

# Datenbanksysteme SS 2011

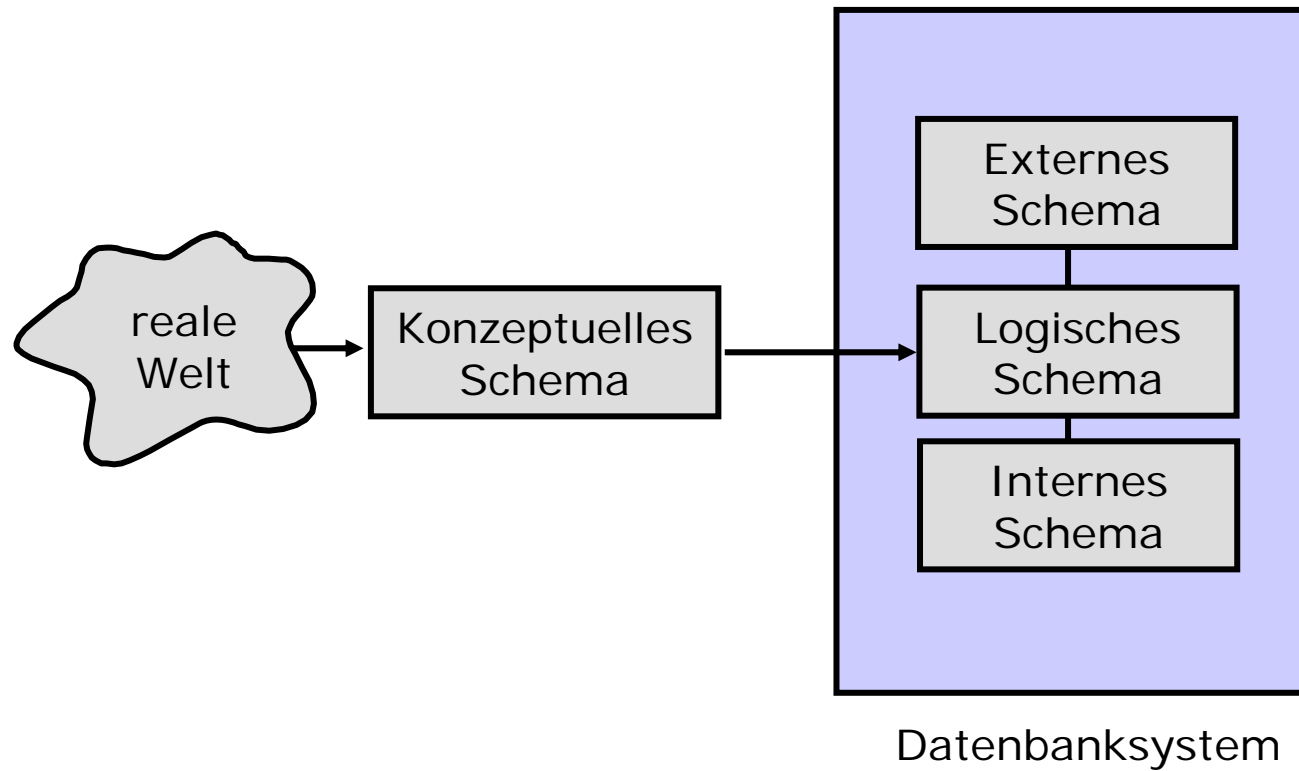
## Kapitel 3: Logische Datenmodelle

Oliver Vornberger

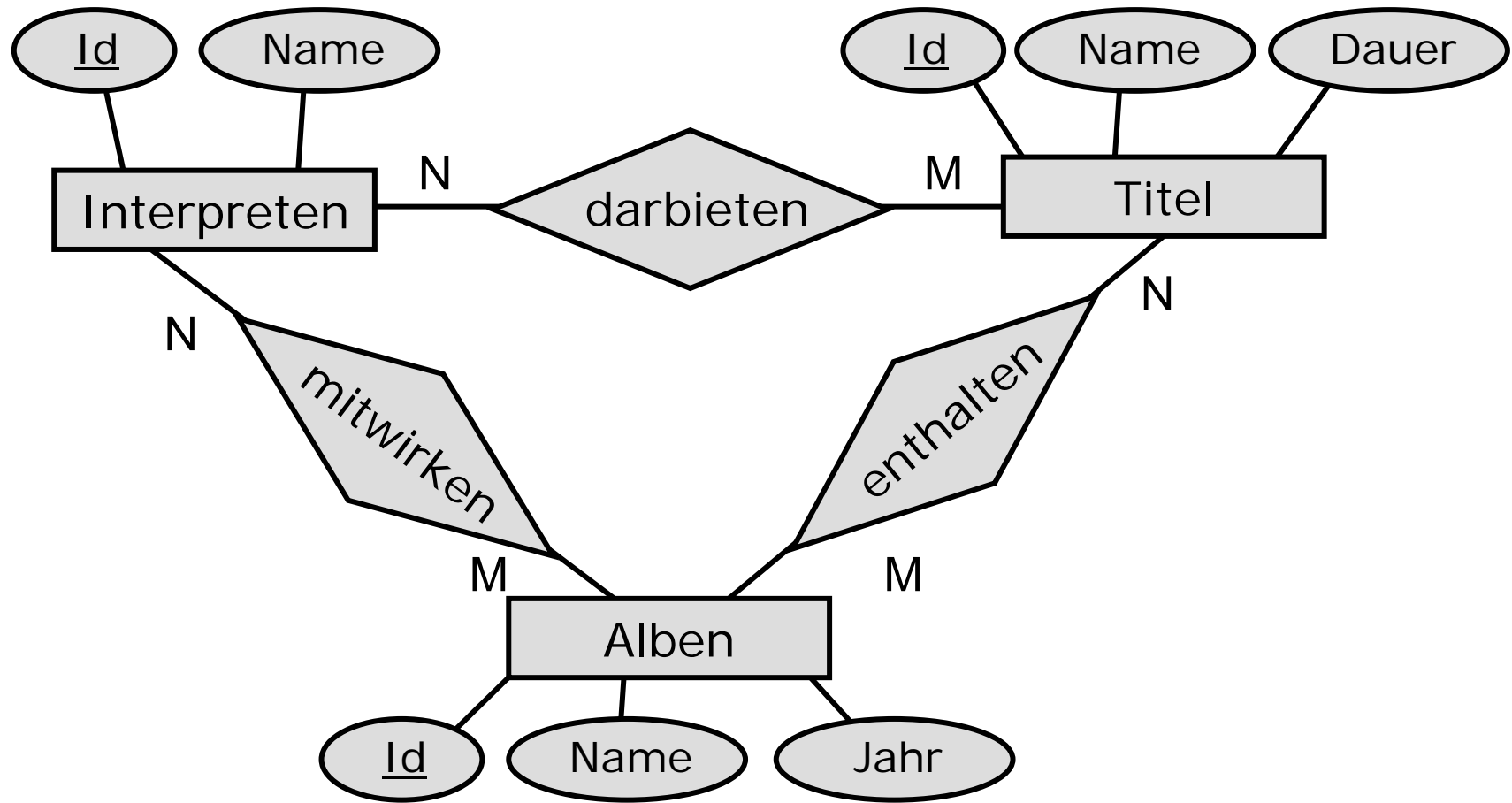
Institut für Informatik  
Universität Osnabrück

Vorlesung vom 18.04.2011

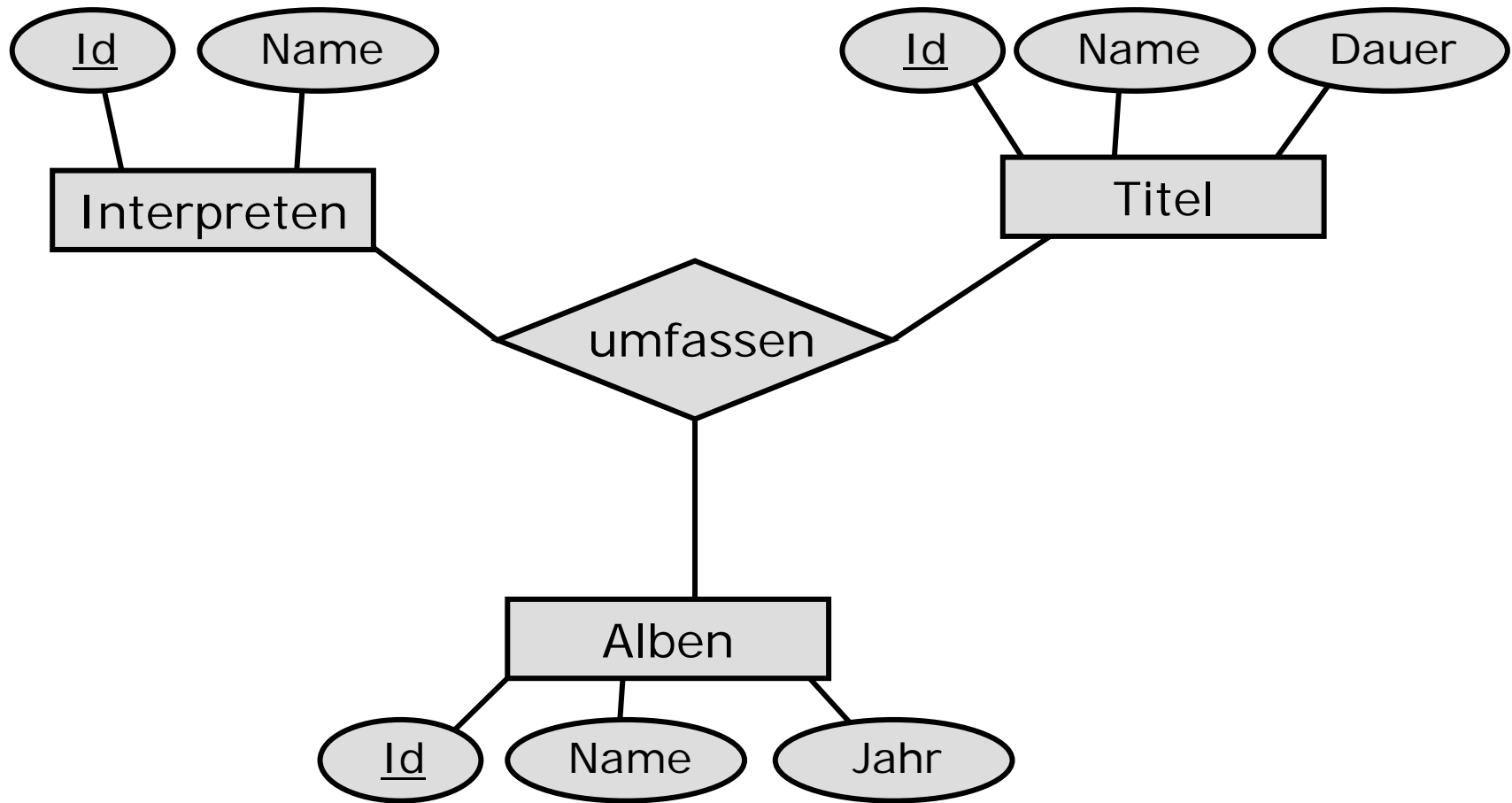
# Konzeptuelles Schema



# Entity Relationship Diagramm



# Entity Relationship Diagramm



# 2er versus 3er Relationships

Beatles	Abbey Road	Here comes the sun
Beatles	Abbey Road	Come together
Beatles	Yellow Submarine	All you need is love

Beatles	Abbey Road
Beatles	Yellow Submarine

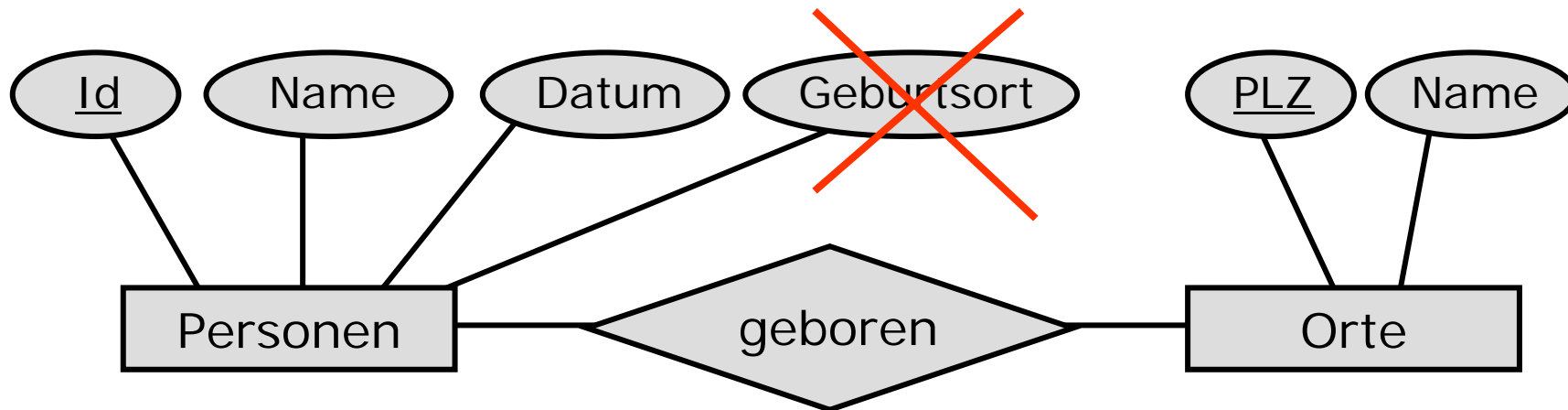
Abbey Road	Here comes the sun
Abbey Road	Come together
Yellow Submarine	All you need is love

Beatles	Here comes the sun
Beatles	Come together
Beatles	All you need is love

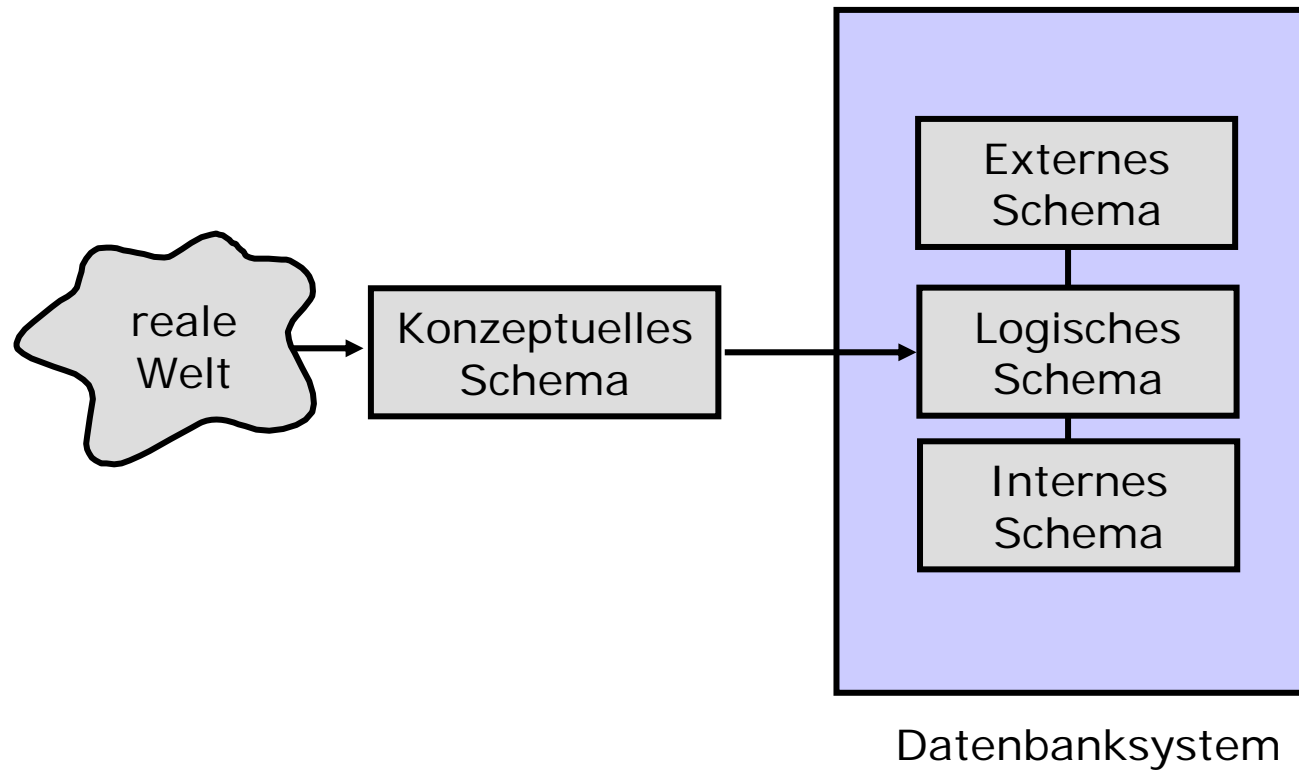
# Personalakte

Susi Sorglos, geboren am 4. Mai 1960 in Ibbenbüren

Willi Wacker, geboren am 12. August 1972 in Osnabrück



# Logische Datenmodelle



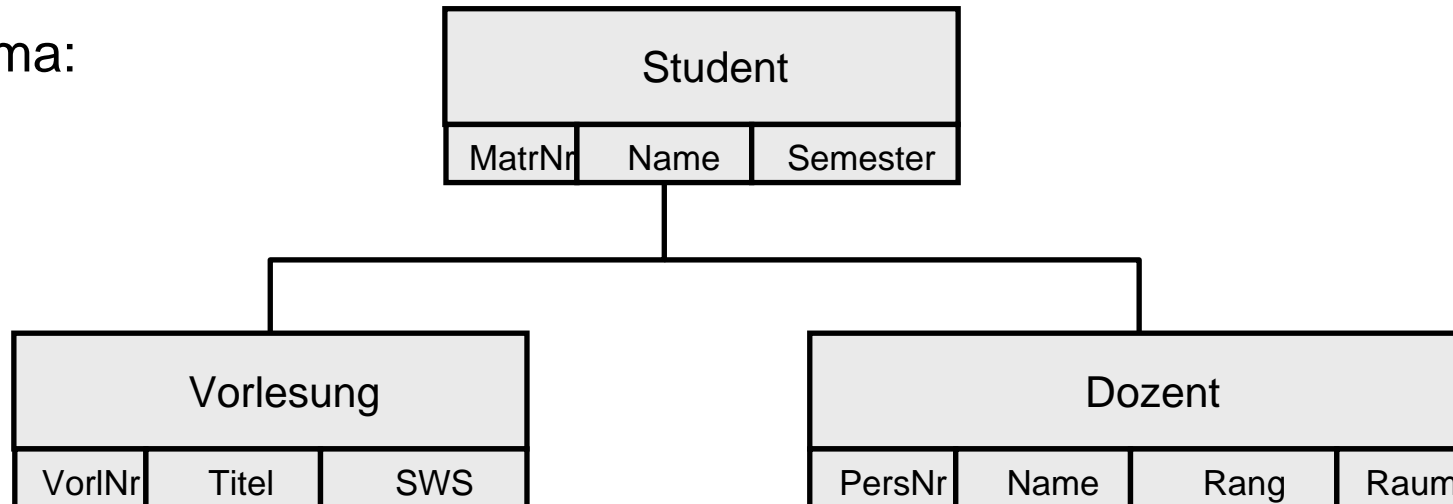
# Logische Datenmodelle

- Hierarchisches Modell (1960 IMS IBM)
- Netzwerkmodell (1970 UDS Siemens)
- Relationales Modell (1980 MS Access 1980)
- Objektorientiertes Modell (1990 O2)



# Hierarchisches Modell

Schema:



Ausprägung (S = Student, V = Vorlesung, D = Dozent):

S	Mustermann
---	------------

V	Stochastik
---	------------

V	Java
---	------

D	Stadje
---	--------

D	Vogt
---	------

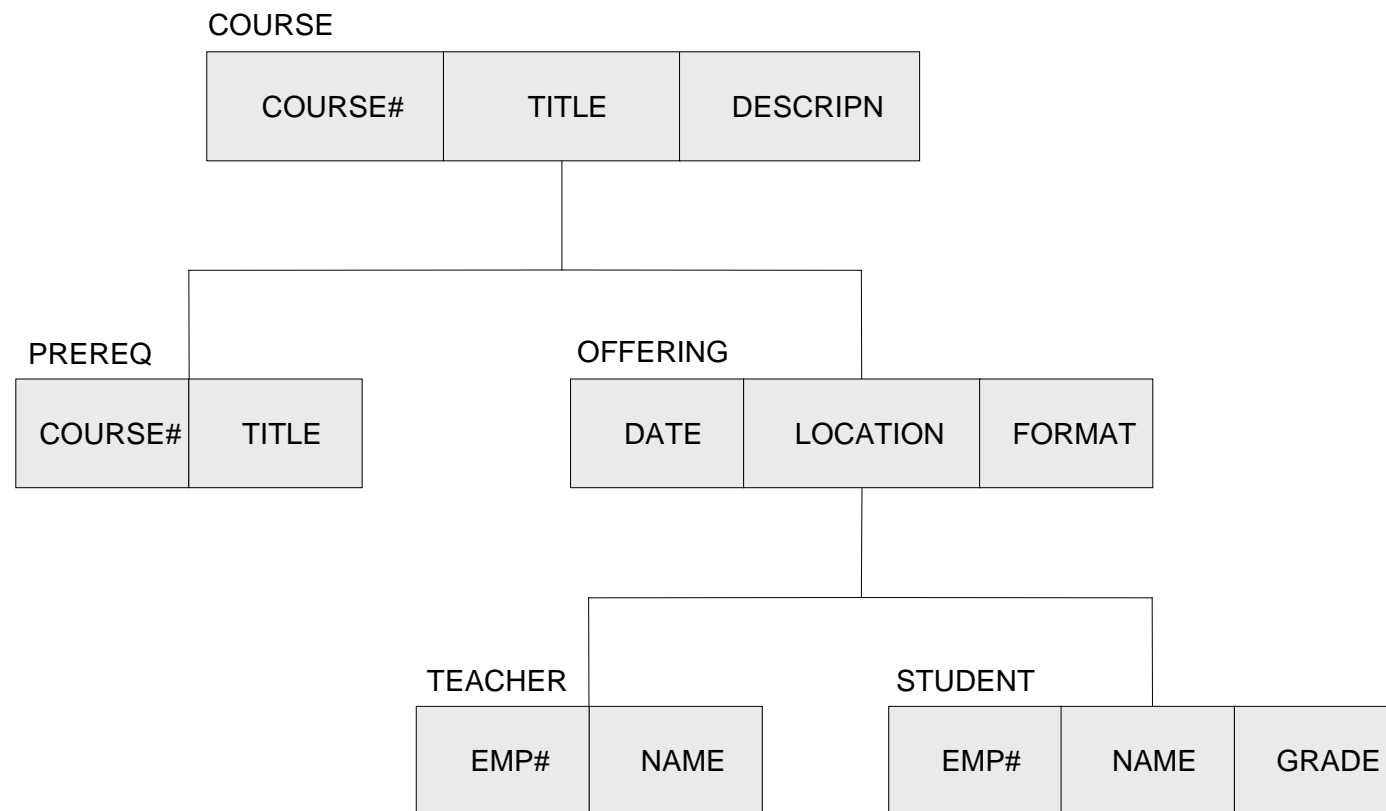
D	Schreiner
---	-----------

S	Wacker
---	--------

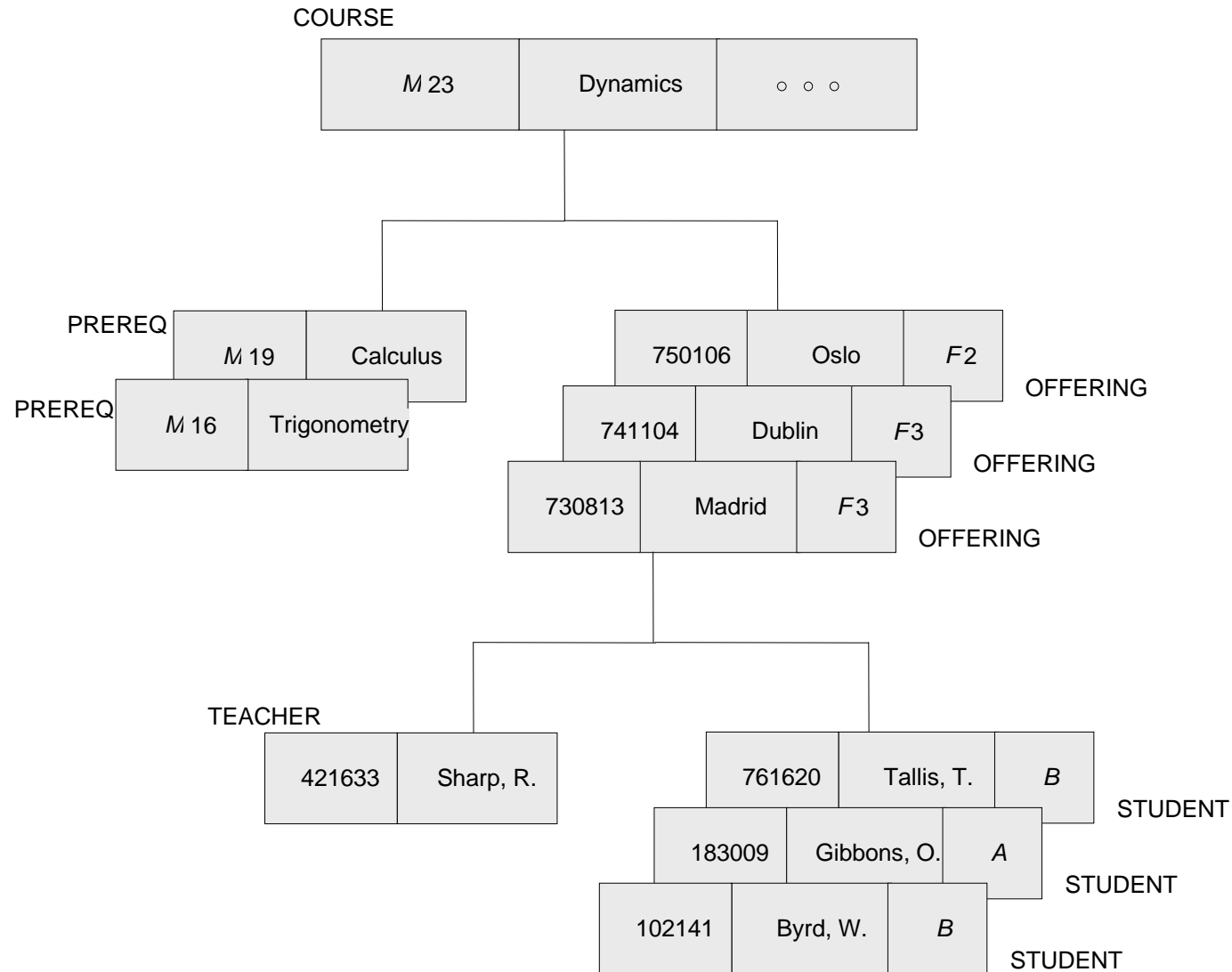
V	Zahlentheorie
---	---------------

D	Vogt
---	------

# Schema im Hierarchischen Modell



# Ausprägung im Hierarchischen Modell



# Navigation in Hierarchischen Modell

Randbedingung: auf Magnetband vorwärts suchen

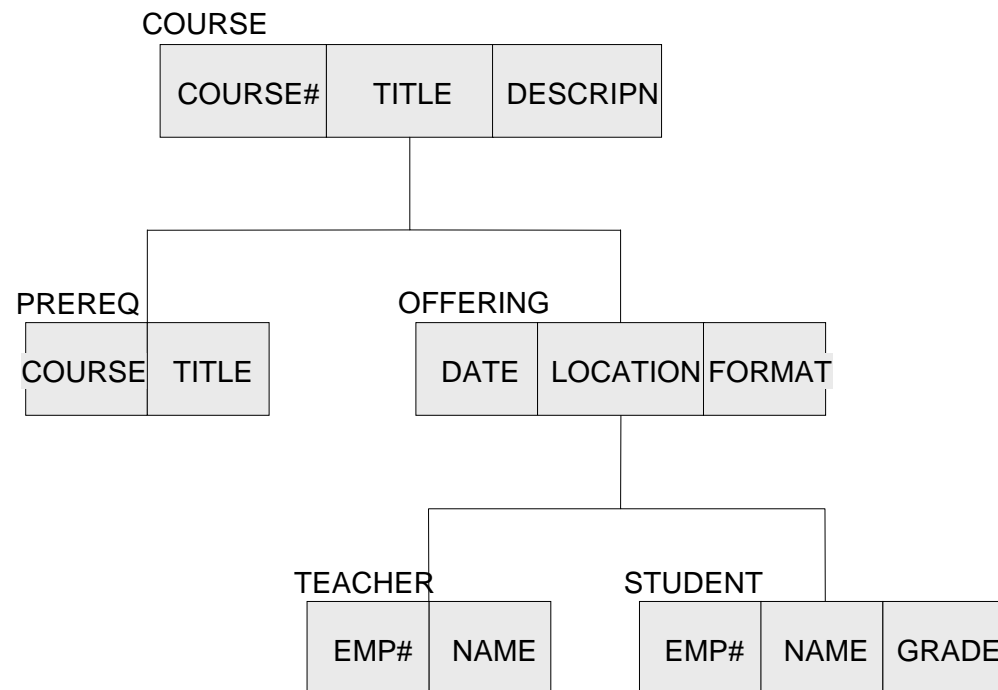
**GU**      **Get Unique <Tpy>**

**GN**      **Get Next <Typ>**

**GNP**     **Get Next <Typ> within Parent**

# Beispiel für Navigation

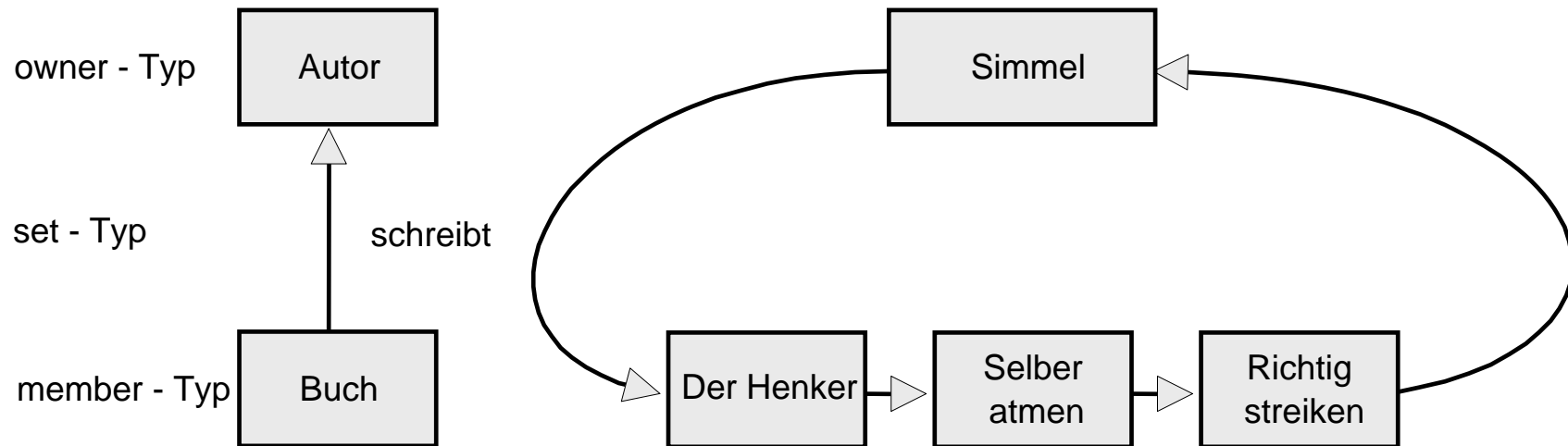
Welche Studenten sind im Kurs M23 am 13.08.1973 ?



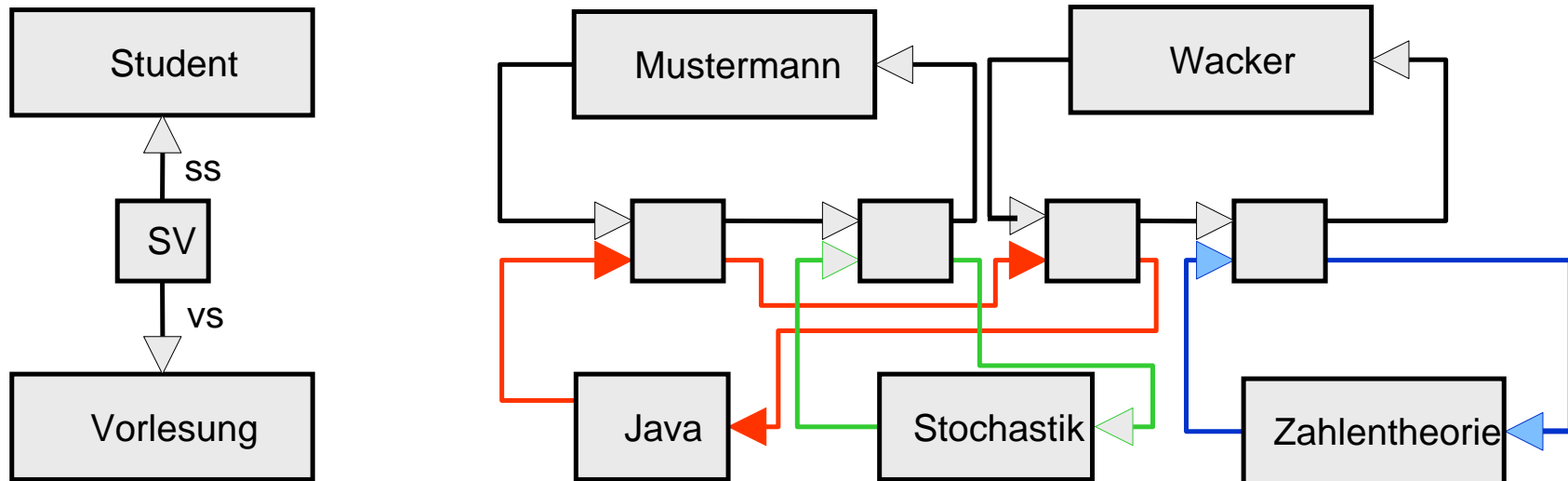
```
GU COURSE(COURSE#='M23')
    OFFERING (DATE='730813')
if gefunden then
begin
    GNP STUDENT
    while gefunden do
    begin
        write(STUDENT.NAME)
        GNP STUDENT
    end
end;
end;
```

# Netzwerkmodell

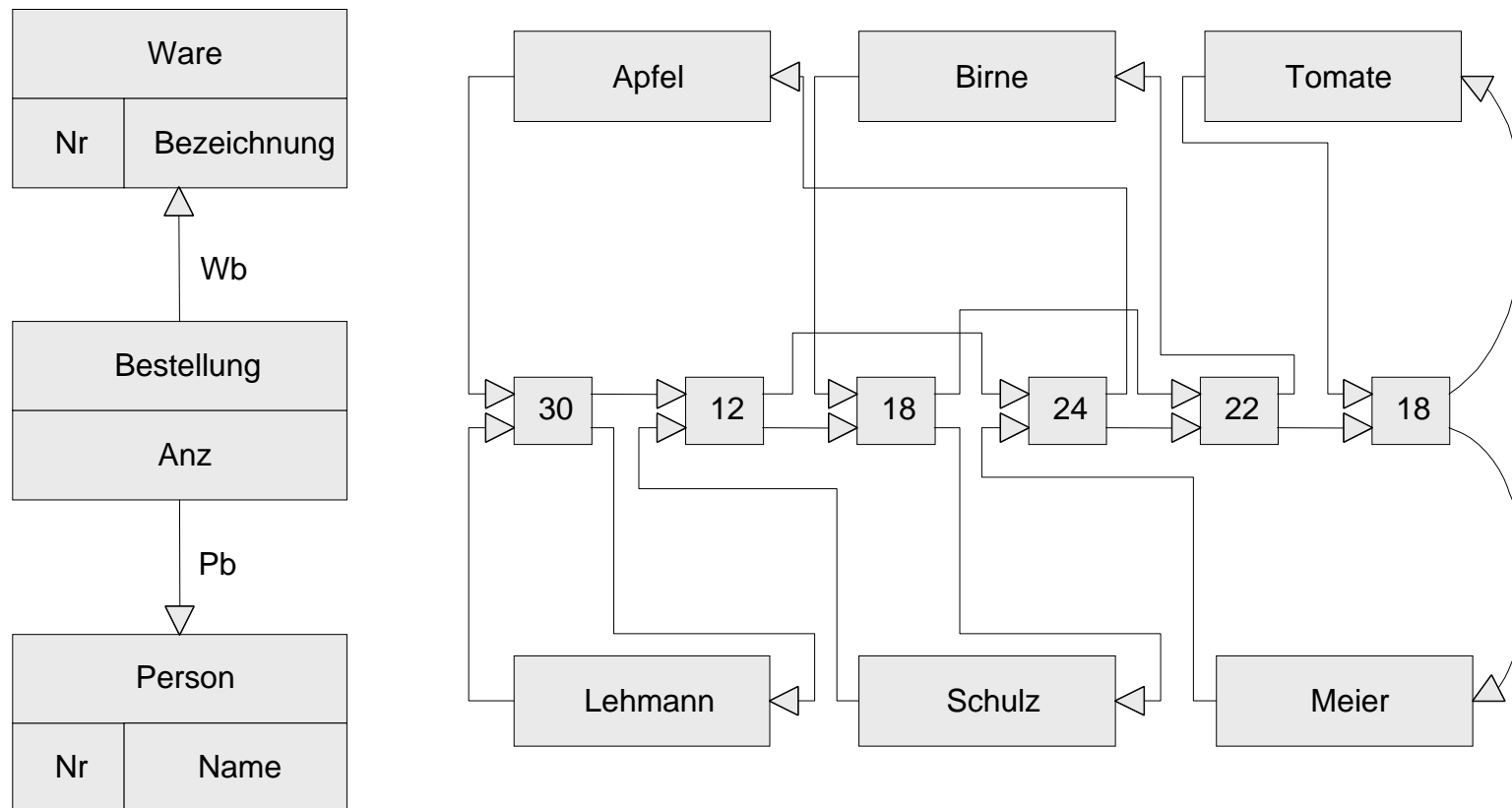
(nur binäre many-one-Beziehungen)



# Netzwerk mit M:N-Beziehung



# Kettrecord mit Attribut im Netzwerkmodell





# Operationen im Netzwerkmodell

Typische Operation :  
Navigation durch die verzeigerten Entities

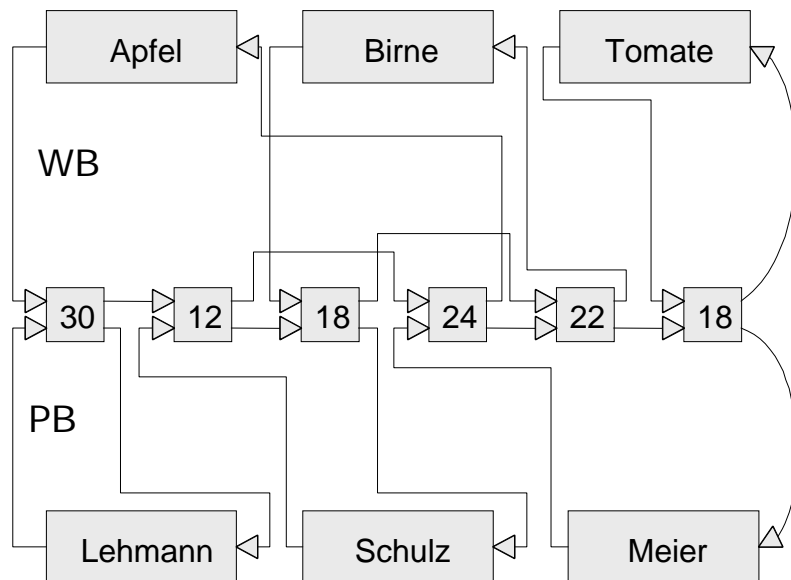
**FIND ANY <TYP> USING <ATTRIBUT>**

**FIND NEXT <TYP> WITHIN <SET>**

**FIND OWNER WITHIN <SET>**

# Operationen im Netzwerkmodell

Drucke Bestellungen  
von Schulz



```
PERSON.NAME := 'SCHULZ';  
FIND ANY PERSON USING NAME;  
IF GEFUNDEN THEN  
BEGIN  
    FIND FIRST BESTELLUNG WITHIN PB;  
    WHILE GEFUNDEN DO  
    BEGIN  
        FIND OWNER WITHIN WB;  
        GET WARE;  
        WRITE(WARE.BEZEICHNUNG);  
        FIND NEXT BESTELLUNG WITHIN PB;  
    END  
END;  
END;
```

# Relationales Modell

1970: Edgar Codd:  
"A relational model for large shared data banks"  
Communications of the ACM

1977: Lawrence Ellison gründet Oracle

1981: Turing Award an Edgar Codd

# Relationales Datenmodell

- pro Entity-Typ:  
Tabelle mit Spalten benannt nach den Attributen.
- pro Relationship-Typ:  
Tabelle mit Spalten für die Schlüssel der beteiligten Entity-Typen und ggf. weitere Spalten.

**Studenten**

MatNr	Vorname	Nachname
653467	Erika	Mustermann
875462	Willi	Wacker
432788	Peter	Pan

**hoeren**

MatrNr	VorlNr
653467	6.718
875462	6.718
432788	6.718
875462	6.108

**Vorlesungen**

VorlNr	Titel	Umfang
6.718	Java	4
6.174	Stochastik	2
6.108	Zahlentheorie	4

# Operationen im Relationalen Modell

**Studenten**

MatNr	Vorname	Nachname
653467	Erika	Mustermann
875462	Willi	Wacker
432788	Peter	Pan

**hoeren**

MatrNr	VorlNr
653467	6.718
875462	6.718
432788	6.718
875462	6.108

**Vorlesungen**

VorlNr	Titel	Umfang
6.718	Java	4
6.174	Stochastik	2
6.108	Zahlentheorie	4

- Selektion: Suche alle Tupel einer Relation mit gewissen Attributeigenschaften
- Projektion: filtere gewisse Spalten heraus
- Verbund: Finde Tupel in mehreren Relationen, die bzgl. gewisser Spalten übereinstimmen.

Welche Studenten hören die Vorlesung *Zahlentheorie*?

```
SELECT Studenten.Nachname
From Studenten, hoeren, Vorlesungen
WHERE Studenten.MatrNr = hoeren.MatrNr
AND hoeren.VorlNr = Vorlesungen.VorlNr
AND Vorlesungen.Titel = 'Zahlentheorie'
```

# Das Objektorientierte Datenmodell

- Eine Klasse repräsentiert einen Entity-Typ, charakterisiert durch Struktur und Verhalten.
- Struktur und Verhalten können an eine Unterklasse vererbt werden.
- Binäre Beziehungen werden durch mengenwertige Attribute modelliert.

# Objektorientierte Modellierung: Klassen

```
class Studenten {  
    attribute long Matrnr;  
    attribute String Name;  
    relationship set <Vorlesungen> hoert  
        inverse Vorlesungen::Hoerer;  
}
```

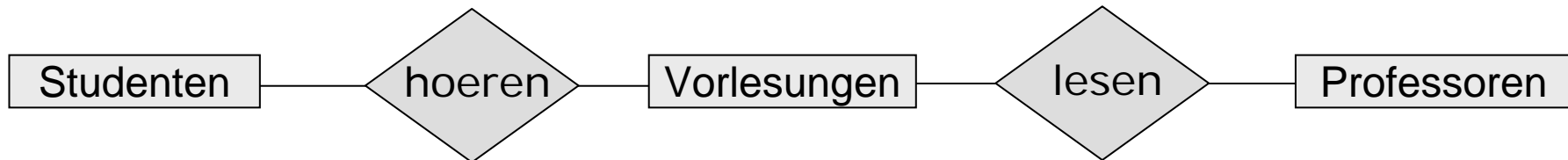
```
class Professoren {  
    attribute long PersNr;  
    attribute String Name;  
    relationship set <Vorlesungen> liest  
        inverse Vorlesungen::gelesenVon;  
}
```

```
class Vorlesungen {  
    attribute long VorlNr;  
    attribute String Titel;  
    relationship Professoren gelesenVon  
        inverse Professoren::liest;  
    relationship set <Studenten> Hoerer  
        inverse Studenten::hoert;  
}
```

# Objektorientierte Modellierung: Query

Welche Studenten besuchen Vorlesungen von Sokrates ?

```
select s.Name
from s in AlleStudenten, v in s.hoert
where v.gelesenVon.Name = "Sokrates"
```



SQL:

```
select Studenten.Name
from Studenten, hoeren, Vorlesungen, lesen, Professoren
where Studenten.MatrNr = hoeren.MatrNr
and hoeren.VorlNr = Vorlesungen.VorlNr
and Vorlesungen.VorlNr = lesen.VorlNr
and lesen.PersNr = Professoren.PersNr
and Professoren.Name = "Sokrates"
```