

# Tabellen

## Aufgabe

Tabellen sind die logischen Entwurfsobjekte der Business Logic Layer und der Data Access Layer in CBA. Sie stellen logische Strukturen dar und können direkt oder per Object-Relation-Mapping den Tabellenstrukturen einer Datenquelle angepasst werden. Durch die programmtechnische Realisierung der Business Logic in Form von Business Logic Units (BLUs) bzw. des Data Access in Form von Data Access Units (DAUs) werden einer Tabelle die Parameter und Methoden dieser Units vererbt. Damit repräsentiert eine Tabelle neben der Datenhaltung objektorientierte Strukturen in Form der Methoden der BLU bzw. DAU.

---

## Anforderungen

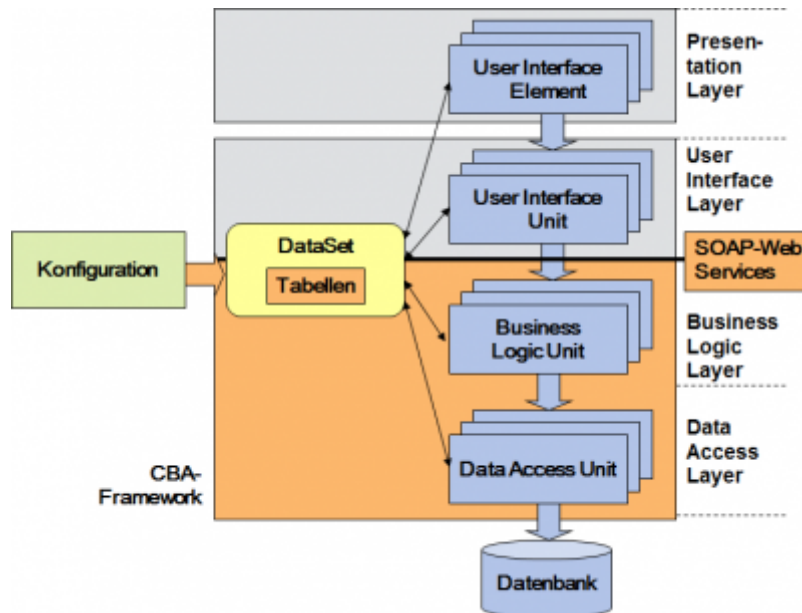
Tabellen setzen im Rahmen des CBA-Frameworks folgende Anforderungen um:

- Es gibt eine ebenenübergreifende Definition der Struktur logischer Tabellen.
  - Daten können in den logische Tabellen unabhängig von der Datenquelle gecached werden.
  - Es gibt einen unabhängigen und ebenenübergreifenden Zugriff auf Daten in logische Tabellen.
  - Den Datensätzen werden Standard- und anwendungsspezifische Methoden zugeordnet und damit ein objektorientierter Zugriff ermöglicht.
  - Die Anbindung des Datenzugriffs an Datenquellen unterschiedlichster Struktur ist frei konfigurierbar.
  - Tabellen unterstützen logische Eigenschaften von Geschäftsobjekten wie logisches Löschen, Historisierung, Datierung, Versionierung und Statusführung.
  - Tabelle implementieren in ihrer Struktur und ihren Methoden Mandantenfähigkeit und Mehrsprachfähigkeit.
  - Tabellen sind frei konfigurierbar.
- 

## Architektur

Tabellen haben eine zentrale Stellung im CBA-Framework. Sie werden durch DataTable-Objekte im ADO.NET-DataSet repräsentiert und implementieren folgende Funktionalitäten:

- frei konfigurierbare zentrale Objekte zur internen Datenhaltung in Form von DataTables im DataSet
- zentrale Konfiguration von Tabellen-, Attribut- und Geschäftsobjekt-Eigenschaften für alle drei Layer des CBA-Frameworks
- Typisierung der Attribute durch logische Datentypen mit daraus abgeleiteten Eigenschaften
- Anbindung konfigurierbarer Business Logic Units zur Implementierung von logischen Funktionen
- Object-Relation-Mapping für den Datenzugriff
- Anbindung konfigurierbarer Data Access Units zur Implementierung des Datenzugriffs auf die verschiedensten Arten von Persistenz-Medien
- WebServices zum Zugriff auf die logischen Funktionen der Tabellen



## Konfiguration

Nachfolgend werden die wichtigsten Konfigurations-Informationen beschrieben:

- **Tabellen als zentrale Entwurfsobjekte:** Tabellen werden mit logischen Namen definiert. Auf sie kann über die im CBA-Framework definierten Funktionen zugegriffen werden.
- **Data Access-Konfiguration:** Einer Tabelle kann eine Data Access Unit zugeordnet werden, die den Datenzugriff programmtechnisch realisiert. Neben dem logischen Tabellennamen kann ein physischer Tabellennamen, ein Prefix bei Umsetzung als Datenbank-Join, Standard-Filter- und Sortierfunktionen, Eigenschaften des Datenzugriffs und ein Nummernkreis zugeordnet werden.
- **Business Logic-Konfiguration:** Einer Tabelle kann eine Business Logic Unit zugeordnet werden, die die logischen Funktionen programmtechnisch realisiert. Neben dem logischen Tabellennamen können logische Eigenschaften von Geschäftsobjekten zugeordnet werden.
- **User Interface-Konfiguration:** Einer Tabelle kann eine Standard-Bedienoberfläche zugeordnet werden, die bei Verlinkung im Programm aufgerufen wird, soweit nicht anderes im Aufruf angegeben ist.
- **Konfiguration der Attribute:** Die Konfiguration der Attribute beinhaltet ebenfalls umfangreiche Informationen zu allen drei Ebenen und wird in einer separaten Hilfe beschrieben.
- **Konfiguration von Standard-Attributen:** Standard-Attribute sind Attribute, die für die Organisation der datenbezogenen Zugriffsrechte (Mandantenfähigkeit) und Systemfunktionen (technischer Schlüssel, Transaktions-Management, logisches Löschen, Historisierung) benötigt werden. Diese werden bei ihrer Konfiguration keiner Tabelle zugeordnet, zur Laufzeit aber in jede Tabelle übernommen. Die Standard-Attribute können tabellenspezifisch modifiziert werden.

Damit sind sowohl die Struktur der Tabelle als auch alle notwendigen Eigenschaften frei konfigurierbar.

## Filter

CBA verwendet in verschiedenen Entwurfsobjekten Filter zur Einschränkung der ausgewählten Daten,

z.B. in Tabellen, Oberflächen oder Datentypen. Außerdem können Filter in verschiedenen Funktionen verwendet werden. Filter können erfasst werden, indem eine spezielle Bedienoberfläche über das Icon Filter aufgerufen wird oder der Filter in der nachfolgend beschriebenen Syntax im Eingabefeld erfasst wird. Für Filter gilt eine einheitliche Syntax. In geschweifte Klammern { } eingeschlossene Ausdrücke können beliebig oft (0 ... n-mal) vorkommen. In eckige Klammern eingeschlossene Ausdrücke können optional (0 oder 1-mal) vorkommen. Vordefinierte Sprachelemente werden in Fettschrift dargestellt. Lexikalische Sprachelemente sind kursiv geschrieben und werden nachfolgend erläutert. Die semantische Bedeutung des Ausdruckes ist nach dem Kommentarzeichen "/" angegeben.

Filter	Element	Bedeutung
filter	filter-element {& filter-element}	Liste AND-verknüpfter Filter-Elemente
filter	[~] ( element {   element } )	Liste OR-verknüpfter Filter-Elemente, '~' bedeutet Negation (NOT)
filter-element	name operator value	Prüfung eines Attributes auf einen Wert
filter-element	filter	Aufruf eines untergeordneten Filter-Konstruktes

Dabei haben die lexikalischen Begriffe folgende Bedeutung:

- **name:** Name eines Attributes (logischer Name ohne Prefix)
- **operator:** anzuwendender Operator entsprechend Tabelle
- **value:** Vergleichswert in datentyp-spezifischer Notation oder Name eines Attributes

CBA unterstützt folgende Operatoren:

Operator	Lexik	Bedeutung
=	=	Attributwert ist gleich dem angegebenen Wert
<	<	Attributwert ist kleiner dem angegebenen Wert
>	>	Attributwert ist größer dem angegebenen Wert
<>	<>	Attributwert ist ungleich dem angegebenen Wert
≤	≤	Attributwert ist kleiner oder gleich dem angegebenen Wert
≥	≥	Attributwert ist größer oder gleich dem angegebenen Wert
UNIFY	#	Attributwert ist gleich Wert oder gleich Default-Wert
CONTAINS	%	Attributwert enthält mindestens ein in Wert enthaltenes Zeichen
LIKE	§	Eine Teilzeichenkette des Attributwertes entspricht dem angegebenen Wert. Als Platzhalter werden die in SQL üblichen Platzhalter '%' für eine beliebige Zeichenkette und '_' für ein einzelnes Zeichen verwendet.
IN	@	Attributwert ist ein Wert aus der Aufzählungsliste (durch Komma seperierte Werteliste oder SELECT-Anweisung)

Zu den definierten Filtern werden automatisch die notwendigen Bedingungen zur Realisierung der Mandantenfähigkeit, der Historisierung, der Datierung und Versionierung hinzugefügt.

## Sortierung

CBA bietet in verschiedenen Entwurfsobjekten Möglichkeiten zum Sortieren von Datensätzen an. Sortier-Möglichkeiten bieten zudem verschiedene Funktionen an. Sortier-Anweisungen werden in der nachfolgend beschriebenen Syntax im Eingabefeld erfasst. Die Syntaxbeschreibung ist analog zu

Filtern In geschweifte Klammern { } eingeschlossene Ausdrücke können beliebig oft (0 ... n-mal) vorkommen. In eckige Klammern eingeschlossene Ausdrücke können optional (0 oder 1-mal) vorkommen. Alternativen werden durch das Trennzeichen '|' gekennzeichnet. Vordefinierte Sprachelement werden in Fettschrift dargestellt. Lexikalische Sprachelemente sind kursiv geschrieben und werden nachfolgend erläutert.

sort	sort-element { , sort-element }	Liste von Sortier-Elementen
sort-element	[+ -]name	optional vorzeichenbehafeter Attribute-Name , '-' bedeutet absteigende Sortierung (DESC) , sonst wird aufsteigend sortiert (ASC)

Dabei haben die lexikalischen Begriffe folgende Bedeutung:

- **name:** Name eines Attributes (logischer Name ohne Prefix)

From:

<https://wiki.tim-solutions.de/> - **TIM Wiki** / [NEW TIM 6 Documentation](#)

Permanent link:

<https://wiki.tim-solutions.de/doku.php?id=software:cba:tables&rev=1418743919>

Last update: **2021/07/01 10:01**

