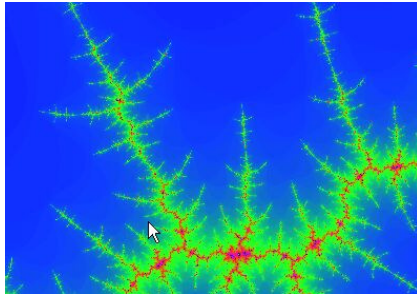


Datenbanken

4. Entity Relationship Modell (ERM)



Fachhochschule Darmstadt
Fachbereich Informatik

Dr. Peter Nevermann

Rückblick

■ Komplexe SQL Abfragen

- ◆ Unterabfragen in WHERE
- ◆ Verbünde
 - SQL2-Schreibweise mit JOIN
 - OUTER JOIN, NATURAL JOIN
- ◆ Unterabfragen in FROM

■ Datenmanipulation

- ◆ INSERT
- ◆ UPDATE
- ◆ DELETE

Problem

- **Finde ein geeignetes Datenbankschema für die Miniwelt "Firma"**

Miniwelt "Firma"

Abteilungen,
Projekte,
Mitarbeiter,
Gehalt,
Wochenstunde
nzahl,
Standorte, ...

Datenbank "Firma"

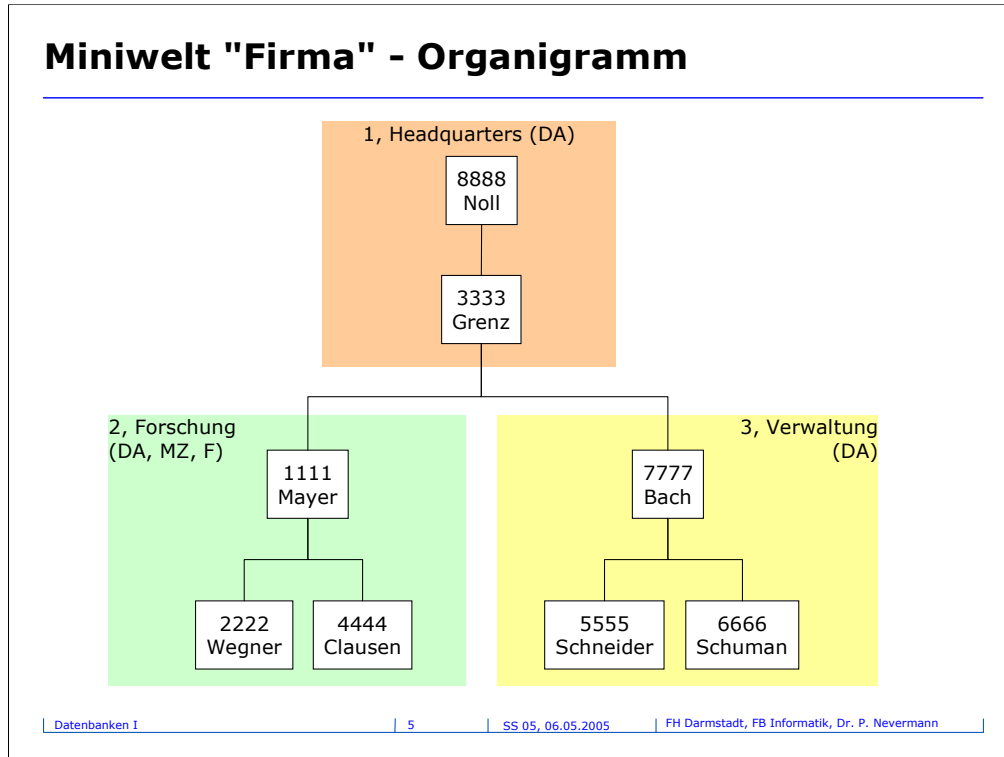
Datenbanken I | 3 | SS 05, 06.05.2005 | FH Darmstadt, FB Informatik, Dr. P. Nevermann

Wir wenden uns in diesem Kapitel dem Thema Datenbank-Entwurf zu. Vorgegeben ist eine Miniwelt, also ein Ausschnitt aus der realen Welt, der in einer Datenbank abgebildet werden soll. Zu dieser Miniwelt gilt es ein optimales Datenbankschema zu finden.

[Die hier verwendete Beispieldatenbank orientiert sich an Beispielen aus: R. Elmasri – S. Navathe, *Fundamentals of Database Systems*, Addison Wesley, 2003]

Im folgenden wird zunächst die Miniwelt "Firma" genau beschrieben.

- Die Firma gliedert sich in Abteilungen
 - Jede Abteilung hat eine eindeutige Nummer und einen eindeutigen Namen
 - Ein bestimmter Mitarbeiter hat die Position des Abteilungsleiters, und es ist von Interesse, seit wann er diese Position inne hat
 - Eine Abteilung kann sich auf mehrere Standorte verteilen
- Eine Abteilung kontrolliert mehrere Projekte
 - Jedes Projekt hat eine eindeutige Nummer und einen eindeutigen Namen
 - Ein Projekt ist an einen Ort gebunden (was nicht unbedingt ein Standort der Abteilung sein muß)
- Die Firma hat natürlich auch Mitarbeiter
 - Jeder Mitarbeiter hat eine eindeutige Personalnummer
 - Folgende Daten eines Mitarbeiters sind von Interesse: Name, Adresse, Gehalt, Geschlecht und Geburtsdatum
 - Ein Mitarbeiter gehört genau einer Abteilung an und hat genau einen anderen Mitarbeiter als Vorgesetzten
 - Ein Mitarbeiter kann an mehreren Projekten arbeiten, die nicht unbedingt alle zur selben Abteilung gehören müssen (oder zur Abteilung des Mitarbeiters)
 - Von Interesse ist außerdem noch die Anzahl von Stunden pro Woche, die ein Mitarbeiter einem Projekt zugeordnet ist (z.B. kann man damit kontrollieren, daß die Gesamtzahl an Projektstunden pro Woche und Mitarbeiter 40 nicht überschreitet)



Die Beschreibung konkreter Instanzen gehört nicht notwendig zu einer Miniwelt-Beschreibung dazu. Hier soll – zur Untermauerung der auf der vorigen Seite gegebenen Miniwelt-Beschreibung - lediglich ein konkreter Zustand der Miniwelt dargestellt werden.

Das abgebildete Organigramm zeigt die 3 Abteilungen mit den zugeordneten Mitarbeitern. In Klammern sind jeweils die Standorte der Abteilungen angegeben. In der Hierarchie ist die Mitarbeiter/Vorgesetzten-Beziehung abgebildet.

Miniwelt "Firma" - Projekte

	11 Pluto (F)	22 Merkur (MZ)	33 Venus (DA)	44 Mars (F)	55 Jupiter (MZ)	66 Saturn (DA)
1111		10		10	10	10
2222		10	10		10	
3333		10		10		20
4444			20	20		
5555	40					
6666	40					
7777	20					20
8888						

Datenbanken I

6




SS 05, 06.05.2005

FH Darmstadt, FB Informatik, Dr. P. Nevermann

Die Tabelle zeigt die Zuteilung (Anzahl Stunden pro Woche) der Mitarbeiter zu den Projekten. Die Gesamtstundenzahl pro Mitarbeiter und Woche soll 40 nicht überschreiten. Durch die Farbgebung der Spaltenüberschriften ist auch die Abteilungszugehörigkeit der Projekte zu erkennen: Projekt 11 → Abteilung 3, Projekte 22-55 → Abteilung 2, Projekt 66 → Abteilung 1.

Entity-Relationship Modell (ERM)

- **Erfinder: Peter P-S. Chen (1976)**

- **Modell-Elemente**
 - ◆ *Entity* (Entität, Datenspeicher) 
 - ◆ *Relationship* (Beziehung) 
 - ◆ *Attribute* (Eigenschaft, Attribut) 

- **ER-Diagramm**

Datenbanken I
7
SS 05, 06.05.2005
FH Darmstadt, FB Informatik, Dr. P. Nevermann

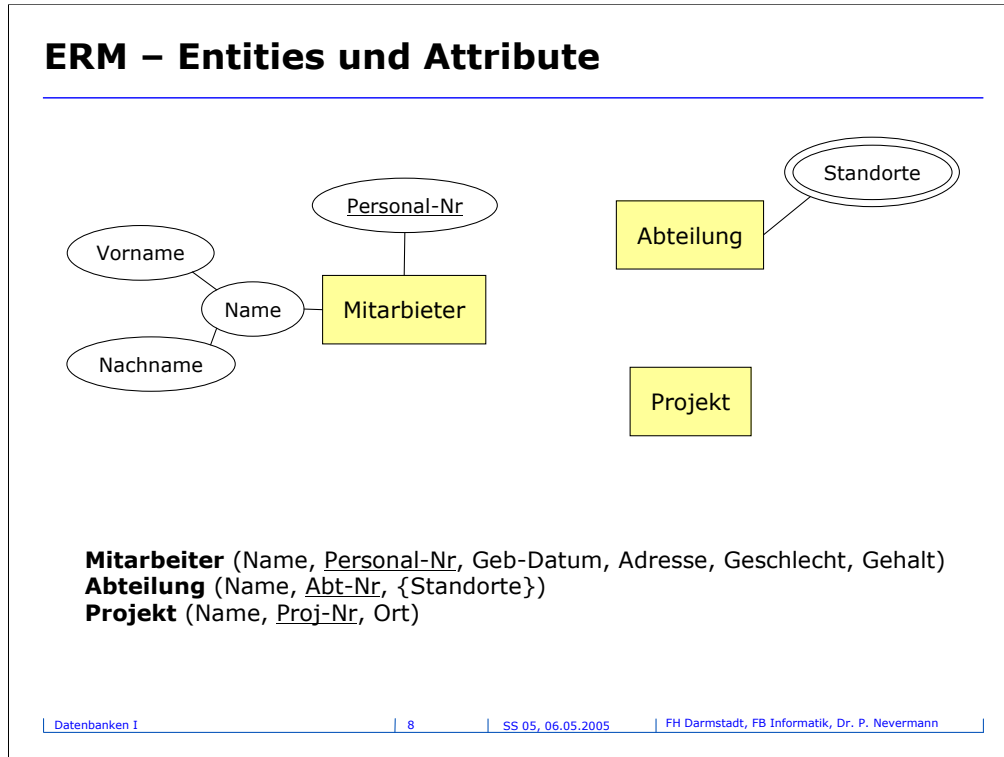
Das 1976 von P. Chen vorgestellte *Entity-Relationship Modell* (ER Modell oder ERM) hat sich bis heute als geeignetes Entwurfsmittel für Datenbanken gehalten und ist weit verbreitet. Es basiert i.W. auf 3 Elementen: *Entity*, *Relationship* und *Attribute*. Besonders hilfreich bei dieser Methode ist das ER Diagramm zur Visualisierung von Datenbankentwürfen.

Nutzen des ER-Modells

- Geeignetes Mittel um Miniwelten "einzufangen"
- Einfache Überführbarkeit in Datenbank-Strukturen
- Erleichtert die Kommunikation: IT-Spezialist ↔ Fachspezialist (Kenner der Miniwelt)

Die Originalarbeit von Chen zum ER-Modell findet man hier:

Chen, Peter P-S. (1976). The Entity-Relationship Model--Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1(1), 9-36.
 (<http://portal.acm.org/citation.cfm?id=320440&coll=GUIDE&dl=ACM&CFID=29527946&CFTOKEN=48861036>)



Was ist ein Entity?

- Personen: Mitarbeiter, Kunde, Mitglied, Student, Patient, Angehöriger
- Gegenstände: KFZ, Buch, Aktie, Immobilie, Artikel, Firma, Abteilung, Verein
- Ereignisse: Auftrag, Kurs, Straftat, Konzert, Projekt

Begriffe:

- Entity: eine konkrete Entität der Miniwelt, z.B. Mitarbeiter "Robert Bach", Abteilung "Verwaltung", Projekt mit der Nr. 55
- Entity-Typ (Schema, Begriffs-Intension): Menge von Eigenschaften (oder Merkmalen) durch die sich eine Klasse (oder Kategorie) von Entitäten beschreiben läßt, z.B. Mitarbeiter, charakterisiert durch die Merkmale {Name, Adresse, Geschlecht, Personal-Nr, ...}
- Entity-Menge (Begriffs-Extension): Menge derzeit in der Miniwelt vorkommender Entitäten eines Typs, z.B. Abteilung {1, 2, 3}, Projekt {11, ..., 66}

In dieser Vorlesung werden wir etwas lax "Entity" sagen, wenn Entity-Typ gemeint ist, und "Entity-Instanz" oder "Entity-Ausprägung" wenn von einem konkreten Entity die Rede ist.

Entities (also korrekt: Entity-Typen) werden im ER Diagramm mit einem Rechteck dargestellt. In der verbalen Beschreibung der Miniwelt erscheinen Entities in der Regel als Substantive.

Attribut

Attribute sind die Eigenschaften (oder Merkmale) mit denen Entities beschrieben werden:

- bei Personen: Name, Geburtsdatum, Adresse, Mitgliedsnummer, Personal-Nr
- bei Gegenständen: Farbe, Größe, Artikelnummer, ISBN-Nummer
- bei Ereignissen: Datum, Dauer, Ort

Attribute können im ER Diagramm als Ovale dargestellt werden. Allerdings wird darauf häufig verzichtet, um das Diagramm nicht zu überladen. Stattdessen werden die Entities mit ihren Attributen separat in der Form *Entity-1 (Attribut-11, Attribut-12, ...), Entity-2 (...), ...* gelistet.

Gegliedertes Attribut, mehrwertiges Attribut, einfaches und komplexes Attribut, Wertebereich

Attribute können *atomar* oder *gegliedert* sein. Eine Komponente einer Attribut-Gliederung kann selbst wieder gegliedert sein. Schreibweise für gegliederte Attribute: *Attribut-1 (Komponente-1, Komponente-2, ...)*.

Beispiele

- Atomar: Geschlecht oder Gehalt
- Gegliedert: Name (Vorname, Nachname) [d.h. Name setzt sich aus Vor- und Nachname zusammen] oder Adresse (Straße, Haus-Nr, PLZ, Ort)

Attribute können *ein-* oder *mehrwertig* sein. Schreibweise für mehrwertige Attribute: *{Attribut-1}*.

- Einwertig: Geb-Datum, Geschlecht
- Mehrwertig: {Standorte} [*Eine Abteilung kann mehrere Standorte haben.*]

Im ER Diagramm können multiple Attribute durch doppelt-umrandete Ovale dargestellt werden.

Gegliederte Attribute können selbst mehrwertig sein, und können gegliederte oder mehrwertige Attribute als Komponente haben. Man bezeichnet ein Attribut, das gegliedert oder mehrwertig ist, oder sich beliebig tief geschachtelt aus solchen zusammensetzt, auch als *komplexes Attribut*. Ein *einfaches Attribut* ist ein einwertiges, nicht-gegliedertes Attribut.

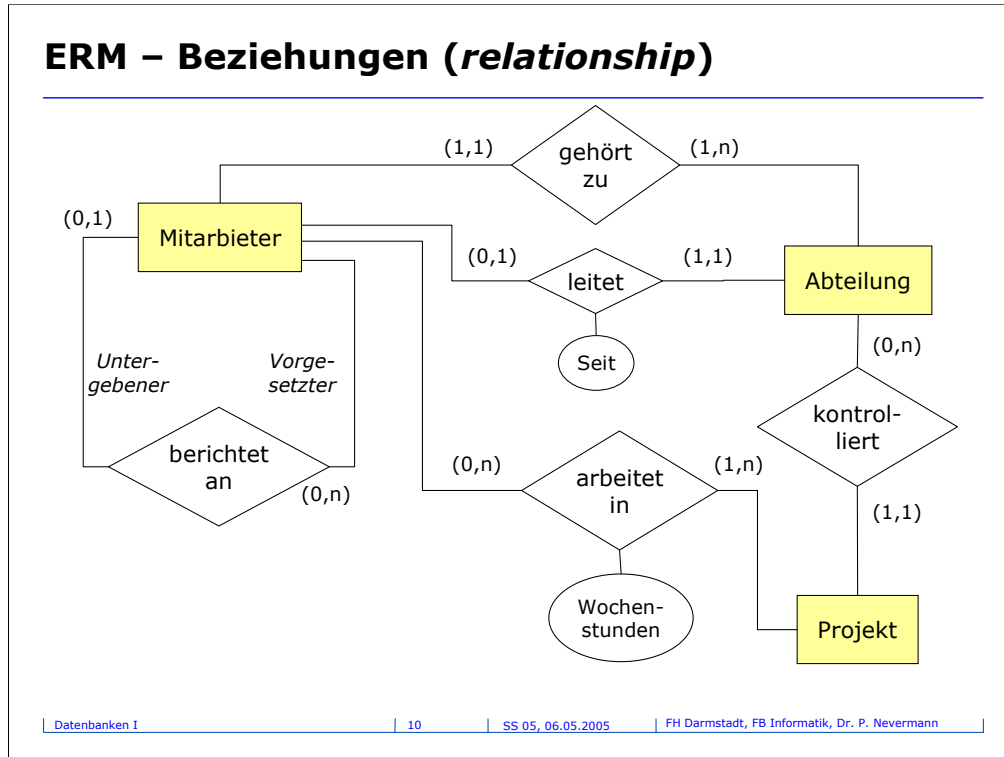
Attribute haben einen Wertebereich (*domain*). Eine konkrete Entity-Instanz hat zu jedem definierten Attribut keinen (*null*), einen oder viele Werte aus dem entsprechenden Wertebereich.

Schlüssel, Primärschlüssel

Ein Attribut eines Entities, welches niemals den gleichen Wert für unterschiedliche Entity-Instanzen haben kann, nennt man ein Schlüssel (*key*). Ein Schlüssel verleiht einem Entity *Identität*. Beispiele: Personal-Nr, ISBN-Nr,

Ein Schlüssel muß nicht aus einem einzigen Attribut bestehen, sondern kann sich aus mehrere Attributen zusammensetzen. Dann haben unterschiedliche Entity-Instanzen immer unterschiedliche Wertekombinationen in den beteiligten Attributen. Gibt es mehrere Kandidaten für einen Schlüssel, kann man einen dieser Kandidaten als Primärschlüssel (*primary key*) wählen. Schreibweise: Attribute, welche den Primärschlüssel bilden, werden unterstrichen.

Entities ohne Schlüssel nennt man Schwache Entitäten (*weak entity*) [kommt später!].

**Beziehung (*Relationship*):**

Neben den Eigenschaften (Attribute) eines Entities werden im ER Modell auch die Beziehungen (Relationship) zwischen den Entities beschrieben:

- Mitarbeiter *gehört zu* Abteilung
- Mitarbeiter *leitet* Abteilung
- Abteilung *kontrolliert* Projekt

Auch bei Beziehungen kann man zwischen einer konkreten Beziehung, einem Beziehungstyp und einer Beziehungs-Menge unterscheiden.

Wieder werden wir etwas lax Beziehung sagen, wenn Beziehungstyp gemeint ist, und Beziehungs-Instanz oder Beziehungs-Ausprägung, wenn von einer konkreten Beziehung die Rede ist.

Beziehungen (also korrekt: Beziehungstypen) werden im ER Diagramm mit einer Raute dargestellt. Üblicherweise haben ER-Diagramme eine "Leserichtung" und zwar von links nach rechts und von oben nach unten.

In der verbalen Beschreibung der Miniwelt erscheinen Beziehungen in der Regel als Verben.

Attribute von Beziehungen:

Wie Entities können auch Beziehungen Attribute haben. Diese sind häufig zeitbezogen, z.B. Datum, seit dem ein Mitarbeiter eine Abteilung leitet, Anzahl Stunden die ein Mitarbeiter pro Woche einem Projekt zugeteilt ist.

Stelligkeit:

Die Stelligkeit einer Beziehung gibt die Anzahl der beteiligten Entity-Instanzen an einer Beziehungs-Instanz an. Im ER-Diagramm der Folie sind ausschließlich 2-stellige Beziehungen zu sehen, welche auch "binär" (*binary relationship*) genannt werden. Beziehungen mit einer Stelligkeit > 2 sind komplex und deshalb selten.

Rollen:

Die an einer Beziehung beteiligten Entities spielen eine bestimmte Rolle in der Beziehung, welche im ER-Diagramm dokumentiert werden kann. Das macht besonders dann Sinn, wenn die Beziehung Instanzen eines einzigen Entities in Beziehung setzt (z.B. "berichtet an" im Diagramm):

- Mitarbeiter leitet Abteilung → Rollen: Abteilungsleiter → geleitete Abteilung
- Mitarbeiter berichtet an Mitarbeiter → Untergebener → Vorgesetzter

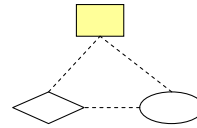
Kardinalitäten:

Für jedes an einer Beziehung beteiligte Entity wird festgelegt, wie häufig eine Entity-Instanz an einer Beziehungs-Instanz teilnehmen kann. Das wird anhand eines Paares (*min, max*) gemacht:

- (0,1) – Eine Instanz kommt höchstens einmal in einer Beziehung vor
- (1,1) – Eine Instanz kommt genau einmal in einer Beziehung vor
- (0,n) – Eine Instanz kommt beliebig häufig in einer Beziehung vor
- (1,n) – Eine Instanz kommt mindestens einmal in einer Beziehung vor

Entwurfsrichtlinien

- Entity Namen [*Singular statt Plural*]
- Gegliederte Attribute
- Entity oder Attribut?
- Attribut oder Beziehung?
- Beziehung oder Entity?
- ER-Modellierung als iterativer Prozeß



Entity Namen

Es ist üblich, Entity Namen im Singular zu wählen, also "Abteilung" statt "Abteilungen".

Gegliederte Attribute

Ob Attribute in kleinere Komponenten gegliedert werden (z.B. Name in (Vorname, Nachname), Adresse in (Straße, Haus-Nr, PLZ, Ort) hängt von den Anforderungen aus der Miniwelt ab: werden die einzelnen Komponenten des Attributs benötigt, oder wird das Attribut immer nur als Ganzes behandelt?

Entity oder Attribut?

Attribute, welche in mehreren Entities vorkommen, können evtl. in ein eigenes Entity herausgezogen werden.

Beispiel: die Attribute "{Standorte}" von Abteilung und "Ort" von Projekt könnten durch ein Entity "Ort" und Beziehungen "hat als Standort" von Abteilung zu Ort, bzw. "findet statt in" von Projekt zu Ort. Manchmal liefern auch gegliederte Attribute Kandidaten für ein zusätzliches Entity ab

Umgekehrt könnte ein initial existierendes Entity "Geschlecht" mit einer Beziehung "hat" von Mitarbeiter zu Geschlecht durch ein Attribut "Geschlecht" bei Mitarbeiter ersetzt worden sein, weil das Entity "Geschlecht" keine Existenzberechtigung hatte (nur ein Attribut und nur zwei mögliche Instanzen).

Attribut oder Beziehung?

Sachverhalte, welche als Beziehung darstellbar sind, sollten möglichst nicht als Attribut dargestellt werden. Wie wir später sehen werden, gehen Beziehungen bei der Überführung eines ER-Modells in ein Datenbankschema häufig in Attribute einer der beteiligten Relationen über. Diesen Schritt sollte man bei der ER-Modellierung nicht vorwegnehmen, um sich später noch alle Möglichkeiten offen zu lassen.

Beziehung oder Entity?

Häufig kann ein Sachverhalt sowohl als Entity als auch als Relationship betrachtet werden. Hilfreich kann dabei folgende Frage sein:

Hat dieses etwas eine ausgeprägte Identität? [ja → Entity, nein → Beziehung].

Beispiel:

- Ehe als Beziehung: Person → (0,1) "verheiratet mit" (0,1) → Person
- Ehe als Entity (Identität: Trauschein-Nr): Person → (0,1) "ist Partner in" (2,2) → Ehe

ER-Modellierung als iterativer Prozeß

ER-Modellierung ist i.d.R. ein iterativer Prozeß, bei dem das Modell durch wiederholte Rücksprache mit den "Miniwelt-Kennern" (Fachabteilung) verfeinert und korrigiert wird.

Aufgabe:

Machen Sie ein ER Teilentwurf für das Kinocenter "CineCity" aus dem 1. Praktikum.

Mit freundlicher Genehmigung von Microsoft Press Deutschland, stammt dieses Beispiel aus dem Buch:

Ralf Albrecht, Natascha Nicol: *Microsoft Office Access 2003 – Das Handbuch*, Microsoft Press Deutschland, 2004 (ISBN 3-86063-176-4)

Es handelt sich dabei um ein Kinocenters namens "CineCity", das auf mehreren Leinwänden Filme zeigt. In der Datenbank sollen die Spielpläne für die einzelnen Kinos verwaltet werden (d.h. welcher Film zu welcher Zeit läuft). Im Prinzip gibt es für jede Kinowoche (beginnend jeweils am Donnerstag) eine Liste von Filmen, die in der Woche laufen sollen. Von Interesse sind folgende Informationen:

- Angaben zum Film (Filmtitel, Originaltitel, Filmlänge, die wievielte Woche der Film läuft - gerechnet ab Bundesstartdatum, Altersbeschränkung [FSK])
- Spielplan, d.h. Kinosäle, Wochentage und Uhrzeiten an denen der Film in der Woche gezeigt wird
- Eintrittspreise,
- Angaben zum Kinosaal (Anzahl Plätze, Turnaround [= Zeit die benötigt wird um den Saal zu räumen und mit dem Publikum des nächsten Films zu füllen]).

Ausblick

■ Erweitertes ER Modell

- ◆ Weak Entities
- ◆ Spezialisierung/Generalisierung