



www.allegro-c.de/allegro

Jahre und Zeiten

*Vier Jahreszeiten - und was dann ?
Was ist, wenn Herbst schon geht und Winter noch nicht kann ?
Kommen Tage
wo die Frage
nach der Uhrzeit keinen Sinn ergibt,
weil Mittag vor den Morgen, dieser vor die Nacht sich schiebt ?
Wenn grüner Regen warm auf weiße Blätter schneit,
dann ist die fünfte Jahreszeit..*

Was es nicht gibt, wie z.B. eine fünfte Jahreszeit, darüber kann man nichts Sinnvolles sagen. Wesentlich häufiger kommt im wirklichen Leben aber das Umgekehrte vor: es wird Unsinn erzählt oder geglaubt über Dinge, die es gibt. Nicht immer fällt das sofort auf.

Zu den ganz merkwürdigen Phänomenen gehört das Schattendasein, das alphabetische Register in den OPACs führen. Fast immer steht im Vordergrund ein Formular oder eine Maske und fordert zum Eingeben von Suchbegriffen auf. Wenn es überhaupt Register zum Blättern gibt, werden sie selten deutlich vorgestellt, bei Web-Katalogen noch seltener als sonst (sogar `catalog.loc.gov`, brandneu, hat keine!).

Bei den Web-Suchmaschinen kommen Register so gut wie gar nicht vor. Das hat zu tun mit der Methodik, mit der solche Geräte intern arbeiten. Natürlich sucht keine im echten Volltext der Dokumente - das würde Monate dauern. Gesucht wird in einem komprimierten, spezifisch aufbereiteten Surrogat, das nicht in einer für menschliche Betrachter geeigneten Weise angeordnet ist. Ohne Frage könnten aber gerade Suchmaschinen durch Register sehr gewinnen, weil sie mit unkontrolliertem Wortmaterial und ohne irgendwelche Ansetzungsregeln arbeiten müssen. Was könnte man beim Register-Browsing für Einblicke gewinnen, was für Zufallsfunde (Serendipity) könnte man machen! Für Bibliotheken wäre interessant, welchen Einfluss die Suchmaschinen auf die Bewusstseinsbildung und die Erwartungshaltung von Informationssuchenden haben, ob Unterschiede zu Katalogen empfunden werden, und wenn ja welche.

Um einmal ein Meinungsbild zu erhalten, wurde vor einigen Wochen in der E-Mail-Liste INETBIB eine Umfrage durchgeführt: die Frage war, ob man in einem OPAC alphabetische Register für unnützlich, entbehrlich, wichtig oder unentbehrlich halte, und es wurde gebeten, nur zu antworten, wenn Praxiserfahrung vorläge. Das Ergebnis war unerwartet deutlich: 54 von 63 Antworten sagten "wichtig" oder "unentbehrlich", *niemand* erklärte Register für unnützlich oder entbehrlich. "Ohne Register stochert man im Dunkeln", so brachte jemand die Ansicht der Mehrheit auf den Punkt. Nun haben zwar Benutzungsstudien gefunden, dass viele Benutzer nicht verstehen, nicht auf Anhieb jedenfalls, was ein Register denn wirklich ist. Soll man schließen, dass der Umgang mit alphabetischen Ordnungen eine Art Kulturtechnik ist, die eigens erlernt werden muss? Ein Formular mit ein bis drei Eingabefeldern dagegen - was könnte einfacher sein? Da sagt vermutlich auf Anhieb erst einmal keiner, er verstünde nicht, was das soll, denn Formulare hat jeder schon mal ausgefüllt. Das sieht schnörkellos direkt aus und suggeriert dem Uneingeweihten, dahinter stecke geballte Intelligenz, die dann die relevanten Dinge aussortieren und zu Tage fördern wird, ohne dass man das Geringste davon verstehen muss. Das grenzt an Aberglauben.

Denn andere Benutzungsstudien haben durch Protokollieren von Benutzersitzungen gefunden: es werden sehr oft Eingaben getätigt, die gar nicht zum Erfolg führen *können*, und die ganz klar zeigen, dass viele Nutzer keinen Schimmer haben (woher auch?), was sie vom System erwarten können. Für den Kenner ist es banal, dass man sehr leicht durch eine geringfügig falsche Eingabe ein falsches, unsinniges oder überhaupt kein Ergebnis bekommt, durch Einblick in ein Register aber sofort *sehen* könnte, was los ist. Das gefürchtete "Null-Treffer-Problem", das den Nutzer so frustriert, tritt beim Blättern in Registern gar nicht auf: man sieht, was es gibt und was nicht, und dass es unterschiedliche Schreibweisen gibt, auf die man nicht gekommen wäre; die Rechtschreibreform kann dies nur weiter verschärfen. Register, mit anderen Worten, stoßen einen mit der Nase darauf, dass man was falsch gemacht hat, und zwingen ganz offen zur Aufmerksamkeit und zum Mitdenken - vielleicht mag das nicht jeder. Das Ausfüllen kleiner Formularkästchen verlangt vom Nutzer scheinbar weniger, und die Antwort "Leider keine Treffer" kann ja auch bedeuten, dass die Bibliothek nichts hat - wenigstens ein schöner Grund zum Schimpfen.

Es scheint, dass der Sinn und Nutzen von Registern noch viel zu wenig untersucht und problematisiert worden ist. Man hat manchmal den Eindruck, dass die Sache als Geschmacks- oder Glaubensfrage unterbewertet wird. Kommt die Rede auf das Null-Treffer-Problem, werden dann gelegentlich schwerste Geschütze vorgeschlagen, etwa Fuzzy Searching, Wörterbuchsysteme, linguistische Analyse etc. Das Unsinnige daran ist: es wird der Eindruck erweckt, das System könne "intelligent" suchen, dabei wird nach wie vor im Trüben gefischt, nur mit einem anderen Netz - da bleiben manchmal ein paar mehr Fische drin hängen, aber immer auch viel mehr Müll. Kann es vertrauensbildend wirken, wenn Suchergebnisse gar nicht mehr nachvollziehbar sind? Beim uneingeweihten Nutzer werden zudem unerfüllbare Erwartungen geweckt, denn kein Programm kann wirklich mitdenken und den *Sinn* einer Anfrage verstehen. Das wahre Dilemma wird man so nicht los: die eigentlich notwendigen Erklärungen sind länger als die Geduld der Betroffenen, denen keine Intuition und keine Institution sagt, dass die Dinge schwierig sind. (Bei der Quantentheorie ist das z.B. anders.) Das Vorstellungsbild, das Nutzer sich machen und nach dem sie sich richten, wird zwangsläufig unvollständig, diffus, konfus, schlimmstenfalls unsinnig. Bibliotheken tendieren anscheinend eher dazu, die Probleme herunterzuspielen, um die Klientel nicht zu verprellen oder weil im Web alles einfach zu sein oder wenigstens zu scheinen hat.

Wenn es ein verbreitetes Unbehagen an alphabetischen Registern wirklich geben sollte, liegen dafür wohl kaum sachlich-sinnvolle Gründe vor. Es könnten schon eher psychologische sein. Vielleicht ist es ja so, dass das Sichtbare weit weniger fasziniert als das Verhüllte, das man nur erahnen kann aber nie ganz zu Gesicht bekommt. Im Finstern tastend stößt man, wer weiß, vielleicht urplötzlich auf Erregendes...

Momentan ist dessenungeachtet eine Arbeitsgruppe dabei, eingesetzt von der Konferenz für Regelwerksfragen, nüchterne Grundempfehlungen für die Indexierung von OPACs auszuarbeiten. Es ist zu hoffen, dass davon Denkanstöße ausgehen. Der Grund ist, dass solche Dinge in RAK bisher völlig fehlen und es deshalb in den OPACs an Einheitlichkeit mangelt. Wenn Doppelnamen oder Bindstrichwörter hier so und dort anders indexiert sind, usw. usf., fördert dies natürlich nicht die Begeisterung für OPACs und insbesondere nicht für deren Register.

Für *allegro* wurden 1989, vor Jahren und Zeiten, *frei parametrierbare* alphabetische Register zum zentralen Konzept gemacht. Man konnte sogar beim DOS-System *nur* über die Register zugreifen. Dabei waren zwar auch logische Kombinationen machbar, doch waren die Möglichkeiten begrenzt und, obwohl sehr leicht auszuführen, nicht so leicht zu vermitteln. Nach allem, was man jetzt weiß, sollte wohl ein OPAC-System beides bieten: Register *und* Suchbefehle. Dies wurde für *allegro* ab 1995 erreicht mit der neuen Programmgeneration: erst *avanti* und jetzt *a99* und *alcarta*. Doch wie die Zeiten auch kommen und die Jahre auch gehen: die Register, die bleiben bestehen.

a99 : Einstieg in die Windows-Klasse (siehe auch www.allegro-c.de/allegro/alcarta)

Die Anleitung in Nummer 53 war gedacht für *Umsteiger*, die schon mit dem DOS-System gearbeitet hatten. Mittlerweile gibt es immer mehr *Einsteiger*, denen ein Systemverwalter ein Icon auf den WinNT-Bildschirm platziert, und dann werden sie konfrontiert mit **a99**, vom DOS-Programm kriegen die nichts zu sehen. Da muss eine etwas andere Anleitung her. Hier kommt sie. Im Anschluss, ab Seite 8, gibt es dann eine ebenfalls neue Einführung in das Grundkonzept des **allegro**-Systems. Diese wendet sich nicht nur an Einsteiger, sondern auch an Systemverantwortliche, die nicht ganz an der Oberfläche bleiben wollen.

Die Beschreibung gilt für die aktuelle Version, die schon wieder etwas mehr kann als die auf der CD-ROM. Anwender mit FTP-Zugang können jederzeit die neueste Version herunterladen. (Datei A99UPD.EXE auf Verzeichnis AC15/A99.) Wir gehen davon aus, dass **a99** installiert ist und funktioniert. Was dabei zu beachten ist, stand in Nr. 54.

Der erste Eindruck vom neuen Programm ist: "Himmel, braucht man so viele Sachen? Diese ganzen Knöpfe und alles ...!" Doch das meiste, was **a99** fertig anklickbar zu bieten hat, das gibt es beim DOS-Vorgängerprogramm PRESTO auch schon. Man muss da nur die richtigen Tasten wissen (oder **F1**, dann kommt die Liste der Tastenbefehle). Datenbanksysteme brauchen nun einmal viel mehr Funktionen und Eigenschaften als man am Anfang ahnt. Wenn das alles offen ausgebreitet wird, ist man im ersten Moment doch etwas überwältigt. Nicht jeder braucht alles, das ist klar, und keiner alles auf einmal.

Aber was ist denn das Wichtige?

Hier eine Momentaufnahme aus einer Arbeitssitzung an einer *bolero*-Datenbank.



Gezeigt wird ein Datensatz (früher sagte man "Titelaufnahme") für eine Aufzeichnung von Beethovens "Fünfter". Der Datensatz, der ausgewählt wurde, der im Anzeigefeld zu sehen ist und mit dem man sich aktuell gerade beschäftigt, wird immer der "aktuelle Datensatz" genannt. Wie das Suchen, Finden und Auswählen geht, dazu kommen wir noch.

Es gibt vier Bereiche, in denen man etwas tun kann. Im Bild oben sind die Namen dieser Bereiche zu sehen: (man sieht sonst diese Namen nicht)

Anzeigefeld

Meistens steht hier drin ein einzelner Datensatz. Aber auch Hilfetexte werden hier eingeblendet.

TIP: Mit Alt+z kann man umschalten zwischen der Normalanzeige und der kategorisierten (internen) Anzeige. (PRESTO: F5)

TIP: Im Anzeigefeld kann man auch herumschreiben, aber was man dort schreibt, wird nicht in der Datenbank gespeichert!

TIP: Der Button mit dem Drucker (unten neben [Find]) schickt das gesamte Anzeigefeld an den Drucker, mit allem was gerade in dem Moment dort steht. Mit Alt+c kann man auch markierte Teile in die Windows-Zwischenablage übernehmen. Wenn man kein Textprogramm hat, könnte man sogar das Anzeigefeld zum Schreiben kleiner Texte benutzen. Auch speichern kann man solche Texte, und zwar über das Menü "Datei", Unterpunkt "Anzeige speichern als..." Gespeichert wird dann im Format RTF, das man auch mit Word einlesen kann.

Auswahlfeld

Im Normalfall stehen hier zur Auswahl die Kategorien des aktuellen Satzes. Im Bild wurde gerade die Kategorie #81 ausgewählt und mit [Enter] ins Schreibfeld kopiert, wo man diese Kategorie dann bearbeiten kann.

Mit der Taste [Entf] kann man einzelne Felder sofort löschen. (Drauf klicken, dann [Entf] und weg ist es.)

TIP: Im Reservespeicher findet man die Kategorien wieder, die man gerade mit [Entf] aus dem Datensatz gelöscht hatte.

So geht's: Alt+r (dann sieht man die gelöschten Kategorien), Kategorie anwählen, dann Alt+k zum Kopieren, und mit Alt+r zurück zur Anzeige des aktuellen Satzes; man sieht, dass die Kategorie wieder da ist.

Im Auswahlfeld können auch erscheinen: Die Liste der Ergebnismengen der laufenden Sitzung (Alt+e), die Kategorien des Reservespeichers (Alt+r), und das Datenschema (Liste der erlaubten Kategorien, Alt+y)

Schreibfeld

Hier bearbeitet man Daten und gibt neue ein. Im Bild wurde im Auswahlfeld die Kategorie #81 angewählt und dann [Enter] gedrückt; die #81 samt Inhalt erscheint im Schreibfeld, und in der Befehlszeile wird als Hilfe eingeblendet, um was für eine Kategorie es sich handelt. *Wichtig:* Man muss [Enter] drücken, um die Eingabe dem Programm zu übergeben. Wenn man etwas geschrieben hat und dann mit der Maus woanders hinklickt, bleibt die Eingabe unwirksam.

TIP: Eine andere Arbeitsweise zum Eingeben und Bearbeiten ist möglich, wenn "Formulare" für die Datenbank eingerichtet wurden (siehe weiter unten). Dafür ist der Button [Formulare] da.

TIP: Drei wichtige Sonderzeichen sind leicht einzugeben: $\text{^} = \text{AltGr}+2$, $\text{¶} = \text{Strg}+\text{t}$, $\text{^} = \text{AltGr}+3$ (Haček)

Auch Phrasen gibt man hier ein: z.B. **p s Shakespeare** eingeben, dann hat man "Shakespeare" unter Phrase gespeichert. Mit Eingabe von \s kann man an jeder Stelle diese Phrase wieder abrufen. (Mehr dazu siehe PHRASE.TXT)

Befehlszeile (man könnte auch "Suchbefehlszeile" sagen oder "Anfragezeile")

In diese Zeile gibt man direkte Suchbefehle ein. Gerade vorher wurde offenbar dieser Befehl eingegeben:
per beethoven? and tit sinfonie?

Denn unten auf der langen Schaltfläche sieht man diesen Befehl, und dass dabei 73 Treffer herauskamen. Der aktuelle Satz ist die Nummer 32 in dieser Treffermenge. Sehen Sie, wo das steht? Neben der Ergebnis-Schaltfläche sind zwei Pfeile: damit geht man zum nächsten oder vorigen Satz der Ergebnismenge, hier also zu Nummer 31 bzw. 33.

GeheimTIP: Wenn man statt Befehl nur einen Punkt eingibt, sieht man in der Anzeige die zum aktuellen Satz gehörigen Registerinträge. (Wie bei der Funktion F7 von PRESTO.) Das geht auch bei *alcarta*.

Der Focus [ein Windows-Begriff, keine *allegro*-Erfindung]

Nicht nur ein Nachrichtenmagazin, auch ein wichtiger Begriff der Windows-Sprache! Der sog. Focus ist kein Symbol oder Objekt, sondern dasjenige Feld "besitzt den Focus", so sagt man, in dem sich gerade der Cursor befindet. Tasten (Zeicheneingaben und Cursorbewegungen) wirken sich nur in dem Feld aus, das gerade den Focus besitzt. Jederzeit kann man den Focus in ein anderes Feld setzen, indem man mit der Maus hineinklickt. In die Auswahl- und Schreibfelder kann man auch mit Alt+c (immer abwechselnd hin und her) springen, in die Befehlszeile mit Alt+f. Auch mit [TAB] kann man herumspringen.



Noch Fragen offen

Dafür ist der Fragezeichen-Button da. Der bringt einen Hilfetext zur Anzeige, der speziell für die Datenbank gemacht wurde, mit der man gerade arbeitet. Der Systemverwalter kann diesen Text frei gestalten, um die Arbeit mit dieser speziellen Datenbank besonders zu unterstützen. Vielleicht hat er einige hilfreiche Links darauf untergebracht, die z.B. automatisch Erfassungformulare aufrufen oder zu besonderen Datensätzen, Ergebnismengen oder Registerstellen hinführen. Eine gut gemachte Hilfeseite kann den Lernbedarf drastisch reduzieren. Ein "Link" sieht so aus wie auf einer Web-Seite: blau und unterstrichen. Es können aber ganz andere Dinge dahinter stecken. Deshalb wird so etwas bei *a99* und *alcarta* nicht "Link" oder "Hyperlink" genannt, sondern "Flip". Ganze Befehlsfolgen, kleine Programme also, können sich hinter einem Flip verbergen. Solche Programmchen werden dann FLEX genannt, weil sie die Flexibilität vermehren.

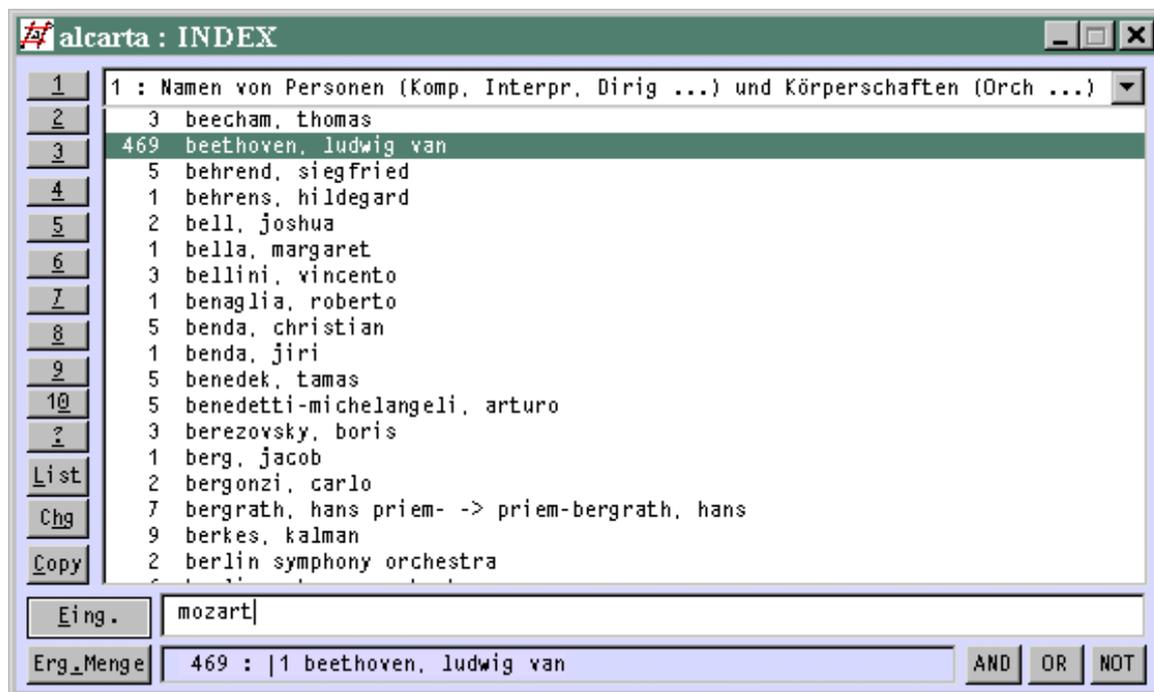
Für Systemverwalter: In *news 54* wurden die FLEX-Befehle beschrieben, s.a. S.15 in dieser Ausgabe. Die aktuelle Beschreibung ist in der Datei FLEX.TXT zu finden, die im Download-Paket von *a99* enthalten ist, noch nicht auf der CD.

Suchen und Blättern

Was man oben in der Abbildung sieht, ist längst nicht alles, was das Programm zu bieten hat. Es gibt vor allem noch drei weitere Fenster, die dem Nutzer beim Suchen und Auswählen helfen:

1. Das Indexfenster (ein- und ausschalten mit Alt+i oder mit dem Button [Index])

Das sieht z.B. so aus, wenn man gerade im Personenregister "Beethoven" aufgeblättert hat:



Im Eingabefeld steht "mozart", offenbar wurde das gerade eingetippt, um zu den Einträgen von Wolfgang Amadeus zu gelangen. *Wichtig:* Sobald man einen Buchstaben tippt, erscheint der im Eingabefeld - man muss also nicht erst hinein klicken. Mit den 10 nummerierten Knöpfen an der linken Seite schaltet man zwischen den 10 Registern um, die es maximal gibt. Die Kopfzeile enthält eine Liste (auf das Dreieck klicken) der vorhandenen Register. Umgeschaltet wird immer auf dieselbe Alphabetstelle, was praktisch ist, wenn man z.B. zuerst das falsche Register gewählt hatte. Das Umschalten funktioniert auch, wie beim alten PRESTO, mit Alt+Ziffer.

Bei Doppelklick auf eine Zeile (oder [Enter]) passiert folgendes:

- Die Trefferzahl ist 1 : der zugehörige Satz wird in das Anzeigefeld geholt. Oder:
- Es gibt mehr als 1 Treffer: man bekommt eine **Kurzliste** der zugehörigen Sätze (auch "Ergebnismenge" genannt; mehr dazu weiter unten), und der Balken geht eine Zeile tiefer.

TIP: Cut-and-Paste geht nicht in diesem Fenster, aber: Balken auf die erste Zeile setzen, die man haben will, ins Eingabefeld eine Zahl schreiben, z.B. 50, dann [List] drücken. Schwupp, hat man die 50 nächsten Zeilen ab dem Balken im Anzeigefeld; dort kann man markieren und Cut-and-Paste machen, also den Registerausschnitt als Text übernehmen.

Weitere nützliche Funktionen:

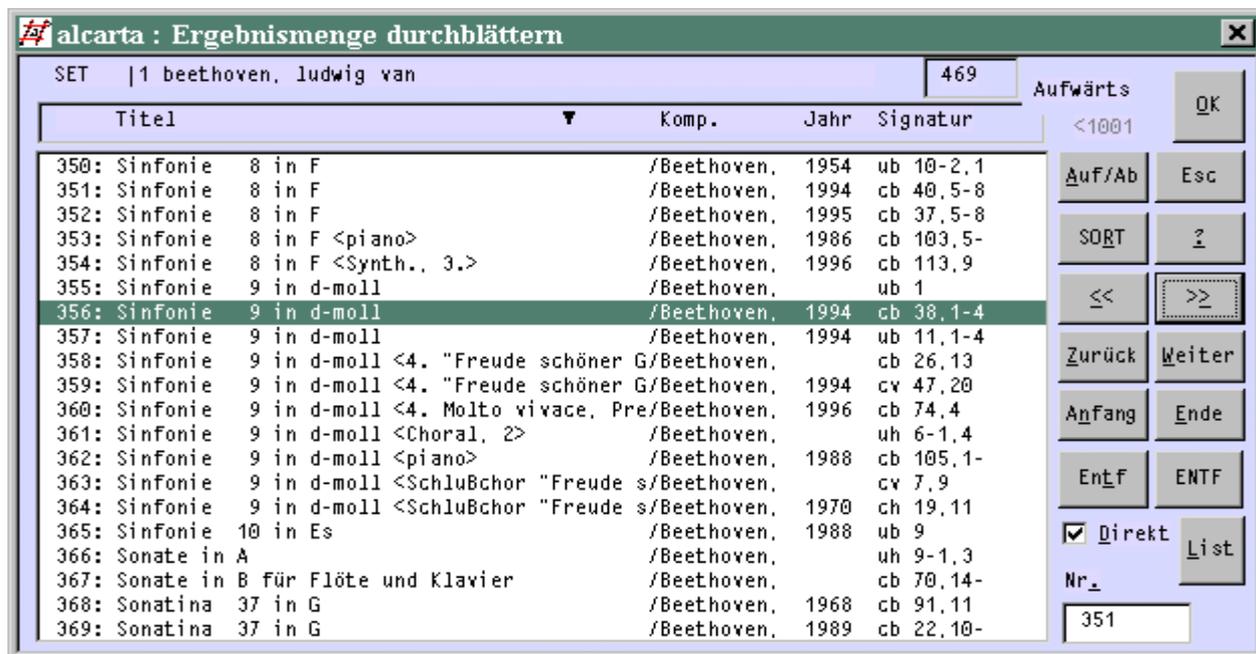
- Der Button [Copy] kopiert die gerade gewählte Zeile in das Schreibfeld. Wenn aber von einem Formular aus in den Index gesprungen wurde, landet die Kopie in dem Formularfeld, von dem man ausging.
- Mit [Chg] wird die angewählte Zeile nach unten ins Eingabefeld kopiert, wo man sie ändern kann, um dann an eine andere Stelle im Register zu springen, wenn das Register sehr lange und zahlreiche Einträge hat, wie z.B. ein Titel- oder Serienregister.
- Ganz unten sieht man, welches Ergebnismenge gerade vorher gebildet wurde. Mit den Buttons [AND] [OR] und [NOT] kann man weitere Registerzeilen mit der Ergebnismenge verknüpfen lassen. Statt dessen können auch, wie beim alten PRESTO, die Tasten +, / und - benutzt werden.
- Mit [Esc] oder Alt+i lässt man das Indexfenster verschwinden, mit Alt+i geht es sofort wieder auf, an derselben Stelle.

Trunkierung: Drücken Sie mal im Register 1 (Personennamen) auf das Komma oder im Register 5 (Serien) auf das Semikolon: alle Einträge werden dann an dem Satzzeichen abgeschnitten. Eine andere Möglichkeit: Geben Sie im Register 1

"shakesp?" ein, also ein Fragezeichen am Ende, dann werden alle Zeilen auf diese Länge gekürzt. So erhält man einerseits ganz neue Überblicke über ein Register und kann andererseits sehr leicht Ergebnisse zusammenfassen.

Die Kurzliste

Das ist ein besonderes Fenster, in dem die Übersicht der aktuellen Ergebnismenge zu sehen ist. Setzt man im Beispiel oben den Balken auf Beethoven und drückt dann Enter, kommt folgende Kurzliste:



Diese Liste kann man sich an jeder sinnvollen Position sortieren lassen. Man verschiebt oben über den Titeln entlang das kleine Dreieck mit den Buttons [<<] und [>>] nach links oder rechts. , dann auf [SORT] drücken.

TIP: nur einmal auf [<<] oder [>>] klicken, dann die [Enter]-Taste niederhalten, das Dreieck verschiebt sich dann sehr schnell; mit Maus allein ist es mühsamer.

In diesem Fall ist nach Titel sortiert worden, dann wurde der Abschnitt der Sinfonien aufgeblättert.

Mit [Auf/Ab] dreht man die Sortierrichtung um. Bei Jahreszahlen ist die umgekehrte Reihenfolge sinnvoll, dann hat man die neuesten Einträge oben.

Für Kenner: Mit Alt+s sortiert man nach der internen Satznummer.

Mit [F1] kommt eine Hilfeseite, die den Rest erklärt.

Noch ein **TIP:** eine Zahl in das Feld "Nr." eintragen, dann [List]. Wie beim Indexfenster erhält man, schwupp, diese Anzahl Zeilen, ab der angewählten Zeile, in das Anzeigefenster.

Das Find-Fenster

Wer noch nicht weiß, wie man Suchbefehle formulieren muss, kann sich das Find-Fenster aufklappen. Man drückt auf den blau-gelben [Find] Button und sieht dieses: (oder Alt+f)

(Auf so etwas mussten und müssen DOS-Anwender verzichten, was aber lange Jahre durchaus gut funktionierte)

Der Nutzer hat in diesem Suchformular verlangt, im Personenregister nach "beethoven" zu suchen und zugleich nach "sinfonie" im Register der Titel- und Schlagwörter. Außerdem sollen nur Einträge von CD-Aufnahmen in die Ergebnismenge aufgenommen werden (also keine Kassetten oder anderen Tonträger).

Damit das Find-Fenster funktioniert, müssen in den Indexparametern der Datenbank die Namen der Register eingetragen sein sowie die Bezeichnungen der Einschränkungen (sog. Restriktionen), wenn solche eingerichtet sind. Bei älteren Datenbanken ist das unter Umständen noch nicht der Fall, dann wird man keine Registernamen zu sehen bekommen, logisch. Ein Fall für den Systemverwalter (siehe Handbuch Kap. 10.2.1, S.176 und *allegro news* Nr.54).

TIP: Nach Benutzung des Formulars hat man den daraus entstandenen Befehl in der Befehlszeile stehen. Dort kann man Änderungen vornehmen und den Befehl erneut ausführen lassen.

TIP: Wenn es keine Registernamen gibt, kann man trotzdem Find-Befehle in die Befehlszeile eingeben: statt per beethoven? and tit sinfonie? könnte man auch eingeben: |1 beethoven? and |3 sinfonie?

TIP: Hat man schon eine Ergebnismenge, kann man den "Volltext" aller Datensätze (nicht der Dokumente!) durchsuchen lassen, indem man in die Zeile ganz unten eine Zeichenkette eingibt. Groß-/Kleinschreibung muss dabei beachtet werden.

Eingeben und Bearbeiten

Bearbeiten vorhandener Daten

Einen schon vorhandenen Satz kann man jederzeit verändern. Im Auswahlfeld links (siehe S. 2) ist die Übersicht der Datenfelder des Satzes zu sehen. So geht's (siehe auch unten unter „Formulare“)

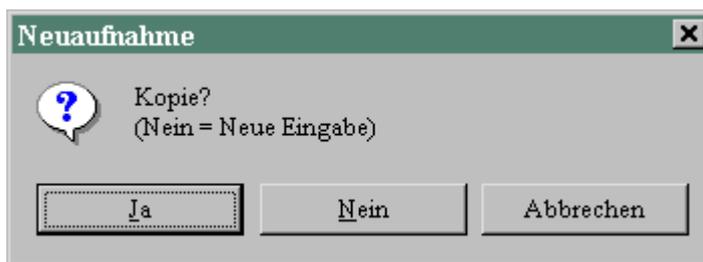
- Man wählt das zu bearbeitende Feld an (Balken drauf und [Enter]). Dann wird der Text in das Schreibfeld kopiert und lässt sich dort beliebig bearbeiten, auch mit Cut-and-Paste.
- Nochmals [Enter], und die Kategorie wird zurück kopiert in den Datensatz, zuerst aber geprüft, und zwar feldspezifisch: z.B. ist das erste Titelwort ein Artikel, ist die Jahreszahl plausibel, die ISBN-Prüfziffer korrekt, usw. usf.

TIP: Auch die Kategoriennummer ist änderbar. Dann allerdings sind plötzlich zwei Kategorien da, denn das alte Datenfeld bleibt erhalten. Was tun? Das überflüssige anwählen, [Entf]-Taste drücken, weg ist es.

Satz fertig bearbeitet? Dann [Speichern] (siehe auch unten unter „Offline-Speicher“)

Eingeben neuer Daten

Das Hauptgeschäft beim Katalogisieren ist das Erfassen neuer Daten. Am schnellsten geht es mit der Übernahme geeigneter Fremddaten, aber das ist für diese Einführung ein zu weites Feld. Man muss immer auch neue Sätze von Hand eingeben können, und darauf konzentrieren wir uns hier. Dafür ist der Button [Neusatz] da. Wird der betätigt, kommt die Frage:



Kopieren!

Oft spart es Arbeit, wenn man zuerst einen Satz aufblättert, der eine Anzahl übernehmbare Kategorien enthält. Besonders ist das der Fall, wenn eine Neuauflage von einem bereits katalogisierten Titel kommt. Wird hier mit [Ja] geantwortet, hat man einen neuen Datensatz vor sich, den man nur noch durch ein paar Korrekturen anpassen muss. Ist kein geeigneter Satz zum Kopieren vorhanden, wählt man hier [Nein]. Es wird dann im Auswahlfeld die Abfrageliste gezeigt. PRESTO-Anwender kennen dieses Konzept. Hier sieht es etwas anders aus, aber das Prinzip ist dasselbe: es kommt eine Reihe von Abfragen, die man durch die richtigen Eingaben im Schreibfeld beantworten muss.

Wenn Formulare eingerichtet wurden, kann man jederzeit den Button [Formulare] betätigen, und zwar in beiden Fällen, bei Neueingaben und bei Bearbeitungen. Ein Formular einschalten heißt **nicht**, dass dabei sofort ein neuer Satz entsteht, sondern die Eingaben der Formulare gehen alle in den aktuellen Satz über. Auf der nächsten Seite ist ein Beispiel für ein Formular zu sehen. Neuaufnahmen beginnen immer nur mit Druck auf [Neusatz] .

Hilfe, wo ist denn meine Neuaufnahme?

Jederzeit, auch während einer Neuaufnahme, kann man [Index] drücken, blättern und sich andere Daten anschauen. Schwupp, verschwindet aber dann die Neuaufnahme vom Schirm - wo ist sie geblieben, wie kriegt man sie zurück? Keine Panik: sie landet im *Offline-Speicher*. (Den gab es bei PRESTO nicht, da musste man einen Datensatz zuerst speichern, vorher konnte man sich keine anderen ansehen.) Der Offline-Speicher öffnet sich mit Alt+q oder Druck auf die Schaltfläche unten rechts unter der Ergebnismengen-Fläche. "Q : Daten in Bearb." steht drauf. Der letzte Eintrag in der Liste, die dann kommt, ist die gesuchte Neuaufnahme. Also: **Alt+q** [Ende] [Enter], schon ist man wieder drin in der Neuaufnahme.

Offline-Speicher?

In dem Offline-Speicher sammeln sich alle Sätze, die man während einer Sitzung "anfasset", d.h. an denen man irgendeine Änderung macht, nicht nur die Neuaufnahmen. Die bearbeiteten, aber noch nicht wieder gespeicherten Sätze erkennt man an der Kennung EDT vor dem Titel. Also: Anders als bei PRESTO muss man bei diesem Programm keineswegs einen Satz zuerst abspeichern, bevor man sich den nächsten vornehmen kann, sondern jederzeit kann man zu anderen Sätzen und zu Neuaufnahmen übergehen. Wird das nicht total unübersichtlich? Und wie leicht kann man am Ende das Speichern überhaupt vergessen! Das verhindern ein paar Zusatzfunktionen im Menü "Extras": Die Funktion "Neu, noch nicht gesp." zeigt die Liste der Neuaufnahmen, die noch nicht gespeichert wurden, und "IN ARBEIT befindl. Sätze" zeigt die geänderten, aber ebenfalls noch nicht gespeicherten Sätze. Die zu einem korrigierten Satz gehörenden Indexeinträge, das ist logisch, ändern sich erst, wenn der Satz wieder gespeichert wird.

Und am Ende einer Sitzung? Muss man da zuerst diese Sätze alle durchgehen und speichern, bevor man raus kann aus dem Programm? So umständlich ist das nicht, sondern das Programm merkt, dass noch solche unerledigten Sachen da sind. Es bringt die Frage: "Sollen die neuen und bearbeiteten (noch nicht gespeicherten) Daten gespeichert werden?" Bei der Antwort "Ja" macht es das dann alles selber, aber man kann auch mit [Abbrechen] dann nochmal zurück und diese Daten nochmals durchsehen, wenn man nicht sicher ist. Einzelne veränderte Sätze kann man mit dem Button [Wechseln] wieder unwirksam machen. Dann wird das Programm die Änderungen nicht speichern. Wer nichts dem Zufall überlassen will (Stromausfall!), kann natürlich vor jedem Übergang zu einem anderen Satz auf [Speichern] drücken.

Formulare

kann man für jede Datenbank einrichten. Sie sehen hier ein Formular, das für eine *bolero*-Datenbank eingerichtet wurde:

The screenshot shows a DOS-style window titled "a99 : Formular-Eingabe/Bearbeitung". The form contains the following data:

- Form: [Musikstück]
- Komponist: Beethoven, Ludwig van
- Musiktitel: Symphony 3 in E-flat "Eroica"
- Einheitstitel: Sinfonie 3 in Es "Eroica"
- Opus: 55
- Dirigent: Masur, Kurt
- Interpret: (empty)
- Orchester: Gewandhaus-Orchester Leipzig
- Sachgruppe: sy
- Jahr [Aufnahme]: 1975
- Sätze: 1: Allegro con brio 15:24|2: Marcia funebre: Adagio assai 16:05|3: Scherzo: All
- Zeitdauer: 49:31
- Tonart: Es-dur
- Signatur: cb 100,5-8

Buttons at the bottom right include Copy, Ende, Index, Reserve, and Get.

Auch hier muss jede eingegebene, bearbeitete oder gelöschte Kategorie mit [Enter] bestätigt werden, sonst passiert nichts. Mit [Index] springt man in die Register; dort gibt es zum Übernehmen von Angaben einen [Copy]-Button (s. oben). Über die Auswahlzeile ganz oben kann man jederzeit eines der anderen Formulare öffnen; die Daten gehen alle in den aktuellen Satz. Mit [Esc] verläßt man das Fenster, beendet also die Formulareingabe. Das Feld, in dem gerade der Cursor ist, wird nicht mehr übernommen. Anders beim Button [Ende]: da wird es übernommen.
TIP: Gibt man in ein Feld nur einen Punkt ein und dann [Enter], wird das Feld aus dem vorigen Datensatz übernommen (genauer: aus demjenigen, der in der Anzeige stand, als man [Neusatz] drückte). Dasselbe macht [Reserve].
TIP: Verschieben Sie das Fenster so nach rechts, daß das Auswahlfeld zu sehen ist.

Eingabe abbrechen

Mit dem Button [Deaktivieren] kann man eine Erfassung komplett abbrechen, egal ob es eine Kopie oder eine Neu-Erfassung ist.

Sitzung unterbrechen

Im DOS-System noch völlig undenkbar: eine laufende Sitzung kann man verlassen, ohne erst die Änderungen zu speichern, und später neu einsteigen und da weitermachen, wo man aufgehört hatte. "Wollen Sie die Sitzung später fortsetzen?" Diese Frage kommt vor dem Verlassen des Programms. Antwortet man mit [Ja], wird der gesamte aktuelle Zustand gesichert, einschließlich Ergebnismengen und Offline-Speicher. Es kommt aber in jedem Fall die Zusatzfrage: "Sollen die neu bearbeiteten (und noch nicht gespeicherten) Daten gespeichert werden?". Mit [Ja] kann man hier erreichen, dass wenigstens

die Bearbeitungen und Neuaufnahmen gespeichert werden. Für die nächste Sitzung werden dann immerhin die Ergebnismengen aufbewahrt, von den bearbeiteten Sätzen aber auch noch die Originalformen, die man mit [Wechseln] sogar nach dem Neustart wieder aktivieren kann!

Muss man wissen, was dahinter steckt?

Hintergrundwissen - braucht man sowas heute noch? Ist nicht eine Software einfach out, die nicht alle Funktionen intuitiv erfassbar und durch wenige Mausklicks erreichbar macht, und für die man sogar noch ein Handbuch benötigt? Man könnte dazu die eine oder andere kulturkritische oder gar philosophische Bemerkung machen, das jedoch soll hier nicht geschehen. Man muss wohl nüchtern feststellen, dass komplexe Werkzeuge ohne Kenntnisse und ohne Einarbeitung nicht angemessen genutzt werden können. Dies gilt auch für Windows im Allgemeinen, für Word und Excel im Besonderen, auch wenn es nicht jedem bewusst wird und die Industrie diese Erkenntnis nicht fördert. Aber das ist hier nicht das Thema.

Auf den nachfolgenden Seiten wird ein Versuch gemacht, die Grundlagen von *allegro* in neuer Weise zu vermitteln. Wer bereits als "Nur-Anwender" mit dem System arbeitet, hat sicher dabei den einen oder anderen Aha-Effekt, aber auch Systemverwalter, die nur mit anderen Datenbanken vertraut sind und *allegro* als exotisch empfinden, können hoffentlich profitieren.

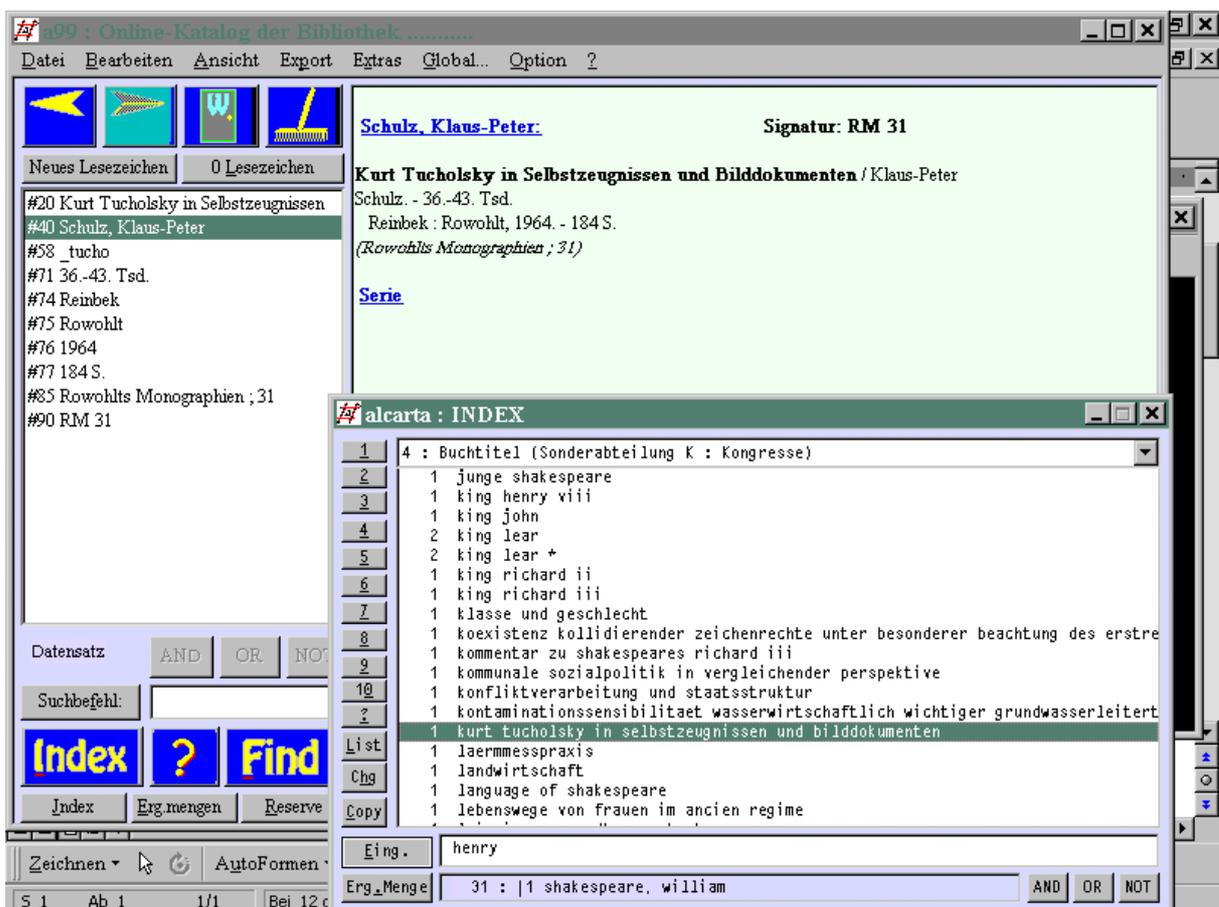
Zielgruppe dieser Seiten (bis S. 14) sind ungefähr alle, denen das Wort "Parametrierung" noch nicht viel sagt. Nicht alles wird "ankommen" oder "hängen bleiben", aber wer wird das bei solcher Materie erwarten.

Daten kann man nicht sehen, das ist das Problem.

Sonst wäre es viel leichter, darüber zu reden.

Aber man sieht sie doch auf dem Bildschirm? Da sieht man nur Leuchtspuren eines kompliziert herumschwirrenden Elektronenstrahls, weiter nichts, hervorgerufen durch den Ablauf von Programmen. Das *sind* nicht die Daten.

Das Bild, das man auf dem Schirm erblickt, *ist* nicht die Datenbank. Das Bild wird gezeichnet von einem Programm, das auf die Datenbank zugreift. Zum Beispiel könnte es ein Bild wie dieses zeigen:



Hier sind Angaben zum Buch "Kurt Tucholsky in Selbstzeugnissen und Bilddokumenten" auf drei verschiedene Arten zu sehen: im Registerfenster unten rechts sieht man nur den Titel und diesen vollständig in Kleinbuchstaben, oben im Anzeigefenster sieht man so etwas wie ein Literaturzitat oder eine Katalogkarte, und links eine Liste von Datenelementen mit Nummern wie #20 oder #74, aus denen man nicht so recht schlau wird. Ist das alles so gespeichert, wie man es hier sieht?

Vor einer Antwort noch ein paar andere Bilder.

Ganz anders als der Windows-Bildschirm sieht ein DOS-Bildschirm aus:



```
MS-DOS - PRESTO
8 x 12
database: CAT file#: 1 record#: 104 U15 length: 217. bytes free: 0
Schulz, Klaus-Peter: RM 31
Kurt Tucholsky in Selbstzeugnissen und Bilddokumenten / Klaus-Peter Schulz. -
36.-43. Tsd.
Reinbek: Rowohlt, 1964. - . - 184 S.
<Rowohlts Monographien ; 31>

>>> tucholsky, kurt
input file # 1 <F10> : execute global exchange
-> = back to index F1 = HELP
```

Dasselbe Buch in einer anderen Datenbank, anders gespeichert? Erfahrene Anwender wissen, dass man auf diesem Bildschirm die Taste **F5** drücken kann, und dann sieht man wieder etwas anderes:



```
MS-DOS - PRESTO
8 x 12
database: CAT file#: 1 record#: 104 U15 length: 217. bytes free: 0
#20 Kurt Tucholsky in Selbstzeugnissen und Bilddokumenten
#40 Schulz, Klaus-Peter
#58 _tuchol_Tucholsky, Kurt
#71 36.-43. Tsd.
#74 Reinbek
#75 Rowohlt
#76 1964
#77 184 S.
#85 Rowohlts Monographien ; 31
#90 RM 31
#9y Normsatz Person, Anwendung

>>> tucholsky, kurt
input file # 1 <F10> : execute global exchange
-> = back to index F1 = HELP
```

Wird hier in eine ganz andere Datenbank geblickt, und sind darin abermals zwei verschiedene Beschreibungen desselben Buches gespeichert?

Solche Vermutungen sind alle falsch. Nochmals: was man sieht, ob Windows oder nicht, das *ist* nicht die Datenbank, es ist ein Bild, erzeugt von einem Programm, weiter nichts. Das Programm selbst aber ist auch nicht die Datenbank! Es ist ein Hilfsmittel, ein Werkzeug, so etwas wie ein Mikroskop, mit dem man in die Datenbank hineinschauen kann.

Bei allen Beispielen handelt es sich um ein und dieselbe Datenbank, aber man schaut mit zwei verschiedenen Programmen, hinein, durch zwei Mikroskope also, und jedes Programm kann mehr als ein Bild vom selben Objekt zeigen.

Damit ist ein beliebtes Missverständnis aufgeklärt: die DOS-Version und die Windows-Version, das sind *nicht* zwei verschiedene Datenbanken! Es sind nur zwei verschiedene Programme, die auf den Bildschirmen gleichzeitig unterschiedliche Bilder vom selben Objekt erscheinen lassen.

(Wohlgemerkt: wir sprechen von *allegro*! Bei anderen Systemen mag manches anders sein.)

Noch anders gesagt: die Daten sind nicht irgendwie im Programm gespeichert oder damit fest verbunden (auch solche Ansichten kann man hören!), sie stehen in Dateien, die von den Programmen völlig getrennt sind. Eben deshalb können ja verschiedene Programme gleichzeitig zugreifen. Deshalb kann man auch eine neuere Programmversion einfach über die alte kopieren, ohne den Daten zu schaden.

Was *ist* nun aber wirklich gespeichert in der Datenbank? Im Fall unseres Beispiels sieht das so aus:

```
20 Kurt Tucholsky in Selbstzeugnissen und Bilddokumenten
40 Schulz, Klaus-Peter
58 _tucho
71 36.-43. Tsd.
74 Reinbek
75 Rowohlt
76 1964
77 184 S.
85 Rowohlts Monographien ; 31
90 RM 31
9y Normsatz Person, Anwendung
```

Das ist ein **Datensatz**. Mehr wurde nicht eingegeben und mehr steht da nicht drin, auch unter Windows nicht.

Eine Besonderheit: Im Feld 58 steht "_tucho". Das ist eine Verknüpfung zu einem Namensstammsatz, in dem der tatsächliche Name steht: "Tucholsky, Kurt". Statt des Kürzels "_tucho" könnte da auch eine Nummer stehen. Oben im zweiten DOS-Bild sieht man, wie das Programm diesen Namen automatisch einblendet; gespeichert ist er nur im Stammsatz.

Ein Datensatz ist noch keine **Datei**, sondern er ist (bei *allegro*) zusammen mit vielen anderen Datensätzen in einer Datei enthalten - so wie ein Katalogzettel mit vielen anderen in einer Schublade oder ein Formular mit vielen anderen in einem Ordner. Die Schublade oder der Ordner, *das* ist die Datei. (Im Englischen ist das leichter zu verstehen: das Wort "file" bedeutet sowohl "Datei" wie auch "Ordner", aber auch "Kartei".)

Die anderen Bilder, die man vom Datensatz sieht, zum Beispiel

Schulz, Klaus-Peter:	Signatur: RM 31
Kurt Tucholsky in Selbstzeugnissen und Bilddokumenten / Klaus-Peter Schulz. - 36.-43. Tsd. Reinbek : Rowohlt, 1964. - 184 S. (Rowohlts Monographien ; 31)	
Serie	

oder

Schulz, Klaus-Peter:	Signatur: RM 31
Kurt Tucholsky in Selbstzeugnissen und Bilddokumenten / Klaus-Peter Schulz. 36.-43. Tsd. Reinbek : Rowohlt, 1964. - 184 S. (Rowohlts Monographien ; 31)	

aber auch die Zeile im Register:

- 1 kommunale sozialpolitik in vergleichender perspektive
- 1 konfliktverarbeitung und staatsstruktur
- 1 kontaminationssensibilitaet wasserwirtschaftlich wichtiger grundwasserleitert
- 1 **kurt tucholsky in selbstzeugnissen und bilddokumenten**
- 1 laermmesspraxis
- 1 landwirtschaft

- 1 language of shakespeare
- 1 lebenswege von frauen im ancien regime

das machen alles nur die Programme. Diese Bilder sehen sehr viel anders aus der Datensatz mit den Nummern, deshalb fragt man sich am Anfang doch, wo sie herkommen. Wie "weiß" das Programm, dass bei der Nummer 85 ein Serientitel steht? Wie weiß es, wo dieser in den anderen Bildern hingehört, und dass er in Klammern gehört - denn die Klammern stehen gar nicht im Datensatz? Sind diese Einzelheiten alle fest in ins Programm eingebaut? Keineswegs! Die *Programme*, ob DOS oder Windows, "wissen" solche Dinge überhaupt nicht. Die Programme können aber gewisse *Vorschriften* ausführen, in denen z.B. steht, dass die Nummer 85 "Serientitel" bedeutet und dass dieser im Anschluss an die Nummer 77 (die bedeutet "Umfangsangabe") ausgegeben werden muss, auf neuer Zeile und in runden Klammern.

Wir müssen also drei verschiedene Dinge auseinander halten:

1. Die eingegebenen **Daten** (Datensätze in Form von Listen mit nummerierten Angaben)
2. Die **Programme**, die darauf zugreifen und verschiedenste Bilder davon anzeigen: mal mit, mal ohne die Nummern! Auch ein Register ist ein Bild, das Teile von Datensätzen zeigt, nicht die Sätze selbst.
3. **Vorschriften**, die den Programmen sagen, **wie** sie das alles machen sollen.

Die Programme sind also neutral. Mit den Datensätzen allein können die Programme deshalb nichts anfangen, sie brauchen die Vorschriften dazu, und jede Datenbank kann ihre eigenen Vorschriften haben. So ist es möglich, dass mit denselben allegro-Programmen Datenbanken betrieben werden, die ganz verschieden aussehen, bis hin zu Daten mit chinesischen und japanischen Schriftzeichen, obwohl die allegro-Entwickler von diesen Sprachen nichts verstehen. Die Vorschriften sind das wahre Geheimnis des Systems, das wichtigste Konzept von allegro : erst dadurch wird es flexibel.

Für Systemleute: hinter oder unter **allegro** steckt kein relationales Datenbanksystem, sondern es ist ein eigenes Konzept. Jedes Datenbanksystem verfügt über ein System von Vorschriften für den Umgang mit konkreten Daten, nur werden sie nie "Vorschriften" genannt, auch bei *allegro* nicht. Man findet die verschiedensten Namen, z.B. "Syntaxtabellen" oder "Skripte" oder "Schablonen" oder "Parameter". Nur nicht "Vorschriften", obwohl das ein genau passendes Wort wäre. Eine *allegro*-Datenbank braucht mehrere Arten von Vorschriften:

1. Für die **Datenstruktur** : die **Konfiguration**, das ist eine Datei vom Typ CFG; darin steht hauptsächlich, welche Felder ein Datensatz haben kann. Die meisten Anwender benutzen die Konfigurationsdatei A.CFG (oder \$.CFG). In der Konfiguration steht aber nichts über die Register oder über die Bildschirmanzeige, das machen andere Vorschriften:

Die anderen drei Arten von Vorschriften müssen natürlich auf die Konfiguration abgestimmt sein:

2. **Datenausgabe:** solche Vorschriften heißen **Exportparameter**. Dazu gehört auch die Bildschirmanzeige von Datensätzen, wie oben zu sehen. Solche Dateien sind vom Typ .APR, wenn man A.CFG verwendet, vom Typ .BPR, wenn man B.CFG verwendet, usw.
3. **Register:** Vorschriften für den Index heißen **Indexparameter** und stehen in einer Datei vom Typ .API (bzw. .BPI, wenn man B.CFG verwendet, usw.) Darin steht, was für Register die Datenbank haben soll, welche Datenfelder indexiert werden sollen, und alle Einzelheiten, wie das gemacht werden soll.
4. **Fremddaten-Umwandlung:** Für jedes Fremdformat braucht man eine eigene Vorschrift, die man **Importparameter** nennt. Das sind Dateien vom Typ .AIM (bzw. .BIM, wenn man B.CFG verwendet, usw.)

Das ist alles **noch viel zu abstrakt**? Dann müssen wir uns eine Abschweifung leisten. Vielleicht wird es mit einem Denkmodell klarer, das ohne Elektronik auskommt. Das Modell arbeitet mit besonderen Formularen. Eine leeres Formular sieht so aus:

In die Spalte links trägt man Feldbezeichnungen ein, in die ganz schmale Spalte kommt eine Ziffer, wenn ein Feld mehrfach auftritt, um es von einem anderen, gleichnamigen Feld zu unterscheiden. Man beachte: vorgedruckt ist nichts, und zwar damit man die Felder alle ausnutzen kann. Ein Formular mit fest vorgegebenen Feldern hätte in der Praxis immer irgendwelche leeren Felder, weil nicht jedes Datenfeld jedesmal gebraucht wird, viele sogar sehr selten!

Ein ausgefülltes Formular sieht so aus: (Mehrfachbelegung kommt nur bei "Verf." vor)

IdNr		123456
Titel		Die Geschichte der Shakespeare-Rezeption
Verf.		Steiger, Klaus-Peter
Verf.	2	Günther, Alfred
Ort		Stuttgart
Verlag		Kohlhammer
Jahr		1987
Umfang		230 S.
Serie		Sprache und Literatur ; 123
ISBN		3-17-009426-2
Sign.		SH-4321
....	

Für Systemleute: hier liegt ein ganz großer Unterschied zu relationalen Datenbanken; die haben feste, vordruckte Formulare. Abgesehen vom Platzverbrauch hat man dabei immer große Probleme mit Feldlängen, selten belegten Feldern, Mehrfachfeldern u.a., aber auch mit der Indexierung von Feldbestandteilen und Stichwörtern. Für **allegro** alles kein Thema. Ein anderes Formular könnte so aussehen: (hier ist "Schlagw." dreifach belegt)

IdNr.		987654
Titel		Das Zeitalter der Extreme : Weltgeschichte des 20. Jahrhunderts
Original		The age of extremes : The short twentieth century 1914-1991
Verf.		Hobsbawm, Eric
Übers.		Badal, Yvonne
Ort		München
Verlag		Hanser
Jahr		1995
Schlagw.		Zeitgeschichte 20. Jahrhundert
Schlagw.	2	Kalter Krieg
Schlagw.	3	Weltgeschichte
Umfang		783 S.
Sign.		GE-5678

Man sieht: es können unterschiedliche Felder auftreten, es bleibt aber kein Feld leer, höchstens am unteren Ende; das kostet in der Realität keinen Speicherplatz. Die Feldbezeichnungen nennen wir kurz "Kategorien". Der Ordnung halber müssen wir uns eine Liste der erlaubten Kategorien aufstellen; ein sehr wichtiges Dokument. Aber halt! Man merkt sehr schnell: es ist unnötig mühsam, immer die Feldbezeichnungen im Klartext hinzuschreiben, außerdem erschwert es die Benutzung durch Personen, die nicht Deutsch können. Eleganter ist es, schlichte Nummern zu verwenden. Dann könnte das zweite Formular so aussehen, wenn man z.B. 20 für "Titel", 74 für "Ort" festlegt usw.:

00		987654
20		Das Zeitalter der Extreme : Weltgeschichte des 20. Jahrhunderts
22		The age of extremes : The short twentieth century 1914-1991
40		Hobsbawm, Eric
47		Badal, Yvonne
74		München
75		Hanser
76		1995
31		Zeitgeschichte 20. Jahrhundert
31	2	Kalter Krieg
31	3	Weltgeschichte
77		783 S.
90		GE-5678

Man hat nun weniger zu schreiben, aber keinen Verlust an Präzision.

In unsere Liste der erlaubten Kategorien schreiben wir die Bedeutung dieser Nummern. Dann braucht man nur die Liste zu übersetzen, damit anderssprachige Nutzer mit dem Katalog umgehen können.

Die Liste der erlaubten Kategorien enthält auch Angaben, welche Nummern mehrfach auftreten dürfen, wie z.B. 40 oder 31, und andere Angaben. Um einen Begriff zu haben, wollen wir die Liste **Konfiguration** nennen, und sie könnte ungefähr so aussehen: (die vollständige Liste findet man in der Konfigurationsdatei \$A.CFG)

```
#00 "Identifikationsnummer"
#20 "Titel"
#22 "Originaltitel (Einheitstitel)"
#31 "Schlagwort" (M)
#40 "Verfasser" (M23)
#41 "Herausgeber" (M)
#47 "Übersetzer"
#74 "Erscheinungsort"
#75 "Verlag"
#76 "Erscheinungsjahr"
#77 "Umfangsangabe"
#85 "Serie"
#90 "Signatur" (M)
```

M23 bedeutet, es können ein zweites und ein drittes Feld vorkommen und in die Wiederholungsspalte muss dann 2 bzw. 3; ein schichtes M heißt: beliebige Wiederholung möglich (Wiederholungszeichen dann 2,3,...., A,B,C,...).

Was soll das Zeichen # ? Es ist praktisch, die Nummern mit einem besonderen Zeichen zu markieren, damit man es in jedem Zusammenhang sofort erkennt, wenn von diesen Nummern die Rede ist. Meistens wird dazu das Doppelkreuz # benutzt. (Daher kommt es übrigens, dass das *allegro*-Logo ein großes Doppelkreuz ist.)

Warum sollte man diese Dinge so penibel genau vorgeben? Dazu kommen wir jetzt.

Wir stellen uns dazu mal vor, wir hätten zwar keinen Computer, aber dafür billiges, fleißiges, nur nicht besonders intelligentes Personal. Der ideale Bibliotheksknecht kann lesen, abschreiben und alphanumerisch sortieren (Zeichen für Zeichen), braucht aber vom Inhalt der Bücher nichts zu verstehen. Er muss nur Vorschriften ausführen können. Diese Vorschriften müssen einfach deshalb sehr genau sein, damit er nichts falsch machen kann, auch wenn er vom Sinn der Dinge nichts kapiert.

Der Betrieb läuft dann so: wir schreiben selber die Formulare, aber alles andere sollen die Knechte machen. Sie sollen die Formulare schlicht in der Reihenfolge der Nummern in Ordnern (= Dateien) abheften. Zum Benutzen des Katalogs wollen wir nicht immer in den Raum wandern müssen, wo dieser steht. Die Knechte sollen uns jedes Mal diejenigen Blätter herausuchen und bringen, die von Interesse sind. Was wir dafür nur brauchen, sind gute Register. Im Register sollen die Nummern der Blätter stehen, so dass ein Knecht dann diese Seiten holen kann.. Oder noch besser (der Mann ist ja billig): er soll die Blätter lieber drin lassen (denn sonst könnte in der Zwischenzeit jemand anders sie nicht finden, und man müsste sie jedes Mal zurücksortieren) und soll nur diejenigen Einzelheiten abschreiben, die wir dann brauchen.

Wir müssen für unsere Knechte also mindestens zwei Vorschriften ausarbeiten: eine für das Anlegen der Register und eine für das Aufschieben der Einzelheiten, die wir jeweils sehen wollen, wenn wir einen Auftrag geben, etwas aus dem Katalog zu holen. Diese Vorschriften können, weil wir das Nummernschema haben, sehr einfach und trotzdem präzise aussehen, d.h. wir müssen nicht viele Worte machen. Genügen würde eine Liste wie zum Beispiel diese: (Statt "Registervorschrift" können wir auch sagen: Indexparameter. Die meisten Anwender nutzen die Index-Parameterdatei CAT.API)

```
|1="Personennamen"          Das bedeutet: Register 1 hat den Titel "Personennamen"
|4="Titel"
|3="Schlagwörter"
|6="Verlage"
#20 " |4"                   das heißt: sortiere den Inhalt von #20 ins Register 2
#22 " |4"                   und den von #22 ebenfalls
#31. " |3"                  #31. heißt: wenn mehrere #31 vorkommen, alle gleich behandeln
#40. " |1"                  genauso geht's mit der #40 und Register 1
#47 " |1"
#41. " |1"
#75 " |6"
```

Und nun zur Vorschrift für die Titelaufschreibung, oder vornehmer: Anzeigeparameter:

```
#40 P": "
#20
#41 p" / hrsg. von: "
#47 p" (übers. von: " P") "
#74 C
#75 p" : "
#76 p", "
```

#77 p". - "
 #31 C p"Schlagwort: "
 #90 C p"Signatur: "

Der Knecht weiß dann: die hier angegebenen Kategorien sollen in dieser Reihenfolge vom Formular abgeschrieben werden, also #40, #20, #22, Dabei soll er noch ein paar verschlüsselte Regeln anwenden: C bedeutet "neue Zeile", p" : " heißt, dass vor das Element die Interpunktionszeichen " : " zu setzen sind, und P") " bedeutet., es soll eine Klammer dahinter. (Das kleine p soll heißen "davor", das große P heißt "dahinter"!)

Dann kommt für das zweite Beispiel folgendes heraus:
 Hobsbawm, Eric: Das Zeitalter der Extreme : Weltgeschichte des 20. Jahrhunderts (übers. von Badal, Yvonne)
 München : Hanser, 1995. - 783 S.
 Schlagwort: Weltgeschichte 20. Jahrhundert
 Signatur: GE-764

Solche Vorschriften könnte sogar ein Bediensteter ausführen, der überhaupt nicht richtig lesen, sondern der nur Buchstaben und Ziffern vergleichen und kopieren kann. (Wie z.B. ein Computer.)

Solche Vorschriften, das ist ein weiterer ganz großer Vorteil, können leicht geändert werden, ohne dass man die Daten oder die Formulare ändern muss.

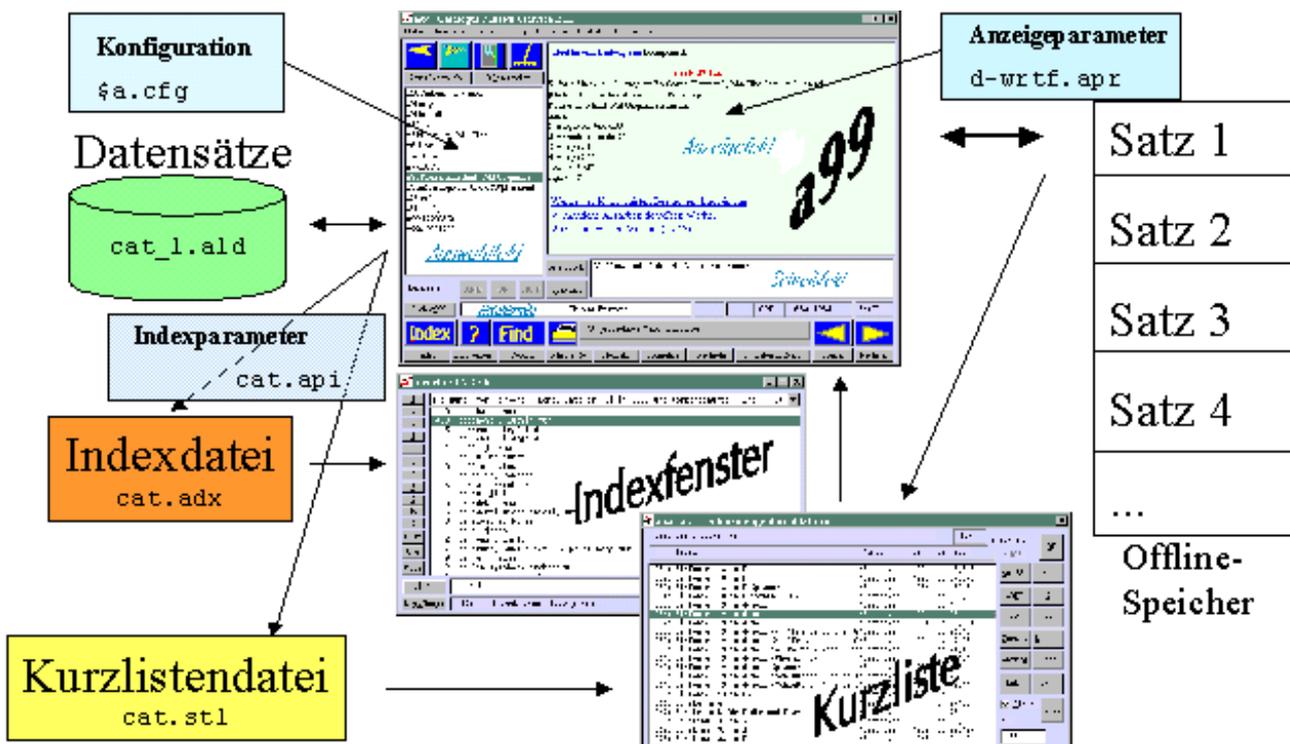
Warum hat nie eine Bibliothek in dieser Weise gearbeitet? Weil menschliche Arbeitskraft dafür eben doch immer zu teuer war. Mit Computern aber kann man genau so (und kaum anders) arbeiten, denn ihre Leistung ist konkurrenzlos billig. Und Zeichen vergleichen, kopieren und sortieren können sie astronomisch schnell - mit Verständnis lesen dagegen überhaupt nicht. Deshalb sind die hübsch einfach aussehenden Vorschriften in Wirklichkeit noch nicht genügend genau. Etwa die Vorschrift, die Titel ins Register 4 zu sortieren. Dabei würde der Titel "Das Zeitalter der Extreme" unter D landen, ziemlich weit weg von Z, wo man ihn gewohnheitsmäßig suchen würde. Eine Vorschrift "Artikel übergehen" kann man für Knechte ohne höheres Sprachverständnis aber nicht formulieren, zumal manchmal ein wie ein Artikel aussehendes Wort gar kein Artikel ist. Hier hilft nur, die Daten von vornherein genauer aufzuschreiben, und zwar den Artikel eindeutig zu kennzeichnen. Wenn man das so macht: -Das- Zeitalter der Extreme, dann kann weder ein Knecht noch ein Computer etwas falsch machen, wenn man ihm sagt: was zwischen -...- eingeschlossen ist, wird nicht mitsortiert! In der Kurzschrift der Parameter sieht das dann so aus (das 'u' ist die Anweisung, die Teile zwischen -...- wegzulassen):

#20 e" : " u p"|4".

Hier ist gleich noch eine Vorschrift eingebaut: die Angabe e" : " bedeutet: Ende der Registerzeile bei "Spatium Doppelpunkt Spatium". Der Titelzusatz soll also nicht mit ins Register, nur der Haupttitel.

Diese wenigen Überlegungen lassen erahnen, dass man die Vorschriften für die Register, aber auch für die Anzeige, noch stark erweitern muss, wenn alle Eventualitäten berücksichtigt werden sollen. Nicht nur bei allegro, sondern bei jedem System, das mit Bibliotheksdaten hantiert, findet man irgendwo solche Vorschriften; manchmal sind sie sehr verborgen und unzugänglich. Die Beispiele oben sind formuliert in der sog. Exportsprache des allegro-Systems; diese Sprache gilt natürlich nur für allegro.

Fassen wir zusammen:



Dieses Bild ist ein grober Überblick, der nur die wichtigsten Zusammenhänge zeigen soll. Man sieht, welche Dateien zusammen die Datenbank bilden, welches die wichtigen Parameterdateien sind, und wo diese sich auswirken.

Indexvorschrift

a99 : FLEX-Verbesserungen

Die in der vorigen Nummer ausführlich beschriebene FLEX-Sprache ist noch ein wenig erweitert worden. Außerdem kann man jetzt einen FLEX noch leichter mit der Hand eingeben: In das Schreibfeld schreibt man z.B. 2 FLEX-Text [Enter]

Dann entsteht #uX2x *FLEX-Text* und kann mit Alt+2 ausgelöst werden.

Übrigens kann auch mit *alcarta* Dateneingabe über Formulare und den FLEX-Befehl **ask** stattfinden. Dazu muß in der INI-Datei aber mindestens die Berechtigung **access=2** stehen, sonst funktioniert das Speichern nicht.

Es gibt die folgenden neuen Befehle:

:marke

Eine Sprungmarke, zu der mit dem Befehl `jump marke` gesprungen werden kann.

if #nnn command

Wenn die Kategorie #nnn besetzt ist, wird der Befehl *command* ausgeführt. Das kann jeder andere FLEX-Befehl sein, der in dem Moment sinnvoll ist. (Dieses **if** kann noch nicht soviel wie das von *avanti* !)

jump marke

Es wird weitergemacht bei dem nächsten Befehl, der auf die Sprungmarke **:marke** folgt. Diese kann an jeder Stelle des FLEX-Programms stehen.

set pM (keine DOS-Entsprechung)

Hiermit wird für den Befehl **update** die Sprungmarke **#-M** festgesetzt, an welcher dann der Primärschlüssel gebildet werden soll. Die Zeile **#-M** muss in den Indexparametern vorkommen. *Vorsicht*: es kann in dem Abschnitt nicht mit #u1 gearbeitet werden, denn #u1 gibt es nur, wenn über einen ak-Befehl gesprungen wird.

Fehlt *M*, wird der Primärschlüssel so gebildet wie vom Programm UPDATE: der erste Schlüssel, der in der Reihenfolge der ak-Befehle entsteht, ist der Primärschlüssel. Dies ist der Standard, wenn der Befehl **set p** fehlt. Wenn mehrere, durch Code 8 getrennte Schlüssel entstehen, gilt nur der erste.

set uxyz (entspricht DOS-Option **-fmxy**)

Update-Modus setzen. Die Werte x und y wirken wie beim DOS-Programm UPDATE, der Wert z ist neu: wenn hier eine 0 steht, wird nicht echt eingemischt, d.h. die Sätze werden nicht gespeichert, sondern in den Offline-Speicher kopiert. Am Ende der Sitzung kann man immer noch alles speichern lassen.

update Dateiname (entspricht DOS-Option **-updateiname**)

Die Datei *dateiname* wird geöffnet. Es muss sich um eine Datei vom Typ .cLG oder .cDT handeln. (Bei Update ist nur .cLG möglich). Die Sätze der Datei werden dann gemäß **set pM** und **set uxyz** eingemischt.

Mit Taste 'x' kann man unterbrechen, danach durch erneuten "update"-Befehl an derselben Stelle fortsetzen!

Anders als bei UPDATE: Ist der Primärschlüssel nicht eindeutig, wird der Satz als Neusatz eingemischt. Außerdem wird die Kategorie des Änderungsdatums beseitigt und dann automatisch neu belegt.

upload Dateiname

Gleichwertig mit update, wenn vorher **set u01** gesetzt wurde. Das bedeutet, alle Datensätze in der Datei werden als neue Sätze eingemischt, es findet keine Identifizierung und Vermischung mit vorhandenen Daten statt.

TIP: Zum Testen der Primärschlüssel-Optionen kann man dieses FLEX benutzen. Man gibt ins Schreibfeld:

```
2 set pX\var "Primaerschl.: " p\ins #ups
```

und dann mit Alt+2 sieht man das Resultat (denn der Befehl `ins #ups` schreibt es zusätzlich ins Schreibfeld).

TIP: Mit Alt+r sieht man im Reservespeicher auch die #u-Variablen, die (anders als beim #a-Befehl von PRESTO) auch mit der Hand verändert werden können, wie jede normale Kategorie.

Zum Befehl "deposit"

Dieser geht jetzt endlich. Es war noch eine Macke drin. Im Gegensatz zu "display" führt "deposit" lediglich die Anzeigeparameter aus, gibt aber nichts aus. Der Ausgabertext landet aber in der Internen Variablen.

Mit dem Befehl `deposit X\ins #123` kann man den Abschnitt #-X in den Anzeigeparametern ausführen und den Ausgabertext dann in #123 speichern lassen. Auf diesem Wege kann man einfacher als per Globaler Manipulation Änderungen in Kategorien erreichen.

Dazu als Information: Wenn in Anzeige- oder Exportparametern Manipulationsbefehle vorkommen, bis hin zum Befehl M, wirken sich diese nur auf die Ausgabedaten aus, nicht auf den eigentlichen Datensatz - der soll sich durch eine Ausgabe gleich welcher Art auf keinen Fall verändern.

Mehrfach-Restriktion

Man kann mehrere Restriktionen nacheinander auf eine Ergebnismenge anwenden. Dazu braucht man nur in der "Suchbefehl"-Zeile die entspr. Befehle mit "and" zu verketteten:

```
per shakesp? and erj>1990 and typ=m
```

Neue INI-Befehle

Alle Befehle sind kommentiert in der Datei **A99.INI**. *Empfehlung*: diese Datei immer als Grundlage nehmen für neue INI-Dateien, damit man nichts vergißt. Die Datei ist unmittelbar funktionsfähig, man muss nur die eigenen Werte einsetzen.

SaveResults=0/1/2

Die Frage am Ende, "Sitzung später fortsetzen", kommt nur bei 0. Bei 1 gilt automatisch "Ja", bei 2 "Nein".

SaveAsk=0/1

Bei 0 unterbleibt die Frage "Speichern? Ja/Nein" nach Druck auf [Speichern] (1 für "Ja" ist Default)

Rdo=0/1

Bei 0 ist die Schreibfunktion im Anzeigefenster abgeschaltet. Default: für **a99**=1, für **alcarta**=0

SubHelp=0/1

Bei 0 ist das Teilfeld-Hilfsfenster abgeschaltet. Dieser Wert ist über das Menü "Optionen" änderbar.

print=0/1

Bei 0 ist die Druckfunktion abgeschaltet (Nützlich bei **alcarta**, wenn kein Drucker angeschlossen)

exflex=0/1

Bei 0 nimmt **a99** keine externen FLEXe an (siehe *news* 54)

closx=0/1

Bei 1 wird das Indexfenster geschlossen, wenn man auf das Hauptfenster klickt, bei 0 nicht

Anzeige von Dateien per FLEX

Jede beliebige ASCII-Datei kann ins Anzeigefeld geholt werden, nicht nur Help-Dateien! Man setzt hinter das Befehlswort "help" den Dateinamen, das ist alles.

Beispiel: in das Schreibfeld eingeben 1 help uifeger

Dann Alt+1, und es kommt die Datei UIFEGGER in die Anzeige. Also kann man auch folgendes tun: zuerst per "download" eine Liste exportieren, dann diese mit dem help-Befehl anzeigen lassen. Die Datei muss ASCII sein, damit das korrekt funktioniert. Mit dem Anzeigefeld kann man aber noch mehr machen:

Anzeigefeld speichern als RTF-Datei

Im Menü "Datei" gibt es jetzt: "Anzeige speichern als ..." (UIF-Zeile 258). Damit kann man nun ohne Umweg über WinWord den Inhalt des Anzeigefeldes unmittelbar als RTF-Datei abspeichern, egal was gerade drin steht. Man muss allerdings die Berechtigung access=4 haben! (Es soll nicht jeder die Hilfedateien bearbeiten können.)

Für Hilfedateien gibt es einen zusätzlichen Trick: Man macht sich zuerst ein FLEX, z.B. 2 help xyz.rtf, also mit der Typangabe ".rtf", die man sonst weglässt. Dann wird die Datei so eingelesen, dass auch die FLIP-Zeilen zu sehen sind und manipuliert werden können! Man hat damit die vollständige Kontrolle über die Hilfedateien und braucht dafür keinen anderen Editor mehr - denn im Anzeigefeld kann man ja schreiben.

Ein Flip-Begriff im Text muss zwischen zwei Codes 160 eingeschlossen werden. Diesen erzeugt man mit Alt+0160 (nicht Alt+160)! Man kopiert am besten einen anderen Flip und ändert den Text, denn die Farbe und die Unterstreichung können sonst nicht erzeugt werden; ein vollständiger RTF-Editor ist das Anzeigefeld leider nicht. Aber Cut-and Paste mit Strg+c/x/v geht immerhin. Es gibt noch eine Möglichkeit:

Mit dem kleinen Hilfsprogramm `xrtf.exe` kann man RTF-Dateien unabhängig von a99 bearbeiten.

UNFREE.EXE

Mit dem Programmchen FREE.EXE kann man eine Datenbank entsperren, d.h. Schreibzugriffe wieder ermöglichen. Dabei wird lediglich das erste Byte der TBL-Datei auf den Wert 0 gesetzt. Nun gibt es UNFREE.EXE. Es macht das Umgekehrte: Alle Schreibzugriffe werden blockiert. Aber nur, wenn die .TBL nicht bereits gesperrt ist, in welchem Fall der ERRORLEVEL auf 10 gesetzt wird. (Es werden 100 Versuche mit je 1 Sekunde Abstand gemacht)

Damit kann man eine Datenbank endlich auch aus einem Batch heraus sperren und wieder freigeben, statt nur vom **CockPit**.

Beispiel:

```
unfree c:\allegro\katalog\cat.tbl
if errorlevel 9 goto fehler
copy .... (oder andere beliebige Befehle, dann wieder freigeben:)
free c:\allegro\katalog\cat.tbl
goto ende
:fehler
echo Sperren ging nicht, Datenbank ist schon gesperrt!
```

: ende

Signalfile

In großen Netzen will man gelegentlich verhindern, dass jemand in eine Datenbank überhaupt einsteigt. Dazu reicht es nicht, sie mit dem Programm UNFREE zu sperren, weil dann immer noch lesender Zugriff möglich ist.

Jetzt kann man ein "Signalfile" anlegen: eine kleine Datei mit dem Namen *dbn.SGF* auf dem Datenverzeichnis. Es kommt auf das erste Zeichen an: ist es '0', kann normal gearbeitet werden, ist es '1', wird jeder Einstieg abgewiesen. Hinter der '1' kann man eine einzeilige Meldung unterbringen, die dann ausgegeben wird, um den hoffnungsvollen Klienten zu verträsten, z.B.

```
1 Zur Zeit wegen Wartung kein Zugriff moeglich. Ab 14:00 Uhr wieder Betrieb.
```

Ändert man die '1' wieder in '0', kann sofort wieder zugegriffen werden. Fehlt die Datei *dbn.SGF*, läuft alles wie bisher.

Aber: diejenigen Nutzer, die schon "drin" sind, werden nicht betroffen! Man muss sie von Hand hinauskomplimentieren, wenn das nötig ist (z.B. wenn sonst das Kopieren von Dateien nicht funktioniert).

Eingabeverhinderung per PV

Bisher konnte man über PV-Routinen zwar Fehlermeldungen ausgeben lassen, aber nicht die Eingabe einer Kategorie überhaupt verhindern. Das geht zwar über die Eigenschaftszahl 5 in der CFG, das ist jedoch nicht immer ein gangbarer Weg, besonders wenn lediglich ein bestimmtes von mehreren Mehrfachfeldern blockiert werden soll.

Jetzt kann man folgendes tun: Man erzeugt in der PV-Routine eine Ausgabe, die mit dem Minuszeichen (= Bindestrich) beginnt oder nur aus diesem besteht, dann wird das betreffende Datenfeld ohne Kommentar nicht in den Satz eingeordnet.

INDEX

Bei Fehlern in den Parametern kam zwar eine Meldung (meistens das Fehlen einer nachzuladenden Tabelle), das Programm lief aber trotzdem an und war dann nur mit Gewalt zu stoppen, falls nicht die Option `-m1` gegeben war.

Jetzt kommt die Fehlermeldung, aber das Programm beendet sich dann, so dass man zuerst den Fehler korrigieren kann, bevor man neu startet.

PRESTO + a99

Wichtige PV-Möglichkeiten

Beide Programme führen an strategischen Punkten einen Aufruf des Hilfsabschnittes in den Indexparametern durch (siehe Handbuch Kap.10.2.8, S. 221).

1. Beginn einer Neuaufnahme durch Kopie eines Datensatzes. In #u2 steht dann "C".
2. Abspeichern eines neuen Datensatzes. In #u2 steht dann "N". Dieser Aufruf findet VOR der Datums-Stempelung und der Nummernvergabe statt. Bei **a99** kommt hinzu: wenn man in der Prüfroutine eine Ausgabe produziert, wird sie als Nummer verstanden. Liegt diese zwischen 1 und 255, wird der Satz in die betr. Datei gespeichert statt in die Default-Datei (Befehl `InputFileNr=...` in der INI-Datei).
3. Letzte Prüfung vor dem Speichern: NACH der Zeitstempelung und Nummernvergabe wird nochmals der Hilfsabschnitt aufgerufen, d.h. man hat noch die Chance, am Satz Prüfungen und Änderungen vorzunehmen und dabei sowohl das Änderungsdatum wie auch die IdNummer mit einzubeziehen. In #u2 steht dann "s" (Kleinbuchstabe). Wird hierbei eine Ausgabe produziert, führt das zum Abbruch der Speicherung, mit dieser Ausgabe als Fehlermeldung.

Nur PRESTO: Teilfelder in Abfrageliste

Aufgabe: es soll in einem Sonderabschnitt eine Abfrage konfiguriert werden, die an eine schon vorhandene Kategorie noch ein oder mehrere Teilfelder anhängt. Sagen wir, an die #90 soll ein ²a mit dem Barcode und ein ²d mit dem Zugangsdatum angehängt werden. Das lässt sich in der Abfrageliste so lösen:

```
-D
90 "Sign. ".           #90 wird hierdurch "angesprochen" (erscheint aber nicht)
90 2a"Barcode: "      Eingaben werden als Teilfelder werden hinten angehängt
90 2d"ZgDatum: "
-x
```

Dazu sollte dann noch der Kennbuchstabe D irgendwo in einer Typzeile stehen, damit der Operator erfährt, dass D einzugeben ist, um an diese Stelle zu kommen.

Für a99 müsste man in die Formulardatei schreiben: (siehe Erklärungen in FORM.TXT)

```
[Signatur-Ergänzung]
$a-#90 "Barcode"
$d-#90 "ZgDatum"
```

Dann werden sogar die Teilfelder, falls sie schon vorkommen, ersetzt statt nur angehängt.

Export

Schaltprobleme

Der Schaltbefehl #< soll im Arbeitsspeicher zum vorigen Satz zurückschalten, den zuletzt nachgeladenen Satz jedoch im Arbeitsspeicher belassen, damit man hinterher evtl. mit #<^ wieder "hochschalten" kann. Wenn jedoch dann eine neue Nachladung stattfindet, soll der betr. Satz hinter den aktuellen Satz gehängt werden, d.h. der Stapel soll nicht um einen Satz verlängert, sondern der Teil hinter dem aktuellen Satz soll ersetzt werden durch den neu nachgeladenen. Das funktionierte bei *a99/alcarta/avanti* noch nicht wie bei PRESTO/SRCH, wurde inzwischen aber korrigiert.

Sehr seltener Fehler

Bei Categoriesystemen der Schwergewichtsklasse (mehr als 240 Deskriptorzeilen in der CFG), konnte ein merkwürdiges Versagen beim Export auftreten: eine bestimmte Sprungmarke wurde manchmal nicht gefunden. Setzte man eine andere ein, z.B. #-b statt #-a, klappte es. Betroffen waren alle Programme, die exportieren können, auch *a99* etc.. Am 6.8.99 wurde der Fehler behoben. Anwender des konsolidierten Formats \$.CFG waren nicht betroffen.

#uxn gibt es nicht

In *news* 53 auf S. 17 stand, dass man in der Variablen #uxn die interne Nummer des nachgeladenen Satzes hätte, wenn gerade eine Nachladung stattgefunden hat. Das stimmt nicht. Richtig ist, dass es dafür die Sonderkategorie #nra gibt. Unter normalen Umständen, d.h. wenn keine Nachladungen stattfinden, steht darin immer dasselbe wie in #nr, die aktuelle interne Satznummer.

Indikatorprüfungen (Handbuch Kap. 10.2.6.3, S.199)

Die Manipulationsbefehle i, I, v, V prüfen jeweils den gesamten Kategorietext, nicht den Arbeitstext. Anders gesagt, sie nehmen die Sonderkategorie #cc, die immer den vollständigen Text der gerade verarbeiteten Kategorie enthält. Wenn man also schreibt #90 +A b" = " i4,C e0, dann funktioniert das nicht, wenn damit das 4. Zeichen des Arbeitstextes geprüft werden soll, also des Teiles, der hinter "=" steht. Geprüft wird das 4. Zeichen in #90 . . . Man müsste also eine #-Variable zwischenschalten und diese dann prüfen, etwa so:

```
#90 b" = " =sg  
#usg +A i4,C e0
```

Manipulationsbefehl y erweitert [am 5.9.99]

Bisher:

- y0 Keine Umcodierung des Arbeitstextes
 - y1 Umcodierung des aktuellen AT mit p-Befehlen
 - y2 Umcodierung mit q-Befehlen
 - yk (2<k<255) Wenn AT mit einem Code >k beginnt, wird nicht umcodiert (das wird von Sinologen gebraucht)
- Die Befehle wirken alle nur, wenn noch kein längerenrelevanter Befehl vorangegangen ist, wie z.B. e20.

NEU:

y" k" mit einer Zahl 0<k<10

k setzt sich zusammen als Summe aus folgenden Werten mit den Bedeutungen:

- 1 Ziffern nehmen
- 2 Kleinbuchstaben nehmen
- 4 Grossbuchstaben nehmen
- 8 Sonderzeichen nehmen (incl. Leerzeichen)

Sollen also alle Buchstaben und Ziffern genommen werden und sonst nichts, muss man

y"7"

sagen, sollen es nur Buchstaben sein, dann

y"6"

Wenn außerdem alle Groß- in Kleinbuchstaben verwandelt werden sollen, setzt man z.B. noch y2 voran, wenn man den Ersetzungsbefehl q A/Z a gesetzt hat.

Wenn man aber Buchstaben, Ziffern *und Leerzeichen* will, wird es noch etwas komplizierter: man muss dann mit lokaler Ersetzung die Leerzeichen zuerst z.B. in "xxxx" umsetzen, diese dann anschliessend wieder in Leerzeichen.

E-Mail-Diskussionsliste

Man schließt sich den ca. 250 Lesern der *allegro*-Liste an, indem man an die Adresse **maiser@buch.biblio.etc.tu-bs.de** eine Botschaft mit nur einer Zeile sendet: **subscribe allegro** . Man wird dann sofort eingetragen und erhält eine Mitteilung mit weiteren Informationen, insbesondere, wie man sich wieder abmeldet. Es gibt auch ein Archiv der Liste.
Frühere Ausgaben, aktuelle Programme etc.: **FTP:** 134.169.20.1 oder **WWW:** www.allegro-c.de/allegro