

 Groupware
Competence Center
GCC



Workgroup Computing Praktikum

Working with Views

University of Paderborn
Business Computing 2 – Information Management & Office Systems
Faculty of Business Administration, Business Computing & Economics
Prof. Dr. Ludwig Nastansky
Warburger Str. 100, D-33098 Paderborn
Tel.: +49--5251--60-3368
<http://gcc.upb.de>

 University of Paderborn
Dept. Business Information Systems
Prof. Dr. Ludwig Nastansky

1

 Groupware
Competence Center
GCC

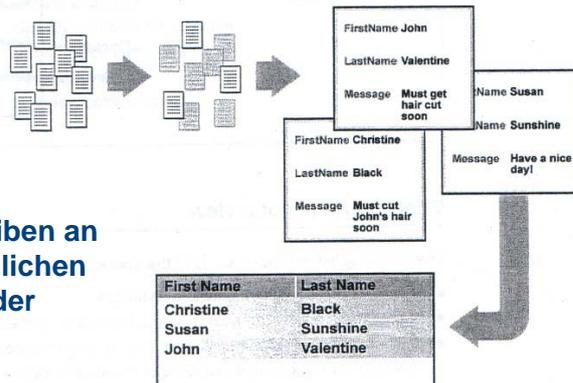
Merkmale von Views (Ansichten)

- **Liste von Dokumenten in einer Datenbank, die individuell organisiert ist und beim Öffnen aktualisiert wird**
 - So genannte „View Selection Formula“ legt die in einer View anzuzeigenden Dokumente aus der bestehenden Datenbank fest
- **Eine View ist aus Zeilen und Spalten aufgebaut**
 - Jede Zeile (row) repräsentiert ein Dokument
 - Jede Spalte (column) zeigt Informationen zu den Dokumenten an
 - Dokumente können mittels eines Doppelklicks in die entsprechende Zeile geöffnet werden
- **Jede Datenbank enthält eine Default-View, die standardmäßig alle Dokumente der Datenbank nach Erstellungszeitpunkt sortiert und durchnummeriert anzeigt**

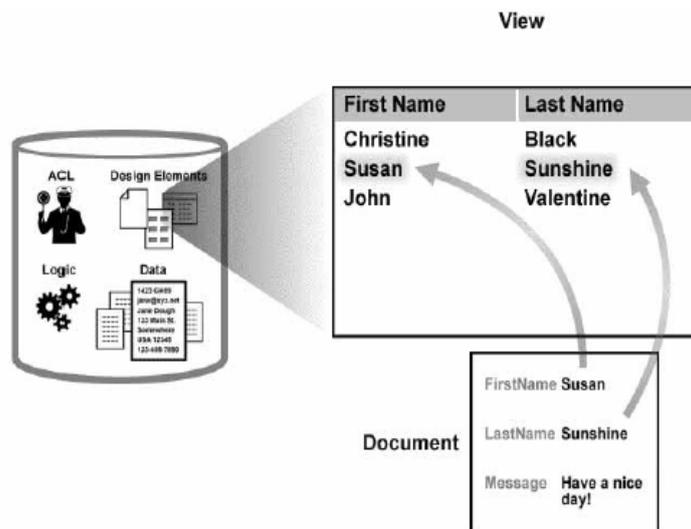
 University of Paderborn
Dept. Business Information Systems
Prof. Dr. Ludwig Nastansky

2

→ Pull-Prinzip: Die mittels der View Selection Formula festgelegten Dokumente werden von Lotus Notes im View-Index referenziert und in der View dargestellt



→ Dokumente bleiben an ihrem ursprünglichen Speicherort in der Datenbank



- ➔ **Zur Erinnerung: Dokumente werden mittels einer Form (Eingabemaske) erstellt**
 - ➔ Die Felder einer Form werden in einer data note (Dokument) in einzelnen items gespeichert
- ➔ **Wegen der Trennung von Daten und Design ist es möglich, ein Dokument mit verschiedenen Forms zu öffnen und zu bearbeiten**
 - ➔ Wenn ein item in einer data note und ein Feld auf einer Form den gleichen Namen haben, so werden die Inhalte dieses items in dem Feld dargestellt und können editiert werden.
 - ➔ Wenn ein item in einer data note kein korrespondierendes Feld auf einer Form besitzt, werden die Inhalte nicht über die Form dargestellt. (Sie können aber trotzdem über die Document Properties Box eingesehen werden.)
 - ➔ Wenn eine Form ein Feld besitzt, zu welchem es kein korrespondierendes item gibt, so werden die Inhalte dieses Feldes beim Speichern der data note als ein neues item hinzugefügt

- ➔ **Lotus Notes legt nach gewissen Regeln in einer bestimmten Reihenfolge fest, welche Form zum Betrachten eines Dokuments genutzt wird**
 1. Form, welche in dem Dokument gespeichert ist
 2. Form, welche in der „Form Formula“ der View angegeben ist
 3. Form, welche im item mit dem Namen „Form“ spezifiziert ist (normalerweise: Form, mit welcher das Dokument erstellt wurde)
 4. Default Form der Datenbank
- ➔ **Trennung von Daten und Design offeriert dem Nutzer wie dem Programmierer mächtige Hilfsmittel zur Gestaltung der Arbeitsabläufe**

- ➔ **View entwerfen**
 - ➔ Welche Dokumente sollen in der View angezeigt werden (view selection formula)?
 - ➔ Welche Informationen sollen in den Spalten dargestellt werden?
 - ➔ Soll die View die Dokumente sortiert oder kategorisiert anzeigen?
 - ➔ Soll es eine private oder gemeinsam genutzte View sein (private view / shared view)?
- ➔ **Eine neue View erstellen, diese benennen und mit sinnvollen Aliassen versehen**
- ➔ **View Type festlegen (shared view / private view)**
- ➔ **View Properties anpassen**
- ➔ **Selektionsbedingungen (view selection formula) festlegen**

- ➔ **Lotus Notes unterscheidet zwischen von allen Nutzern einer Datenbank gemeinsam genutzten und von einzelnen Nutzern „privat“ genutzten Views**
 - ➔ Der Typ einer View muss beim Erstellen festgelegt werden und kann nachträglich nicht mehr geändert werden
- ➔ **Shared Views können von allen Nutzern genutzt werden**
 - ➔ Nutzer können bei entsprechenden Zugriffsrechten in der ACL der Datenbank shared views erstellen, welche allen Nutzern zugänglich sind
- ➔ **Private Views können für eine individuelle Konfiguration der Arbeitsumgebung genutzt werden**
 - ➔ Jeder Nutzer kann sich eigene private Views erstellen, jedoch werden diese nur bei entsprechenden Nutzerrechten in der ACL auf den Server übertragen

Eigenschaft	Beschreibung
View name	Ein kurzer und aussagekräftiger Name, aus dem hervorgeht, welche Dokumente in der View zusammengefasst sind
View type	<u>shared (gemeinsam)</u> : alle Benutzer dürfen auf die View zugreifen <u>private</u> : persönliche Ansicht für einen einzelnen Benutzer
Location for the view	<ul style="list-style-type: none"> ▪ Nur zwei Ebenen in der Hierarchie ▪ Gruppieren von Ansichten, die gleiche Dokumente enthalten

Eigenschaft	Beschreibung
Copy style	Das Design einer anderen View als Schablone übernehmen
Selection conditions	<ul style="list-style-type: none"> ▪ Alle Dokumente in der DB oder nur bestimmte Dokumente über Auswahlbedingung (Selection condition)

➔ **Spalte (column) einfügen / anhängen**

➔ **Wert für eine Spalte festlegen**

- ➔ Feld
- ➔ Einfache Funktion (Simple Function)
- ➔ Formeln (mittels Formula Language)

➔ **Spalte / Spaltenüberschrift formatieren**

- ➔ Spaltenbreite
- ➔ Schrifttyp, Schriftgrad, Stil

➔ **Werte in der Spalte formatieren**

- ➔ Schrifttyp, Schriftgrad, Stil

Um folgendes anzuzeigen...	Wähle...	Und...
Einen einzelnen Wert, der in einem Feld gespeichert ist	Field	Wähle ein Feld aus der erscheinenden Auswahlliste aus
Informationen über das Dokument, die nicht in einem Feld gespeichert sind	Simple Function	Wähle eine simple function aus der erscheinenden Auswahlliste aus
Mehrere Feldwerte in einer Spalte oder neue Werte	Formula	Gebe die entsprechende Formel an

- seit Domino 7
- Bietet die Möglichkeit eine Spalte identisch in mehreren Views zu verwenden (z.B. Kundennr. in verschiedenen Ansichten)
- Hat das gleiche Aussehen und die gleichen Eigenschaften, wie eine „normale“ Spalte in der view
- Kann Alias und Kommentar haben
- Erstellen einer Shared Column über Shared Code → Columns → new Shared Column
- Einfügen einer Shared Column in eine View: Create → Insert / Append New Shared Column

- **Normale Sortierung (Standard)**
 - Aufsteigend / Absteigend
 - Case-sensitive
 - Accent-sensitive
- **Kategorisierte Sortierung (Categorized)**
 - Zusammengehörige Dokumente werden in Abhängigkeit von ihrem Spaltenwert in einer gemeinsamen Kategorie gruppiert

Last Name	First Name
▼ Accounting	
Desai	Lewis
Moon	Gloria
▼ Development	
Nakahara	Brandon
▼ Education	
Ortiz	Victoria
Ortiz	John

- **Kategorisiert wird stets die äußerste linke Spalte**
- **Die kategorisierte Spalte bekommt keinen Titel**
- **Die Spalte sollte die Breite 1 haben**
- **Die Option „Resizable“ der Spalte sollte deaktiviert werden**
- **Dreieckssymbol soll angezeigt werden, wenn die Zeile erweitert werden kann (show twistie).**
- **Verwenden von fetter und farbiger Schrift, um die Werte der kategorisierten Spalte gegenüber den weiteren Informationen hervorzuheben.**
- **Sortierung und Kategorisierung muss ausgewählt werden**

- **Das Indizieren der Dokumente einer Datenbank für die Anzeige in Views beansprucht relativ viel Prozessorkapazität**
 - Viele Views in einer Datenbank können gravierende Einbußen bei der Performance zur Folge haben
 - **Beachte:** jede Sortierung verhält sich in Bezug auf die Performance wie ein weiterer View!
- **Ein gutes Datenbankdesign schließt automatisch ein gut durchdachtes View-Konzept mit ein, um unnötige und redundante Views zu vermeiden**
- **Zu wenige Views hingegen können den Benutzer bei der täglichen Arbeit behindern und den Nutzen der Datenbank einschränken**
 - Die Applikation soll den Bedürfnissen des Nutzers angepasst sein, nicht der Benutzer an die Fähigkeiten der Applikation
- **Daher: Erstelle so viele Views wie nötig, aber so wenige wie möglich**

- ➔ **Daten, welche in einer View angezeigt werden sollen, sollten nach Möglichkeit in entsprechenden items des Dokuments abgespeichert sein**

- ➔ **Das Berechnen von Werten mittels einer Formel aus einzelnen items eines Dokuments in einer View kann sich sehr negativ auf die Performance der Datenbank auswirken**
 - ➔ Berechnungen werden bei jedem Neuaufbau des View-Index erneut durchgeführt
 - ➔ Ein Trade-Off zwischen Performance und Speicherplatz sollte in diesen Fällen immer zugunsten der Performance ausfallen

- ➔ **View erstellen**
 - ➔ Im Design Pane „Views“ anklicken
 - ➔ Im Work Pane „New View“ anklicken
 - ➔ Im Dialogfenster „Name“, „Typ“, „Auswahlbedingung“, usw. angeben
- ➔ **Auswahlbedingung** *(falls nicht zuvor im Dialogfenster eingestellt)*
 - ➔ Im Object Pane „View Selection“ wählen („Simple Search“ aktivieren!)
 - ➔ Mit dem „Add Condition“ - Dialog Dokumente selektieren
- ➔ **Spalten zur View hinzufügen**
 - ➔ Mit dem Menüpunkt „Create“ → „Insert Column“ / „Append Column“ Spalten erstellen
 - ➔ Im Object Pane „Column Value“ (Feld, dessen Inhalt in der Spalte stehen soll) auswählen
 - ➔ Die Spalten in den Column-Properties formatieren

➔ Sortierung / Kategorisierung zur View hinzufügen

- ➔ Column-Properties, im Tab Sorting: „Sort“ / „Categorized“ einstellen;
für die Kategorisierung:
 - ➔ „Categorized“ und „show twisties when row is expandable“ aktivieren,
ggfs. Twistie Bild auswählen (Richtlinien beachten!)
 - ➔ Nur in Sonderfällen „Click on column header to sort“ aktivieren
 - ➔ nur soweit absolut notwendig aktivieren, da jede Sortierung die
Performance eines Views beanträchtigen kann. Dies fällt spätestens mit
rapide steigender Anzahl Dokumente später negativ auf!

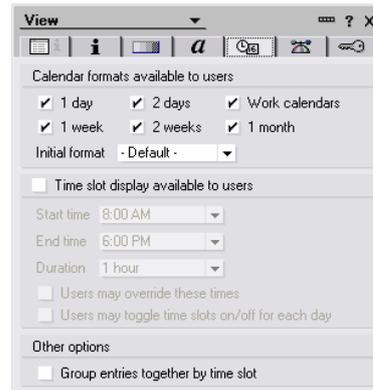
➔ Abschließende Formatierung der View (View-Properties)

- ➔ Allgemeine Darstellung im Tab „Style“ einstellen

- ➔ **Erstelle eine View mit dem Namen „Orders“**
- ➔ **Zeige darin Dokumente an, die mit der Form „Orders“ erstellt wurden**
- ➔ **Kategorisiere die Dokumente nach dem Order – „Type“**
- ➔ **Stelle die Werte der Felder „owner“, „vtype“, „vehicle“, „costs“, „date“ jeweils in einer eigenen Spalte dar.**

→ Calendar Views

- Zeigt Dokumente in Form eines Kalenders an
 - 1 Tag, 2 Tage, 1 Woche, 2 Wochen, Arbeitskalender, oder 1 Monat
- Das Datum wird in der ersten Spalte definiert und sollte versteckt werden
- Aufsteigende Sortierung für das Datum verwenden
- Definition der Uhrzeit analog in der zweiten Spalte



- Warum spricht man im Zusammenhang mit dem Aufbau des View Index in Lotus Notes auch von einem Pull-Prinzip ?