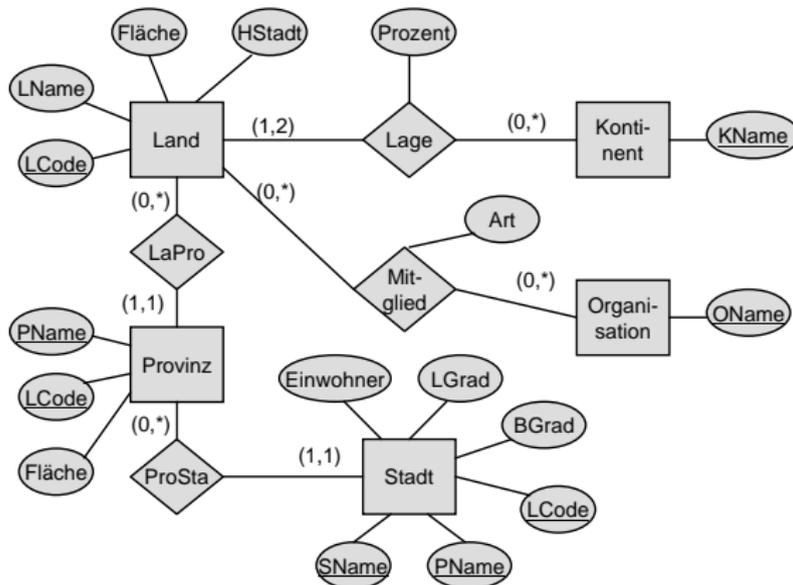


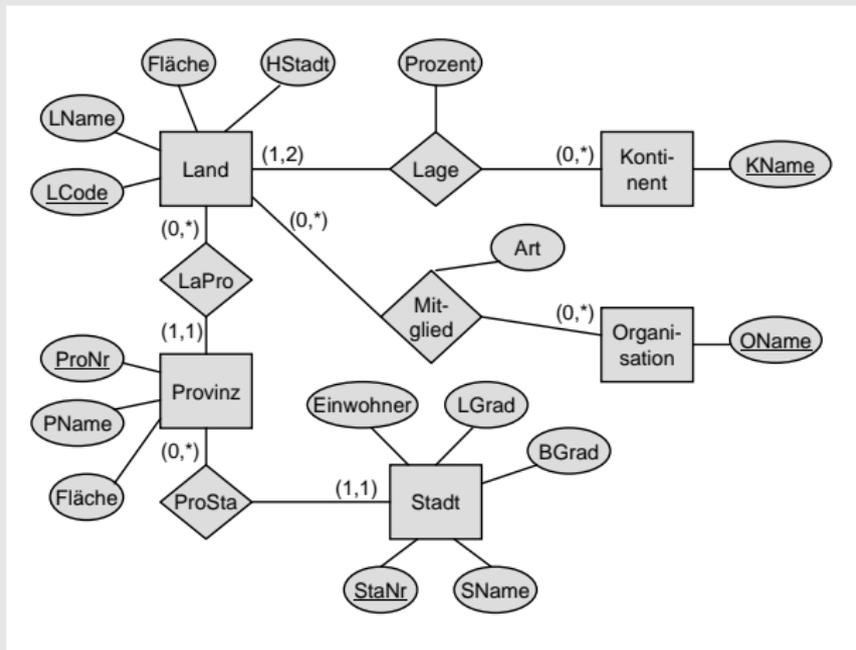
# Kapitel 4: Entwurf der Relationsschemata

- ▶ *konzeptueller Entwurf*: finde eine umfassende Strukturierung der gesamten Informationsanforderungen der Miniwelt. Resultat: *konzeptuelles Schema*, typischerweise ein ER-Schema, oder vergleichbares.
- ▶ *logischer Entwurf*: bilde die Zusammenhänge des konzeptuellen Schemas in Relationsschemata ab. Resultat: *logisches Schema*.  
Das logische Schema ist Teil des zum Einsatz kommenden (relationalen) Datenbanksystems.
- ▶ *physischer Entwurf*: definiere die Abspeicherung der Relationen auf den Speichermedien und lege Indexstrukturen zur Effizienzsteigerung der Anwendungen fest.

## Mondial ER-Schema (M1)



## Mondial ER-Schema alternativ (M2)



## 4.1 Transformation in Relationsschemata

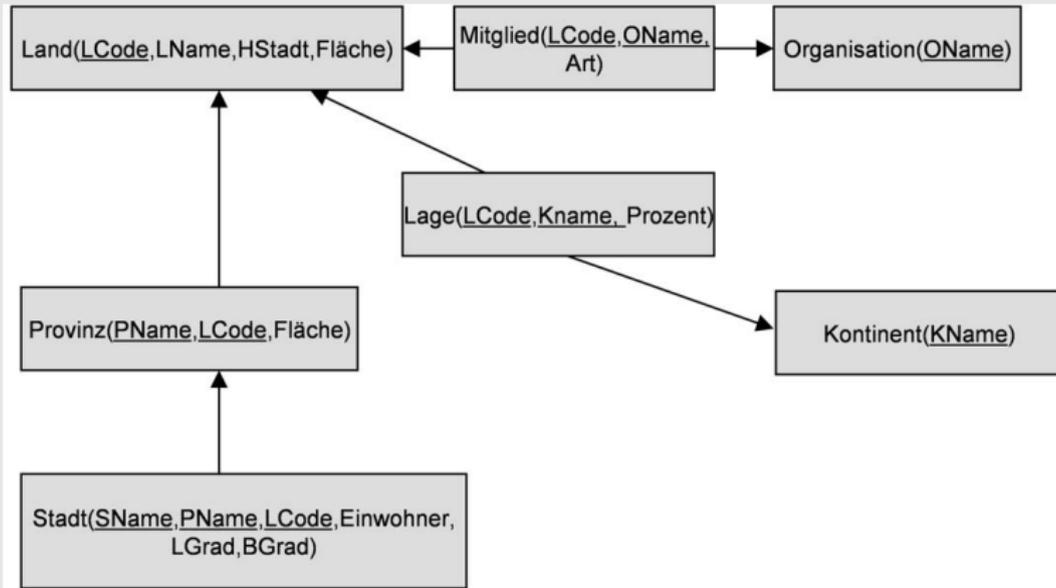
- ▶ Wir ordnen zunächst jedem Entitätstyp und jedem Beziehungstyp ein Relationsschema zu.
- ▶ Fremdschlüsselbeziehungen unter den relationalen Schemata kennzeichnen wir graphisch durch gerichtete Kanten (*child*  $\rightarrow$  *parent*).
- ▶ Unter Umständen können mehrere Relationsschemata wieder zu einem Relationsschema zusammengefasst werden, um die Anzahl Schemata nicht unnötig groß werden zu lassen:
  - ▶ Ist ein Beziehungstyp mit einem Entitätstyp über eine Kardinalität mit  $max = 1$  verbunden, dann kann das Relationsschema des Beziehungstyps in das Schema des Entitätstyps integriert werden.
- ▶ Ein so erhaltenes logisches Schema ist als Ausgangspunkt für eine weitere Verfeinerung zu verstehen, in die dann menschliches, nicht im ER-Schema dokumentiertes Wissen über die geplanten Anwendungen der Miniwelt eingehen kann.

### Mondial relational

```
Land(LCode, HStadt, LName, Fläche),  
Provinz(PName, LCode, Fläche),  
Stadt(SName, LCode, PName, Einwohner, LGrad, BGrad),  
Organisation(OName),
```

```
Mitglied(LCode, OName, Art),  
Lage(LCode, KName, Prozent).
```

## Mondial relational als Graph



## 4.2 Relationsschemata in SQL

### grundlegende Datentypen

SQL bietet eine Fülle von unterschiedlichen Datentypen an, mittels derer die Wertebereiche der Spalten einer Tabelle festgelegt werden können.

- ▶ INTEGER, SMALLINT
- ▶ NUMERIC, DECIMAL. Angabe Anzahl Ziffern insgesamt und Anzahl Kommastellen.
- ▶ REAL, DOUBLE PRECISION, FLOAT
- ▶ CHARACTER, CHARACTER VARYING, CHARACTER LARGE OBJECT
- ▶ BIT, BIT VARYING, BINARY LARGE OBJECT
- ▶ BOOLEAN
- ▶ DATE, TIME, TIMESTAMP, INTERVALL, . . . .

## Definition von Tabellen am Beispiel

```
CREATE TABLE Land (  
  LCode      CHAR(4),  
  LName     CHAR VARYING (50),  
  LName     CHAR VARYING (50),  
  Fläche    NUMBER,  
  PRIMARY KEY (LCode) )  
  
CREATE TABLE Provinz (  
  PName     VARCHAR(35),  
  LCode     VARCHAR(4),  
  Fläche    NUMBER  
  PRIMARY KEY (PName, LCode),  
  FOREIGN KEY (LCode) REFERENCES Land (LCode) )  
  
CREATE TABLE Stadt (  
  SName     VARCHAR(50),  
  PName     VARCHAR(50),  
  LCode     CHAR(4),  
  Einwohner INTEGER,  
  LGrad     NUMBER,  
  BGrad     NUMBER,  
  PRIMARY KEY (SName,PName,LCode)  
  FOREIGN KEY (LCode) REFERENCES Land (LCode),  
  FOREIGN KEY (LCode, PName) REFERENCES Provinz (LCode, PName) )
```