ADMIN & DEVELOPER 2005

Session 14 - Domino 7 und DB2: Effektive Nutzung von Access und Query Views

Ingo Erdmann

Groupware Competence Center
Universität Paderborn

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

Referent

- **Groupware Competence Center**
  - Leitung: Prof. Dr. Ludwig Nastansky
  - Praxisnahe Lehre
  - Zertifizierung
  - Forschung
  - Projekte und Technologietransfer mit Praxispartnern
- **Ingo Erdmann**
  - Wissenschaftlicher Mitarbeiter
  - Leitung des Notes/Domino 7 beta Programms
  - 8 Jahre Erfahrung im Notes/Domino Umfeld
  - IBM Certified Advanced Instructor
  - Trainer, Entwickler, Admin

Agenda

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **1 Einleitung**
  - N/D ⇔ DB2 Paradigmen
  - Einführung Domino Access und Query Views
- **2 Domino Access Views (DAV)**
  - Szenarien
  - Technologie, Tips
  - Live Demo
- **3 Query Views**
  - Szenarien
  - Technologie, Tips
  - Föderierte Daten
  - Live Demos
- **4 Zusammenfassung**

1 Einleitung

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Session Fokus auf DAV & Query Views in NSFDB2**
- **NSFDB2 Grundlagen werden nicht behandelt**
- **Betrachtung spezifischer Aspekte von NSFDB2**
- **Ziele**
  - Einordnung von N/D ⇔ DB2 Integration
  - Präsentation innovativer & effizienter N/D ⇔ DB2 Integrationsmöglichkeiten
  - Aufzeigen neuer Ansätze für N/D & DB2 Anwendungsentwicklung
  - Anregung über notwendige Infrastrukturänderungen nachzudenken
  - Vermittlung von Tips für erste Implementierungen

Notes/Domino & DB2 Integration

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **RDB Integration wird seit Release 2 angestrebt**
- **Viele erfolgreiche Ansätze für Integration**
  - DECS, LEI, SAP connector, …
- **NSFDB2 ist der radikalste Integrationsansatz**
  - Sehr einfach zu handhaben
- ***Nun müssen die DB2 Leute endlich zugeben, dass Domino auf einer "richtigen" Datenbank Engine basiert***
- **Jedoch: Die Kernparadigmen unterscheiden sich weiterhin deutlich**
- **N/D ⇔ RDB Integration betrifft nicht ausschließlich technologische Aspekte**
- **Vielmehr geht es um Anwendungsdomänen, IT-Strategie, Top-down System Architekturen, Funktionale Konzepte**

Documente vs. Relationale Daten

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

N/D Welt

- Strategische Orientierung & Kommunikationszentriert
- Knowledge & Information Management
- Werkzeug Paradigmen objektorientiert – Wiederverwendung von Code
- Verbunddokumente, semistrukturierte Daten, flexible Datenstrukturen und -typen
- Multimedia, Links, Eingebettete Methoden und Objekte
- Dezentral, Bottom-up, User-Workplace & Collaboration zentriert
- Replikation, Information Sharing, robuste Verteilung, Redundanz, Message Objekte
- Unterstützung von mobilen, nomadischen und offline User Workplaces

DB2/RDB Welt

- Operative Orientierung & Datenzentriert
- Verarbeitung hoher Transaktionsvolumen
- Automatisierungsparadigma auf Daten – Code Effizienz
- Datensätze, Tabellen, strukturierte Daten, Begrenzte Flexibilität, strikte Formate
- Transaktionen, dynamische Daten
- Zentrale Organisation, Top-down, systemzentriert
- Zugriffskoordination, Datenintegrität, Redundanzfreiheit durch Normalisierung, 2-Phase commit, ACID
- Statischer Office- und Server-basierter Workplace

Integration, Föderation

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Herausforderung: Das Beste aus beiden Welten zusammenbringen**
  - Soweit, wie es benötigt wird
  - Soweit es aus wirtschaftlicher Sicht Sinn macht
- **Position: N/D ⇔ RDB/DB2 Integration ist nur eine von vielen System Integrationsaufgaben, die aktuell stattfinden**
- **Integration zwischen gleichen Partnern**
  - Community: Cool bleiben, Glaubensfragen ausklammern, nicht verteufeln
- **Zweck**
  - Datenintegration über Systemgrenzen hinaus
  - Zusammenarbeit unterschiedlicher Applikationswelten
  - Föderation von N/D Applikationen mit DB2-Daten
  - Föderation von DB2 Applikationen mit N/D Daten
- **NSFDB2 bringt RDB/SQL Funktionen in die N/D Welt**
- **Zu Beginn: Diskussion eines typischen Modellierungsproblems in einem Standard N/D Szenario**

Szenario 1 - Modell Employee - Room Relation

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

Start: Zwei unabhängige Listen

N/D-View: Employees

Last Name	First name	Department
Geyer	John	Finance
Haas	Christine	Sales
Kwan	Sally	Procurement
Thompson	Michael	Procurement

N/D-View: Room list

Room (Building-floor.number)	Type	Floorspace
A1.122	Office	16
A2.035	Canteen	26
C1.245	Office	12
D2.133	Meeting	38
C4.034	Office	16

Ziel: Modellierung von Employee-Room Relation

N/D-View: Employee in room

Last Name	First name	Department	Room (Building-floor.number)
Geyer	John	Finance	A1.122
Haas	Christine	Sales	C4.034
Kwan	Sally	Procurement	C1.245
Thompson	Michael	Procurement	C1.245

#1

N/D-Form: Employee

Last name:	Haas
First name:	Christine
Department:	Sales
... ... etc.	
Room:	C4.034

N/D-Form: Room

Room:	C4.034
Type:	Office
Floorspace:	16
... ... etc.	
Last name:	@DBLookup in Employee View

Lösung in N/D ist nicht sauber

Relationale Abhängigkeiten müssen hart codiert werden indem Daten kopiert und redundant gespeichert werden

- Option 1: Lookup von "LastName" in "Employees" Ansicht und redundante Speicherung in Room Dokument
- Option 2: Lookup von "Room" in "Room list" Ansicht und redundante Speicherung in Employee Dokument
- [Option 3: Konsistente Benutzerführung in abhängigen Dokumenten forcieren]

#2

N/D-Form: Employee

Last name:	Haas
First name:	Christine
Department:	Sales
... ... etc.	
Room:	@DBLookup in Room View

N/D-Form: Room

Room:	C4.034
Type:	Office
Floorspace:	16
... ... etc.	
Last name:	Haas

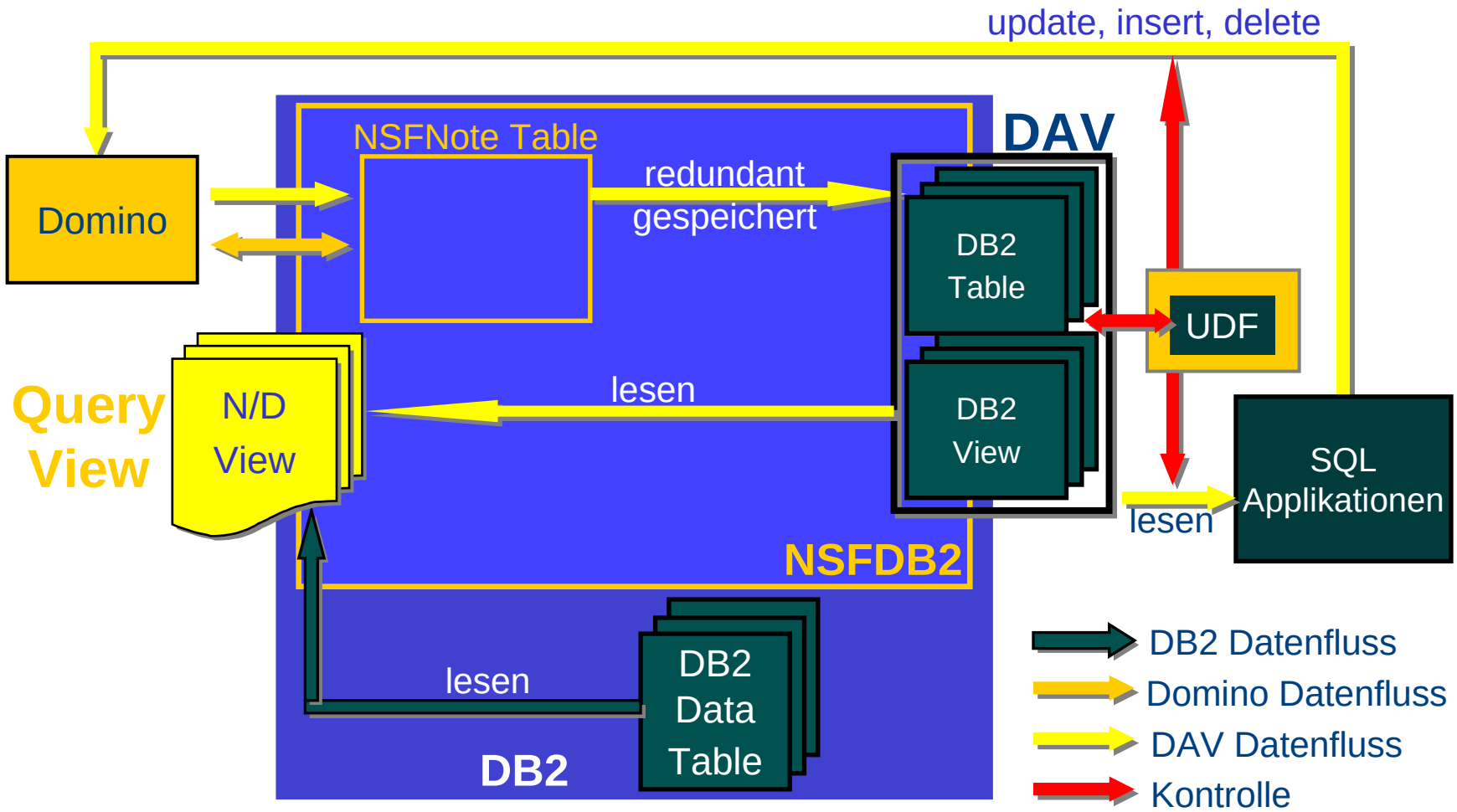
Szenario 1 - Modell Employee - Room Relation

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Problem:**
  - Ansicht "EmployeeInRoom" kann nicht ohne Duplizierung des "Room"-Feldes realisiert werden
  - Gleiches gilt für das "LastName"-Feld in "Employee"- und "Room"-Dokument
  - Grund: @DBLookup kann nicht in Ansichten verwendet werden
- **Synchronisation muss modelliert werden**
  - Möglich aber komplex und schlecht wartbar
  - Agenten sammeln Änderungen und führen Aktualisierung durch
  - Kopieraktionen von Dokumenten u. ä. müssen überwacht werden
  - Sehr ausgefeilte User Navigationsmethoden können implementiert werden, um Konsistenz zu wahren
  - Redundanzvermeidung würde dieses Problem deutlich verringern
- **Lösung durch neue Optionen von NSFDB2**
- **Daten Duplikation kann durch den Einsatz von Domino Access Views (DAV) und Query Views vermieden werden**

DAV vs. Query Views

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky



Agenda

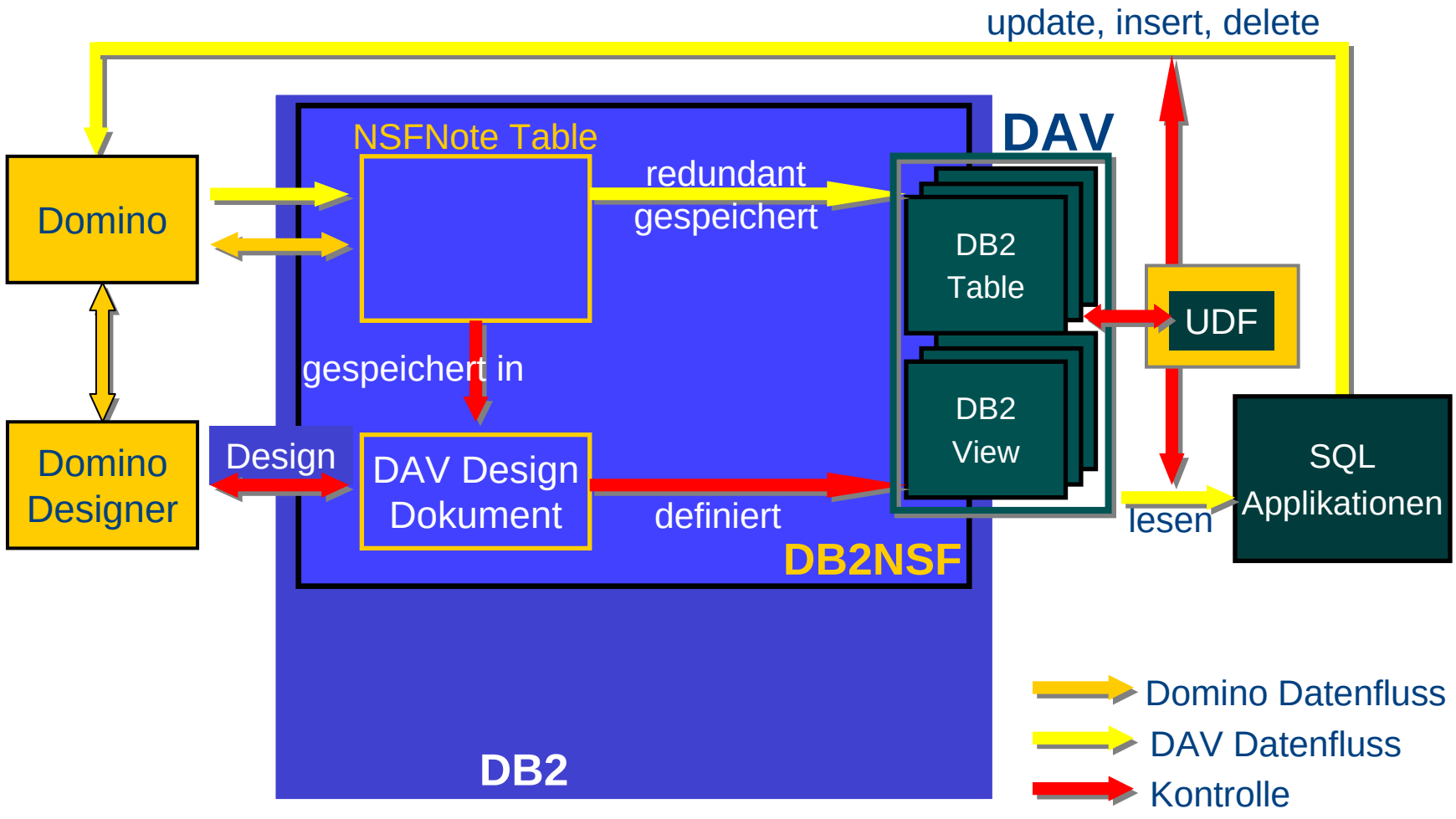
- 1 Einleitung
  - N/D ⇔ DB2 Paradigmen
  - Einführung Domino Access und Query Views
- **2 Domino Access Views (DAV)**
  - Szenarien
  - Technologie, Tips
  - Live Demo
- 3 Query Views
  - Szenarien
  - Technologie, Tips
  - Föderierte Daten
  - Live Demos
- 4 Zusammenfassung

DAV - Übersicht

- **Redundante Speicherung von N/D Daten in DB2 Table**
- **Zur DB2 Table korrespondierender DB2 View als Interface für SQL Applikationen**
- **DAV ist kein N/D UI Element, also keine N/D Standard Ansicht**
- **DAV ist eine Tabellen-orientierte Entität für N/D⇔DB2 Datenaustausch**
- **Analog zu N/D Ansicht, aber nicht durch den Notes Client zugreifbar**
- **Stellt N/D Daten zur Verfügung für**
  - SQL Applikationen
  - JDBC, ODBC
  - DB2 Tools und Applikationen
  - Relationale Reporting Tools (z. B. Crystal Reports)
  - Domino Query Views
- **DECS, LEI und Connectors werden nicht benötigt**

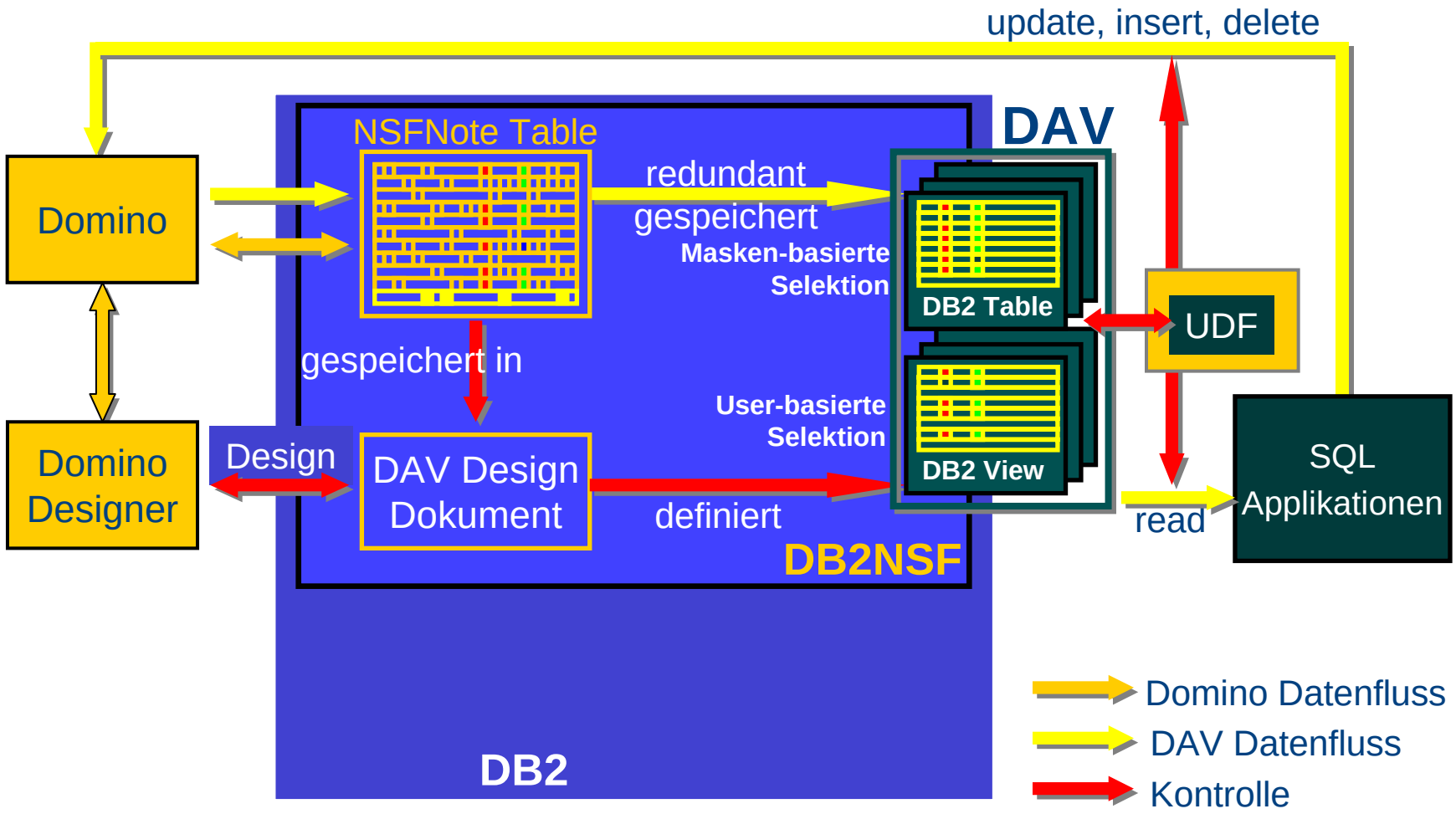
DAV Architektur

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky



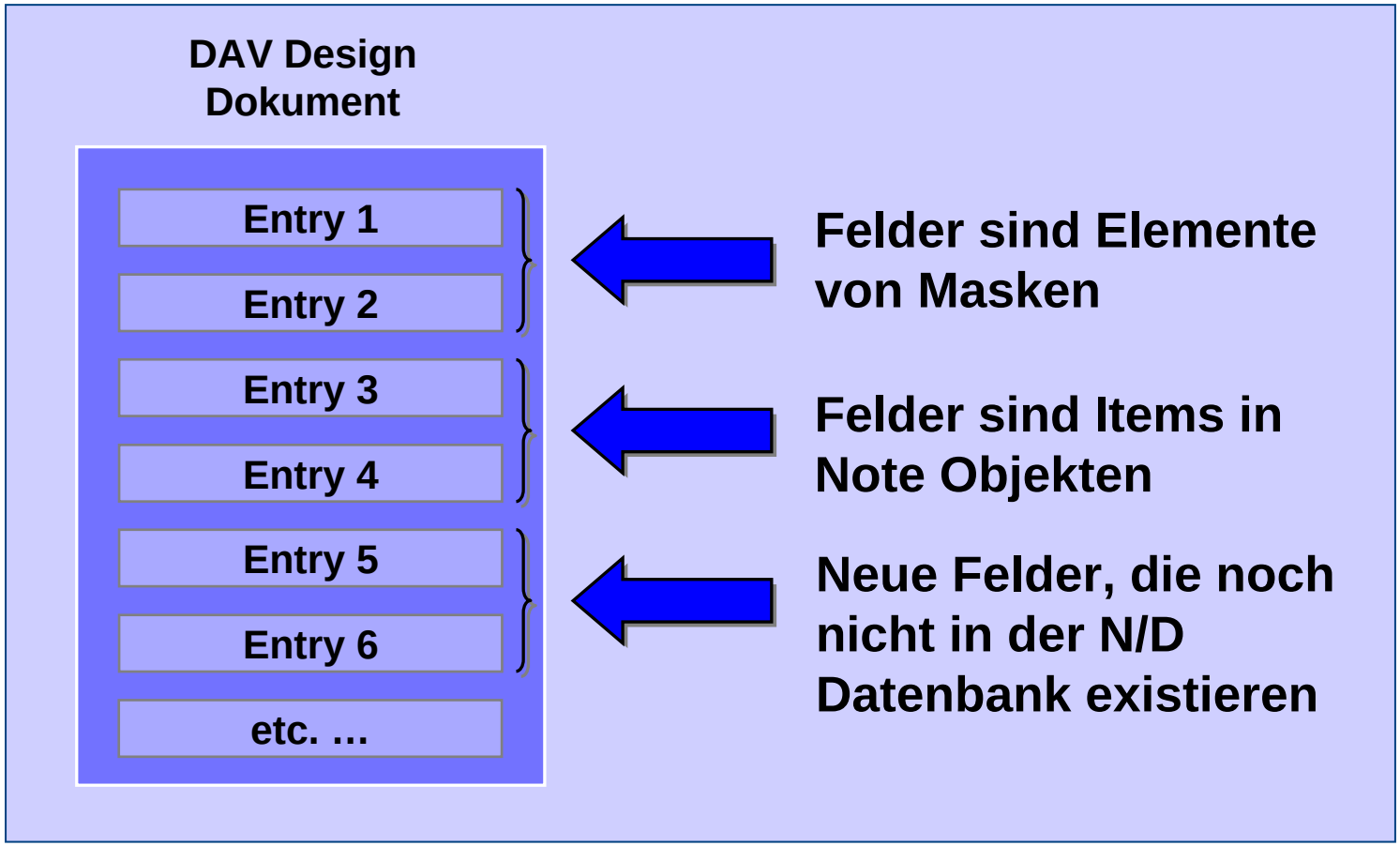
DAV Data: N/D⇔DB2 Interaktion

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky



DAV Entry Design

DAV Entries definieren die Zuordnung von N/D Feldern und DB2 Feldern (Spalten)



Architektur - Zusammenfassung

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Verwendung von DAV ermöglicht in SQL:**
  - N/D Daten unter Einhaltung von N/D Sicherheit zu lesen
  - SQL insert, update und delete unter Einhaltung von N/D Sicherheit
- **Aus SQL Sicht erweitern DAV die DB2 Konzepte um "Zeilensicherheit"**
- **DB2 ist für Sicherheit bei Lese-Operationen zuständig**
- **DB2 Access for Lotus Domino (aka UDF server) sorgt für Einhaltung von N/D Sicherheit bei insert, update, delete**
  - DAV richtet Operationen an UDF Server
  - Domino übernimmt Management von Replication- und Save-Konflikten, Document locking etc.
  - Domino wertet ACL, Reader Felder usw. aus
  - User mapping ist notwendig für Sicherheitsfunktionen (Domino Administrator)

Szenario 2: Integration #1

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **N/D und DB2 Applikations- und Datenintegration wird notwendig**
  - Datenanalyse mit MIS
  - Reporting Funktionalitäten mit Crystal Reports
  - HR verwendet existierende DB2 Applikation
  - Salesforce setzt mobile N/D Applikation ein
  - Bestellungen werden durch DB2 Applikation verarbeitet
  - Kunden verwenden J2EE Applikation mit DB2 Backend um Bestellungen aufzugeben
- **Herausforderung**
  - Applikationsintegration
  - Synchronisierung von Daten

Szenario 2: Integration #2

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Lösung: Integration durch DAV**
  - N/D Dokument-basierte Daten werden durch DAV konzernweiten DB2 Applikationen zur Verfügung gestellt
    - N/D Daten stehen für DB2 Lese- und Schreiboperationen zur Verfügung
  - Populierung von N/D Dokumenten mit DB2-basierten Daten via DAV
    - DB2 Daten stehen N/D Dokumenten zur Verfügung
    - DB2 Daten stehen im flexiblen N/D Ansichts Kontext zur Verfügung
- **Vorteile**
  - N/D Umgebung wird durch DB2 Applikationsoptionen aufgewertet
  - DB2 Umgebung wird durch N/D basierte Applikationsoptionen aufgewertet

Szenario 2: Mobile Salesforce

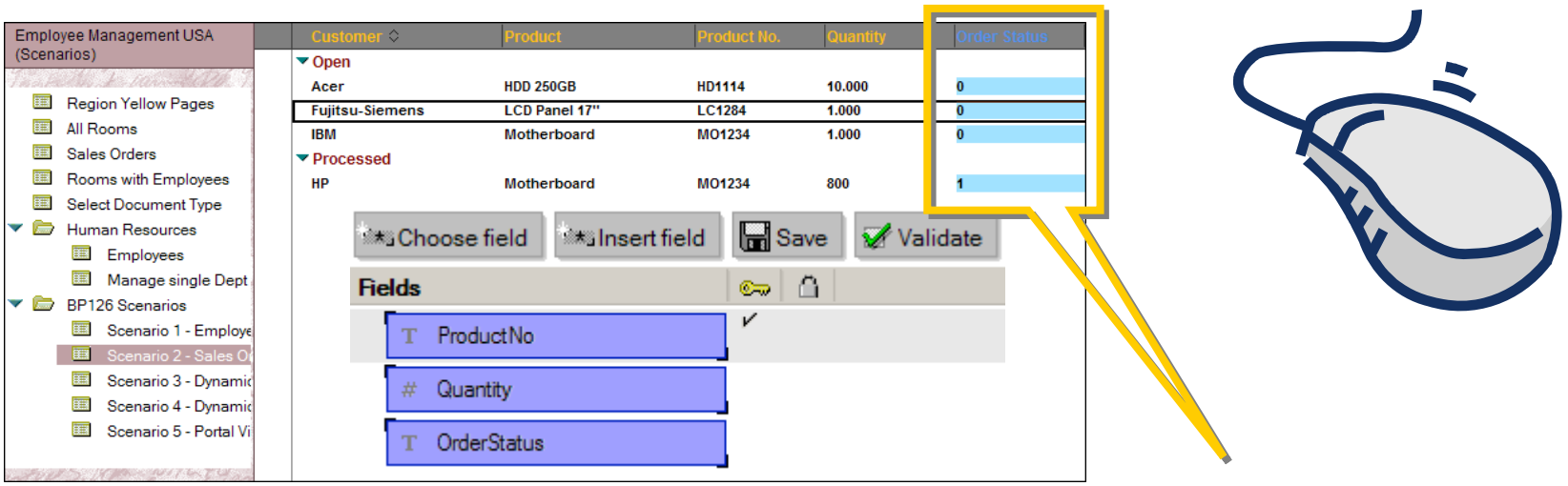
University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Salesforce verwendet mobile N/D Applikation**
- **Bestellungen werden in eine zentrale NSFDB2 repliziert**
- **DB2 Applikation verarbeitet Daten, die über DAV zur Verfügung gestellt werden**
  - N/D fügt DB2 Daten über Replikation offline Funktionalität hinzu
  - Transaktionen werden in DB2 verarbeitet
- **Salesforce repliziert erneut und erhält den Transaktionsstatus auf die mobilen Geräte**

Demo 1: DAV Design und Nutzung

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

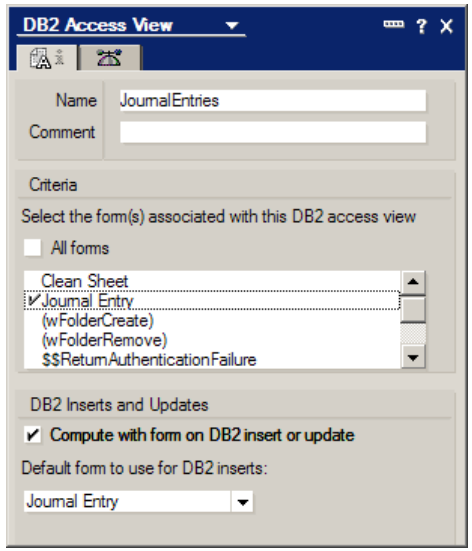
- Design einer Domino Access View
- Erstellen und populieren einer DAV
- Untersuchung der erstellten Elemente in DB2
- Modifikation von Daten aus einer SQL Applikation



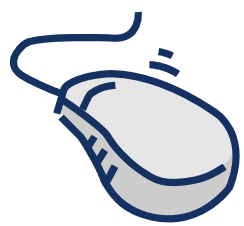
Wird durch DB2 Applikation aktualisiert

DAV Design - #1

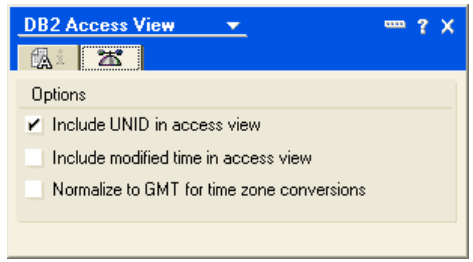
- Datenbank muss NSFDB2 Format aufweisen
- Erstellung in DB2 mit Domino Designer
- Neue Shared Resource
- DAV Selection basiert auf einer oder mehreren Masken
- Zum Erstellen neuer Dokumente wird eine Maske definiert



Definieren der erweiterten Eigenschaften

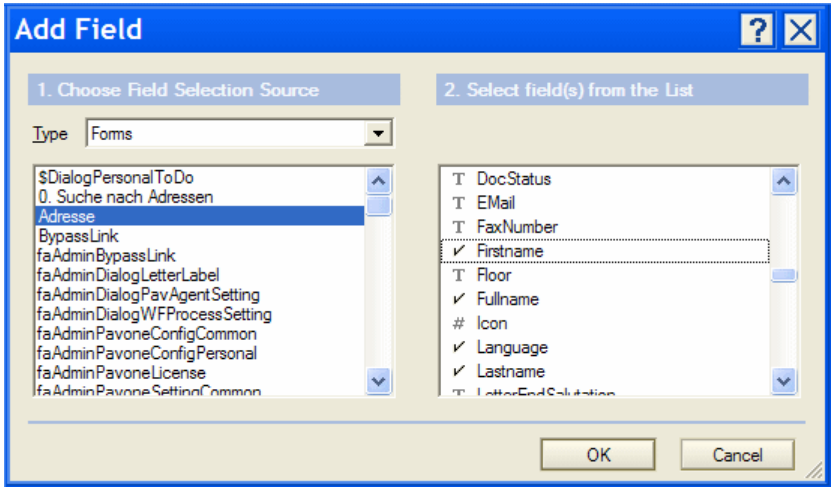


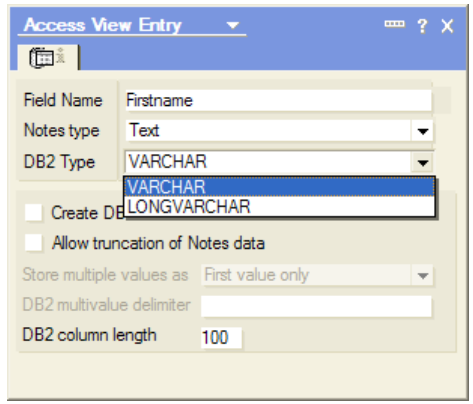
- UNID muss enthalten sein, um Dokumente in Query Views öffnen zu können
- Änderungsdatum



DAV Design - #3

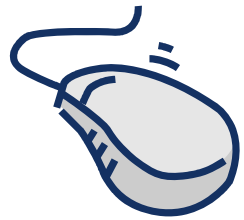
- **Auswahl einer Maske**
- **Auswahl von N/D Feldern**
- **Modifikation der ausgewählten Feld Definitionen**
  - DB2 Datentyp (Automatisches Default Mapping)
  - Definition der DB2 Feldlänge (Spalte)

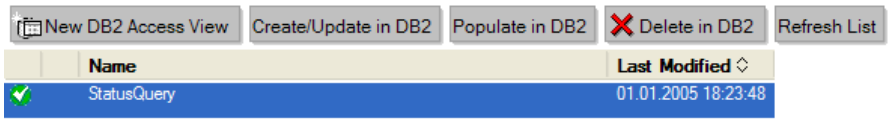




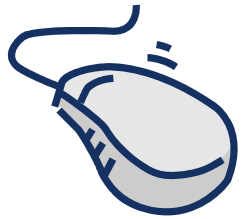
DAV Design - #4

- **Erstellen des DAV in DB2**
- **Populieren des DAV**
  - Asynchroner Hintergrund Task DAVPOP
  - Kann in großen Datenbanken einige Zeit in Anspruch nehmen

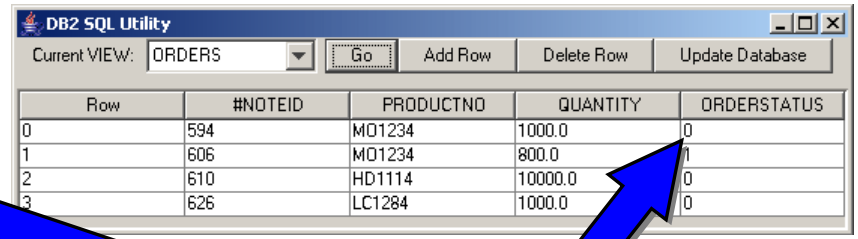


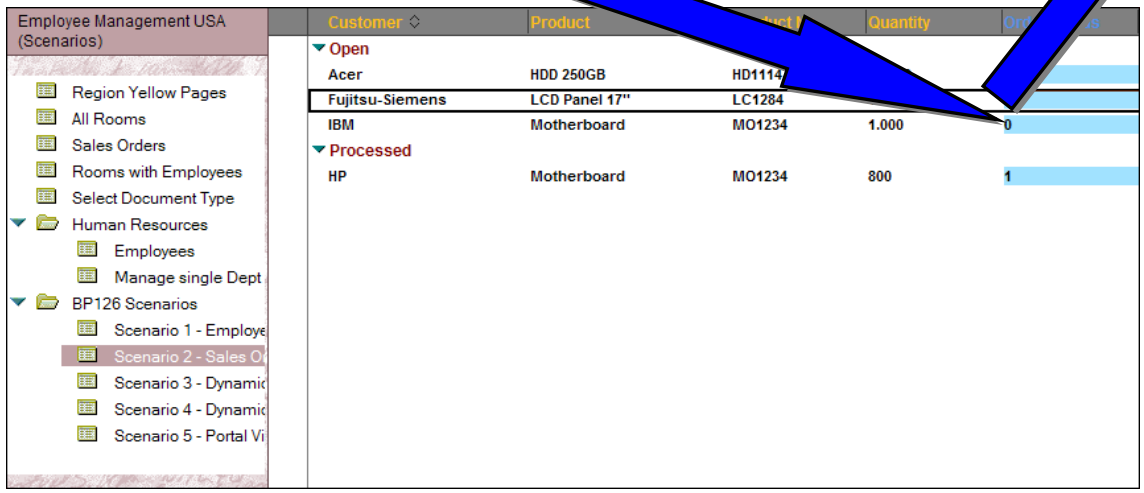


DAV Nutzung



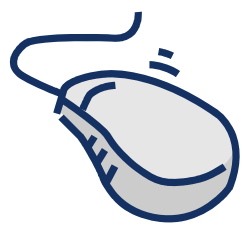

Order
ORDER
Customer Name: IBM
Customer No.: 1234
Sales Rep: CH-32124-SA
Product: Motherboard
Product No: MO1234
Quantity: 1.000
Order Status: Open


DB2 SQL Utility
Current VIEW: ORDERS
Go
Add Row
Delete Row
Update Database
Row | #NOTEID | PRODUCTNO | QUANTITY | ORDERSTATUS
0 | 594 | MO1234 | 1000.0 | 0
1 | 606 | MO1234 | 800.0 | 1
2 | 610 | HD1114 | 10000.0 | 0
3 | 626 | LC1284 | 1000.0 | 0

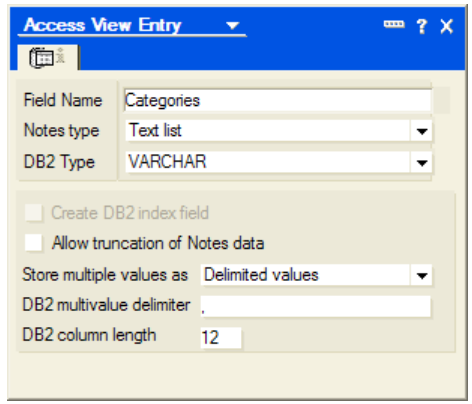

Employee Management USA (Scenarios)
Region Yellow Pages
All Rooms
Sales Orders
Rooms with Employees
Select Document Type
Human Resources
Employees
Manage single Dept
BP126 Scenarios
Scenario 1 - Employe
Scenario 2 - Sales O
Scenario 3 - Dynamic
Scenario 4 - Dynamic
Scenario 5 - Portal Vi
Customer
Product
Quantity
Open
Acer HDD 250GB HD1114
Fujitsu-Siemens LCD Panel 17" LC1284
IBM Motherboard MO1234 1.000 0
Processed
HP Motherboard MO1234 800 1

DAV Anmerkungen - #1

- **Strukturbruch zwischen N/D Flexibilität und statischen DB2 Tabellen**
  - DB2 setzt feste Feldlängen (Spalten) voraus
  - Insbesondere N/D Textfelder müssen bedacht werden
  - Truncated Data sollte nicht in SQL modifiziert werden



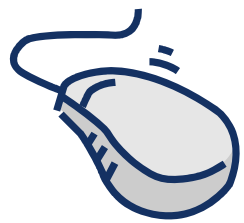
- **Verarbeitung von Mehrfachwerten muss berücksichtigt werden**
  - Summierung der Länge der Werte
  - Delimiter
- **Listenfelder**
  - Ggf. DB2 Feldlänge reduzieren
  - Verwendung eines Alias spart Speicherplatz



DAV Anmerkungen - #2

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- UDF Server wird benötigt
- Das Design Dokument repliziert in NSF, ist aber lokal nicht sichtbar
- Formula, RichText und RichText Light Datentypen werden nicht unterstützt
- Es sollten nur Felder verwendet werden, die unbedingt benötigt werden (Redundanz)



DAV Zusammenfassung

- **Domino Designer ist Entwicklungswerkzeug für DB2**
- **N/D Daten werden SQL Applikationen zur Verfügung gestellt**
- **N/D Funktionalitäten werden DB2 Applikationen zur Verfügung gestellt**
  - N/D Mechanismen ermöglichen Zeilensicherheit in SQL Applikationen
  - DB2 wird durch offline Optionen aufgewertet
- **Kosten für die Integration von Applikationen in bestehende Infrastrukturen können potentiell gesenkt werden**

Agenda

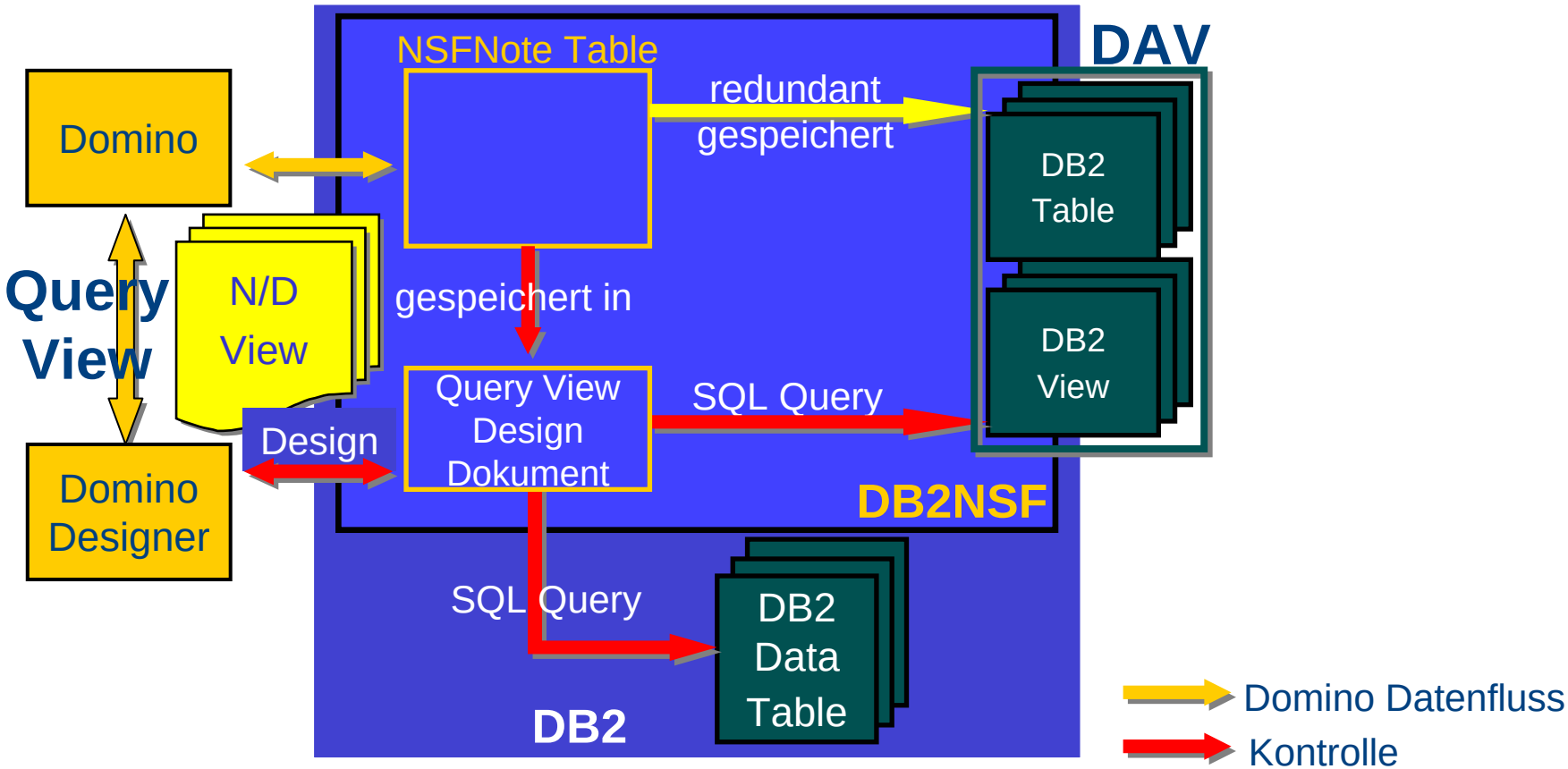
- 1 Einleitung
  - N/D ⇔ DB2 Paradigmen
  - Einführung Domino Access und Query Views
- 2 Domino Access Views (DAV)
  - Szenarien
  - Technologie, Tips
  - Live Demo
- **3 Query Views**
  - Szenarien
  - Technologie, Tips
  - Föderierte Daten
  - Live Demos
- 4 Zusammenfassung

Query Views - Übersicht

- **Zeigen die Ergebnistabelle einer SQL Query (Result Set) in einer N/D Ansicht an**
- **SQL Queries werden im View Selection Formula Kontext verwendet**
- **Beispiele für Nutzung**
  - Filtern von Dokumenten (Dynamische View Selection Formula)
  - Anzeigen von DB2 Daten in einer N/D Ansicht
  - Verwendung von DB2 Daten in Spaltenformeln
  - Kombinierte Darstellung von N/D Daten und externen DB2 Daten

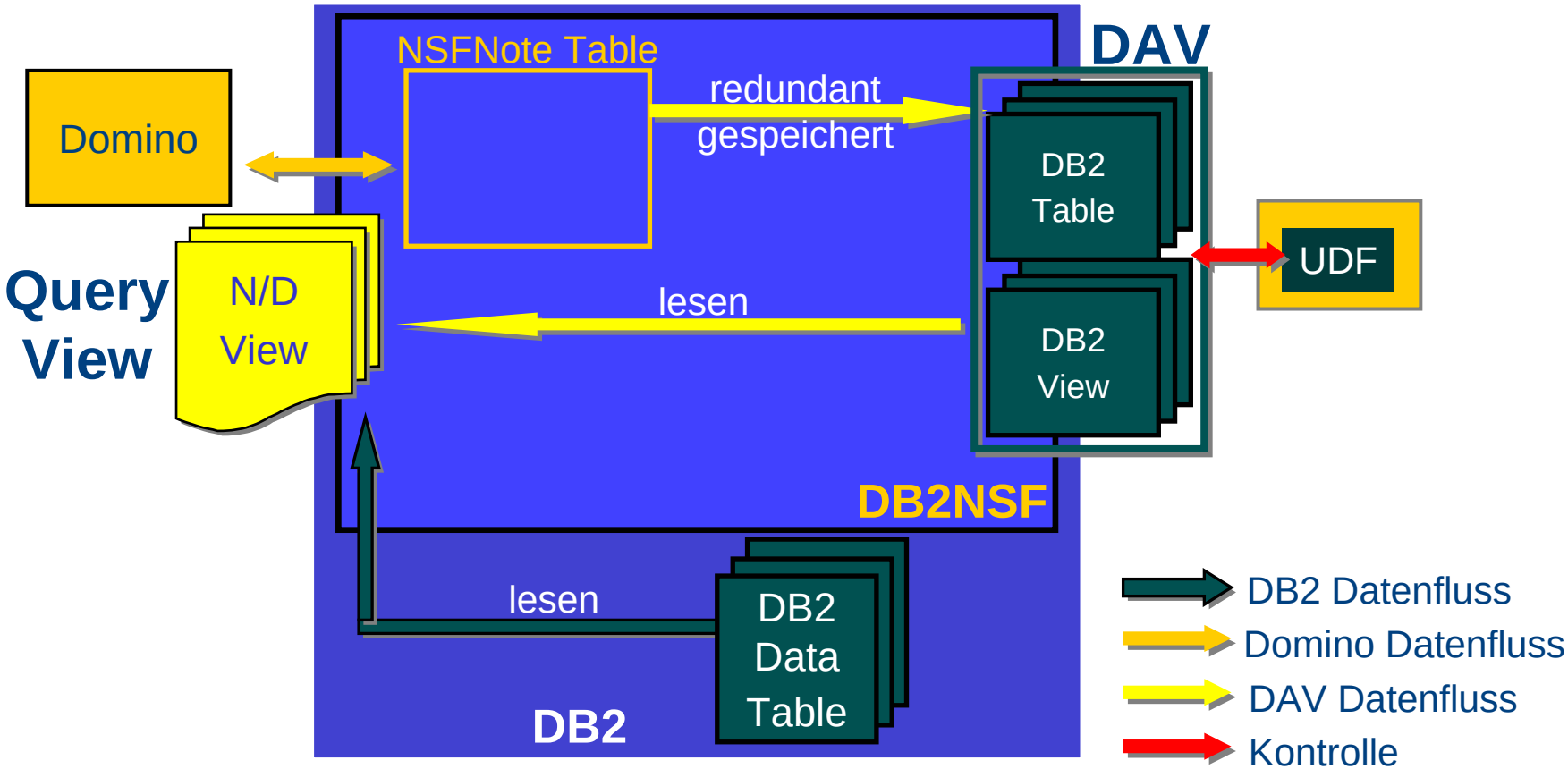
Query Views - Design

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky



Query Views – Datenfluss

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky



Query Views – Daten Föderation #1

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Zeilen können aus Daten verschiedener Quellen bestehen**
  - N/D Dokumente der aktuellen Datenbank (nur via DAV)
  - DAV Daten aus DB2, die nicht aus der aktuellen Datenbank stammen
  - DB2 Daten
  - Kombinationen aus den genannten Quellen via SQL JOIN
- **Query Views können also Daten aus folgenden Quellen darstellen**
  - Aktuelle N/D Datenbank über DAV
  - Andere N/D Datenbanken über DAV
  - N/D Dokumente (über DAV) gemeinsam mit weiteren Daten über SQL JOIN
  - Native (nicht aus N/D stammende) DB2 Daten
- **Aber was passiert beim Doppelklick auf Zeilen in Query Views?**

Query Views – Daten Föderation #2

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **"Föderierte Daten" stammen nicht aus der aktuellen N/D Datenbank**
- **Eine Zeile kann Daten aus Dokumenten gemeinsam mit föderierten Daten anzeigen**
  - DB2 Datenobjekte, die per SQL Query abgefragt wurden
  - "Normale" N/D Feldinhalte
    - Müssen in DAV vorhanden sein
    - Müssen im SQL Result Set enthalten sein
  - Ergebnis einer Spaltenformel
  - "Doppelklick" kann zum Öffnen eines Dokumentes führen
    - UNID muss in DAV enthalten sein

Query Views – Daten Föderation #3

- **Eine Zeile kann ausschließlich aus föderierten Daten bestehen**
  - Alle Daten stammen aus DB2 Objekten
  - "Doppelklick" auf eine Zeile macht (in den meisten Fällen) keinen Sinn und führt zu einer Fehlermeldung
  - Daten aus mehreren NSF Datenbanken können für den User konsolidiert werden
    - Aus einer Applikation
    - Aus mehreren Applikationen
    - Siehe "Szenario 5"

Query Views – SQL Query Regeln

- **Die Query wird im N/D Formula Language Kontext definiert**
- **Die Query unterstützt Standard SQL**
  - SQL JOIN
  - SQL UNION
  - ORDER BY
  - etc.
- **Queries die kein Result Set liefern sind nicht zugelassen**
  - Sicherheitsgründe
  - Verhindert Löschung und Veränderung von Dokumenten aus dem View Selection Kontext

Query Views sind dynamisch

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Es existiert kein persistenter View Index**
  - Effizientes DB2 Indexing wird verwendet
  - Queries können User-spezifisch sein
  - Queries können parametrisiert und personalisiert sein
  - Lookups auf N/D Daten zur Konstruktion der Query sind erlaubt

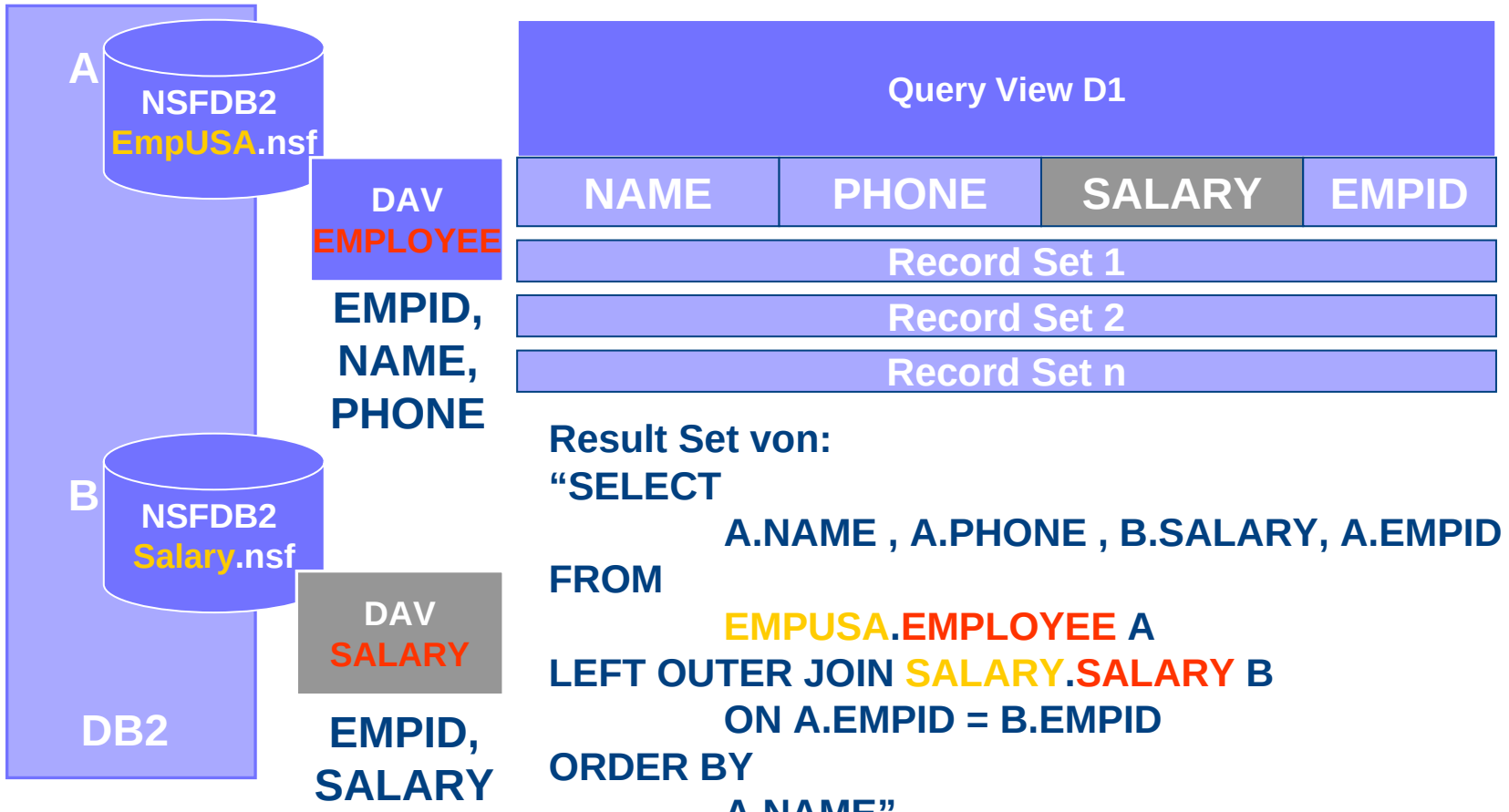
Scenario 3 – Föderierte Daten in Ansichten

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **HR verwendet N/D um Daten von Beschäftigten zu verwalten**
- **Jeder darf den Namen und die Telefonnummer sehen**
- **Jedoch hat nur HR das Recht, den Lohn einzusehen**
- **Herausforderung für Sicherheitsmanagement**
  - Feldverschlüsselung erfordert kompliziertes Schlüsselmanagement
  - "Hide when" Optionen sind keine Sicherheitsfeatures
  - @DBLookup ist in Spaltenformeln nicht erlaubt
- **Lösung**
  - Lohndaten werden in einer anderen Datenbank oder einem anderen Dokument gespeichert
  - Query Views und SQL JOIN werden verwendet, um die nativen Dokumente plus die föderierten Daten anzuzeigen

SQL JOIN

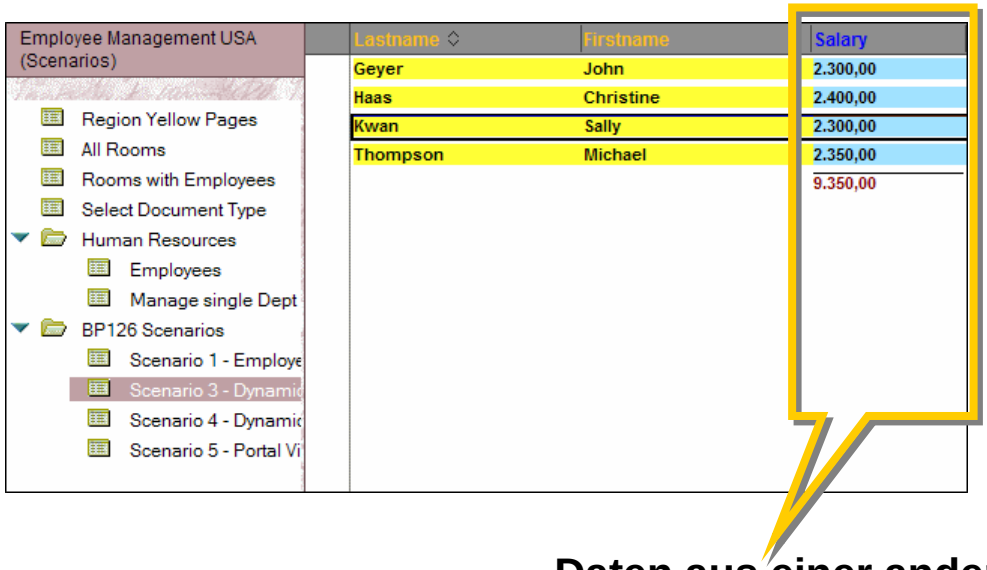
University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

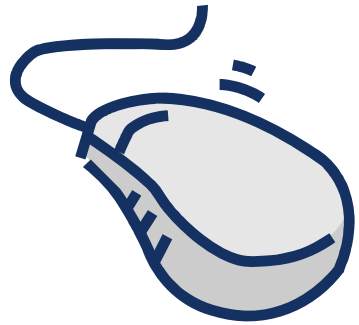


Result Set von:

```
"SELECT
        A.NAME , A.PHONE , B.SALARY, A.EMPID
FROM
        EMPUSA.EMPLOYEE A
LEFT OUTER JOIN SALARY.SALARY B
        ON A.EMPID = B.EMPID
ORDER BY
        A.NAME"
```

- Design einer Query View
- JOIN von Daten aus einer anderen N/D Datenbank

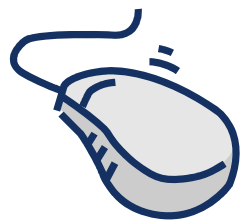




Daten aus einer anderen Datenbank

Anmerkungen

- **In den meisten N/D Szenarien ist LEFT OUTER JOIN die richtige Wahl**
  - Stellt sicher, dass alle Dokumente angezeigt werden
  - Wenn kein JOIN Match existiert bleibt die Spalte leer
- **Nur eine NoteID pro Result Set**
  - Spezifisches Select Statement anstatt "*" verwenden
  - Ansonsten könnte das falsche Dokument geöffnet oder eine Fehlermeldung ausgelöst werden



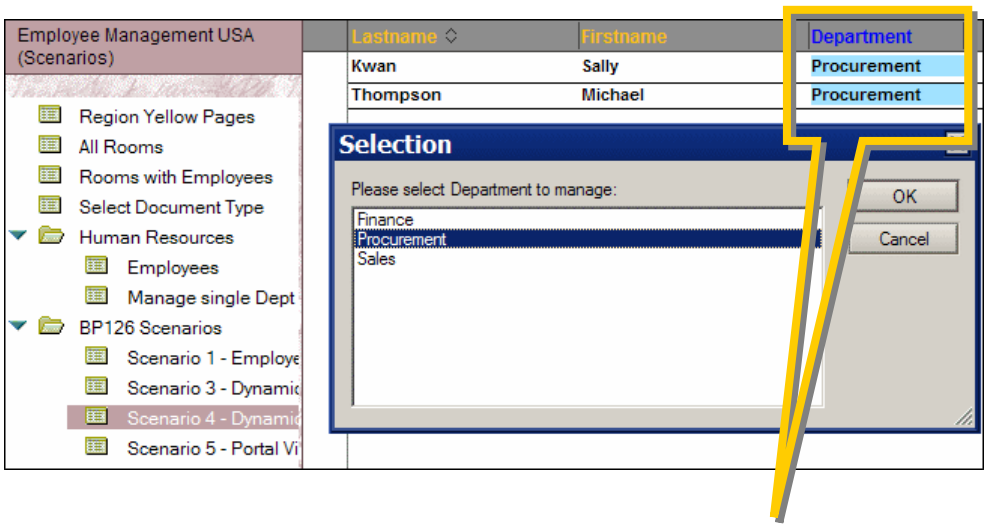
Scenario 4 – Dynamische Selektion

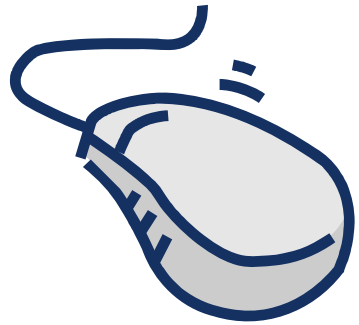
University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Workflow Applikation**
- **Große Anzahl von Dokumenten**
- **Nutzer benötigen lediglich eine geringe Anzahl der Dokumente**
- **Nutzer wollen regelbasiert auswählen können, welche Dokumente angezeigt werden sollen**
- **Nutzer wollen persönliche Voreinstellungen bei der Selektion von Dokumenten nutzen**
- **Herausforderung (im N/D Kontext)**
  - @DBLookup ist in Selection Formula nicht zulässig
  - @UserName funktioniert nur in privaten Ansichten
  - Schlechte Performance durch großen View Index
- **Lösung**
  - Query Views haben keinen persistenten statischen Index
  - Queries können aus dynamischen N/D Formeln konstruiert werden

Demo 3

- Design einer Query View
- Dynamische Selektion mit @Prompt
- Dynamische User Voreinstellungen





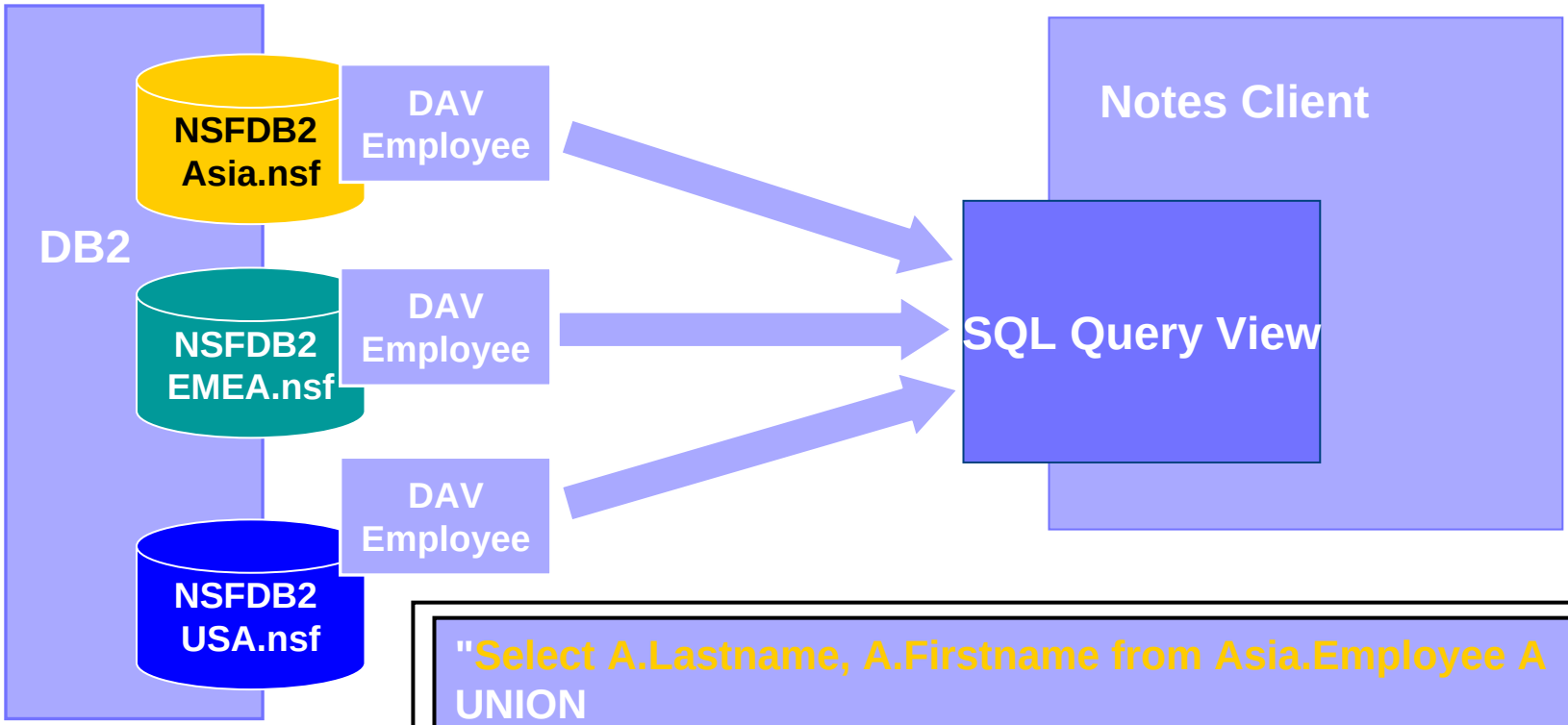
Durch User selektierte Dokumente

Query Views – SQL UNION

- **Liefert föderierte Datensätze und Dokumente**
- **Erlaubt das Design von aggregierten Ansichten**
  - Dynamische Formeln erlauben User-spezifische Aggregation von Dokumenten
  - Dokumente aus mehreren N/D Datenbanken können in einer einzelnen Ansicht zu einem "Single Point of Access" aggregiert werden

SQL UNION – Aggregierte Ansichten

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky




```
"Select A.Lastname, A.Firstname from Asia.Employee A
UNION
Select B.Lastname, B.Firstname from EMEA.Employee B
UNION
Select C.Lastname, C.Firstname from USA.Employee C"
```

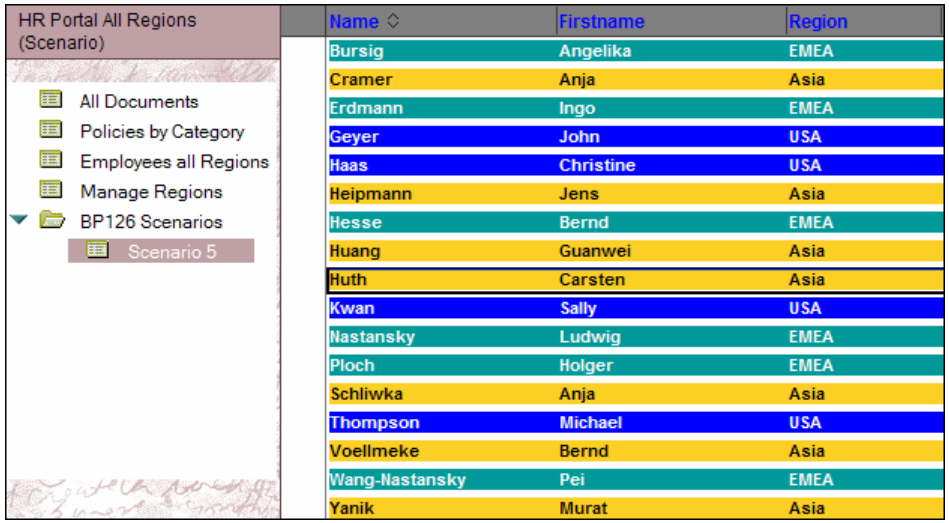
Scenario 5 – Aggregierte Ansichten

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

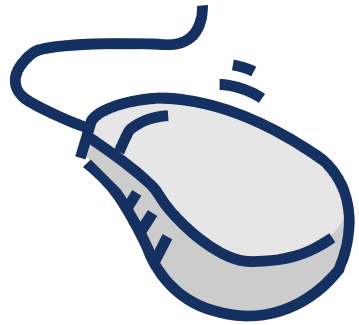
- **Aufgaben der User sind über mehrere Applikationen verstreut**
  - Aktueller Bearbeiter im Workflow
  - Beteiligung in mehreren Diskussionsforen/Teamrooms
  - To Do in einer Projektmanagement Applikation
- **Herausforderung**
  - User muss mehrere Datenbanken öffnen, um Aufgaben zu bearbeiten
  - User hat keinen Überblick über alle aktuellen Aufgaben, kann schlecht Priorisieren
- **Lösung**
  - Eine zentrale Ansicht zeigt alle relevanten Dokumente aus mehreren Datenbanken an, die ein User benötigt
  - Verwendung von Query Views und SQL UNION um föderierte (virtuelle) "Dokumente" anzuzeigen
  - Die Dokumente, die zu den föderierten Daten passen müssen programmatisch referenziert werden

Demo 4

- Design einer aggregierten Ansicht
- Anzeige von Daten externer Dokumente aus anderen Datenbanken in einer Query View
- Implementierung des Programmcodes zum Öffnen der externen Dokumente

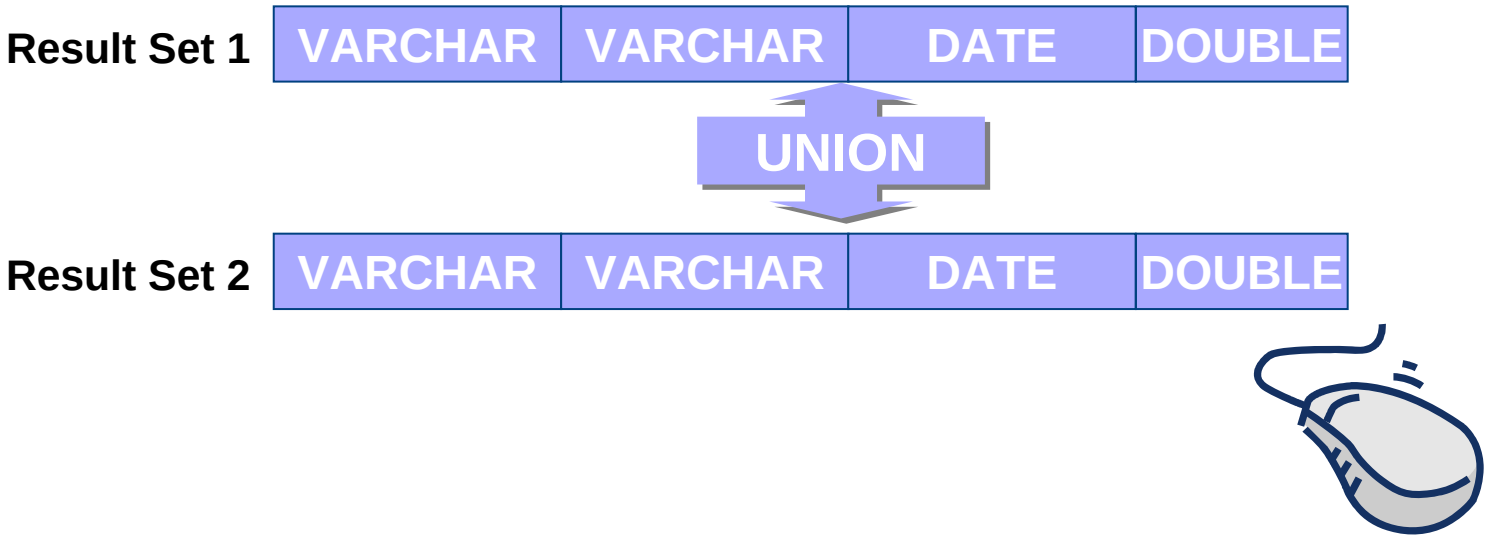


Name	Firstname	Region
Bursig	Angelika	EMEA
Cramer	Anja	Asia
Erdmann	Ingo	EMEA
Geyer	John	USA
Haas	Christine	USA
Heipmann	Jens	Asia
Hesse	Bernd	EMEA
Huang	Guanwei	Asia
Huth	Carsten	Asia
Kwan	Sally	USA
Nastansky	Ludwig	EMEA
Ploch	Holger	EMEA
Schliwka	Anja	Asia
Thompson	Michael	USA
Voellmeke	Bernd	Asia
Wang-Nastansky	Pei	EMEA
Yanik	Murat	Asia



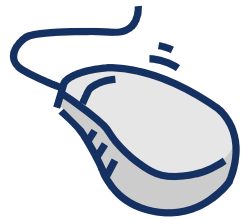
SQL UNION Anmerkungen

- Zu berücksichtigen ist, dass die Reihenfolge der DAV Felder die Spaltenreihenfolge der DB2 View bestimmt
- Die Anzahl der Felder in allen Result Sets muss gleich sein
- Die Datentypen von zusammengefügten Spalten müssen übereinstimmen



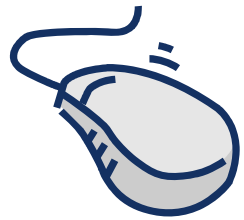
Zuordnung von föderierten Daten zu Dokumenten - #1

- **Eine Query View enthält Zeilen mit föderierten Daten**
  - Stehen in keiner Beziehung zu Dokumenten aus der aktuellen Datenbank
  - Daten stammen aus N/D Dokumenten externer Datenbanken
- **Dokument-bezogene N/D Objekte sind nicht verfügbar**
  - Document collection
  - Document context
  - Caret note ID
- **View entry Objekte (Zeilen) sind verfügbar**
  - Enthalten die föderierten Daten
- **Caret Category kann verwendet werden, um den Zeilenkontext festzustellen**
  - Setzt eindeutigen Wert in der ersten sortierten Spalte voraus



Zuordnung von föderierten Daten zu Dokumenten - #2

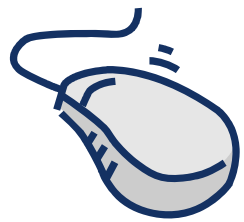
- **Notwendige Daten zum Zugriff auf das externe Dokument müssen in (verborgenen) Spalten enthalten sein**
  - UNID / Note ID
  - Servername
  - Pfad oder Replica ID
- **Abfangen des QueryOpenDocument Events**
- **Erstellung eines Backend Document Objects, welches das externe Dokument repräsentiert**
- **Öffnen des externen Dokumentes im Notes Client UI**



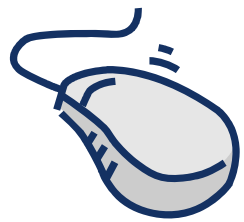
Föderierte Daten - Anmerkungen

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Funktioniert (aktuell) nicht mit kategorisierten Ansichten**
  - Caret Category enthält dann nicht den eindeutigen Wert
- **Dynamisches Umsortieren muss gesondert berücksichtigt werden**
  - Umsortieren ändert die Caret Category
- **Wunschliste für zukünftige Releases**
  - Programmatische Identifizierung des View Selection Context (nicht Caret Category)
  - Notwendige Referenzdaten für Dokumente sollten automatisch verfügbar sein
    (Servername, Pfad oder Replica ID zusätzlich zur UNID)
  - Antworthierarchien sollten unterstützt werden
    - Können aber aktuell auch programmatisch per SQL Query erzeugt werden



- **Die Nutzung von aggregierten Ansichten setzt standardisierte Infrastruktur voraus**
- **Anforderungen an Applikationen**
  - Datenstruktur Policies müssen implementiert und befolgt werden
  - Applikationsdesign muss auf wenige Templates beschränkt sein
- **DB2 als alternatives Data Store muss sorgfältig evaluiert werden**



Query Views - Performance

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- **Performance ist geringer als NSF, wenn nicht spezielle Features genutzt werden**
- **Beta Code – Performance Aussagen können nicht auf Release bezogen werden**
- **DB2 Result Sets**
  - Live SQL Query beim Öffnen oder Aktualisieren
  - Kein View Index muss gespeichert oder aktualisiert werden
  - Vorteile entstehen bei kleinen Result Sets in großen Dokumentenmengen
  - Vorteile entstehen bei vielen Updates
- **In bestimmten Szenarien**
  - Könnten Query Views Performance Probleme adressieren, wenn ein Redesign der Applikation vorgenommen wird
  - Aber: Redesign evtl. problematisch, da Query Views nicht lokal verfügbar sind

Query Views Anmerkungen #1

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- DB2 User Mapping ist notwendig um Sicherheitsmechanismen zu verwenden
- Default für die maximale Anzahl an Zeilen im Result Set ist 500 (Server Level)
- Föderierte Daten müssen in DAV oder DB2 Tabelle verfügbar sein
- Föderierte Daten sind lokal nicht verfügbar
- Query Views werden nicht automatisch aktualisiert
  - Query wird nicht automatisch neu ausgeführt
  - SHIFT-F9 führt zur Neuausführung der Query

Query Views Anmerkungen #2

- Überprüfung des Schemanamens ist wichtig
- @DB2Schema sollte immer verwendet werden um den Schemanamen zu bestimmen (verfügbar seit Beta 3)
- N/D Spaltensortierung hat Präzedenz vor der SQL Sortierung

Query Views Zusammenfassung

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- JOINS lösen grundsätzliche Probleme von N/D mit Relationen
- Aggregierte Ansichten ermöglichen die Schaffung eines "Single Point of Access" zu allen Dokumenten die ein User benötigt
- Potenzial Applikationen zu entwerfen, welche die Arbeitseffizienz erhöhen
- Potenzial um aktuelle N/D Problemszenarien in Bezug auf Performance/Scalability zu adressieren

Videoclip

Agenda

- 1 Einleitung
  - N/D ⇔ DB2 Paradigmen
  - Einführung Domino Access und Query Views
- 2 Domino Access Views (DAV)
  - Szenarien
  - Technologie, Tips
  - Live Demo
- 3 Query Views
  - Szenarien
  - Technologie, Tips
  - Föderierte Daten
  - Live Demos
- 4 Zusammenfassung

Zusammenfassung

University of Paderborn
Business Computing 2
Prof. Dr. Ludwig Nastansky

- Es sollten Einsatzszenarien identifiziert und diskutiert werden
- Die Applikationsinfrastruktur sollte konsolidiert werden (gilt ohnehin immer)
- DB2 Administrationsskills sollten rechtzeitig erworben werden
- Gleiches gilt für SQL Programmierung
- Applikationen sollten für die Nutzung von DAV and Query Views vorbereitet werden

Sonstiges

- Fragen und Diskussionen sind herzlich Willkommen
- Bitte die Evaluationsbögen ausfüllen!

- Sample Code und aktualisierte Folien:
  http://gcc.upb.de/k-pool/Admin2005_S14
- Ingo Erdmann
  mailto:Ingo.Erdmann@notes.upb.de
  http://gcc.upb.de/IngoErdmann
- Universität Paderborn
  Groupware Competence Center
  http://gcc.upb.de

Credits

- **Dank an das GCC N/D7 beta Test Team**
- **Besonderen Dank an**
  - Prof. Dr. Ludwig Nastansky
  - Holger Ploch, Olaf Hahnl
  - Murat Yanik