

Inhaltsverzeichnis	1
Masken-Tutorial	2
Orientierung	2
Grundlagen	3
Abrufen einer Ergebnistabelle	3
Die Felddefinitionen	3
Speichern der Felddefinition	4
Änderung einer Felddefinition	5
Maskendefinition	5
Abfragen in Maskendefinitionen	6
Masken erstellen aus Kopien	7
Masken oder Felder löschen	8
Konventionen	8
Abfragenentwurf mit SuperX-Sichten	8
Einträge verstecken oder nicht-selektierbar machen	10
User-/Gruppenrechte	11
Benutzen der Sichten in Masken	11

# Masken-Tutorial

---

## Orientierung

---

Die Abfragemasken liefern die Daten aus den Basissystemen an das SuperX-Frontend aus. Einige Abfragen zur Administration sind im Kernmodul enthalten, die Abfragen zu den Basissystemen sind in den jeweiligen Modulen enthalten. Die Abfragen in der Administration erlauben es, neue Masken anzulegen, zu kopieren und zu löschen.

Um den Austausch von Abfragen innerhalb der Hochschulen zu erleichtern ("Abfragen-Pooling" über die SuperX-Website), sollten die Masken immer im Nummernkreis xxxx0000 bis xxxx9970 liegen, wobei xxxx der von der HIS verwandten Hochschulnummer entspricht. Es sind Dreißigerschritte zu wählen, da die dazwischen liegenden Nummern für die Maskenfelder (Tabelle '[felderinfo](#)') reserviert sind.

Im Folgenden finden Sie allgemeine Hinweise für die Verwaltung der Masken. Die Masken lassen sich browserbasiert, über UNIX-Shellscripte, und über Access administrieren. Weitergehende Möglichkeiten bietet aber das XML-Frontend (Möglichkeit der Editierung von großen text -Feldern bei Postgres als Datenbanksystem). Nach der Anmeldung haben Administratoren das Recht, Masken zu löschen, zu kopieren und erzeugen. Die einzelnen Felder der Masken lassen sich direkt in der Datenbank oder z.B. mit MS Access verändern. Im Applet sind nur grundlegende Verwaltungsoperationen möglich. Sie sind als Ersatz für die UNIX- Scripte gedacht.

Maske kopieren: Wie im [UNIX - Script](#) wird eine Maske in eine neue Maske kopiert, und alle zugehörigen Tabellen werden aktualisiert. Zusätzlich wird auch der Eintrag im Themenbaum gemacht. Bei der Nummer der Maske (tid) sollten Sie das Nummernschema von SuperX einhalten, um in Zukunft Abfragen-Pooling zu ermöglichen.

Maske löschen: Wie im [UNIX-Script](#) werden Masken aus allen dazugehörigen Tabellen entfernt. Zusätzlich wird auch der Eintrag im Themenbaum gelöscht. Zur Sicherheit muss die Nummer der Maske manuell eingegeben werden.

Maske suchen: Sie können Masken suchen und im XML-Frontend komfortabel editieren. Schränken Sie Ihre Auswahl auf ein Sachgebiet ein, und drücken Sie "Abschicken". Sie erhalten eine Liste mit "Treffern", und rechts befinden sich jeweils Buttons zum ansehen bzw. editieren einer Maske.

Die Maske läuft nur unter Postgres, weil Informix kein direktes Bearbeiten von Blob-Feldern mit sql unterstützt.

Analog können Felder kopiert, gelöscht oder gesucht werden.

Die Abfragen sind selbsterklärend; das Erzeugen neuer Masken, Löschen vorhandener Masken und Kopieren vorhandener Masken ist nur für Userkennungen möglich, die in der Tabelle userinfo im Feld administration den Wert 1 haben. Natürlich sollten die Abfragen sehr vorsichtig benutzt werden, sie sind die einzigen Abfragen in SuperX, die tatsächlich Änderungen an der Datenbank vornehmen können.

## Grundlagen

---

In den folgenden Abschnitten werden die Grundlagen der Maskenentwicklung erläutert. Beginnend beim Abrufen einer Ergebnistabelle aus einer Maske, über die Definition der Felder dieser Maske, bis hin zur Definition der dahinterliegenden SQL-Anweisungen.

### Abrufen einer Ergebnistabelle

---

Der Ausgangspunkt ist ein Beispiel aus dem SOS-Modul. Auf der folgenden Abbildung ist der Themenbaum ersichtlich, welcher sich in verschiedene Bereiche gliedert.

imgsrcThemenbaum.PNGwidth600captionThemenbaum

Wählt man aus der Navigation auf der linken Seite einen Bereich aus, werden Abfragen sichtbar. In diesem Fall ist der Bereich "Bewerbung, Zulassung" ausgewählt und es wird unter anderem die Abfrage "Bewerbungsprozess nach Fach/Studiengang" sichtbar.

imgsrcMaske.PNGwidth600captionMaske

Wählt man eine Abfrage aus, erscheint eine Maske, welche verschiedene Möglichkeiten zur Parametrisierung bietet.

imgsrcErgebnistabelle.PNGwidth600captionErgebnistabelle

Nach einem Klick auf "Abschicken" erscheint die entsprechende Ergebnistabelle.

### Die Felddefinitionen

---

Gehen wir kurz zurück zur Auswahlmaske. Jedes Feld der Maske, z.B. "Semester", ist ein Datensatz in der Tabelle 'felderinfo'. Dort finden Sie Angaben zum Namen, Inhalt und Layout des Feldes. Gehen wir zunächst zum Inhalt des Feldes: Die Liste der Semester.

imgsrcMaskenfeldSemester.pngwidth400captionMaskenfeldSemester

Beim Klick auf das Feld Semester erhalten wir eine Reihe von Semestern zur Auswahl. Die Liste ist absteigend sortiert. Sie wird durch die SQL Anweisung `select tid, eintrag from zul_semester order by tid DESC`; erzeugt. Die SQL-Anweisung liefert aus der Tabelle 'zul\_semester' die Felder 'tid' ("Tupelidentifizier") und 'eintrag' (der Volltext des Semesters). Der Schlüssel des Feldes 'tid' ist unsichtbar, sorgt aber dafür, dass die Sortierung richtig erfolgt.

imgsrcDatenbanktabelleZulSemester.pngwidth400captionDatenbanktabelleZulSemester

Hier sehen Sie einen Screenshot der Tabelle 'zul\_semester' direkt in der Datenbank. Die Nummerierung ist fünfstellig und besteht aus Jahr (vier Stellen) und 1 für Sommer- und 2 für Wintersemester.

## Speichern der Felddefinition

---

Wo wird nun in SuperX die Felddefinition gespeichert? Viele Scripte in SuperX werden selbst in Datenbanktabellen abgelegt, die Tabelle [felderinfo](#) enthält die relevanten Angaben für die Felder.

Um dies zu sehen, öffnen wir ein Formular im XML-Frontend. Dort befinden sich Bearbeitungsformulare für Felder und Masken.

imgsrcFeldSuchen.pngwidth600captionFeldSuchen

Im Themenbaum des XML-Frontends finden wir den Menüpunkt **Administration → Felder → Feld suchen**

imgsrcFeldSuchenMaske.pngwidth600captionFeldSuchenMaske

In dem Formular wählen wir die Abfrage "Bewerbungsprozess nach Fach/Studiengang" aus. Zusätzlich sehen wir auch die Nummer der Maske (26020), was bei der Maskenbearbeitung ganz nützlich ist. Mit einem Klick auf "Abschicken" werden die Eingaben bestätigt und abgeschickt.

imgsrcFeldSuchenErgebnis.pngwidth600captionFeldSuchenErgebnis

Es wird eine Liste mit den vorhandenen Feldern geliefert. Wir sehen die Nummer des Feldes und den Namen. Rechts daneben befindet sich ein Knopf zum Bearbeiten des Feldes.

Ein kleiner Hinweis an dieser Stelle: Die Felder werden in der Tabelle `masken_felder_bez` der Maske Nr. 26020 zugeordnet. Wir zählen also bei Feldnummern in Einer-Schritten von der Maskennummer aus hoch. Aus diesem Grunde wählen wir bei Maskennummern größere Intervalle.

imgsrcBearbeitungsformularFeld.pngwidth600captionBearbeitungsformularFeld

Wir wählen nun das Feld "Semester" und gelangen in ein Bearbeitungsformular der Tabelle 'felderinfo'. Wir sehen Name, Nummer, Position auf der Maske, Breite und Typ des Feldes (ganzzahlig). Das Feld ist obligatorisch und von der Art Nr. 1 (Nummer + Text, mit Dialog). Im Feld 'relation' steht nach dem

---

Steuerungszeichen <<SQL>> der SQL-Befehl. Unten ist noch der Defaultwert für das Feld angegeben, ebenfalls ein SQL-Ausdruck.

SuperX liest also aus der Datenbank die Scripte für eine Maske bzw. für ein Feld aus einer Tabelle, und führt Sie dann in der Datenbank aus.

## Änderung einer Felddefinition

---

imgsrcFeldBearbeiten.pngwidth600captionFeldBearbeiten

Um die Felddefinition zu ändern, kann nun beispielsweise beim SQL-Befehl DESC durch ASC ersetzt und gespeichert werden.

imgsrcMaskenfeldSemesterASC.pngwidth400captionMaskenfeldSemesterASC

Um das neue Script aus der Datenbank zu laden, muss die Maske nun erneut über den Themebaum aufgerufen werden. Wenn wir dann in der Maske auf "Semester" klicken, erscheinen die Semester in aufsteigender Reihenfolge.

Auf diese Art und Weise können wir alle Maskenfelder bearbeiten. Die restlichen Attribute in der Tabelléfelderinfo' sind im [Administrationshandbuch Kernmodul](#) erläutert.

## Maskendefinition

---

So weit so gut, wir können nun also Felder ändern. Wie können wir nun die Ergebnistabellen bearbeiten?

SuperX arbeitet hier ebenfalls mit SQL-Scripten, die als Felder in einer Tabelle gespeichert sind. Die Tabelle lautet 'maskeninfo'. Wir können uns diese Tabelle ebenfalls im XML-Frontend anschauen:

imgsrcMaskeSuchenMaske.pngwidth600captionMaskeSuchenMaske

Im Themenbaum wählen wir "Administration → Masken verwalten → Maske suchen". Es erscheint eine Auswahlmaske ohne Vorbelegung. Es lässt sich nun eine Maske anhand einer Auswahlliste oder Maskennummer suchen. Um das Feld für die Suche mit Maskennummer einzublenden, muss das Schlüsselsymbol geklickt werden.

imgsrcMaskeSuchenErgebnis.pngwidth600captionMaskeSuchenErgebnis

Als Ergebnis sehen wir unsere Maske sowie zwei Bearbeitungsbuttons. Wir wählen den ersten Button, **Bearbeiten**.

imgsrcMaskeninfoVerwalten.pngwidth600captionMaskeninfoVerwalten

Wir gelangen in das Bearbeitungsformular der Maske. Neben der Nummer der Maske sehen wir unter anderem den Namen und die Felder 'Select-Statement' und 'Spaltenlayout'. Das Feld 'Select-Statement' enthält das SQL-Script und 'Spaltenlayout' die Ergebnisdarstellung.

## Abfragen in Maskendefinitionen

---

Allgemein formuliert arbeiten wir so:

1. Zunächst werden die Eingaben in den Auswahlfeldern ausgewertet und eine Ergebnismenge ermittelt, meist in Form einer temporären Tabelle.
2. Diese Tabelle wird mit den Hilfstabellen in der Datenbank gejoined, und es wird eine Ergebnistabelle berechnet. Ggf. werden noch Summen oder Prozente berechnet, meist benötigen wir dazu weitere temporäre Tabellen.
3. Der letzte select im Feld 'Select-Statement' enthält die Ergebnistabelle, die das Applet empfängt. Im Feld 'Spaltenlayout' werden die Spaltenüberschriften- und Breiten gesetzt und das Ergebnis wird angezeigt.
4. Direkt danach wird die letzte temporäre Tabelle gedroppt, und die Datenbankverbindung wird an das SuperX-Servlet zurückgegeben.

imgsrcMaskeSuchenPruefungsstatistik.pngwidth600captionMaskeSuchenPruefungsstatistik

Im Folgenden wird als Beispiel der Bericht "Studierende, Prüfungen → Einzelprüfungen → Prüfungsstatistik" herangezogen. Das Script in 'Select-Statement' ist relativ lang, wir wollen es daher nur Auszugsweise kommentieren.

imgsrcSelectStatement1.pngwidth600captionSelectStatement1

Zu Beginn des Select-Statements wird über `select ... into temp tmp_ergebnis` die Ergebnismenge der temporären Tabelle 'tmp\_ergebnis' übergeben. Einige Felder unterliegen bestimmten Kriterien, wie bspw. das Feld 'bestanden': `(case when P.pstatus='BE' then 1 else 0 end) as bestanden`

imgsrcSelectStatement2.pngwidth600captionSelectStatement2

In der folgenden where-Klausel werden verschiedene Bedingungen festgelegt. Hier finden sich Platzhalter wie `<<Seit Semester>>`. Diese Platzhalter entsprechen den Maskenfeldern und werden mit den Benutzereingaben gefüllt. Passagen innerhalb `/* ... */` werden nur dann ausgeführt, wenn das entsprechende Maskenfeld gefüllt wurde. Da es sich bei `<<Seit Semester>>` um ein Pflichtfeld handelt, ist diese Einschränkung dort nicht nötig.

imgsrcSelectStatement3.pngwidth600captionSelectStatement3

Um diverse Berechnungen anzustellen werden, wird die Ergebnismenge in eine zweite temporäre Tabelle überführt: `select ... into temp tmp_ergebnis2`. Die erste temporäre Ergebnistabelle wird gelöscht: `drop table tmp_ergebnis;`

imgsrcSelectStatement4.pngwidth600captionSelectStatement4

Der letzte Select greift auf die zweite temporäre Tabelle zu und enthält die Ergebnistabelle.

imgsrcSpaltenlayout.pngwidth600captionSpaltenlayout

Im Feld 'Spaltenlayout' wird das Layout der Spalten der Ergebnistabelle festgelegt. Der markierte Bereich definiert das Layout der ersten Spalte. Spaltenüberschriften sind von 0 aufsteigend durchnummeriert, und die Überschriften selbst können durch "\n" mit Zeilenumbrüchen versehen werden. Pro Überschrift wird ein Absatz formuliert, u.a. auch die Breite der Spalte. Dabei ist zu beachten, dass die Darstellung in HTML vom verwandten Browser abhängt. 'Width' beispielsweise hat auf die HTML-Darstellung keinen Einfluss.

imgsrcCleanupStatements.pngwidth600captionCleanupStatements

Wichtig ist außerdem, dass die zweite temporäre Tabelle gelöscht wird. Dies geschieht durch ein kurzes SQL-Statement in 'Cleanup Statements'.

## Masken erstellen aus Kopien

---

Zur Maskenerstellung werden Kopien bereits existierender Masken erstellt. Es ist ratsam Masken zu kopieren, welche der neu zu erstellenden Masken thematisch ähneln. Das bringt den Vorteil, dass diverse Felder der Kopie weiterverwendet werden können. Unter AdministrationMasken verwaltenMaske kopieren findet sich die entsprechende Maske.

Es wird zunächst die zu kopierende Maske gewählt und anschließend eine tid und ein Name, sowie eine Erläuterung für die neue Maske vergeben. Nach Patch 2022-01-28 kann wahlweise die neue Maske direkt als Makro erstellt werden.

imgsrcMaske\_kopieren\_1.pngwidth600captionAuswahlmaske Maske kopieren

Bei diesem Beispiel würde die Maske "Studierende nach Alter" kopiert und eine neue Masken "Studierende nach Geschlecht" mit der tid 15980 erstellt.

Bei der Vergabe der neuen tid unbedingt darauf achten, dass diese nicht bereits belegt ist.

Bei der Vergabe der neuen tid wird mindestens in Dreißigerintervallen vorgegangen, damit für die Felder der Masken ausreichend tids verfügbar sind.

Um die belegten Masken-tid zu ermitteln eignet sich AdministrationMasken verwaltenMaske suchen, da dort beim Feld Maske die Masken nach tids aufsteigend sortiert sind.

imgsrcMasken\_tids.pngwidth600captionAuswahlmaske Maske suchen

Die Feld-tids werden beginnend ab der Masken-tid direkt aufeinanderfolgend vergeben. Ein Dreißigerintervall zwischen den Masken-tids ermöglicht es somit dreißig Felder pro Maske zu vergeben. Trotzdem sollte immer geprüft werden, ob die vorherige Maske gegebenenfalls mehr als dreißig Felder besitzt und entsprechend ein größeres Intervall zu wählen ist. Hierfür eignet sich AdministrationMasken verwaltenFelderFeld suchen

imgsrcFeld\_suchen\_1.pngwidth600captionAuswahlmaske Felder suchen

imgsrcFeld\_suchen\_2.pngwidth400captionErgebnistabelle Felder suchen

Die Maske "16340-Studierende nach Alter" besitzt 17 Felder. Ein Dreißigerintervall wäre somit ausreichend.

## Masken oder Felder löschen

---

Masken und Felder werden über AdministrationMasken verwaltenFelderFeld löschen und AdministrationMasken verwaltenMaske löschen gelöscht. Dies hat den Vorteil, dass anhand der entsprechenden tid die Einträge für diese Maske oder dieses Feld aus sämtlichen Tabellen gelöscht wird. Falls die Einträge für ein Feld oder eine Maske nur in den Tabellen maskeninfo und felderinfo gelöscht würden, wären beispielsweise in der Feldverwaltung leere Felder zu finden. Um den Datenbestand sauber zu halten wird daher empfohlen, die angesprochenen Formulare "Feld löschen" und "Maske löschen" zu nutzen.

## Konventionen

---

Für die Erstellung von SQL-Abfragen gibt es ein paar wichtige Konventionen:

- wenn Sie Abfragen schreiben, sollten Sie temporäre Tabellen immer mit "tmp\_" vorab benennen. Grund: es könnte mal sein, dass wir die Tabelle auch als statische Tabelle nutzen. Dann gibt es einen SQL-Fehler.
- vermeiden Sie bei temporären Tabellen Umlaute in Spaltennamen (z.B. "Fakultät"). Das klappt zwar unter Informix, aber nicht unter Postgres.
- Nutzen Sie, wenn möglich, die ANSI-SQL-Standards.

## Abfragenentwurf mit SuperX-Sichten

---

In SuperX können bei Auswahldialogen verschiedene Sichten angeboten werden (vgl.[Alternative Hierarchien zur Datenselektion](#)).

Alternative Hierarchien bzw. Auswertungshierarchien aus COB werden automatisiert übernommen. Jede Hierarchie bekommt einen Eintrag in der Sichtentabelle.

imgsrcSichtenAuswahl.pngwidth600captionSichtenAuswahl

Achtung: Die Inhalte und Rechte der Sichten werden gecached. Nach einer Änderung der Inhalte einer

Sicht im laufenden Betrieb musst man im SuperXManager den Cache leeren und sich neu anmelden. Die Sichten werden in der Tabelle 'sichten' definiert. Eine Übersicht über die Tabellenstruktur inklusive Beschreibung findet sich [hier](#).

Die Tabelle 'sichten' lässt sich wie auch die Masken und Felder (s. Abschnitt [Maskendefinition](#) und [Speichern der Felddefinition](#)) in SuperX über ein Bearbeitungsformular manipulieren.

imgsrcSichtSuchen.pngwidth600captionSichtSuchen

Im Folgenden wird die Sicht "FB/Fak, Lehrinheit, Fach/Abschluss" näher betrachtet.

imgsrcSichtFachAbschluss.pngwidth600captionSichtFachAbschluss

Bei sos\_org\_lehre\_fach\_absch\_ude handelt es sich um eine "View". Diese "View" beinhaltet mehrere Select-Statements, welche quasi die verschiedenen Ebenen der Sicht "FB/Fak, Lehrinheit, Fach/Abschluss" darstellen.

imgsrcViewFachAbschUDE.pngwidth600captionViewFachAbschUDE

Das Organigramm stellt eine integrierende Sicht für verschiedene Datenquellen zusammen und ist somit die Voraussetzung für eine integrierte Betrachtung. Alle Einrichtungen, Institutionen und Projekte sind im Organigramm hierarchisch angeordnet (vgl. [Das Organigramm](#)). Das Feld 'lehre' wird entweder mit 1 oder 0 gefüllt, um festzustellen ob die Institution relevant für Auswertungen im Bereich Lehre ist (1=ja, 0=nein).

Erläuterung des parent-Felds:

Das Parent-Feld gibt die key\_apnr der übergeordneten organisatorischen Einheit an. Das root-Element des Baums besitzt als parent einen null-Wert. Das Organigramm darf dementsprechend nur ein Element haben, das keinen parent besitzt. Hier sollte grundsätzlich die Hochschulnummer eingetragen werden die auch in anderen HIS-Programmen verwendet wird. In Duisburg ist dies z.B. der Basiseintrag key\_apnr = "70" ( GMU Duisburg ). Es ist praktisch der oberste "Knoten" im Baum. Dann werden alle Einträge in der Tabelle organigramm gesucht, die parent="70" haben – also direkte Kinder des Basiseintrags.

Hier findet sich u.a. Einrichtungen Forschung und Lehre (key\_apnr="7", parent="70"). Dieser Eintrag hat wiederum u.a. folgende Kinder:

- Fakultät 1 (key\_apnr="1100", parent="7")
- Fakultät 2 (key\_apnr="1200", parent="7")
- Fakultät 3 (key\_apnr="1300", parent="7")

usw.

Die weiteren Äste des Baums werden rekursiv abgefragt.

Die Struktur der Tabelle 'organigramm' inklusive Beschreibung findet sich [hier](#).

Die Tabelle 'dim\_studiengang' enthält Datensätze zu Studiengängen (Die Struktur der Tabelle inklusive Beschreibung findet sich [hier](#)).

Für alle Sichten wird erwartet, dass mindestens die drei Felder name, key und parent geliefert werden (der konkrete Name der Spalten ist irrelevant).

Anschließend können optional noch Strukturinformationen folgen (entweder Integer oder String).

Beispiel:

```
sp_fin_inst_hier(<,<,<);select name,key,parent,strukturint from tmp_hier order by name; drop table tmp_hier;
strukturint (gefüllt aus fin_inst.orgstruktur) gibt mit Zahlen an, ob ein Eintrag eine Lehrinheit oder ein
Fachbereich ist.
```

Auf diese Information kann später in Abfragen mit FreeMarker zugegriffen werden.

Beispiel:

imgsrcFreemarkerStrukturInt.pngwidth500captionFreemarkerStrukturInt

## Einträge verstecken oder nicht-selektierbar machen

Es kann gewünscht sein, dass Einträge versteckt hinterlegt werden. Betrachtet sei weiterhin die Sicht '[FB/Fak](#), [Lehrinheit](#), [Fach/Abschluss](#)'. Die unterste sichtbare Ebene ist Fach/Abschluss. Man könnte also annehmen, es würde genügen, diese Sicht bis zur Ebene Fach/Abschluss aufzubauen. Dem ist allerdings nicht so. Es benötigt eine weitere, darunterliegende, aber nicht sichtbare Ebene "Studiengang".

imgsrcViewFachAbschUDESichtbarUnsichtbar.pngwidth600captionViewFachAbschUDESichtbarUnsichtbar

Zum Verständnis ein Beispiel anhand der Auswertung "Studierende nach Alter". Zunächst wird eine Kombination "Fach/Abschluss" ausgewählt.

imgsrcAuswahlSichtBWLBA.pngwidth400captionAuswahlSichtBWLBA

Es erscheint folgende Ergebnistabelle:

imgsrcErgebnistabelleStudierendeNachAlter.pngwidth600captionErgebnistabelleStudierendeNachAlter

Im SQL erfolgt die Einschränkung auf den im Sichtenbaum ausgewählten Eintrag durch  
 where ... 's\_' || L.tid in <@printkeys Studiengang.allNeededKeysList/<@printkeys  
 Studiengang.allNeededKeysList/> ... .

Für unser Beispiel sieht diese Passage nach der FreeMarker-Transformation folgendermaßen aus:

```
where ... 's_' || L.tid in ('_LE-WI_021_BS','s_3','s_36','s_33','s_65','s_270','s_87') ....
```

Würde die unterste, unsichtbare Ebene 'Studiengang' entfallen (s. View '[sos\\_org\\_lehre\\_fach\\_absch\\_ude](#)'), sähe der SQL stattdessen so aus:

```
where ... 's_' || L.tid in ('_LE-WI_021_BS') ....
```

Das hätte zur Folge, dass keine Einträge gefunden würden, denn es wird nach dem Muster 's\_tid' gesucht.

Um eine Ebene unsichtbar zu machen benötigt es das Attribut 'nodeattrib'. Wird diesem der Wert 1 übergeben, so ist die Ebene unsichtbar. Mit dem Wert 0 bleibt die Ebene sichtbar (s. View

'[sos\\_org\\_lehre\\_fach\\_absch\\_ude](#)', Zeile 21 und 30).

Weiterhin kann es gewünscht sein, dass man in einer Sicht nur spezielle Einträge auswählen kann. So können man eine Lehreinheitsauswahl vielleicht so darstellen:

Uni XY

- Fak 1
  - Lehreinheit A
  - Lehreinheit B
- Fak 2
  - Lehreinheit C
  - Lehreinheit D

Die Kategorisierung Fak 1/2 dient der Klarheit der Darstellung, für eine Abfrage soll aber vielleicht nur eine konkrete Lehreinheit auswählbar sein. In dem Fall kann wie oben das nodeattrib 2 vergeben werden. Dadurch wird ein Eintrag als nicht-selektierbar markiert.

## User-/Gruppenrechte

---

Außer Administratoren dürfen normale User zunächst einmal keine der nicht-internen Sichten benutzen. In den Tabellen `user_sichten` und `user_sichtarten` kann hinterlegt werden, dass ein User eine einzelne Sicht (z.B. `name_intern` Kostenstellen-Hauptsicht) oder eine ganze Sichtart (z.B. Kostenarten) benutzen darf. Für Gruppen gibt es analog die Tabellen `group_sichten` und `group_sichtarten`. Außerdem `sachgeb_sichten` und `sachgeb_sichtarten`, wenn Leute die ganze Sachgebiete sehen dürfen auch Sicht sehen sollen.

Die Pflege dieser Tabellen kann bequem über Administrationsformular im XML-Frontend vorgenommen werden.

Achtung:

Nach Änderungen muss der Server-Cache aktualisiert werden und der User muss sich neu anmelden.

Alternativ könnte auch Tomcat neu gestartet werden.

Im Dateisystem unter `$$SUPERX_DIR/db/bin` gibt es zwei Skripte `userrechte_sichern.x` und `userrechte_einspielen.x`.

Die entladen alle usertabellen (`userinfo`, `groupinfo`, `user_sachgeb_bez` etc) und spielen sie wieder ein.

Diese können z.B. dafür benutzt werden um alle Einstellungen von Test auf Echtrechner zu übernehmen.

## Benutzen der Sichten in Masken

---

---

In 'felderinfo' (s. [Speichern der Felddefinition](#)) muss als Art 12-Sicht eingetragen werden.

Im Feld Relation muss ein Select stehen, der die 'tids' der gewünschten Sichten zurückliefert.

imgsrcFeldBearbeitenArtSicht.pngwidth600captionFeldBearbeitenArtSicht

Die aktiv-Einschränkung ist praktisch für Entwicklungs- und Wartungszwecke, weil man dann leicht einzelne Sichten de/aktivieren kann - Sortierung nach type, damit erst Hauptsichten und dann alternative Hierarchien angezeigt werden, dann nach Name.

Der select sollte alle Sichtentids liefern, die ein Administrator sehen darf.

Wenn ein User für einzelne Sichten keine Berechtigung hat, filtert der Server diese automatisch raus.