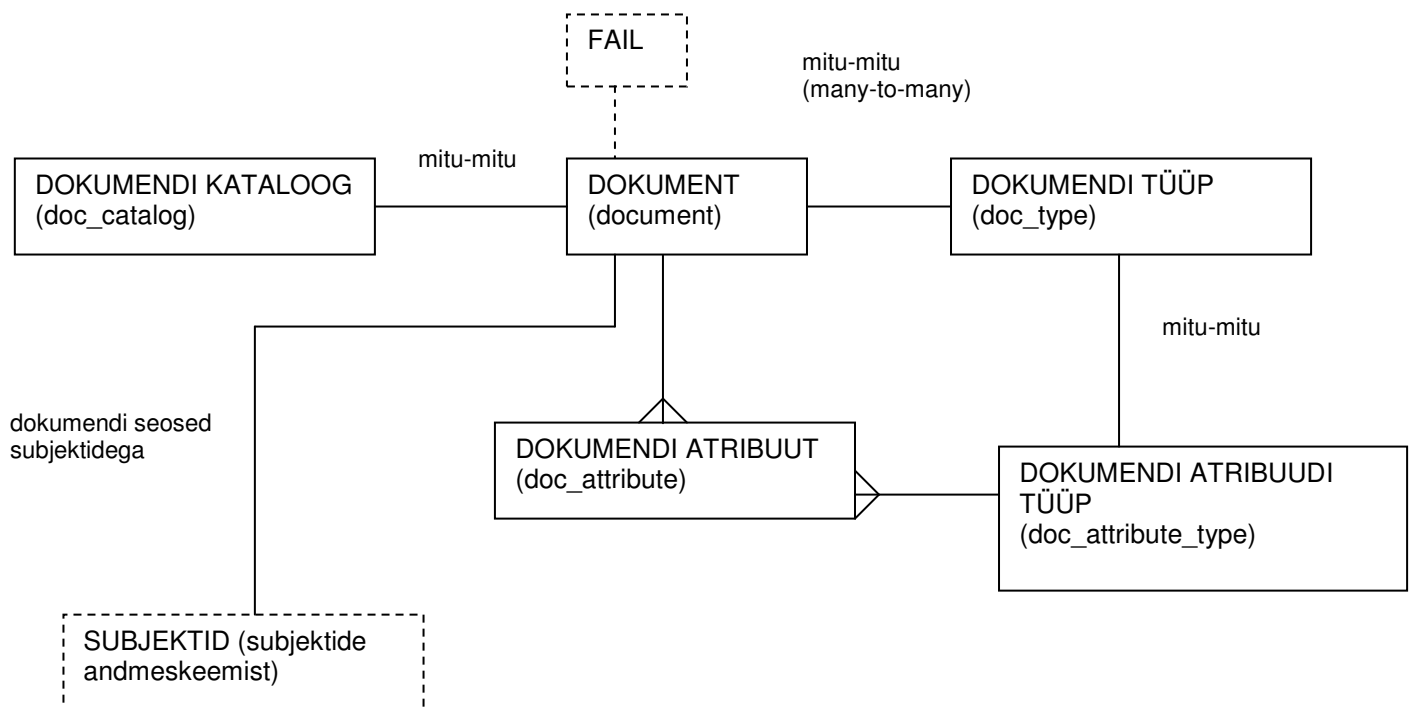


IDU0200

Praktikaülesande „DOKUMENDID” kirjeldus: andmebaasi struktuurid ja rakenduse funktsionaalsuse kirjeldus.

1. Sissejuhatus , ülevaade andmetest ja funktsionaalsusest	2
2. Andmed.	3
2.1. Andmebaasi loomine.	3
2.2. Andmebaasi skeem.....	4
2.3. Andmebaasi ülesehituse üldised põhimõtted.....	4
2.4. Andmetabelite ja väljade selgitused.	6
3. Rakenduselt oodatav funktsionaalsus	22
3.1. Funktsionaalsuste loetelu	22
Dokumendi tüüpide ettevalmistamine INSERT-lausetega:.....	22
Uute dokumentide sisestamine , sisestatud dokumentide andmete muutmine ja dokumentide kustutamine.	23
Dokumentide seostamine subjektide SUBJEKTIDE allsüsteemist.	23
Dokumentide otsing	24
Dokumentide tõstmine puhvrissi ja puhvrist liigutamine uude kataloogi („cut and paste”).	27
Dokumendifailide upload brauserist ja failide alla-laadimine – pole vaja teha.	29
Töötaja autentimine. Sisse- ja väljalogimine.	29
3.2. Ärireeglid ja andmete kontrollid. 10 reeglit.	30
4. Mis teha siis kui mingi osa ülesande funktsionaalsusest tundub ebaselge, kui midagi ei ole täpselt kirjeldatud?	30

1. Sissejuhatus , ülevaade andmetest ja funktsionaalsusest.



Tegemist on dokumentide metaandmete houdmise süsteemiga.

Süsteemi keskseks infoobjektiks on DOKUMENT.

Dokumendil on tüüp, dokumendi tüüp määrab ära millised on dokumendi atribuudid atrinbuutide tabelis. Dokumendi kui infoobjekti andmed on jagatud dokumendi tabeli ja atribuudi tabeli kirjete vahel.

Dokument asub dokumendi kataloogis, dokumenti on võimalik tõsta ühest kataloogist teise.

Dokumente on võimalik seostada isikute ja ettevõtetega SUBJEKTIDE allsüsteemist.

Dokumente saab ostida süsteemis dokumendi põhiantmete (nimi, kirjeldus) , dokumendiga seotud subjektide, dokumendi kataloogi , dokumendi tüübi , doukendi staatuse ja dokumendi atribuudi andmete järgi (täpsemalt vaata alt.)

Dokumendi staatuse muutmisel (täpsemalt – selle muudatuse salvestamisel) lisatakse see uus staatus ka dokumendi staatuste ajaloo tabelisse ja lõpetatakse selles tabelis eelmise staatuse kehtivus.

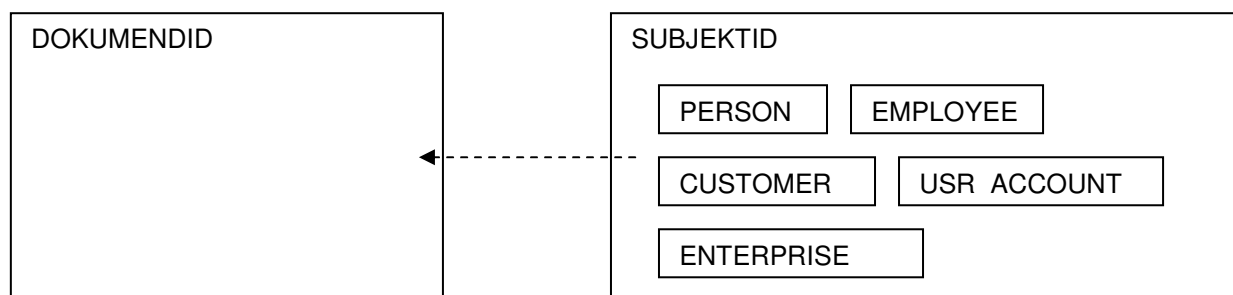
2. Andmed.

2.1. Andmebaasi loomine.

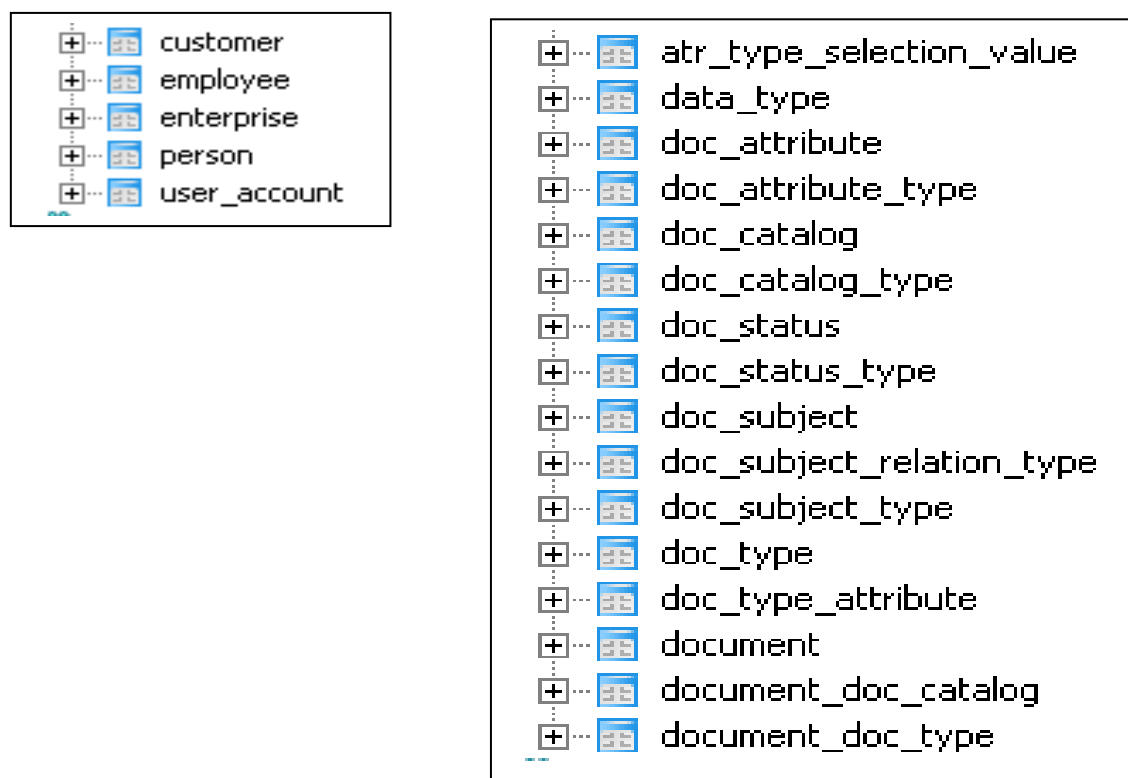
Selle ülesande andmebaasi loomiseks tuleb andmebaasis käivitada järgmiste failide sisu:

SUBJEKTID/CREATE_DB_SUBJEKT.txt
SUBJEKTID/INSERT_DATA_SUBJEKT.txt

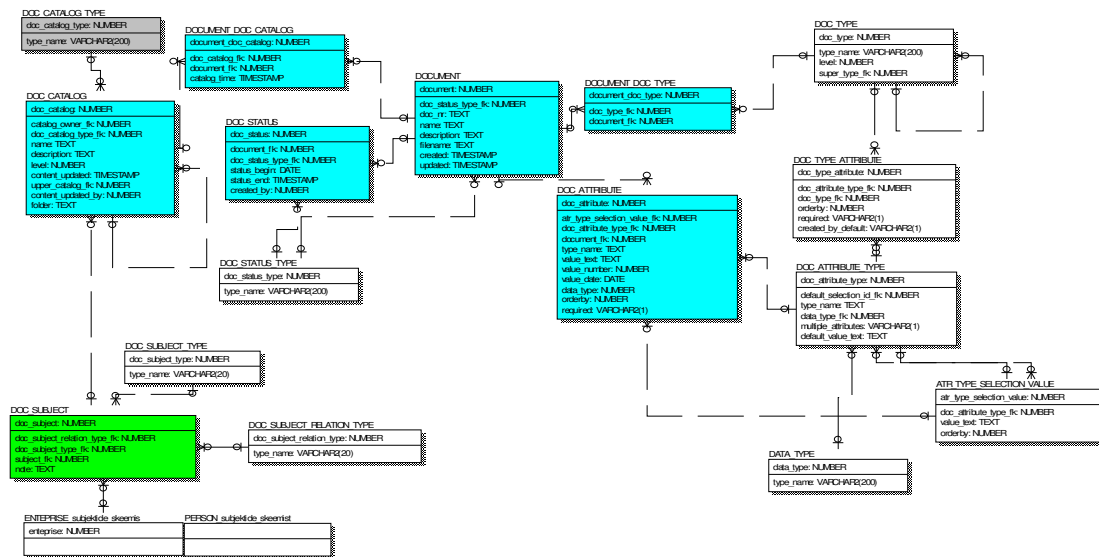
DOKUMENDID/CREATE_DB_DOKUMENT.txt
DOKUMENDID/INSERT_DATA_DOKUMENT.txt



Lisaks ülesande enda andmebaasi skeemi tabelitele (CREATE_DB_REMONDITEENUS) kasutatakse selles ülesandes tabelleid ülesande SUBJEKTID andmebaasi skeemist: PERSON, CUSTOMER, ENTERPRISE, EMPLOYEE, USR_ACCOUNT



2.2. Andmebaasi skeem.



„sinised” ja „rohelistes” on sellised tabelid kus läbi teie tehtava rakenduse peaks olema võimalik andmeid lisada, muuta, kustutada.

„rohelistes” tabelitega seotud funktsionaalsuse võivad need kes teevad tööd üksi kõrvale jätta.

2.3. Andmebaasi ülesehituse üldised põhimõtted.

* tabelite võtmeväljad on number-tüüpi ja sama nimega mis tabeli nimi.

```

CREATE TABLE doc_catalog_type
( doc_catalog_type numeric(10,0) NOT NULL ,
  ...
  CONSTRAINT doc_catalog_type_pk PRIMARY KEY (doc_catalog_type)
);
    
```

* suurema osa tabelite puhul tehakse tabelite võtmeväljade väärtused andmebaasisüsteemi poolt (nn. autonumbrid). Erandiks on ainult need tabelid milles on vähe kirjeid ja kuhu kirjeid rakenduse töö jooksul ei lisata (liigid, tüübi – sellised klassifikaatorite tabelid kus on 5-6 kirjet)

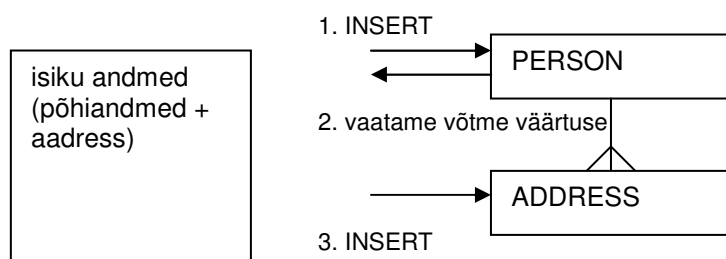
```

CREATE SEQUENCE document_id ;
    
```

```

CREATE TABLE document
( document numeric(10,0) NOT NULL DEFAULT nextval('document_id'),
  ..
  CONSTRAINT document_pk PRIMARY KEY (document)
);
    
```

* Autonumbrite tõttu tuleb andmete sisestamisel rakenduses järgida teatud andmete sisestamise (INSERT-lausete) järjekorda – enne tuleb sisestada andmed nendesse tabelutesse millele on vaja teistes kirjetes viidata.

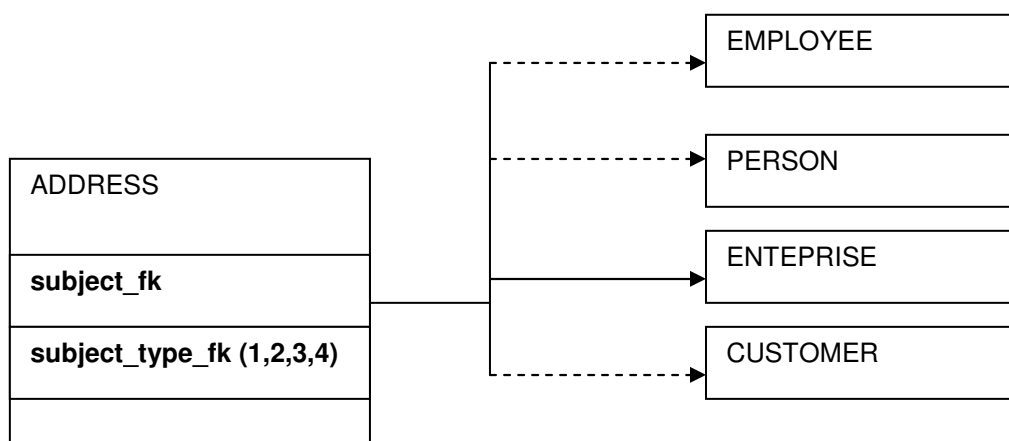


Näiteks : ühe andmevormi pealt isiku ja tema aadressis andmeid sisestades tuleb esimesena sisestada kirje PERSON tabelisse, siis (programmselt) vaadata mis sai PERSON-kirje võtmevälja väärtuseks ja siis sisestada PERSON-iga seotud kirje ADDRESS-tabelisse.

* „foreign key” piiranguid andmebaasides ei ole.

* andmeväljad mille sisu viitab teistele tabelitele (relatsioonid) on nimetatud enamasti nii et nende lõpus on „_fk”.

* mõned relatsioonid võivad viidata erinevatel juhtudel erinevatele andmebaasi tabelitele.




2.4. Andmetabelite ja väljade selgitused.

atr_type_selection_value

„Dokumendi atribuudi tüübi valikväärtused”

Mingi dokumendi atribuudi tüübi valikväärtused (sellel atribuudi tüübi peab data_type=4) mille hulgast kasutaja saab ekraanivormil valida atribuudile väärtuseid, väärtused ei ole siis kasutaja poolt sisestavad vaid ta saab ainult etteantud „listist” valida.

Sellesse tabelisse ei ole selles ülesandes vaja läbi rakenduse andmeid lisada, andmed sisestatakse INSERT—lausetega otse andmebaasi.


 atr_type_selection_value	Võtmeväli, sisu autonummerduv
value_text	Valikväärtuse nimetus
orderby	Järhekord – näitab millises järjekorras näidatakse kasutajale valikväärtusi (näiteks „combo-box”-is)

data_type

dokumendi atribuudi andmetüüp. Tabelites [doc_attribute_type] ja [doc_attribute] viidatakse sellele andmetüübile.

Võimalikud väärtused:

- 1 – atribuut on teksti tüüpi
- 2 – atribuut on number tüüpi
- 3 – atribuut on kuupäev/timestamp tüüpi
- 4 – atribuut on valiku tüüpi

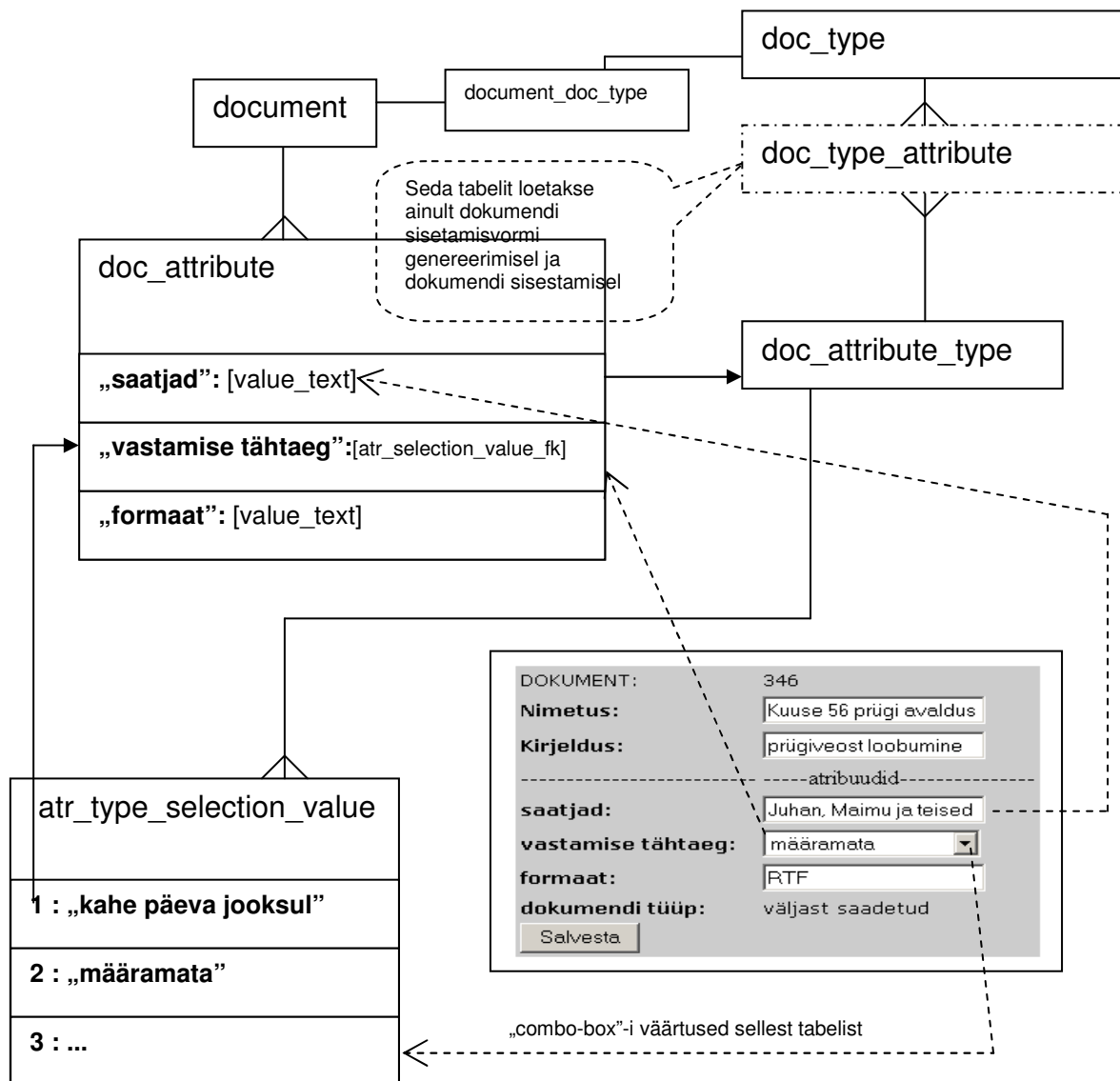
 data_type	Võtmeväli, sisu ei ole autonummerduv
type_name	Andmetüübi nimi

doc_attribute

dokumendi tunnus, dokumendi omadus ehk teiste sõnadega – dokumendi atribuut.

Dokumendi atribuut sisaldab väärtust mille sisestab ekraanivormilt kasutaja ja atribuudi tüüpi (mida väärtus tähendab), iga dokumendil võib süsteemis olla 0...N atribuuti. Dokumendi atribuut on seotud atribuudi tüübiga tabelis [doc_attribute_type]. [doc_attribute_type] määrab

- mis on atribuudi tüübi nimi („vastamise tähtaeg”, „turvatase”)
- mis on atribuudi andmetüüp (string, number, kuupäev või valik nimekirjast)
- mis on atribuudi vaikimisi väärtus tekstitüüpi atribuudi korral.
- mis on atribuudi väärtus valiku-tüüpi atribuudi korral (default_selection_id_fk)



Informatsioon dokumendi muutmise vormi näitamiseks võetakse antud näites kokku 6-st tabelist:

1. tabelist **[document]** dokumendi põhiaandmed (nimi, kirjeldus)
2. – 3. tabelitest **[document_doc_type]** ja **[doc_type]** dokumendi tüüp
4. tabelist **[doc_attribute]** dokumendi atribuutide väärtused („saatjad”, „vastamise tähtaeg”, „formaat”)
5. tabelist **[doc_attribute_type]** atribuutide nimed (mis tüüpi atribuudiga on tegemist). Kuigi seda tüübi nime võib dubleerida ka dokumendi atribuudi kirjes, tabelis **[doc_attribute]** on selle jaoks väli

„type_name”.

6. tabelist **[atr_type_selection_value]** valikväärtused juhul kui atribuudi andmetüüp on „valik” (data_type=4). Olemuselt on ikka tegemist number-tüübiga aga numbrid selle atribuudi tüübi väärtusena on tegelikult viited tabelisse **[atr_type_selection_value]**.

Valikväärtused tabelis **[atr_type_selection_value]** on seotud atribuudi tüübiga **[doc_attribute_type]** nii et kui me teame dokumendi atribuudi tüüpi siis me saame alati küsida **[atr_type_selection_value]** tabelist mis on sellele atribuudi tüübile vastavad valikud.

Millal sisestatakse dokumendi atribuudid tabelisse **[doc_attribute]** ?

Lühike vastus on – dokumendi sisestamisel tabelisse **[document]**, siis kui kasutaja vajutab „Sisesta uus dokument” nuppu.

Dokumendi atribuudid sisestatakse dokumendi lisamisel. Millised atribuudid tuleb dokumendile lisada see sõltub dokumendi tüübist – igal dokumendi tüübiga on seotud atribuudid tüübid – seda seost hoitakse tabelis **[doc_type_attribute]** („dokumendi tüübi atribuudid”). Selles ülesandes on tehtud selline eeldus et dokumendi lisamisel (lisamisvormi avamisel) on alati teada (on valitud) dokumendi tüüp. Kui dokumendi tüüp on teada siis on võimalik lugeda tabelist

[doc_type_attribute] millised atribuudi tüübid on selle dokumendi tüübiga seotud ja millises järjekorras tuleb neid atribuute ekraanivormil näidata (**[doc_type_attribute].orderby**). See informatsioon on piisav et genereerida dokumendi sisestamise vorm nii et seal oleksid ka väljad selle dokumendi tüübi atribuutide jaoks:

The screenshot shows a form titled "DOKUMENDI LISAMINE" with fields for "Nimetus:", "Kirjeldus:", "saatjad:", "vastamise tähtaeg:" (with a dropdown menu showing "määramata"), "formaad:", and "dokumendi tüüp:" (with a dropdown menu showing "väljast saadetud"). A button "Lisa uus dokument" is at the bottom. A dashed line separates the main fields from the "atribuudid" section. To the right, a diagram shows a box labeled "doc_type_attribute" connected by a line to a box labeled "doc_attribute_type", with arrows pointing from the boxes to the diagram.

Näiteks dokumendi tüüp 10 puhul saame sellise atribuudi tüüpide komplekti:

```
SELECT DAT.doc_attribute_type, DAT.type_name, DTA.orderby,
DTA.required, DTA.created_by_default, DAT.default_selection_id_fk AS valiku_id,
DAT.data_type_fk FROM doc_attribute_type DAT
INNER JOIN doc_type_attribute DTA ON
DAT.doc_attribute_type=DTA.doc_attribute_type_fk WHERE DTA.doc_type_fk=10;
```

	doc_attribute_type numeric(10,0)	type_name text	orderby numeric(10,0)	required character	created_by_default character varying(1)	valiku_id numeric(10)	data_type_fk numeric(1,0)
1	1	saatjad	1	N	Y		1
2	2	vastamise	2	Y	Y	4	4
3	3	formaad	3	N	Y		1

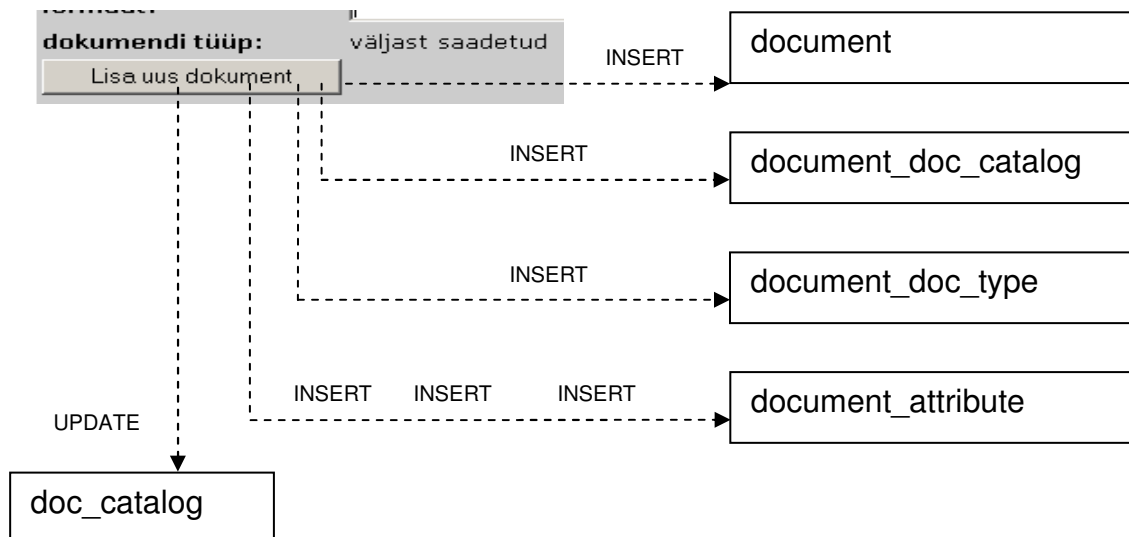
Kui on tegemist atribuudiga mille andmetüübiks on 4 („valik”) siis peame tabelist **[atr_type_selection_value]** küsima selle atribuudi tüübiga seotud valikud „como-box”-is näitamiseks:

doc_attribute_type numeric(10,0)	type_name text	orderby numeric(10,0)	required character	created_by_default character varying(1	valiku_id numeric(10	data_type_fk numeric(1,0)
1	saatjad	1	N	Y		1
2	vastamise	2	Y	Y	4	4
3	formaad	3	N	Y		1

SELECT ATSV.atr_type_selection_value, ATSV.value_text FROM atr_type_selection_value ATSV WHERE ATSV.doc_attribute_type_fk = 2 ORDER BY orderby;

	atr_type_selection_value numeric(10,0)	value_text text
1	1	kahe paeva jooksul
2	2	nadala jooksul
3	3	kuu aja jooksul
4	4	maaramata

Kui kasutaja nüüd vajutab dokumendi lisamise vormil lisamise nuppu tuleb rakenduses teha 6 INSERT-lauset.



1. INSERT tabelisse **[document]**
2. INSERT tabelisse **[document_doc_catalog]** (eeldame et dokumendi lisamisel on mingi dokumendi kataloog alati valitud) et siduda dokument dokumendikataloogiga
3. INSERT tabelisse **[document_doc_type]** et siduda dokument tema tüübiga
4. neli INSERT-i tabelisse **[document_attribute]** . Atribuudi väärtused võetakse dokumendi lisamise vormilt. Ilmselt on sellesse vormi , peidetud väljadesse („hidden fields”) võimalik vormi genereerimisel lisada ka infot slele kohta mis tüüpi atribuudiga on tegemist ja seda infot saab siis INSERT lausete konstrueerimisel rakenduse kasutada.

Lisaks tuleks uue dokumendi lisamisel uuendada dokumendi kataloogi tabelis **[doc_catalog]** selle kataloogi sisu uuendamise aega („content_updated”) ja kes sisu uuendas („content_updated_by”).

Atribuudi lisamisel loetakse tabelist [doc_type_attribute] ka info selle kohta kas selle dokumendi tüübi puhul on selle atribuudi väärtus nõutud (st. kasutaja ei tohi seda välja ekraanivormil tühjaks jätta) ja millises järjekorras atribuute ekraanivormil kuvatakse. Ja see info kopeeritakse [doc_attribute] tabelisse ka.

[doc_type_attribute].required = [doc_attribute].required
[doc_type_attribute].orderby = [doc_attribute].orderby

Atribuudi väärtused erinevatesse andmeväljadesse tabelis [doc_attribute].

Andmebaasi sisestatud dokumendi andmed saame (muutmisvormi genereerimiseks) kätte nüüd sellise päringuga:

```
SELECT D.document,DAT.type_name, DA.value_text,DA.doc_attribute_type_fk,
DA.value_number, DA.value_date,DA.atr_type_selection_value_fk FROM document D
LEFT JOIN doc_attribute DA ON D.document = DA.document_fk
INNER JOIN doc_attribute_type DAT ON DA.doc_attribute_type_fk =
DAT.doc_attribute_type
WHERE D.document = 1 ORDER BY DA.orderby;
```

	document numeric(10,0)	type_name text	value_text text	doc_attribute_type_fk numeric(10,0)	value_number numeric	value_date date	atr_type_selection_value_fk numeric(10,0)
1	1	saatjad	Juhan, Maimu ja teised	1			
2	1	vastamise tähtaeg		2			4
3	1	formaad	RTF	3			

DOKUMENT: 346
Nimetus: Kuuse 56 prügi avaldus
Kirjeldus: prügiveost loobumine
-----atribuudid-----
saatjad: Juhan, Maimu ja teised
vastamise tähtaeg: määramata
formaad: RTF
dokumendi tüüp: väljast saadetud
Salvesta

Ülemiselt pildilt näete et erinevat tüüpi atribuutide (tekst, number, kuupäev, valik) väärtused tuleb salvestada erinevatesse andmeväljadesse tabelis [doc_attribute] .

Et ülesanne ei läheks liiga keeruliseks ütleme et kehtivad järgmised eeldused:

- dokumendil saab olla ainult üks tüüp (andmebaasi skeem võimaldab dokumendile korraga mitu tüüpi aga teeme nii et paneme sinna iga dokumendi puhul ainult ühe seose tabeliga [doc_type]
- dokumendi tüüp pannakse paika dokumendi lisamisel ja seda ekraanivormilt hiljem muuta ei saa
- dokumendi atribuudid sisestatakse andmebaasi dokumendi lisamisel ja neid hiljem lisada või kutsutada ei saa (saab ainult väärtust „tühjaks” teha kui selle atribuudi „required”=’N’. See tähendab – dokumendi struktuur, tema atribuudid ja nende atribuutide järjestus pannakse paika dokumendi lisamisel ja seda hiljem muuta ei saa.

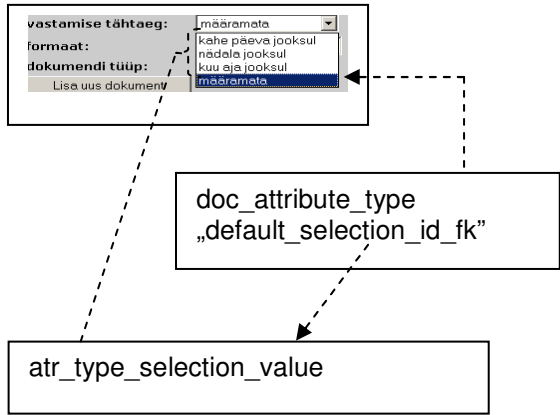


doc_attribute

atr_type_selection_value_fk

võtmeväli, autonummerdub

Viit atribuudi väärtusele kui valikule , tabelisse [atr_type_selection_value]. Täidetud juhul kui data_type=4 (valiku-tüüpi atribuut).

	 <p>Atribuudi väärtus (see tähendab viide tabeli [atr_type_selection_value] võib olla atribuudi lisamisel andmebaasi juba olla täidetud – juhul kui sellel atribuudi tüübil on vaikimisi väärtus olemas – [doc_attribute_type].default_selection_id_fk</p>
doc_attribute_type_fk	Viit dokumendi atribuudi tüübile tabelisse [doc_attribute_type_fk]
document_fk	Viit dokumendile mille atribuudiga on tegemist, tabelisse [document]
type_name	Dokumendi atribuudi tüübi nimi. Tegelikult on see kirjas tabelis [doc_attribute_type] aga soovi korral võib (atribuudi lisamisel) selle tüübi nime siia ka dubleerida.
value_text	Atribuudi väärtus. Kasutaja sisestab siia andmeid ekraanivormilt. Täidetud siis kui tegemist on tekst-tüüpi atribuudiga (data_type=1)
value_number	Atribuudi väärtus. Kasutaja sisestab siia andmeid ekraanivormilt. Täidetud siis kui tegemist on number-tüüpi atribuudiga (data_type=2)
value_date	Atribuudi väärtus. Kasutaja sisestab siia andmeid ekraanivormilt. Täidetud siis kui tegemist on kuupäev- või timestamp-tüüpi atribuudiga (data_type=3)
data_type	<p>Atribuudi andmetüüp</p> <p>data_type=1 – teksti-tüüpi atribuut data_type=2 – number-tüüpi atribuut data_type=3 – kuupäeva-tüüpi atribuut</p> <p>data_type=4 – valiku-tüüpi atribuut Selle välja sisust sõltub millisest andmeväljast näidatakse andmeid ekraanivormil ja millisesse andmevälja kirjutatakse kasutaja muudetud andmed tagasi (samuti päringud oskavad atribuudi väärtust selle välja järgi õigest andmeväljast vaadata)</p>
orderby	järjekord , näitab millises järjekorras näidatakse

	atribuute ekraanivormil Võetakse atribuudi lisamisel tabelist [doc_attribute_type]
required	Näitab kas kasutaja tohib andmeid sisestades ja muutes ekraanivormil selle atribuudi väärtust tühjaks jätta (required='N')või mitte (required='Y').Võetakse atribuudi lisamisel tabelist [doc_attribute_type]


doc_attribute_type

dokumendi atribuudi tüüp. Dokumentidel võib süsteemis olla erinevaid atribuute. Atribuutidel on tüübid, tüüp määrab mis on selle atribuudi tähendus, mis informatsiooni seda tüüpi atribuut hoiab. Millised atribuudid on seotud milliste dokumendi tüüpidega – see on kirjas tabelis [doc_type_attribute].

Üks atribuudi tüüp võib olla seotud mitme dokumenditüübiga.

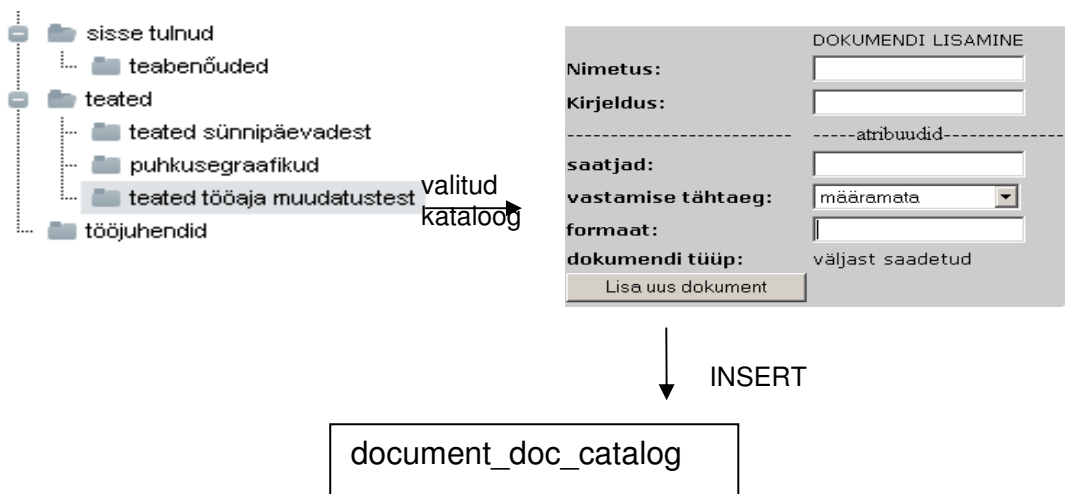
Dokumendi atribuudi tüüp määrab ka ära mis on seda tüüpi atribuudi andmetüüp (tekst, number, kuupäev, valik) ja mis on vaikumisi atribuudi väärtus (kui tegemist on teksti-tüüpi või valiku-tüüpi atribuudiga).

Sellesse tabelisse ei ole selles ülesandes vaja läbi rakenduse andmeid lisada, andmed sisestatakse INSERT—lausetega otse andmebaasi.

 doc_attribute_type	Võtmeväli, autonummerduv
default_selection_id_fk	Vaikumisi valiku id, viit tabelisse [atr_type_selection_value] („atribuudi tüübi valikväärtused“)
type_name	Atribuudi tüübi nimi
data_type_fk	Andmetüüp, viit tabelisse [data_type]
multiple_attributes	Selles ülesandes ei kasuta
default_value_text	Seda tüüpi doku,endi atribuudi vaikumisi väärtus ,võib olla täidetud siis kui data_type=1

doc_catalog

Dokumendi kataloog. Selles tabelis hoitakse dokumendi-kataloogide hierarhiat („puud”). Kõige ülemisel tasemel level=1 ja upper_catalog_fk=0



Andmeid sisestades võiks tekitada maksimaalselt kolmetasemelise kataloogipuu, rohkem pole vaja.

Dokumendi lisamisel võiks teha nii et (uue) dokumendi salvestamisel on alati mingi kataloog kataloogipuust valitud – ja selle kataloogile hakkab siis viitama kirje mis seostab dokumenti selle kataloogiga kus ta on – tabelis [document_doc_catalog]. Uue dokumendi lisamise tuleb tabelisse [document_doc_catalog] seega selline seostav kirje lisada.

Dokumendi kataloogil on kaks andmevälja mille sisu tuleb uuendada siis kui muutub kataloogi sisu või selles kataloogis olevate dokumentide (kui andmebaasi kirjete) sisu.

„content_updated” – kataloogi sisu muutmise aeg , timestamp


„content_updated_by” – kes sisselõigatud töötajatest muutis kataloogi sisu

Neid välju tuleb uuendada siis kui


1. kataloogi lisatakse uus dokument
2. kataloogi tõstetakse olemasolev dokument mõnest teisest kataloogist
3. kustutatakse dokument mis asub selles kataloogis
4. kataloogist tõstetakse dokument ära mõnda teise kataloogi („cut and paste”).
5. salvestatakse dokument mis asub selles kataloogis

NB! Kui peate keeruliseks kasutajaliidesel näidata avatavate harudega kataloogipuud (mingi haru avaneb siis kui tema peal klõpsata) siis võite selle kataloogipuu ka tervikuna , koos kõigi harudega välja joonistada veebilehele (nii nagu ülemisel joonisil on näha).

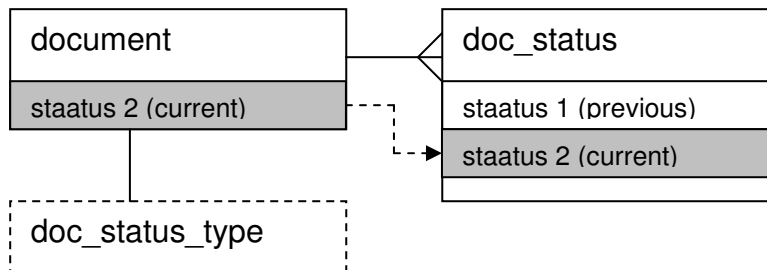
Sellesse tabelisse ei ole selles ülesandes vaja läbi rakenduse andmeid lisada, andmed sisestatakse INSERT—lausetega otse andmebaasi.

 doc_catalog	Võtmeväli, sisu on autonummerduv
catalog_owner_fk	Ei ole selles ülesandes kasutusel
doc_catalog_type_fk	Ei ole selles ülesandes kasutusel
name	Kataloogi nimi
description	Kataloogi kirjeldus, ei ole vaja kasutada
level	Kataloogi tase kataloogipuu, kõige esimesel tasemel level=0
content_updated	Millal kataloogi sisu viimati muutus, timestamp

upper_catalog_fk	Viit ülemkataloogile samasse tabelisse [doc_catalog]
content_updatde_by	Viit sisselloginud töötajale kes muutis viimati kataloogi sisu, viit SUBJEKTIDE allsüsteemi tabelisse [employee]
folder	Kataloogile vastav failikataloog serveri failisüsteemis. Kui teete selles ülesandes dokumendifailide üleslaadimise (mida ei ole otseselt nõutud) siis saab selles väljas hoida failikataloogi „path”-i (sellise andmbaasis hoitava virtuaalse „doc_catalog”-iga võib seostada reaalse failisüsteemis oleva kataloogi milles hoitakse siis sellesse kataloogi uploaditud dokumendifaile).

doc_catalog_type dokumendi kataloogi tüüp. Ei ole selles ülesandes kasutusel	
 doc_catalog_type	
type_name	

doc_status



Dokumendi staatuste ajaloo ja jooksva staatuse tabel.

Dokumendi peaks saama ekraanivormil valida dokumendi staatust, mis need staatuse tüübid on – see pole tähtis, suvalised.

Kui kasutaja nüüd vajutab dokumendi salvestamise nuppu siis tuleb muuta dokumendi staatust dokumendi tabelis ([**document**].doc_status_type_fk) ja lisada jooksva staatuse kohta kirje tabelisse [doc_status], staatuse (status_begin) alguseks salvestamise hetk ja staatuse lõpp määramata (status_end=NULL).

Kui dokumendil juba eelnevalt oli staatus siis tuleb tabelis [doc_status] vana staatus lõpetada – täita ära [**doc_status**].status_end vanal kirjel. Vana staatuse lõpetamise ajaks on uue staatuse algus.

doc_status	Võtmeväli, sisu autonummerdub
document_fk	Viit dokumendile mille staatusega on tegemist, tabelisse [document]
doc_status_type_fk	Viit staatuse tüübile tabelisse [doc_status_type]
status_begin	Staatuse algus. Timestamp tüüpi
status_end	Staatuse lõpp. Täidetakse siis kui tekib uus staatus
created_by	Viit dokumendi sisestanud sisse logitud töötajale, viit SUBJEKTIDE allsüsteemi tabelisse [employee]

doc_status_type

dokumendi seisundi tüüp

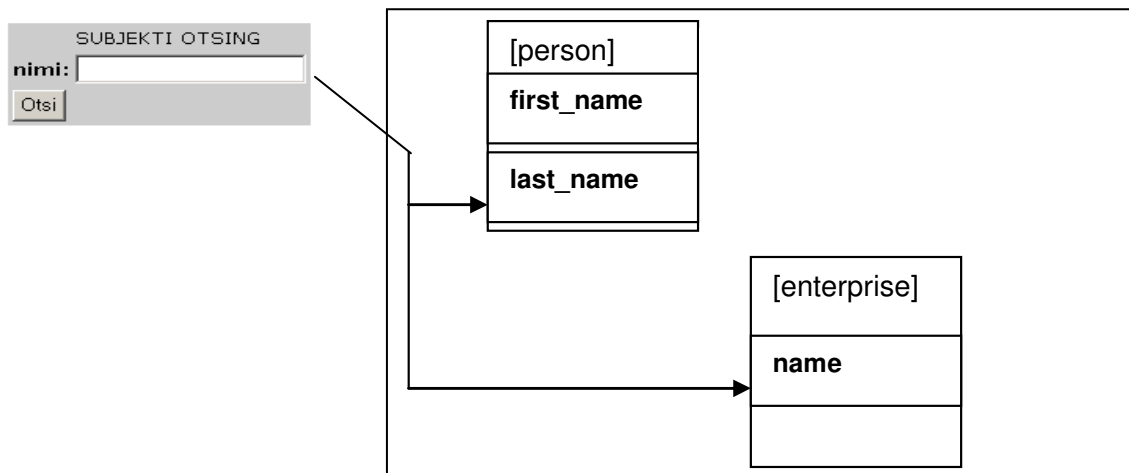
doc_status_type	Võtmeväli, sisu autonummerdub
type_name	Dokumendi seisundi tüübi nimetus

doc_subject

dokumendi seos ettevõtte või isikuga SUBJEKTIDE allsüsteemist.

Dokumendiga peab olema võimalik siduda ettevõtteid ja isikuid.

Selleks et siduda tuleb need isikud või ettevõtted subjektide andmebaasist välja otsida, otsinguvorm võib olla väga lihtne – otsitakse ainult ühe andmevälja järgi.



Selline nime järgi otsing tähendab et otsitakse ettevõtete hulgast selliseid millel „name” on selline nagu otsinguvormil ja isikute hulgast selliseid kelle „last_name” on selline nagu otsinguvormil.


Kui andmebaasist on isik või ettevõtte leitud siis peaks saama teda seostada dokumendiga. Seostamisel sisestatakse kirje tabelisse **[doc_subject]** . Seosed peab saama ka kustutada.

Seosel on mingi tähendus, roll, seda rolli saab seose loomisel määrata , rollid on tabelis **[doc_subject_relation_type]**


seose tüüp: autor
autor
tema kohta
talle saadetud


Olemasolevate seoste muutmise võimalust pole vaja teha (ainus asi mis seal muuta saaks oleks seose roll ja välja „note” sisu. Aga pole vaja seda muuta.)

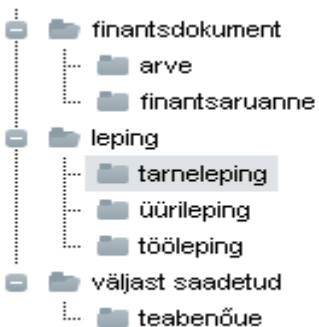
Kui seos lisatakse tabelisse **[doc_subject]** peab rakendus kirje välja „doc_subject_type_fk” panema väärtuseks 1 või 2 – sõltuvalt sellest kas valitud subjekt oli ettevõtte **[enterprise]** või isik **[person]**


 doc_subject	Võtmeväli, sisu autonummerdub
doc_subject_relation_type_fk	Viit dokumendi ja subjekti vahelise seose tüübile tabelisse [doc_subject_relation_type]
doc_subject_type_fk	Viit subjekti tüübile (kas isik või ettevõtte) tabelisse [doc_subject_type]

subject_fk	Viit subjektidel subjektide allsüsteemi tabelisse [person] või [enterprise] (kummale tabelile viidatakse see sõltub subjekti tüübist väljas „doc_subject_type_fk”)
document_fk	Viit dokumendile mis on subjektiga seotud, viit tabelisse [document]
note	Märkus, mingi suvaline tekst selle seose kohta

doc_subject_relation_type	
dokumendi ja subjekti vahelise seose tüüp	
 doc_subject_relation_type	Võtmeväli, sisu on autonummerduv
type_name	Seose tüübi nimi

doc_subject_type	
dokumendi subjekti tüüp. Näitab kas seosetabeli [doc_subject] kirje viitab SUBJEKTIDE allsüsteemi isikule [person] või ettevõttele [enterprise]	
 doc_subject_type	Võtmeväli, sisu ei ole autonummerduv
type_name	subjekti tüübi nimi, „ettevõtte”, „isik”


doc_type	
dokumendi tüüp. Kahetasemeline hierarhia, ülemtüübid ja alamtüübid. Konkreetse dokumendi tüübiks sobib teise taseme tüüp (level=2).	
	
<p>Määratakse dokumendi lisamise ajal – kas enne dokumendi lisamise vormi kuvamist (see tähendab – enne kui rakendus näitab dokumendi lisamise vormi küsib ta kasutaja käest millist tüüpi dokumenti ta tahab sisestada) või dokumendi lisamise vormil .</p> <p>NB! Kuna dokumendi lisamise ekraanivorm genereeritakse valitud dokumendi tüübi alusel siis tuleb see ekraanivorm ilmselt uuesti genereerida/kuvada kui kasutaja valib ja vahetab lisamise vormil dokumendi tüüpi.</p>	

dokumendi tüüp:	finantsdokument ▼
Lisa uus dokument	finantsdokument ...arve ...finantsaruanne leping ...tarneleping ...üürileping
 doc_type	Võtmeväli, sisu autonummerdub
type_name	Dokumendi tüübi nimi
level	Dokumendi tüübi tase (level=1 – ülemtüüp, level=2 – mingi ülemtüübi alamtüüp)
super_type_fk	Viit ülemtüübile samasse tabelisse [doc_type] . Kui level=1 siis väärtuseks 0

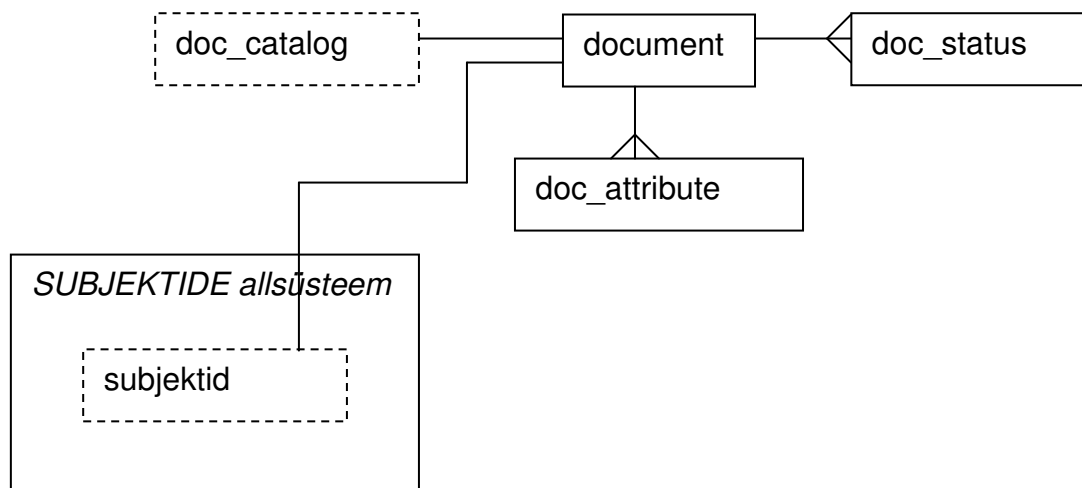
doc_type_attribute

„dokumendi tüübi atribuut (atribuudi tüüp)”, seosetabel

Näitab millised atribuudi tüübid on seotud millise dokumendi tüübiga. Kui rakendus siestab tabelisse [document uue dokumendi siis peab rakendus sellest tabelist lugema millised atribuudi tüübid sellisel dokumendi tüübil on ja sisestama vastava arvu kirjeid tabelisse **[doc_attribute]** .

 doc_type_attribute	Võtmeväli, sisu on autonummerdub
doc_attribute_type_fk	Viit dokumendi atribuudi tüübile tabelisse [doc_attribute_type]
doc_type_fk	Viit dokumendi tüübile tabelisse [doc_type]
orderby	Dokumendi atribuudi järjestus ekraanivormil. Selle välja sisu kopeeritakse tabelisse [doc_attribute].orderby kui seda tüüpi dokumendi atribuut andmebaasi lisatakse
required	Kas seda tüüpi atribuudi väärtus peab kindlasti olema täidetud (selle dokumendi tüübi puhul) Selle välja sisu kopeeritakse tabelisse [doc_attribute].orderby kui seda tüüpi dokumendi atribuut andmebaasi lisatakse
created_by_default	alati 'Y', sisuliselt ei ole selles ülesandes kasutuses


document



Dokument, selle süsteemi põhiobjekt.

Dokumendi kui infoobjekti omadused.

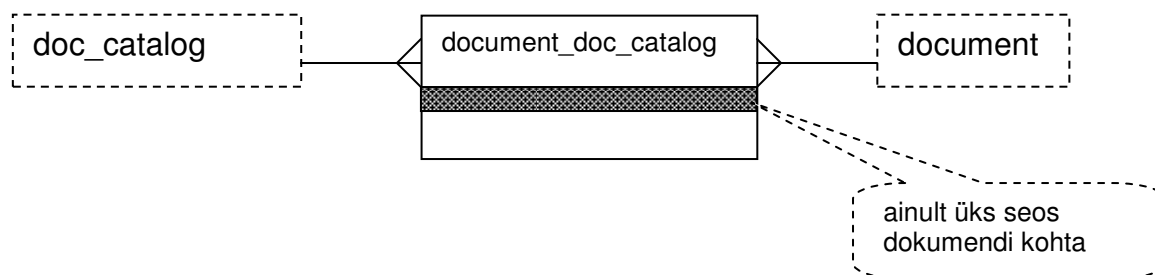
- dokumendil on tüüp (seos tüübiga vahetabelis) ja tüübist sõltuvad atribuudid
- dokument asub dokumendi kataloogis (seos vahetabeli kaudu)
- dokument on seotud subjektidega (isikute ja ettevõtetega) SUBJEKTIDE allsüsteemist
- dokumendil on staatus , staatuste ajalugu hoitakse eraldi tabelis **[doc_status]**

 document	Võtmeväli, sisu autonummerdub
doc_status_type_fk	Dokumendi praegu kehtiv staatus. Kehtivale staatuse vastab ka kirje tabelis [doc_status] mille „status_end” on NULL Staatuse muutmisel lõpetatakse tabelis [doc_status] eelmine staatus („status_end” = NOW()) ja sisestatakse uus jooksev staatus („status_begin”=NOW()). Viit tabelisse [doc_status_type]
doc_nr	Ei ole selles ülesandes kasutusel
name	Dokumendi nimi, suvaline tekst
description	Dokumendi kirjeldus, suvaline tekst
filename	Dokumendi failini. Kui teete rakendusse ka dokumendifailide üleslaadimise (mida ei ole nõutud) siis sellesse välja saab salvestada faili nime. Muidu tühi.
created	Dokumendi andmebaasi sisestamise aeg
created_by	Viit dokumendi sisestanud sisse logitud töötajale,

	viit SUBJEKTIDE allsüsteemi tabelisse [employee]
updated	Dokumendi andmete salvestamise aeg
updated_by	Viit dokumendi andmeid salvestanud sisse logitud töötajale, viit SUBJEKTIDE allsüsteemi tabelisse [employee]

document_doc_catalog


dokumendi kataloog. Näitab millises kataloogis dokument on.



Kuigi selline skeem võimaldab ühte dokumenti siduda mitme dokumendi kataloogiga võtame selles ülesandes eelduseks et dokument saab korraga olla ainult ühes kataloogis.

Dokumendi kataloog salvestatakse siia tabelisse dokumendi lisamisel (rakendus võiks olla tehtud nii et dokumendi lisamisel on mingi kataloog alati valitud)

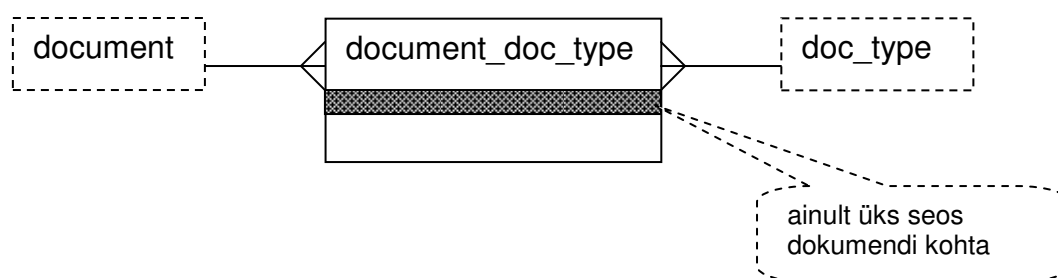
Dokumenti peaks olema võimalik ka ühest kataloogist teise tõsta („cut and paste”), vaata selle kohta täpsemalt altpoolt, funktsionaalsetest nõuetest. Dokumendi tõstmisel ühest kataloogist teise kustutakse vana seos tabelist **[document_doc_catalog]** ja lisatakse uus seos tabelisse **[document_doc_catalog]** mis nüüd viitab juba dokumendi uuele kataloogile. Muudatuste selles tabelis peavad kaasnema ka muudatused dokumendi kataloogis **[doc_catalog]** – kui siit tabelist midagi kustutatakse või lisatakse siis tuleb salvestatud tabelisse **[doc_catalog]** selle kataloogi viimase muutmise aeg ja viimane muutja (sisseloginud kasutaja) sest kataloogi sisu muutus (**[doc_catalog].updated** ja **[doc_catalog].updated_by**)

 document_doc_catalog	Võtmeväli, sisu autonummerdub
doc_catalog_fk	Viit dokumendi kataloogile tabelisse [doc_catalog]
document_fk	Viit dokumendile tabelisse [document]
catalog_time	Dokumendi kataloogi lisamise aeg (aeg millal tehti selle kirje INSERT lause)


document_doc_type

dokumendi tüüp.

Dokumendi tüüp ei ole salvestatud dokumendi tabelisse [document] vaid dokumendi tüübi [doc_type] ja dokumendi vahetabelisse – [document_doc_type]



Kuigi selline skeem võimaldab ühte dokumenti siduda mitme dokumenditüübiga võtame selles ülesandes eelduseks et dokumendil saab olla ainult üks tüüp.

 document_doc_type	Võtmeväli, sisu autonummerdub
doc_type_fk	Viit dokumendi tüübile tabelisse [doc_type]
document_fk	Viit dokumendile mille tüübiga on tegemist, viit tabelisse [document]

3. Rakenduselt oodatav funktsionaalsus

3.1. Funktsionaalsuste loetelu

Dokumendi tüüpide ettevalmistamine INSERT-lausetega:

Lisage INSERT-lausetega dokumendi atribuudi tüüpide tabelisse [doc_attribute_type] **16-20** atribuudi tüüpi. Lisage selliseid tüüpe mis on string tüübid (atribuudi väärtus salvestatakse [doc_attribute] tabelis „value_text” välja) aga lisage ka mõned atribuudi tüübid millel data_type=2 (atribuudi väärtus salvestatakse [doc_attribute] tabelis „value_number” välja) , data_type=3 (atribuudi väärtus salvestatakse [doc_attribute] tabelis „value_date” välja) ja data_type=4 (atribuudi väärtused võetakse valikväärtusena tabelist [atr_type_selection_value])

```
INSERT INTO doc_attribute_type (type_name, data_type_fk,multiple_attributes)
VALUES ('saatjad',1,'N');
```

```
INSERT INTO doc_attribute_type (type_name, data_type_fk,default_selection_id_fk,multiple_attributes)
VALUES ('vastamise tahtaeg',4,4,'N');
```

```
INSERT INTO doc_attribute_type (type_name, data_type_fk,multiple_attributes)
VALUES ('formaad',1,'N');
```

Kui lisasite atribuudi tüüpe millele on valikväärtused siis lisage ka nendele tüüpidele vastavad valikväärtused tabelisse [atr_type_selection_value] .

```
INSERT INTO atr_type_selection_value (doc_attribute_type_fk, value_text,orderby)
VALUES (2,'kahe paeva jooksul',1);
```

```
INSERT INTO atr_type_selection_value (doc_attribute_type_fk, value_text,orderby)
VALUES (2,'nadala jooksul',2);
```

```
INSERT INTO atr_type_selection_value (doc_attribute_type_fk, value_text,orderby)
VALUES (2,'kuu aja jooksul',3);
```

```
INSERT INTO atr_type_selection_value (doc_attribute_type_fk, value_text,orderby)
VALUES (2,'maaramata',4);
```

Rakenduse töö ettevalmistamiseks valige nüüd välja vähemalt 4 dokumendi tüüpi ([doc_type]) ja siduge iga tüübiga vähemalt **4-5** atribuudi tüüpi (andmebaasis tähendab see INSERT lauseid [doc_type_attribute] tabelisse). Sisestage andmed nii et erinevatel dokumendi tüüpidel vähemalt mõned atribuudi tüübid kattuvad .

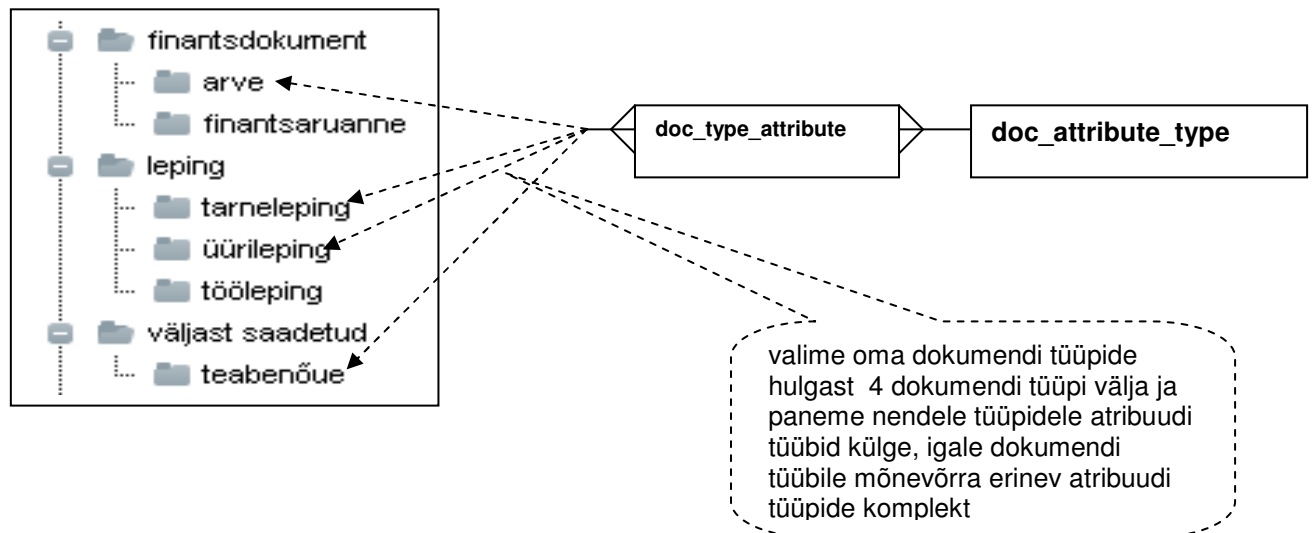
```
INSERT INTO doc_type_attribute (doc_type_fk, doc_attribute_type_fk,created_by_default,
orderby,required)
VALUES (10,1,'Y',1,'N');
```

```
/* doc_attribute_type-fk=2 (vastamise tahtaeg) */
INSERT INTO doc_type_attribute (doc_type_fk, doc_attribute_type_fk,created_by_default,
orderby,required)
```

```
VALUES (10,2,'Y',2,'Y');
```

```
/* doc_attribute_type-fk=3 (dokumendi formaadi kohta mingid tapsemad andmed) */  
INSERT INTO doc_type_attribute (doc_type_fk, doc_attribute_type_fk, created_by_default,  
orderby, required)  
VALUES (10,3,'Y',3,'N');
```

jne.



Uute dokumentide sisestamine , sisestatud dokumentide andmete muutmine ja dokumentide kustutamine.

Dokumentide sisestamisel eeldame et:

- sisestamisel on alati määratud dokumendi tüüp, sellest tüübist sõltub dokumendi sisestamise ja muutmise vormi sisu
- sisestamisel on alati valitud dokumendi kataloog

Dokumendi andmete muutmisel dokumendi tüüpi muuta ei saa, „read-only”

Dokumenti peab saama kustutada, kustutamisel kustutakse ära ka kõik dokumendiga seotud kirjed atribuutide, staatuste, kataloogi-seoste ja subjekti-seoste tabelitest.

Dokumentid lisamisel, kustutamisel ja andmete salvestamisel tuleb muuta dokumendi kataloogi andmeid tabelis [doc_catalog] et salvestada kataloogi sisu muutmise aeg ja viimane muutja.

Dokumendi lisamise ja andmete salvestamise kohta vaata kindlasti tabeli [doc-attribute] selgitust.

Dokumentide seostamine subjektide SUBJEKTIDE allsüsteemist.

NB! Need kes teevad seda ülesannet üksi võivad selle punkti vahele jätta.

Dokumentile peab saama lisada seoseid SUBJEKTIDE allsüsteemi isikutega [person] ja ettevõtetega [enterprise].

Seose loomisel peab saama määrata seose tüüpi (seose tähendus, semantika) ja peab saama sisestada märkust seose kohta („note”).

Dokumendi seoseid subjektidega peab saama kustutada.

Dokumendi seoste andmete muutmist pole vaja teha.

Dokumentide otsing

Dokumentide otsinguvormil on kaks võimalikku olekut:

1. siis kui mingi dokumendi tüüp (mille seast otsida) on valimata
2. siis kui dokumendi tüüp on valitud

Dokumendi tüüp valimata:

DOKUMENDI OTSING

id:

nimi:

kirjeldus:

viimane muutja:

kataloogi nimi:

subjekt:

dokumendi staatus:

----- atribuutidest -----

atribuudi väärtus:

dokumendi tüüp:

[document]

* „description” sisaldab: „suusad” **AND**
* viimase muutja [person].last_name LIKE „juhan%” **AND**
* „doc_status_type” = „vastu võetud” **AND**

[doc_attribute] **AND**

Dokumendi atribuudi

„value_text” või
„value_number” või
„value_date”
sisaldab
väärtust „2011”

(„value_number” väli ei saa arusaadavatel põhjustel sellist väärtus antud näites sisaldada)

Otsida tuleks:

- dokumendi (andmebaasi) id järgi
- dokumendi nime järgi
- dokumendi kirjelduse järgi
- viimase muutja perekonnanime järgi ([employee]->[person].last_name)
- kataloogi nime järgi kus dokument asub
- dokumentidega seotud subjektide nime järgi ([enterprise].name, [person].last_name)
- dokumendi staatuse järgi

- dokumendi atribuudi väärtuste järgi. Kuna atribuudi tüüpi pole siin otsinguvormil määratud siis tuleks vaadata kõikide selle dokumentidega seotud atribuudi andmetesse, väljadesse „value_text”, „value_number” ja „value_date”. Millistest atribuudi vääryuse väljatest otsime see sõltub nüüd mõnevõrra ka sisestatud atribuudi väärtusest – kui sisestatud on „tallinn” siis ei ole ilmslet mõtet vaadata atribuute mille sisu on number-tüüpi.

NB! Kui selle päringu atribuutide osa ja eriti erinevatel juhtudel erinevatest atribuudi andmeväljadest otsimine tundub teile keeruline siis tehke lihtsamalt – otsige ainult „value_text” väljast. Te võite atribuudi andmete salvestamisel „value_number” ja „value_date” väljade sisu alati dubleerida ka „value_text” välja ja nii saate te otsinguga sellest tabelist andmeid kätte kui vaatate alati ainult „value_text” välja sisse. Aga siis peate mõnevõrra täiendama atribuudi andmete salvestamist.

Dokumendi tüüp valitud:

Kui otsinguvormil on dokumendi tüüp valitud siis otsinguvormi atribuutide osa muutub – nüüd on teada mis tüüpi atribuudid antud tüüpi dokumendil on ja saab nende atribuutide järgi ka otsingut teha.

Muu otsinguloogika jääb samaks mis dokumendi tüübita otsingu puhul, atribuutide tabelis [doc_attribute] andmete otsimise loogika aga muutub täpsemaks:

DOKUMENDI OTSING

id:

nimi:

kirjeldus:

viimane muutja:

kataloogi nimi:

subjekt:

dokumendi staatus:

----- atribuutidest -----

saatjad:

vastamise tähtaeg:

formaad:

dokumendi tüüp:

[doc_attribute]
AND
 („value_text” sisaldab „Juhan” **AND** atribuudi tüüp on „saatjad”)

AND
 („atr_type_selection_value_fk =4 **AND** atribuudi tüüp on „vastamise tähtaeg”)

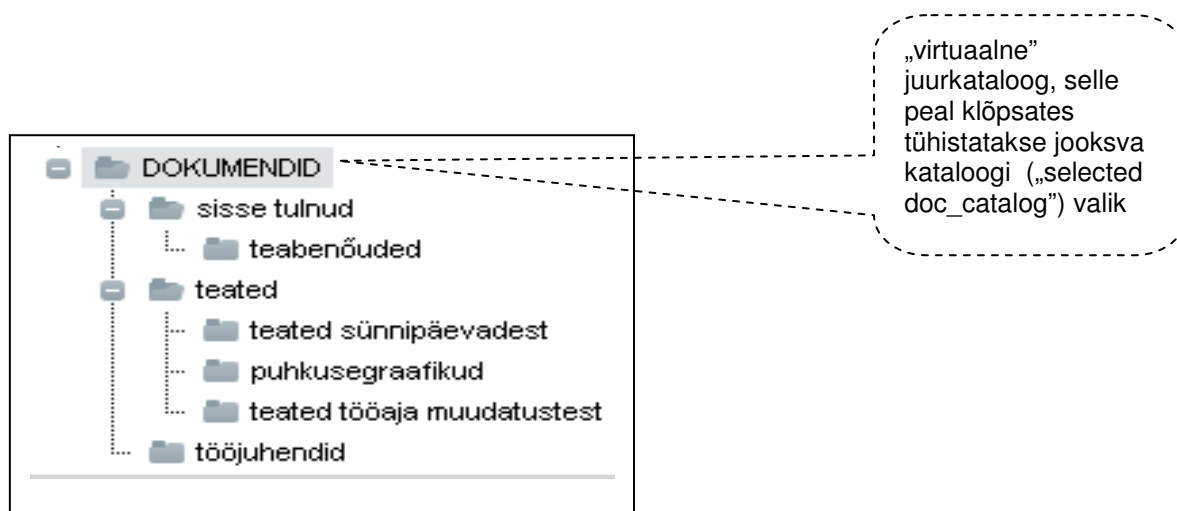
AND
 („value_text” sisaldab „RTF” **AND** atribuudi tüüp on „formaad”)

Tekstiotsinguid sisaldavad päringud peaksid üldjuhul olema tõstutundetud mis tähendab et PostgreSQL-i SELECT lauseid tuleb täiendada UPPER() või LOWER(9 funktsioonidega kuna vaikimis on PostgreSQL-i LIKE päring tõstutundlik. PostgreSQL-i täistekstipäring (full-text search) ei ole tõstutundlik)

Kui kataloogipuust on dokumendi kataloog valitud:

Siis võiks otsing vaikimisi toimuda selle valitud dokumendi kataloogi dokumentide hulgast – kui otsinguvormi väli „kataloogi nimi” on täitmata.

Kataloogipuus peaks olema võimalik vajutada mingile juurharule („/root”) mis tühistaks kataloogi valiku kasutajaliidesel – siis saaksid otsingud toimuda üle kõikide kataloogide (see tähendab sellisel juhul et kataloogi arvestatakse ainult siis kui kasutaja täidab „kataloogi nimi” välja, jooksvat kataloogi ei ole valitud)



Kuidas otsida:

Kui tegemist on selliste tekstiväljadega mis sisaldavad ühte stringi (dokumendikataloogi nimi, toote kood, kliendi nimi, ...) siis võiks otsida sõna alguse järgi , LIKE-tüüpi päringuga. Kuna PostgreSQL-i LIKE-otsing on vaikimisi tõstutundlik (case-sensitive) siis on mõistlik WHERE-klausli mõlemad pooled näiteks UPPER-funktsiooniga teisendada, tekstiotsingus me tavaliselt ei taha tõstutundlikkust.

Tootja kood:
WHERE UPPER(producer_code) LIKE
UPPER('6M76P%')

Kui tegemist on selliste tekstiväljadega mis sisaldavat pikemat teksti (mitu sõna) siis võiks otsida täistekstiotsinguga mis tähendab et otsitakse kas otsitavad sõnad sisalduvad väljas. Näiteks aadressid, märkused, tegevusalad vms.

Atribuut:
WHERE (to_tsvector(SA.value_text) @@
to_tsquery('800x250'))

Kui tegemist on kuupäevade ja numbritega (summad, aastaarvud) mille puhul on mõistlik otsingul ette anda vahemik siis võiks seda teha (kuid kohustust nii teha ei ole). Seega võib teha nii:

Müügihind vahemikus:
WHERE sale_price >= 100 AND sale_price <= 200

Aga võib teha ka ni:

allahindluse %:

10

WHERE discount_xtra= 10

Otsinguvormi sisestatud andmete säilitame pärast otsingu nupule vajutamist:

Otsinguvormile sisestatud andmed peaksid olema sellel otsinguvormil alles ka pärast nupule „Otsi” vajutamist (pärast veebilehe ümberlaadumist). Otsinguvorm ise võiks ka muidugi ekraanile alles jääda. Et uue otsingu tegemiseks ei

Dokumentide tõstmine puhvrissse ja puhvrist liigutamine uude kataloogi („cut and paste”).

Peaks olema korraga mitut dokumenti liigutada ühest kataloogist teise meetodil mis on sarnane näiteks failisüsteemis failide liigutamisega „clipboard”-i abil, „cut and paste” meetodil.

1. On võimalik liikudes erinevates kataloogides märkida ära sleles kataloogis olevadi dokumente ja vajutada mingit nuppu või linki „Tõsta puhvrissse” (midagi analoogilist siis „cut”-käsuga kui me failide liigutamise analoogiat operatsioonisüsteemist meelde tuletame).



dok. nr.	dokumendi nimi	kustuta
45	2010 novembri lyh. tööpäevad	<input type="checkbox"/>
102	detsembri töö III osakonnas	<input checked="" type="checkbox"/>
108	vabad päevad jaanuaris	<input checked="" type="checkbox"/>

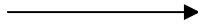
Puhvrissse tõstmisel dokumentide kataloogi-seostega tabelis [document_doc_catalog] veel midagi ei juhtu (nii nagu „cut”-i tegemisel failisüsteemis faile veel tegelikult kuhugi ei liigutata). Informatsioon selle kohta et mingist kataloogist mingid dokumendid tõstetu puhvrissse tuleb aga meeles pidada rakenduses ja tuleb meeles pidada nii et kasutaja võib rakenduses teha muid tegevusi, liiguda suvalistele lehtedele ja vormidele rakenduses – info puhvri sisu kohta jääb alles. See tähendab et puhvri sisu tuleb hoida näiteks sessiooni objektis (vaadake sessioonihalduse ja autentimise teemalisi näiteid ja materjale) või siis mingis (teie enda poolt andmebaasi juurde tehtud ajutiste andmete hoidmise jaoks mõeldud) andmebaasi tabelis. Rakendusserveri sessiooniobjektis puhvri infot hoida on lihtsam kui andmebaasi tabelis.

Nüüd peab kasutajal olema võimalik liikuda ka teistesse kataloogipuu harudesse ja ka sealt tõsta väljavalitud dokumente puhvrissse lisaks.

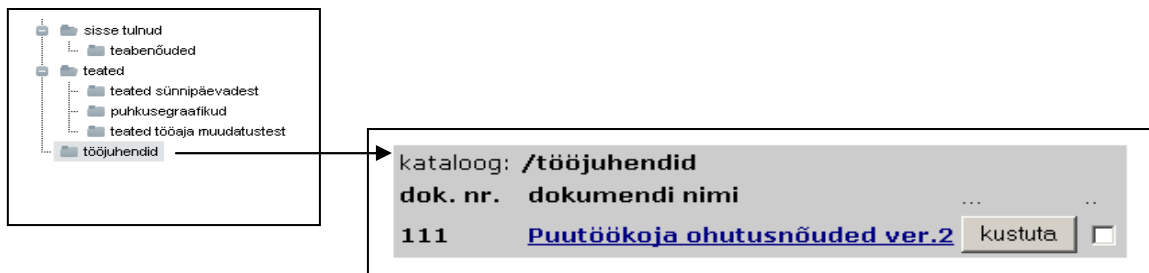


dok. nr.	dokumendi nimi	kustuta
55	Juhani avaldus	<input checked="" type="checkbox"/>

see ka puhvrissse



2. Kui dokumendid on kõik „kokku korjatud” kataloogidest siis valib kasutaja kataloogi kuhu ta tahab dokumente puhvrast kopeerida:



Seal on praegu üks dokument.

3. Soovi korral peaks kasutaja saama alati vaadata puhvri sisu (mingi valik kuskil menüüs – „vaata puhvrissse”)



4. Kui kasutaja nüüd (õiges kataloogis olles) vajutab puhvri vormi peal nuppu „Tõsta jooksvasse kataloogi” siis kopeerib rakendus kõik puhvriss olevad dokumendid sellesse jooksvasse kataloogi.

Andmebaasioperatsioonide mõttes tähendab see muudatusi tabelites **[document_doc_catalog]** ja **[doc_catalog]**

Tabelist **[document_doc_catalog]** võetakse maha kirjed dokumentide vanade seostega ja lisatakse uued kirjed mis seovad dokumendid selle kataloogiga kuhu nad tõsteti.

Tabelis **[doc_catalog]** tuleb muuta kõigi asjasse segatud dokumendikataloogide sisu muutmise aega ja sisu viimast muutjad („content_updated” ja „content_updatde_by”) sest kataloogide sisu muutus.

5. Avatud puhvri vomril peaks saama puhvrist ka dokumente ära kustutada
- üksikult, dokumendi kaupa
 - dokumentide valikut (valitud „check box”-id)
 - kõik dokumendid puhvrist maha

Puhvrist kustutamine ei muuda dokumentide seisu andmebaas, andmebaasi jäävad alles, ainult puhvrist (ehk siis sessiooniobjektist või ajutisi andmeid hoidvast tabelist) kustutatakse.

Kataloogi „tööjuhendid” uus sisu peale failide liigutamist:

108 /teated/teated tööaja muudatustest/vabad pä

tõsta jooksvasse kataloogi (move)

↓

kataloog: /tööjuhendid

dok. nr.	dokumendi nimi
111	Puutöökoja ohutusnõuded ver.2	kustuta	<input type="checkbox"/>
55	Juhani avaldus	kustuta	<input type="checkbox"/>
102	detsembri töö III osakonnas	kustuta	<input type="checkbox"/>
108	vabad päevad jaanuaris	kustuta	<input type="checkbox"/>

Dokumendifailide upload brauserist ja failide alla-laadimine – pole vaja teha.

Dokumendifailide upload”-i ei ole selles rakenduses vaja teha. Kui selle siiski lisate rakendusse (nii et saab dokumendifaili uploadida brauserist ja uploaditud faili pärast alla tõmmata) saate kindlasti „mitteformaalseid” plusspunkte. Aga tegeleda selle teemaga oleks mõtet (kui üldse) alles siis kui kõik ülejäänud osad on tehtud.

Töötaja autentimine. Sisse- ja väljalogimine.

Rakendust saab kasutada peale sisselogimist. Kasutajanime ja parooli kontrollimiseks (autentimispäring) kasutatakse andmeid SUBJEKT-i skeemi tabelis **[user_account]**.

Kõik ülesande variandid kasutavad sisselogimiseks tabelite **[employee]** ja **[user_account]** andmeid – kui kasutaja sisestab kasutajanime ja parooli siis otsitakse seda asutajanime ja parooli tabelist **[user_account]** ja kui leitakse siis leitakse ka sellele kasutajakontole vastava **[employee]** identifikaator (**[user_account].employee_fk**). Sisselogimisel leitud viidet „employee” tabeli kirjele kasutatakse vajadusel rakenduses „created_by” ja „updated_by”väljade täitmiseks.

Üldiselt on soovitatav parooli väljas („passwd”) hoida paroole krüpteeritult (nt. MD5).

```
/* tootaja kasutajanimega 'marten' logib süsteemi sisse */  
SELECT E.employee,UA.user_account, P.first_name, P.last_name FROM employee E INNER  
JOIN user_account UA ON E.employee = UA.subject_fk  
INNER JOIN person P ON E.person_fk = P.person  
WHERE UA.subject_type_fk = 3 AND UA.username='marten' AND  
UA.passw='37b4931088193a73b6561bae19bf06d9';
```

Rakenduses peab olema tehtud ka väljalogimine.

3.2. Ärireeglid ja andmete kontrollid. 10 reeglit.

Praktikaülesande üldistes nõuetes on kirjas et tuleb rakendusse teha ka „validaator” tüüpi objektid (klassid) mis kontrollivad sisendandmete õigsust. Millised need reeglid on millele andmed peavad vastama – need mõelge ise välja.

Mõelge välja vähemalt 10 reeglit millele sisendandmed peavad vastama ja kontrollige rakenduses neid reegleid (kui kasutajad andmeid sisestavad või muudavad) ning andke kasutajale ekraanivormidel teada kui sisend-andmed neid kontrolli reegleid ei rahulda. Mõned reeglid on selle praktikatöö juhendis erinevates kohtades ka juba toodud, neid võib ka 10 hulka arvestada.

Näiteks:

- millised väljad ei tohi olla tühjad
- milliseid andmeid ei tohi kustutada kuna nendele andmetele viidatakse teistes tabelites
- millised summad või kuupäevad peavad jääma teatud piiridesse
- millal võib tellimusele teha arvet
- millal võib tellimust kustutada
- ja nii edasi, ja nii edasi.

4. Mis teha siis kui mingi osa ülesande funktsionaalsusest tundub ebaselge, kui midagi ei ole täpselt kirjeldatud?

Kui midagi jääb ebaselgeks siis võib alati näiteks harjutustunnis küsida aga võib ka ise otsustada ebaselgetes situatsioonides kuidas rakendus peaks toimima. Ilmselt on mõistlik otsustada nii et peaks vähem programmeerima, et rakenduse toimimine oleks lihtsam .