

3 Übungsblatt: Relationale Datenbanken: Sichten und Transaktionen

3.1 Terminologische Grundlagen

Syntax beschreibt die Struktur/Grammatik einer Sprache. Ein syntaktischer Fehler ist z.B. das Vergessen des Semikolons in einer Java-Zeile.

Semantik beschreibt die tatsächliche Bedeutung einer syntaktisch korrekten Zeile Code. Ein semantischer Fehler ist z.B. das Benutzen von Multiplikation, obwohl man fuer das gewollte Ergebnis eigentlich Addieren sollte.

Operationale Semantik: Durch eine Relation werden die moeglichen Ausfuehrungsschritte als Paare (Zustand, Nachfolgezustand) beschrieben.

Denotationelle Semantik: Die Semantik eines Programms wird durch eine Funktion zugewiesen.

3.2 Relationale Datenbanken

1.

```
aktuellerpreis(TITEL, AUTOR, VERLAG, ID, PREIS) :-
  findall(JAHR, verkauft(ID,JAHR,_,_), LISTE), max_list(LISTE, MAX_JAHR),
  produkt(ID,_,TITEL,AUTOR,VERLAG,_,_), verkauft(ID,MAX_JAHR,PREIS,_).
```

2.

```
letztesKatalogJahr(ID, LETZTES_JAHR) :-
  findall(JAHR, verkauft(ID,JAHR,_,_), LISTE), max_list(LISTE, LETZTES_JAHR).
```

3.

Unsere Interpretation eines nicht mehr angebotenen Produktes ist eine Verkaufszahl von 0 im Jahre 2016.

```
nichtAngebotenImBestand(ID) :-
  verkauft(ID,2016,_,0), produkt(ID,_,_,_,_,BESTAND), BESTAND>0.
```

3.3 Aggregation in Relationalen Datenbanken

1.

```
anzahlInKategorie(KATEGORIE, ANZAHL) :-
  aggregate_all(count, produkt(_,KATEGORIE,_,_,_,_), ANZAHL).
```

2.

```
inKategorieVerkauft(KATEGORIE, EXEMPLARE) :-
  findall(SUM, (produkt(PID, KATEGORIE,_,_,_,_),
  findall(ANZAHL, verkauft(PID,_,_,ANZAHL), LISTE), sumlist(LISTE, SUM)), SUMLIST),
  sumlist(SUMLIST, EXEMPLARE).
```

3.

Da nicht in jedem Jahr die gleiche Anzahl von Produkten verkauft wird, ist der Preisnachlass ja nicht der einzige Faktor fuer verringerte Verkaufserloese. Deshalb berechnet unser Praedikat die allgemeinen Einnahmen in einem Jahr und bildet dann die Differenz zum Vorjahr.

```
preisnachlassInJahr(JAHR, NACHLASS) :-
  findall(ERLS, (verkauft(_,JAHR, PREIS, ANZAHL), ERLS is PREIS * ANZAHL), GELD),
  sumlist(GELD, EINNAHMEN), VORJAHR is JAHR-1, findall(ERLS2,
  (verkauft(_,VORJAHR, PREIS2, ANZAHL2), ERLS2 is PREIS2 * ANZAHL2), GELD2),
  sumlist(GELD2, EINNAHMEN2), NACHLASS is EINNAHMEN - EINNAHMEN2.
```

3.4 Deduktive Datenbanken (1): Hierarchische Strukturen**1.**

```
unterkategorieZu(KATEGORIE, ULISTE) :-
  kategorie(ID,KATEGORIE,_), findall(NAME, kategorie(_, NAME, ID), ULISTE).
```

2.

```
duplikatKategorien(DLISTE) :-
  findall(X, (kategorie(ID1,X,_), kategorie(ID2,X,_), ID1\=ID2), LISTE),
  sort(LISTE, DLISTE).
```

3.

```
kategorienOhneUnterkategorien(LISTE) :-
  findall((ID,KATEGORIE), (kategorie(ID, KATEGORIE,_),
  findall(NAME, kategorie(_,NAME,ID), ULISTE), length(ULISTE, 0)), LISTE).
```

4.

```
verletzteHirarchien(SORTED) :-
  findall((KID,NAME,UID), (kategorie(KID, NAME, UID), kategorie(KID, NAME, UID2),
  UID\=UID2), LISTE), sort(LISTE, SORTED).
```