden alle Daten einfach in einer Tabelle gespeichert, so ergeben sich häufig zahlreiche Probleme. Wir betrachten das Beispiel aus dem Buch auf S. 143.

BESTELLUNG									
KdNr	Name	Vorname	Adresse	Telefon	ArtNr	Artikel	Preis	Anzahl	BDatum
...
123	Mueller	Max	Postweg 8	555180	3434	Hemd	22,90 €	1	12.07.20
123	Mueller	Max	Postweg 8	555180	2827	Jeans	69,95 €	1	12.07.20
123	Mueller	Max	Postweg 8	555180	6001	Rasierer	49,00 €	1	12.07.20
224	Nowak	Beate	Goldstr. 7	555788	3434	Hemd	22,90 €	2	13.07.20
224	Nowak	Beate	Goldstr. 7	555788	1007	Bluse	43,50 €	1	13.07.20
...

Hier wird ein Versandhandel modelliert, der seine Bestellungen in einer einzelnen Tabelle verwaltet. Auf Anblick kann man erkennen, dass viele Informationen unnötig mehrfach in der Datenbank gespeichert werden (Speicherplatzverschwendung!), z.B. der Name und die Adresse des Kunden, im Fachbegriff wird dies **Redundanz** genannt.

Es gibt jedoch auch weitere Probleme (auch Anomalien genannt), die durch Redundanzen entstehen, z.B. die **UPDATE-Anomalie**.

Wird einer oder mehrere Einträge einer Tabelle geändert, so spricht man von einem update (da der zugehörige SQL-Befehl ebenfalls UPDATE heißt!). Offensichtlich ist z.B. eine Namensänderung eines Kunden – beispielsweise durch Heirat – hier sehr problematisch, da jede einzelne seiner Bestellungen geändert werden muss, ansonsten ist die Datenbank **inkonsistent**. Das bedeutet, dass sie Widersprüche enthält.

Möchte man etwas aus einer Tabelle löschen benutzt man den SQL-Befehl **DELETE**. Möchte ein Kunde aus dem System gelöscht werden, so muss auch hier jede einzelne seiner Bestellungen gelöscht werden, um seine Daten vollständig zu entfernen. Dabei kann leicht etwas vergessen werden. Zusätzlich besteht die Gefahr auch die Informationen über einen Artikel vollständig zu löschen, wenn die Bestellung die einzige zu diesem Artikel war. Im zweiten Fall spricht man auch von der **DELETE-Anomalie**.

Angenommen, man möchte einen neuen Artikel zum Sortiment hinzufügen, so ist dies mit obigem Tabellenschema nicht ohne weiteres möglich. Zu jedem Artikel muss eine Bestellung existieren!

Auf den ersten Blick scheint es möglich, bei allen Werten einfach *Null*, also „nichts“ einzutragen, die bisher nicht bekannt sind. Dies wird jedoch durch den Primärschlüssel verhindert, der hier aus *KdNr*, *ArtNr* und *BDatum* besteht. Es muss also eine fiktive Bestellung zwangsweise angelegt werden, um den Artikel aufnehmen zu können, es kommt zu einer **INSERT-Anomalie**, also zu einer Anomalie bzw. Inkonsistenz beim Einfügen neuer Daten.

Zusammengefasst ist das obige Schema ein wunderbares Beispiel dafür, wie Tabellen **NICHT** angelegt werden sollten.