AnyApp macht's möglich:

Die Entwicklung einer kompletten SOAP-Anwendung

CICS-Backend / Java–Frontend

In nur 2 Stunden!

Unser Werkzeug:

AnyApp – CICS-Kernel

+

AnyApp – Java – Standard-Frontend

Prolog

Das ursprüngliche Vorhaben, Art und Funktion der Applikation erst direkt in der Sitzung von den Anwesenden festlegen zu lassen, musste aufgegeben werden. Eine Testsession zeigte, dass alleine die Diskussion um die zu erstellende Applikation sowie die Definition der Grunddaten und der Topologie die gesamte veranschlagte Präsentationszeit verbrauchte.

Es wurde daher eine allgemeine Applikation gesucht, die folgenden Kriterien genügen sollte:

- Sie sollte jedermann vertraut sein,
- möglichst zu keiner der Branchen der anwesenden Professionisten einen besonderen Bezug haben,
- von überschaubarer Komplexität sein,
- dennoch einiges an Businesslogik erfordern,
- Funktionen benötigen, die nur durch die Spezifikation einer Non-Standard-Oberfläche möglich sind.

Eine zu diesem Profil passende Abwicklung ist zum Beispiel die ***Verwaltung einer öffentlichen Bücherei***.

Das Prozedere beim Entlehnen von Büchern (und anderen Artikeln wie die Diskussion ergab) kann als allgemein bekannt vorausgesetzt werden. Die Herstellung der IT-Unterstützung von Kundenbestand, Ausleihung, Mahnung, Bestellung und Lagerhaltung ist ausreichend anspruchsvoll und die Realisierung dieser Anforderung unter einer CICS/DB2-Backend – JAVA-Frontend Anwendung unter SOA binnen nur zwei Stunden ist selbst für **AnyApp** eine absolute Herausforderung!

Wie gehen daher davon aus, dass uns von einer Organisations-abteilung zumindest das Datenmodell, und für dieses von der DBA ein fertiges SQL-Member mit den nötigen DBD-Gen-Statements vorliegt. Die genaue Topologie der Applikation, das Funktions-schema, die Datenbanken und das Detailregelwerk werden wir jedoch, soweit es die Zeit erlaubt, in dieser Sitzung herstellen.

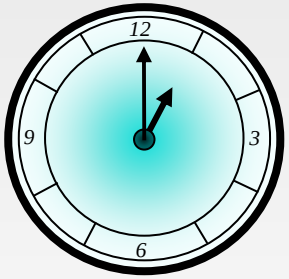
HOST-Voraussetzungen:

- Ein DB-Space wurde eingerichtet (hier SPADB01)
- die allgemeinen Steuertabellen sind in diesem mit eigenem Creator aufgebaut (hier SPA)
- Die CICS-Komponenten von **AnyApp** sind installiert, die Zentralroutinen mit obigen Tabellen (Creator SPA) statisch im DB2 gebunden
- für diese Demo ist ein eigener Creator mit ausreichenden Rechten verfügbar (hier SPA00).

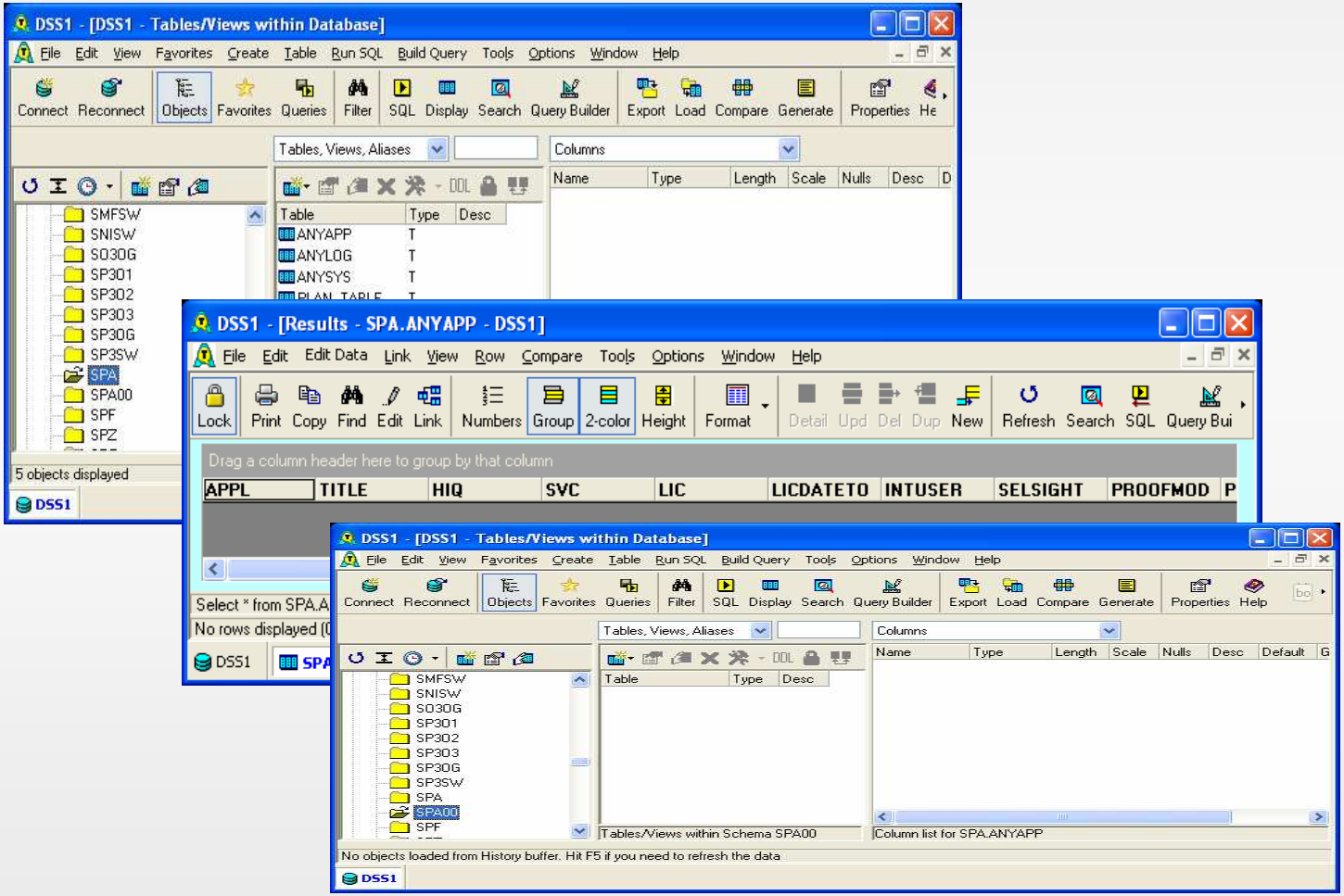
Anmerkungen zum optional erhältlichen integrierten **AnyApp** -Framework unter PHP/mySQL:

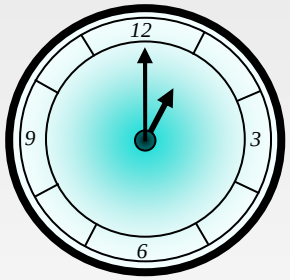
Da die Datenbankbeschreibung für DB2 fertig vorliegt, entfällt das sonst in **AnyApp** empfohlene Prototyping unter PHP/mySQL. Änderungen unter mySQL können im Gegensatz zu DB2 sehr einfach und direkt in einer auch bereits vitalen Datenbank vorgenommen werden, ohne diese jedesmal ins Zielformat entladen, und neu aufbauen zu müssen.

Die integrierte **AnyApp**-Version unter PHP/mySQL ist systembedingt toleranter im Umgang mit Datenformaten und Konvertierungen, was das Designen und den Detailtest bei Neuerstellung oder Änderung deutlich vereinfacht und beschleunigt. Die integerierte **AnyApp**-Version unter PHP/mySQL läuft stand-alone auf jedem PC, bedarf keiner der oben angegebenen Voraussetzungen und kann die erstellte Applikation auch im DB2-Format exportieren.



1. Datenbasis herstellen
   Status: leere ANY-Systemtabellen, keine Applikationsdaten





1.a) Primärload der Steuertabellen (liegen der Distribution bei)

demogen_SPA00.sql - Editor

Datei Bearbeiten Format Ansicht ?

```
set current sqlid = 'SPA00';

drop tablespace spadb
Create Tablespace dem
        USING STOGROU
        BUFFERPOOL BP


-- DOMACCESS

DROP TABLE DOMACCESS;
CREATE TABLE domacces
  tbl varchar(26) NOT
  rolle varchar(64) N
  zugriff char(3) NOT
  bem varchar(100)
) in spadb01.demospc
;

create unique index d
alter table domaccess
GRANT DELETE,INSERT,S


-- DOMFIELD

set current sqlid = '

Drop Table DOMFIELD;
```

DSS1 - Run Multiple SQL Statements

Close Edit View Help

Auto Commit: On / Off | Stop On Error?: Yes / Ignore errors on Drops / No / More | Select: All / None | Start from Stmt: 1 | Check Syntax: Yes | Parameters: Always Prompt

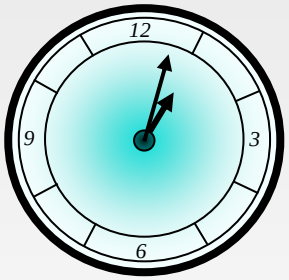
	Run?	Line No	Statement	Elapsed	Result
602	Completed	805	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,08	SQL executed OK. Number of rows affected=1
603	Completed	806	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,08	SQL executed OK. Number of rows affected=1
604	Completed	807	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,10	SQL executed OK. Number of rows affected=1
605	Completed	808	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,07	SQL executed OK. Number of rows affected=1
606	Completed	809	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,06	SQL executed OK. Number of rows affected=1
607	Completed	810	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,08	SQL executed OK. Number of rows affected=1
608	Completed	811	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,08	SQL executed OK. Number of rows affected=1
609	Completed	812	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('error', ..	0,08	SQL executed OK. Number of rows affected=1
610	Completed	813	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('usergr..	0,08	SQL executed OK. Number of rows affected=1
611	Completed	814	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('usergr..	0,06	SQL executed OK. Number of rows affected=1
..	Completed	815	Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('usergr..	0,07	SQL executed OK. Number of rows affected=1

Insert into DOMSEL (ITEM,AKTIV,INTERN,EXTERN) values('usergrp', 1, 'guest', 'Gast')

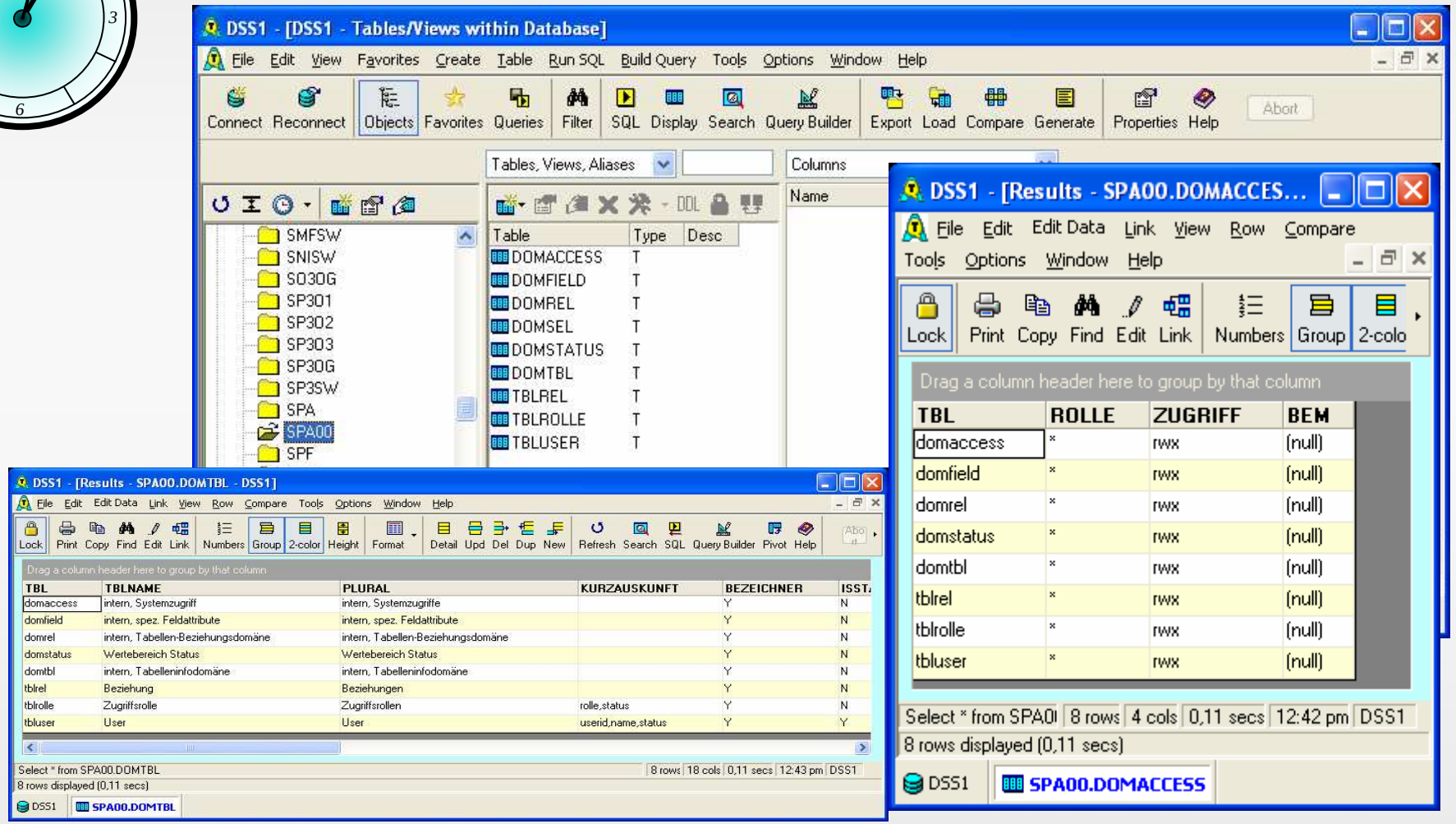
SQL executed OK. Number of rows affected=1

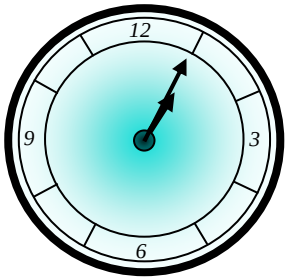
Run | Exit | Abort | Pause

Completed processing 612 statements (9 errors)



Kontrolle: Die Steuertabellen sind in DB2/PAR00 korrekt angelegt



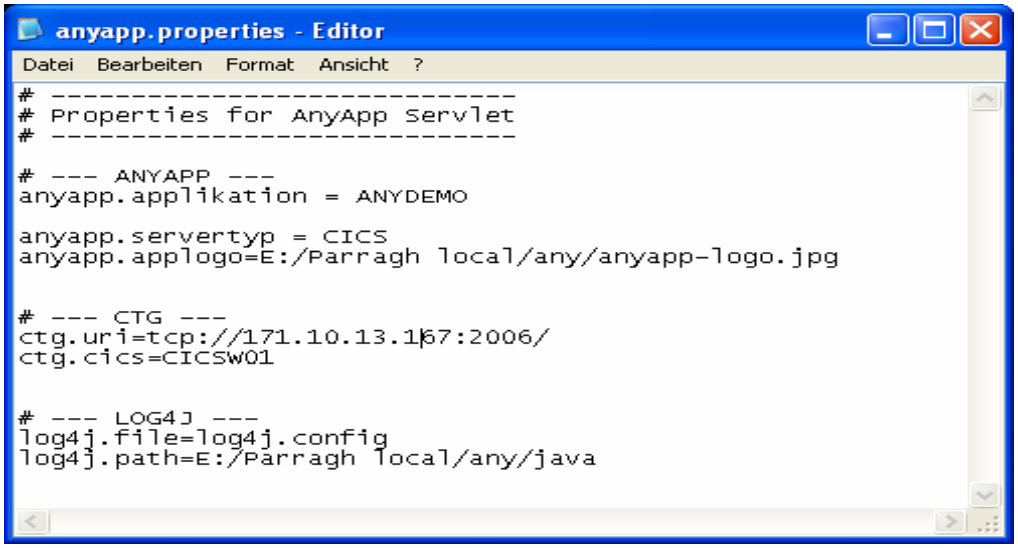


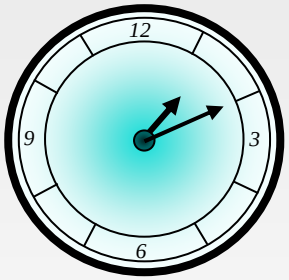
1.b) Die aktuelle Demoapplikation ist in die Konfiguration des/r unterstützenden WEB-Server(s) einzutragen.

WAS →

TOMCAT →

```
<?xml version="1.0" encoding="UTF-8" ?>
- <web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name>AnyApp</display-name>
- <servlet>
  <description />
  <display-name>AnyAppServlet</display-name>
  <servlet-name>AnyAppServlet</servlet-name>
  <servlet-class>com.compitence.anyapp.AnyAppServlet</servlet-class>
- <init-param>
   <description />
   <param-name>propertyfile1</param-name>
   <param-value>D:\develop\RAD75\AnyAppConfig\anyapp.properties</param-value>
  </init-param>
- <init-param>
   <description />
   <param-name>propertyfile</param-name>
   <param-value>E:\Parragh local\any\java\anyapp.properties</param-value>
  </init-param>
  </servlet>
- <servlet-mapping>
  <servlet-name>AnyAppServlet</servlet-name>
  <url-pattern>/AnyAppServlet</url-pattern>
  </servlet-mapping>
- <welcome-file-list>
  <welcome-file>AnyAppServlet</welcome-file>
  </welcome-file-list>
  </web-app>
```



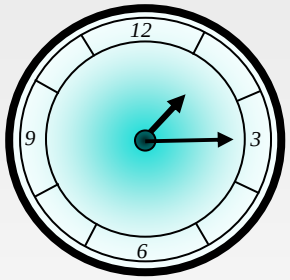


1.c) Die Datenfeldtabelle ist zu initialisieren. Dies kann bei einer neuen Applikation nur über einen Batchjob (ANYBFLD, Teil der Distribution) erfolgen, da ohne Datentabelle kein Einstieg in die Administrationssicht, und damit kein Aufruf des Online-Refresh möglich ist.

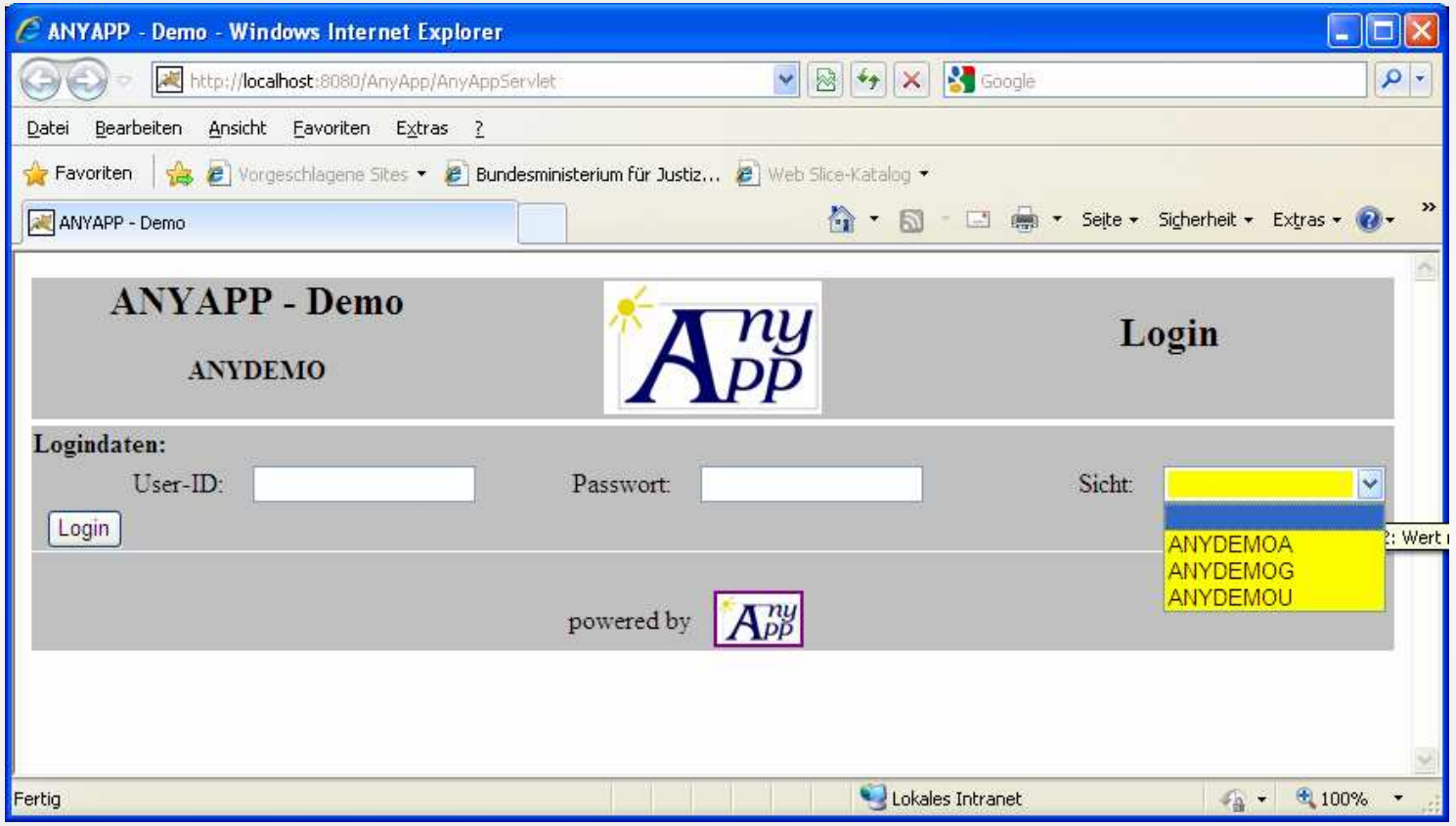
Creator

```
****** ***************************** Top of Data ***********************
000001 //ANYAPP  JOB (SPA,SPA),PARRAGH,CLASS=A,MSGCLASS=Z,NOTIFY=SPA
000002 //*
000003 //ANYTEST  EXEC PGM=IKJEFT01,REGION=0M
000004 //SYSTSIN  DD *
000005  DSN SYSTEM(DSS1)
000006  RUN PROGRAM(ANYBFLD) PLAN(BATCHX) -
000007  PARM('/SPA00')
000008 /*
000009 //STEPLIB  DD DSN=XXX.LOADLIB,DISP=SHR
000010 //         D
000011 //         D
000012 //SYSPRINT D
000013 //SYSTSPRT D
000014 //SYSDMP   D
000015 //PLIDUMP  D
000016 //SYSUDMP  D
000017 //SYSUDUMP D
****** ************
```

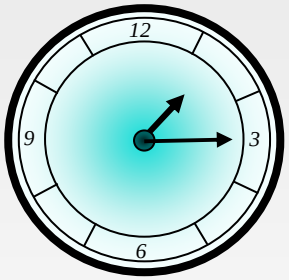
```
******************************** TOP OF DATA **************
Programm ANYBFLD, Refresh Feldtabelle
Start Programm ANYBFLD, 20110531130749405
Duchführung für Creator SPA00

TBLREL                                                  11
TBLROLLE                                                 9
TBLUSER                                                 19
Anzahl Tabellen                      3
Anzahl Felder                       39
Anzahl Delete 1                      0
Anzahl Delete 2                      0
Anzahl Inserts                      39
Anzahl Updates                       0
Ende Programm ANYBFLD,  20110531130749878
******************************** BOTTOM OF DATA ************
```



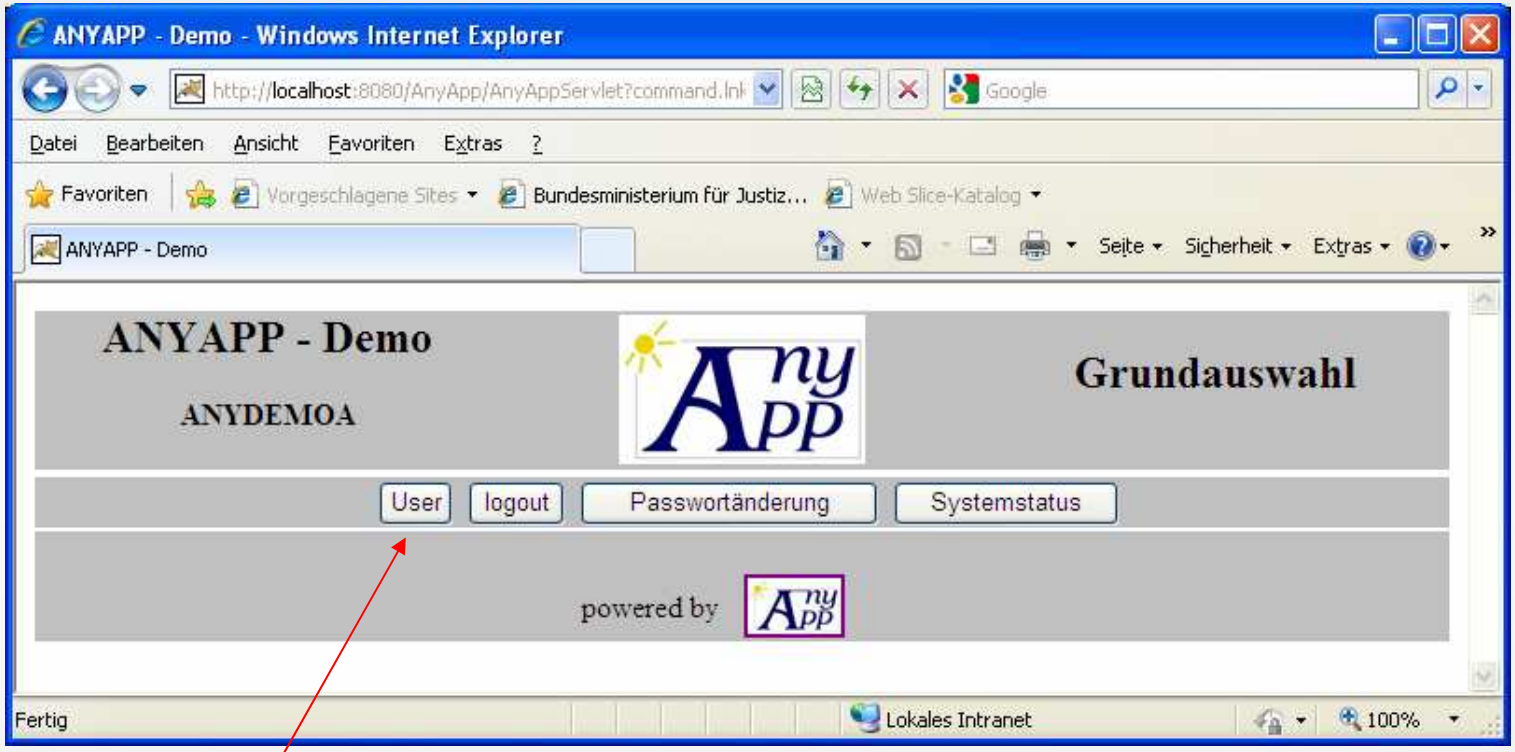
Nun können wir erstmals in die neue Applikation einsteigen:



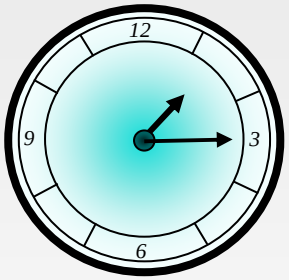
Ist die integrierte Userverwaltung aktiviert, muss man sich ins System einloggen. Läuft das System hinter einem Portal, das User-ID und User-Rollen selbst administriert und diese weitergeben kann, entfällt das Login (Commands werden ausgeblendet). Ein Portaleinstieg bestimmt auch die Applikationssicht. Eine Anwahl, wie hier, ist dann nicht vorgesehen. Default User: *anyapp*, Default PW: *anyapp*, wir wählen Sicht ANYDEMOA (Administrationssicht).



Der Standardeinstieg liefert immer die Grundauswahl:



Wären bereits Applikationstabellen eingebunden, suchbar und für den Anwender autorisiert, so würden sie hier, so wie die Usertabelle (User), angezeigt.



Das SOA-Protokoll am Beispiel des Login:

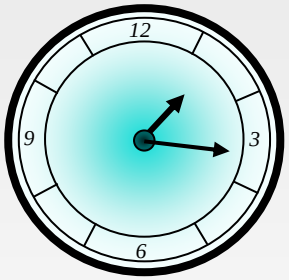
Das Bespiel zeigt gut die Arbeitsweise des Frameworks. Das Java-Frontend selbst hat keine Businesslogik. Es präsentiert nur den Datenstrom vom Server. Der CICS-Kernel bestimmt zu jedem Zeitpunkt vollständig das Daten- und Prozessverhalten: So wird <command> immer zu einem Button, <list> zu einer Auswahlliste, <fields> zu einer <field>Liste, etc. Ist bei einem Key-Tag ein Link angegeben, so wird beim Anklicken der zugehörige Command mit diesem Link wieder zum Server geschickt. Links identifizieren immer exakt einen Knoten im Daten-Cube. Mit dieser Konstruktion lässt sich jedes Prozess- und Strukturschema einfach und sicher abbilden.

Request

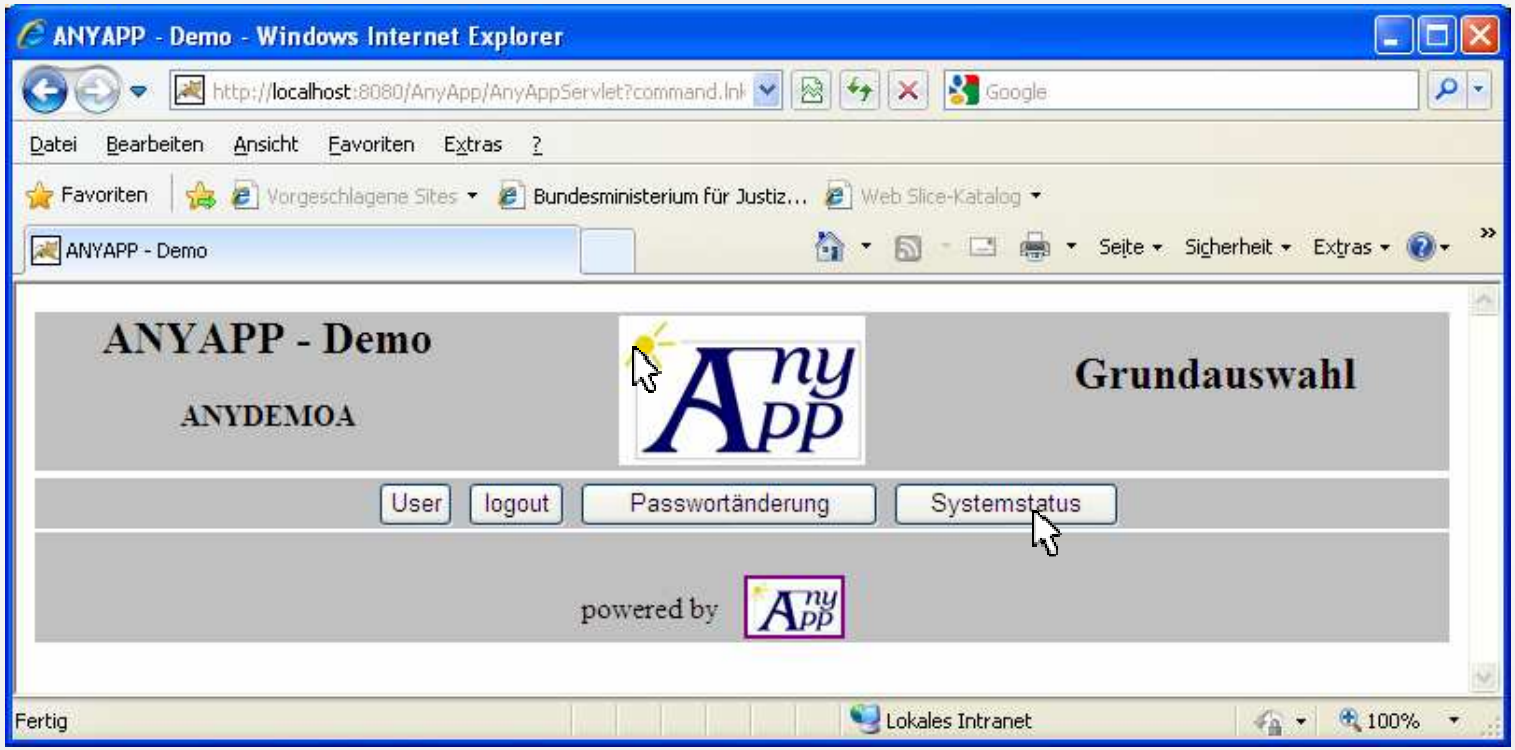
```
<?xml version="1.0" encoding="UTF-8" ?>
- <request>
- <hdr>
  <userid>anyapp</userid>
  <token>2de4968c96e1b22b28ce0d68761ebb34</token>
  <sicht>ANYDEMOA</sicht>
  <applikation>ANYDEMO</applikation>
  </hdr>
- <data>
  <command lnk="SYSTEM:0">login</command>
- <fields>
  <field name="password">2de4968c96e1b22b28ce0d68761ebb34</field>
  <field name="userid">anyapp</field>
  <field name="sicht">ANYDEMOA</field>
  </fields>
  </data>
  </request>
```

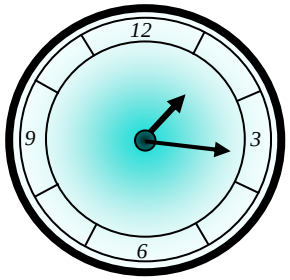
Response

```
- <anyapp timestamp="20110531131617245" dok="response">
- <hdr>
  <userid>ANYAPP</userid>
- <rollen>
  <rolle>*</rolle>
  </rollen>
- <return>
  <retcode>0</retcode>
  <messages />
  </return>
  <applikation title="ANYAPP - Demo">ANYDEMO</applikation>
  <sicht>ANYDEMOA</sicht>
  <token>2de4968c96e1b22b28ce0d68761ebb34</token>
  <service name="Grundauswahl" />
  </hdr>
- <data>
- <commands>
  <command name="User" lnk="TBLUSER:0">get-searchkrits</command>
  <command name="logout" lnk="ANYAPP">logout</command>
  <command name="Passwortaenderung" lnk="ANYAPP">get-pwaend</command>
  <command name="Systemstatus" lnk="SYSTEM:0">adm-status</command>
  </commands>
  </data>
  </anyapp>
```



Wir aktivieren den Menüpunkt Systemstatus:





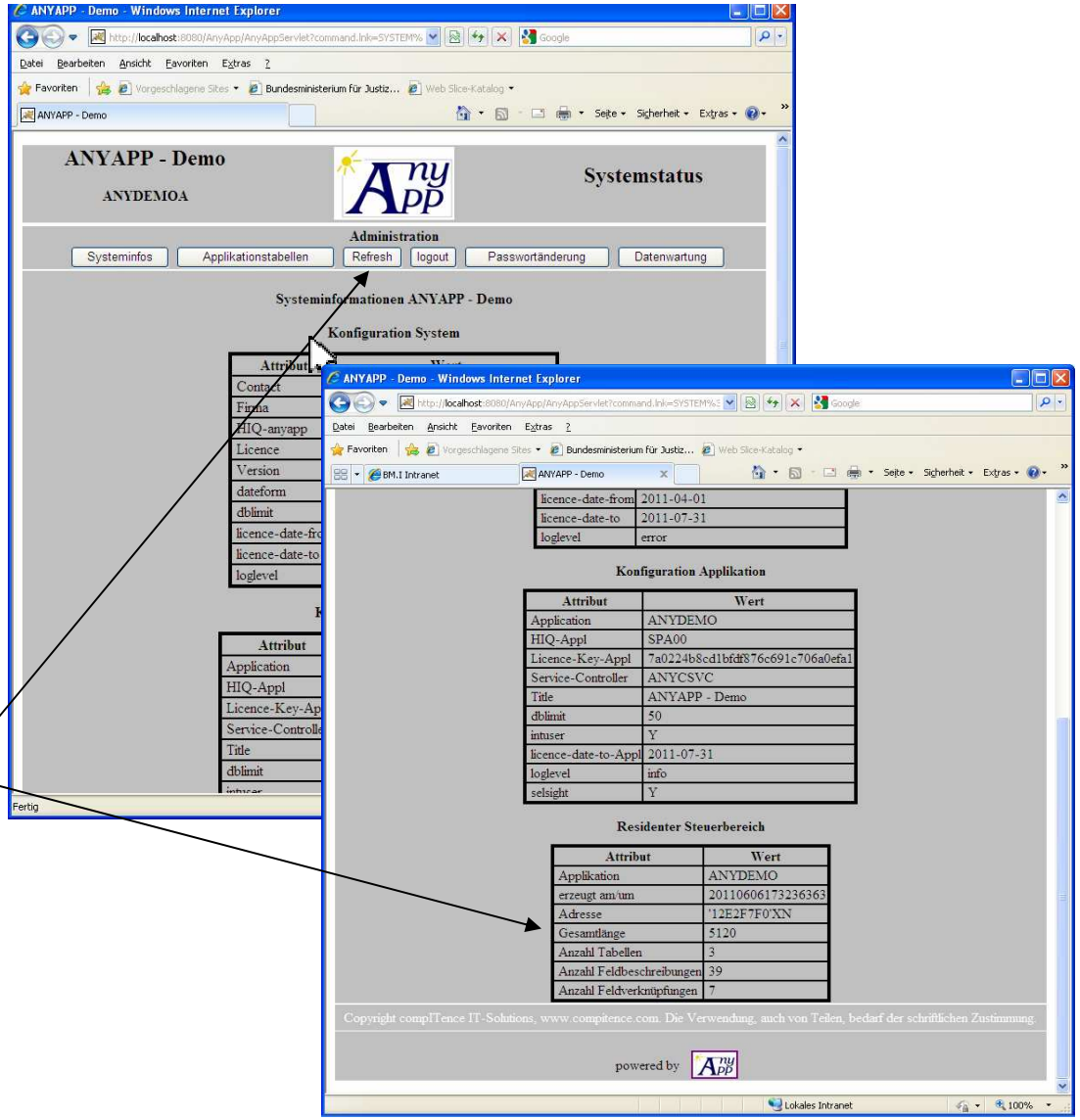
AnyApp unter CICS ist hochperformant. Die Steuertabellen werden im Echtbetrieb pro Applikation resident geladen und bedingen daher keine DB-Zugriffe zur Laufzeit.

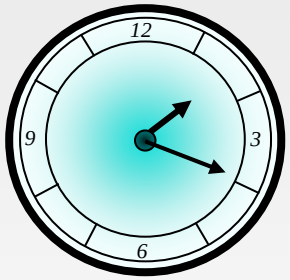
Dies ist möglich, da die Eintragungen in die Steuertabellen das Applikationsverhalten bestimmen, ähnlich einem Programmeinbau in der klassischen Programmierung. Sie ändern sich daher nicht während des Betriebs.

Üblicherweise wird für AnyApp in der CWA ein dedizierter Basispointer vergeben. Der erste Systemaufruf generiert dann über diesen den residenten und komprimierten Steuerbaum.

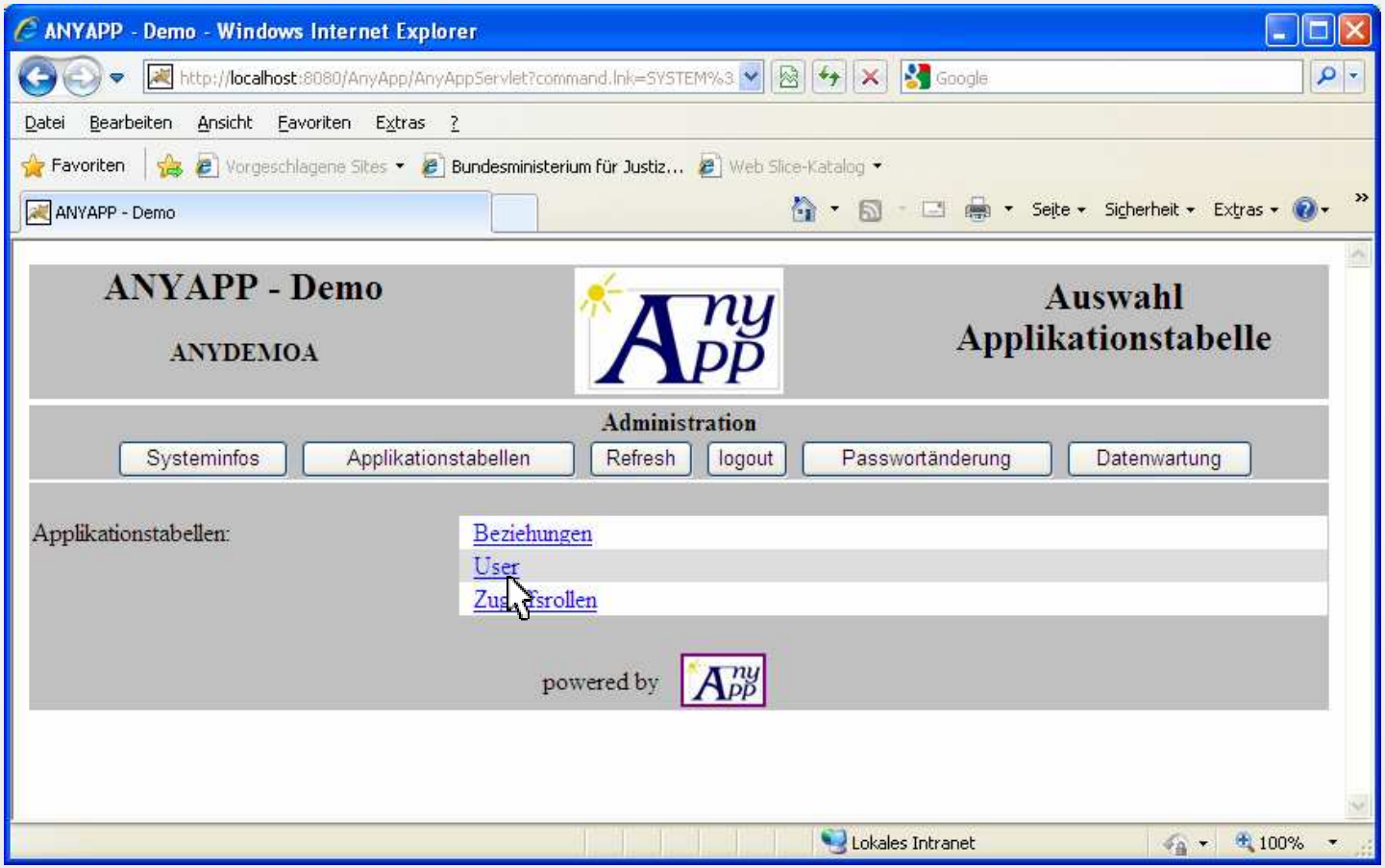
Der Platzbedarf ist überschaubar: Das Basissetup benötigt dafür 5120 Bytes (siehe Tabelle rechts), der IFA-Prototyp mit 23 Tabellen und etwa 500 Datenfeldern ca. 50KB.

Sind laufende Änderungen nötig, so kann mittels des ‚Refresh'-Buttons (rechts) ein Neugenerieren/Aktualisieren der Datenfeld-tabelle nebst Reload des Steuerbaums erzwungen werden.

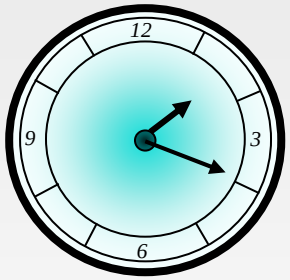




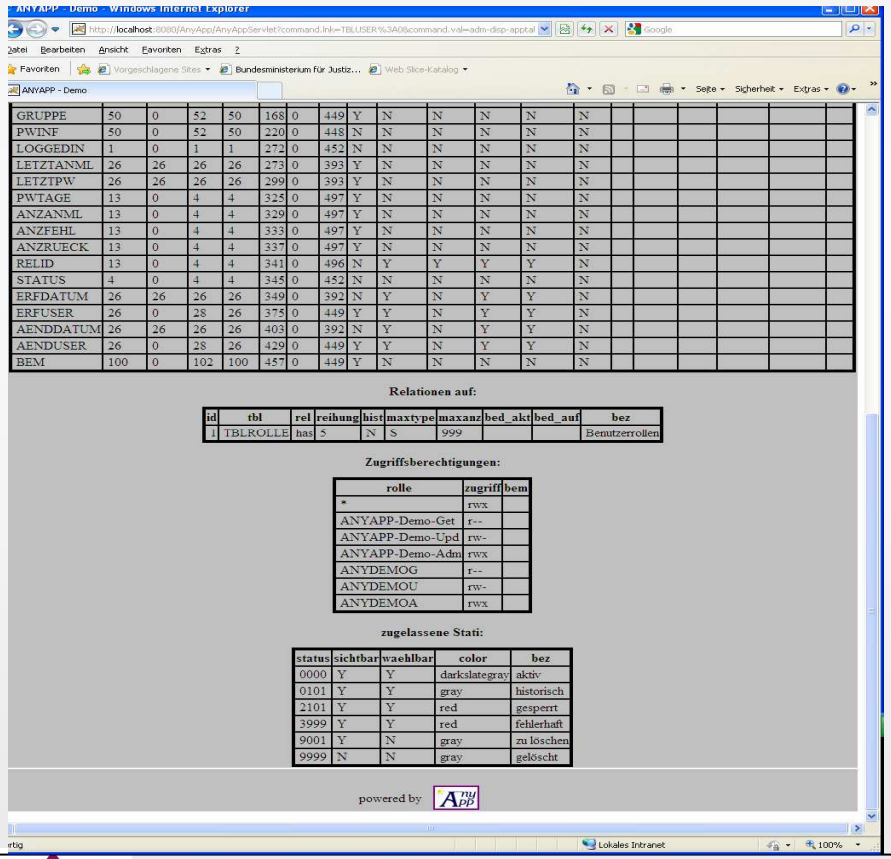
Klick auf ‚Applikationstabellen' bringt die Liste aller eingetragenen Tabellen



AnyApp – Universelles Applikationsframework - Leistungsschau



Bei Anwahl von z.B. Tabelle ‚User‘ kommen deren Details...



GRUPPE 50 0 52 50 168 0 449 Y N N N N N
PWINF 50 0 52 50 220 0 448 N N N N N N
LOGGEDIN 1 0 1 1 272 0 452 N N N N N N
LETZTANML 26 26 26 26 273 0 393 Y N N N N N
LETZTPW 26 26 26 26 299 0 393 Y N N N N N
PWTAGE 13 0 4 4 325 0 497 Y N N N N N
ANZANML 13 0 4 4 329 0 497 Y N N N N N
ANZFEHL 13 0 4 4 333 0 497 Y N N N N N
ANZRUECK 13 0 4 4 337 0 497 Y N N N N N
RELID 13 0 4 4 341 0 496 N Y Y Y Y N
STATUS 4 0 4 4 345 0 452 N N N N N N
ERFDATUM 26 26 26 26 349 0 392 N Y N Y Y N
ERFUSER 26 0 28 26 375 0 449 Y Y N Y Y N
AENDDATUM 26 26 26 26 403 0 392 N Y N Y Y N
AENDUSER 26 0 28 26 429 0 449 Y Y N Y Y N
BEM 100 0 102 100 457 0 449 Y N N N N N

Relationen auf:

id	tbl	rel	reihung	hist	maxtype	maxanz	bed_akt	bed_auf	bez
1	TBLROLLE	has	5	N	S	999			Benutzerrollen

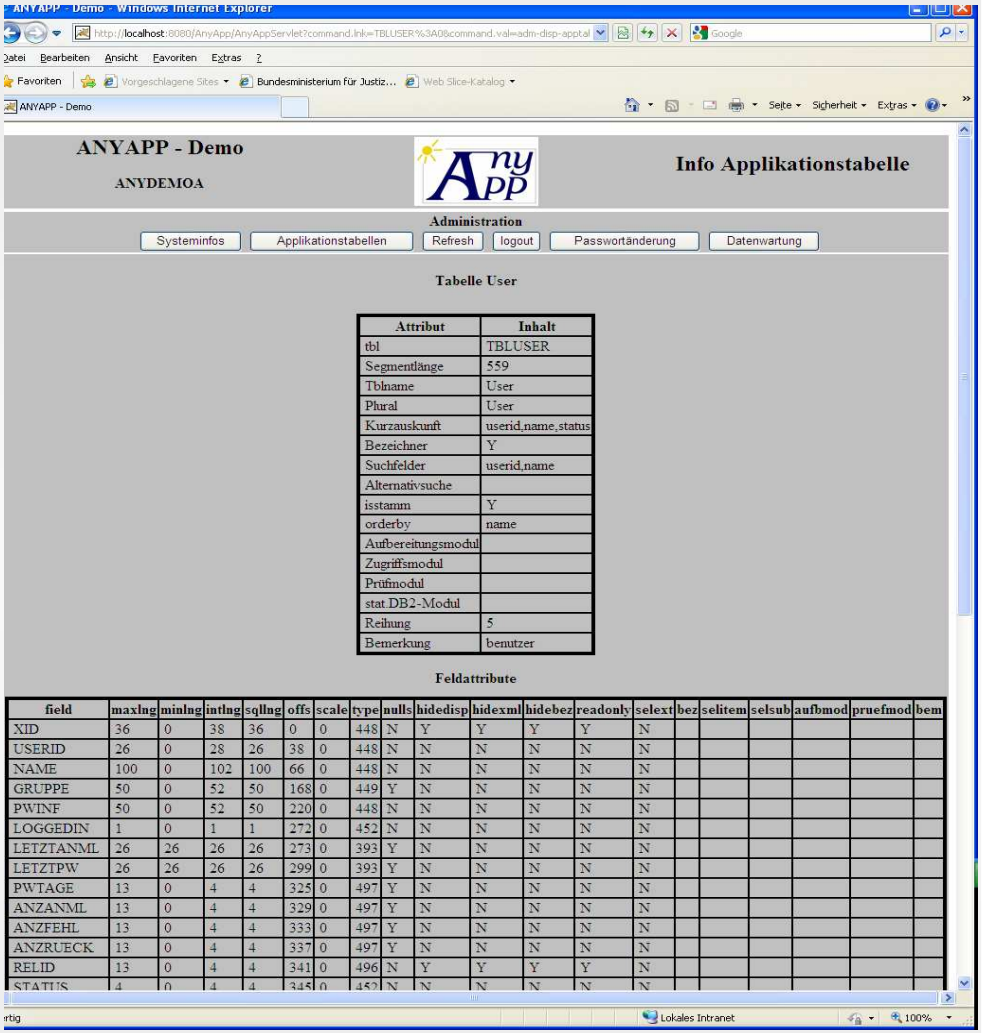
Zugriffsberechtigungen:

rolle	zugriff	bem
*	rwx	
ANYAPP-Demo-Get	r--	
ANYAPP-Demo-Upd	rw-	
ANYAPP-Demo-Adm	rwx	
ANYDEMOG	r--	
ANYDEMOU	rw-	
ANYDEMOA	rwx	

zugelassene Stati:

status	sichtbar	waehlbar	color	bez
0000	Y	Y	darkslategray	aktiv
0101	Y	Y	gray	historisch
2101	Y	Y	red	gesperrt
3999	Y	Y	red	fehlerhaft
9001	Y	N	gray	zu löschen
9999	N	N	gray	gelöscht

powered by



ANYAPP - Demo

ANYDEMOA

Info Applikationstabelle

Administration

Systeminfos | Applikationstabellen | Refresh | logout | Passwortänderung | Datenwartung

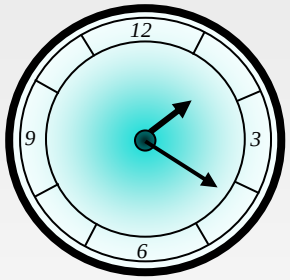
Tabelle User

Attribut	Inhalt
tbl	TBLUSER
Segmentlänge	559
Tblname	User
Plural	User
Kurzauskunft	userid,name,status
Bezeichner	Y
Suchfelder	userid,name
Alternativsuche	
isstamm	Y
orderby	name
Aufbereitungsmodul	
Zugriffsmodul	
Prüfmodul	
stat DB2-Modul	
Reihung	5
Bemerkung	benutzer

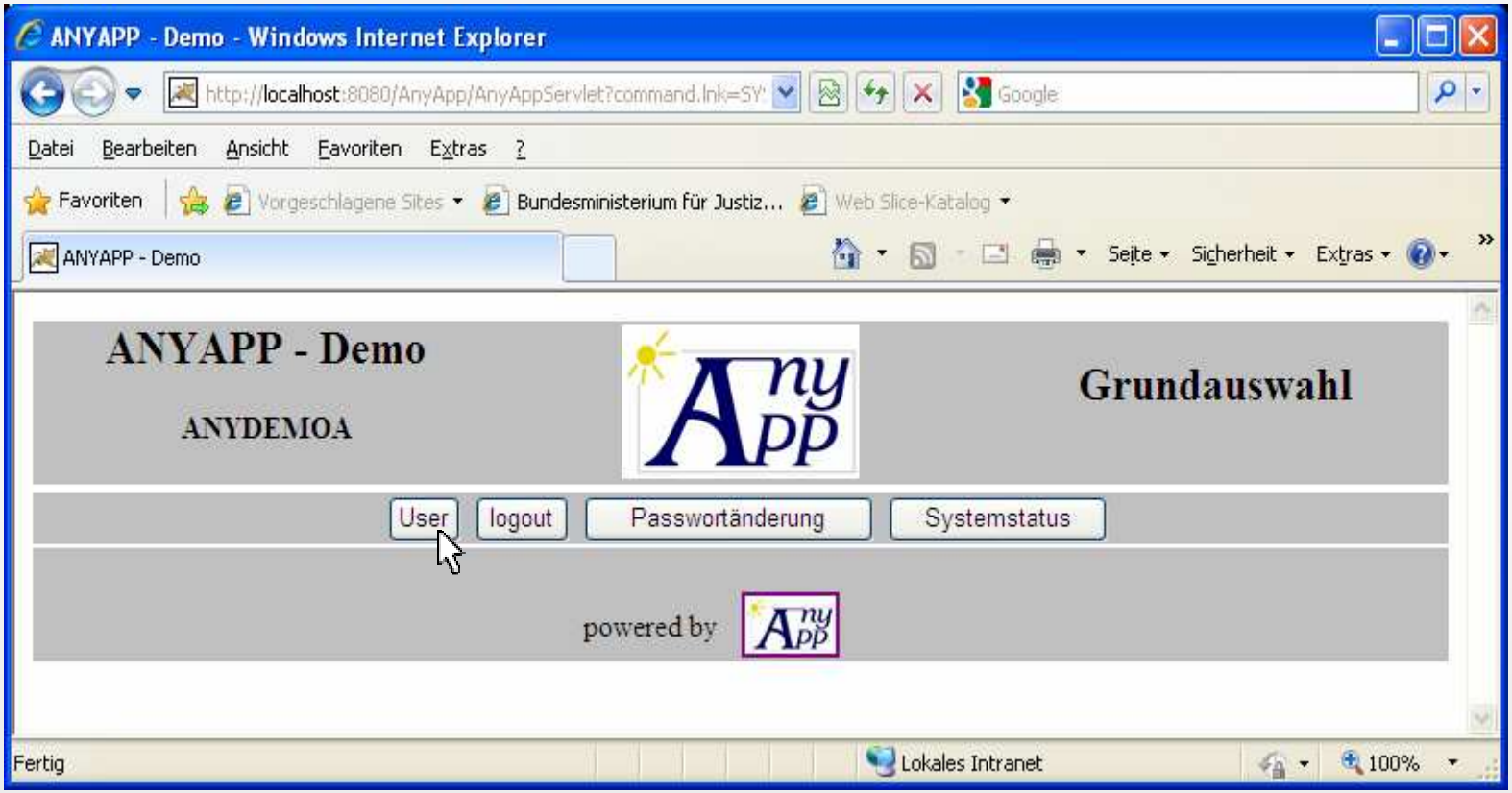
Feldattribute

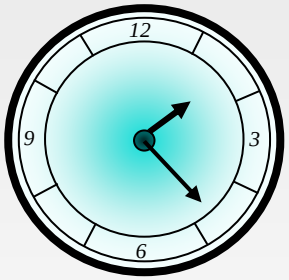
field	maxlng	minlng	intlng	sqllng	offs	scale	type	nulls	hidedisp	hidexml	hidebez	readonly	selext	bez	selitem	selsub	aufbmod	pruefmod	bem
XID	36	0	38	36	0	0	448	N	Y	Y	Y	Y	N						
USERID	26	0	28	26	38	0	448	N	N	N	N	N	N						
NAME	100	0	102	100	66	0	448	N	N	N	N	N	N						
GRUPPE	50	0	52	50	168	0	449	Y	N	N	N	N	N						
PWINF	50	0	52	50	220	0	448	N	N	N	N	N	N						
LOGGEDIN	1	0	1	1	272	0	452	N	N	N	N	N	N						
LETZTANML	26	26	26	26	273	0	393	Y	N	N	N	N	N						
LETZTPW	26	26	26	26	299	0	393	Y	N	N	N	N	N						
PWTAGE	13	0	4	4	325	0	497	Y	N	N	N	N	N						
ANZANML	13	0	4	4	329	0	497	Y	N	N	N	N	N						
ANZFEHL	13	0	4	4	333	0	497	Y	N	N	N	N	N						
ANZRUECK	13	0	4	4	337	0	497	Y	N	N	N	N	N						
RELID	13	0	4	4	341	0	496	N	Y	Y	Y	Y	N						
STATUS	4	0	4	4	345	0	452	N	N	N	N	N	N						

wie globale Tabelleneinstellungen, Datenfelder, Feldattribute, Beziehungen, Zugriffe, Stati, etc.



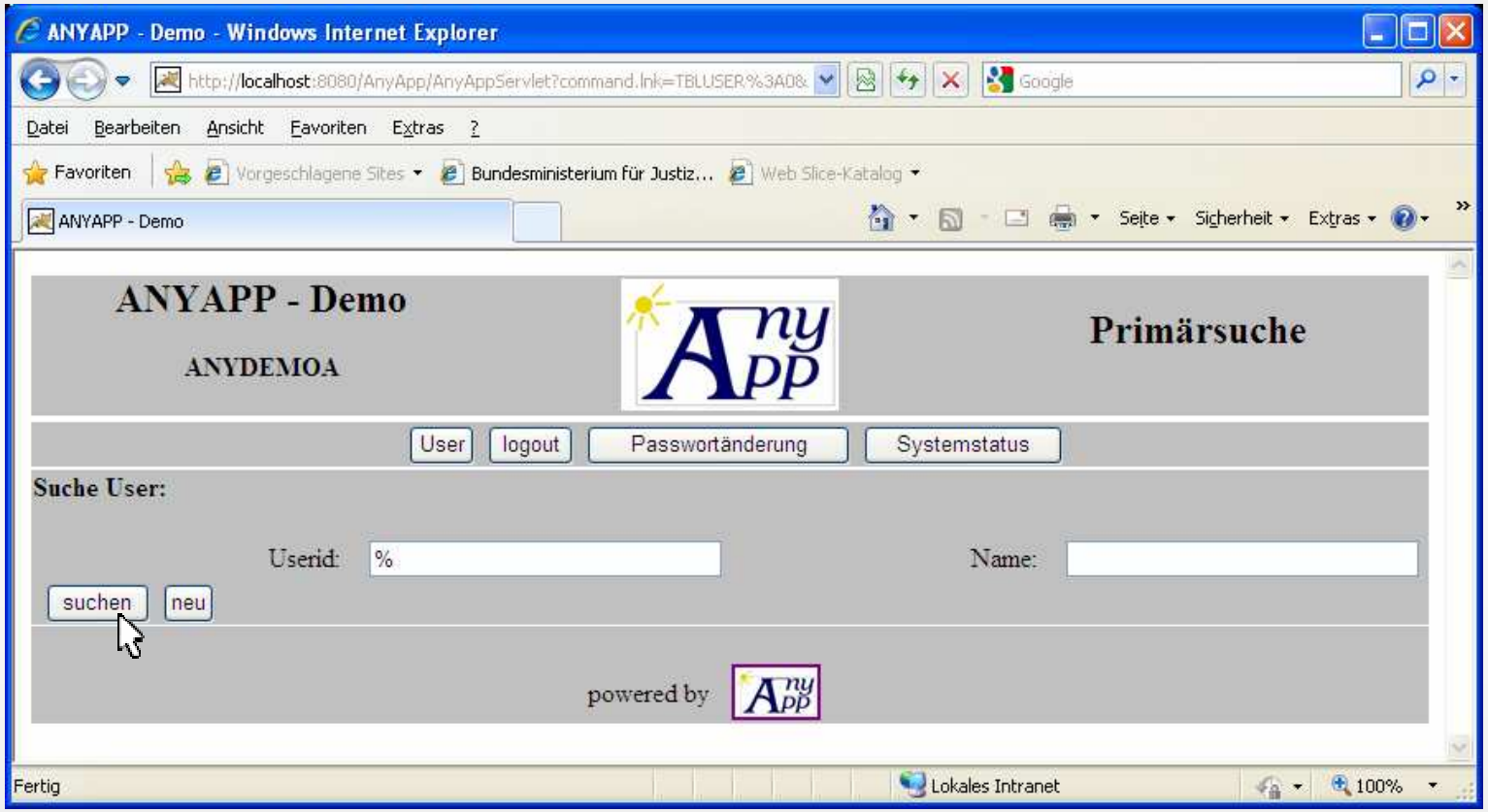
Zurück zur Grundauswahl und ‚User'-Button drücken ...



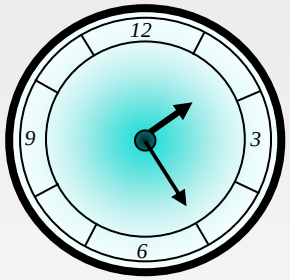


... liefert die erste echte Frameworkfunktion: Suche .

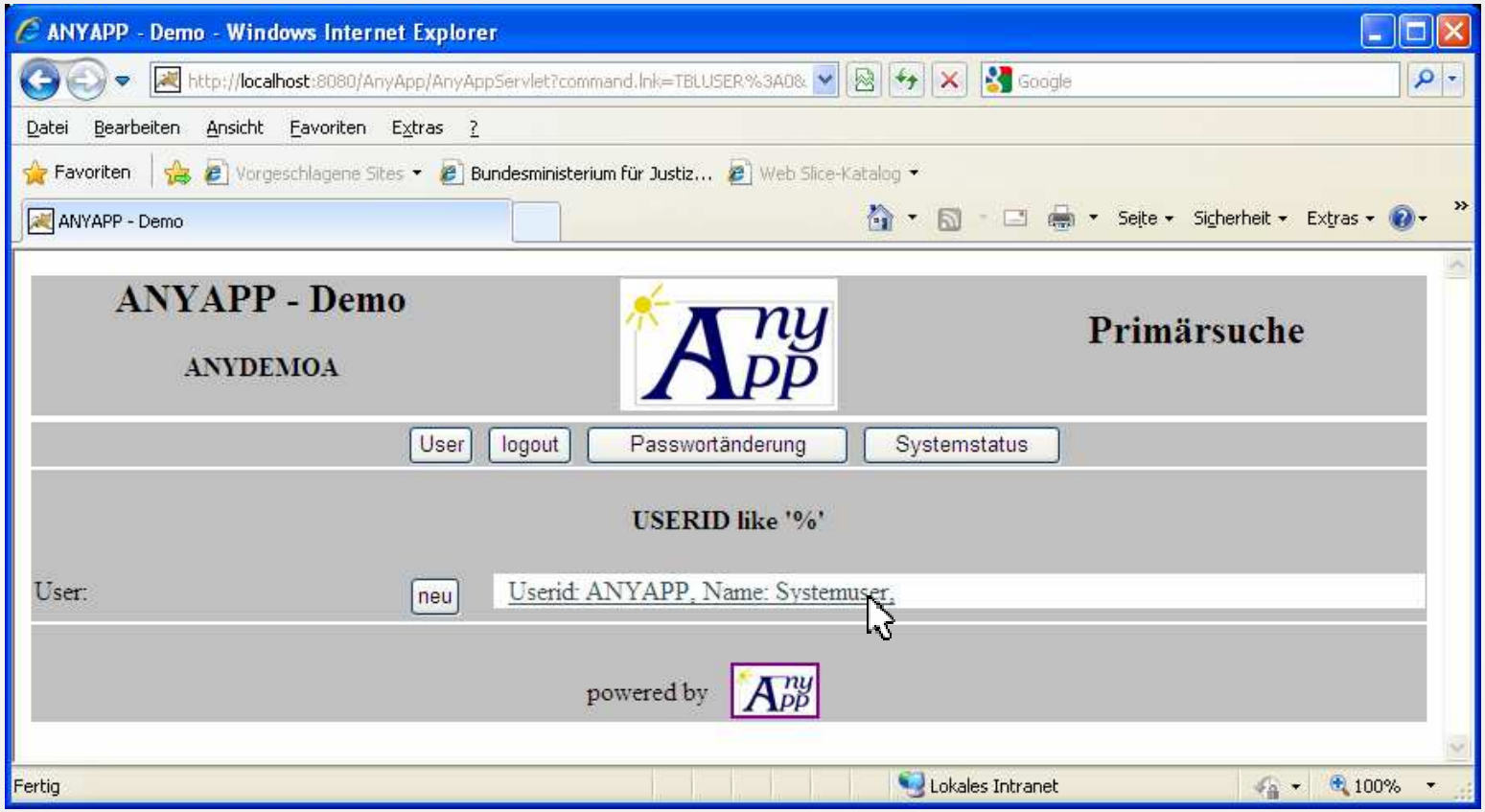
Suchfelder lt. Eintragung in domtbl



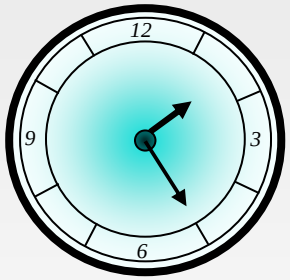
Werden bei einer Tabelle Suchfelder angegeben, so erscheint sie als Button im Commandblock und kann nach diesen Werten durchsucht werden. Als Wildcards werden die SQL-Standardwildcards ‚%' und ‚_' akzeptiert. Eine Suche nach ‚%' liefert somit alle Ergebnisse (Default max. 100, ist einstellbar).



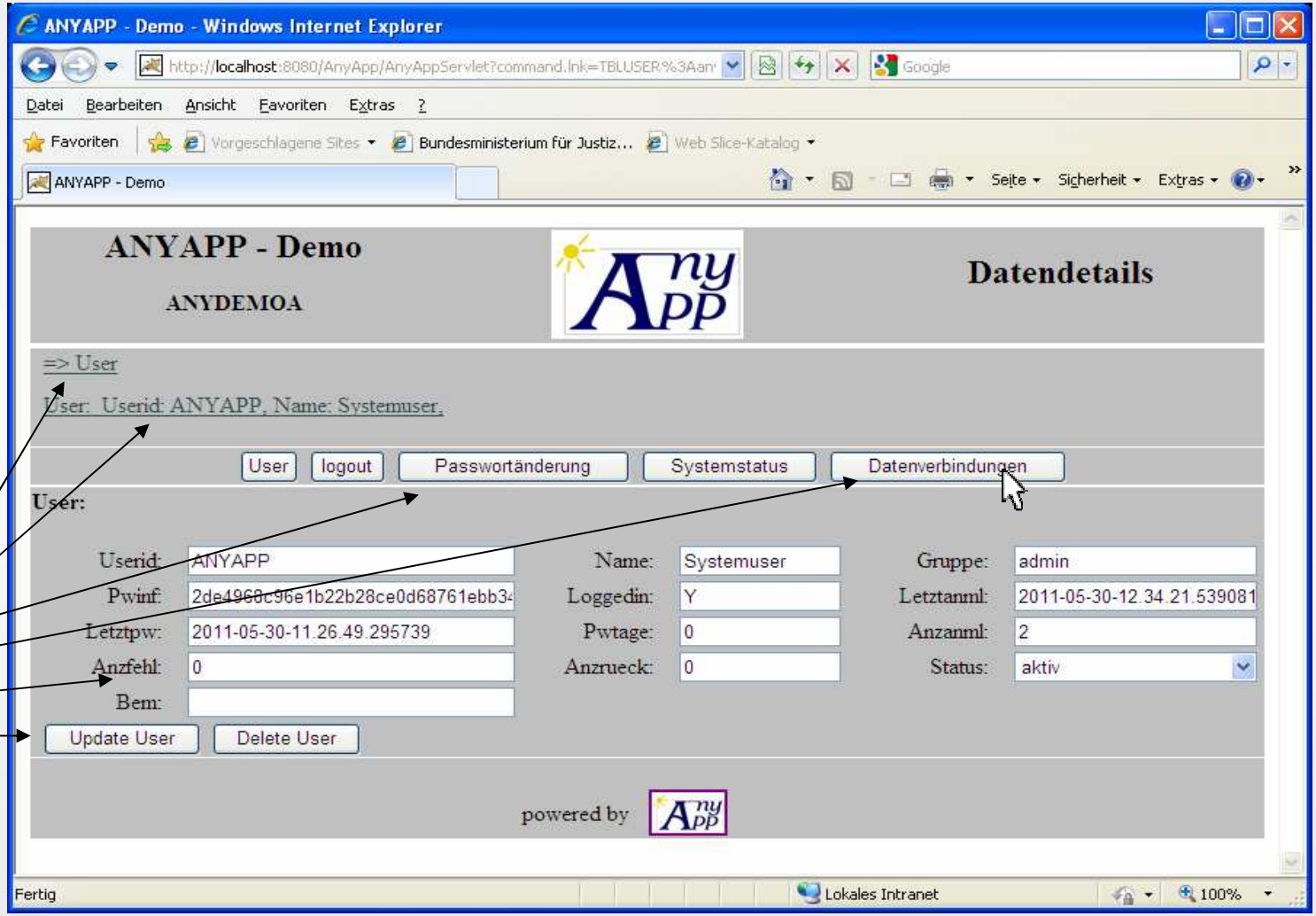
Das Ergebnis ist, wenig überraschend, der zur Zeit einzige User



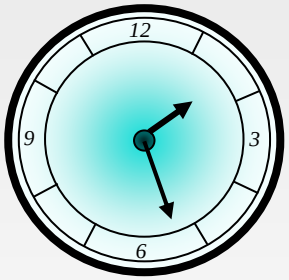
Ergebnisse sind immer auch Hyperlinks, der Klick auf die (noch einzige) Eintragung liefert …



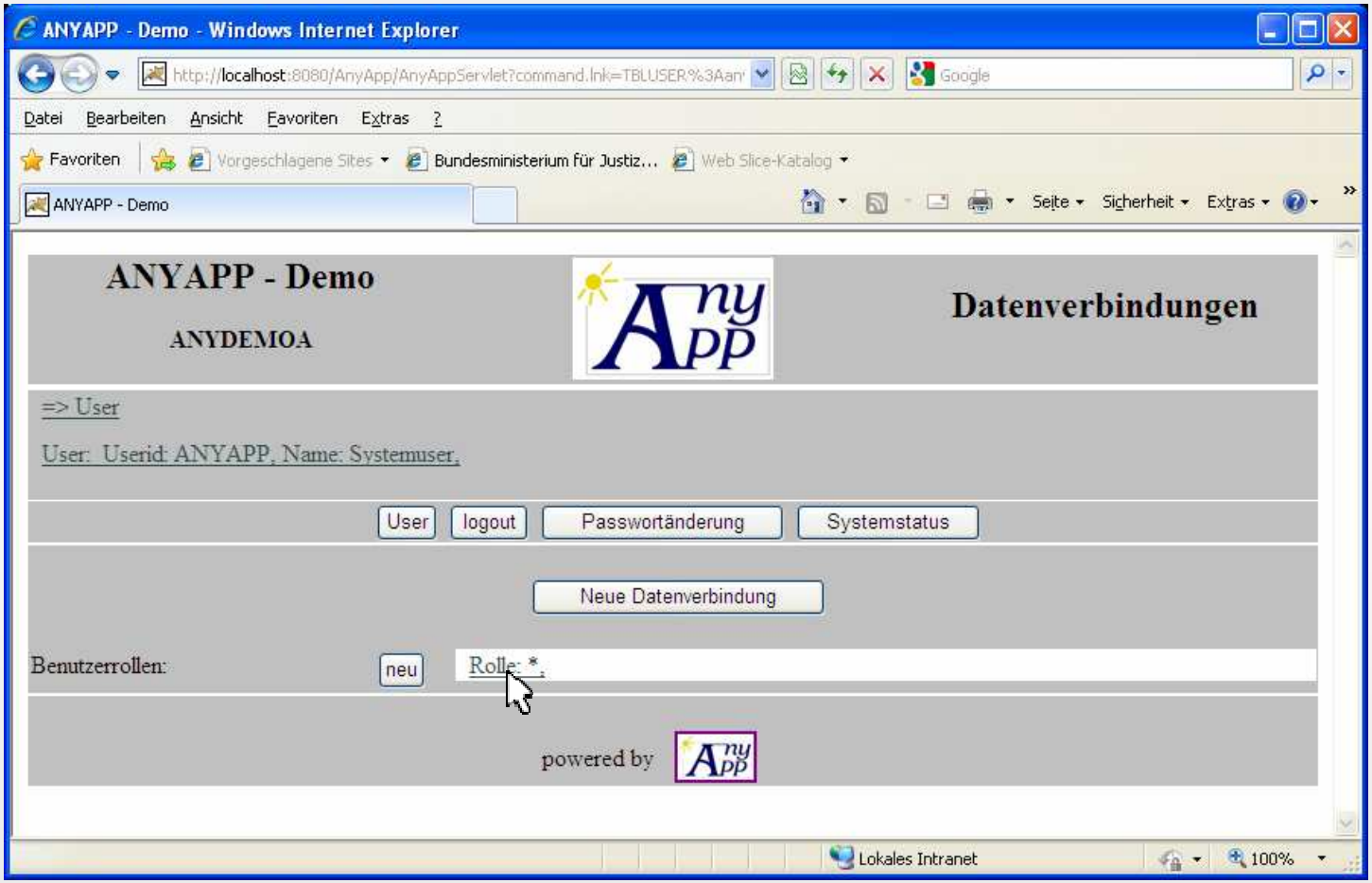
… die Details des Datenknotens: User ANYAPP.

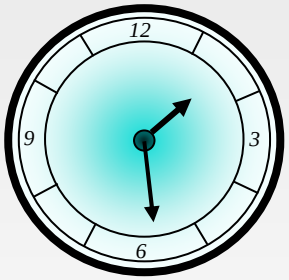


Aktueller Pfad
Knoten im Pfad
Grundfunktionen
Kontextfunktionen
Knotendetails
Knotenfunktionen

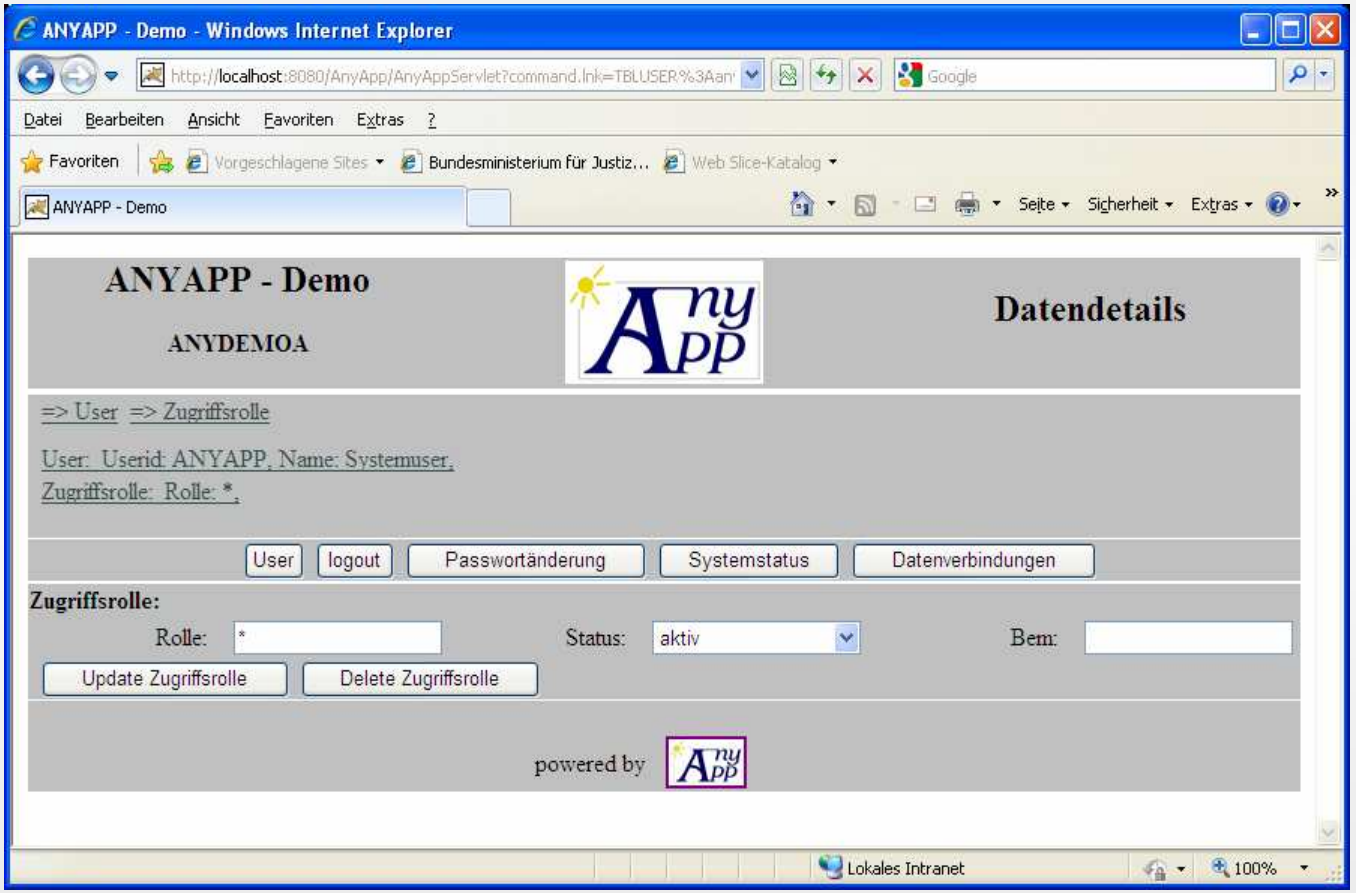


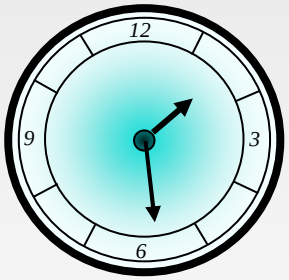
Klick auf Datenverbindungen bringt alle zu diesem Knoten verbundenen Datenknoten (in diesem Fall nur eine Verbindung zur Userrolle)



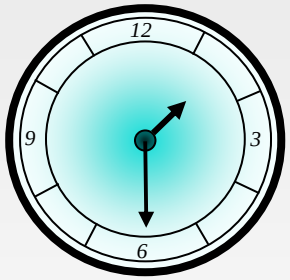


Der Klick auf ein verbundenes Element liefert wiederum dessen Details:





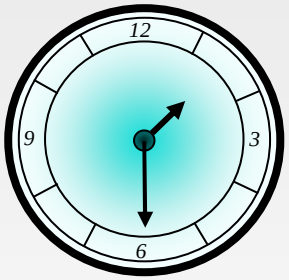
Wir haben erfolgreich unser erstes **AnyApp** – Framework installiert.



2. Applikationsdesign ‚Büchereiverwaltung'

   Folgende Grunddaten wurden identifiziert:

   1. Kunde (Nr, Vorname, Nachname, Titel, Geschlecht, Geburtsdatum, Strasse, Hausnr, PLZ, Ort, Dokument, Dokumentennr, Einschreibung_am, Gebührensaldo, letzte Einzahlung, Mahnstufe, Anzahl_Mahnungen)
   2. Buch (Nr,Titel, Verfasser, Verlag, Ausgabe, Erschienen, ISBN, Sachgebiet, Datum_Anschaffung, Status, Lagerort, Anzahl-Ausleihungen, letzte Ausleihung)
   3. Ausleihung (Status, Ausleihung_am, Rückgabe_am, Zustand)
   4. Vorschreibung (Datum, Status, Betrag, Gebührenart)
   5. Mahnung (Mahndatum, Mahnstufe, Mahngebühr, Mahnbetrag, Status)
   6. Verlag (Firmenname, Kontaktperson, Adresse)
   7. Bestellung (Status, bestellt_am, geliefert_am, Anzahl_betsellt, Anzahl_geliefert)



Die Datentabellen

AnyApp benötigt in der aktuellen Version V1.0 des CICS-Kernels zur Steuerung einige spezielle Datenfelder in jeder Datentabelle (blau eingefärbt), allen voran den eindeutigen Schlüssel XID. Es wäre zwar einfach, dieses Verhalten zu dynamisieren und über Tabellen- bzw. Feldattribute zu lösen. Darauf wurde jedoch zu Gunsten besserer Performance verzichtet.

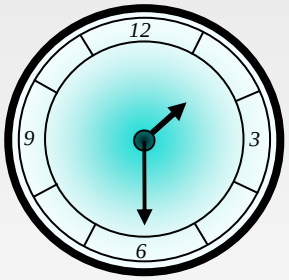
```
CREATE TABLE tblbuch (
  id integer NOT NULL generated by default as identity,
  xid    varchar(36) NOT NULL with default,
  isbn  varchar(32) ,
  verfasser          varchar(100),
  titel   varchar(255),
  sachgebiet        char(3),
  ...
  relid integer NOT NULL with default value(0),
  status char(4) NOT NULL with default value('0000'),
  erfdatum timestamp NOT NULL with default ,
  erfuser varchar(26) ,
  aenddatum timestamp generated by default FOR
          EACH ROW ON UPDATE AS ROW CHANGE
          TIMESTAMP NOT NULL,
  aenduser varchar(26) ,
  bem  varchar(100)
) in spadb01.demospc ;

create unique index tblbuch_pk on tblbuch(id);
alter    table tblbuch add constraint tblbuch_pk primary
          key(id);
create unique index tblbuch_i1 on tblbuch(xid);
create index tblbuch_i2 on tblbuch(isbn);
create index tblbuch_i3 on tblbuch(titel);
create index tblbuch_i4 on tblbuch(verfasser);
create index tblbuch_i5 on tblbuch(sachgebiet);
```

Im Allgemeinen ist diese Forderung bei der Neuerstellung einer Applikation einfach zu erfüllen, bei der Modifikation bestehender Tabellen oder wenn andere Schlüssel-strukturen vonnöten sind, bedarf es evtl. spezieller Definitionen.

Als Beispiel die Buch-Tabelle unserer Demo:

Um Kundenkarten und Bücher einfach mit einem Barcode versehen zu können, ist ein numerischer Schlüssel vorzuziehen, der möglichst automatisch unique, somit also als Identitätsschlüssel definiert ist. Damit muss für den systemseitig notwendigen **AnyApp**-Schlüssel XID ein weiterer Unique-Key definiert werden.

Der ISBN-Schlüssel hingegen muss z.B. als nonunique angelegt werden, sonst könnte man nicht mehrere gleiche Bücher auflegen.



4. Generieren der Applikationstabellen wie von der DBA zur Verfügung gestellt.

DSS1 - Run Multiple SQL Statements

Close Edit View Help

Auto Commit: On / Off | Stop On Error?: Yes / Ignore errors on Drops / No / More | Select: All / None | Start from Stmt: 1 | Check Syntax: Yes | Parameters: Always Prompt

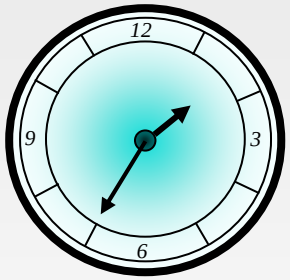
	Run?	Line No	Statement	Elapsed	Result
34	Completed	143	drop table tblverlag	0,36	SQL executed OK
35	Completed	144	CREATE TABLE tblverlag (xidvarchar(36) NOT NULL with default, ..	0,06	SQL executed OK
36	Completed	163	create unique index tblverlag_pk on tblverlag(xid)	0,19	SQL executed OK
37	Completed	164	alter table tblverlag add constraint tblverlag_pk primary key(xid)	0,08	SQL executed OK
38	Completed	165	create index tblverlag_i1 on tblverlag(verlag)	0,19	SQL executed OK
39	Completed	166	GRANT DELETE,INSERT,SELECT,UPDATE ON TABLE tblverlag ..	0,08	SQL executed OK
40	Completed	170	drop table tblbestellung	0,22	SQL executed OK
41	Completed	171	CREATE TABLE tblbestellung (xidvarchar(36) NOT NULL with defa..	0,08	SQL executed OK
42	Completed	186	create unique index tblbestellung_pk on tblbestellung(xid)	0,17	SQL executed OK
43	Completed	187	alter table tblbestellung add constraint tblbestellung_pk primary key(xi..	0,06	SQL executed OK
..	Completed	188	GRANT DELETE,INSERT,SELECT,UPDATE ON TABLE tblbestellu..	0,06	SQL executed OK

GRANT DELETE,INSERT,SELECT,UPDATE ON TABLE tblbestellung TO USWDB2

SQL executed OK

Run | Exit | Abort | Pause

Completed processing 44 statements



Die neuen Tabellen in domtbl eintragen ..

Die Wartungstätigkeit der Applikationsbetreuung besteht vor allem in der Pflege der Steuertabellen (domtbl, domrel, domfield, domstatus, sowie der Wertebereichstabelle domsel).

DSS1 - [Results - SPA00.DOMTBL - DSS1]

File Edit Edit Data Link View Row Compare Tools Options Window Help

Lock Print Copy Find Edit Link Numbers Group 2-color Height Format Detail Upd Del Dup New Refresh Search SQL Query Builder Pivot Help Abort

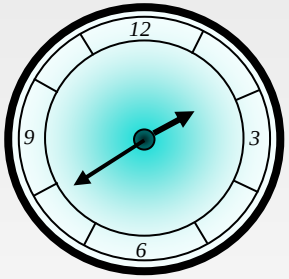
Drag a column header here to group by that column

TBL	TBLNAME	PLURAL	KURZAUSKUNFT	BEZEICHNER	ISSTAMM
domaccess	intern, Systemzugriff	intern, Systemzugriffe		Y	N
domfield	intern, spez. Feldattribute	intern, spez. Feldattribute		Y	N
domrel	intern, Tabellen-Beziehungsdomäne	intern, Tabellen-Beziehungsdomäne		Y	N
domstatus	Wertebereich Status	Wertebereich Status		Y	N
domtbl	intern, Tabelleninfodomäne	intern, Tabelleninfodomäne		Y	N
tblrel	Beziehung	Beziehungen		Y	N
tblrolle	Zugriffsrolle	Zugriffsrollen	rolle,status	Y	N
tbluser	User	User	userid,name,status	Y	Y
tblbuch	Buch	Bücher	titel,verfasser,status	Y	Y
tblkunde	Kunde	Kunden	vorname,titel,famname	Y	Y
tblausleihung	Entlehnung	Entlehnungen	dat_ausl,status	Y	N
tblvorschreibung	Vorschreibung	Vorschreibungen	datum,betrag,status	Y	N
tblmahnung	Mahnung	Mahnungen	mahndatum,mahnbetrag,status	Y	N
tblverlag	Verlag	Verlage	verlag	Y	Y
tblbestellung	Bestellung	Bestellungen	dat_best	Y	N

Select * from SPA00.DOMTBL | 15 rows | 18 cols | 0,13 secs | 04:46 pm | DSS1

15 rows displayed (0,13 secs)

DSS1 SPA00.DOMTBL SPA00.DOMTBL



Nun noch die möglichen Relationen in domrel eintragen und alles autorisieren (allgemeine Autorisierungs-SQL liegt bei):

DSS1 - [Results - SPA00.DOMREL - DSS1]

File Edit Edit Data Link View Row Compare Tools Options Window Help

Lock Print Copy Find Edit Link Numbers Group 2-color Height Format Detail Upd Del Dup New Refresh Search SQL Query Builder Pivot Help Abort

Drag a column header here to group by that column

ID	TBL1	TBL2	REL	REIHUNG	HIST	MAXTYPE	MAXANZ	BED1	BED2	FORWARD	BACKWARD
1	tbluser	tblrolle	has	5	N	S	999	(null)	(null)	Benutzerrollen	User
2	tblkunde	tblausleihung	has	5	Y	S	1	(null)	(null)	Ausleihung	Kunde
3	tblausleihung	tblbuch	use	5	N	S	5	(null)	(null)	Buch	Ausleihung
4	tblbuch	tblverlag	use	5	N	S	32760	(null)	(null)	Verlag	Buch
5	tblkunde	tblvorschreibung	has	5	N	S	9	(null)	(null)	Vorschreibungen	Kunde
6	tblkunde	tblmahnung	has	5							
7	tblmahnung	tblvorschreibungn	use	5							
8	tblverlag	tblbestellung	has	5							
9	tblbestellung	tblbuch	use	5							

Select * from SPA00.DOMREL

9 rows displayed (0,10 secs)

DSS1 SPA00.DOMTBL SPA00.DOMREL

DSS1 - Run Multiple SQL Statements

Close Edit View Help

Auto Commit: On Off | Stop On Error?: Yes Ignore errors on Drops No More | Select: All None | Start from Stmt: 1 | Check Syntax: Yes | Parameters: Always Prompt

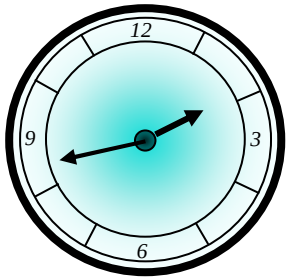
	Run?	Line No	Statement	Elapsed	Result
1	Completed	1	set current sqlid = 'SPA00'	0,08	SQL executed OK
2	Completed	3	delete from domaccess	0,08	SQL0513W Durch die SQL-Anweisung wird die gesa
3	Completed	4	insert into domaccess(tbl,rolle,zugriff) select tbl,'*','rwx' from domtbl	0,08	SQL executed OK. Number of rows affected=15
4	Completed	5	insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYAPP-Demo-Ger','..	0,06	SQL executed OK. Number of rows affected=15
5	Completed	6	insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYAPP-Demo-Upd',..	0,08	SQL executed OK. Number of rows affected=15
6	Completed	7	insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYAPP-Demo-Adm'..	0,06	SQL executed OK. Number of rows affected=15
7	Completed	8	insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYDEMOG','r--' fro..	0,08	SQL executed OK. Number of rows affected=15
8	Completed	9	insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYDEMOU','rw-' fro..	0,06	SQL executed OK. Number of rows affected=15
9	Completed	10	insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYDEMOA','rwx' fro..	0,08	SQL executed OK. Number of rows affected=15

insert into domaccess(tbl,rolle,zugriff) select tbl,'ANYDEMOA','rwx' from domtbl

SQL executed OK. Number of rows affected=15

Run Exit Abort Pause

Completed processing 9 statements



Wir gehen wieder zum Systemstatus :

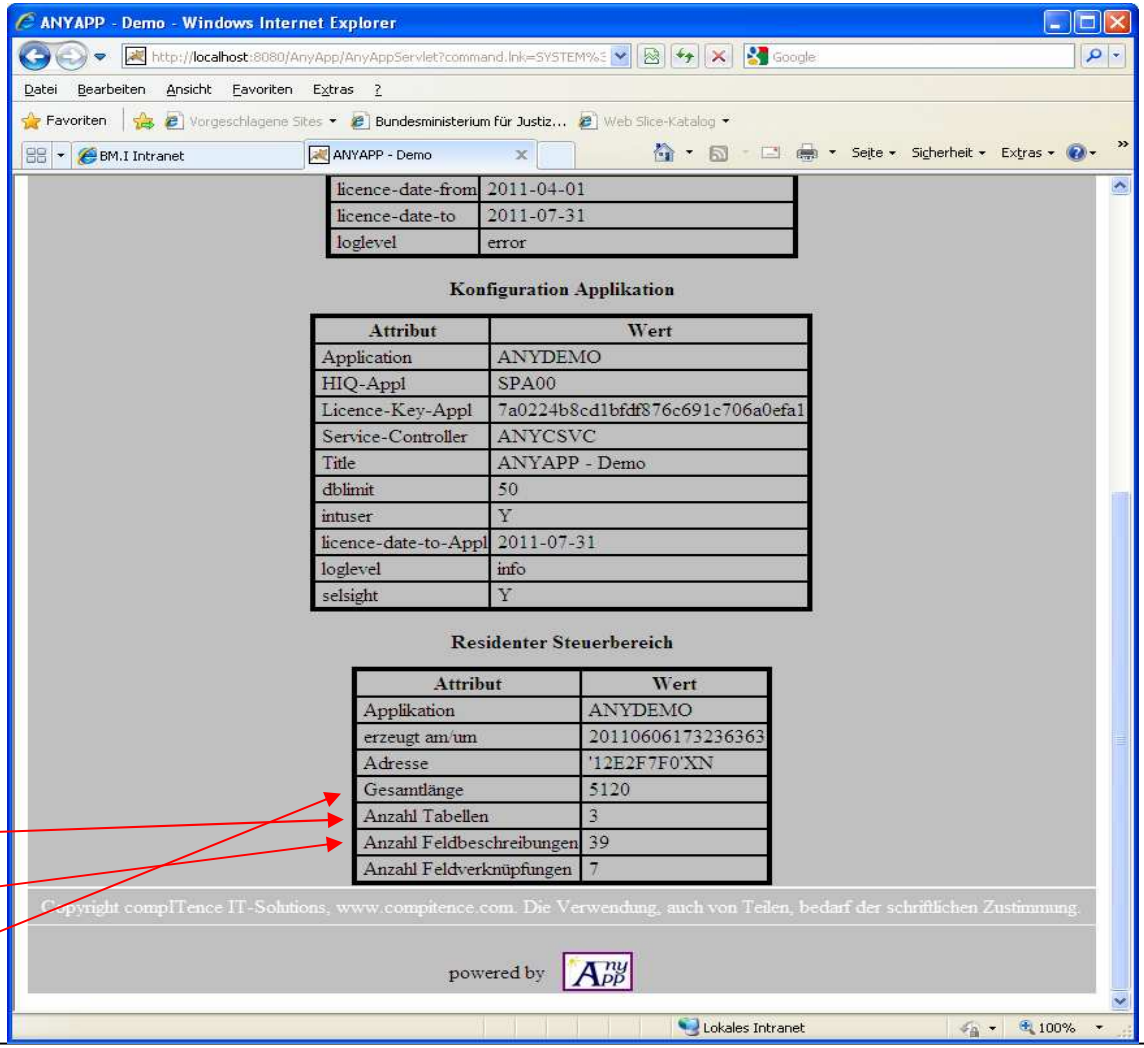
... und scrollen zum Block: ‚Residenter Steuerbereich‘

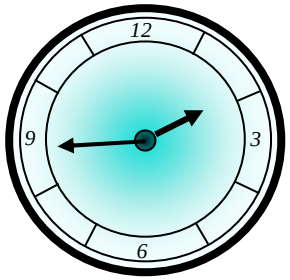
Merken:

3 Tabellen

39 Felder

5120 Bytes





… und aktualisieren mittels Refresh-Button:

nun:

10 Tabellen

147 Felder

17.408 Bytes

ANYAPP - Demo - Windows Internet Explorer

ANYAPP - Demo

ANYDEMOA

Systemstatus

Returncode: 0

0100: Änderung durchgeführt (,)

Administration

Systeminfos | Applikationstabellen | Refresh | logout | Passwortänderung | Datenwartung

Systeminformationen ANYAPP - Demo

licence-date-from	2011-04-01
licence-date-to	2011-07-31
loglevel	error

Konfiguration Applikation

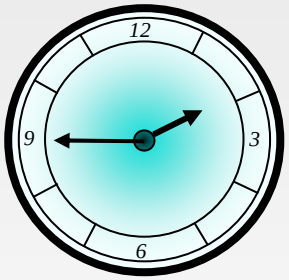
Attribut	Wert
Application	ANYDEMO
HIQ-Appl	SPA00
Licence-Key-Appl	7a0224b8cd1bfdf876c691c706a0efa1
Service-Controller	ANYCSVC
Title	ANYAPP - Demo
dblimit	50
intuser	Y
licence-date-to-Appl	2011-07-31
loglevel	info
selsight	Y

Residenter Steuerbereich

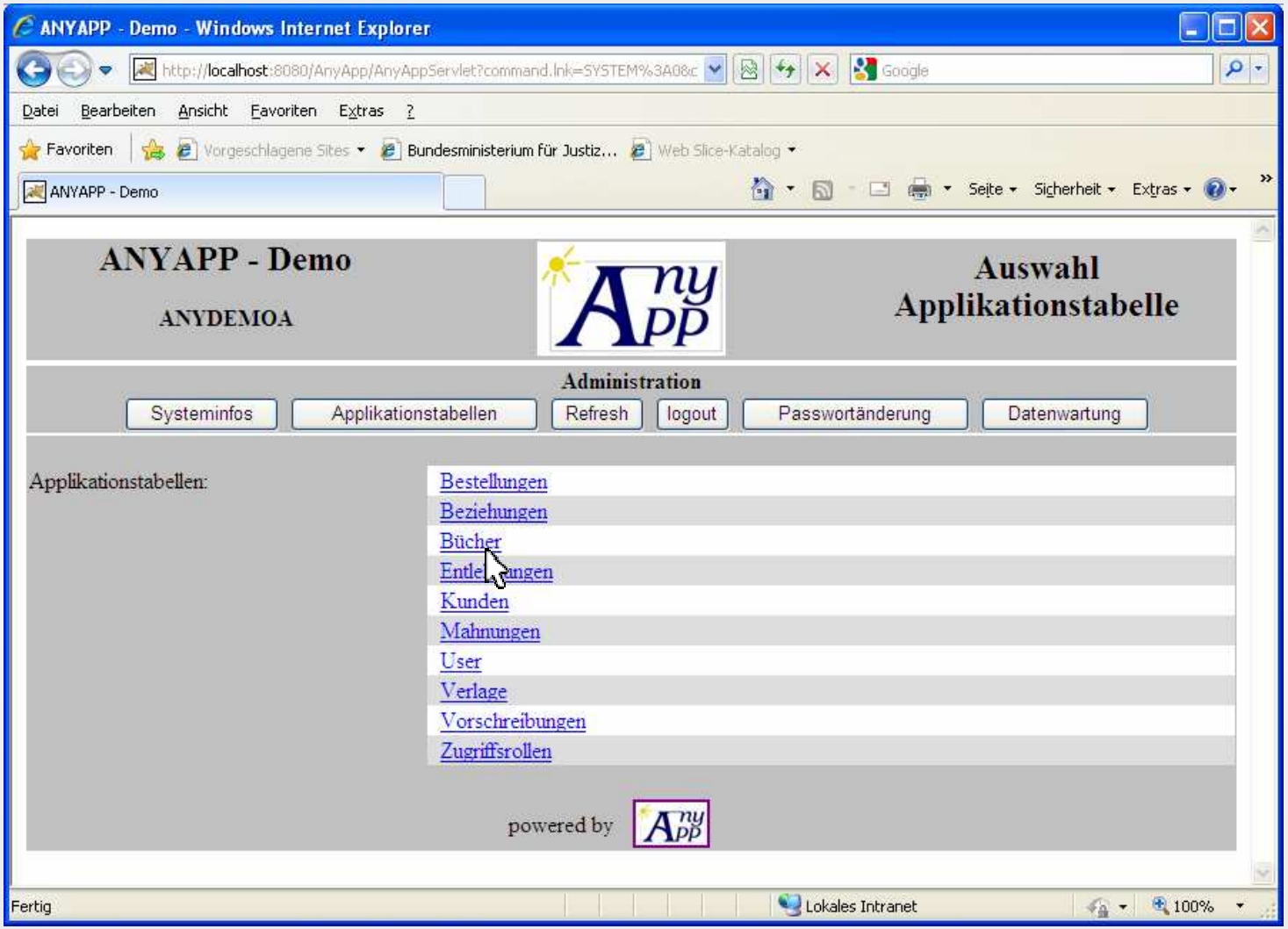
Attribut	Wert
Applikation	ANYDEMO
erzeugt am/um	20110606102425065
Adresse	'12E2F7F0'XN
Gesamtlänge	17408
Anzahl Tabellen	10
Anzahl Feldbeschreibungen	147
Anzahl Feldverknüpfungen	34

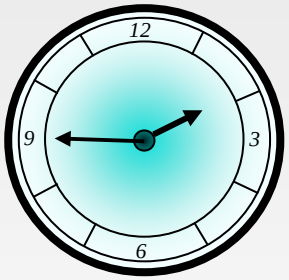
Copyright compITence IT-Solutions, www.compitence.com. Die Verwendung, auch von Teilen, bedarf der schriftlichen Zustimmung.

powered by

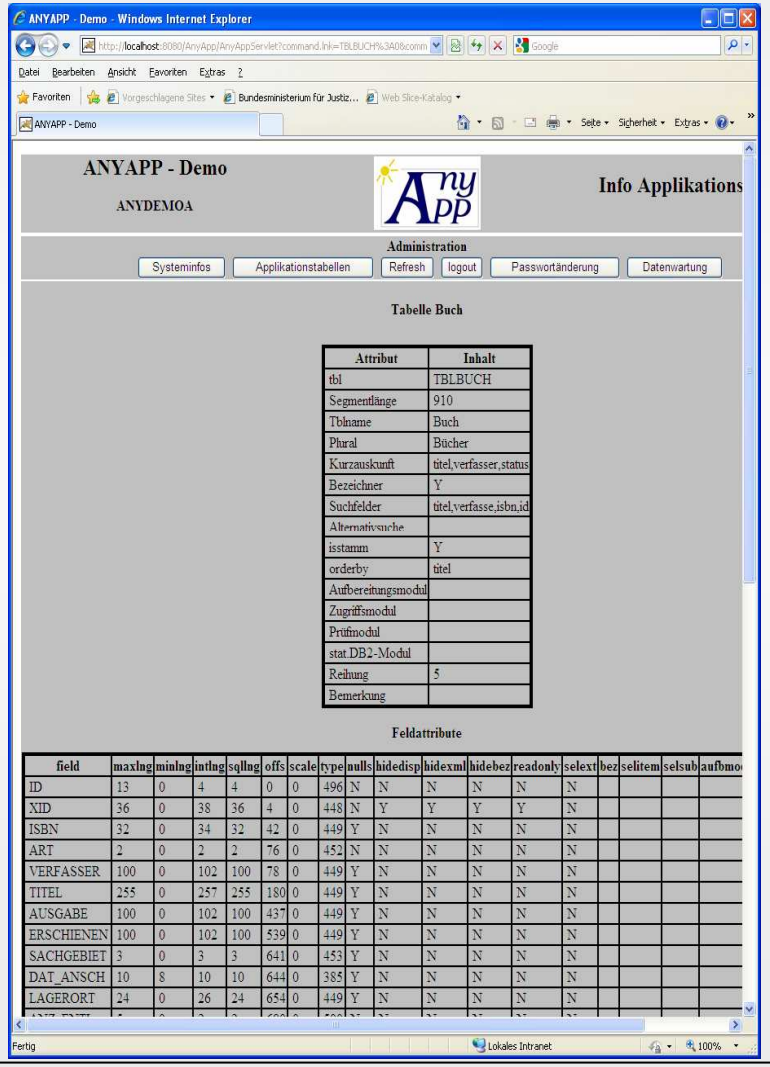


und werfen einen Blick auf die Einbindung der neuen Tabellen:





Bzw. Im Detail nach Anwahl von z.B. Bücher:



ANYAPP - Demo - Windows Internet Explorer

ANYAPP - Demo

ANYDEMOA

Info Applikations

Administration

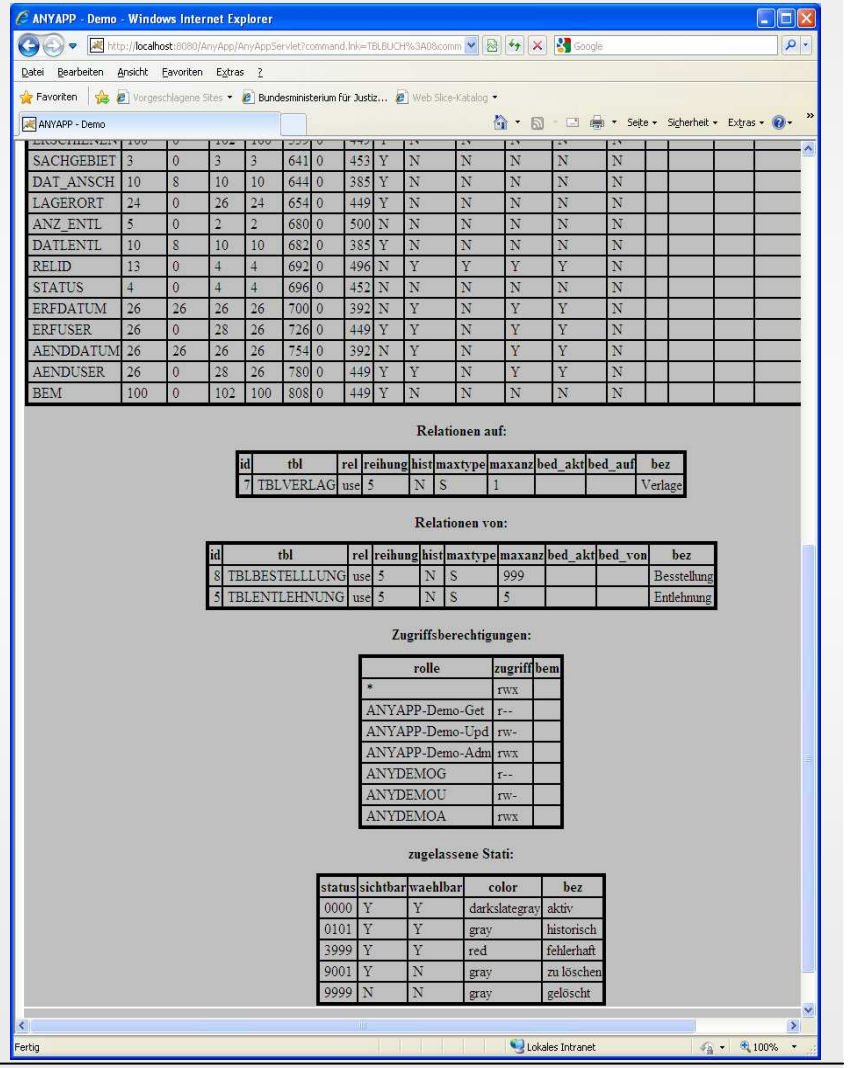
Systeminfos | Applikationstabellen | Refresh | logout | Passwortänderung | Datenwartung

Tabelle Buch

Attribut	Inhalt
tbl	TBLBUCH
Segmentlänge	910
Tblname	Buch
Plural	Bücher
Kurzauskunft	titel,verfasser,status
Bezeichner	Y
Suchfelder	titel,verfasse,isbn,id
Alternativsuche	
isstamm	Y
orderby	titel
Aufbereitungsmodul	
Zugriffsmodul	
Prüfmodul	
stat.DB2-Modul	
Reihung	5
Bemerkung	

Feldattribute

field	maxlng	minlng	intlng	sqllng	offs	scale	type	nulls	hidedisp	hidexml	hidebez	readonly	selext	bez	selitem	selsub	aufbmo
ID	13	0	4	4	0	0	496	N	N	N	N	N	N				
XID	36	0	38	36	4	0	448	N	Y	Y	Y	Y	N				
ISBN	32	0	34	32	42	0	449	Y	N	N	N	N	N				
ART	2	0	2	2	76	0	452	N	N	N	N	N	N				
VERFASSER	100	0	102	100	78	0	449	Y	N	N	N	N	N				
TITEL	255	0	257	255	180	0	449	Y	N	N	N	N	N				
AUSGABE	100	0	102	100	437	0	449	Y	N	N	N	N	N				
ERSCHIENEN	100	0	102	100	539	0	449	Y	N	N	N	N	N				
SACHGEBIET	3	0	3	3	641	0	453	Y	N	N	N	N	N				
DAT_ANSCH	10	8	10	10	644	0	385	Y	N	N	N	N	N				
LAGERORT	24	0	26	24	654	0	449	Y	N	N	N	N	N				



ANYAPP - Demo - Windows Internet Explorer

SACHGEBIET	3	0	3	3	641	0	453	Y	N	N	N	N	N				
DAT_ANSCH	10	8	10	10	644	0	385	Y	N	N	N	N	N				
LAGERORT	24	0	26	24	654	0	449	Y	N	N	N	N	N				
ANZ_ENTL	5	0	2	2	680	0	500	N	N	N	N	N	N				
DATLENTL	10	8	10	10	682	0	385	Y	N	N	N	N	N				
RELID	13	0	4	4	692	0	496	N	Y	Y	Y	Y	N				
STATUS	4	0	4	4	696	0	452	N	N	N	N	N	N				
ERFDATUM	26	26	26	26	700	0	392	N	Y	N	Y	Y	N				
ERFUSER	26	0	28	26	726	0	449	Y	Y	N	Y	Y	N				
AENDDATUM	26	26	26	26	754	0	392	N	Y	N	Y	Y	N				
AENDUSER	26	0	28	26	780	0	449	Y	Y	N	Y	Y	N				
BEM	100	0	102	100	808	0	449	Y	N	N	N	N	N				

Relationen auf:

id	tbl	rel	reihung	hist	maxtype	maxanz	bed_akt	bed_auf	bez
7	TBLVERLAG	use	5	N	S	1			Verlage

Relationen von:

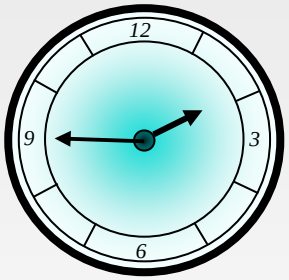
id	tbl	rel	reihung	hist	maxtype	maxanz	bed_akt	bed_von	bez
8	TBLBESTELLLUNG	use	5	N	S	999			Besstellung
5	TBLENTLEHNUNG	use	5	N	S	5			Entlehnung

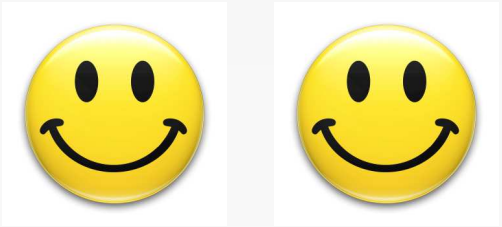
Zugriffsberechtigungen:

rolle	zugriff	bem
*	rwx	
ANYAPP-Demo-Get	r--	
ANYAPP-Demo-Upd	rw-	
ANYAPP-Demo-Adm	rwx	
ANYDEMOG	r--	
ANYDEMOU	rw-	
ANYDEMOA	rwx	

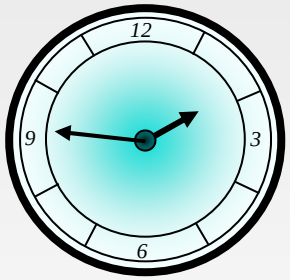
zugelassene Stati:

status	sichtbar	waehlbar	color	bez
0000	Y	Y	darkslategray	aktiv
0101	Y	Y	gray	historisch
3999	Y	Y	red	fehlerhaft
9001	Y	N	gray	zu löschen
9999	N	N	gray	gelöscht

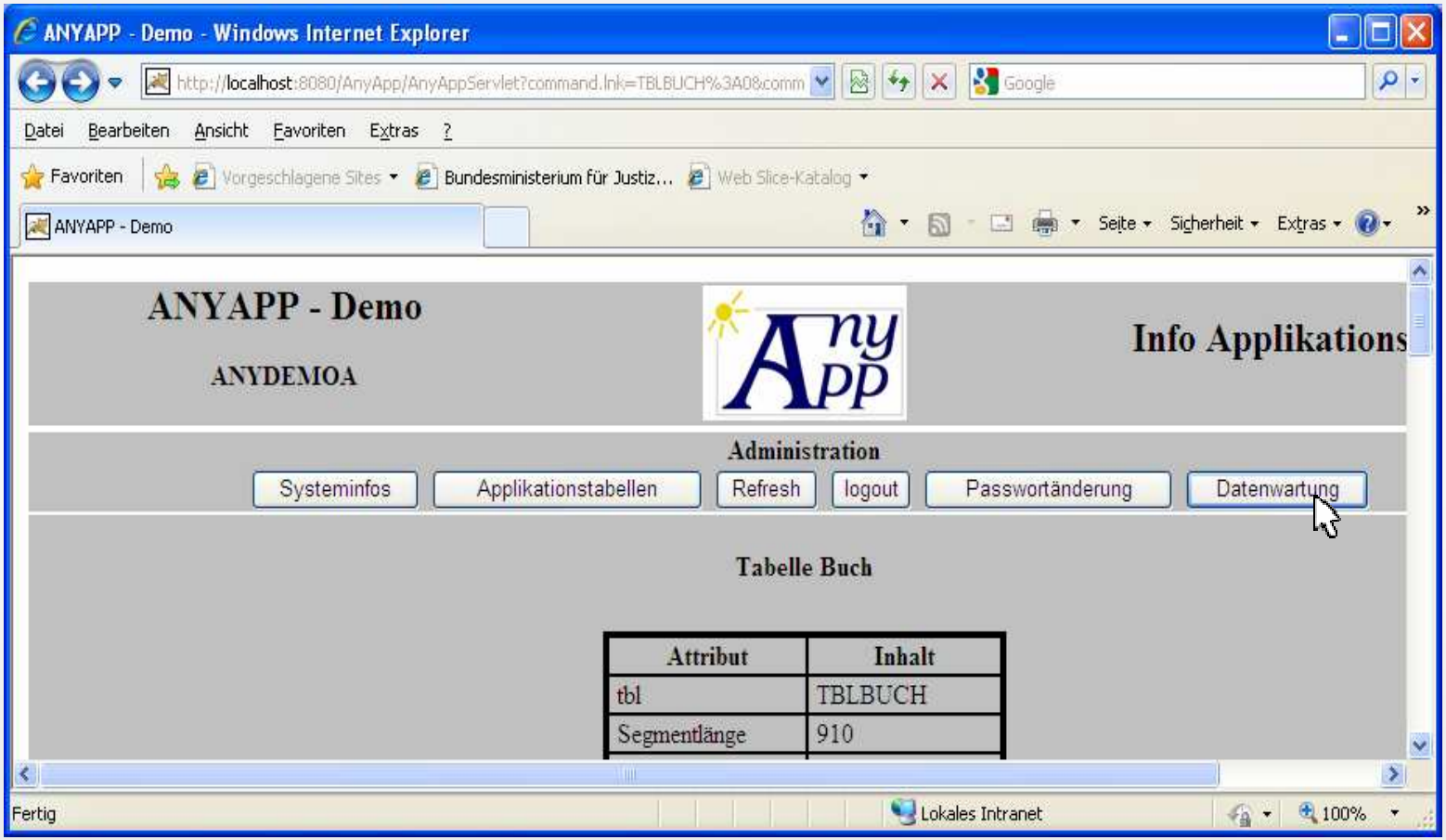


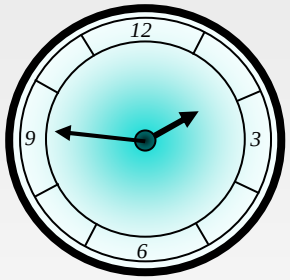


Wir haben erfolgreich unsere erste **AnyApp** – Applikation installiert.

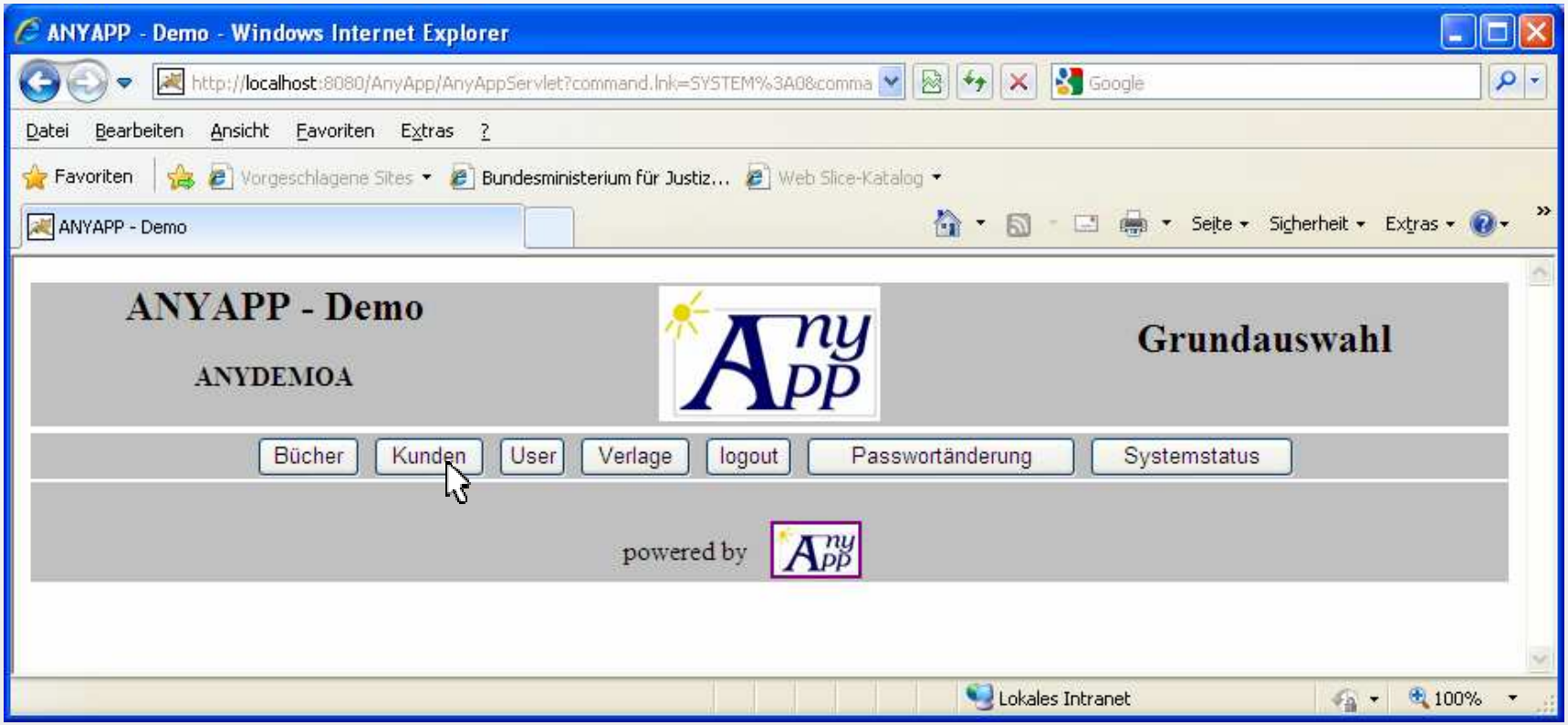


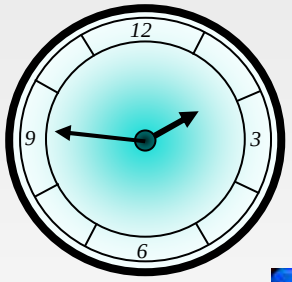
Nun gilt es ein paar Daten zu erfassen:



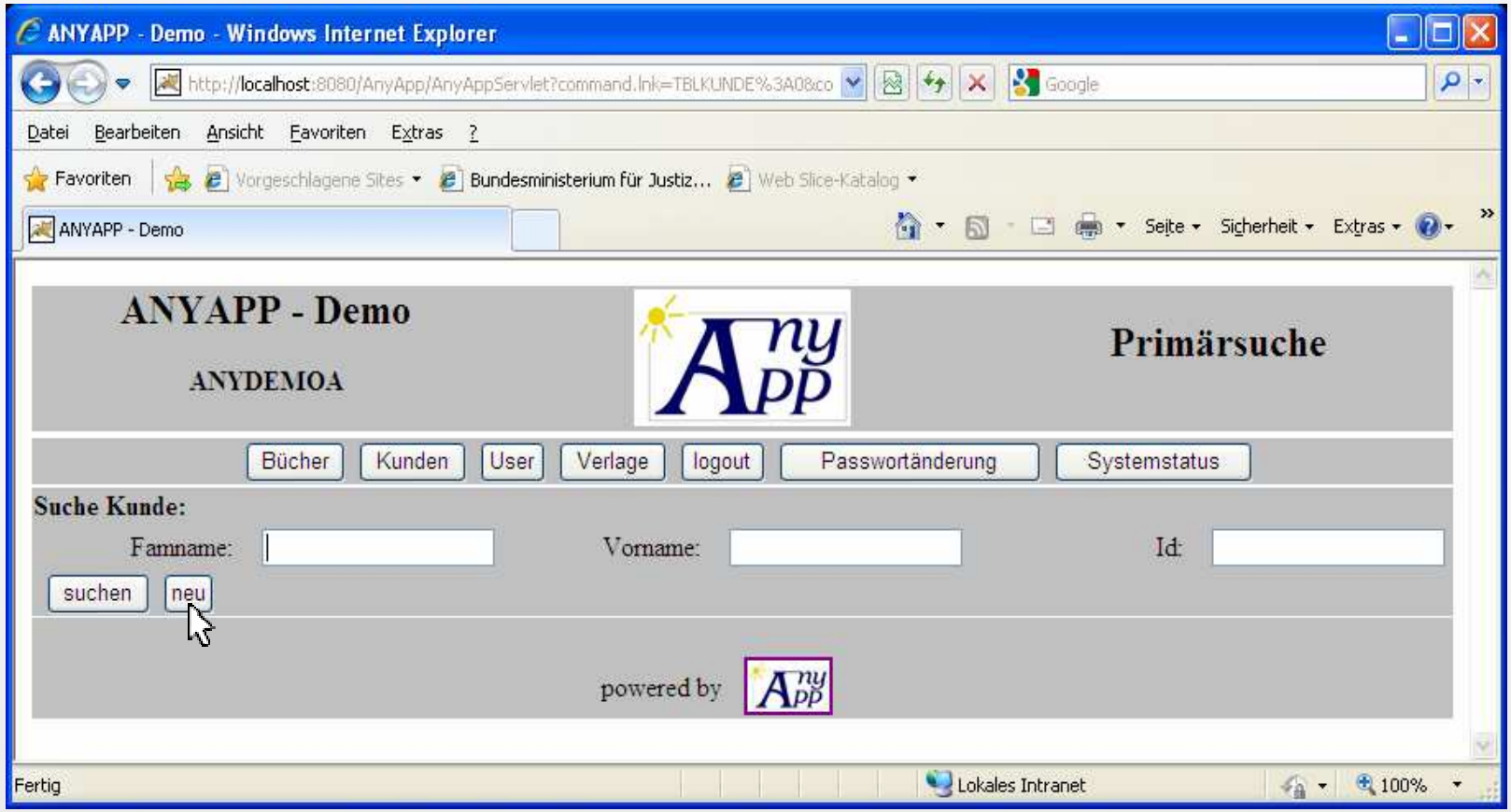


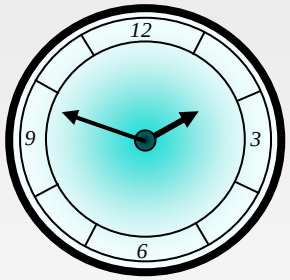
Wir beginnen mit einem Kunden:





Die Grundauswahl zu Basistabelle geht immer über eine Suche (beachte die Suchfelder lt. Eintragung in der domtbl). Bei entsprechender Markierung wird jedoch auch die Neuerfassung ohne vorhergegangene Suche zugelassen (siehe hier Button neu):



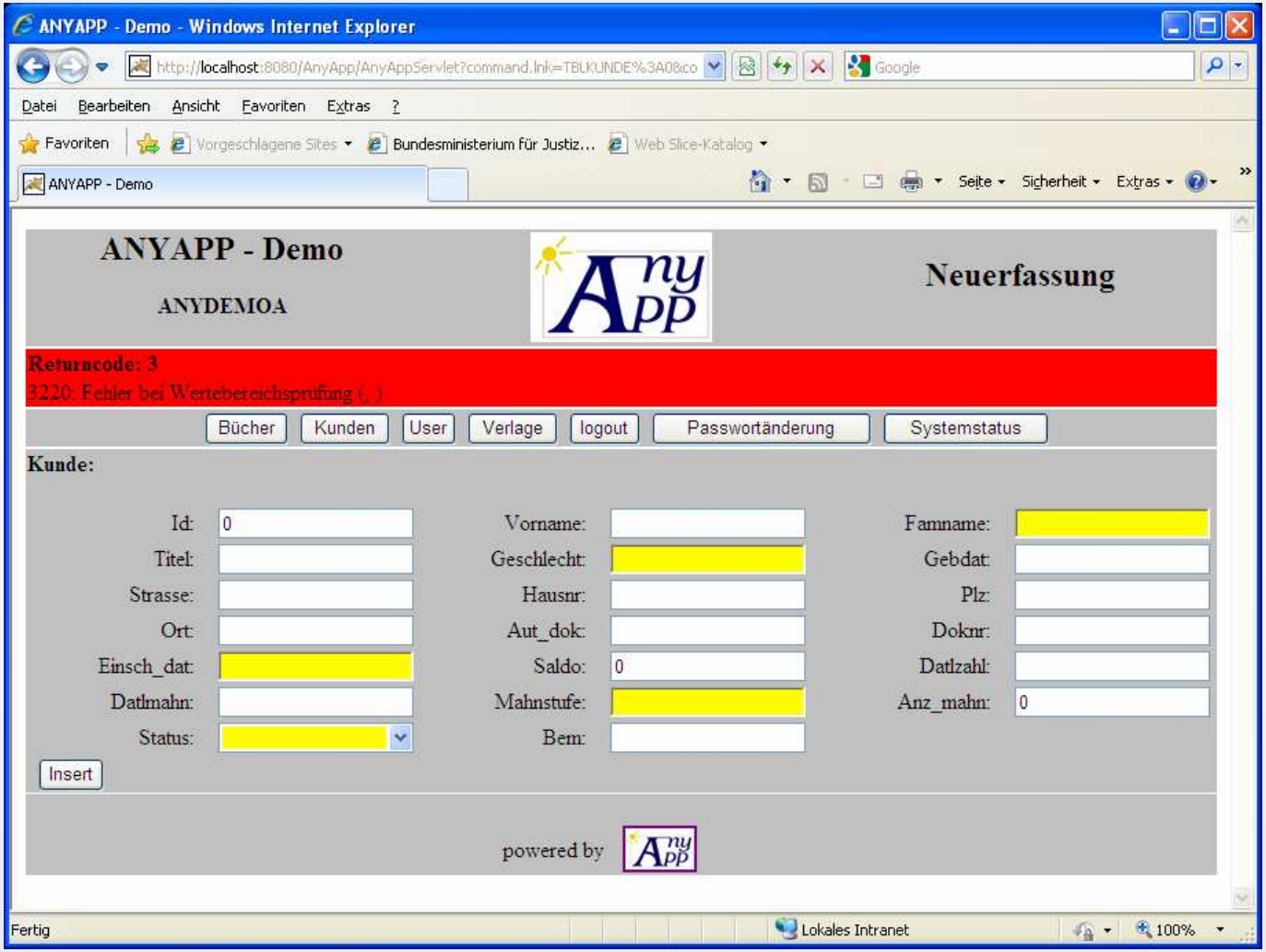


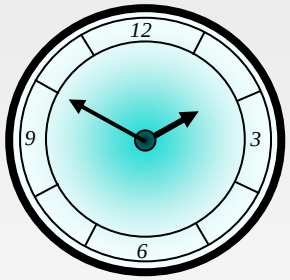
Neuerfassung ‚Leer' – schirm

AnyApp überprüft ein Segment generell bei jedem Senden. Daher finden sich Fehlerhinweise auch bei der ‚leeren' Neuerfassung.

Dies ist zwar evtl. ungewohnt, erspart jedoch oft einen zusätzlichen Dialogschritt,

Gelb unterlegte Felder sind leer aber mandatory (NULLS = ‚N')

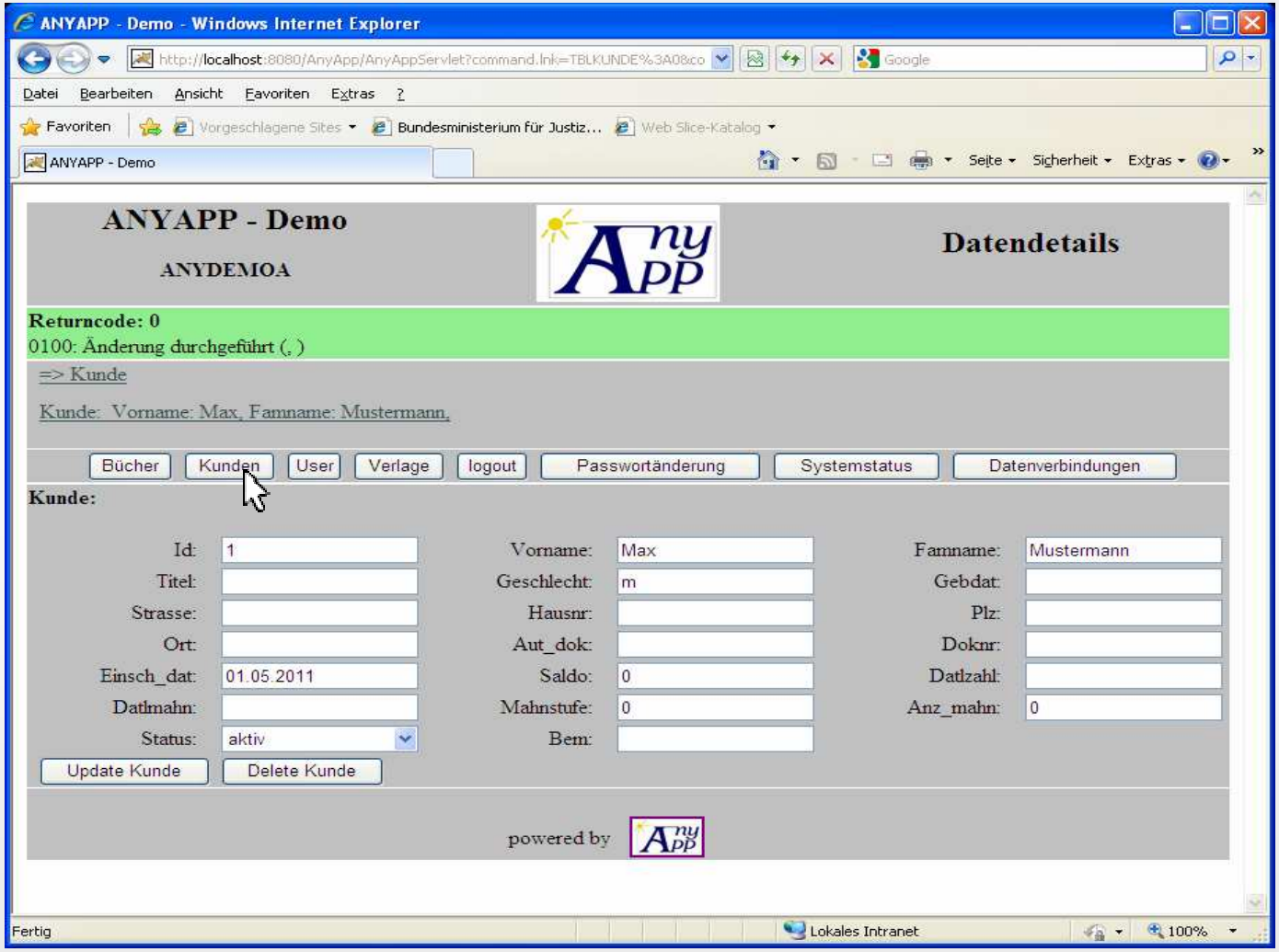


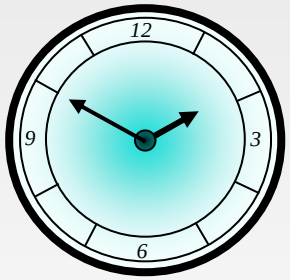


Neuerfassung durchgeführt .

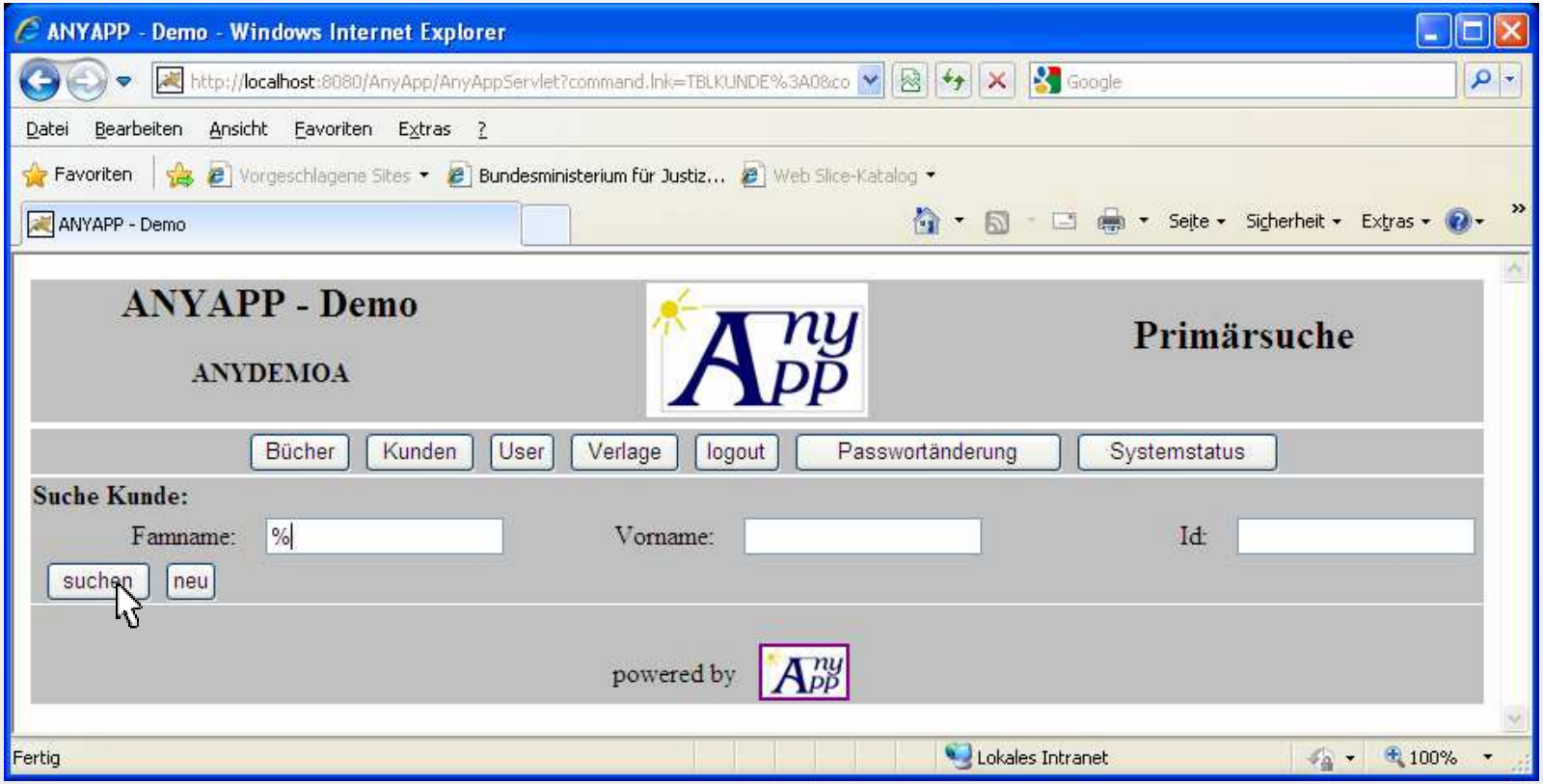
Ist alles korrekt eingetragen, wird das Segment bei Drücken von ‚insert' angelegt.

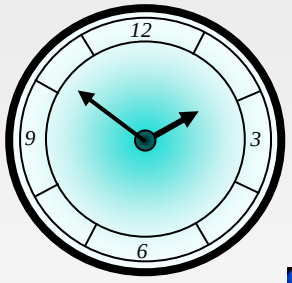
Wir gehen noch einmal zur Kundensuche



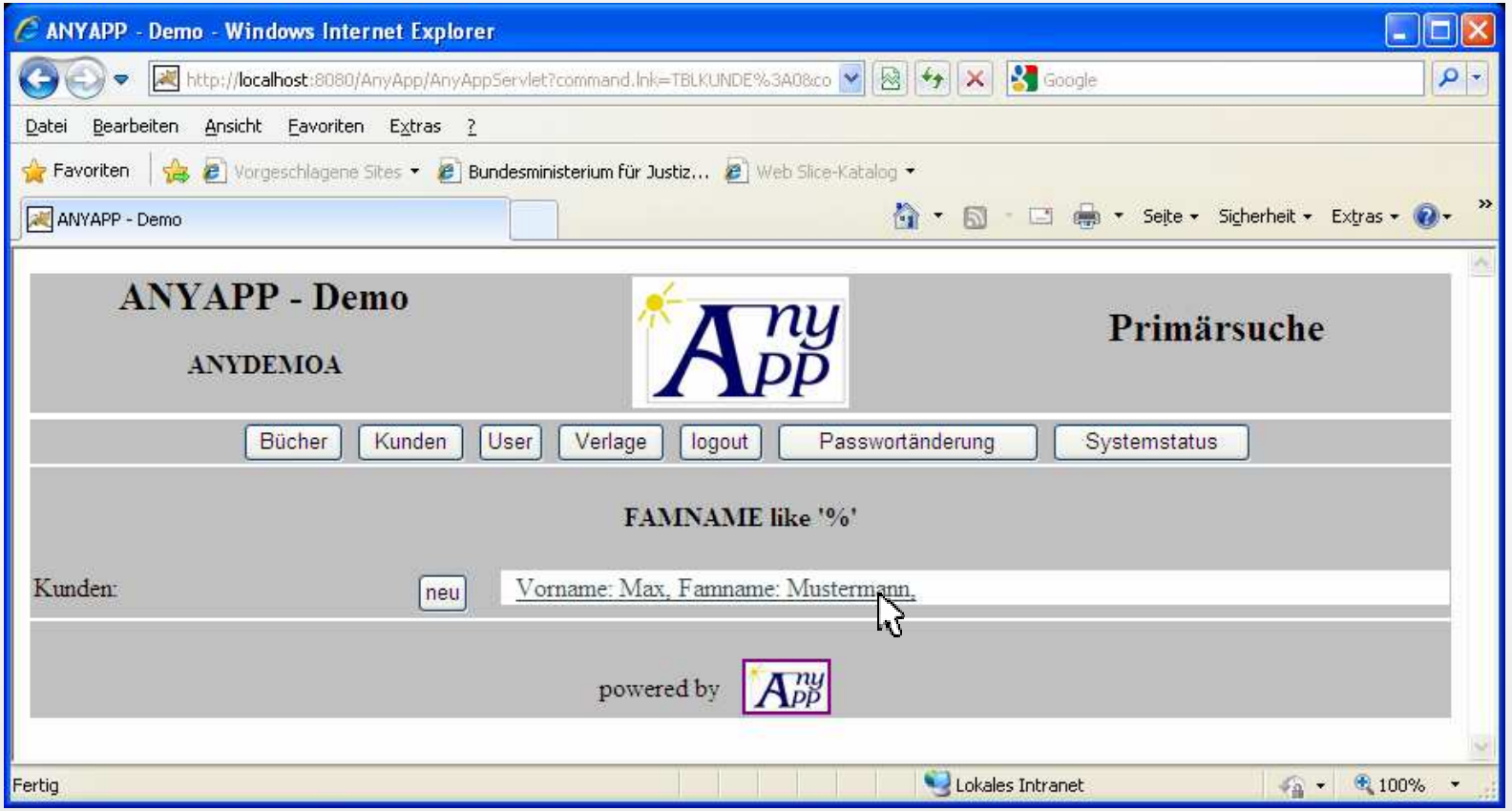


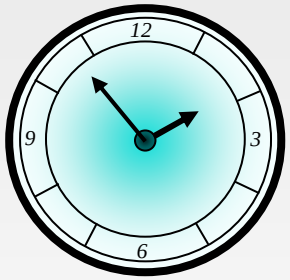
Wir schauen, welche Kunden es nun gibt.





… und sehen unseren neuen Kollegen:

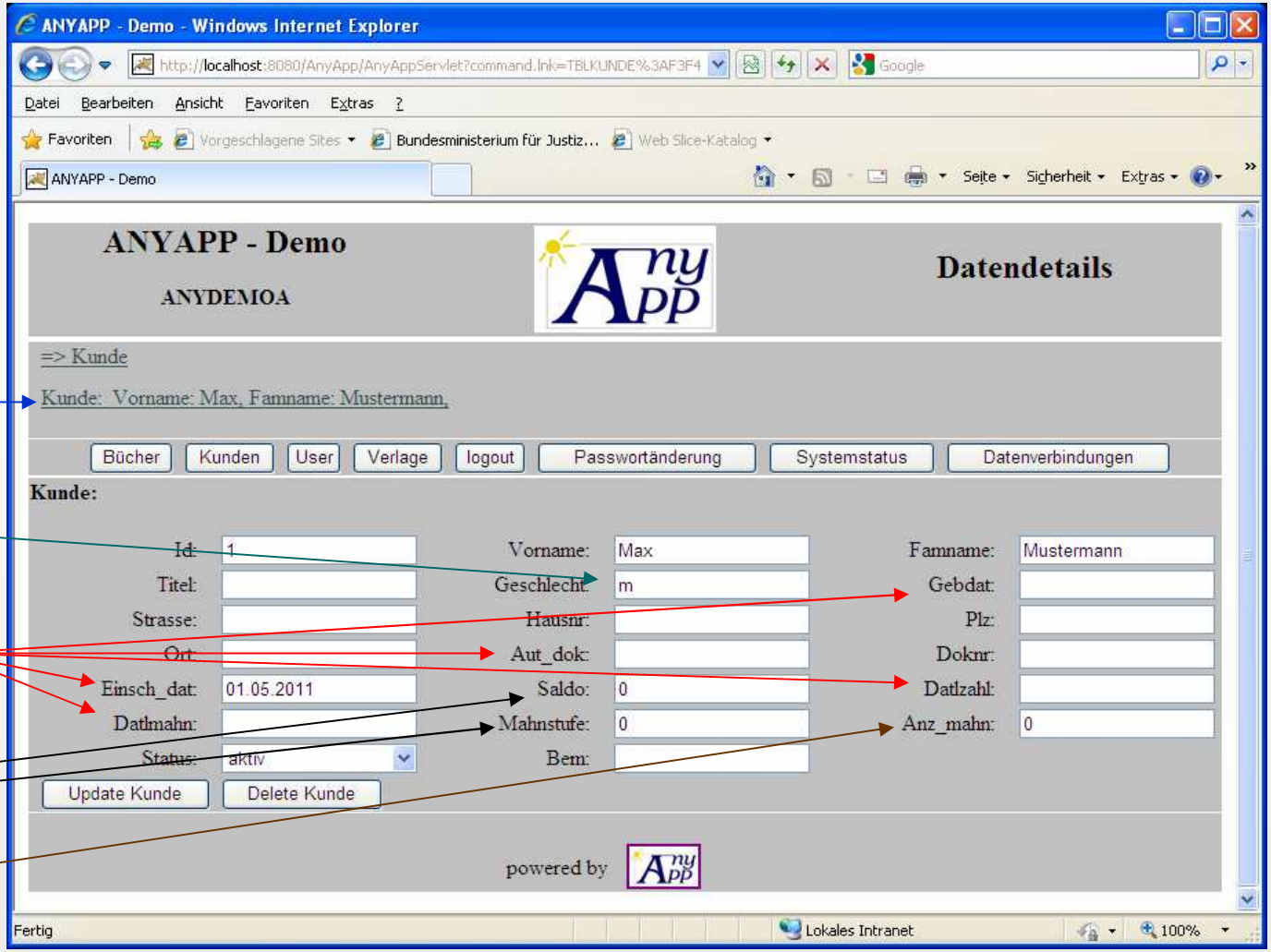


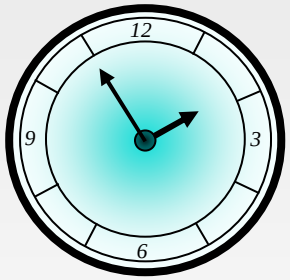


Durch Anklicken des Eintrags geht‘s zurück zum Detailschirm:

Da gibt es Verbesserungs-potential:

- Beim Knoten Kunden sollen keine Feldnamen stehen
- Geschlecht sollte im Klartext angezeigt werden
- Div ‚hässliche‘ Feldbezeichnungen
- Einige Felder sollten geschützt werden
- Einige nicht angezeigt





Die Textfarben entsprechen den Farben der Pfeile im vorhergegangenen Bild

So lassen sich die Anzeigeeigenschaften bei sämtlichen Datenfeldern individuell einstellen. Neben diesen Attributen gibt es noch eine Menge weiterer Einstellmöglichkeiten, Details finden sich im Manual.

Diese Änderungen hätte man natürlich auch manuell in den Tabellen durchführen können.

Wir werden daher ein wenig Tuning vornehmen:

Wertebereiche erfinden:

```
insert into domsel(item,intern,extern) values('sex','m','männlich');
insert into domsel(item,intern,extern) values('sex','w','weiblich');
insert into domsel(item,intern,extern) values('sex','f','firma');
insert into domsel(item,intern,extern) values('idpapier','Reisepass','Reisepass');
insert into domsel(item,intern,extern) values('idpapier','Führerschein','Führerschein');
```

Und mit den Datenfeldern verbinden:

```
update domfield set selitem = 'sex' where tbl = 'tblkunde' and field = 'geschlecht';
update domfield set selitem = 'idpapier' where tbl = 'tblkunde' and field = 'aut_dok';
```

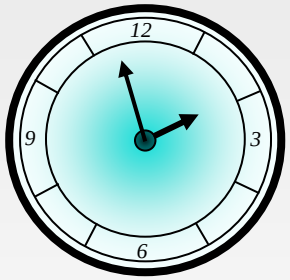
Feldbezeichnungen:

beim Knoten ‘Kunde‘ generell ausblenden:

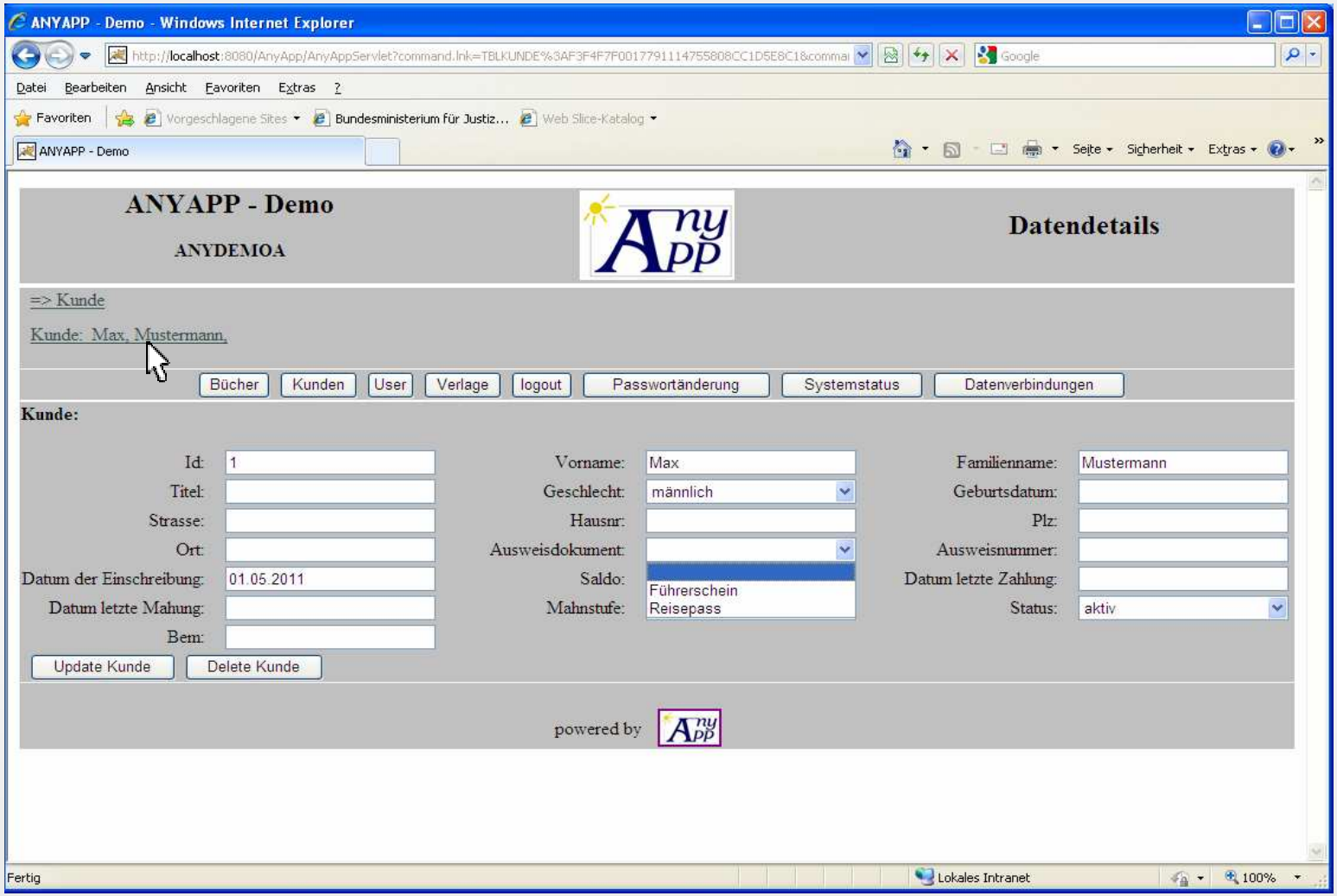
```
update domtbl set bezeichner = 'N' where tbl = 'tblkunde';
```

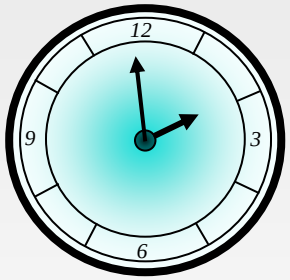
andere ändern:

```
update domfield set bez = 'Familienname' where tbl = 'tblkunde' and field = 'famname';
update domfield set bez = 'Datum der Einschreibung' where tbl = 'tblkunde' and field =
'einsch_dat';
update domfield set bez = 'Mahndatum' where tbl = 'tblkunde' and field = 'datmahn';
update domfield set bez = 'Datum letzte Zahlung' where tbl = 'tblkunde' and field =
'datlzahl';
update domfield set bez = 'Datum letzte Mahung' where tbl = 'tblkunde' and field =
'datlmahn';
update domfield set bez = 'Ausweisdokument' where tbl = 'tblkunde' and field = 'aut_dok';
update domfield set hidedisp = 'Y' where tbl = 'tblkunde' and field = 'anz_mahn';
update domfield set bez = 'Ausweisnummer' where tbl = 'tblkunde' and field = 'doknr';
update domfield set bez = 'Geburtsdatum' where tbl = 'tblkunde' and field = 'gebdat';
update domfield set readonly = 'Y' where tbl = 'tblkunde' and field = 'saldo';
update domfield set readonly = 'Y' where tbl = 'tblkunde' and field = 'mahnsufe';
update domfield set readonly = 'Y' where tbl = 'tblkunde' and field = 'id';
update domfield set readonly = 'Y' where tbl = 'tblbuch' and field = 'id';
```



Refresh des Bildschirms (z.B. via Pfad-Link) bringt:



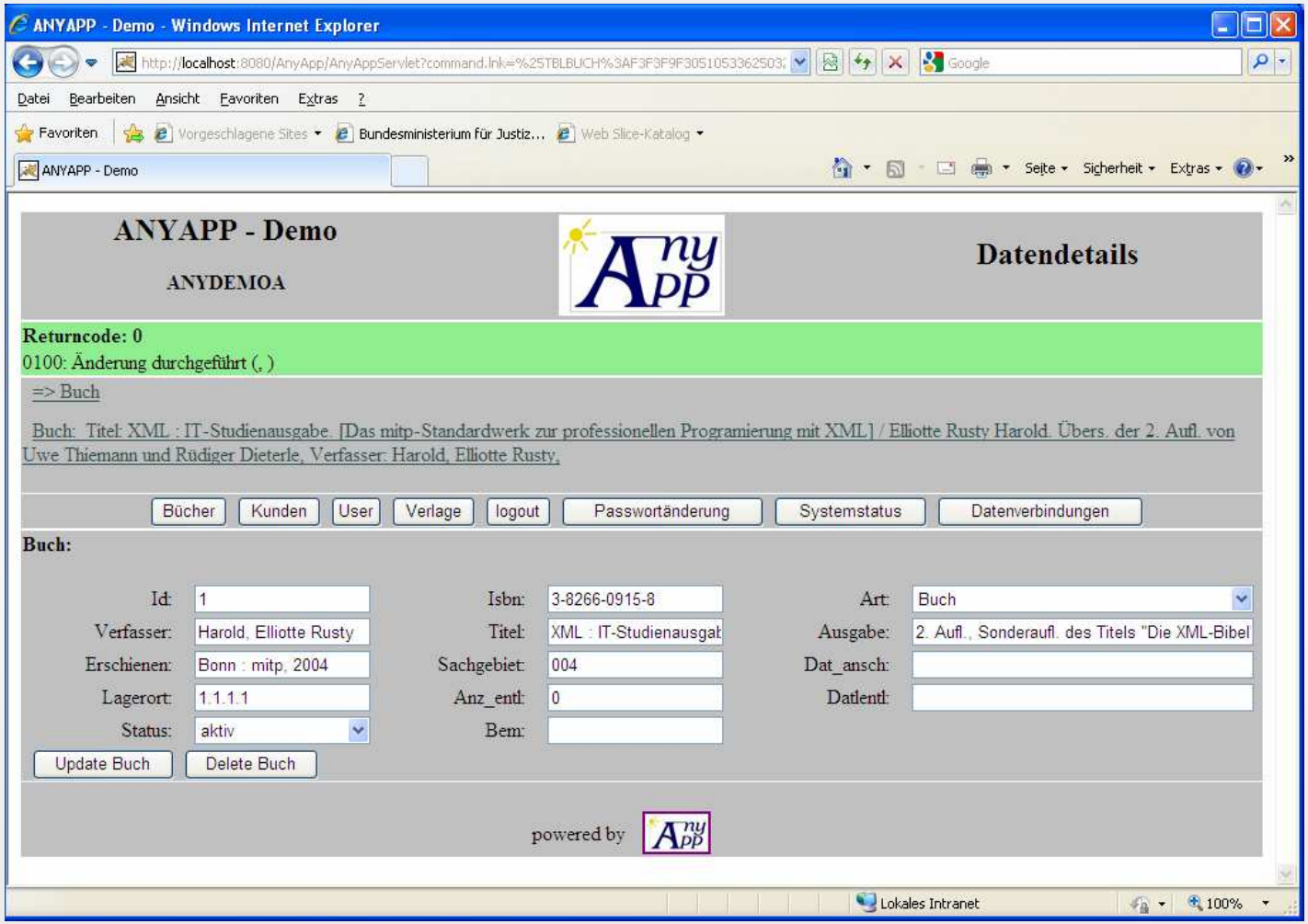


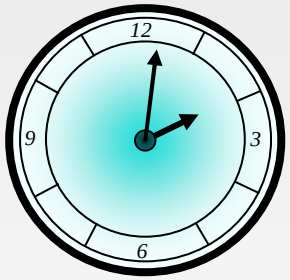
Wir erfassen nun ein Buch (‚Bücher' -> ‚neu')

Schon bemerkt?

Infos werden grün unterlegt, Warnungen gelb und Fehler rot

Die Publikationsart (Art) wurde hier ebenfalls mit Wertebereichen hinterlegt. Weitere Behübschungen müssen nunmehr leider aus Zeitgründen ausbleiben.





Bisher haben wir nur Segmente erfasst, für die auf dem Grundschirm Suchbuttons existierten. Diese Tabellen sind sog. Basistabellen. Deren Existenz ist nicht von einer anderen Tabelle abhängig. Diese Tabellen stehen daher auch niemals als Zieltabellen in has-Relationen (physischer Pfad, Feld TBL2) in der Steuertabelle domrel :

DSS1 - [Results - SPA00.DOMREL - DSS1]

File Edit Edit Data Link View Row Compare Tools Options Window Help

Lock Print Copy Find Edit Link Numbers Group 2-color Height Format Detail Upd Del Dup New Refresh Search SQL Query Builder Pivot Help Abort

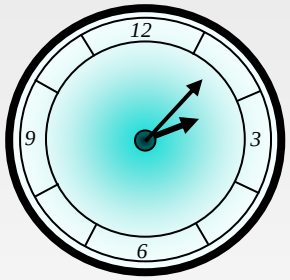
Drag a column header here to group by that column

ID	TBL1	TBL2	REL	REIHUNG	HIST	MAXTYPE	MAXANZ	BED1	BED2	FORWARD	BACKWARD
1	tbluser	tblrolle	has	5	N	S	999	(null)	(null)	Benutzerrollen	User
2	tblkunde	tblausleihung	has	5	Y	S	1	(null)	(null)	Ausleihung	Kunde
3	tblausleihung	tblbuch	use	5	N	S	5	(null)	(null)	Buch	Ausleihung
4	tblbuch	tblverlag	use	5	N	S	32760	(null)	(null)	Verlag	Buch
5	tblkunde	tblvorschreibung	has	5	N	S	9	(null)	(null)	Vorschreibungen	Kunde
6	tblkunde	tblmahnung	has	5	Y	S	1	(null)	(null)	Mahnung	Kunde
7	tblmahnung	tblvorschreibungn	use	5	N	S	1	(null)	(null)	Vorschreibung	Mahnung
8	tblverlag	tblbestellung	has	5	N	S	9	(null)	(null)	Bestellungen	Verlag
9	tblbestellung	tblbuch	use	5	N	S	999	(null)	(null)	Bücher	Bestellung

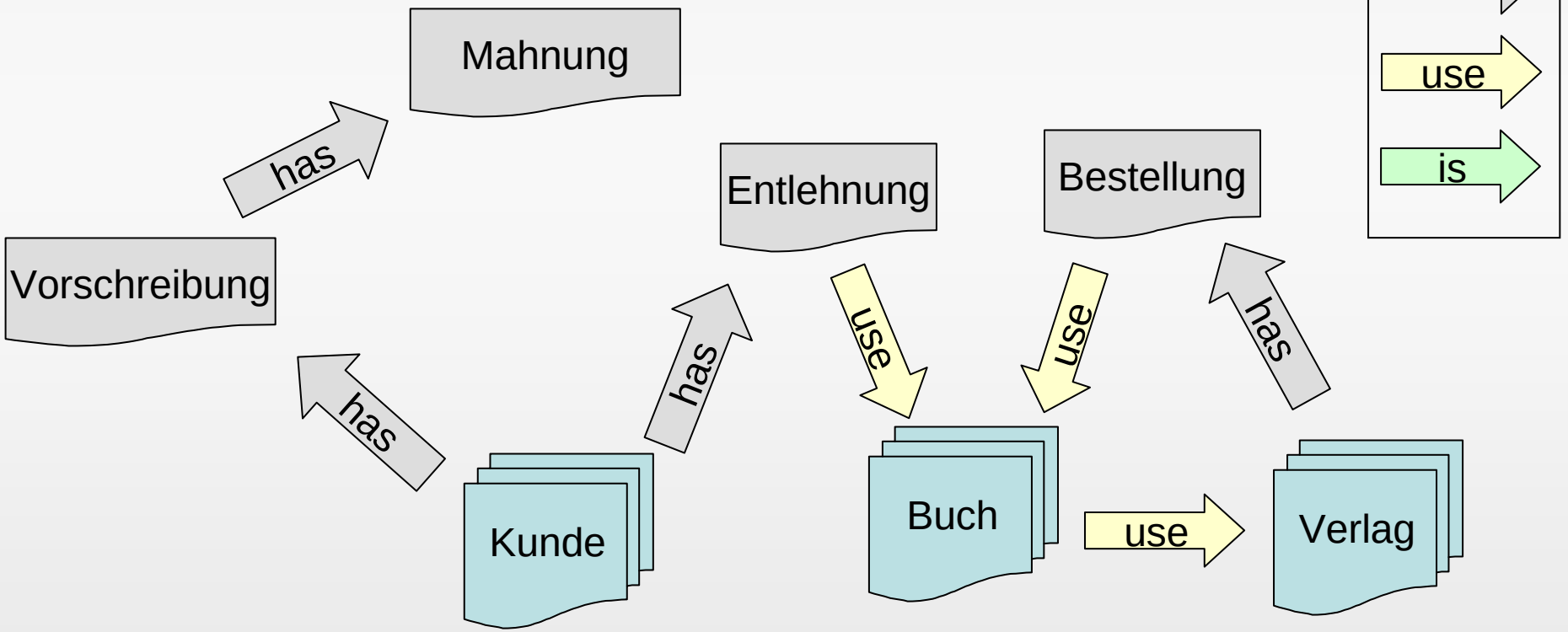
Select * from SPA00.DOMREL | 9 rows | 12 cols | 0,12 secs | 07:18 pm | DSS1

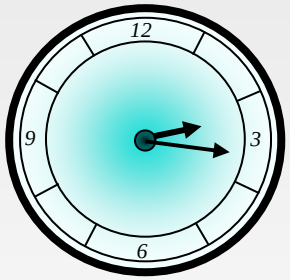
9 rows displayed (0,12 secs)

DSS1 SQL SPA00.DOMFIELD SPA00.DOMREL

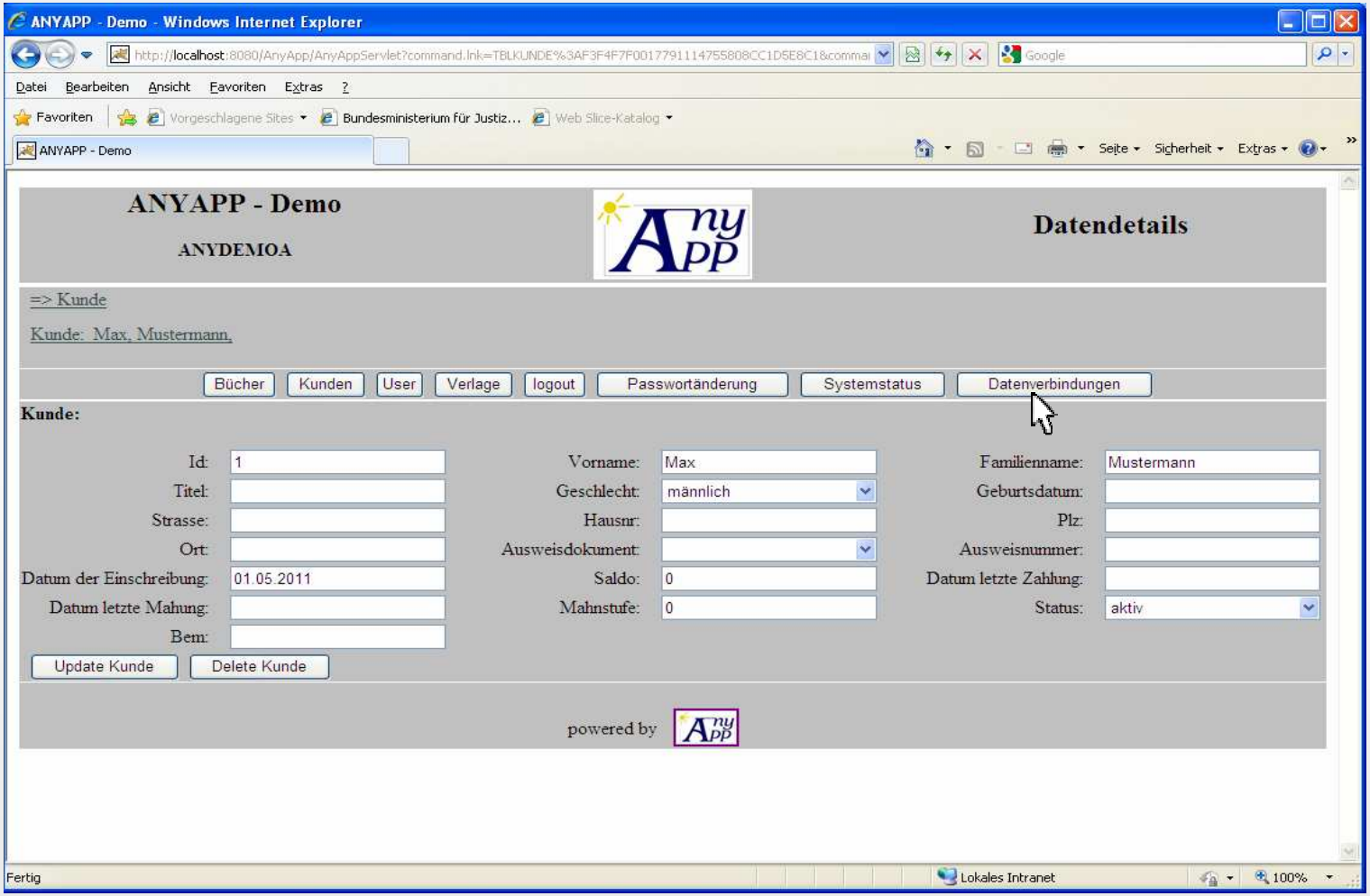


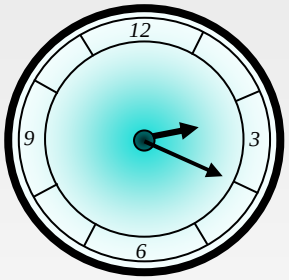
Die Tabelle domrel definiert die Topologie einer Applikation (hier die unserer Applikation ‚Bücherei'). Dieser von uns gewählte Aufbau ist jedoch nur eine von vielen denkbaren Strukturen.





Wir erstellen ein abhängiges Segment zu unserem Testkunden. Dazu klicken wir bei ihm auf ‚Datenverbindungen'

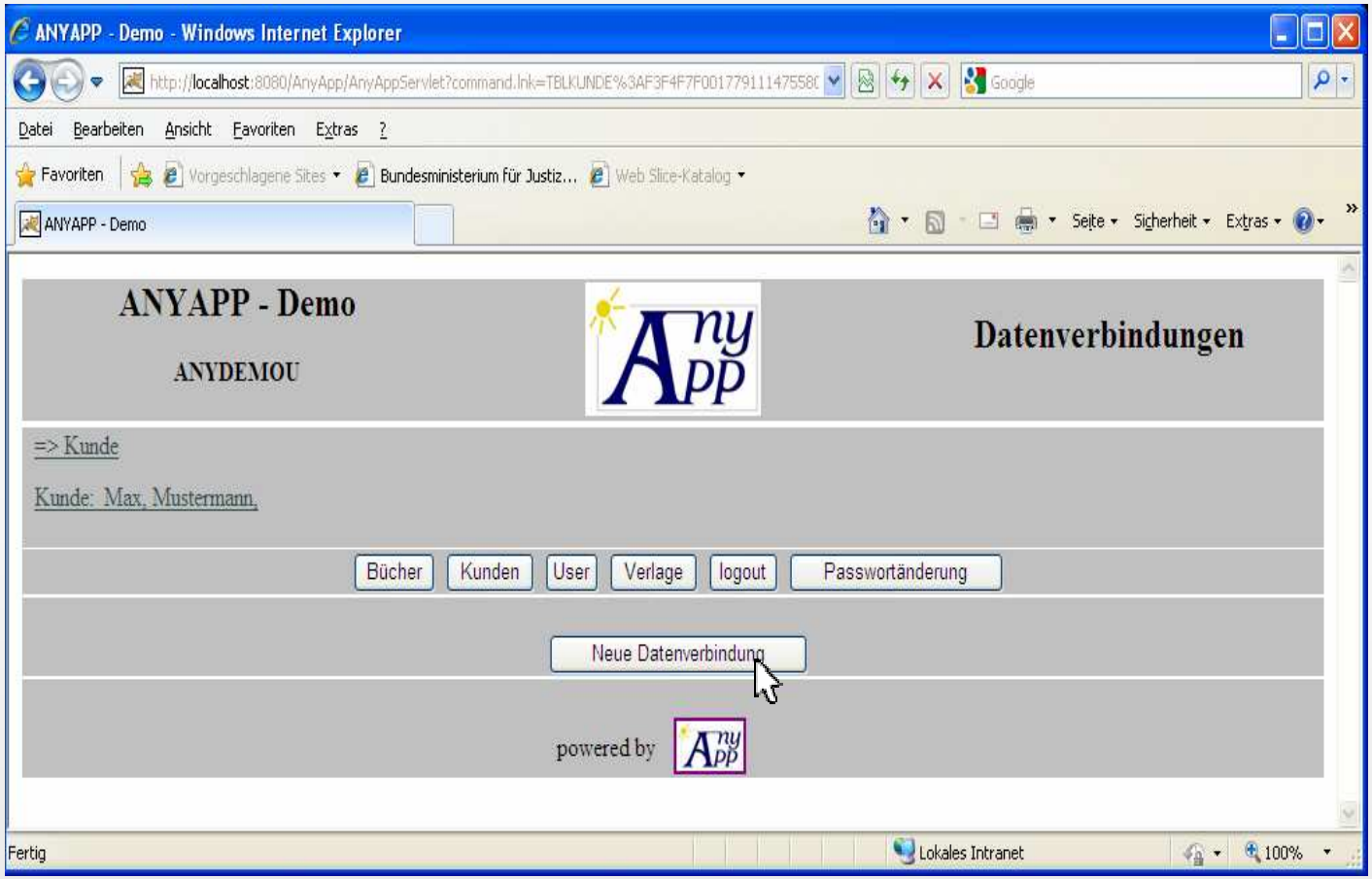


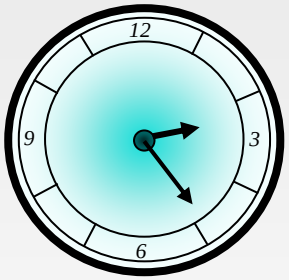


War hier jemand aufmerksam?

Zwischenzeitlich ist die Loginzeit abgelaufen und ich musste mich neu einloggen, nun jedoch als Datenpfleger unter der Update-Sicht ANYDEMOU. Augenscheinlichster Unterschied – unter Update hat man keinen Zugriff auf die System-administration, der Button Systemstatus fehlt.

Wären an diesem Knoten bereits Datenverbindungen angelegt, so würden sie in einer Liste angezeigt. So drücken wir ‚Neue Datenverbindung'

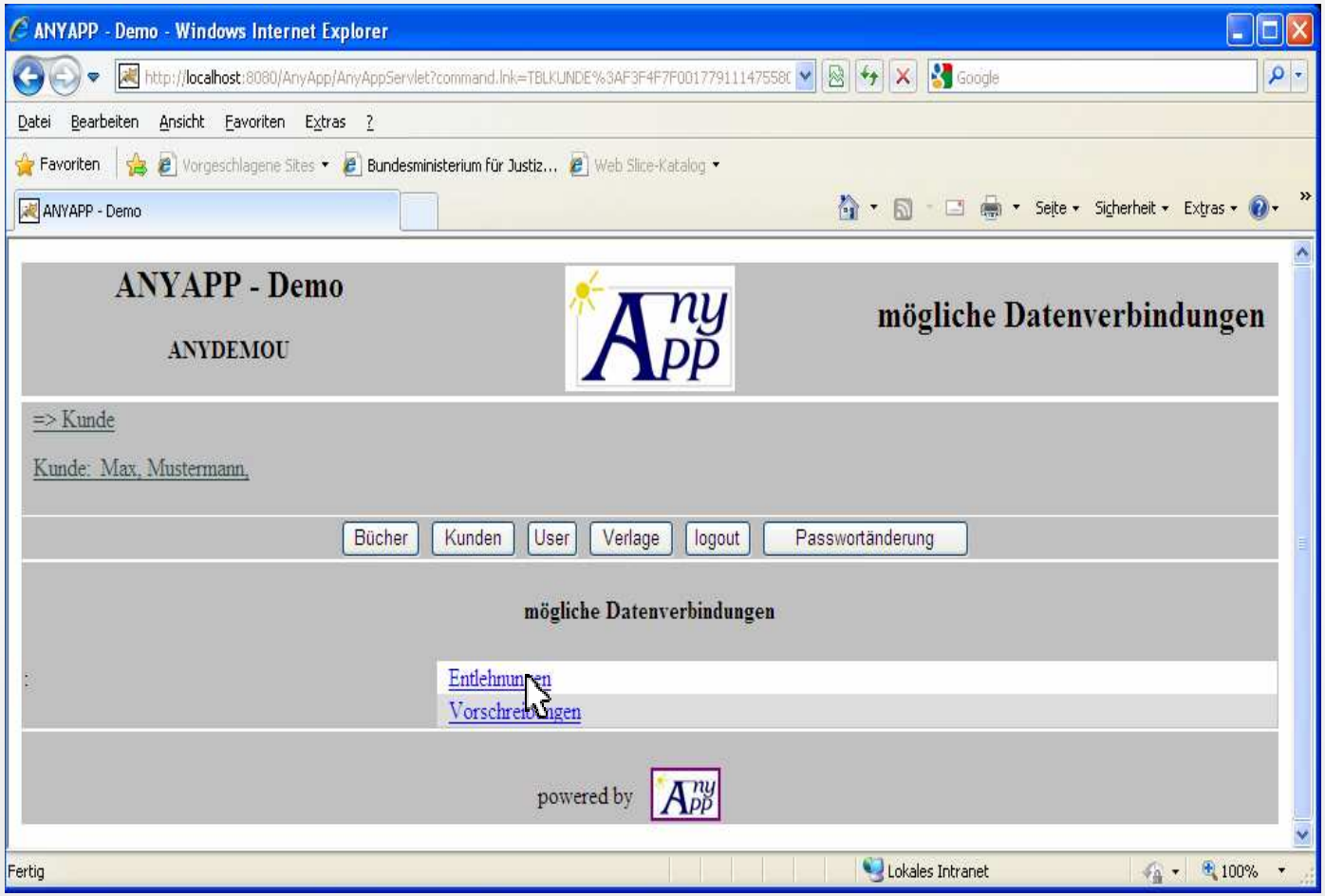


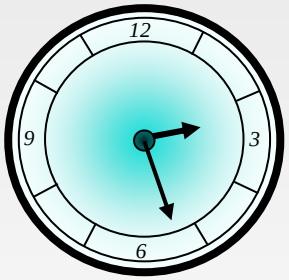


Es öffnet sich eine Liste mit möglichen neuen Verbindungen dieses Knotens wir drücken ‚Entlehnungen‘.

Hier werden alle Eintragungen aus der Tabelle domrel gelistet, mit:

- tbl1 = ‚tblkunde‘,
- Bedingung 1 ist nicht eingegeben oder trifft auf den aktuellen Kunden zu
- maximale Anzahl gleicher aktiver Verbindungen noch nicht erreicht (maxanz).

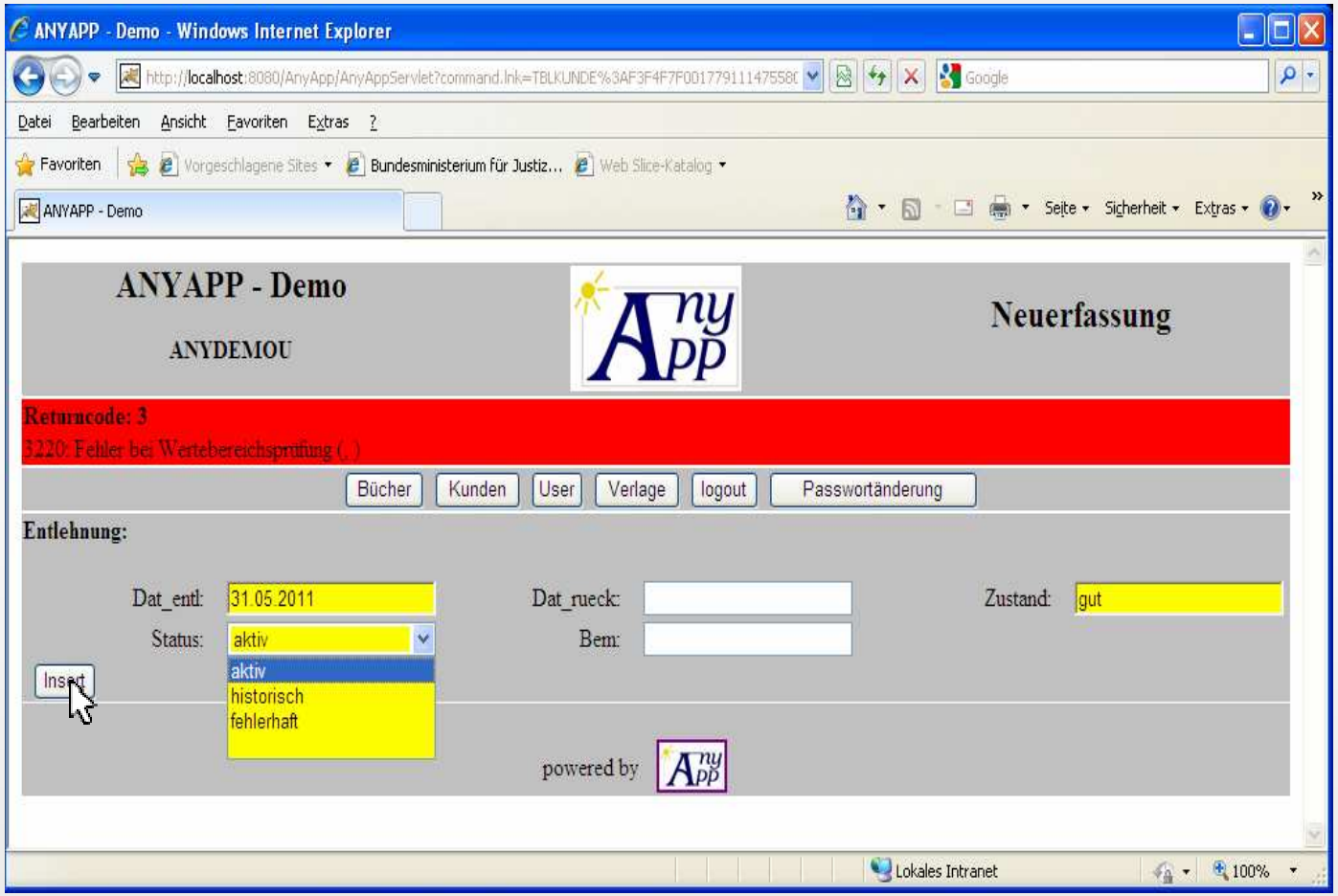


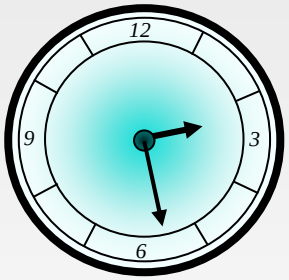


Wir stehen in der Neuerfassung Entlehnung ausfüllen und Feuer …

Anmerkungen:

- Stehen in der domrel Bedingungen 2, so wird das Zielsegment nach diesen vorbefüllt
- In den Systembasis-definitionen kann ein generelles Datums-format für die Bildschirm-darstellung gewählt werden. Hier ist tt.mm.jjjj eingestellt. Das System gibt somit alle ‚Datümer' in diesem Format aus und erwartet so auch alle Eingaben.

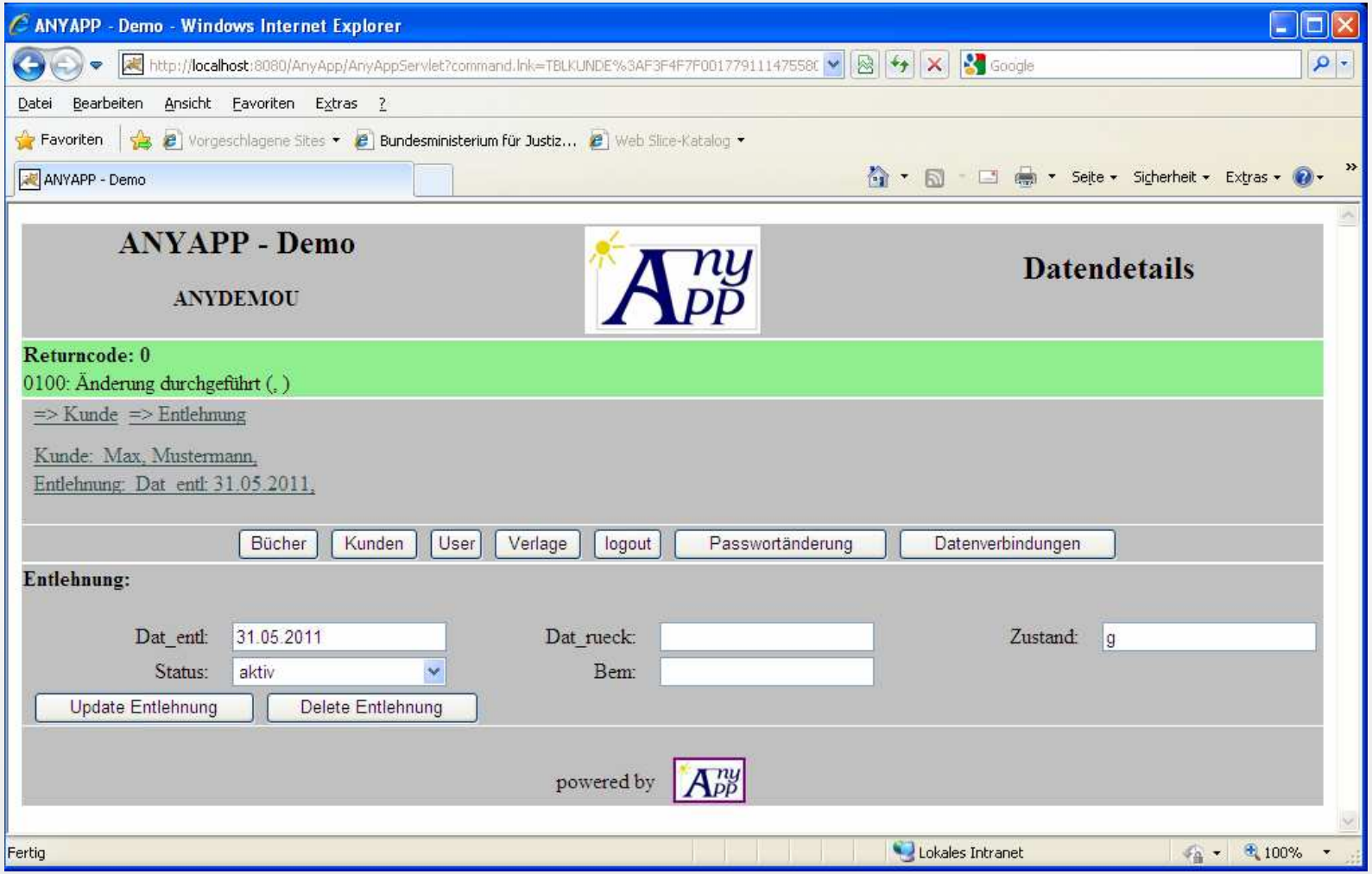


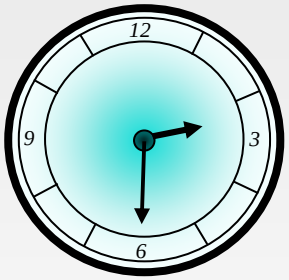


Die Entlehnung wurde erfasst.

Aha:

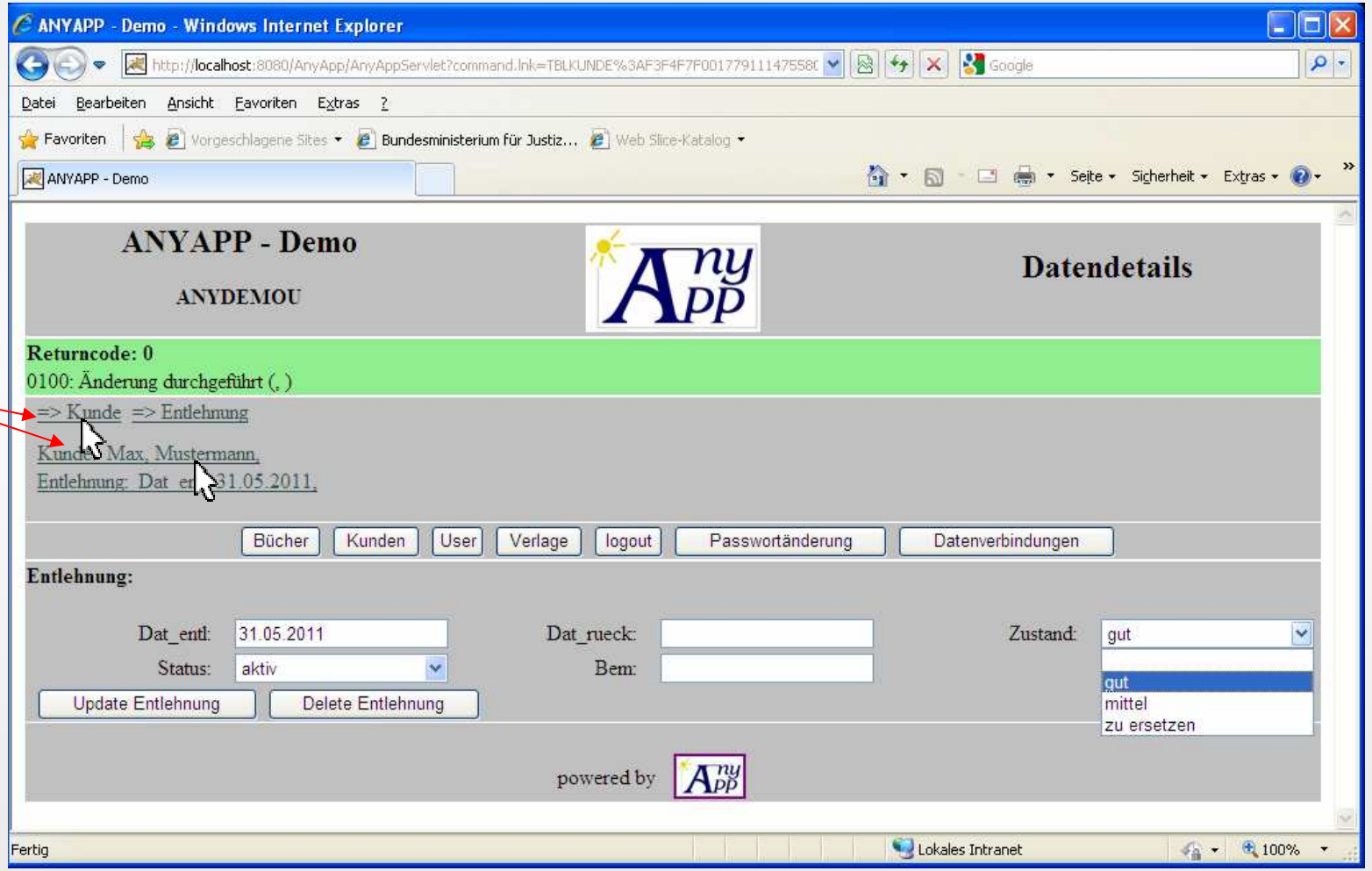
Zustand ‚g‘
Da fehlt noch eine Werte-bereichstabelle (wie sinnvoll ein Zustand bei einer Entlehnung auch sein mag …)

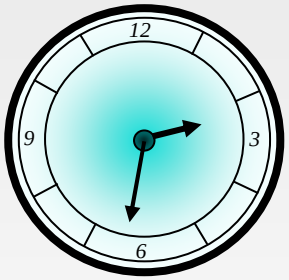




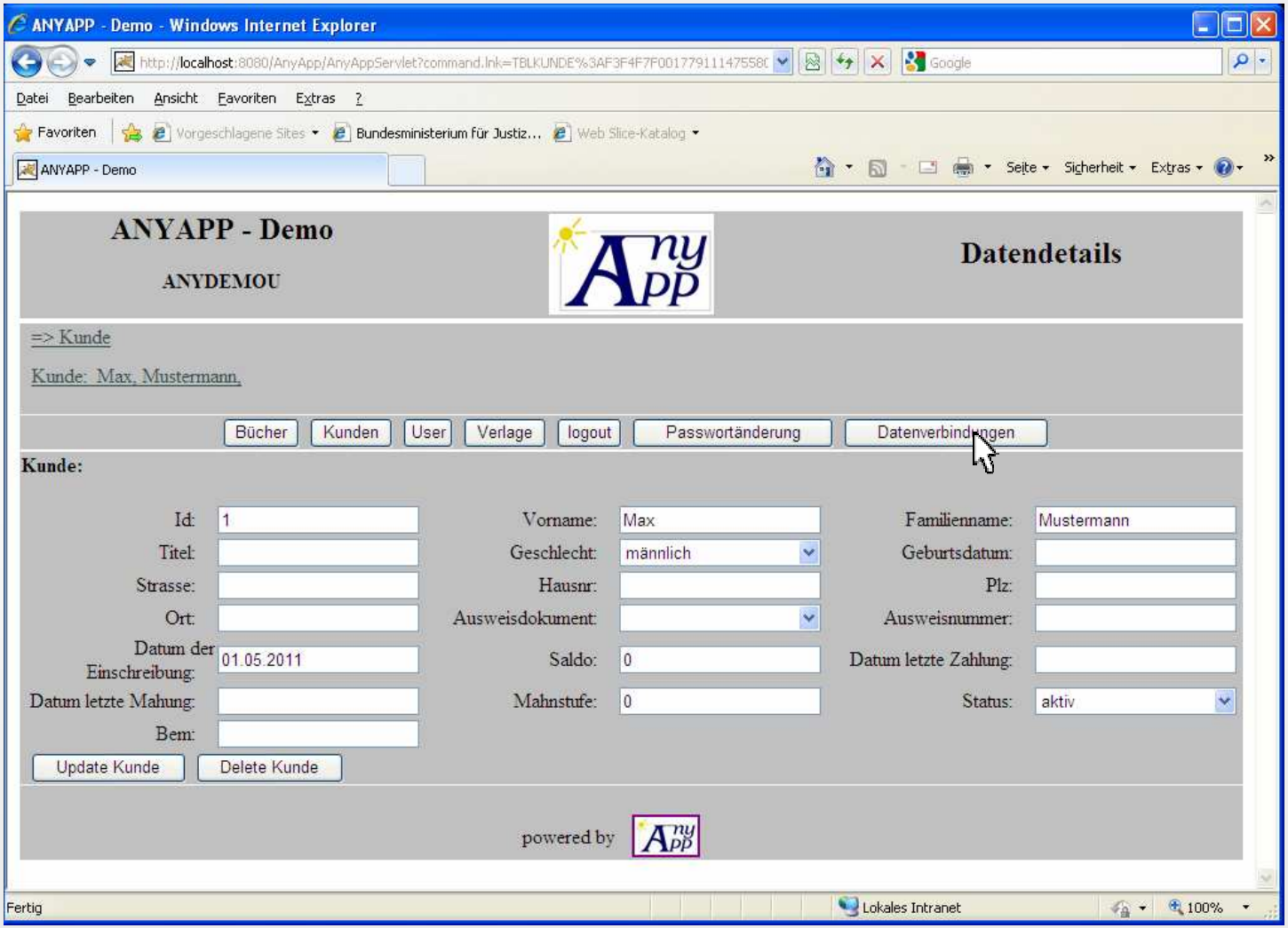
Besser! Nun bitte zur Kontrolle Kunden-Node anclicken

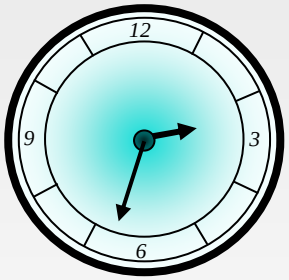
Beachte:
Beide Links führen zum Kunden





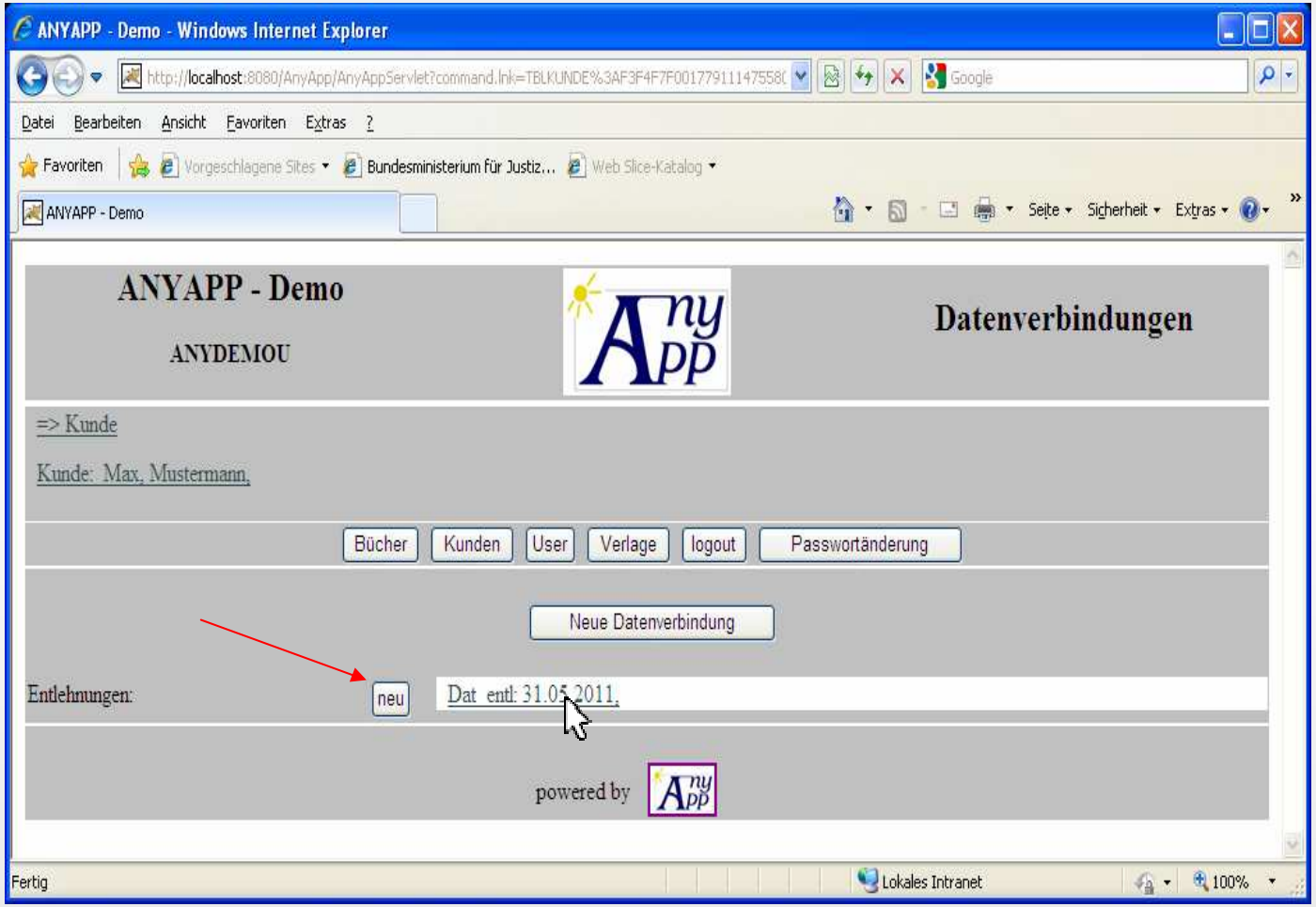
Und hier wieder ‚Datenverbindungen'

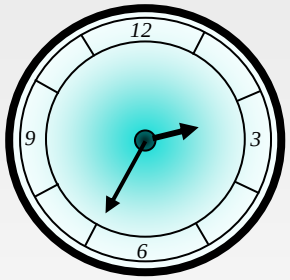




Die neue Ausleihung wird korrekt als Datenverbindung angezeigt

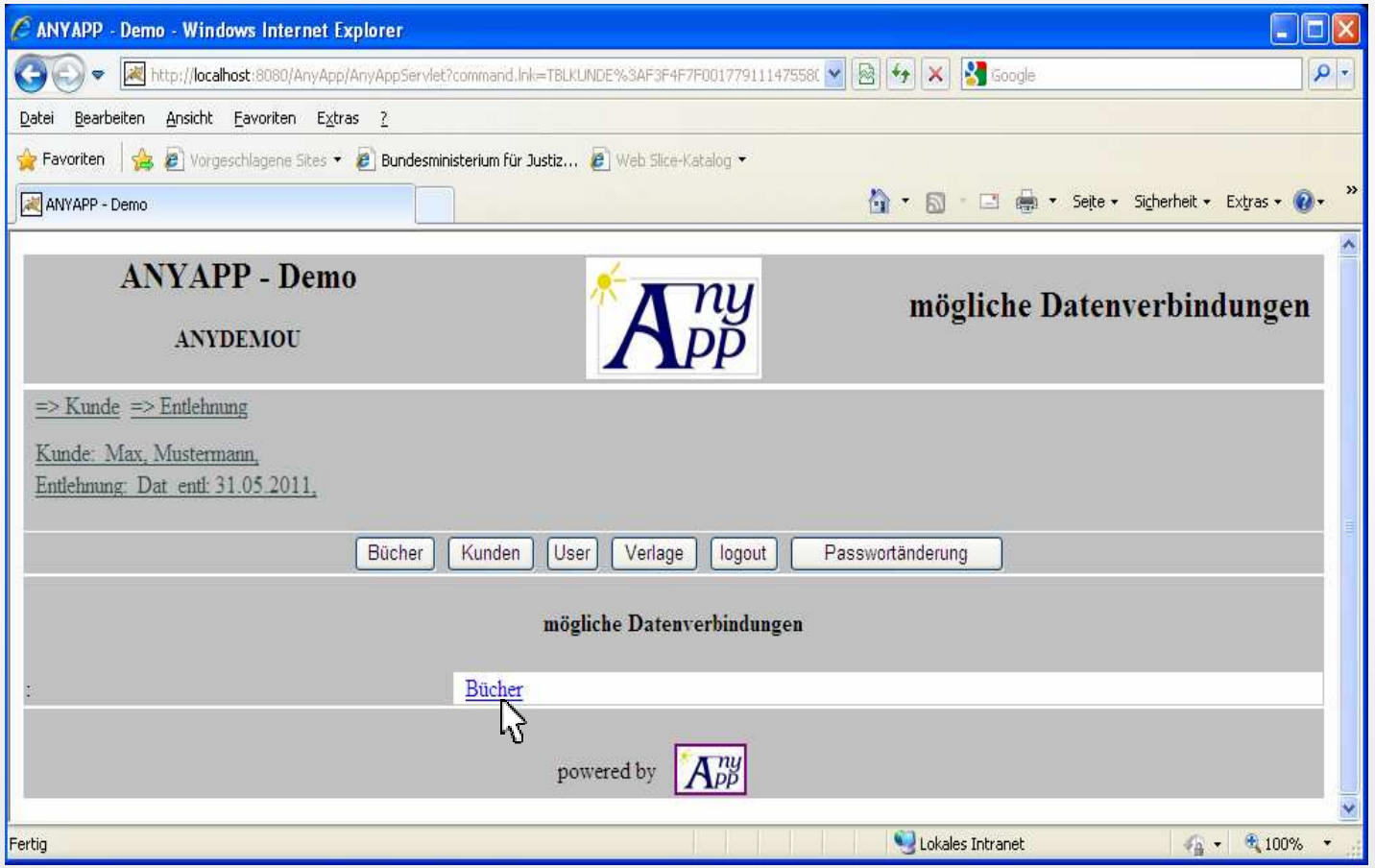
Da steht aber auch ein ‚neu'-Button, bei dieser Verbindung! Das scheint der maxanz von 1 bei dieser Relation lt. domrel zu widersprechen!
Dort haben wir jedoch bei HIST ein ‚Y' eingetragen, was soviel heißt wie ‚historisierend'. Bei Insert einer neuen gleichartigen Segmentverbindung wird die alte automatisch auf historisch gesetzt. Es bleibt somit bei nur einer aktiven Verbindung und die Neueingabe ist OK.

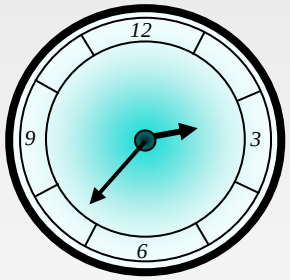




Nun muss die Entlehnung aber auf Bücher verweisen. Wir gehen zur Entlehnung und drücken dort ‚Datenverbindungen' und dann ‚Neue Datenverbindungen'

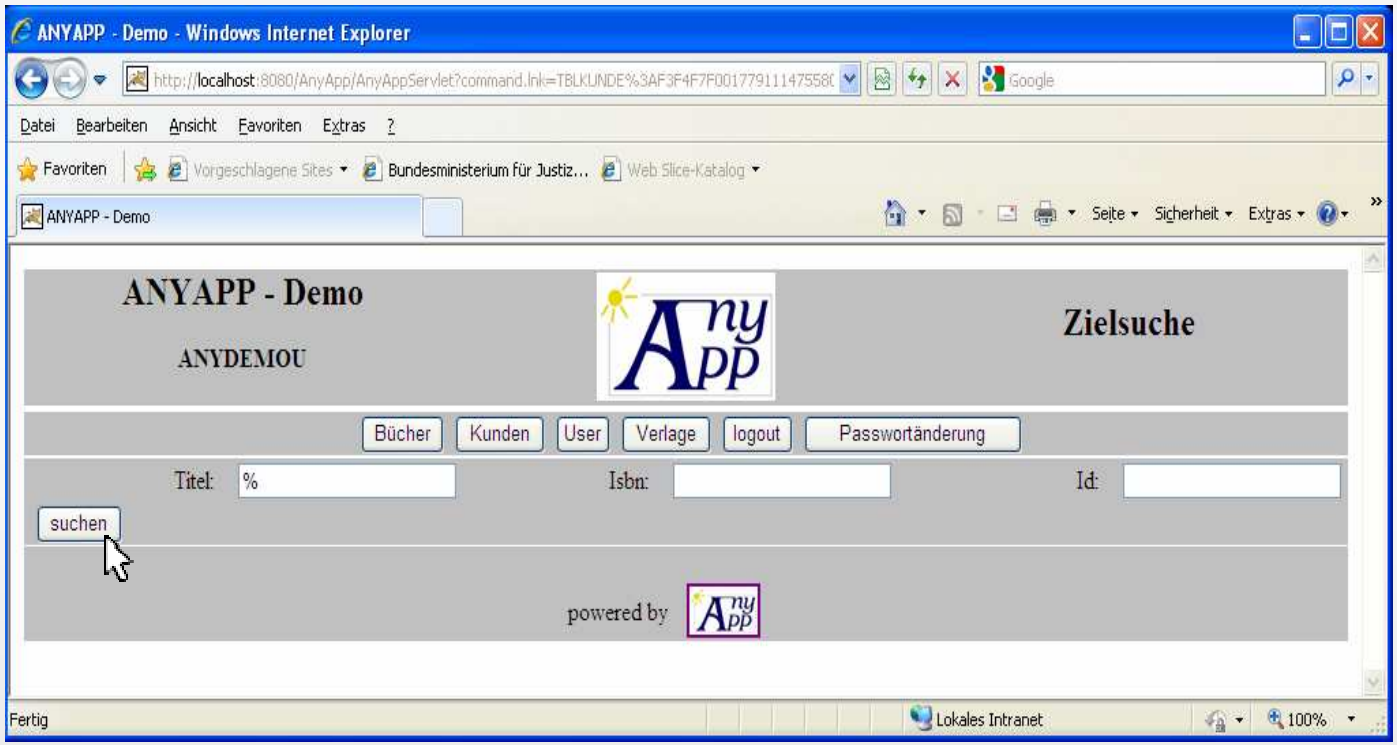
Als einzige mögliche weitere Verbindung in diesem Kontext wird ein Buch angezeigt. Wir aktivieren den Link.

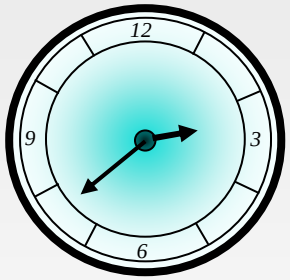




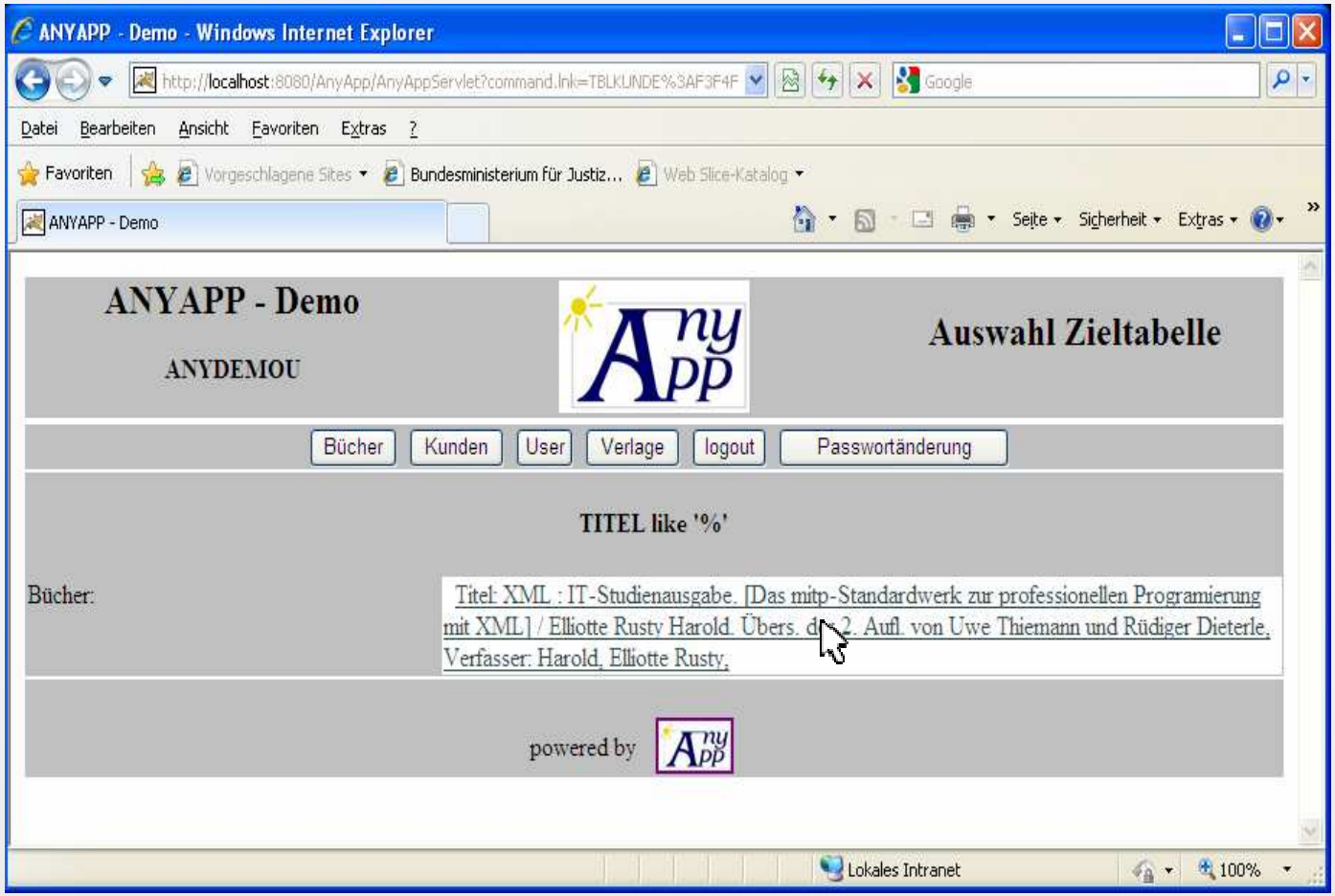
Da es sich bei der Verbindung ‚Entlehnung zu Buch' um eine ‚use'-Verbindung handelt, muss das Zielelement erst gefunden werden. Es öffnet sich ein Suchschirm.

Auch hier die Standardsuche mit Wildcards. Da eine Suche nur eine beschränkte Azahl von Resultaten listet (Default 100, kann in den Systemeinstellungen geändert werden), sind die Anfragen mit anwachsender Datenbasis zu konkretisieren



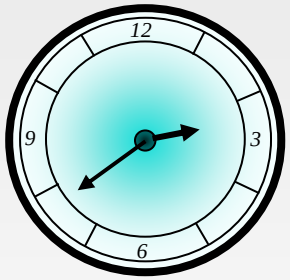


Wir finden unser bisher einziges Buch. Um eine Verbindung herzustellen ist die Eintragung in der Liste anzuklicken

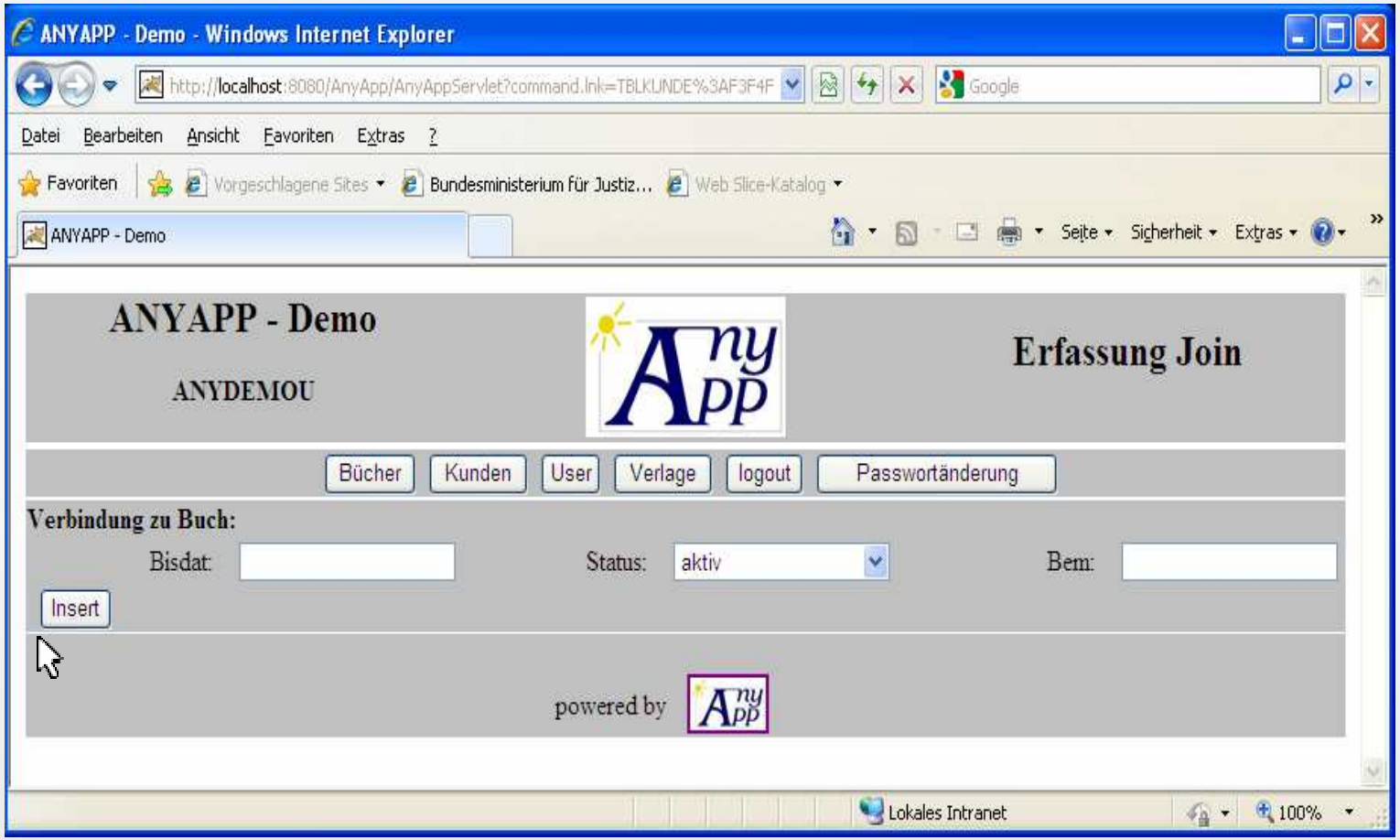


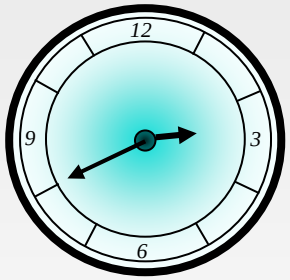
Anmerkung:

Stünden in der domrel bei dieser Verbindung Bedingungen, so würden nur Sätze gelistet, die diese Bedingungen auch erfüllen



Wir stehen beim Service Erfassung Join (Verbindung).



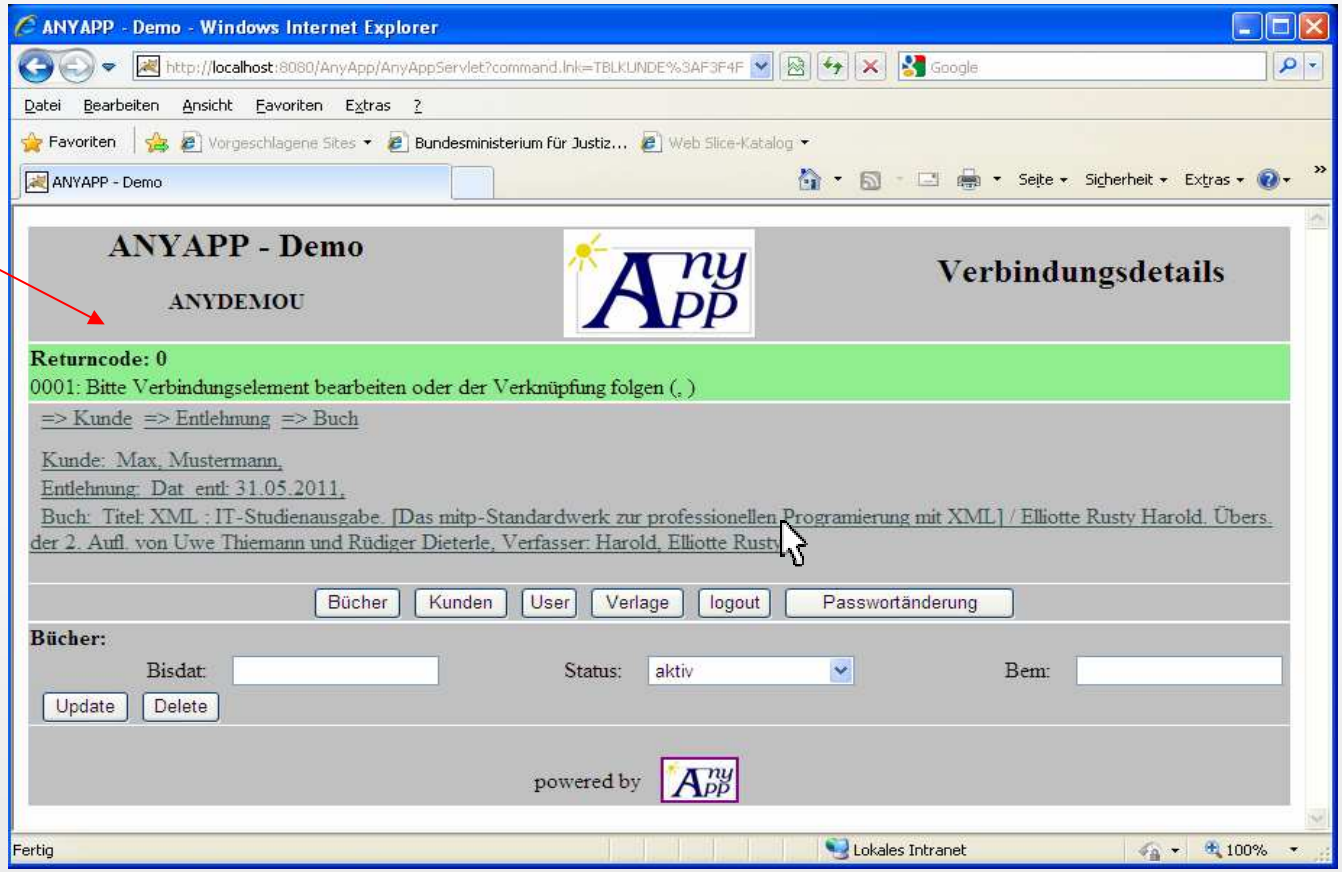


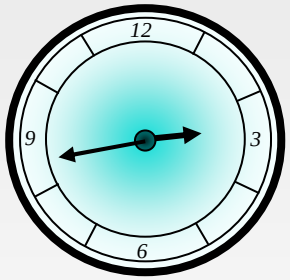
Der Klick auf Insert stellt die Verbindung dauerhaft her. Das Display steht jedoch weiterhin auf dem Verbindungs-element. Um zum verbundenen Zielelement zu kommen ist dieses in den Pfad-Links explizit anzuklicken

Bitte beachten Sie den Hinweis.

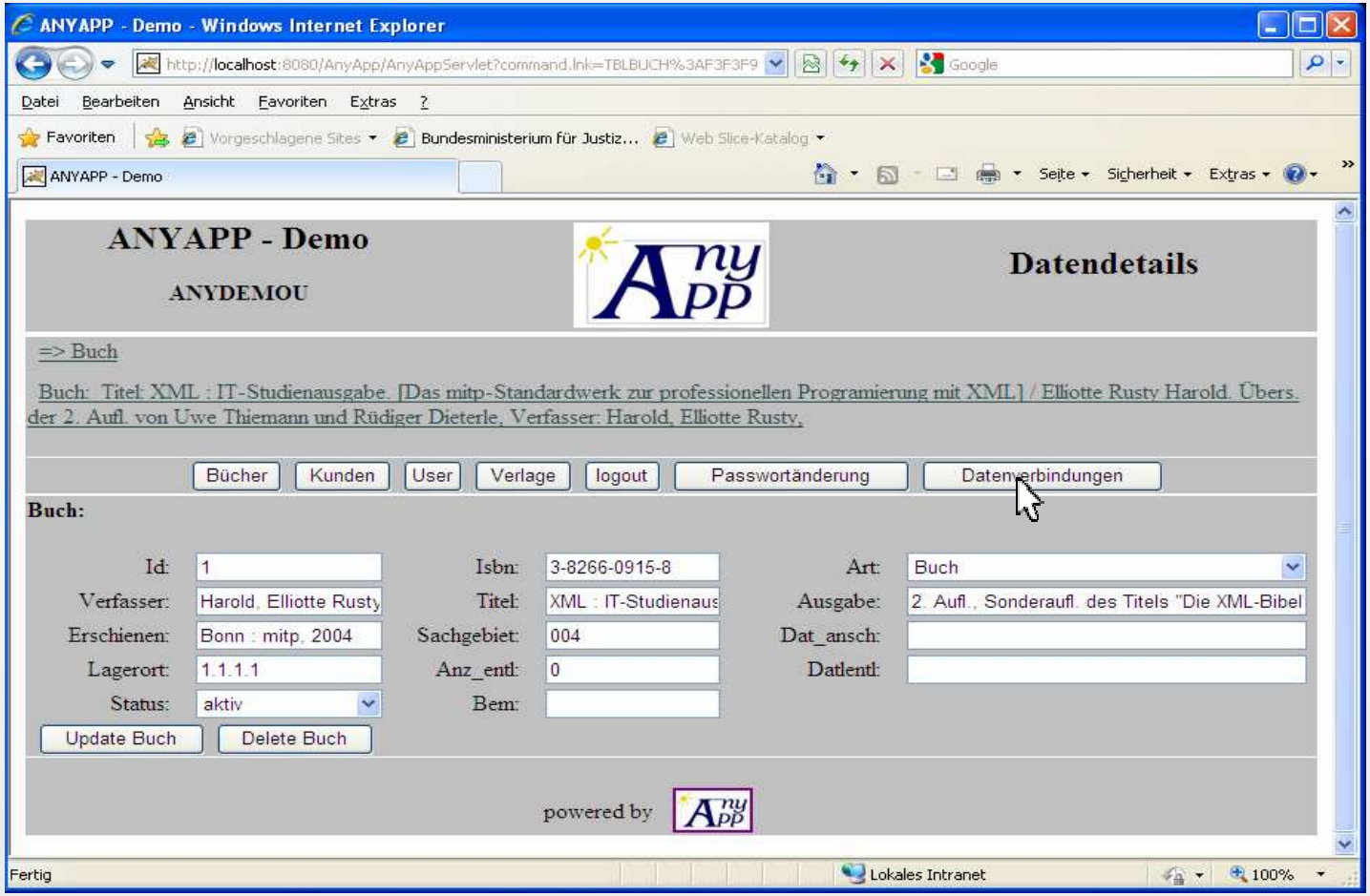
Bei logischen use- und is- Verbindungen muss die Verbindung im Update-Modus wartbar bleiben und der Service verzweigt daher in die Detailanzeige.

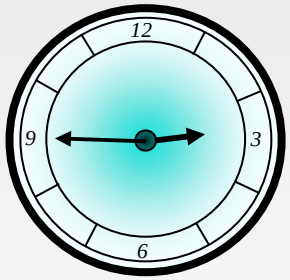
Physische Verbindungs-segmente (has-Relationen) zwischen den Knoten werden nicht angezeigt. Bei Klick auf eine Verbindung wird direkt zu dessen Endknoten verzweigt.



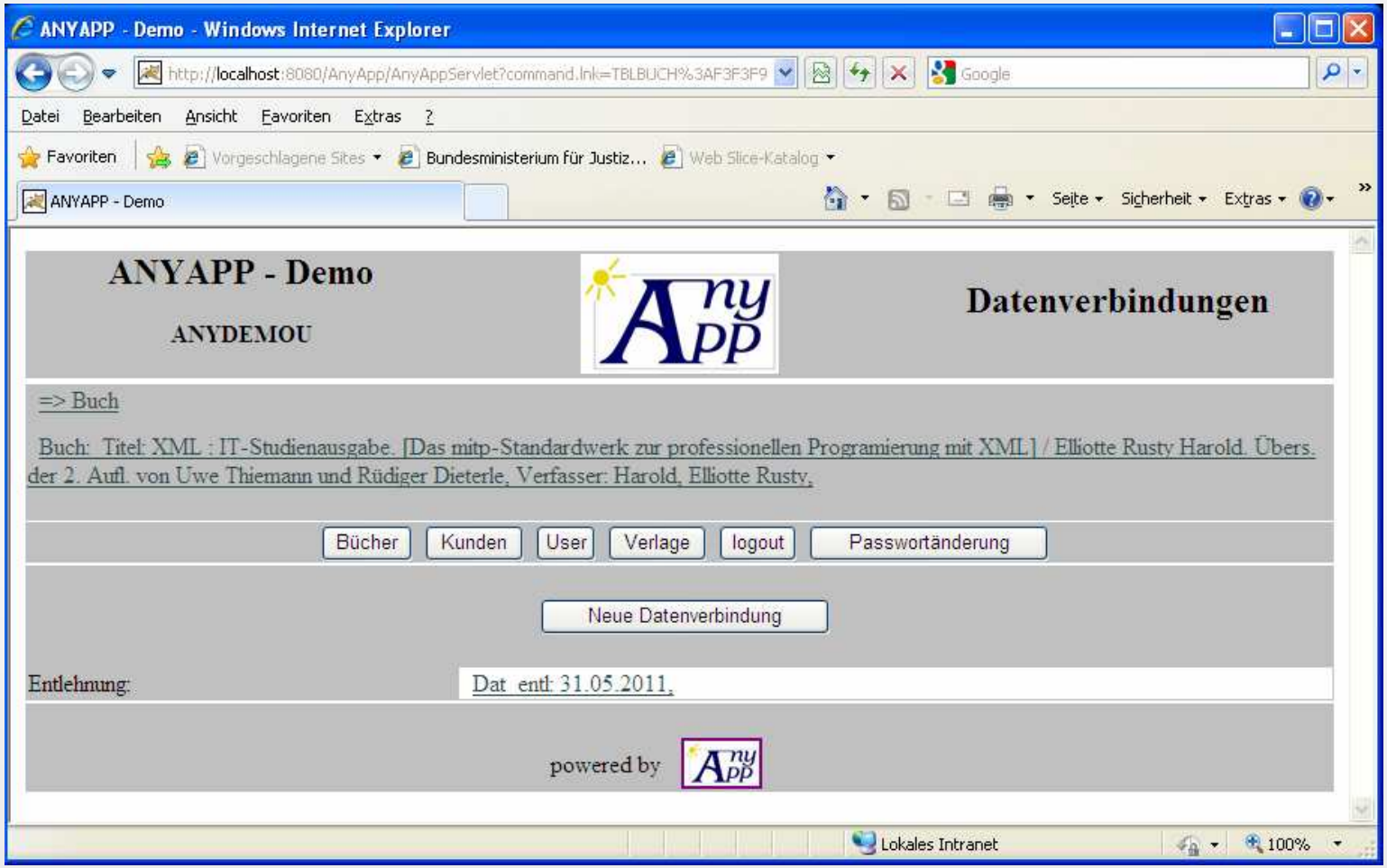


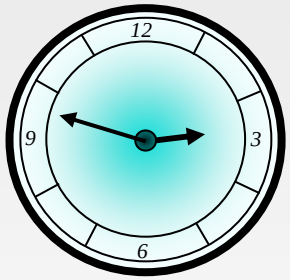
Nun kann man gleich sehen, wie diese Verbindung aus der Sicht des Zielements aussieht. Dazu bitte bei diesem auf Datenverbindungen‘ drücken.





Wir sehen die Rückwärtsverkettung zur Entlehnung.

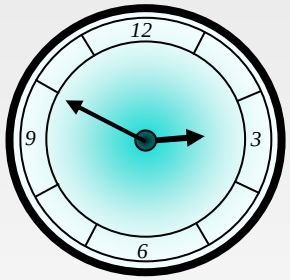




Nun braucht unser System noch ein wenig Businesslogik

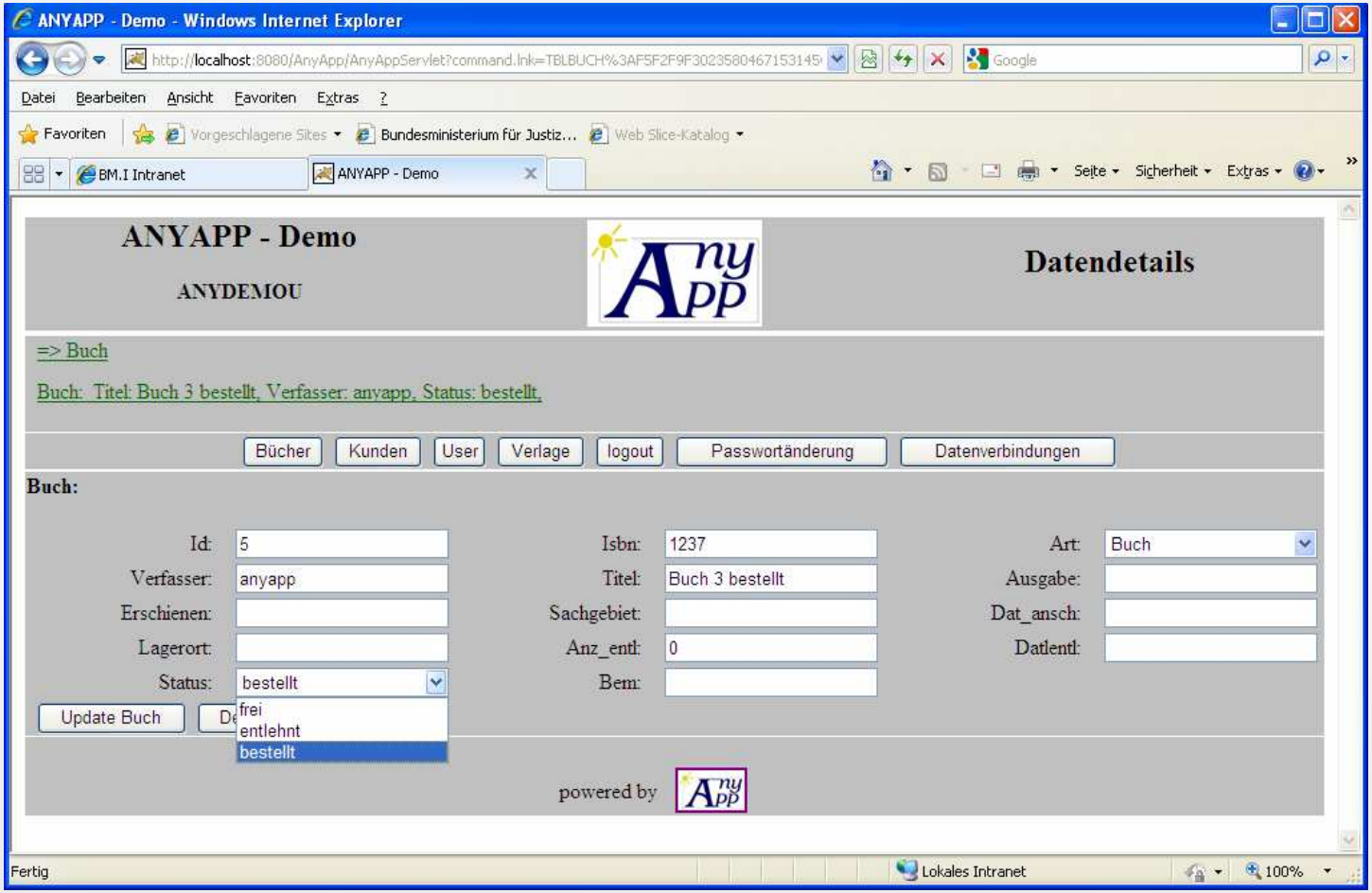
- Wir erweitern die Status-Domäne (domstatus) bei der Buchtabelle (tblbuch) um die Stati
  - •Status ‚1501' – ‚entlehnt' (1xxx ist Warnung)
  - •Status ‚1502' – ‚bestellt'
  - •Sowie ‚0000' – ‚frei', was hier den Standard ‚aktiv' ersetzt
- Und setzen die Bedingung 2 in der domrel bei der use-Verbindung Ausleihung -> Buch auf status=frei, wodurch nur noch freie Bücher, die den Suchkriterien entsprechen, in die Verknüpfungsauswahl kommen

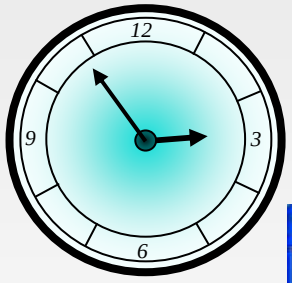
```
Insert into domstatus(tbl,status,color,bez)
       values(,tblbuch','0000''darkslategray','frei');
Insert into domstatus(tbl,status,color,bez)
       values(,tblbuch','1501','gray','entlehnt');
Insert into domstatus(tbl,status,color,bez)
       values(,tblbuch','1502','darkgreen','bestellt');
Update domrel set bed2 = 'status= ''0000''' where id = 3;
Update domrel set bed2 = 'status= ''1502''' where id = 7;
```



Zur Demonstration erfassen wir noch drei Bücher, eines frei, eines bestellt, eines entlehnt.

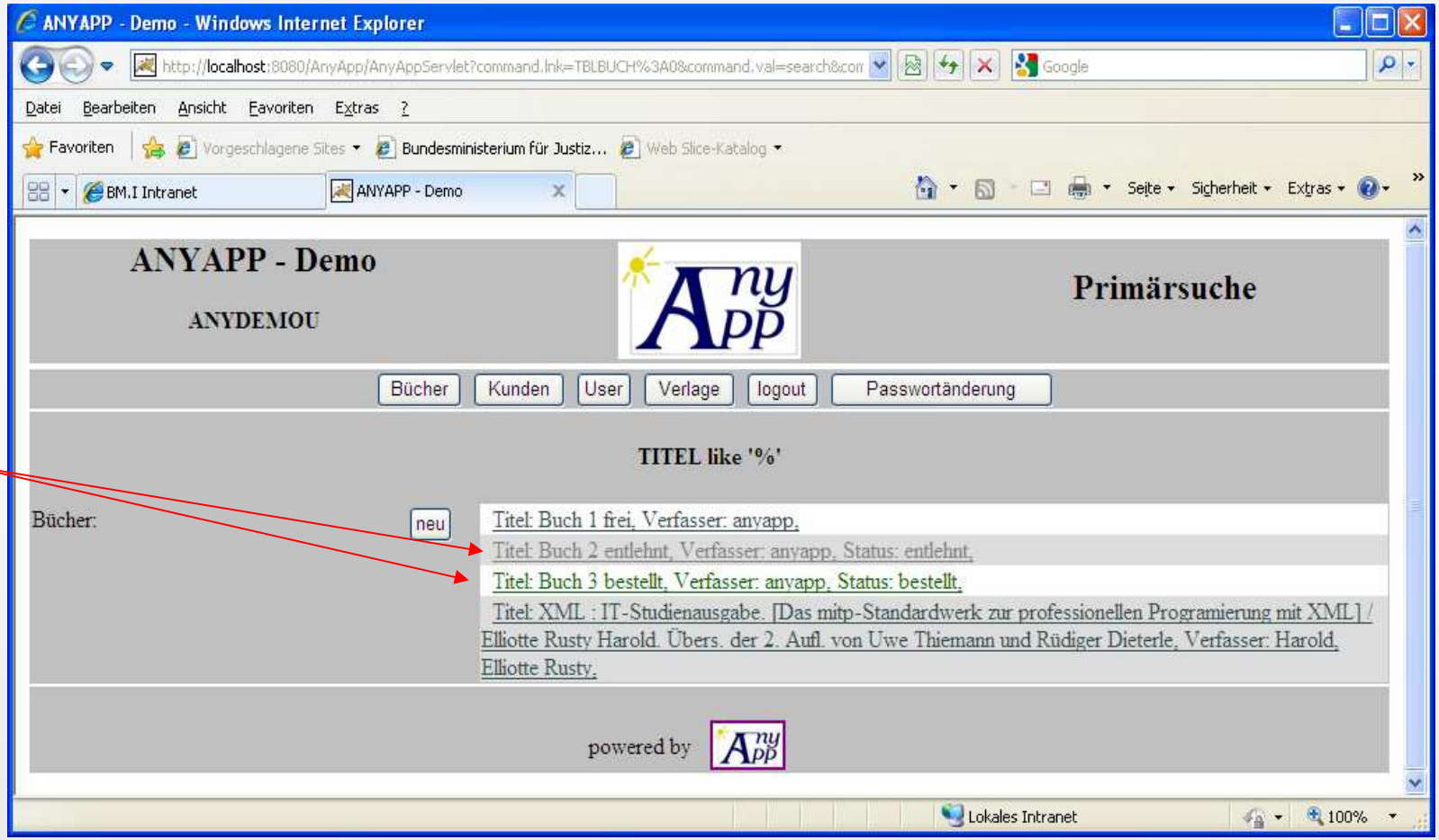
Beachte die neuen Wahlmöglichkeiten im Statusfeld

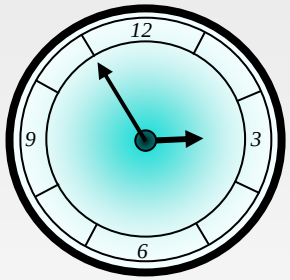




Unser aktueller Buchbestand

Wir hatten den Stati Farben zugeordnet

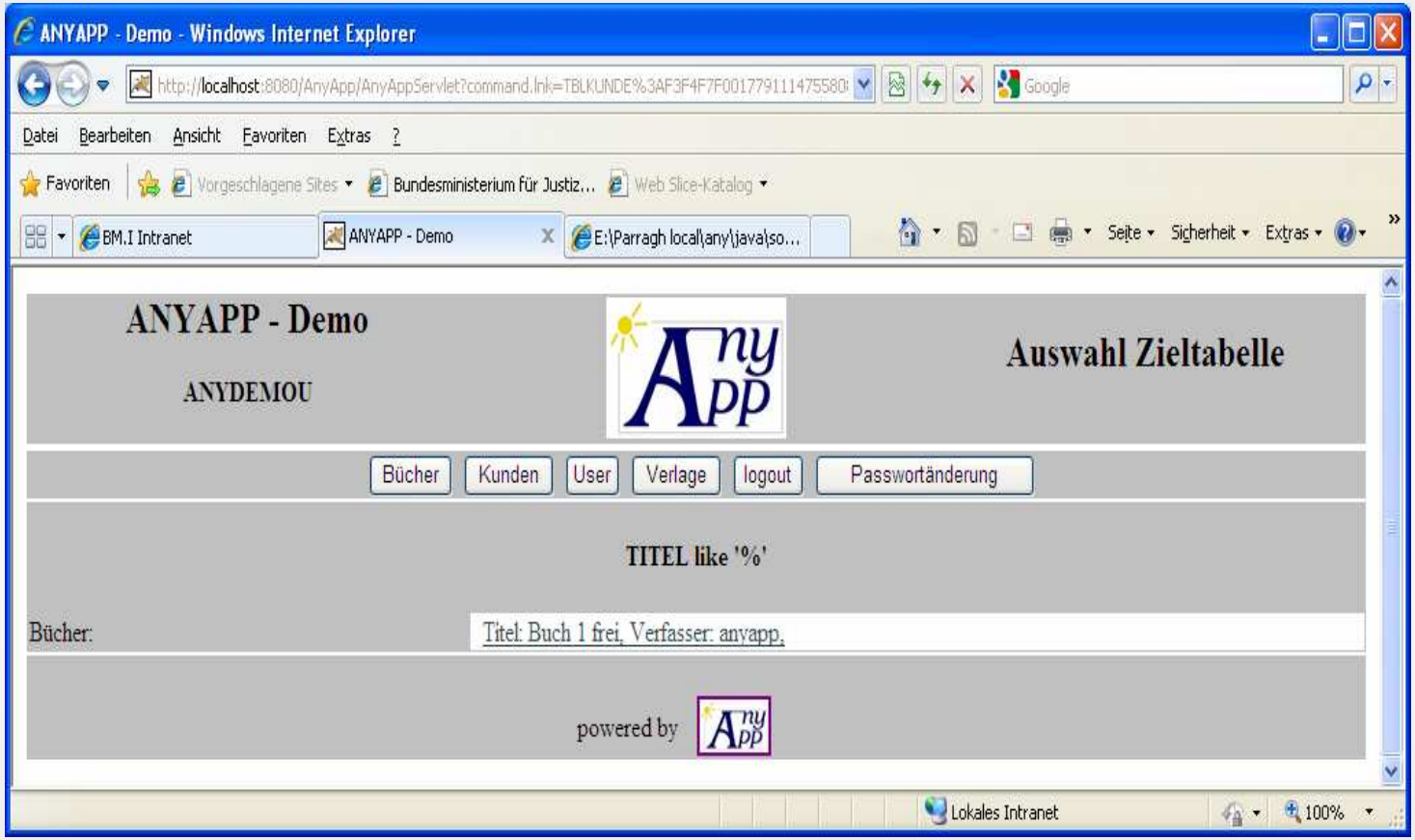




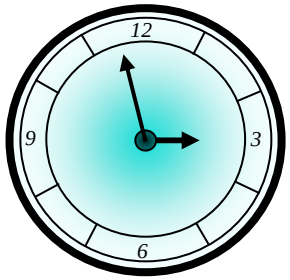
Wir entlehnen noch einmal.

Nur das freie Buch ‚Buch 1' wird zur Entlehnung angeboten.

Merke: auch unser XML-Buch wird nicht gelistet, obwohl es noch auf Status ‚frei' steht, weil es bereits an dieser Stelle verbunden ist



Mit der sorgfältigen Pflege dieser Standardregeln lassen sich ohne Programmieraufwand bereits die meisten nötigen Prüfungen bezüglich Daten- und Strukturintegrität abdecken.



knapp 2 Stunden ‚Entwicklung'

- und läuft …

AnyApp – genial einfach

Individueller Frontend

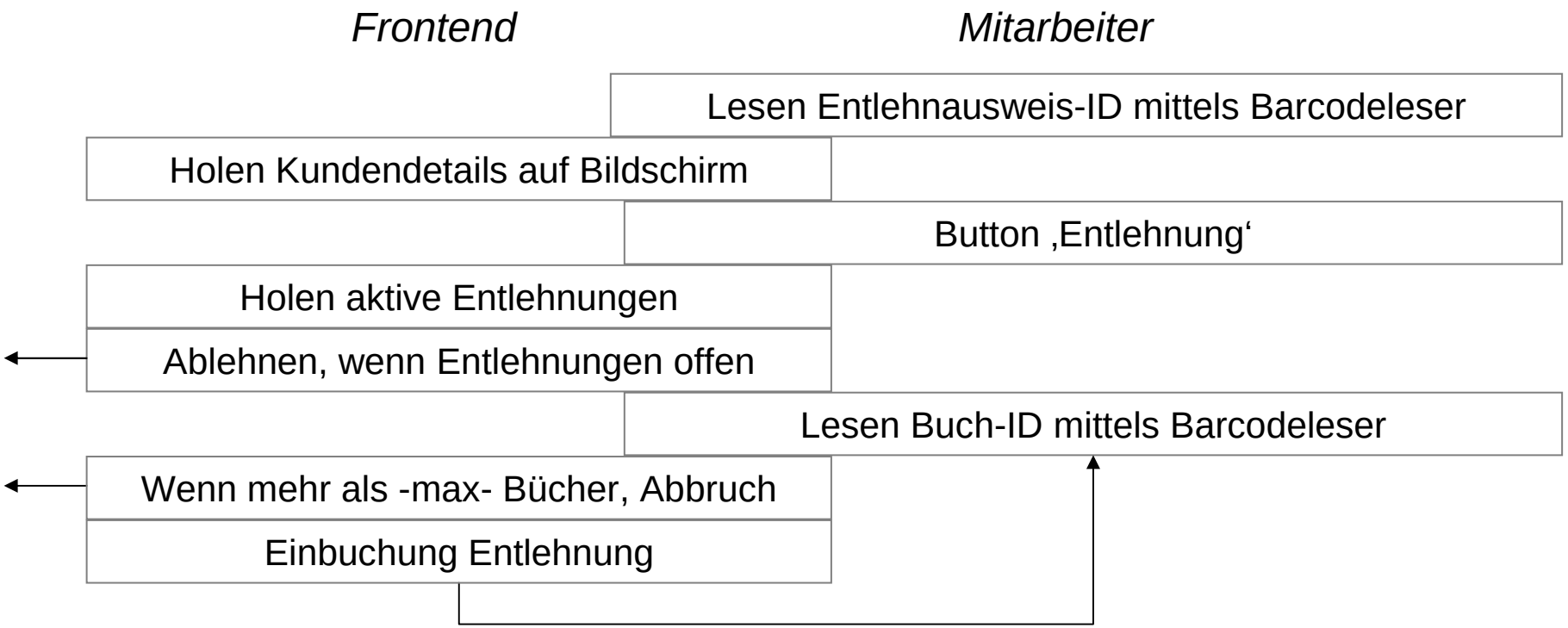
Mit Hilfe der Java-Standardoberfläche kann jeder Dateneintrag einer unter **AnyApp** entwickelten Applikation eingesehen, analysiert und gewartet werden. Die Oberfläche ist daher ein unverzichtbares Instrument zur Sicherstellung des korrekten Datenmaterials und stimmiger Applikationsregeln.

Als alleinige Applikationsoberfläche ist sie nicht gedacht und wird diesem Anspruch auch nur in Ausnahmefällen genügen. Am Beispiel unserer Demoapplikation lässt sich das gut zeigen. Um z.B. die Entlehnung eines Buches über die Standardoberfläche zu erfassen sind 12(!) Verfahrensschritte nötig:

- Anfordern Suchschirm Person
- Suche Person
- Auswahl Person
- Anwahl ‚Neue Datenverbindung'
- Auswahl Entlehnung
- Ausfüllen Entlehnschirm und speichern
- Anwahl ‚Neue Datenverbindung'
- Anfordern Suchschirm Buch
- Suche Buch
- Auswahl Buch
- Speichern Join
- Update Status bei Buch auf ‚entlehnt'

Dieses Verfahren ist für einen Echtbetrieb eher weniger geeignet.

Der **AnyApp**-Kernel kommuniziert mit jedem Frontend und das ausschließlich über XML. Ein applikationsspezifischer Frontend braucht daher nur zu ‚wissen', was der Businessprozess fordert und die nötigen SOA-Requests zusammenstellen und senden. Der Entlehnungsprozess sollte eher so aussehen:



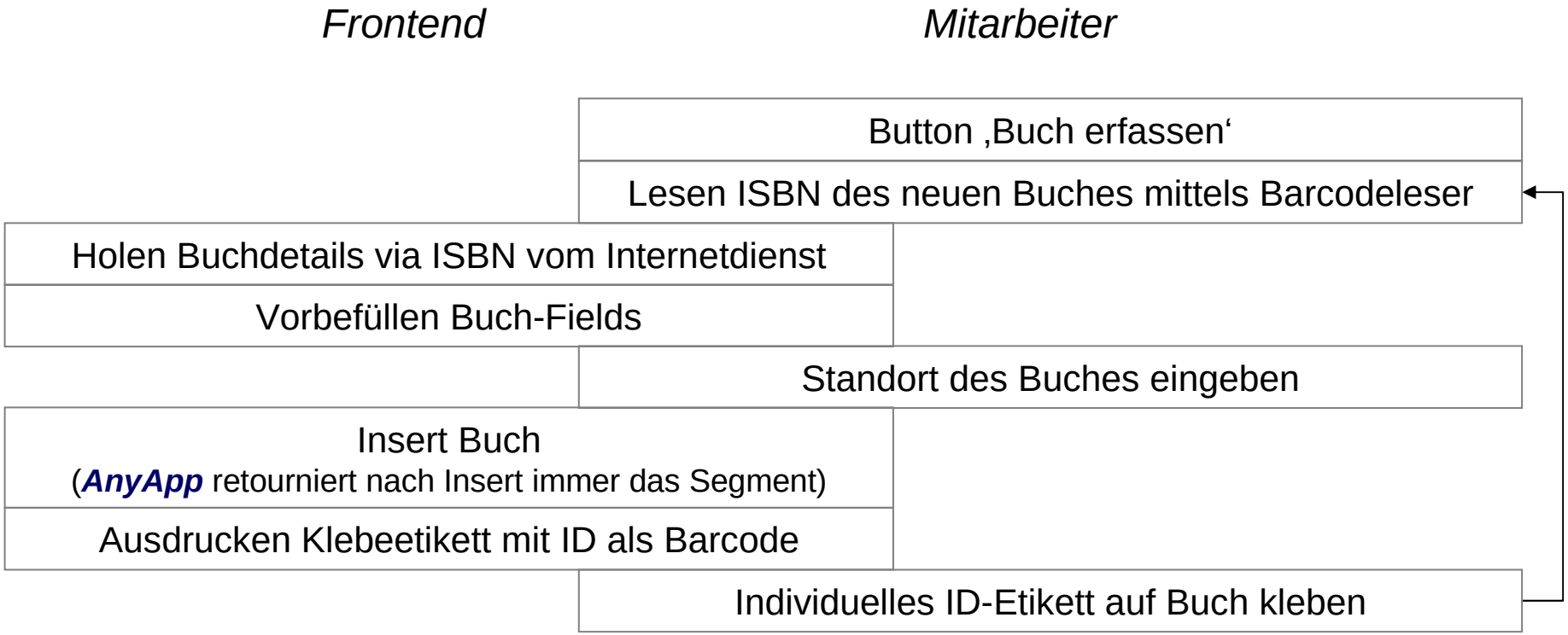
Wie kann das über **AnyApp** realisiert werden?

Der **AnyApp**-Kernel stellt sämtliche Methoden für die Sichtung und Wartung des Datencubes zur Verfügung. Darunter sind auch Methoden, die die Standardoberfläche nicht verwendet, wie z.B. der Command *pos-detail*. Diesem sind, wie bei der üblichen Suche, Suchfelder für das gewünschte Segment mitzugeben. Die Methode retourniert direkt die Knotendetails, allerdings nur dann, wenn die Ergebnismenge der Suche genau 1 ist. Dies entspricht somit einem Get-Unique über dynamische Schlüsselbegriffe. Diese Methode zusammen mit einem DB-Trigger verkürzt die obige Prozesskette wie folgt:

1. *pos-detail* auf tblkunde mit Kunden-ID = Barcode-ID
2. für jede Buch-ID:
   a. *pos-detail* auf tblbuch mit Buch-ID = Barcode-ID des Buches,
   b. *insert* auf tblausleihung (Schlüssel tblkunde aus 1. + neues Segment)
   c. *join* auf Buch (Schlüssel neue Ausleihung + Schlüssel Buch aus a., initiiert auch den DB-Trigger zur Wartung der Felder anz_ausl (Anzahl Entlehnungen) sowie des Buchstatus).

Bei der Herstellung eines kundenspezifischen Frontends reduziert sich die Programmierung der Kommunikation zum **AnyApp**-Kernel auf das Senden der fertigen Requests, die zudem meist von einer gleichartigen manuellen Prozesskette kopiert werden können.

Prozess bei der Erfassung neuer Bücher:



Werden Bücher bereits bei der Bestellung ins System eingebucht, ist statt des Inserts eine Suche beim Verlag nach einem Buch mit selber ISBN und Status ‚bestellt‘ abzusetzen.

Fazit:

Applikationen unter **AnyApp**

- Bedürfen i.A. keiner HOST-Programmierskills
- Bedürfen kaum programmtechnischer Wartung
- Bedürfen keiner speziellen Kommunikationsprozesse
- Minimieren Formatprobleme
- Minimieren Probleme mit Datenstrukturen und Topologie
- Minimieren Plausibilitätsprobleme
- Werden extern über Regeln gesteuert
- Können den Frontend steuern
- Können von jedem Frontend bedient werden
- Sind binnnen weniger Tage produktiv einsetzbar
- Ersparen 80% bis 90% des traditionellen Entwicklungsaufwands
- Ersparen 50% bis 80% des traditionellen Wartungsaufwands
- Benötigen kaum Dokumentation (selbstdokumentierendes Regelwek)
- Sind extrem performant (diese Demoversion: 65 ms / CTG-Transition)

AnyApp – einfach genial

Danke für die Mitarbeit