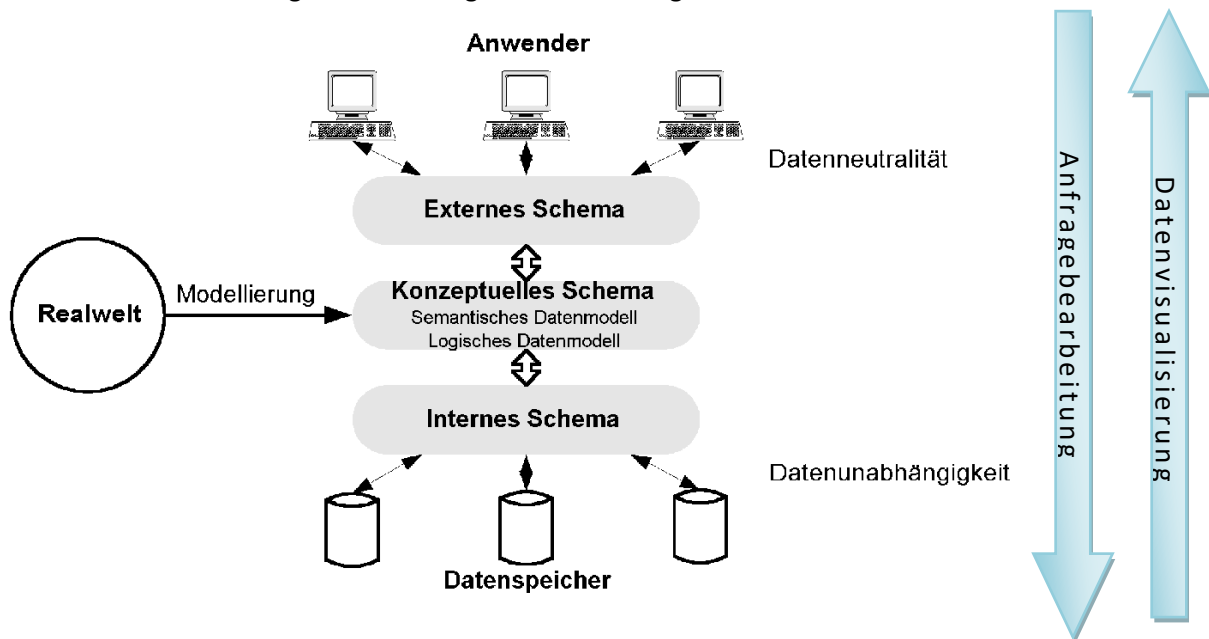


Datenbanken - Entwurf + Modellierung (ERM + RDM)

ANSI-SPARC-Architektur (ANSI-Architekturmodell)

Die **ANSI-SPARC-Architektur**, auch **Drei-Schema-Architektur**, beschreibt die grundlegende Trennung verschiedener Beschreibungsebenen für Datenbankschemata.

Die Architektur wurde 1975 vom Standards Planning and Requirements Committee (SPARC) des American National Standards Institute (ANSI) entwickelt und hat das Ziel den Benutzer einer Datenbank vor nachteiligen Auswirkungen von Änderungen in der Datenbankstruktur zu schützen.



Die drei Ebenen sind:

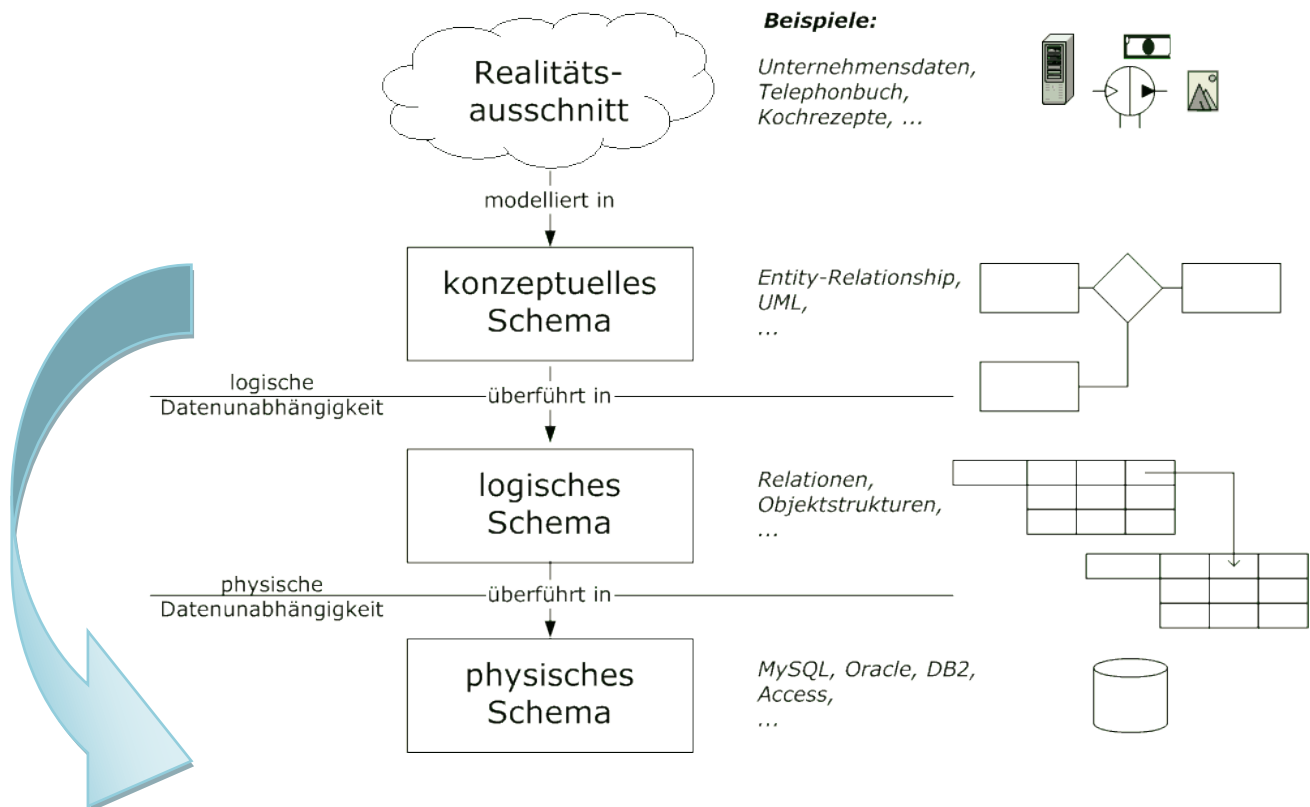
1. Die **externe Ebene**, die den Benutzern und Anwendungen **individuelle Benutzersichten** bereitstellt – Beispiele: Formulare, Masken-Layouts, Listen, Schnittstellen.
2. Die **konzeptionelle Ebene**, in der die in der Datenbank gespeicherten Daten sowie deren Beziehungen zueinander logisch formal beschrieben werden. Designziel ist hier eine vollständige und redundanzfreie Darstellung aller zu speichernden Informationen. Hier findet u.a. auch die Normalisierung (vgl. später) eines relationalen Datenbankschemas statt.
3. Die **interne Ebene** (physische Ebene), die die physische Sicht der Datenbank im Computer darstellt (Speicherort und -art). Designziel ist hier ein effizienter Zugriff auf die gespeicherten Informationen, wobei oft eine gewisse Redundanz in Kauf genommen werden muss.

Die **Vorteile** des 3-Ebenen-Modells liegen in der hohen Robustheit gegenüber Änderungen. Es besteht eine

- **physische Datenunabhängigkeit**, da die interne von der konzeptionellen und der externen Ebene getrennt ist. Physische Änderungen, z.B. des Speichermediums oder des Datenbankprodukts, wirken sich somit nicht auf die konzeptionelle oder externe Ebene aus.
- **logische Datenunabhängigkeit**, da die konzeptionelle und die externe Ebene getrennt sind. Dies bedeutet, dass Änderungen an der Datenbankstruktur (konzeptionelle Ebene) keine Auswirkungen auf die externe Ebene, also die Masken-Layouts, Listen und Schnittstellen haben.

Datenbankentwicklung

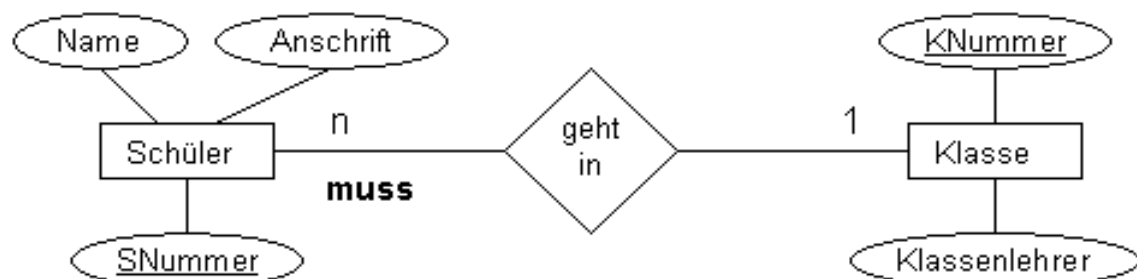
Der Entwurf einer Datenbank beschreibt den Prozess der Umsetzung einer Mini-Welt in ein Datenbankschema, das in der Lage ist, die gewünschten Daten dieser Welt mit ihren Eigenschaften und Beziehungen darzustellen. Er ist von zentraler Bedeutung bei der Entwicklung einer umfangreichen Anwendungssoftware, da diese Systeme zumeist viele Jahre betrieben werden und viele komplexe Herausforderungen zu bewältigen haben. Der Entwurf, an den sich dann die Implementierung anschließt, besteht im Wesentlichen aus den der Abbildung entnehmbaren Schritten:



Datenmodellierung (konzeptuell) – das Entity-Relationship-Modell (ERM)

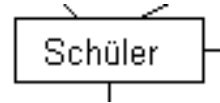
Die Modellierung dient der systematischen Darstellung einer Abstraktion der abzubildenden Miniwelt sowie der Kommunikation mit Nichtfachleuten, die nicht das Denken in Relationen und Attributen gewohnt sind (und zu denen u.a. der Auftraggeber eines großen Projektes gehören kann!)

Als eine Möglichkeit zur Modellierung (vor allem relationaler) Datenbanken hat das so genannte **Entity-Relationship-Modell (ERM)** weite Verbreitung erfahren. Es wurde 1976 von Peter Chen veröffentlicht und seitdem mehrfach weiterentwickelt. Zur Darstellung eines ERM werden E-R-Diagramme benutzt:

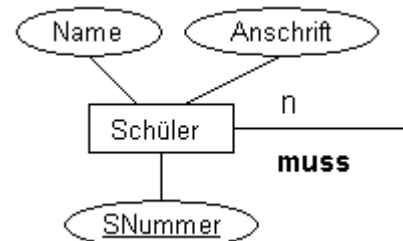


Die grundlegenden Begriffe in einem ERM seien im Folgenden erläutert:

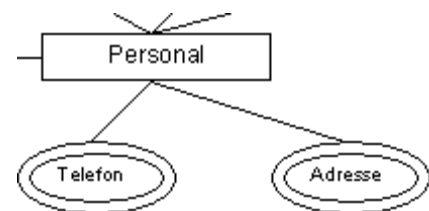
Unter einer **Entität** versteht man ein eindeutig identifizierbares Objekt der realen Welt, d.h. normalerweise eine Person, einen Prozess oder einen Gegenstand (z.B. ein Student, eine Lieferung, ein spezielles Buch). Gleichartige Entitäten bilden zusammen einen **Entitätstyp**, z.B. alle Studenten einer Universität oder alle Bücher einer Bibliothek. In einem E-R-Diagramm wird ein Entitätstyp durch ein Rechteck dargestellt, das den entsprechenden Namen umschließt.



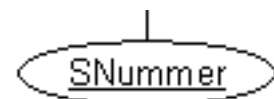
Eine Entität (bzw. ein Entitätstyp) wird durch eine Menge von **Attributen** beschrieben. In einem E-R-Diagramm umschließt eine Ellipse den Namen des Attributes und eine Linie verbindet es mit dem zugehörigen Entitätstyp. Ein Attribut von Student ist z.B. der Name. Man spricht hier speziell von einem **zusammengesetzten Attribut**, weil ihm wiederum verschiedene Attribute zugeordnet sind (Vor- und Nachname). In einem E-R-Diagramm werden zusammengesetzte Attribute genau wie atomare Attribute dargestellt – am besten wäre es aber in Anbetracht der späteren Weiterentwicklung der Datenbank, sie im E-R-Diagramm direkt zu zerlegen.



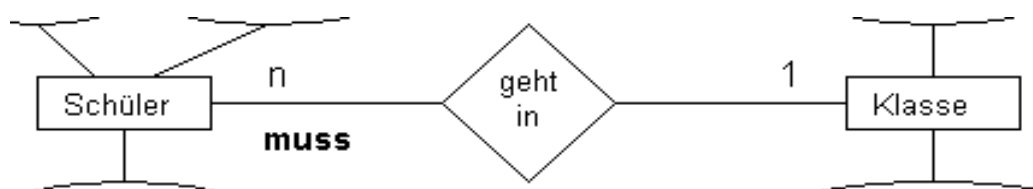
Mehrwertige Attribute sind Attribute, die als Wert eine Liste oder eine Menge annehmen können. In einem E-R-Diagramm werden diese durch den Attributnamen in einem Oval dargestellt. Meist ist es sinnvoll, mehrwertige Attribute durch die Einführung eines neuen Entitätstyps zu ersetzen.



Attributkombinationen, die zur eindeutigen Identifikation einer Entität dienen, werden **Schlüssel** genannt. Beim Entitätstyp Student könnte z.B. Name in Kombination mit einer zu speichernden Adresse als Schlüssel dienen. Unter allen möglichen Schlüsseln wird ein Schlüssel (eine Schlüsselkombination) ausgezeichnet, dieser heißt **Primärschlüssel**. In einem E-R-Diagramm werden die Namen aller Attribute, die zum Primärschlüssel gehören, unterstrichen dargestellt.



Durch **Beziehungen** werden im E-R-Modell Verknüpfungen, Zusammenhänge und Interaktionen zwischen Entitätstypen modelliert (z.B. Teil-von, arbeitet-für, ...). Im E-R-Diagramm werden Beziehungen durch eine Raute dargestellt, die den Namen der selbigen enthält und mit allen Entitätstypen verbunden ist, die an dieser Beziehung teilnehmen. Außerdem werden Zahlen an den Verbindungslinien notiert, die die **Kardinalität** (Anzahl möglicher beteiligter Entitäten) der Beziehung ausdrücken – im unteren Fall handelt es sich z.B. um eine „1:n-Beziehung“ „geht in“ zwischen den Entitäten Klasse und Schüler, was konkret bedeutet: eine Klasse kann mehrere Schüler beinhalten (n), wobei ein Schüler nur zu einer Klasse gehören kann (1). Beziehungen können im Übrigen (hier nicht zutreffend) – genauso wie Entitäten – Attribute besitzen.



Im Allgemeinen sind folgende Kardinalitätstypen denkbar:

Anzahl der Entitäten, die einer anderen Entität zugeordnet werden können	Symbol	Bezeichnung der Beziehung
genau eine	1	einfach
keine oder eine	c	einfach-bedingt (konditionell)
mindestens eine	n/ m	komplex (mehrfach)
keine, eine oder mehrere	nc/ mc	komplex-bedingt

(Spezialfälle von Beziehungen wie die Generalisierung oder die Aggregation müssen hier nicht näher behandelt werden, da sie stets in eine der obigen Kategorien einzuordnen sind.)

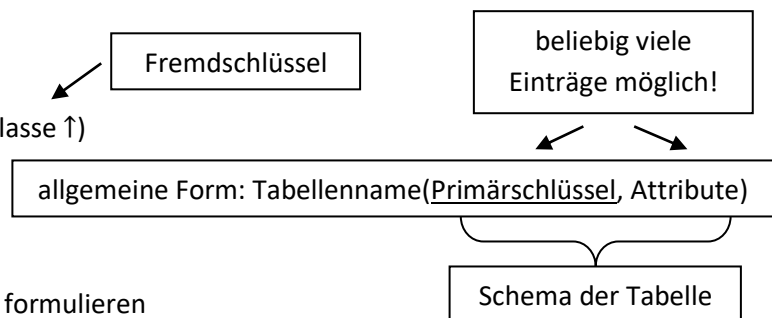
Datenmodellierung (logisch) – vom ERM zum relationalen Datenbankmodell (RDM)

Wir beschränken uns bei der Umwandlung unseres konzeptuellen Modells im Folgenden auf das 1970 von Codd entwickelte relationale Datenbankmodell, da die aktuelle Marktdominanz relationaler Datenbanken unumstritten ist. Im RDM sind die Daten stets in Form von Tabellen angeordnet – die genaue Schreibweise soll anhand des folgenden (obigen) Beispiels verdeutlicht werden soll:

Beispiel Schule:

Schüler(SNummer, Name, Anschrift, Klasse ↑)

Klasse(KNummer, Klassenlehrer)



Allgemein lassen sich folgende Regeln formulieren

(zur Überführung **vom ERM zum RDM**):

- **Entitätstypen** werden zu **Tabellen** (und bei Bedarf durch einen Primärschlüssel ergänzt)
- **Attribute** werden zu Attributen/ Spalten der zugehörigen Tabelle, die in ihrer Gesamtheit als **Schema** der Tabelle bezeichnet werden
- für **Beziehungen** bedarf das Vorgehen einer genaueren Analyse im konkreten Einzelfall:

Beziehung	Abbildungsregel für das RDM
1:1	Die beiden beteiligten Tabellen lassen sich zu einer einzelnen verschmelzen (wobei der Primärschlüssel einer der beiden Entitäten übernommen werden kann). Allerdings kann es auch Sinn machen, im Zusammenhang die Entitäten logisch voneinander getrennt zu belassen – dann ist wie im Fall 1:c zu verfahren.
1:c 1:n 1:nc	Die „c“/ „n“/ „nc“-Tabelle erhält ein zusätzliches Fremdschlüssel-Attribut, welches den betreffenden Primärschlüsselwert der referenzierten „1“-Tabelle annimmt.
(n)(c):(m)(c)	Es wird eine weitere Beziehungstabelle benötigt, die jeweils vorhandene Primärschlüssel als Fremdschlüssel übernimmt. Gleichzeitig bildet die Kombination aus Fremdschlüsseln den Primärschlüssel der neuen Tabelle.

Beispiele hierzu werden natürlich im Unterricht folgen ☺!