



# Access

## Andmebaaside loomine

**Alo Linntam**

**Tabelid • Seosed tabelite vahel**  
**Vormid • Vormide sidestamine tabelitega**  
**Päringud • Aruanded • Andmebaasi turvamine**  
**Andmebaasi paigaldamine arvutivõrku**

### Sissejuhatus

Kõige paremini õpib näidete varal. Käesolev raamat võimaldab lugejal kõiki kirjeldatud näidis-andmebaase samm-sammult ise luua. Abiteavet lugedes ei tohiks siis enam raske olla luua edasi oma andmebaase.

Iga järgmine raamatus käsitletud andmebaas on üles ehitatud eelmise baasil. Esimene neist on lihtsaim: ainult ühest tabelist koosnev andmebaas. Seega ei nõuta lugejalt mitte mingisugust eelnevast kogemust. Oluline on ainult hea motiveeritus. Kui ka siis mõni asi selgusetuks jääb, saab sellest süüdistada ainult autorit, kes on küll püüdnud iga pisimagi momendi hästi selgeks tsha, kuid pole järelikult oma ülesandtega toime tulnud.

Kuna raamatus on lõppkokkuvõttes ainult üks läbiv näidisandmebas, siis huvitab lugejat kindlasti see ka elektroonilisel kujul omandada. Seda tohib igaüks, kes käesoleva raamatu on ostnud, oma arvutisse vabalt maha laadida internetileheküljelt <http://www.online.ee/~alo>. Vajadusel on lihtne autoriga ühendust võtta ka e-posti kaudu aadressil [alo@online.ee](mailto:alo@online.ee).

### Access'i põhiolemus

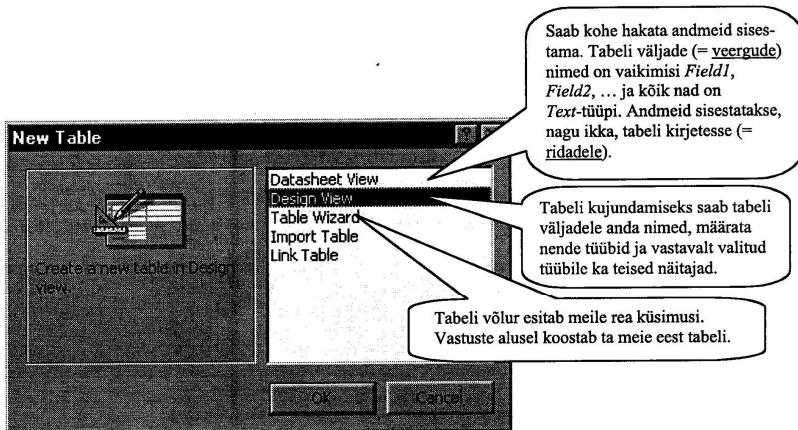
Nii nagu näiteks *Microsoft Word* tegeleb dokumentidega (= DOC-failidega) – loob ja muudab, avab ja salvestab neid –, nii tegeleb *Microsoft Access* andmebaasidega (ehk MDB-failidega; MDB = *Microsoft Database*). Andmebaas koosneb järgmisi tüüpi objektidest.

- **Tabelid (tables).** Tabelid on igasuguse andmebaasi vundament, kõik ülejäänu on ainult pealisehitus – tableid „teenindavad organid“, et nendesse võimalikult mugavalt andmeid sisestada ja neist kätte saada.
- **Päringud (queries).** Päring on eeskiri, mis määrab, millistest tabelitest missuguseid (näiteks teatud tingimustele vastavaid) andmeid on vaja esitada. Seeja, päringu väljund on omakorda tabel, kuid juba toletatud tabel, mitte fundamentaalne, ning seetõttu ei kuulu tabelite rubriiki.
- **Vormid (forms).** Vormid on dialoogikastid, milles olevate kontrollelementide (nt. tekstikast või nupp) abil saab tabelite andmeid mugavalt esitata ja muuta.
- **Aruanded (reports).** Aruanded nimetatakse tabelites esinevate andmete esitust dokumenti kujul, mis on väga hea igasuguste kokkuvõtete ja ülevaadete tegemiseks. Aruandest saab ühe näpuliigutusega tekitada *MS Word*'i dokumendi.
- **Makrod (macros).** Makroks nimetatakse käskude jada. Selle asemel et iga kord ühtesid ja samu käsklusi jagada, võib need makrona vormistada ning lasta neid automaatselt täita.
- **Moodulid (modules).** Mooduliks nimetatakse *Visual Basic*-keelse programmiteksti lehekülge.

# 1 Ainult ühte tabelit sisaldav andmebaas

Loome uue tühja andmebaasi.

1. Käivitame Microsoft Access'i.
2. Menüküsaga File → New Database → leht General → Blank Database loome uue andmebaasis faili, millele paneme nimeks nt. MÜÜK.MDB. Pärast nupul Create klõpsutamist luuakse uus andmebaas, mille peaaken kuvatakse. Kõik selle andmebaasi liikmesobjektid – tabelid, pärngud, vormmid, aruanded, makrod ja moodulid – on tühjad.
3. Loome esimese tabeli: klõpsame nupul New, kui ees on leht Tables.
4. Access soovib teada, mida me uue tabeliga ette võtta tahame:



*Import Table* võimaldab sisse tuua mõnes teises andmebaasis oleva tabeli koopia, *Link Table* aga haakida teises andmebaasis olev tabel käesoleva tabeli külge nii, et edaspidi, toimetades käesolevat tabelit toimetatakse tegelikult külgehaagitud tabelit. Lingitud tabel on virtuaalselt meie andmebaasis, tööliselt aga mujal. Andmebaas, milles ta asub, ei pruugi olla samaaegselt avatud.

Valime *Design View* ning hakkame sisestama tabeli väljade (= veergude) nimetusi, nende tüüpe ja muid näitajaid.

Olgu esimese välja nimetuseks „Mitmes tehing” (välja nimetus saab olla maksimaalselt 64 märki pikk) ning tüübiks *AutoNumber*. Viimane määrab, et edaspidi hakkab Access ise hoolitsema sellele väljale väärustele andmise eest, ja nimelt märkima, mitmenda kirjega on tegemist (iga tehing olgu üles tähendatud parajasti ühes kirjes, s.t. tabeli real). Olemasolevate kirjete kustutamine seejuures numeratsiooni ei muuda: kui meil näiteks on 6 kirjet ja me kustutame mõned neist, on ikkagi järgmisse kirje number 7.

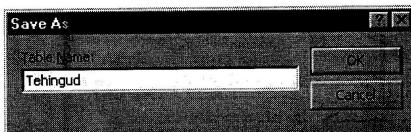
Väli „Eesnimi” on loomulikult *Text*-tüüp; **selle omadus** *Field Size* (= välja suurus) olgu võrdne näiteks 30 märgiga (vaikimisi 50, **maksimaalselt 255**). Teised *Text*-tüüpi välja omadused (*Field Properties*) on järgmised:

Nimetus	Eestikeeline	Omaduse sisu
<i>Format</i>	Formaat	<b>Andmete esitamise</b> vorm <b>sellel</b> tabeli väljal (= veerus).
<i>Input Mask</i>	Sisestusmask	<b>Andmete</b> kuju, millega nad tuleb sisestada <b>sellele</b> tabeli <b>väljale</b> .
<i>Caption</i>	Pealkiri	<b>Väljale</b> vastava tabeli veeru pealkiri. Sama tekst kirjutatakse <b>ka</b> mingil vormil olevale kontrollile (nt. tekstikastile), mis <b>on seotud</b> selle <b>väljaga</b> .
<i>Default Value</i>	Vaikevääratus	<b>Vääratus</b> , mis peab olema vaikimisi kirjas sellel väljal, kui <b>midagi pole veel</b> sisestatud.

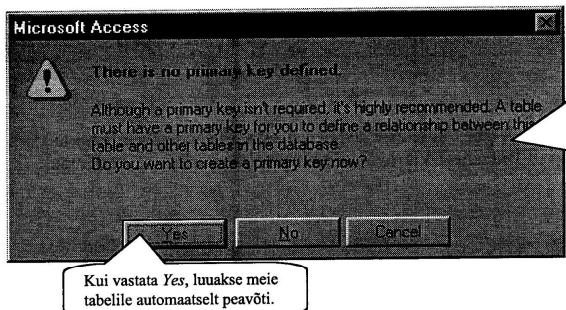
<i>Validation Rule</i>	Väärtustusreegel	Tingimus, millele peab vastama sellele väljale sisestatav tekst.
<i>Validation Text</i>	Väärtustustekst	Hoiatus, mis väljastatakse, kui sisestatav tekst ei vasta väärtustusreegliga määratud nõuetele.
<i>Required</i>	Nõutav	=Yes: kui kirje on loodud, siis ei tohi see väli olla tühi. Kirje aga luuakse niipea, kui vastavale tabeli reale on midagi sisestatud.
<i>Allow Zero Length</i>	Nulliline pikkus lubatav	=No: sellel väljal ei tohi olla nullilise pikkusega tekstilöök, mida saab asetada ainult Visual Basic-keelse programmi abil; muidu reguleerib seda omadus <i>Required</i> .
<i>Indexed</i>	Indekseeritud	=Yes ( <i>Duplicates OK</i> ): sellel väljal olev väärtus määrab üheselt mingi kirje teises tabelis, mis on käesoleva tabeliga seotud ( <i>related</i> ). =Yes ( <i>No Duplicates</i> ): sellel väljal olev väärtus määrab üheselt käesoleva tabeli kirje (seetõttu ei lubata ka korduvaid väärtusi).

Igal väljatüübil on oma omaduste komplekt. Sageli võivad nad jäädä oma vaikeväärtuste juurde. Praegusel juhul muutsimme ainult omadust *Field Size*, aga ka selle oleksime võinud tegemata jäätta.

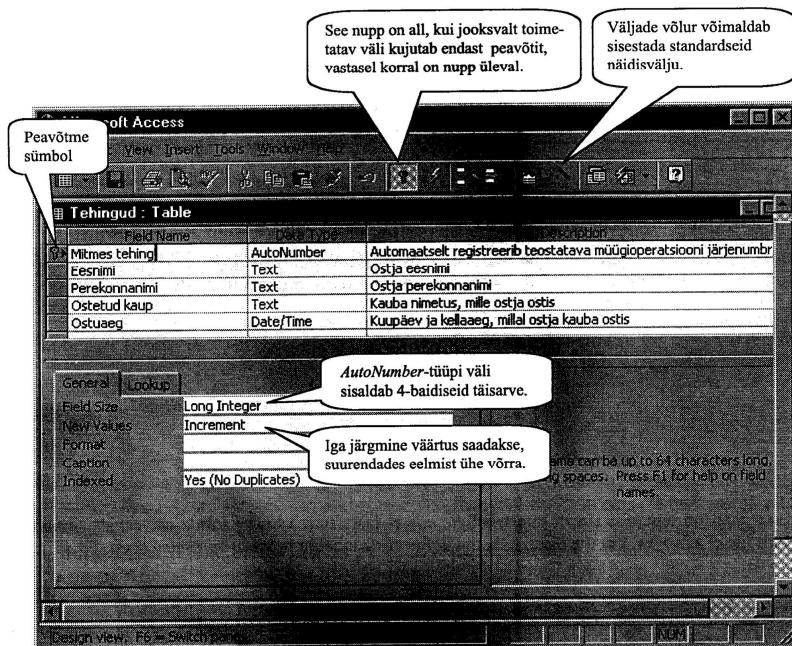
Salvestame tabeli: menüükäsk **File→Save** või vahendiriba (*toolbar*) vastav nupp või kiirkorraldus **Ctrl+S**. Et veel ei ole määratud tabeli nime, siis küsitakse seda meilt:



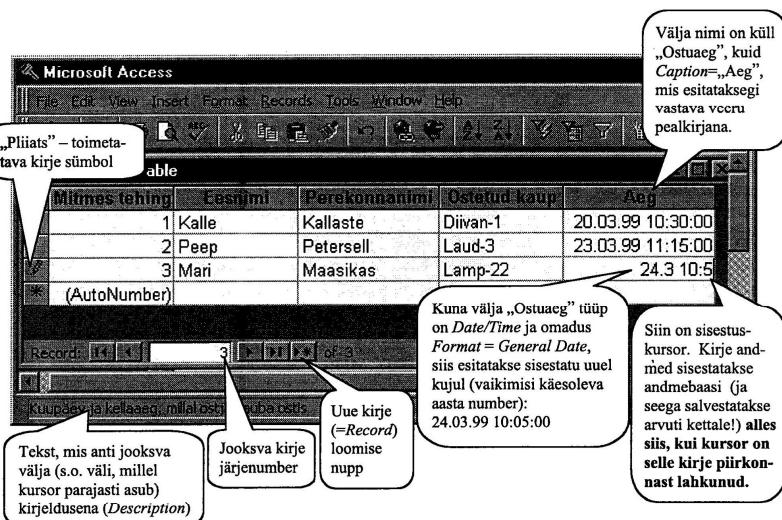
Kuna meil on tegemist täiesti uue tabeliga, siis ei lasta seda esimest korda salvestada, ilma et meilt üle ei küsitaks, kas me töesti ei soovi defineerida tabelile peavõtit (*primary key*):



Peavõti – tabeli indeks – on niisugune tabeli väli, millel olev väärtus määrab üheselt tabeli kirje. Selle välja omadus *Indexed = Yes (No Duplicates)*. Korduvväärtusi ühe ja sama tabeli piires ei lubata. Tavaliselt on peavõtmeks *AutoNumber*-tüüpi väli, kuid võib olla ka näiteks *Text*-tüüpi väli. Tabelil saab olla ainult üks peavõti. Kui ülaatoodud dialoogikastis vastata *Yes*, ositatakse mõne tabeli väljade hulgast väli, millel tütüp on *AutoNumber* (neid ei saa olla rohkem kui üks) ja omistatakse sellele peavõtme staatust. Kui niisugust ei leidu, luuakse selleks otstarbeks automaatselt uus *AutoNumber*-tüüpi „ID”-nimeline väli (ID = identifikaator, määraja). Käesoleval juhul oli meil vajalik väli juba olemas, mistõttu peavõtme õigused antakse väljale „Mitmes tehing”.



Tabel on disainitud. Jääb üle ainult tabeli kirjetesse andmed sisestada ja andmebaas ongi loodud. Selleks valime menüüst **View→Datasheet View** või klöpsutame vahendiriba vastaval nupul.



Ühest tabelist koosnev andmebaas on valmis. Suleme Access'i peaakna (nt. menükäsguga File→Exit). Meilt ei küsita, kas muudetud andmebaasi on vaja ka salvestada või mitte – kõik selle obektid salvestati kettale jooksvalt nende muutmise käigus. Avame sama andmebaasi uesti topelt-klopsuga vastava MDB-faili ikoonil. Automaatselt käivitatakse Access ning me võime veenduda, et kõik sisestatud andmed on säilinud:



### 3 Mitut tabelit ja nendevahelisi seoseid sisaldaandmebaas

Siiiani oli meil ainult üks tabel, „Tehingud”. Oletame, et me soovime andmebaasi salvestada tehtingu teostanud kliendi kohta rohkem andmeid kui ainult ees- ja perekonnanime, nt. aadressi, või ostetud kauba kohta muid andmeid peale nimetuse. Kas me peaksime siis iga kord, kui üks ja sama klient ostma tuleb, sisestama uesti ka tema aadressi, sest me loome ju iga kord uue kirje? Või kas me peaksime alati, kui mingit kaupa ostetakse, uesti sisestama ka selle kauba teised andmed peale nimetuse? Piisaks ju ainult nimetusest, mis on piisavalt täpne selleks, et kõik teised andmed oleksid selleks juba määratud.

Mitme tabeliga andmebaasi luuakse nii selleks, et ühtesid ja samu andmeid kunagi ei dubleeritakse, kui ka selleks, et „üks asi” saaks olla kirjas ka „ainult ühes kohas” – et kui midagi töepooltest muutub, nt. aadress, siis piisaks ainult ühest parandusest, pealegi tuleb andmebaasi maht nii-moodi tunduvalt väiksem.

Samas ei taha me ka, et need tabelid üksteisest täiesti sõltumatud oleksid: me soovime näiteks, et kui me oma tabelis „Tehingud” oleme sisestanud ostja identifikaatori, siis kontrollitakse automaatselt, kas ikka selle ostja andmed on klientide tabelis olemas, ja kui ei ole, siis sunnits meid selle sinna kirjutama, enne kui me oma kirjet salvestada saame. Samuti on ka loomulik, et Access ei lubaks klientide tabelist ära kustutada neid kirjeid, millele viidatakse tabelis „Tehingud”. Et see kõik nii toimiks, tuleb tabelile vahel defineerida seosed (relationships), mis määrasid ära, millise tabeli milline välja missuguse seostatud tabeli (related table) kirjetele viitab.

Et tabelis „Tehingud” on meil väljad „Eesnimi” ja „Perekonnanimi”, mis hakkavad edaspidi paiknema uues tabelis „Kliendid”, siis kustutame need väljad praegusest tabelist ja lisame nende asemel uue number-tüüpi välja „Kliendi ID”. See on väli, millel olev arv hakkab viitama tabeli „Kliendid” vastavale kirjele. Sealt leitakse siis juba õigest kirjest „Eesnimi”, „Perekonnanimi” ja muud andmed.

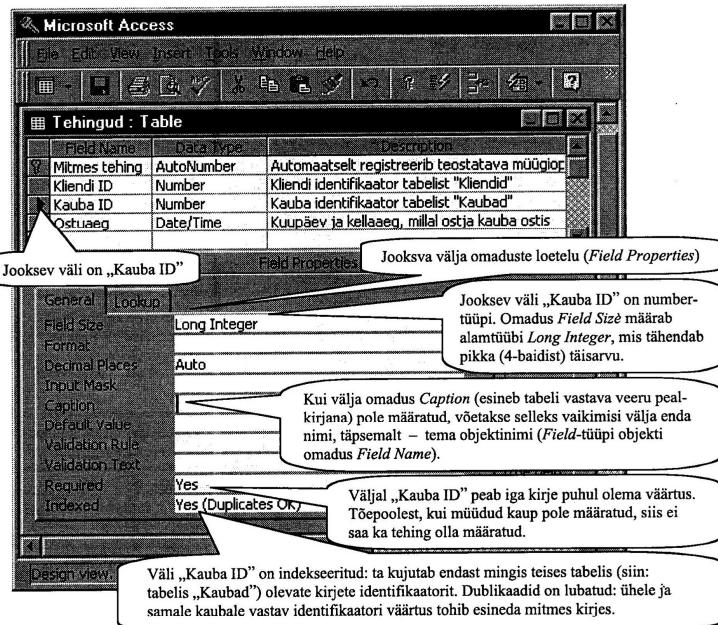
Analoogiliselt asendame tekst-tüüpi välja „Ostetud kaup” number-tüüpi väljaga „Kauba ID”.

Vastav toiming tabeli „Tehingud” kujundusvaates:

- Kustutamiseks** selekteerime väljad „Eesnimi”, „Perekonnanimi” ja „Ostetud kaup” ning vajutame klahvi **Delete**.
- Lisamiseks** viime sisestuskursori sellele reale, mille ette soovime uue rea vahele panna (mis hakkab kirjeldama tabelisse lisatavat uut välja) ning valime menükäsu **Insert→Rows**. Nii sisestame välja „Ostuaeg” kohale uued väljad „Kauba ID” ja „Kliendi ID”.
- Salvestame** tabeli kujunduse.

Seejärel võtame ette tabeli „Tehingud” andmelehevaate (*Datasheet View*) ja sisestame vastloodud väljadele mingid arvud, nt. „Kauba ID”=1,2,3 ja „Kliendi ID”=1,2,3. See on vajalik selleks, et me saaksime seejärel nendele väljadele omistada omaduse *Required* = Yes, mis määrab, et nendel väljadel on väärtsuse olemasolu nõutud iga kirje puhul.

Tulemusena tekib tabelile „Tehingud” järgmine kujundus:



Tabel „Tehingud” on seega modifitseeritud. Esimene uue tabelina loome näiteks klientide tabeli, mille kirjetele hakkab viitama tabeli „Tehingud” välja „Kliendi ID” (s.o. kliendi identifikaator). Seetõttu peab loodava tabeli peavõti samuti kujutama endast kliendi identifikaatorit. On hea tava hoida ka välja nime täpselt samana, „Kliendi ID”. Loodud tabeli võib kokkuvõtlikult esitada järgmiste väljade loendina:

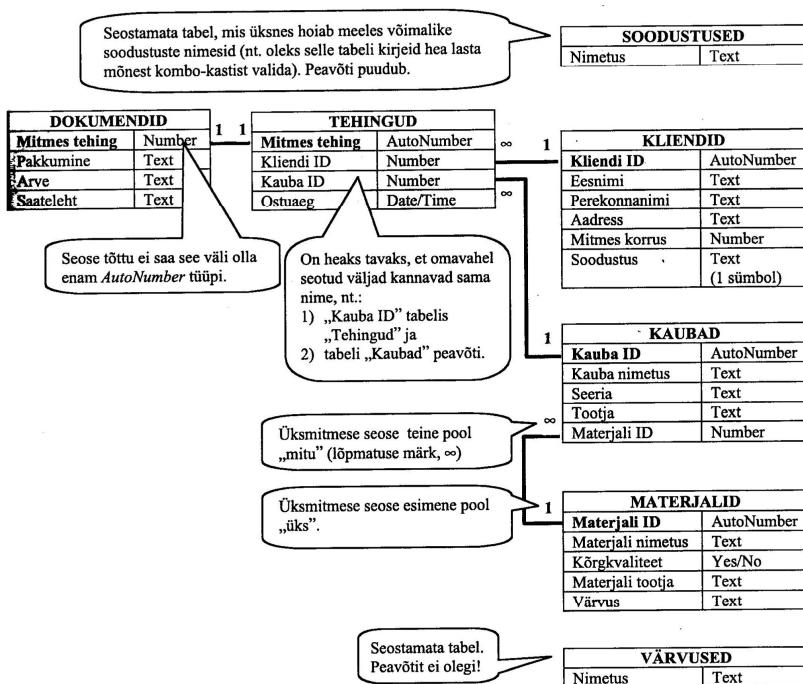
KLIENDID		
Kliendi ID	Data Type	Description
	AutoNumber	Automaatselt registreerib teostatava müügioru
Eesnimi	Text	Kliendi eesnimi
Perekonnanimi	Text	Kliendi perekonnanimi
Aadress	Text	Kliendi aadress
Mitmes korras	Number	Mitmendal korral kliendi elab
Soodustus	Text	Kliendi soodustuse näitaja

Sama klientide tabeli kujundusvaade (Design View) näeb välja nii:

Salvestame tabeli „Kliendid”. Loome uue tabeli „Dokumendid”, mille ülesandeks on meeles pidada tehingutega seotud dokumentide (nt. MS Word'i failide) asukohti ja nimesid.

## Andmebaasi struktuur

Lisame andmebaasi veel tabelaid, kujundame need ja loome niisugused tabelitevahelised seosed, et tulemusena tekiks järgmiste struktuuriga andmebaas:



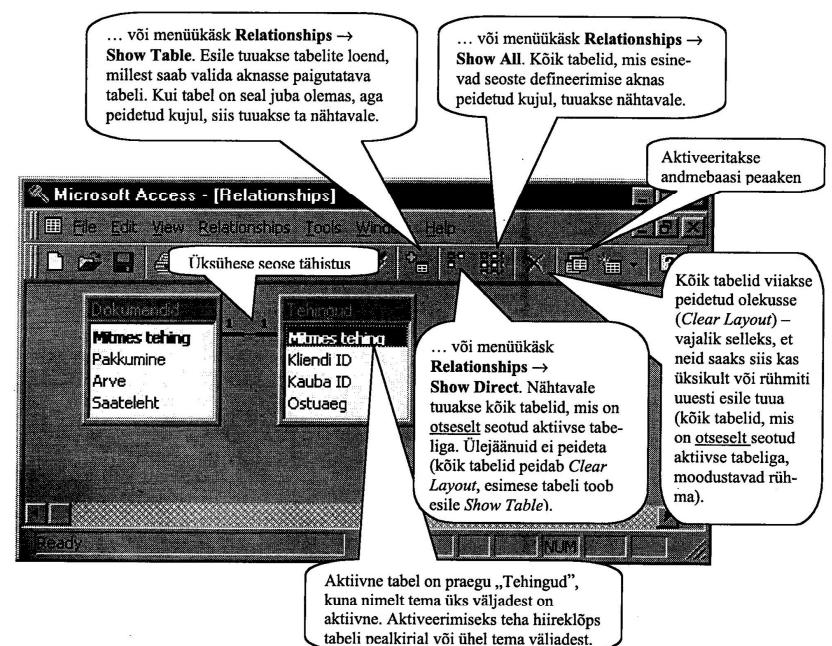
Kuna tabelid ja nendevahelised seosed on igasuguse andmebaasi vundament, millele kõik ülejäänud paramatamult tugineb, siis alati, alustades andmebaasi tegemist, on väga soovitatav kõigepealt visandada analoogiline andmebaasi struktuur.

## Tabelitevaheliste seoste defineerimine

Menüükäsk **Tools → Relationships** avab seoste defineerimise akna **Relationships**. Sama teeb ka vahendiribal **Database** asuv nupp **Relationships**, kuid see on nähtav ainult siis, kui andmebaasi peaaiken on parajasti aktiivne (viimast saab aktiveerida, vajutades klahvi **F11** või valides peamenüü käsu **Window** alt „Müük: Database”).

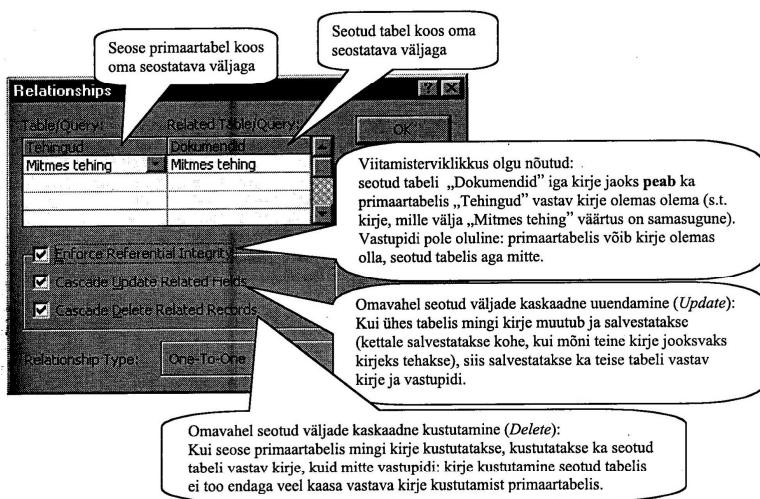
Kogu kunst on ainult selles, et seoste defineerimise aknasse tuleb paigutada seostatavate tabelite kujundid ja seejärel nendevahelised seosed hiire abil määratleda. Seose defineerimiseks tuleb lohistada üks seostatav välja teise peale. Edasi küsitakse meilt loodava seose omadusi. Näiteks, defineerime üksühese vastavuse tabelite „Tehingud” ja „Dokumendid” kirjete vahel, määrates, et

esimese tabeli igale kirjele, mille puhul välja „Mitmes teeing” omab teatud väärust, vastab teise tabeli nimelt see kirje, mille puhul samanimeline välja omab täpselt samasugust väärust:

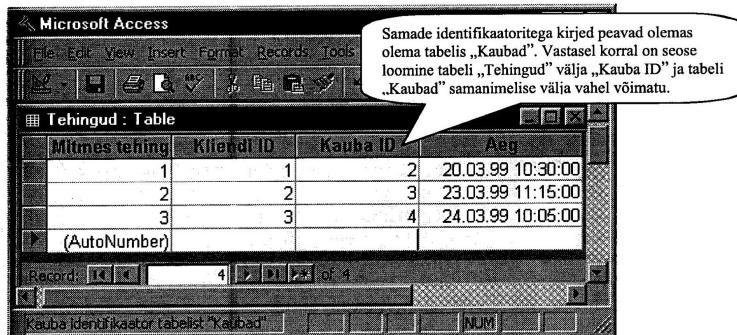


Soovitud seose loomiseks tuleb hiire vasapoolne klahv alla vajutada tabeli „Tehingud” välja „Mitmes teeing” ning vabastada tabeli „Dokumendid” samanimelisel väljal. **Vastupidi ei saa:** selle seose **primaartabeliks (primary table)** on „Tehingud” (kuna tema peavõti „Mitmes teeing” on **AutoNumber**-tüüpi) ja **seotud tabeliks (related table)** on „Dokumendid” (kuna samanimeline peavõti on **Number**-tüüpi: talle ei ole antud voli automaatselt muutuda, vaid ta peab järgima tabeli „Tehingud” peavõtme väärtsi, kui ta tahab seotud olla). Seos tuleb alati vedada **primaartabelilt seotud tabelile**, mitte vastupidi.

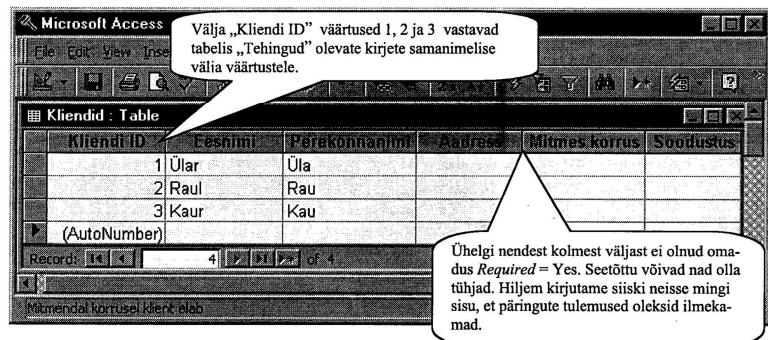
Et mõlema tabeli väljad „Mitmes teeing” on oma tabeli jaoks peavõtmel, tekib üksühene seos (*one-to-one relationship*) tabelite „Tehingud” ja „Dokumendid” kirjete vahel. Seose omadusi saab määra järgmises dialoogikastis:



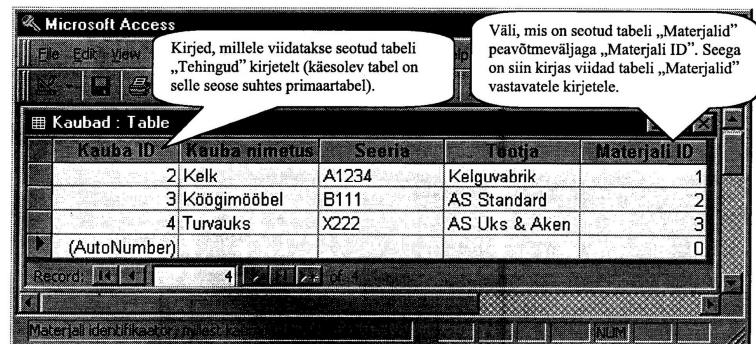
Järgmisena loome seose tabeli „Kliendid“ (primaartabel) ja „Tehingud“ (seostatud tabel) vahel, kusjuures seostatavaks väljaks on mõlemas tabelis „Kliendi ID“. Selleks lohistame nimetatud välja hiire abil primaartabelist seostatud tabeli vastavale väljale, nõudes samuti viitamisterviklike forssseerimist (Enforce Referential Integrity on märgistatud). On oluline, et meie tabelis „Tehingud“ oli 3 kirjet, mis viitasid klientidele identifikaatoritega 1, 2 ja 3.



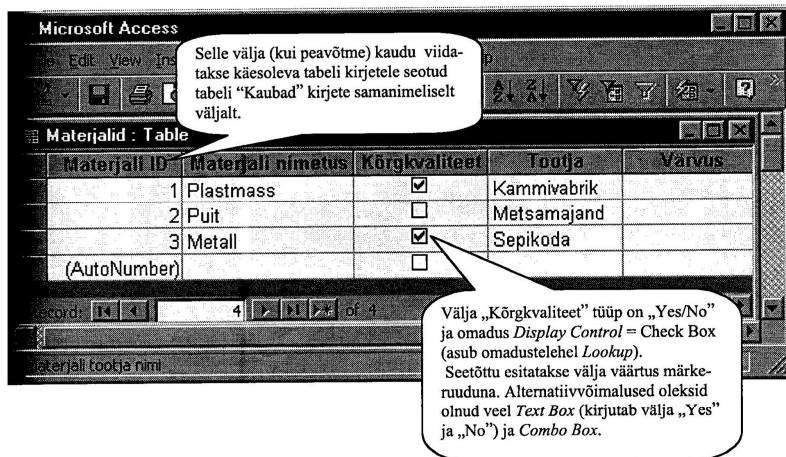
Me saame oma seose luua **ainult siis**, kui primaartabelis „Kliendid“ on vastavad kliendid ka töepooltest olemas.



Edasi peame looma üksmitmese seose tabeli „Kaubad“ peavõtme „Kauba ID“ ja tabeli „Tehingud“ välja „Kauba ID“ vahel. Seda saame me teha ainult siis (kui me muidugi nõuame viitamisterviklikust), kui iga kaup, millele viitab tabel „Tehingud“, on töepooltest olemas tabelis „Kaubad“.



Selleks, et luua viimane seos – üksmitmene seos tabeli „Materjalid“ peavõtme „Materjali ID“ ja tabeli „Kaubad“ välja „Materjali ID“ vahel – peame kõigepealt varustama tabeli „Materjalid“ nende kirjetega, millele soovivad viidata tabeli „Kaubad“ kirjet:



Tulemusena oleme loonud meie andmebaasi tabelite vahel kõik soovitud seosed. Proovime, kas nad ka toimivad.

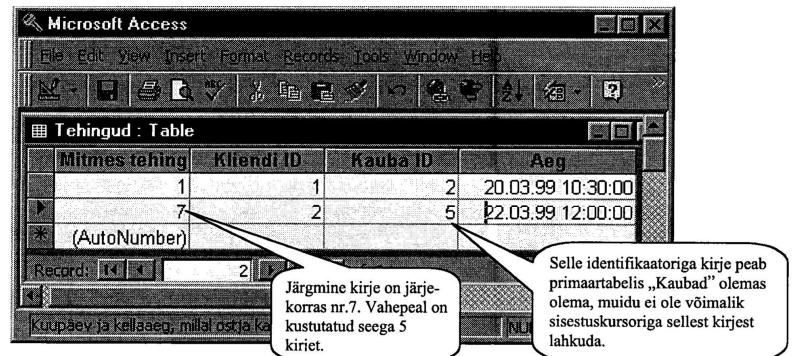
- Omavahel vastavuses olevad kirjad kustuvad üheskoos:** kustutades nt. tabelist „Materjalid“ kirje, mille [Materjali ID] = 3, kustutatakse sellele vastav kirje seotud tabelist „Kaubad“ automaatselt (selle kirje peavõtme väärthus [Kauba ID] = 4), mis omakorda tingib vastava kirje kadumise seotud tabelist „Tehingud“ (mille peavõtme väärthus [Mitmes teing] = 3).

Põhjus:

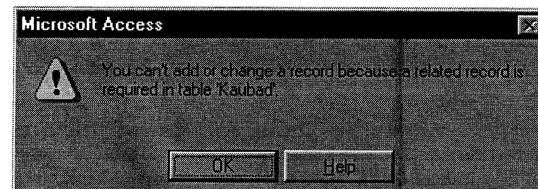
- Ahelas „Materjalid“ → „Kaubad“ → „Tehingud“ on iga tabel temale järgneva tabeli suhtes primaartabeliks (seega kujutab iga järgnev tabel endast eelmise suhtes seotud tabelit): kui mingi kirje kadus tabelist „Materjalid“, polnud vastaval kirjel tabelis „Kaubad“ enam mõtet ning ta kadus samuti jne.
- Tabelitevahelistel seostel on omaduse *Cascade Delete Related Records* väärtsuseks „Jah“ (vastavas märkeruudus on „linnuke“).

Veendume, et vastupidi ei juhi: kirje kustutamine seotud tabelist ei too endaga kaasa kirje kadumist primaartabelist. Kustutame tabelist „Tehingud“ (seotud tabel) kirje, mille peavõtme väärthus [Mitmes teing] = 2 ning millegi viidataksse primaartabeli „Kaubad“ kirjele, mille puhul [Kauba ID] = 3. Kirje kadus küll tabelist „Tehingud“, kuid vastava kirjega tabelis „Kaubad“ ei juhtunud midagi.

- Seotud tabelisse saab lisada uut kirjet ainult siis, kui vastav kirje on primaartabelis olemas:** püüame näiteks lisada uue kirje tabelisse „Tehingud“:



Kuna niisugust kirjet, mille puhul [Kauba ID] = 5, primaartabelis „Kaubad“ ei ole, siis tabelisse „Tehingud“ ut kirjet, mis sellele viitab, meil lisada ei lubata:



Selle kirje saame lisada alles siis, kui oleme eelnevalt loonud vastava kirje primaartabelis:

Microsoft Access

Kaubad : Table

Kauba ID	Kauba nimetus	Seeria	Tootja	Materjali ID
3	Kõögimõõbel	B111	AS Standard	2
5	Metaluks	w222	AS Sepikoda	5
(AutoNumber)				1

On nõutav, et selle identifikaatoriga kirje oleks primaartabelis „Materjalid” olemas.

(tabeli „Tehingud” aken võib seejuures avatuks jäädä, et me hiljem saaksime kirje loomise lõpule viia – kirje andmed kirjutatakse andmebaasi alles siis, kui sisestuskursor on sellelt kirjelt lähkunud, s.t. kui jooksvaks kirjeks on tehtud mõni teine kirje)

Selle kirje salvestamine nõuab omakorda vastava kirje olemasolu primaartabelis „Materjalid” – kirje, mille puhul peavõtme väärthus [Materjali ID] = 5:

Microsoft Access

Materjalid : Table

Materjali ID	Materjali nimetus	Kõrgkvaliteet	Tootja	Värvus
1	Plastmass	<input checked="" type="checkbox"/>	Kammivabrik	
2	Puit	<input type="checkbox"/>	Metsamajand	
5	Messing	<input checked="" type="checkbox"/>	Sulamitehas	
(AutoNumber)				

Kirje, millele viitab seotud tabelisse „Kaubad” lisatav kirje.

Olles salvestanud uue kirje primaartabelis „Materjalid”, saame lõpule viia uue kirje salvestamise ka tabelis „Kaubad”, mis omakorda võimaldab meil salvestada vastav uus kirje tabelisse „Tehingud”.

Aulemusena oleme saanud järgmise struktuuriga andmebaasi (menükäsk **Tools → Relationships**):

