

12. Funktionen die bei Suchformaten (besondere Auswahl) und Berichten eingesetzt werden können

WINSV bietet dem erfahrenen Benutzer Funktionen ähnlich wie in einer Programmiersprache an, die Berechnungen oder die Manipulation von Merkmalsinhalten gestatten.

Funktionen können bei Suchformaten (besondere Auswahl) und bei der Erstellung von Berichten mit dem Berichtsgenerator verwendet werden.

In jeder Funktion können Konstanten oder/und Variable (Merkmalsnamen) verwendet werden. Dabei gilt:

- Jede Funktion beginnt mit dem Schlüsselwort der Funktion.
- Die Parameter einer Funktion werden in runde Klammern gesetzt.
- Die erwarteten Parameter werden durch ; voneinander getrennt.
- Konstante werden in Hochkommas (") gesetzt.
- Als Variable werden nur Merkmalsnamen aus einer der vorhandenen Dateien und selbst definierte "berechnete Merkmale" aus Berichten in der originalen Schreibweise akzeptiert;
- Kleinschreibung wird in Großschreibung umgesetzt (Merkmalsnamen können der Datensatzbeschreibung entnommen, bei Suchformaten auch mit der Funktionstaste <F9> am Bildschirm angezeigt werden).
- Funktionen können selbst als Parameter einer anderen Funktion eingesetzt werden.
- Funktionen, in Berichten oder Suchformaten eingesetzt, verändern nicht den gespeicherten Wert eines Merkmals, sondern ändern nur das Ausgabeformat des Wertes.
- Alle Funktionen werden sofort nach Abschluß auf zulässige Syntax geprüft, als Variable verwendete Merkmalsnamen werden auf Existenz in der Datenbank geprüft, falsche Schreibweisen führen zur Zurückweisung oder Stornierung.

12.1 Beschreibung einzelner Funktionen

Großschreibung einer Zeichenkette

Cap(Zeichenkette) oder **Upp**(Zeichenkette) dient zur Umsetzung einer Zeichenkette in Großbuchstaben.

Cap("Huber") oder		
Upp("Huber")	ergibt	HUBER
Cap(FAMILIENNAME)		
Upp(FAMILIENNAME)	ergibt	den Familiennamen des aktuellen Datensatzes in Großbuchstaben
Cap("1234567890")		
Upp("1234567890")	ergibt	1234567890, also wirkungslos

Kleinschreibung einer Zeichenkette

Low(Zeichenkette) dient zur Umsetzung einer Zeichenkette in Kleinbuchstaben. Syntax wie **Cap**()

Zusammenfügen von Zeichenketten (Merkmalsinhalten)

Con(Zeichenkette_1;Zeichenkette_2;...)

In Berichten kann es zweckmäßig sein, die Inhalte mehrerer Merkmale zu einer einzigen Zeichenkette zusammenzufügen. Dabei muß die unterschiedliche Länge der Merkmalsinhalte berücksichtigt werden.

Con(Zeichenkette_1;Zeichenkette_2;...) verwendet nur die tatsächliche Länge einer Zeichenkette, Leerzeichen werden abgeschnitten. Die Anzahl der Teilzeichenketten ist nicht beschränkt.

Con("Huber";"Renate")	ergibt	HuberRenate
Con("Huber";" Renate")	ergibt	Huber Renate (zu beachten: das Leerzeichen vor Renate)
Con ("Huber ";"Renate")	ergibt	Huber Renate (zu beachten: das Leerzeichen nach Huber)
Con ("Huber";" ";"Renate")	ergibt	Huber Renate
Con(FAMILIENNAME;" ";"RUFNAME)	ergibt	Huber Renate, sofern Huber der FAMILIENNAME und Renate der RUFNAME im aktuellen Datensatz ist.
Con(VORANGESTELLT;" ";"FAMILIENNAME;" ";"RUFNAME)	ergibt	von Huber Renate, sofern dies die Inhalte des aktuellen Datensatzes sind.
Con(Cap(FAMILIENNAME);" ";"RUFNAME)	ergibt	HUBER, Renate
Con(FAMILIENNAME;" ";"Cap(RUFNAME)	ergibt	Huber, RENATE

Ganzzahliger Teil einer Zahl

Int(Wert) gibt den ganzzahligen Teil einer Zahl zurück. **Int** schneidet alle Nachkommastellen der Zahl ab, rundet also nicht!
Int(Zeichenkette) wandelt eine Zeichenkette in eine Ganzzahl um, mit der gerechnet werden kann.

Int(276,6789)	ergibt	276	
Int(276,9999)	ergibt	276	
Int(-276,999)	ergibt	-276	
Int(Mid(#D;9;2))	ergibt	94	sofern das Datum TT.MM.19JJ mit 94 endet. Mit diesem Wert kann gerechnet werden.

Formatierung einer Zeichenkette

Oftmals ist es insbesondere in Listen oder Formbriefen erforderlich, Zeichenketten in einem bestimmten Format auszugeben. Die Funktion **Jst**() bietet dazu ein sehr breites Spektrum an Bearbeitungsmöglichkeiten an:

a) Links- bzw. rechtsbündig/ zentriert ausrichten, mit Füllzeichen bis zu einer bestimmten Länge auffüllen

Jst("Huber";20)	ergibt	Huber
Jst("Huber";-20)	ergibt	Huber
Jst("Huber";"20P.")	ergibt	Huber.....
Jst("Huber";"-20P.")	ergibtHuber
Jst(FAMILIENNAME;"20_")	ergibt	Huber_____ , sofern der FAMILIENNAME des aktuellen Datensatzes Huber lautet.
Jst("Huber";"^11")	ergibt	Huber
Jst(FAMILIENNAME;"^11")	ergibt	Huber
Con(Jst("Huber";"-20P.");", ";Jst("Renate";"20P."))	ergibtHuber, Renate.....

b) Abschneiden von Zeichenketten

Jst("Huber";"1X")	ergibt	H
Jst("Huber";"4X")	ergibt	Hube
Jst(FAMILIENNAME;"2X")	ergibt	Hu
Cap(Jst(FAMILIENNAME;"3X"))	ergibt	HUB

c) Umwandlung Groß/Kleinschreibung

Jst("Huber";"U")	ergibt	HUBER (nur Großbuchstaben)
Jst("Huber";"L")	ergibt	huber (nur Kleinbuschstaben)
Jst("huber";"C")	ergibt	Huber (erster Buchstabe groß)
Jst(Jst("Huber";"-20p.");"U")	ergibtHUBER

d) Zahlen formatieren

Jst(0,235;"N")	ergibt	0,235
Jst(0,235;"N2")	ergibt	0,24

e) Formatierung von Datum und Zeit

Die Systemvariablen **#D** (Systemdatum) und **#T** (Systemzeit) stehen in Suchformaten und Berichten stets aktuell zur Verfügung. In den folgenden Beispielen wird ihre Formatierung dargestellt. Zur Formatierung sind folgende Symbole vereinbart:

D stellt Daten als Datum dar

: interpretiert die nachfolgende Zeichenkette als Formatierungsanweisung

Y Jahr (99)	y Jahr (1999)	
C Jahrhundert (19)		
M Monat (06)	m Monat (JUN)	n Monat (Juni)
D Tag (12)	d Tag (12ter)	
W Wochentag (5)	w Wochentag (Freitag)	
H Stunde (0...23)	h Stunde (1...12)	
N Minuten		
S Sekunden	s Hundertstel	

Jst(#D;"D:w, d n CY")	ergibt	Samstag, 29. November 1997 bei einem Systemdatum 29.11.1997
Jst(#D;"D:d n Y")	ergibt	29. November 94
Jst(#D;"D:d n y")	ergibt	29. November 1994
Jst(GEBURTSDATUM;"D:w, d n C ..")	ergibt	Samstag, 29. November 19 .. sofern der Schüler am 29.11.19.. geboren ist

Jst(AUSTR_DATUM;"D:DMY")	ergibt	291199	sofern der Schüler am 29.11.99 die Schule verlassen hat
Jst(#T;"T:H-N")	ergibt	14-07	bei einer Systemzeit 14.07 Uhr

Die Systemvariablen **#FD** und **#FDT** enthalten die aktuellen Datums- und Zeitformate, sie können zur Kontrolle verwendet werden, eine Ausgabe in Berichten ist nicht sinnvoll. Werden **#D** oder **#T** ohne Formatierungsangaben verwendet, werden **#FD** bzw. **#FDT** zur Formatierung verwendet.

Länge einer Zeichenkette

Len(Zeichenkette) gibt die Anzahl von Buchstaben einer Zeichenkette als Ganzzahl wieder

Len("Huber")	ergibt	5	
Len("Huber") + 20	ergibt	25	
Len(FAMILIENNAME)	ergibt	5	sofern Huber der Familienname des aktuellen Datensatzes ist.

Zeichenfolgen aus einer Zeichenkette kopieren/untersuchen

Mid(Zeichenkette;ab Stelle der Zeichenkette;Gesamtlänge der Teilzeichenkette)

Mid() erlaubt es, bekannte Teile eines Feldinhaltes zu untersuchen und herauszukopieren. Die behandelte Zeichenkette wird dadurch nicht verändert.

Mid("29.11.1999";4;2)	ergibt	11	
Mid(#D;4;2)	ergibt	11	sofern das Systemdatum den Monat November enthält
jst(Mid(#D;4;2);"D:n")	ergibt	November	sofern das Systemdatum den Monat 11 enthält
Mid(S_SCHULJAHR;3;2)	ergibt	93	im Schuljahr 1993/94
Mid(S_SCHULJAHR;6;2)	ergibt	94	im Schuljahr 1993/94
Mid(S_SCHULJAHR;3;5)	ergibt	93/94	im Schuljahr 1993/94

Einem numerischen Wert eine Zeichenkette zuweisen

Pick(Wert;wenn Wert=0;wenn Wert=1; ...)

Pick() erlaubt es numerische Inhalte eines Datenfeldes in einen anderen Wert oder in eine Zeichenkette umzuwandeln.

Pick("1";"kein zulässiger Monat";"Januar";"Februar";"März";"...")

ergibt Januar

Pick(Mid(#D;4;2);" "; "Januar";"Februar";"März";"...")

ergibt den Monat aus dem Systemdatum in Worten

Wichtig: Die Zuweisungen zum angegebenen Wert beginnen stets von links bei 0, soll z.B. einem Wert 8 eine Zeichenkette zugewiesen werden, müssen vorher 8 weitere Zuweisungen (0 - 7) angegeben werden, die Zuweisung für 8 stünde also an 9ter Stelle der Zuweisungskette.

Ist der Wert nicht numerisch, wird er als 0 behandelt.

Pick(Feldinhalt <> = Wert;**sonst**;dann) kann als WENN-SONST-DANN-Funktion eingesetzt werden. Bei Vergleichen wird der Wahrheitswert 0 oder 1 zurückgegeben. Dadurch kann auf bestimmte Datenfeldinhalte gezielt reagiert werden.

Pick(GESCHLECHT="W";"Schüler";"Schülerin")

ergibt Schülerin

sofern das Geschlecht weiblich ist

ergibt Schüler

sofern das Geschlecht männlich ist

Pick(ERZB1_ART="EL";"Liebe(r) Erziehungsberechtigte(r)";"Liebe Eltern,")

ergibt **Liebe Eltern**

sofern der Eintrag "EL" im Feld "Art der Erziehungsberechtigung" gefunden wird.

ergibt **Liebe(r) Erziehungsberechtigte(r)**

sofern andere Personen erziehungsberechtigt sind.

Jeder Parameter von **PICK**() kann wiederum durch ein **PICK**() ausgedrückt werden, womit sich sehr komplexe Zusammenhänge bilden lassen.

Pick(ERZB1_ART="EL";**Pick**(ERZB1_ART="MU";**Pick**(ERZB1_ART="VA";"Sehr geehrte Damen und Herren";"Sehr geehrter Herr");"Sehr geehrte Frau");"Sehr geehrte Eltern,")

In diesem Beispiel gilt:

EL	ergibt	Sehr geehrte Eltern
MU	ergibt	Sehr geehrte Frau
VA	ergibt	Sehr geehrter Herr
...	ergibt	Sehr geehrte Damen und Herren

Wichtig: **Pick**() arbeitet umgekehrt wie sonst in Programmiersprachen gewohnt. Nach der Bedingung kommt die Anweisung für falsch, dann die Anweisung für wahr.

Die Position einer Zeichenfolge in einer Zeichenkette berechnen

POS(Vergleichszeichenfolge;Zeichenkette, in der gesucht wird)

Pos() gibt den Beginn der Vergleichszeichenfolge innerhalb der Zeichenkette, in der gesucht wird, als Wert zurück.

Pos("u";"Huber")	ergibt	2	
Pos("U";"Huber")	ergibt	0	
Pos("be";"Huber")	ergibt	3	
Pos("W";GESCHLECHT)	ergibt	1	sofern das GESCHLECHT "W" gespeichert ist,
	ergibt	0	sofern das GESCHLECHT "M" gespeichert ist
Pick(Pos("W";GESCHLECHT);"Knabe";"Mädchen")			
	ergibt	"Mädchen"	sofern im Merkmal GESCHLECHT "W" gespeichert ist,
		"Knabe"	sofern im Merkmal GESCHLECHT "M" gespeichert ist

Systeminformationen

Die Funktion SYS() gibt Systeminformationen zurück, eine Änderung der Werte ist nicht möglich. Sys() kann bei der Suche von Systemfehlern hilfreich sein.

Sys(10)	Pfadname der aktuellen Anwendungsdatei
Sys(24)	Name der aktuellen Ausgabeeinheit
Sys(80)	Aktueller Berichtsname
Sys(81)	Aktueller Suchformatsname (besondere Auswahl)
Sys(82)	Name der Hauptdatei
Sys(83)	Name der in der Hauptdatei gespeicherten Datensätze
Sys(101)	Name des aktuellen Druckers
Sys(109)	nicht benötigter Speicher
Sys(110)	CPU-Typ
AHF_NAME	Name des aktuellen Berichts
#D	aktuelles Systemdatum
#FD	Datumsformat (sh. auch Jst())
#FT	Zeitformat (sh. auch Jst())
#FDT	Datums- und Zeitformat (sh. auch Jst())
#T	Aktuelle Systemzeit
dat(#T,#FDT)	ergibt die vollständige Zeit und das Datum
#P	Aktuelle Seitennummer eines Berichtes
#SUBFLD	gibt den Namen des Sortierfeldes zurück, das bei Zwischensummen in Berichten die Zwischensumme ausgelöst hat.
Nam(#SUBFLD)	gibt eine Zeichenkette zurück,
[#SUBFLD]	enthält den Zwischensummenwert.
#R	Aktuelle Anzahl gedruckter Datensätze bei Berichten

12.2 Beispiele zum Einsatz von Funktionen in Berichten als "Berechnetes Merkmal"

In Berichten (*.AHR) genügt es häufig nicht, den reinen Merkmalsinhalt zu drucken. Um den Informationsgrad des Dokumentes zu erhöhen, sind Berechnungen, Umcodierungen, Zusammenfassungen notwendig. In Serienbriefen müssen aus dem Geschlecht des Adressaten die geschlechtspezifischen Formulierungen errechnet werden, sofern diese nicht als eigene Merkmale geführt werden.

In den folgenden Beispielen werden stets Merkmalsnamen verwendet, die in der Datensatzbeschreibung von WINSV enthalten sind (sh. Anhang zum Handbuch). Merkmalsnamen werden immer in Großbuchstaben geschrieben. Namen links vom Gleichheitszeichen beziehen sich auf berechnete Merkmale, die in den mitgelieferten Berichten enthalten sind und dort bei ihrem Einsatz beobachtet werden können.

Berechnete Merkmale können als eigenständige Funktionen im aktuellen Bericht weiterverwendet werden. Sie werden beim Bericht mitgespeichert und stehen nur dort zur Verfügung. Die Namen der berechneten Merkmale können frei gewählt werden, dürfen aber nicht dem Namen eines Merkmals der Datenbank entsprechen.

Zusammenfassen der Namensbestandteile eines Schülers:

Gespeichert in getrennten Merkmalen sind:

- Familienname
- vorangestellte Namensbestandteile
- nachgestellte Namensbestandteile
- Vornamen
- Rufname

Im einfachsten Fall kann für jedes Merkmal eine eigene Spalte angelegt werden. Dabei bliebe bei der Mehrzahl der Schüler die Spalte "vorangest." bzw. "nachgest." leer und die Eintragslänge müßte bei jeder Spalte getrennt berücksichtigt werden. Es bietet sich also Con() an, um alle Teile in einer einzigen Zeichenkette zusammenzufügen.

ALLE_NAMEN	=	con(VORANGESTELLT;" ";FAMILIENNAME;"
		";NACHGESTELLT;" ";VORNAMEN)
	ergibt	von Huber de la Fuente, Maria Desire Louise
	oder	Bauer , Helene

Solange alle Namensbestandteile besetzt sind, funktioniert diese Methode einwandfrei, fehlen aber Teile, wird das Druckbild fehlerhaft bzw. unregelmäßig. Fehlt ein vorangestellter Namensbestandteil, beginnt der Familienname mit einem Leerzeichen, da dieses von Con() stets wie oben definiert eingefügt wird. Ähnliches geschieht bei fehlendem nachgestelltem Namensbestandteil. Vor dem Komma, das den Familiennamen beendet, wird ein überflüssiges Leerzeichen eingefügt.

Zur Lösung des Problems wird zunächst je ein eigenes berechnetes Feld für die voran- und nachgestellten Namensbestandteile gebildet, um die Gesamtformel nicht zu unübersichtlich zu machen.

VON	=	pick(VORANGESTELLT<>" ";con(VORANGESTELLT;" ")
HINTEN	=	pick(NACHGESTELLT<>" ";con(" ";NACHGESTELLT)

In VON bzw. HINTEN wird durch PICK() geprüft, ob das Merkmal belegt ist, ein leeres Merkmal ergibt den Wert 0. Bei 0 wird nichts getan, bei besetztem Merkmal (Wahrheitsgehalt = 1) wird mit der Con()-Funktion dem Merkmal VORANGESTELLT / NACHGESTELLT ein abschließendes/führendes Leerzeichen, das den Abstand zum Familiennamen herstellt, angefügt. In der bisherigen Formel werden diese Leerzeichen dann nicht mehr benötigt (;" ");!

Somit ergibt die folgende Formulierung von ALLE_NAMEN immer eine korrekte Zeichenkette, die linksbündig ohne führendes Leerzeichen bleibt und keine überflüssigen Leerzeichen enthält.

ALLE_NAMEN	=	con((VON;FAMILIENNAME;HINTEN;" ";VORNAMEN)
	ergibt	von Huber de la Fuente, Maria Desire Louise
	oder	Bauer, Helene

Das berechnete Merkmal liefert Zeichenketten, die für Listen sicherlich zu lang sind, in Briefen stört es dagegen nicht, bzw. ist sogar erwünscht. Mit Jst() läßt sich die gewünschte maximale Länge einstellen:

ALLE_NAMEN	=	jst(con((VON;FAMILIENNAME;HINTEN;" ";VORNAMEN);20X)
	ergibt	von Huber de la Fuen
	oder	Bauer, Helene

ALLE_NAMEN	=	jst(con((VON;FAMILIENNAME;HINTEN;" ";VORNAMEN);30X)
	ergibt	von Huber de la Fuente, Maria
	oder	Bauer, Helene

Der Einsatz von Jst() kann noch durch den Parameter "P." erweitert werden, um bei kurzen Namen die Spalte aufzufüllen (Gliederungs-/Lesehilfe).

Diese Methode hat allerdings nur beim Einsatz von nicht proportionalen Schriften die gewünschte Wirkung.

ALLE_NAMEN	=	jst(con((VON;FAMILIENNAME;HINTEN;" ";VORNAMEN);30XP.)
	ergibt	von Huber de la Fuente, Maria
	oder	Bauer, Helene.....

Um den Familiennamen hervorzuheben, kann dieser in Großbuchstaben dargestellt werden:

ALLE_NAMEN	=	jst(con((VON;Upp(FAMILIENNAME);HINTEN;" ";
		VORNAMEN);30XP.)
	ergibt	von HUBER de la Fuente, Maria
	oder	BAUER, Helene.....

Aufbau einer Standardfußzeile für Berichte

Jeder gedruckte Bericht sollte eine Fußzeile enthalten, die Auskunft über den Berichtsnamen, den Stand der enthaltenen Daten, sowie den Umfang der ausgewählten Daten gibt.

Beispiel: NOTEN1.AHR Stand: 3.3.93 besondere Auswahl: Hauswirtschaft

BERICHT_NAME = con(AHF_NAME;" Stand: ";#D;" besondere Auswahl: ";SYS(81))

Zusammensetzung der Formel:

- die Systemfunktion AHF_NAME enthält den Berichtsnamen
- #D gibt das Systemdatum wieder
- die Systemfunktion Sys(81) enthält den Namen des aktiven Suchformats
- Con() faßt die Teile zu einer Zeichenkette zusammen

In der gezeigten Form wird die Fußzeile stets in gleicher Weise ausgegeben, ist kein Suchformat aktiv, erscheint trotzdem "besondere Auswahl:". Besser wäre es, wenn der Hinweis auf die Auswahl nur erschiene, wenn tatsächlich ein Suchformat

verwendet wurde. Um dies zu verwirklichen, wird mit dem "berechneten Feld" FILTER die benötigte Zeichenkette erzeugt.

FILTER = **Pick(Sys(81)<>"";" ";con(" besondere Auswahl: ";Sys(81)))**

Sys(81) enthält den Namen des aktuellen Suchformates. Ist kein Suchformat gesetzt, ist der Wahrheitsgehalt = 0.

Mit der Pick()-Funktion läßt sich aufgrund dieser Werte entschlüsseln:

0	entspricht " " (Leerzeichen)
1	entspricht " besondere Auswahl: ";Sys(81)

Die Teilfunktion FILTER wird in Berichtsname eingesetzt:

BERICHT_NAME = **con(AHF_NAME;" Stand: ";#D;**FILTER**)**

gesetztes Suchformat ergibt **NOTEN1.AHR Stand: 3.3.93 besondere Auswahl: Hauswirtschaft**

kein Suchformat ergibt **NOTEN1.AHR Stand: 3.3.93**

Berechnung des Schulbesuchsjahres

An Volksschulen/Sondervolksschulen kann das Jahr des Schulbesuchs anhand der Merkmale S_SCHULJAHR (aktuelles Schuljahr) und EINSCHUL_JAHR (Eintritt in die erste Jahrgangsstufe) berechnet werden.

Mit Mid() wird aus dem aktuellen Schuljahr (z.B. 1994/95) der erste Teil ausgeschnitten:

Mid(S_SCHULJAHR;1;4) ergibt z. B. 1994

Subtraktion des Einschulungsjahres (z.B. 1987) und Addition von 1 ergibt daraus das Schulbesuchsjahr:

BESUCH_JAHR = **Mid(S_SCHULJAHR;1;4) - EINSCHUL_JAHR + 1**

Obwohl es sich bei den beteiligten Merkmalseinträgen um Zeichenketten handelt, berechnet WINDS das Schulbesuchsjahr 8 richtig.

Unter verschiedenen Umständen reagiert WINDS jedoch nicht ordnungsgemäß, die automatische Umwandlung Zeichenkette in Zahlenwert gelingt nicht. Damit die Umwandlung in jedem Fall erzwungen wird, kann die Int() - Funktion herangezogen werden. Besser also:

int(Mid(S_SCHULJAHR;1;4))

In einem Suchformat würde die folgende Berechnungsformel

int(Mid(S_SCHULJAHR;1;4)) > int(EINSCHUL_JAHR) + 8

bewirken, daß nur Schüler aufgelistet werden, die bereits 9 oder mehr Schulbesuchsjahre hinter sich gebracht haben.

Zusammenfügen eines Briefdatums

In Elternbriefen etc. wird eine vollständige Datumsangabe gewünscht. Folgende Kombination aus den Funktionen Con(), Jst() und #D erzeugen ein solches Datum auf der Grundlage des Systemdatums.

Jst(#D;D:w) ergibt aktueller Wochentag (Montag, Dienstag, ...)

Jst(#D;"D:d n y) ergibt Tagesdatum in der Form TT.MONAT.19JJ

Con() fügt beide Teilzeichenketten zusammen und fügt ", den " ein

D: kündigt der Funktion Jst(), an daß nach dem Doppelpunkt eine Formatieranweisung folgt

BRIEFDATUM = **Con(Jst(#D;D:w);", den ";Jst(#D;"D:d n y))**
ergibt Mittwoch, den 03.03.1993

Aus dem Merkmal GESCHLECHT die geschlechtspezifische Formulierung entnehmen

Pos() gibt die Position einer Teilzeichenkette in einer anderen Zeichenkette wieder. Ist die Teilzeichenkette nicht enthalten, erhält man den Wert 0.

Pos("W";GESCHLECHT) ergibt bei einem Knaben den Wert 0, bei einem Mädchen den Wert 1

Pick(WERT;0;1;2;3;4;....) ordnet den Werten 0...n die angegebenen Zeichenketten zu.

Ein berechnetes Merkmal ER_SIE kann daher die richtige geschlechtsspezifische Anrede in einem fortlaufenden Text liefern.

ER_SIE = **Pick(Pos("W";GESCHLECHT);"Er";"Sie")**

ergibt Er bzw. Sie

= **Low(Pick(Pos("W";GESCHLECHT);"Er";"Sie"))**

ergibt er bzw. sie

SCHÜLER_IN = **Pick(Pos("W";GESCHLECHT);"der Schüler";"die Schülerin")**

ergibt der Schüler bzw. die Schülerin

SEINE_IHRE	=	Pick(Pos("W";GESCHLECHT);"Seine";"Ihre")
	ergibt	Seine bzw. Ihre
	=	Low(Pick(Pos("W";GESCHLECHT);"Seine";"Ihre"))
	ergibt	seine bzw. ihre
SOHN_TOCHTER	=	Pick(Pos("W";GESCHLECHT);"Ihr Sohn";"Ihre Tochter")
	ergibt	Ihr Sohn bzw. Ihre Tochter
	=	Low(Pick(Pos("W";GESCHLECHT);"Ihr Sohn";"Ihre Tochter"))
	ergibt	ihr Sohn bzw. ihre Tochter

Da es nur die Einträge "M" oder "W" geben kann, können mit einer Formulierung beide Geschlechter bearbeitet werden.

Pick() mit direktem WENN-SONST-DANN-Vergleich liefert schneller Ergebnisse:

Pick(GESCHLECHT="W";"Sohn";"Tochter")
 Pick(GESCHLECHT="W";"Er";"Sie")
 Pick(GESCHLECHT="W";"Schüler";"Schülerin")
 Pick(GESCHLECHT="W";"Seine";"Ihre")
 ...

Aus dem Merkmal ERZB1_ANREDE die richtige Anrede für Etiketten/Briefe an die Erziehungsberechtigten etc. entnehmen

Pos(ERZB1_ANREDE;"HFLUK")	ergibt	einen Wert zwischen 0 und 5, die zusammengefaßte Zeichenkette "HFLUK" repräsentiert alle möglichen Merkmalseinträge, Pos() übergibt die Position des gefundenen Eintrags in der Zeichenkette.
ERZB1_ANREDE	kein Eintrag	ergibt 0
	H	ergibt 1
	F	ergibt 2
	L	ergibt 3
	U	ergibt 4
	K	ergibt 5

Pick() wertet entsprechend aus:

ERZB_ANREDE	=	Pick(Pos(ERZB1_ANREDE;"MHFLUK");"";"Herrn";"Frau";"Fräulein"; ;"Herrn und Frau";"Damen und Herren")	
ERZB1_ANREDE	kein Eintrag	ergibt	
	H	ergibt	Herrn
	F	ergibt	Frau
	L	ergibt	Fräulein
	U	ergibt	Herrn und Frau
	K	ergibt	Damen und Herren

Das Merkmal GASTSCHUELER entschlüsseln

Die Zeichenkette "NEPKOA" enthält alle zulässigen Schlüssel für die Gastschülereigenschaft der Volksschulen. POS() ermittelt die Position des tatsächlichen Eintrags beim Schüler in dieser Zeichenkette und gibt einen numerischen Wert zurück.

GASTSCHUELER	kein Eintrag	ergibt	0
	N	ergibt	1
	E	ergibt	2
	P	ergibt	3
	K	ergibt	4
	O	ergibt	5
	A	ergibt	6

Pick() ordnet anschließend dem numerischen Wert 0 - 6 die angegebene Entschlüsselung (Zeichenkette) von links mit 0 beginnend zu:

Pick(2;"";"keiner";"auf Antrag";"aus päd. Gründen";"Klassenbildung";"Ordnungsmaßnahme";"anderer Aufwandsträger")	ergibt	auf Antrag
Pick(5;"";"keiner";"auf Antrag";"aus päd. Gründen";"Klassenbildung";"Ordnungsmaßnahme";"anderer Aufwandsträger")	ergibt	Ordnungsmaßnahme
Pick(3;"";"keiner";"auf Antrag";"aus päd. Gründen";"Klassenbildung";"Ordnungsmaßnahme";"anderer Aufwandsträger")	ergibt	aus päd. Gründen
Pick(6;"";"keiner";"auf Antrag";"aus päd. Gründen";"Klassenbildung";"Ordnungsmaßnahme";"anderer Aufwandsträger")	ergibt	anderer Aufwandsträger

Pick(Pos(GASTSCHUELER;"NEPKOA");"";"keiner";"auf Antrag";"aus päd. Gründen";"Klassenbildung";"Ordnungsmaßnahme";"anderer Aufwandsträger")

ergibt für jeden zulässigen Eintrag die richtige Entschlüsselung

Weitere Beispiele zur Entschlüsselung von Merkmalen

In gleicher Weise lassen sich auch andere Merkmale entschlüsseln, wobei der Entschlüsselungstext völlig frei gestaltet werden kann. Die Vorgaben des Programms in der Datensatzbeschreibung haben dabei keine kontrollierende Bedeutung, eine dauerhafte Veränderung dieser Vorgaben ist nicht möglich.

Entschlüsseln "Grund der ersten Wiederholung":

Pick(Pos(WDH1_GRUND;"PKFRAS");"";"Pflicht";"Krankheit";"freiwillig";"Rücktritt";"andere Schulart";"Schulartwechsel")

Entschlüsseln "Art der Erziehungsberechtigung 1":

Pick((Pos(ERZB1_ART;"/EL/MU/VA/VO/VW/PF/HL/K")+1)/3;"";"Eltern";"Mutter";"Vater";"Vormund";"Verwandter";"Pflegeeltern"; "Heimleiter";"keine")

Die Trennzeichen zwischen den einzelnen Schlüsseln verhindern die zufällige Bildung von gültigen Zeichenfolgen durch Zusammenhängen der einzelnen mehrstelligen Abkürzungen.

Zähler herstellen

Um in Berichten (*.AHR) zählen zu können, muß Pos() in Verbindung mit dem Merkmalsattribut AUFSUMMIEREN verwendet werden. Durch den Einsatz von Pos() wird aus der Zeichenkette "M" bzw. "W" ein numerischer Wert.

ALLE_KNABEN = Pos("M";GESCHLECHT) ergibt den Wert 1 bei Knaben, bei Mädchen 0

ALLE_MADL = Pos("W";GESCHLECHT) ergibt den Wert 0 bei Knaben, bei Mädchen 1

ALLE_KNABEN = GESCHLECHT="M" ergibt den Wert 1 bei Knaben, bei Mädchen 0

ALLE_MADL = GESCHLECHT="W" ergibt den Wert 0 bei Knaben, bei Mädchen 1

Der direkte Vergleich kann keine UND/ODER-Verknüpfung enthalten!

Wird in einem Bericht einem Merkmal das Attribut AUFSUMMIERT zugeordnet (siehe Berichtsgenerator), werden numerische Werte bis zum Berichtsende aufaddiert. POS() oder ein direkter Vergleich liefern den gewünschten numerischen Wert 1 oder 0.

ALLE_SCHÜLER = ALLE_KNABEN + ALLE_MADL liefert die Gesamtschülerzahl

weitere Zählerbeispiele:

ALLE_RK = Pos("RK";BEKENNTNIS) ergibt 1 bei Schülern mit dem Bekenntnis RK, sonst 0
 ALLE_DEUTSCH = Pos("D";STAAT) ergibt 1 bei Schülern mit der Staatsangehörigkeit D, sonst 0
 ALLE_GMD_Z = Pos("162000";GEMEINDEKENNZ) ergibt 1 bei Schülern aus der Gemeinde 162000, sonst 0

ALLE_JG_82 = Pos(Mid(GEBURTSDATUM;9;2);"82") ergibt 1 bei Schülern des Jahrgangs 82, sonst 0

ALLE_RK = BEKENNTNIS="RK" ergibt 1 bei Schülern mit dem Bekenntnis RK, sonst 0

ALLE_DEUTSCH = STAAT="D" ergibt 1 bei Schülern mit der Staatsangehörigkeit D, sonst 0

ALLE_GMD_Z = GEMEINDEKENNZ=162000 ergibt 1 bei Schülern aus der Gemeinde 162000, sonst 0

ALLE_JG_82 = Mid(GEBURTSDATUM;9;2)="82" ergibt 1 bei Schülern des Jahrgangs 82, sonst 0

In einen Bericht einen Dialog einfügen

Manche Angaben eines Berichts/Formulars, wie etwa das Ausstellungsdatum oder ein Abgabetermin, können nicht bei der Erstellung der Berichtsbeschreibung bestimmt werden, sondern lassen sich erst zum Zeitpunkt des Druckes festlegen. Für solche Fälle kann eine Dialogvariable in den Bericht eingefügt werden. Sie ist stets ein berechnetes Merkmal mit der Bezeichnung X1 ... X15. Daraus ergibt sich die Höchstzahl der möglichen Dialogfelder von 15.

Als Berechnung ist einzutragen:

X1 LST(8;1;#S1)

X2 LST(8;2;#S1)

X3 LST(8;3;#S1)

...

Wichtig ist der zweite Parameter, er gibt an, um welches X es sich handelt. Die Formulierung dient der Übergabe an die Berichtsbibliothek, die daran erkennt, daß ein Dialog abzufragen ist.