

Einführung in Programmierung mit ABAP:

Teil 4: Data Dictionary

Prof. Dr. Peter Hohmann

Technische Hochschule Mittelhessen

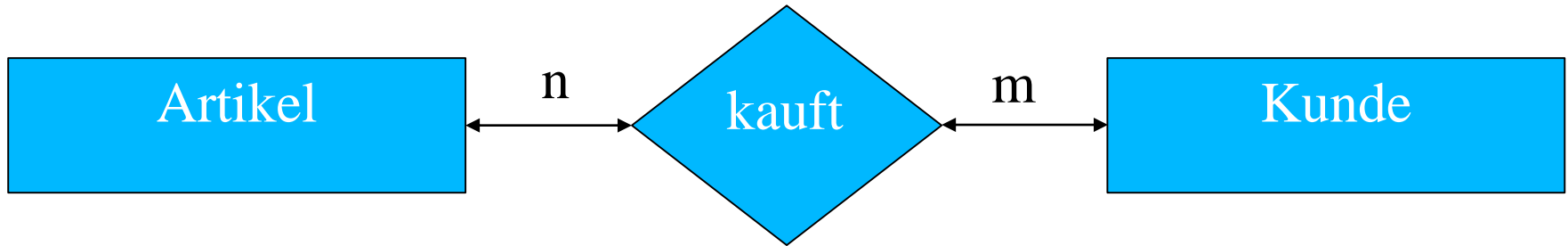
FB MNI

www.prof-dr-hohmann.de

Version 2017

Einfaches Datenmodell

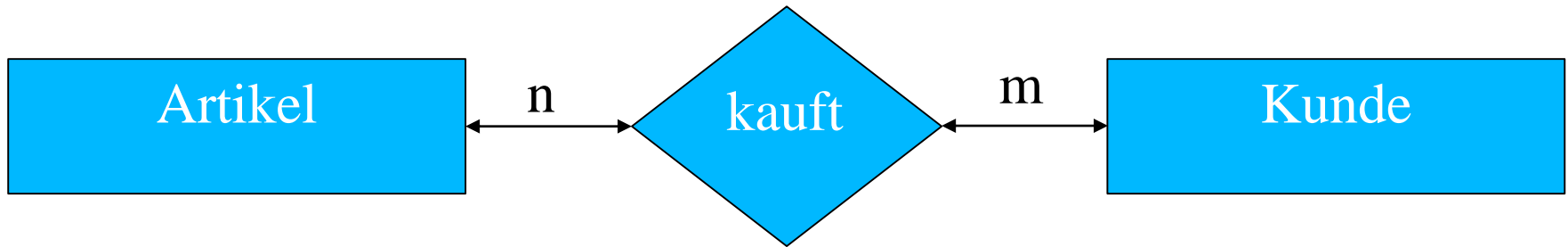
Auftragsbearbeitung



Ableitung der Tabellen? Auflösung der n:m Beziehung in 1:n Beziehungen

Einfaches Datenmodell

Auftragsbearbeitung



ARTIKEL

Attribut	Typ
Artnr	Int4
Bez	Char(50)
Preis	Dec(12,2)
..	..
..	...

APOS

Attribut	Typ
Auftrnr	int4
Artnr	??
Kunr	??
Menge	??
..	...

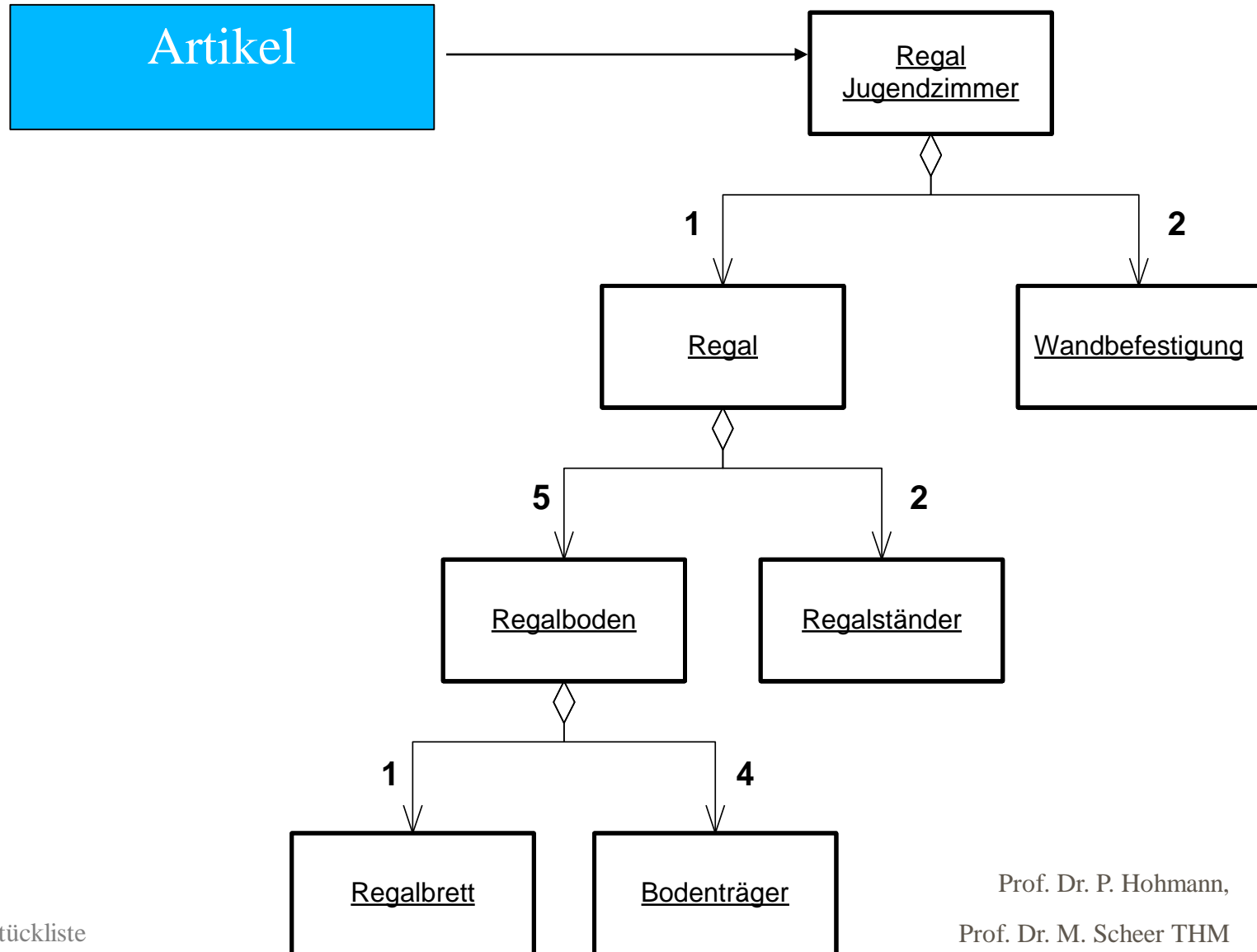
KUNDE

Attribut	Typ
Kunr	Int4
Name	Char(50)
Strasse	Dec(12,2)
PLZ	??
Ort	??

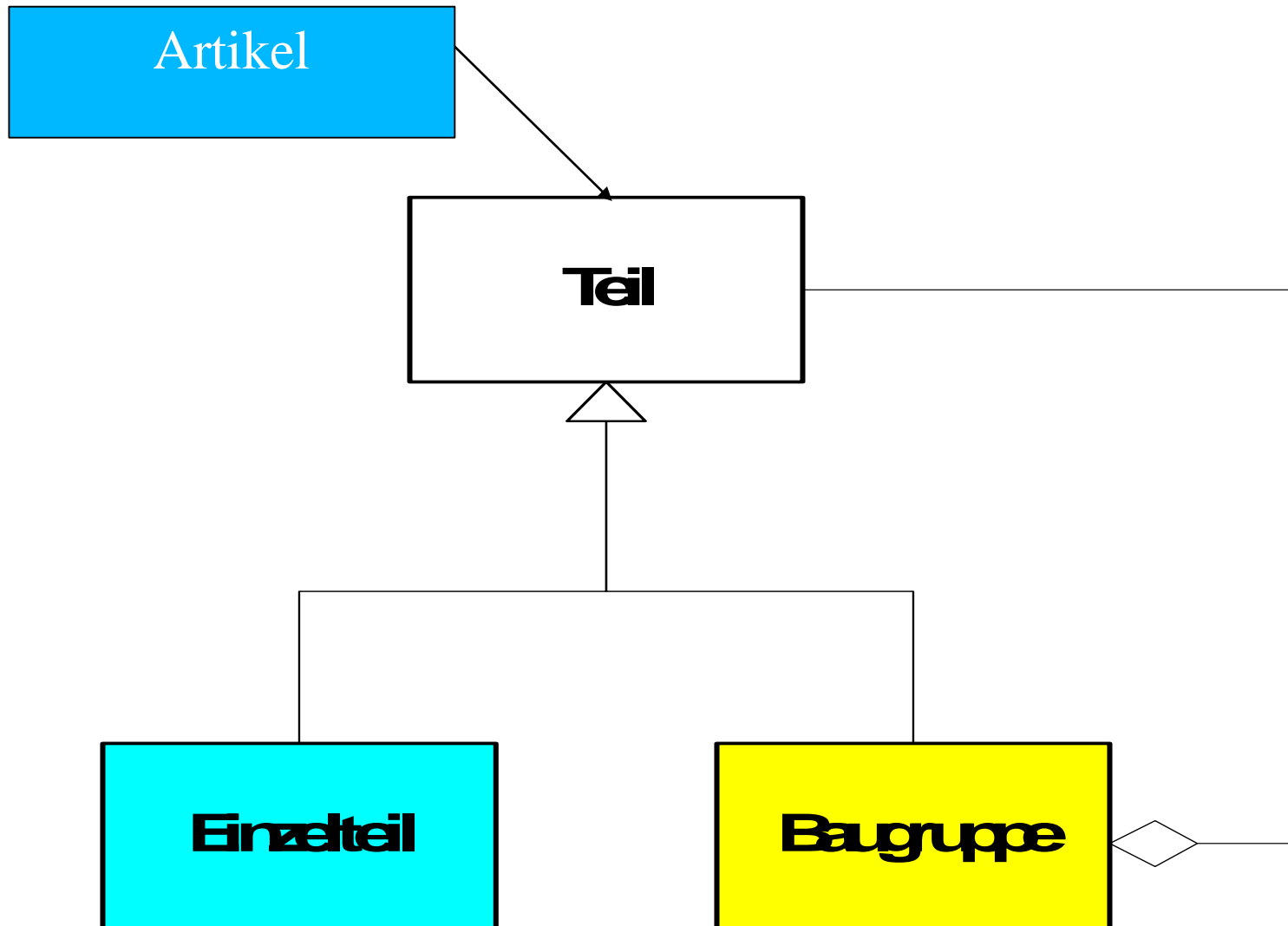
Prof. Dr. P. Hohmann,

Prof. Dr. M. Scheer THM

Artikelverwendung in Stücklisten



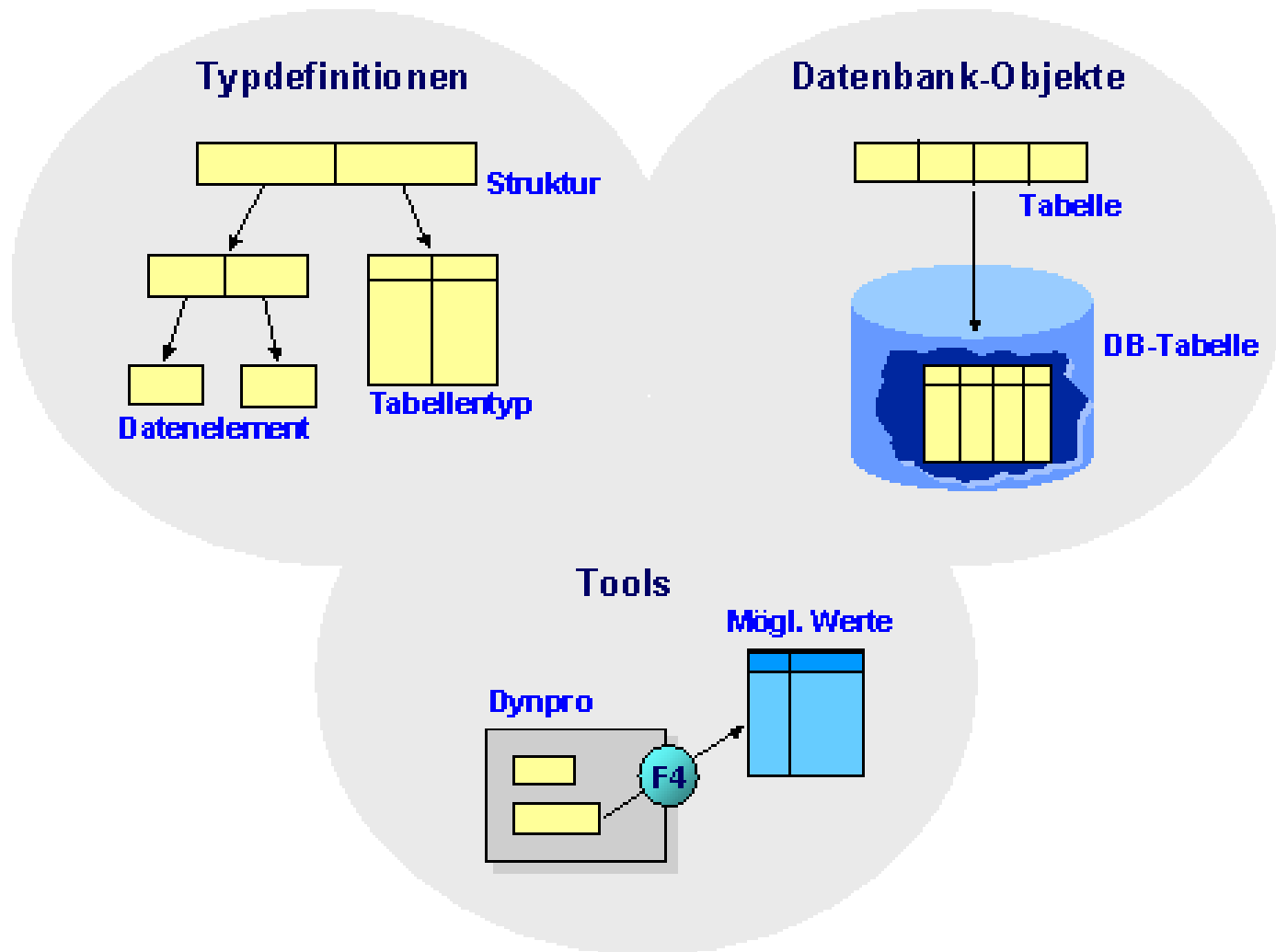
Verwendung Artikel z.B. Kompositum-Pattern



Prof. Dr. P. Hohmann,


Prof. Dr. M. Scheer THM

ABAP Dictionary




Einstiegsbild: ABAP Dictionary

ABAP Dictionary: Einstieg



<input type="radio"/> Datenbanktabelle	
<input type="radio"/> View	
<input checked="" type="radio"/> Datentyp	Z_TEIL
<input type="radio"/> Typgruppe	
<input type="radio"/> Domäne	
<input type="radio"/> Suchhilfe	
<input type="radio"/> Sperrobject	



**Metabe-
schreibungen**

Typ Z_TEIL anlegen

☒ Datenelement

☐ Struktur

☐ Tabellentyp

☒ ☐

technische Eigenschaften

Datenbanktabellen und Views

- Metabeschreibung von Tabellen und Views zum Erzeugen von physischen Tabellen
- Zusammensetzung der Zeilen aus Einzelfeldern/Spalten
- jede Spalte erhält einen elementaren Datentyp (keine Struktur, interne Tabelle etc.)
- Datentyp der Zeile ist immer eine flache Struktur

Datentypen im ABAP Dictionary

- Datenelemente

- beschreiben Einzelfelder (elementare Typen, Referenztypen)
- Zuweisung von Domänen (technische Eigenschaften)

- Strukturen

- Folge von beliebigen Datentypen des Diktionärs
- jede Komponente erhält einen Namen und einen Typ

- Tabellentypen

- Konstruktionsvorschriften interner Tabellen
- Zeilentyp, Zugriffsart, Schlüssel

- Typgruppen (Type-Pools, veraltet)

Beispiel: Typgruppen

Stück ABAP-Code, wird im ABAP-Editor gepflegt:

```
TYPE-POOL    poolname .  
TYPES:  typ1          TYPE p DECIMALS 2 ,  
          typ2 (5)      TYPE c .  
  
* ABAP-Programm:  
TYPE-POOL    poolname .  
  
DATA:  dat1    TYPE poolname_typ1 ,  
        dat2    TYPE poolname_typ2 VALUE 'xyz' .
```

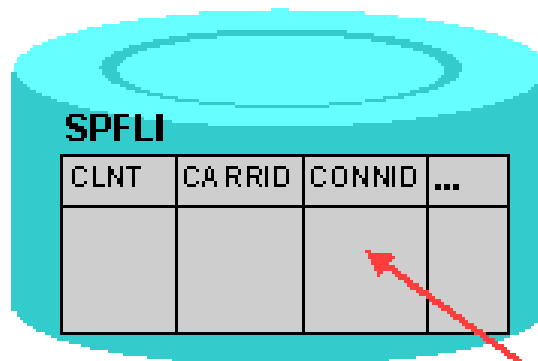
Umsetzung der Datentypen

Tabellendefinition im ABAP Dictionary

Tabelle SPFLI			
CLNT	CARRID	CONNID	...

S_CONN_ID

Datenelement
Datentyp NUMC
Länge 4



Feld CONNID mit DB-Typ VARCHAR2 (bei ORACLE) und Länge 4

ABAP Programm

```
TYPES CONN TYPE S_CONN_ID.
```

Legt Typ CONN mit
ABAP-Typ N und
Länge 4 an

Einschränkungen ABAP Dictionary

- Schlüsselfelder müssen zusammenhängend am Anfang der Tabelle stehen.
- Max. 16 Schlüsselfelder pro Tabelle erlaubt, max. Länge des Schlüssels 255 Byte.
- Falls Schlüssellänge > 120 Byte: Einschränkungen beim Transport von Tabelleneinträgen.
- Tabelle darf max. 249 Felder enthalten.
- Besonderheiten bei Typen LRAW und LCHAR.

Weiter Möglichkeiten ABAP Dictionary

- Suchhilfe für ein Feld oder Tabelle
- Prüftabelle (siehe Fremdschlüsseleintrag)
- Sperrobjekte
- Datenbank-Hilfsprogramme
 - SE14: Datenbank-Utility
 - SE30, SE31, SE32: Standardpflegeprogramme
 - Im Dictionary: Menü Hilfsmittel - Tabellenpflegegenerator

Suchhilfe

Parameter	LPos	Suchhilfe H_SPFLI Suche nach Flugnummern
CARRID	1	
CONNID	2	
CITYFROM	3	
CITYTO	4	

CA	CO	FROM	TO	Trefferliste
LH	67	Bonn	Berlin	
LH	103	München	Berlin	
LH	24	Frankfurt	Boston	
AA	188	New York	Miami	
AA	15	New York	London	
...	

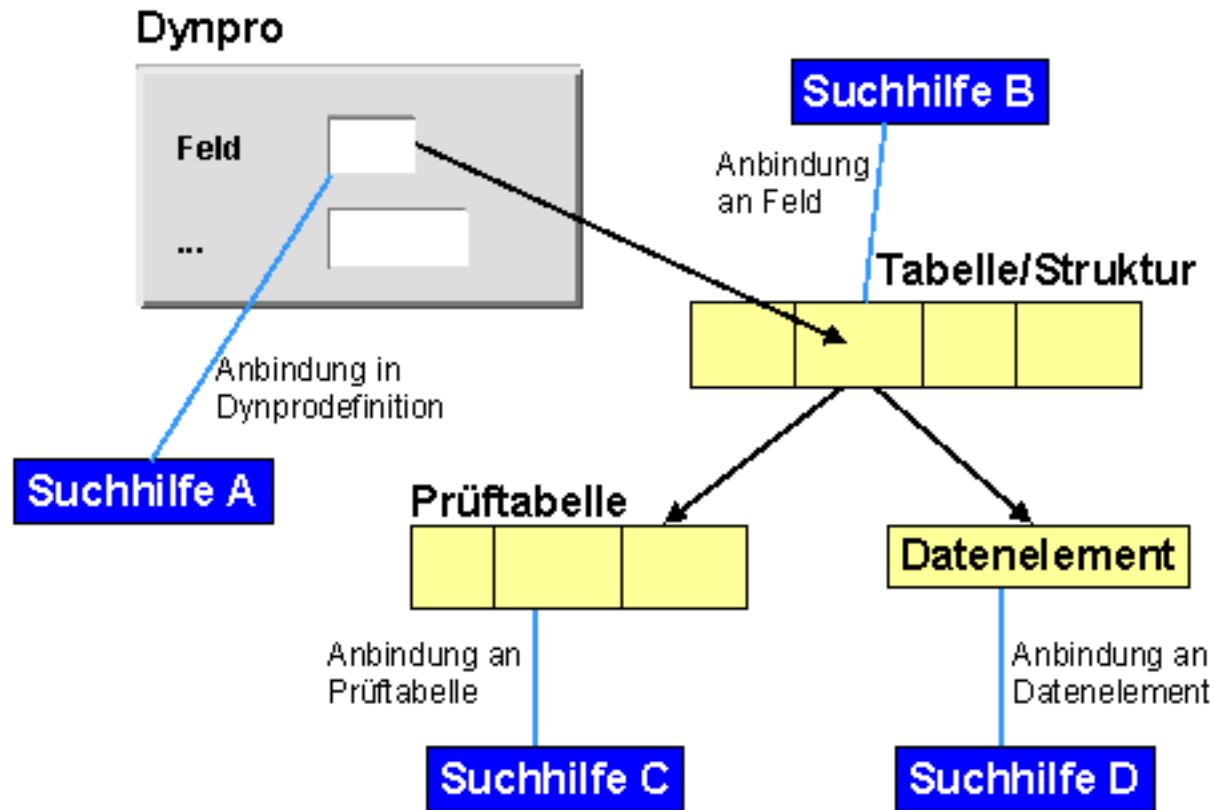
Suchhilfe H_SPFLI (Suche nach Flugnummern)

Parameter	SPos
CARRID	1
CONNID	2
CITYFROM	3
CITYTO	4

Dialogfenster zur Werteselektion

Flugges.	<input type="text"/>
Flugnr.	<input type="text"/>
Abflug	<input type="text"/>
Ankunft	<input type="text"/>

Eingabehilfe / Suchhilfe



Tabellenzeilen einfügen/löschen

Tabellenzeilen einfügen:

```
INSERT INTO db_tabelle VALUE satz .
```

```
INSERT db_tabelle FROM satz .
```

```
INSERT db_tabelle FROM TABLE itab  
[ACCEPTING DUPLICATE KEYS] .
```

Tabellenzeilen löschen:

```
DELETE db_tabelle FROM satz .
```

```
DELETE FROM db_tabelle WHERE <bedingung> .
```

```
DELETE db_tabelle FROM TABLE itab .
```


Tabellenzeilen ändern

Tabellenzeilen ändern:

```
UPDATE db_tabelle FROM satz .
```

```
UPDATE db_tabelle FROM TABLE itab .
```

```
UPDATE db_tabelle  
    SET s1 = wert1 [s2 = wert2] ...  
    [WHERE <bedingung>] .
```

Tabellenzeilen einfügen oder ändern:

```
MODIFY db_tabelle FROM satz .
```

```
MODIFY db_tabelle FROM TABLE itab .
```

Sonstiges zu Datenbankoperationen

- Systemfeld **sy-subrc**
 - 0 : Operation war erfolgreich
 - > 0 : Operation war nicht (vollständig) erfolgreich
- Systemfeld **sy-dbcnt**
 - Anzahl der verarbeitenden Sätzen
- Angabe der Datenbanktabelle
 - dynamisch: (**variable**)
 - statisch: **db_tabelle**

Noch Fragen?

