

**IZGLĪTĪBAS UN ZINĀTNES MINISTRIJAS
RĒZEKNES PROFESIONĀLĀ VIDUSSKOLA**



RĒZEKNE, 2007



Mācību materiāli izstrādāti Rēzeknes Profesionālajā vidusskolā, īstenojot ESF projektu ***„Profesionālās vidējās izglītības programmu „Viesnīcu serviss” un „Datorsistēmas” uzlabošana”***, projekta identifikācijas Nr. VPD1/ESF/PIAA/04/APK/3.2.1./0028/0145, laikā no 07.11.2005. līdz 06.12.2007.

Projektu administrēja valsts aģentūra ***„Profesionālās izglītības attīstības aģentūra”*** (līgums Nr. 2005/0266/VPD1/ESF/PIAA/04/APK/3.2.1./0028/0145).

SATURS

(I DAĻA)

IEVADS	4
1. IEVADS DATU BĀZĒS	5
1.1. DATU BĀZES JĒDZIENS.	5
1.2. DATU BĀZU PĀRVALDĪBAS SISTĒMA.	5
2. RELĀCIJAS DATU BĀZU PROJEKTĒŠANAS PAMATPRINCIPI	7
2.1. DBVS PROJEKTĒŠANA.	7
2.2. DB SPECIFIKĀCIJA.	7
3. RELĀCIJAS STARP TABULĀM	8
3.1. RELĀCIJAS (SAITES) JĒDZIENS.	8
3.2. PRIMĀRĀS ATSLĒGAS.	8
3.3. ĀRĒJĀS ATSLĒGAS.	11
3.4. SAITE VIENS-PRET-VIENS (ONE-TO-ONE).	12
3.5. SAITE VIENS-PRET-DAUDZ (ONE-TO-MANY).	13
3.6. SAITE DAUDZ-PRET-DAUDZ (MANY-TO-MANY).	14
3.7. INTEGRITĀTES BŪTĪBA, TĀS NOSACĪJUMI.	15
3.8. INTEGRITĀTES PAMATNOSACĪJUMI.	16
3.9. SPECIFISKIE INTEGRITĀTES NOSACĪJUMI.	17
4. DATU BĀZU NORMALIZĒŠANA	18
4.1. RELĀCIJAS (STARPTABULAS)	18
4.2. PIRMĀ NORMĀLFORMA	18
4.3. OTRĀ NORMĀLFORMA	20
4.4. TREŠĀ NORMĀLFORMA	22
4.5. AUGSTĀKAS PAKĀPES NORMĀLFORMAS.	22
5. DATU BĀZES PROJEKTĒŠANA	23
5.1. ER-MODELIS, REALITĀŠU-SAIŠU DIAGRAMMAS	24
5.2. KĀ IZVEIDOT RE DIAGRAMMU?	26
5.3. DATU PLŪSMU DIAGRAMMAS	28
5.4. DP DIAGRAMMU ELEMENTI UN TO APZĪMĒJUMI	28
5.5. KĀ IZVEIDOT DP DIAGRAMMU?	30
6. DATU BĀZES STRUKTŪRAS PROJEKTĒŠANAS GRAFISKIE LĪDZEKĻI	34
6.1. MS VISIO VIDE	34
6.2. MS VISIO FAILU TIPI.	36
6.3. DARBS AR FORMĀM	36
6.4. FORMU UN SAVIENOTĀJU NOFORMĒŠANA	44
6.5. DARBS AR TEKSTU	46
6.6. DARBS AR SLĀŅIEM (LAYER)	49
6.7. FORMAS STILI UN KRĀSU SHĒMAS	50
6.8. DARBS AR LAPĀM, DRUKĀŠANA.	53
6.9. DIAGRAMMU PUBLICĒŠANA INTERNETĀ, HIPERSAITES. SŪTĪŠANA PA E-PASTU.	54
6.10. JAUNU FORMU UN TRAFARETU IZVEIDE	57
6.11. DARBS AR VEIDNĒM, FONA LAPĀM.	62
6.12. PIESAISTĪŠANAS PUNKTU UN LĪNIJU PIELIETOŠANA. PERSPEKTĪVAS ZĪMĒŠANA.	64
7. LIETOJUMPROGRAMMAS MS ACCESS VIDE	67
8. MS ACCESS DATU BĀZES TABULAS ATRIBŪTI	71
9. DATU BĀZES TABULAS IZVEIDE	72
9.1. TABULAS IZVEIDE, LAUKU ĪPAŠĪBAS	72
10. TABULAS LAUKU DEFINĒŠANA	74
10.1. LAUKU DEFINĒŠANA	74

10.2.	LAUKU SECĪBAS MAIŅA	74
11.	LAUKU ĪPAŠĪBAS <i>FIELD PROPERTIES</i>	76
11.1.	ĪEVADA MASKA	78
11.2.	TABULU SAGLABĀŠANA	80
11.3.	PRIMĀRĀS ATSLĒGAS NOTEIKŠANA	81
11.4.	OBJEKTU NOSAUKUMI	83
12.	DARBS AR TABULAS DATIEM	84
12.1.	DATU IEVADS	84
12.2.	HIPERSAIŠU IZMANTOŠANA TABULĀ	85
13.	RELĀCIJU (SAIŠU) IZVEIDE	91
14.	DATU KĀRTOŠANA, MEKLĒŠANA UN AIZVIETOŠANA	94
14.1.	DATU KĀRTOŠANA	94
14.2.	DATU MEKLĒŠANA UN AIZVIETOŠANA	94
15.	FILTRI	97
16.	VAICĀJUMI	102
16.1.	VAICĀJUMA VEIDOŠANA	103
16.2.	<i>QBE</i> REŽĢIS	104
16.3.	VAICĀJUMA SKATI	106
16.4.	KRITĒRIJU PIELIETOJUMA PIEMĒRI	106
16.5.	APRĒĶINĀTIE LAUKI – <i>CALCULATED FIELD</i>	110
16.6.	VAICĀJUMA SAGLABĀŠANA	111
16.7.	PARAMETRU VAICĀJUMI – <i>PARAMETER QUERIES</i>	111
16.8.	<i>ZOOM LOGS</i>	114
16.9.	SAITES STARP VAICĀJUMA TABULĀM	114
16.10.	VAICĀJUMS AR VIENA VIRZIENA SAITI – <i>OUTER JOIN</i>	114
16.11.	SUMMĒŠANAS VAICĀJUMI – <i>TOTALS QUERIES</i>	116
16.12.	VAICĀJUMA ĪPAŠĪBA <i>TOP VALUES</i>	119
16.13.	ŠĶĒRSTABULAS VAICĀJUMI – <i>CROSSTAB QUERIES</i>	120
16.14.	KOLONNU VIRSRĀKSTU KĀRTOŠANA ŠĶĒRSTABULAS VAICĀJUMIEM	121
16.15.	BIEŽĀK LIETOTĀS FUNKCIJAS UN OPERATORI	122
16.16.	DATU REDIGĒŠANA VAICĀJUMOS	123
16.17.	DARBĪBAS VAICĀJUMI – <i>ACTION QUERIES</i>	123
16.18.	DATU PIEVIENOŠANAS VAICĀJUMS – <i>APPEND QUERY</i>	124
16.19.	DATU MAIŅAS VAICĀJUMS – <i>UPDATE QUERY</i>	125
16.20.	DATU DZĒŠANAS VAICĀJUMS – <i>DELETE QUERY</i>	126
16.21.	TABULAS IZVEIDOŠANAS VAICĀJUMS – <i>MAKE-TABLE QUERY</i>	128

Ievads

Šis materiāls ir izveidots „Datu bāzes” kursa ietvaros. Materiālā ir apskatītas gan teorētiska rakstura problēmas, gan arī praktiskie piemēri, sākot no pašiem pamatiem ar mērķi iemācīties izmantot Microsoft Access programmas iespējas, t.i., lasītājs tiek iepazīstināts ar svarīgākajām darbībām, funkcijām un pamatjēdzieniem. Lasītājam nav vajadzīgas nekādas priekšzināšanas par datu bāzēm, jo viss pakāpeniski tiek izskaidrots no pašiem pamatiem.

Lietotie saīsinājumi un apzīmējumi:

Datu bāze – tā ir sistematizēta informācijas glabātuve. Kā par spilgtu piemēru varētu ņemt telefonu abonementu grāmatu.

Datu bāzes pārvaldības sistēma (Database management system, DBMS) – lietojumprogrammatūra, kas organizē datus datu bāzē, nodrošinot to uzglabāšanu, ieguvī, drošību un integritāti.

Datu bāzes Tabula – ir vieta, kur ievieto dažāda apjoma informāciju.

Specifikācija – ir sistēmas īss apraksts ar tās galvenajām funkcijām un sasniedzamajiem mērķiem.

Relācija – vairāku objektu saistības izpausme, kas attēlo tiem kopīgi piemītošo.

Atslēga – ir ieraksta (*objekta*) īpašība, pēc kuras ieraksts var tikt unikāli identificēts.

Primārā atslēga (*Primary key*) – relāciju datu bāzes pārvaldības sistēmā unikāls lauks, kas identificē datu bāzes tabulas rindiņu. Šo atslēgu dažkārt sauc arī par galveno atslēgu.

Autonumber – ir speciāls Access datu tips, paredzēts tieši primārajām atslēgām. Lauks ar šo datu tipu ir tikai lasāms (tajā neko nevar ievadīt) – MS Access pats rūpējas par laukā esošajām un jaunajām vērtībām.

Indeksēšana – atvieglo darbības ar indeksēto lauku. Īpaši uzlabojas kārtošana, meklēšana un grupēšana, bet samazinās labojumu veikšanas ātrums.

Ārējā atslēga (*Foreign Key*) – ir lauks dotajā tabulā ar kura palīdzību tiek norādīta atsauce uz citas tabulas primāro atslēgu.

Integritāte (*Integrity*) – sistēmu, programmu un datu aizsardzība pret netīšu vai ļaunprātīgu bojāšanu vai pārveidošanu.

ER-modelis, RE diagrammas, relitāšu-saišu diagrammas (*ER model*) – ir grafisks līdzeklis datu bāzes struktūras attēlošanai. Tas sastāv no divu veidu objektiem – **entītēm** (lōģiskais objekts) un **relācijām** (attiecības).

Datu plūsmu diagrammas ir modelēšanas rīks, kas ļauj parādīt sistēmu kā funkcionālu procesa tīklu (Yourdan).

SQL (Structured Query Language – strukturizētu pieprasījumu valoda

1. Ievads datu bāzēs

(Datu bāzes jēdziens, lietojums, piemēri. Tabulas lauka un ieraksta jēdzieni. Populārāko datu bāzu vadības sistēmu pārskats, to priekšrocības, trūkumi, lietojums.)

Pirms uzsākt Datu bāzes izveidi, jānoskaidro, kas ir datu bāze, jāiepazīstas ar galvenajiem jēdzieniem un jāiemācās atpazīt gadījumus, kādās situācijās ir nepieciešamas datu bāzes.

1.1. Datu bāzes jēdziens.

Datu bāze – tā ir sistematizēta informācijas glabātuve. Datu bāzes piemērs ir telefonu abonementu grāmata.

Datu bāze – ir informācijas kopums ar noteiktu struktūru. Jebkurš informācijas kopums var tikt saukts par datu bāzi. Termins tika radīts, lai apzīmētu datora datus un to praktiski lieto tikai datorikā. Dažkārt to lieto, lai apzīmētu vēl nedatorizētus datus, plānojot to datorizēšanu.

Programmatūru, kas radīta, lai pārvaldītu vispārējas nozīmes datu bāzes, sauc par datu bāzu pārvaldības sistēmām jeb DBPS.

Datu bāze (Database, DB) – savstarpēji saistītu informacionālu objektu tematisks kopums, kas ar speciālas pārvaldības sistēmas starpniecību organizēts tā, lai nodrošinātu ērtu informācijas ieguvu, izdarītu tās atlasu un kārtošanu. Informācija datu bāzē parasti ir sadalīta ierakstos (tabulās), no kuriem katram var būt viens vai vairāki lauki (kolonnas).

1.2. Datu bāzu pārvaldības sistēma.

Datu bāze ir īpašā veidā organizēts datu sakārtojums, kuram datus var pievienot, meklēt, izdzēst, labot, utml.

Datu bāzes pārvaldības sistēma (Database management system, DBMS) – lietojumprogrammatūra, kas organizē datus datu bāzē, nodrošinot to uzglabāšanu, ieguvu, drošību un integritāti. Datu bāzes pārvaldības sistēma parasti veic izdrukājamo pārskatu formatizēšanu, kā arī eksportu uz un importu no citām lietojumprogrammām, izmantojot datņu standartformātus.

Par datu bāzēm pieņemts uzskatīt konkrētas datu kolekcijas, piemēram, bibliotēkas kartotēku, "čēkas maisus" u.c. Tās var uzglabāt kā papīra kartotēkas, kā sagrafētu kladīti, kā vienkārša teksta failu, u.c. Lielas datu bāzes visērtāk glabāt un apstrādāt elektroniski ar īpaša veida programmnodrošinājumu (*datu bāzu vadības sistēmām*). Pazīstamākās datu bāzu vadības sistēmas ir "Oracle", "Informix", "Microsoft SQL Server", "MySQL" utml. Par tām var uzskatīt arī Microsoft Office programmu paketes rīkus "Access" un "Excel". Visas modernās DBMS nodrošina datu bāzu lietotājiem iespēju piekļūt (lasīt, rakstīt) datu bāzēm caur vienu un to pašu valodu: SQL.

Datoru programmas ir ieteicams izmantot kā darba instrumentu, kuru katru dienu izmanto darbā. Datu bāzes ir domātas informācijas glabāšanai. Apskatīsim dažus tipiskus informācijas piemērus, kuri visbiežāk glabājas datu bāzēs:

- Informācija par firmas darbiniekiem;
- Produkcijas uzskaite;
- Ziņas par pircējiem;
- Darba plāni;
- Pārdošanas uzskaite;
- Piegādātāji;
- Kompaktdisku kolekcija; u.c.

Jāpievērš uzmanība tam, ka visos dotajos piemēros ir minētas glabāšanas funkcijas un manipulācija ar datiem, nevis aprēķini. Kaut arī datu bāzē ir iespējams veikt daudz finansiālu un statistisku aprēķinu, tomēr šie aprēķini netiek veikti tik ātri kā to var veikt ar elektroniskajām tabulām (MS Excel).

Ja ir nepieciešams apstrādāt, analizēt un uzglabāt kādu liela apjoma informāciju, tad datu bāze labāk noder, lai to savāktu un uzglabātu.

Kopš Microsoft Access 1.0 versijas iznākšanas 1992. gadā pagājis krietns laiks. Ir mainījušās vairākas Microsoft Access versijas, katrai jaunākai nākot klajā ar lielākām priekšrocībām un iespējām, un vienlaikus, sekojot vispārējai neapvaldītajai datortehnikas attīstībai, ar prasībām pēc jaudīgākiem datoriem, ar kuriem strādā, izmantojot Microsoft Access.

Datu bāzes, salīdzinot ar citu programmatūru, ir intuitīvi grūtāk saprotamas un sarežģītāk apgūstamas, bet vienlaikus tās dod iespēju apstrādāt liela apjoma informāciju. Daudz lietotāju pievēršas Microsoft Access, jo viņiem ir nepieciešama ērta un efektīva datu bāzu vadības sistēma, kura darbojas tīklā.

Viena no Microsoft Access¹ priekšrocībām ir tā pilna integrācija ar Microsoft Office programmu paketēm un iespēja strādāt ar vairumu citu datu bāzu vadības sistēmu un datu bāzu serveru datiem.

¹ MS Access – ir datu bāzu vadības sistēma, kas nodrošina:

- Ikdienu vajadzīgu datu ievadu,
- Vienkāršu lietotāja sistēmu izveidošanu, netērējot daudz laika programmēšanai,
- Ekrāna formas un izdrukas sev pieņemamā veidā, iekļaujot visu nepieciešamo informāciju,
- Automātisku saglabāšanu,
- Datu bāzes integritāti, izmainot un likvidējot ierakstus.

2. Relācijas datu bāzu projektēšanas pamatprincipi

(Datu bāzes veidošanas galvenie etapi. Datu bāzes struktūras projektēšana. Relācijas jēdziens. Datu bāzes specifikācija..)

Pazīstamas dažādas datu bāzu sistēmas, kuras organizētas pēc dažādiem principiem. 60-tajos gados situācija šajā nozarē bija ļoti sarežģīta – tika ieviesti hierarhiskie, tīklu, utml. modeļi. Vēsturisku pārskatu par datu bāzu attīstību var izlasīt šajā WDVL rakstā (<http://www.wdvl.com/Authoring/DB/Intro/toc.html>).

2.1. DBVS projektēšana.

Lai uzbūvētu jaunu sistēmu, galvenais uzdevums ir projektēšana un modelēšana (nevis fiziskā sistēmas radīšana).

Projektēšanas jautājumi:

- Kam paredzēta sistēma?
- Kādu informāciju vajadzēs apstrādāt?
- Kā apstrādāt informāciju (lietotājs varēs informāciju labot vai tā būs pieejama tikai aplūkošanai)?

2.1. tabula

Cilvēka un datu bāzes pasaules objekti.

Pasaules tips	Fiziskie objekti	Loģiskie objekti
Cilvēku pasaule	Skapis ar mapēm, papīrs, zīmuli	Domas, idejas
DB pasaule	Datu fails	Tabula, indekss, datu attēlošana

Ko nozīmē DB tabula? To var salīdzināt ar skapi ar izvelkamajām atvilktnēm: atvilktnes var būt dažādas pēc izmēra, katrā no tām var kaut ko ielikt, bet var uz kādu laiku atstāt tukšu (vai arī uz visiem laikiem).



Datu bāzes Tabula – ir vieta, kur ievieto dažāda apjoma informāciju.

2.2. DB specifikācija.

Galveno atbilžu kopumu, kas sniedz atbildes uz globāliem sistēmas jautājumiem, sauc par sistēmas specifikāciju. (Datu bāzes specifikācijas fragmenta piemērs dots 1. pielikumā)

Specifikācija – ir sistēmas īss apraksts ar tās galvenajām funkcijām un sasniedzamajiem mērķiem.

3. Relācijas starp tabulām

(Relācijas (saites) jēdziens, nosacījumi to definēšanā. Relācijas 1-1 (viens pret viens) un 1-∞ (viens pret daudziem), to piemēri. Primārās un ārējās atslēgas būtība, definēšana, prasības attiecībā uz primārā lauka datu tipu un vērtībām. Datu integritāte.)

IBM pārstāvis E. F. Codd ap 1970. gadu ierosināja visu aizstāt ar relācijām.

Relācija – vairāku objektu saistības izpausme, kas attēlo tiem kopīgi piemītošo.

3.1. Relācijas (saites) jēdziens.

Relācijas ir starp dažādu kopu elementiem. Kopas jeb **entītes** (database entities) tiek reprezentētas ar datu bāzu tabulām, kuras sastāv no rakstiem. Katrā rakstā ir vieni un tie paši lauki. Viens raksts ir vienas entītes elements jeb instance, jeb konkrētais iemiesojums. Katram laukam fiksēts tips. Attiecības starp kopu elementiem īsteno īpaša veida lauki, kurus sauc par primārajām atslēgām (**primary keys**) un ārējām atslēgām (**foreign keys**).

Visas datu bāzes lietotājam ļauj ievadīt, rediģēt, skatīt un izdrukāt informāciju, ko satur viena vai vairākas tabulas. Šīs tabulas ne ar ko neatšķiras no parastajām elektroniskajām tabulām (piem. MS Excel tabulas). Lai iepazītu atšķirību starp vienkāršām elektroniskām tabulām un relāciju datu bāzes tabulām ir jāatzīmē trīs būtiskas atšķirības:

- Visas relāciju DB ir paredzētas liela apjoma informācijas apstrādei – nesalīdzināmi lielāks informācijas daudzums, kāds tiek iekļauts vienkāršās elektroniskās tabulās.
- Relāciju DB tabulas var viegli saistīt savā starpā – tabulu uztverot kā vienotu objektu. Turpretī elektroniskās tabulās šo iespēju ir grūti vai gandrīz neiespējami realizēt.
- Relāciju DB datu dublēšanās tiek novesta līdz minimumam. Tiek atvieglota datu rediģēšana, jo tā kā informācija nav dublēta, šīs izmaiņas ir jāveic tikai vienu reizi. Tas tiek realizēts ar saitēm, tādejādi tiek dublētas tikai tabulu atslēgas.

Tās datu apstrādes sistēmas, kas nenodrošina tabulu savstarpējo saistību sauc par “plakano failu procesoriem”. Šīs sistēmas galvenokārt izmanto vienkāršu sarakstu apstrādei. Piemēram: firmas darbinieki un to telefonu numuri.

Pateicoties relāciju datu bāzēm (galvenokārt datu nedublēšanās īpašībai) tiek taupīti datora resursi – operatīvā atmiņa un vieta cietajā diskā datu fiziskajai glabāšanai.

3.2. Primārās atslēgas.

Katra tabula relāciju modelī reprezentē kādu reālās pasaules objektu, tikai vienu objektu (*skolēns, darbinieks, grāmata*). Šis objekts jeb kopa (*entīte*) var

būt reāls objekts vai notikums (*atzīmes saņemšana, pasūtījums, telefona zvans u.c.*).

Relāciju DB modelis pieprasa, lai katrs tabulas ieraksts (*objekts*) būtu unikāls. Ja tiek pieļauti divi identiski ieraksti, tad tiek apgrūtināta un izslēgta objekta unikāla identificēšana izstrādes un programmēšanas laikā. Tas var radīt vēl virkni dažādu problēmu sistēmas darbības laikā, tāpēc lai novērstu ierakstu dublēšanos, lieto **atslēgas**.

Atslēga – ir ieraksta (*objekta*) īpašība, pēc kuras ieraksts var tikt unikāli identificēts. Atslēgai nevar būt tukša vērtība (*nultā vērtība, null*). Atslēga var būt viens lauks vai arī vairāku lauku kombinācija. Nav arī svarīgs lauka datu tips – tas var būt gan cipars, gan teksts.

Atslēga var būt arī vairāki lauki, kas kopā nodrošina ieraksta identifikāciju. Piemēram, *valūtu kursu tabulā vajadzētu būt laukiem Datums, Valūta un Kurss. Lauku Datums un Valūta vērtības, protams, var atkārtoties, taču šo lauku kombinācijai tabulā ir jābūt vienai.*

Lai izprastu unikālo atslēgu pielietojumu, aplūkosim piemēru – tabulu *tblPartneris*, kurā esam nolēmuši glabāt datus (*vārds, uzvārds, adrese, rajons, pasta indekss un telefons*) par partneriem.



Vards	Uzvards	Adrese	Rajons	Indekss	Telefons
Andis	Roze	Gaujas 15	Rīga	LV-2150	9123456
Ainārs	Mitris		Rīga	LV-1001	7234678
Jānis	Bērziņš	Sigulda, Jaunā 27	Rīgas rajons		7560978
Jānis	Bērziņš	Matīsa 14-9	Rīga		7128689
Kristīne	Rīna	"Jānīši"	Bauskas rajons	LV-2300	6125689

3.1.att. Tabulas *tblPartneris* fragments.

Lai unikāli identificētu partneri, vajadzētu izvēlēties unikālu atslēgu. Šim mērķim varētu kalpot lauku *vārds* un *uzvārds* kombinācija. Redzot nelielu datu piemēru (divi partneri ar identisku vārdu un uzvārdu – *Jānis Bērziņš*), ir jāatsakās no šīs atslēgas, jo tā neveic tai uzdotās unikalitātes prasības. Varētu atslēgu papildināt ar vēl kādu lauku – *adrese, telefons*, bet arī šī saliktā atslēga nevarēs nodrošināt ieraksta unikalitāti, jo satur tukšas vērtības. Lai nodrošinātu ieraksta unikalitāti, tabulai pievieno lauku, kas nodrošinās šo iespēju. Mūsu gadījumā partneris ir persona, tāpēc viena iespēja būtu pievienot lauku – *personas kods*. Arī šī iespēja var radīt problēmas – var nezināt partnera personas kodu, vai tas var arī būt nepieejams konfidencialitātes dēļ, vai arī tas var būt nepilnīgs. Ieviesīsim katram partnerim savu identifikatoru – numuru, pēc kura varēsīm unikāli identificēt personu.

tblPartneris : Table						
Partneris ID	Vards	Uzvards	Adrese	Rajons	Indekss	Telefons
1	Andis	Roze	Gaujas 15	Rīga	LV-2150	9123456
2	Ainārs	Mitris		Rīga	LV-1001	7234678
3	Jānis	Bērziņš	Sigulda, Jaunā 27	Rīgas rajons		7560978
4	Jānis	Bērziņš	Matīsa 14-9	Rīga		7128689
5	Kristīne	Rīna	"Jānīši"	Bauskas rajons	LV-2300	6125689
* (AutoNumber)						

3.2.att. Jauns lauks *Partneris_ID* tabulā *tblPartneris* .

Lai nodrošinātu relāciju DB svarīgāko īpašību – relācijas (tabulu savstarpējo saistību) jālieto *primārās atslēgas*. Katrai tabulai var būt tikai viena primārā atslēga un vairākas unikālās atslēgas. Par primāro atslēgu var kalpot jebkura atslēga, kas nodrošina ieraksta unikalitāti. Parasti katrai tabulai ievieš speciālu primārās atslēgas lauku (nosaucot to par *identifikatoru*, *ID* vai biežāk lieto kombināciju *<tabulas nosaukums>_ID*). MS Access programmā šim laukam kā datu tipu izmanto *Autonumber* datu tipu.

Primārā atslēga (*Primary key*) – relāciju datu bāzes pārvaldības sistēmā unikāls lauks, kas identificē datu bāzes tabulas rindiņu. Šo atslēgu dažkārt sauc arī par galveno atslēgu.

Arī mūsu piemērā laukam *Partneris_ID* mēs izmantojām šo datu tipu. Šis lauks mums kalpo gan kā unikāla atslēga, gan kā primārā atslēga.

Autonumber – ir speciāls Access datu tips, paredzēts tieši primārajām atslēgām. Lauks ar šo datu tipu ir tikai lasāms (tajā neko nevar ievadīt) – MS Access pats rūpējas par laukā esošajām un jaunajām vērtībām.

Piezīme. *Par datu tipiem un to īpašībām tiek minēts 8. nodaļā „MS Access datu bāzes tabulas atribūti”.*

Autonumber ir skaitļu tipa lauks, kur katra jaunā vērtība saglabā unikalitāti ar iepriekšējām vērtībām. Ir pieejamas arī papildus īpašības šī tipa laukiem:

General	Lookup
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)

3.3.att. Īpašības *Autonumber* datu tipa laukam .

viena no tām ir *New values* (jauno vērtību pievienošanas veids) ar iespējamām vērtībām – *Increment* (jaunās vērtības tiek liktas pēc kārtas – katra jaunā par

viens lielāka nekā iepriekšējā) un *Random* (unikālas vērtības sajauktā kārtībā). Otra svarīga īpašība ir *Indexed* (indeksēšana) ar vērtībām – *No*, *Yes* (*Duplicates Ok*), *Yes* (*No Duplicates*).

Atstājot noklusēto vērtību – *Yes* (*No Duplicates*), kas tieši nodrošinās vērtību unikalitāti – tās neatkārtosies. Bet galvenā šīs īpašības pamatnozīme ir *indeksēšana*.

Indeksēšana – atvieglo darbības ar indeksēto lauku. Īpaši uzlabojas kārtošana, meklēšana un grupēšana, bet samazinās labojumu veikšanas ātrums.

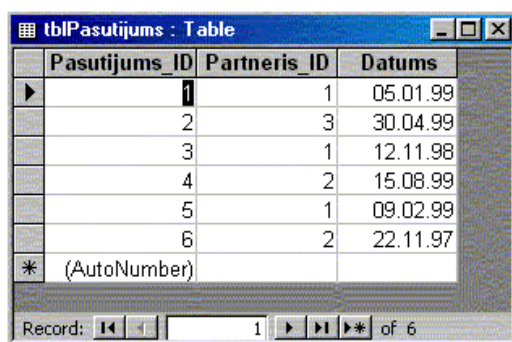
Jāievēro arī to, ka nedrīkst pārcensties ar lauku indeksēšanu – to vajadzētu pielietot tikai laukiem, ar kuriem notiek visintensīvākās darbības (meklēšana, kārtošana). Primāro atslēgu indeksēšana ir obligāta, MS Access programmā tas notiek automātiski. Indeksēšana notiek pēc atslēgām, tāpēc tā var tikt pielietota gan vienam laukam, gan lauku kombinācijai (saliktai atslēgai, t.i. atslēgai no vairākiem laukiem). Viena lauka indeksēšanu veic, uzstādot papildus īpašību *Indexed*, bet salikto atslēgu indeksēšanu, kā arī pašu atslēgu definēšanu izmanto indeksēšanas logu (*Indexes*). MS Access programmā – tas ir izvēlne *View/Indexes* (kad tabula atrodas konstruktora režīmā). Par indeksu izveidi ir aprakstīts 9. nodaļā „Datu bāzes tabulas izveide”.

3.3. Ārējās atslēgas.

Primārās atslēgas izmanto katrai tabulai individuāli, bet, ja DB tabulas netiek savstarpēji saistītas, tad no primārajām atslēgām nav reāla pielietojuma un jēgas, jo to galvenā funkcija ir starptabulu saišu definēšanas nodrošinājums. Primāro atslēgu patiesā būtība sāk izpausties tikai tad, kad tiek definētas tabulu savstarpējās attiecības – relācijas. Definējot relācijas, parādās vēl viens atslēgu veids – *ārējās atslēgas*.

Ārējā atslēga (*Foreign Key*) – ir lauks dotajā tabulā ar kura palīdzību tiek norādīta atsauce uz citas tabulas primāro atslēgu.

Tabulā *tblPasutijums* ir lauki – *Pasutijums_ID*, *Partneris_ID*, *Datums*:



Pasutijums_ID	Partneris_ID	Datums
1	1	05.01.99
2	3	30.04.99
3	1	12.11.98
4	2	15.08.99
5	1	09.02.99
6	2	22.11.97

3.4. att. Ārējās atslēgas lauki tabulā .

Lauks *Partneris_ID* ir ārēja atslēga, kas norāda uz tabulas *tblPartneris* primāro atslēgu *Partneris_ID*. Rezultātā – katram pasūtījumam unikāli piesaista atbilstošo partneri no tabulas *tblPartneris*.

Lai izveidotu saiti starp primāro un ārējo atslēgu, ir jāsakrīt to datu tipiem. Kas attiecas uz *Autonumber* (primārajai atslēgai) datu tipu – tam kā atbilstošo ārējo atslēgas tipu lieto *Number* datu tipu (ar formātu *Long Integer*).

Projektējot DB, ir jāveido tabulas un to primārās atslēgas atbilstoši DB specifikai. Tabulu relācijas definē atbilstoši reālās dzīves aspektiem. Šo aspektu attēlošana – relāciju definēšana var būt diezgan komplicēts process, jo MS Access pieļauj saiti tikai tabulu pārim. Access pieļauj trīs saišu veidus: **viens-pret-viens** (*one-to-one*), **viens-pret-daudz** (*one-to-many*) vai **daudz-pret-daudz** (*many-to-many*). Bet fiziski norādīt var tikai divus: **viens-pret-viens** (*one-to-one*) un **viens-pret-daudz** (*one-to-many*).

3.4. Saite viens-pret-viens (one-to-one).

Divas tabulas ar šo saiti – *viens-pret-viens* (1-1) tiek saistītas gadījumos, kad katram pirmās tabulas ierakstam otrajā tabulā atbilst ne vairāk kā viens ieraksts.

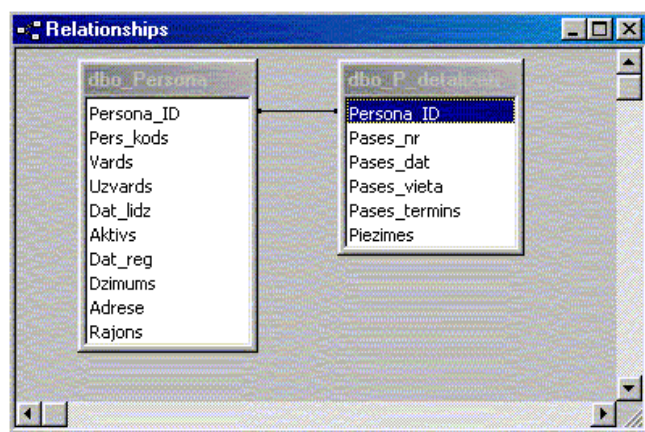
Saite ***viens-pret-viens (one-to-one)*** – katram ierakstam tabulā **A** var būt tikai viens saistīts ieraksts tabulā **B** un otrādi.

Šī veida saites izmanto reti, jo lielākā ierakstu daļa, kas saistīta pēc šī principa, var būt apvienota vienā tabulā.

Saites veidu viens-pret-viens izmanto situācijās, lai:

- sadalītu tabulas, kurām ir liels lauku skaits,
- atdalītu daļu no tabulas drošības pēc,
- saglabātu ierakstus, kas attiecas uz galvenās tabulas ierakstu kopumu.

Īsta šāda veida attiecība (viens-pret-viens) reālajā pasaulē ir reti sastopama. Kā jau bija atzīmēts, DB šo saiti visbiežāk izmanto, lai sadalītu vienu tabulu vairākās daļās. Viens no iemesliem, kāpēc to izmanto MS Access programmā, ir tabulas fiziskie parametri. MS Access pieļauj tabulas ar maksimālo kolonnu skaitu 255. Gadījumos, kad objekta īpašību definēšanai ir nepieciešamas vairāk kā 255 kolonas, tad arī lieto šo tabulas mākslīgo sadalīšanu. Ir arī citi gadījumi, kad pielieto šo saiti – tabulu sadala datu drošības vai konfidencialitātes dēļ. Vienā daļā (tabulā) glabā pamatdatus – šo tabulu izmanto visām DB pamatdarbībām un operācijām, otrā (vai vairākās citās) pārējo objekta informāciju – šie dati netiek tik bieži izmantoti vai arī saistīti ar drošību un konfidencialitāti.



3.5. att. Saite viens-pret-viens piemērs.

Tabulām, kuras tiek saistītas ar šī veida saiti vienmēr ir jābūt identiskām primārajām atslēgām (šīs saites gadījumā nav ārējās atslēgas).

3.5. Saite viens-pret-daudz (one-to-many).

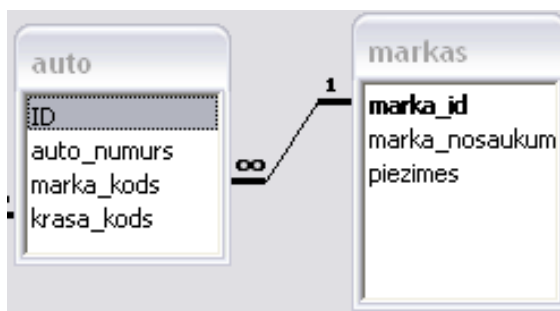
Divas tabulas ar šo saiti – *viens-pret-daudz* ($1-\infty$) tiek saistītas gadījumos, kad katram pirmās tabulas laukam otrajā tabulā atbilst viens vai daudz ieraksti (var būt arī nulle ierakstu), bet katram otrās tabulas ierakstam pirmajā tabulā atbilst tieši viens ieraksts.

Saite ***viens-pret-daudz*** (***one-to-many***) – katram ierakstam tabulā **A** atbilst vairāki ieraksti tabulā **B**, bet ierakstam tabulā **B** nevar eksistēt vairāk par vienu ierakstu tabulā **A**.

Šis ir viens no biežāk izmantojamiem saišu veidiem starp tabulām. Plaši izmanto modelī identifikācijas numuri un to apraksts

One-to-Many piemēri:

Piemērs 1. Galvenā tabulā „Automašīnas” glabājas automašīnu krāsu *ID* numuri. Savukārt tabulā „Krāsas” glabājas *ID* numuri un krāsu nosaukumi



3.6. att. Saite viens-pret-daudz piemērs.

Piemērs 2. Viens piegādātājs var piegādāt vairākās preces, savukārt katrai precei ir tikai viens piegādātājs.

3.6. Saite daudz-pret-daudz (many-to-many).

Divas tabulas tiek saistītas ar šo saiti – *daudz-pret-daudz* gadījumos, kad pirmās tabulas ierakstam otrajā atbilst daudz ierakstu un arī katram otrās tabulas ierakstam pirmajā atbilst daudz ierakstu. Šī saite DBVS, arī MS Access programmā nevar tikt tieši realizēta. **Lai realizētu šo saiti, tā tiek sadalīta divās viens-pret-daudz saitēs. Sadalīšanas procesā rodas vēl viena tabula, kas nodrošina saistību starp divām jaunajām saitēm.**

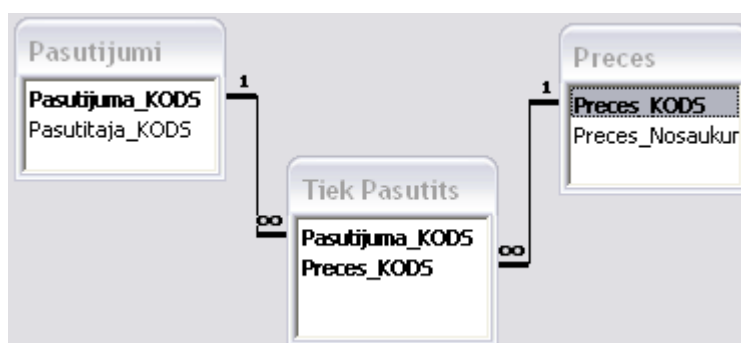
Saite *daudz-pret-daudz* (*many-to-many*) – katram ierakstam tabulā **A** atbilst vairāki ieraksti tabulā **B**, savukārt katram ierakstam tabulā **B** atbilst vairāki ieraksti tabulā **A**.

Šāda veida saites starp divām tabulām ir iespējams izveidot, izmantojot trešo (sasaistošu), palīgtabulu, kuras primārā atslēga sastāv no diviem laukiem (kolonnām), kas savukārt ir tabulu A un B ārējās atslēgas lauki.

Saiti “daudzi-pret-daudziem” var attēlot kā divas saites “viens-pret-daudziem” ar trešo tabulu

Many-to-Many piemēri:

Saite “*daudz-pret-daudz*” ir starp tabulām “*Pasutijumi*” un “*Preces*”. Tā tiek nodefinēta, izmantojot divas saites *viens-pret-daudz* ar tabulu “*Tiek Pasutits*”. Vienā pasūtījumā var būt vairākas preces, bet katra no precēm var atrasties dažādos pasūtījumos:

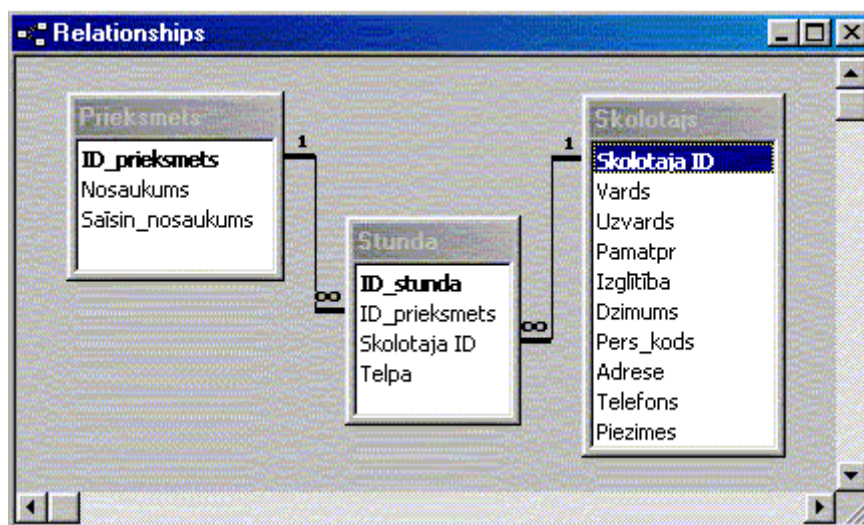


3.7. att. Saites *daudz-pret-daudz* 1. piemērs.

Aplūkosim vēl vienu piemēru par skolotājiem un mācību priekšmetiem, ko pasniedz šie skolotāji. Katrs skolotājs var pasniegt vairākus mācību priekšmetus un arī katru mācību priekšmetu var pasniegt vairāki skolotāji.

Kāpēc nevar realizēt šo saiti? Aplūkosim to no „skolotāja” viedokļa: Lai varētu glabāt informāciju par skolotāja mācību priekšmetiem, tabulai vajadzētu

pievienot tik laukus, cik katrs skolotājs māca priekšmetus (mēs nevaram viennozīmīgi noteikt šo lauku skaitu, jo tas ar laiku var mainīties). Tāda pati situācija ir arī aplūkojot šo problēmu no „mācību priekšmeta” viedokļa – priekšmetam jāpievieno tik lauku (priekšmetiem), cik skolotāji var to mācīt.



3.8. att. Saites daudz-pret-daudziem 2. piemērs.

Izveidojot tabulu *Stunda*, kas mums kalpos kā starptabula, jeb sasaistoša tabula, var realizēt šo *n-n* saiti ar divu *1-n* saišu palīdzību.

Saite 1-1 tiek lietota reti, saite *n-n* neeksistē, bet tā tiek noreducēta uz divām saitēm 1-n. Rezultātā mēs varam secināt, ka DB visbiežāk ir sastopama saite 1-n.

3.7. Integritātes būtība, tās nosacījumi.

Relācijām DB ir daži specifiski integritātes noteikumi, kurus nevar definēt ar normālformu palīdzību.

Integritāte (*Integrity*) – sistēmu, programmu un datu aizsardzība pret netīšu vai ļaunprātīgu bojāšanu vai pārveidošanu.

Piezīme. Par tabulu normālformām un normalizēšanu tiek minēts 4.nodaļā „Datu bāzu normalizēšana”

Integritātes nosacījumi iedalās divās grupās:

1. Pamatnosacījumi.
2. DB specifiskie integritātes nosacījumi.

3.8. Integritātes pamatnosacījumi.

Relāciju modelī ir divi specifiski integritātes pamatnosacījumi. Tos klasificē kā pamatnosacījumus, jo tie attiecas uz visu DB kopumā un ir obligāti pielietojami visām relāciju DB. Tie ir *entīšu integritāte* un *referenciālā integritāte*.

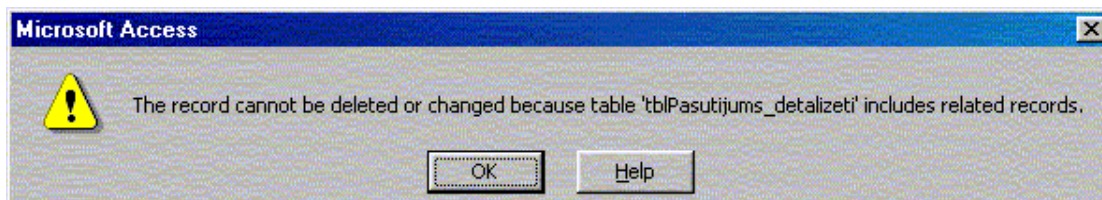
Entīšu integritātes nosacījums ir ļoti vienkāršs – tas nosaka, ka primārā atslēga nevar saturēt tukšas (null) vērtības. Iemesls šim nosacījums ir triviāls, jo tukšas primārās atslēgas gadījumā mēs nevaram unikāli identificēt ierakstu un uz to atsaukties citā tabulā vai relācijā. Šis pats nosacījums attiecas arī uz saliktajām atslēgām – neviena no komponentēm nedrīkst saturēt tukšu vērtību.

Referenciālās integritātes nosacījums aizliedz kā ārējo atslēgu vērtības lietot primārās atslēgas neeksistējošas vērtības. Tas nozīmē, ka:

- jaunu (neesošu) ārējās atslēgas vērtību var pievienot tikai tad, ja iepriekš ir izveidota atbilstošā primārā atslēga,
- primārās atslēgas rediģēšanas vai arī ieraksta dzēšanas gadījumā ir jāveic atbilstošas darbības ar atbilstošajām ārējo atslēgu vērtībām.

Šim gadījumam ar datu rediģēšanu (dzēšanu) ir paredzētas speciālas iespējas. Izšķir trīs gadījumus, kā notiek šis process:

Aizliegt (*Disallow*) – Izmaiņas tiek aizliegtas. Access parāda atbilstošu ziņojumu.

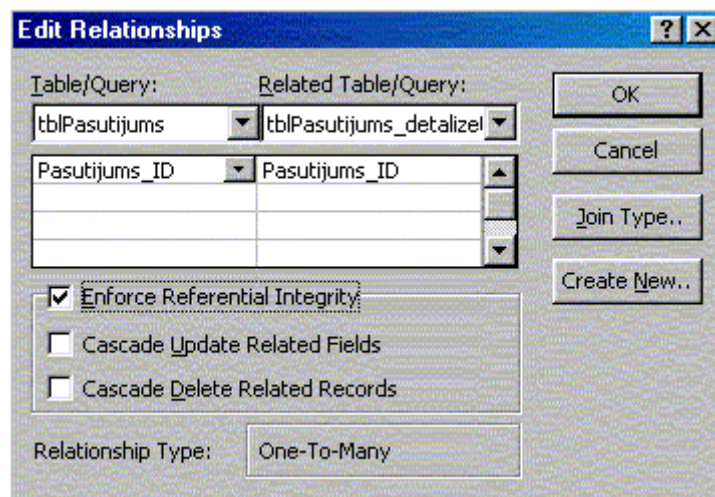


3.8. att. MS Acces paziņojums pie *Disallow*.

Kaskāde (*Cascade*) – izmaiņu gadījumā ārējās atslēgas mainās atbilstoši primārās atslēgas izmaiņām. Dzēšanas gadījumā līdzīgi – automātiski tiek izdzēstas atbilstošās ārējās atslēgas (viss ieraksts).

Atcelšana (*Nullify*) – dzēšanas (primārās atslēgas) gadījumā atbilstošās primārās atslēgas tiek aizstātas ar tukšām (nultām) vērtībām. MS Access nepiedāvā iespēju *Nullify*.

Šīs referenciālās integritātes nosacījumus definē *Edit Relationships* (Tools/Relationship) logā:



3.8. att. *Edit Relationships* dialoglogs.

3.9. Specifiskie integritātes nosacījumi.

Specifiskie integritātes nosacījumi ir jāievēro un jāizstrādā atbilstoši DB specifikai. Katrai DB tie var būtiski atšķiries atbilstoši DB specifikācijai un pielietojumam. Piemēram, DB par skolu:

- Katrā klasē skolēnu skaits nedrīkst pārsniegt, piemēram, 30 skolēnus.
- Skolēniem klasē jābūt samērīgās proporcijās attiecībā pret dzimumu.
- Skolēnu iedalījumam pa klasēm jābūt atbilstoši vecumam (vadoties no personas koda). utt.

4. Datu bāzu normalizēšana

Kā jau iepriekš redzējām, izstrādājot datu bāzi saskaramies ar daudz dažādām DB izstrādes iespējām: cik un kādas DB būs tabulas, kāda būs to struktūra, kāds būs datu kolonu izvietojums pa tabulām, kāda būs tabulu savstarpējā saistība. Atbildes uz visiem šiem jautājumiem sauc par normalizēšanu. Normalizēšana ir DB vienkāršošana un optimizēšana atbilstoši relāciju DB prasībām.

Normalizēšanas teorija piedāvā normālformu koncepciju, kas ļauj optimizēt DB struktūru. Normālformas ir nosacījumu kopas, kam būtu jāapmierina DB. Normalizēšanā tiek izdalītas šādas normālformas:

- Pirmā normālforma
- Otrā normālforma
- Trešā normālforma
- Boyce Codd normālforma
- Ceturtā normālforma
- Piektā normālforma

Pielietojot projektējamai DB arvien augstākas pakāpes normālformu, iegūstam arvien optimālāku un relāciju DB teorijai atbilstošāku datu bāzi.

4.1. Relācijas (starptabulas)

Normālformu pielietojums vairāk tiek lietots saistībā ar relācijām nekā ar tabulām. Šinī gadījumā relācija nav pati saite, bet gan tabula, ko mēs aplūkojam saites n-n gadījumā. Relācija ir specifisks tabulas veids kam ir šādi nosacījumi:

- Tā apraksta vienu entīti,
- Nav atkārtoto rindīņu (ierakstu),
- Kolonas nav sakārtotas,
- Rindas nav sakārtotas.

4.2. Pirmā normālforma

Pirmā normālforma (1NF) pieprasa, lai kolonu vērtības būtu atomāras. Atomāras – nozīmē nedalāmas, katra kolona satur mazāko iespējamo vērtības kopumu, kas sīkāk nav sadalāms. Piemēram, adresi pareizi būtu jāglabā sadalītā veidā (lauki – rajons, pilsēta, iela, u.c.), nevis visu vienā teksta laukā.

1NF pieprasa to, lai katra pozīcija (rinda/kolona) saturētu atomāru vērtību un nevis vairāku vērtību uzskaitījumu. Ar šādām atomārām vērtībām ir daudz vieglāk manipulēt un veikt visas nepieciešamās darbības. Darbības ar neatomāriem datiem ir komplicētākas – tās prasa vairāk resursu.

Pasutijums_ID	Partneris_ID	Preces
1	1	5 datori, 3 printeri, 6 galda lampas
2	3	dators
3	1	2 datoru galdi, 2 galda lampas
4	2	15 krēsli
5	1	dators
6	2	5 printeri, dators, 2 galda lampas
(AutoNumber)		

Record: 7 of 7

4.1. att. Tabulas lauks *Preces*.

Aplūkojam lauku *preces* – šeit skaidri redzama projektēšanas kļūda. Dati nav atomāri, bet ir preču uzskaitījums ar vairākām atomārām vienībām. Iedomāties uzdevuma sarežģītību, ja tiek vaicāta atskaite par preču kustību (jāsummē preces pēc nosaukumiem).

1NF aizliedz arī atkārtotojošās grupas, pat tad, ja tās tiek glabātas dažādās saliktās kolonās. Aplūkojam šo pašu tabulu tikai modificētu, arī tā neatbilst 1NF prasībām, jo nav ievērotas atkārtotojošās grupas.

Pasutijums_ID	Partneris_ID	Daudzums1	Prece1	Daudzums2	Prece2	Daudzums3	Prece3
1	1	5	dators	3	printeris	6	lampa
2	3	1	lampa	1	dators		
3	1	2	datora galds	2	lampa		
4	2	15	krēsls				
5	1	1	dators				
6	2	5	printeris	2	lampas		
(AutoNumber)							

Record: 7 of 7

4.2. att. Tabulas *tblPasutijums* jauna struktūra.

Kāpēc nevar lietot šādu struktūru (sk. att. 4.2.)? Veidojot vaicājumus un atskaites, viennozīmīgi nevarēsim veikt operācijas ar kādu konkrētu preci, jo tās uzskaitījums katrā ierakstā var parādīties citā laukā. Šo problēmu varētu atrisināt – katrai precei rezervēt savu uzskaites kolonu, bet tas rada preču skaita ierobežojumu (255 lauki tabulā). Šāds glabāšanas veids ir arī ļoti neoptimāls no resursu viedokļa – pasūtījums var sastāvēt no vienas preces, bet mēs rezervējam vietu visām iespējamajām preču vienībām. Problēmas arī jaunas preču vienības gadījumā.

Lai izvairītos no grupām, modificējam šo tabulu (sk. att. 4.3.).

	Pasutijums_ID	NPK	Partneris_ID	Daudzums	Prece
	1	1	1	5	dators
	1	2	1	3	printeris
	1	3	1	6	lampa
	2	1	3	1	lampa
▶	2	1	3	1	dators
	3	1	1	2	datora galds
	3	2	1	2	lampa
	4	1	2	15	krēsls
	5	1	1	1	dators
	6	1	2	5	printeris
	6	2	2	2	lampa
*					

Record: 5 of 11

4.3. att. Tabulas *tblPasutijums* 1.pārveidojums.

Vienam pasūtījumam vairs neatbilst viens ieraksts, bet gan vairāki ieraksti. Tagad katrs modificētās tabulas ieraksts reprezentē kāda pasūtījuma vienu preci. Preces piederību pasūtījumam mēs nosakām pēc lauka *Pasutijums_ID*. Primāro atslēgu vairs neveido lauks *Pasutijums*, bet gan lauku *Pasutijums* un *NPK* kombinācija.

Tagad tabula *tblPasutijums* atbilst visām 1NF prasībām – dati ir atomāri un tie neveido grupas, kas atkārtojas.

Šāda tabulas struktūra viegli ļauj operēt ar konkrētu preču vienību – varam atrast, grupēt un summēt vajadzīgos datus.

4.3. Otrā normālforma

Tabula ir otrajā normālformā 2NF, ja tā ir 1NF un katrs neatslēgas lauks ir norāde uz kādu primāro atslēgu.

	Pasutijums_ID	NPK	Partneris_ID	Datums	Daudzums	Prece_ID	Prece
	1	1	1	05.01.99	5	2	dators
	1	2	1	05.01.99	3	5	printeris
	1	3	1	05.01.99	6	4	lampa
	2	1	3	30.04.99	1	4	lampa
	2	2	3	30.04.99	1	2	dators
	3	1	1	12.11.98	2	1	datora galds
	3	2	1	12.11.98	2	4	lampa
	4	1	2	12.11.98	15	3	krēsls
	5	1	1	09.02.99	1	2	dators
	6	1	2	22.11.97	5	5	printeris
▶	6	2	2	22.11.97	2	4	lampa
*							

Record: 11 of 11

4.4. att. Tabulas *tblPasutijums* 2.pārveidojums.

Tabula ir modificēta iepriekšējā varianta tabulā, tās kolonas joprojām ir atomāras, un nav grupu, kas atkārtojas.

Lai noskaidrotu vai šīs tabulas variants atbilst 2NF, jānoskaidro primārā atslēga. Primārā atslēga ir salikta atslēga, kas sastāv no laukiem – *Pasutijums_ID* un *Pasutijuma_NPK*. Tabulā bez atslēgām ir divu veidu īpašības – tādas, kas attiecas uz pasūtījumu un tādas, kas attiecas uz precēm.

Lai Tabula atbilstu 2NF, to sadala divās daļās. Šādu procesu – nenormalizētas tabulas sadalīšanu divās (vai n) normalizētās tabulās sauc par dekompozīciju.

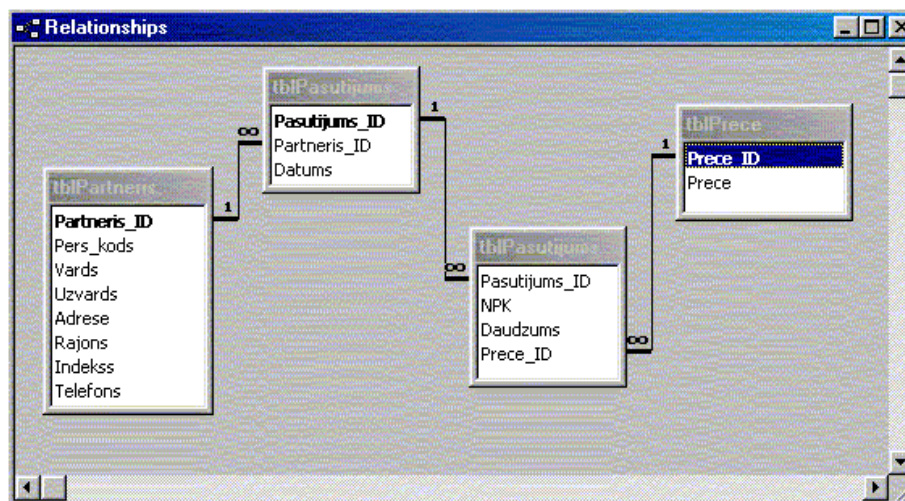
The image shows two screenshots of database tables. The top screenshot is titled 'tblPasutijums : Table' and contains 6 records with columns Pasutijums_ID, Partneris_ID, and Datums. The bottom screenshot is titled 'tblPasutijums_detalizeti : Table' and contains 11 records with columns Pasutijums_ID, NPK, Daudzums, Prece_ID, and Prece.

Pasutijums_ID	Partneris_ID	Datums
1	1	05.01.99
2	3	30.04.99
3	1	12.11.98
4	2	12.11.98
5	1	09.02.99
6	2	22.11.97

Pasutijums_ID	NPK	Daudzums	Prece_ID	Prece
3	1	2	1	datora galds
1	1	5	2	dators
5	1	1	2	dators
2	2	1	2	dators
4	1	15	3	krēsls
2	1	1	4	lampa
1	3	6	4	lampa
3	2	2	4	lampa
6	2	2	4	lampa
6	1	5	5	printeris
1	2	3	5	printeris

4.5. att. Tabulas *tblPasutijums* 3.pārveidojums.

Tabula *tblPasutijums* atbilst 2NF, bet tabula *Pasutijums_detalizeti* neatbilst 2NF, lauki *Prece_ID* un *Preces_nosaukums* neatbilst nosacījumiem par primārajām atslēgām. Lauks – *Preces_nosaukums* neatbilst 2NF, jo tā ir preces īpašība, bet tabula ir domāta pasūtījuma īpašībām. Veicam vēl vienu dekompozīcijas procesu – sadalām tabulu *tblPasutijums_detalizēti* nošķirot atsevišķi tabulu *tblPrece*.



4.6. att. DB relāciju shēma.

Rezultātā DB atbilst shēmai, kuru var apskatīt 4.6. attēlā.

4.4. Trešā normālforma

Tabula ir trešajā normālformā 3NF, ja tā ir 2NF un visi neatslēgu lauki ir savstarpēji neatkarīgi (tabula nesatur laukus, kas ir iegūstami no pārējo lauku kombinācijām un matemātiskām operācijām). Piemēram, neglabājam summu, ja ir zināmi visi summas locekļi. Mūsu pasūtījumu piemērā: ja ir zināma katras pasūtījuma preces cena, tad neglabājam visu pasūtījuma kopīgo summu.

4.5. Augstākas pakāpes normālformas.

Pēc 3NF visas normalizēšanas problēmas saistās tikai ar tabulām, kurām ir trīs un vairākas kolonas un tās visas ir atslēgas.

DB speciālisti apgalvo, ka, ja entīte ir 3NF, tad tā bieži ir arī 4NF un 5NF. Pārveidojot tabulu no 3NF uz augstākas pakāpes normālformām (4NF un 5NF), iegūtais normalizēšanas rezultāts ir salīdzinoši niecīgs, salīdzinot ar procesā ieguldīto darbu. Tomēr, neskatoties uz to, specifiskos gadījumos tiek pielietotas augstākās pakāpes normālformas.

5. Datu bāzes projektēšana

Lai uzbūvētu jaunu sistēmu galvenais uzdevums ir nevis pati fiziskā sistēmas radīšana, bet gan šīs sistēmas projektēšana un modelēšana. Sistēmu varētu salīdzināt ar ēku, ko vēlamies uzbūvēt, vispirms mēs veicam projektēšanu - izdomājam, iztēlojamies un apspriežam savu topošo ēku, paredzam tai visas nepieciešamās detaļas, izanalizējam visas funkcijas, kādām kalpos katra konkrētā mājas detaļa - viss tiek iepriekš labi pārdomāts un analizēts. Un tikai tad, kad ir izstrādāts ēkas projekts (kaut kādā veidā, piemēram, grafiski iemūžināts) mēs varam pilnvērtīgi ķerties pie šīs savas ieceres – uzbūvēt ēku.

Protams, var arī neaizrauties ar projektēšanu. Bet, kā rāda pieredze, pēc tam var būt rūgta vilšanās tik neapdomīgam "celtniekam". Var sistēmu būvēt bez projekta - "lipināt" no maziem fragmentiem, tādejādi iegūt vēlamo. Bet, kā jau minēts, tas var radīt zināmas nepatīkšanas un šādi "audzējot" sistēmu var rasties situācija, ka vairāk nav iespējama šāda "pielipināšana".

Kas tad īsti ir projektēšana? Projektēšanu varētu nosaukt par "atbildēšanu" uz mums interesējošiem jautājumiem, piemēram:

1. Kam paredzēta sistēma?
2. Kādu informāciju mēs gribēsim apstrādāt?
3. Kā apstrādāsim informāciju (lietotājs varēs informāciju labot, vai tā būs pieejama tikai aplūkošanai)?

Šie jautājumi ir ļoti vispārīgi, bet tie varētu būt arī ļoti specifiski, kas saistīti ar konkrēto nozari, kurai paredzēta dotā sistēma. Galveno atbilžu kopu, kura sniedz atbildes uz globāliem sistēmas jautājumiem, sauc par sistēmas specifikāciju. Specifikācija (*Specification*) ir sistēmas īss apraksts ar tās galvenajām funkcijām un sasniedzamajiem mērķiem.

Piemēram, sistēmas "SKOLA" īsa specifikācija varētu būt šāda:

Sistēma "SKOLA" paredzēta skolas datu glabāšanai, uzskaiti, apstrādei. Sistēma ļauj ievadīt, apstrādāt un saglabāt informāciju par skolēniem, skolotājiem, klasēm, priekšmetiem, skolēnu sekmēm. Sistēma ļauj izdrukāt skolēnu sarakstu, informācijas par klasēm, skolēnu sekmes.

Nākamais solis ir jau sīkāk un precīzāk aprakstīt informāciju, ko mēs gribam glabāt savā datu bāzē. Sākumā tas varētu būt vārdiskais datu apraksts (to izplāno, lai visa vajadzīgā informācija tiktu uzkrāta un arī nekas lieks netiktu uzglabāts):

Sistēmā "SKOLA" glabāsies dati par skolēniem, klasēm, skolotājiem, mācību priekšmetiem, stundām un skolēnu sekmēm:

- *Par skolēniem: vārds, uzvārds, dzimums, klase, personas kods, adrese, telefons, vai ir apmeklējis bērnudārzu, u.c. ziņas.*
- *Par klasēm: nosaukums, telpa, audzinātājs, profils, u.c. ziņas.*
- *Par skolotājiem tiks glabāts: vārds, uzvārds, izglītība, dzimums, personas kods, pamatpriekšmets, adrese, telefons, u.c. ziņas.*

- *Par mācību priekšmetiem: nosaukums, saīsināts nosaukums.*
- *Par stundām: mācību priekšmets, skolotājs, telpa.*
- *Par sekmēm: mācību priekšmets, skolēns, datums, vērtējums, u.c.*

Nākamais solis ir aprakstīt, kādā veidā šos datus ievadīs un kādā veidā tie saistās savā starpā:

Sākmā ievada visu informāciju par skolotājiem, tad informāciju par klasēm (audzinātāju izvēloties no jau ievadītajiem skolotājiem). Pēc tam varētu ievadīt informāciju par mācību priekšmetiem un stundām (katram norādot vienu no jau ievadītajiem skolotājiem (kurš pasniedz šo stundu) un mācību priekšmetiem. Nākamais solis ir skolēnu informācijas ievadīšana (katram skolēnam norādot klasi, kurā viņš mācās (izvēloties no klašu saraksta)). Pēc tam laika gaitā varētu fiksēt skolēnu sekmes: izvēloties, kuram skolēnam, kurā priekšmetā un datumā ir noteikts vērtējums.

Tātad, dati par skolēnu ir saistīti ar datiem par klasēm un datiem par sekmēm; dati par skolotājiem ar datiem par klasi un datiem par stundām; un dati par stundām ar datiem par priekšmetiem, datiem par sekmēm un datiem par skolotājiem.

Tādā veidā var aprakstīt, kāda būs mūsu datu bāze. Vienīgā problēma ir tā, ka šis datu bāzes aprakstīšanas veids nav pārāk pārskatāms, tāpēc IT (informācijas tehnoloģijas) ir daudz dažādu veidu (ER- modelis, DP diagrammas u.c.), kā nofiksēt datu bāzes projektu. Parasti visi šie DB aprakstīšanas līdzekļi ir grafiski. Standarts, kā pierakstīt DB, ir ER modelis.

5.1. ER-modelis, realitāšu-saišu diagrammas

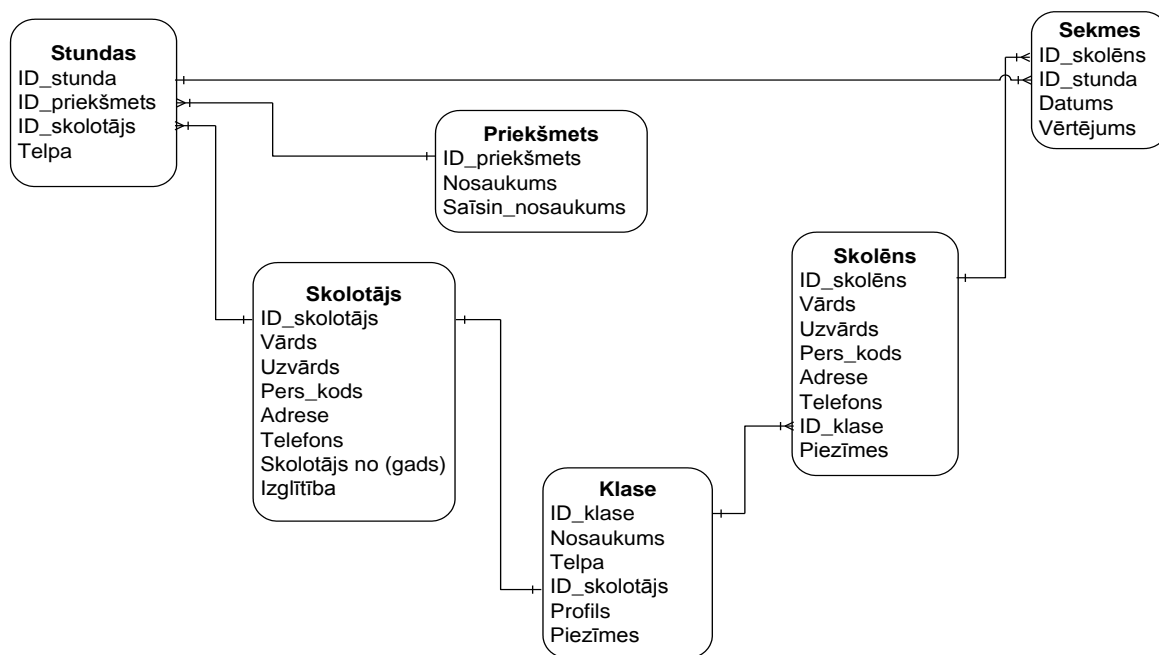
Projektējot informācijas sistēmas, ir jāņem vērā, ka šī informācijas sistēma ir kādas noteiktas organizācijas vai uzņēmuma apakšsistēma. Tāpēc, pirms informācijas sistēmas projekta, vispirms jāiegūst priekšstats par pašas organizācijas struktūru un funkcionēšanu. Šim nolūkam noderīgas ir iepriekšējā nodaļā iegūtās zināšanas. Taču, lai organizācijas vai citas sistēmas analīze būtu pilnīgāka, noderīgi ir izmantot dažādu veidu diagrammas. Mācību priekšmeta kursā galvenokārt tiek aplūkotas trīs veidu diagrammas: Realitāšu-saišu diagrammas, Datu-plūsmu diagrammas un Stāvokļu pāreju diagrammas.

Realitāšu-saišu diagrammas sākotnēji tika lietotas datu bāzu projektēšanā datu kopu vai grupu un saišu starp tām attēlošanai. Vēlāk tās sāka lietot arī problēmsfēras modelēšanai. Realitāšu-saišu diagrammu, projektējot IS organizācijās un uzņēmumos, parasti attēlo vismaz divos līmeņos. Viens no tiem atbilst organizācijai kopumā, bet otrs līmenis raksturo tikai informācijas sistēmu.

ER-modelis, RE diagrammas, relitāšu-saišu diagrammas (ER model) – ir grafisks līdzeklis datu bāzes struktūras attēlošanai. Tas sastāv no divu veidu objektiem – **entītēm** (loģiskais objekts) un **relācijām** (attiecības).

Entītes apzīmē ar taisnstūriem ar noapaļotiem stūriem. Katra entīte reprezentē vienu tabulu. Entītēm ir atribūti (lauki tabulās). Relācijas ir tabulu saišu grafisks attēlojums. Relāciju attēlo, kā taisni ar specifiskiem galiem (skatīties modelī), kas apzīmē saites veidu. Saišu veidi ir viens pret viens (starp tabulām *Klase* un *Skolotājs*), viens pret n (daudz) (*Skolēns* – *Klase*).

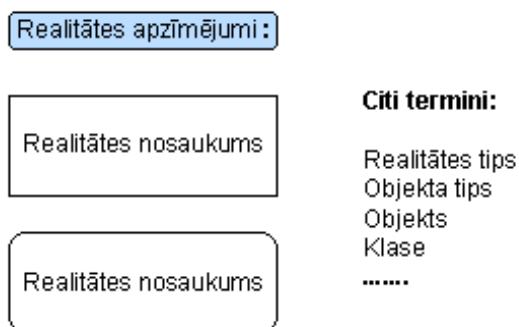
Iepriekš aprakstītas sistēmas „SKOLA” ER-modelis varētu izskatīties šādi: att. Sistēmas „SKOLA” ER-modelis.



5.1. att. ER modeļa piemērs.

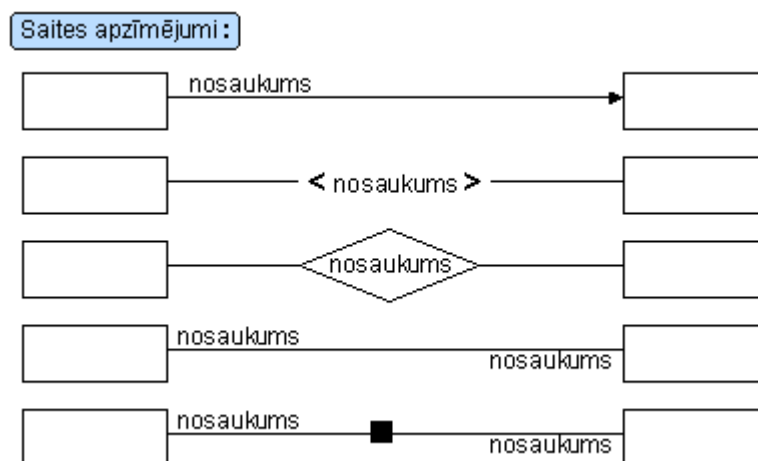
Realitāšu-saišu diagrammai ir divi galvenie elementi:

Entītija, realitāte (entity) - objekts vai objektu grupa. Objekts var būt konkrēta lieta uzņēmumā (persona, vieta, organizācijas funkcija), par kuru mēs vēlamies iegūt informāciju. Visbiežāk ar realitāti apzīmē objekta tipu (sk. att.5.2.).



5.2. att. Realitātes apzīmējums.

Saite (relationship) – noteikta veida attieksme starp objektiem (sk. att.5.3.).



5.3. att. Saites apzīmējums

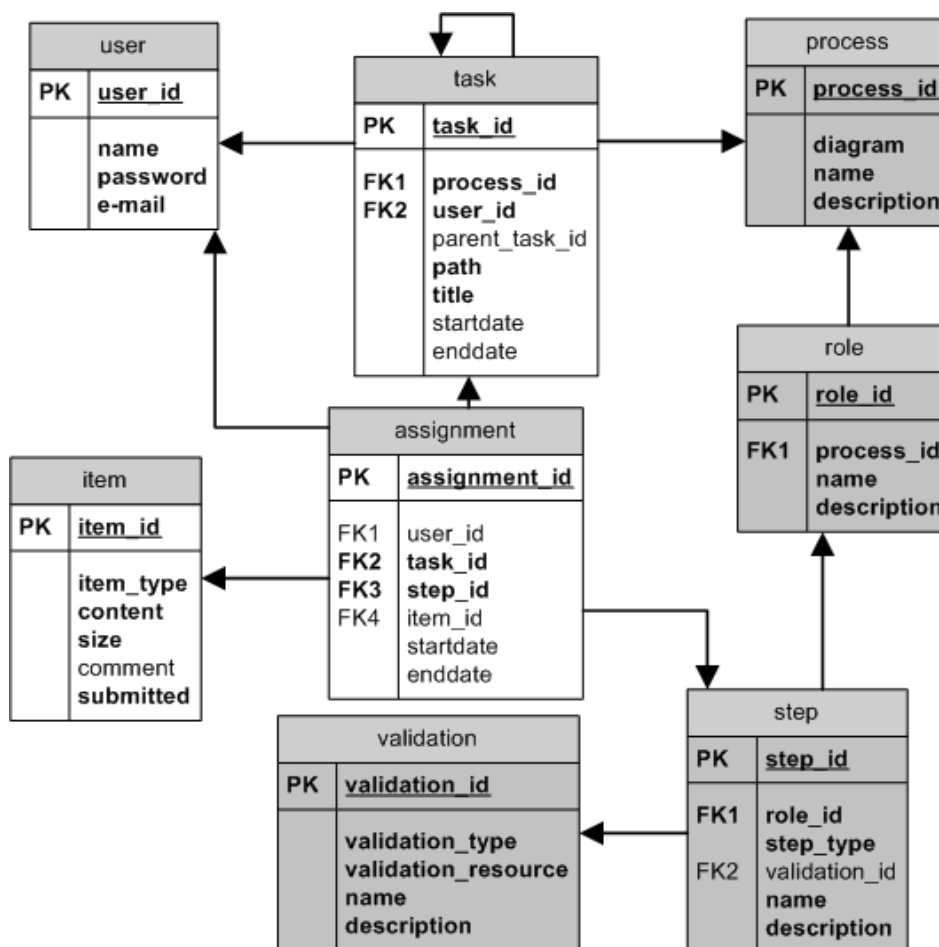
5.2. Kā izveidot RE diagrammu?

Lai realitāšu – saišu diagramma būtu pareiza un viegli saprotama, svarīgi zināt, kā izvēlēties realitātes, saites un atribūtus, kā arī kādus nosaukumus piešķirt diagrammas elementiem.

Parasti realitātes ir fiziskas lietas – organizācijas, personas, projekti vai darba vietas. Tās var būt lietas, kas notiek – piegādes, defektu pārbaudes un tamlīdzīgas. Ar realitātēm var apzīmēt arī abstraktus jēdzienus un, protams, datu elementus, tipus, kopas un grupas. Realitāšu – saišu diagrammā attēlo tās saites, kas atbilst konkrētajiem modelēšanas mērķiem.

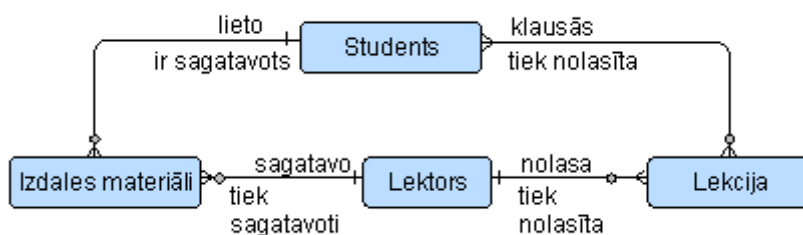
Pareiza nosaukuma izvēle dod iespēju pareizi saprast modeli gan lietotājiem, gan datorsistēmas izstrādātājiem. Parasti realitātes nosaukumam izmanto lietvārdu, bet saites nosaukumam lieto darbības vārdu. Dažkārt kādas bieži sastopamas saites vārdu aizstāj ar speciālu apzīmējumu.

ER modelis (Entity Relationship) piemērs:



5.4. att. ER modeļa piemērs zīmēt ar MS Visio.

Viens no veidiem, kā sākt zīmēt realitāšu – saišu diagrammu, ir šāds: vispirms uz lapas sazīmē taisnstūrus, kuros ieraksta sistēmas galvenās sastāvdaļas jeb elementus (realitātes), tad pārdomā, ko šie elementi sistēmā dara un kā tie savā starpā saistīti. Kad tas ir skaidrs, savieno atbilstošos taisnstūrus. Kad visi taisnstūri savienoti, izveidojas modeļa uzmetums. Pēc tā zīmē realitāšu – saišu diagrammu.



5.5. att. Realitāšu saišu diagrammas 2. piemērs.

5.3. Datu plūsmu diagrammas

Datu plūsmu diagrammas ir modelēšanas rīks, kas ļauj parādīt sistēmu kā funkcionālu procesu tīklu (Yourdan).

Datu plūsmas diagrammu citi apzīmējumi:

- DFD
- Procesu modelis
- Funkcionālais modelis
- Darba plūsmu diagramma
- “a picture of what is going on around here”
- Bubble chart
- Bubble diagram

Datu plūsmu diagrammas izmanto, lai parādītu sistēmas funkcionēšanu. Gan sistēma kopumā, gan tās elementi, saņemot ieejā kādu informāciju vai materiālus, pārveido to, kas saņemts, un pārveidojuma rezultātu var nodot tālāk ārējai videi vai citiem sistēmas elementiem. Attēlojot datu plūsmu diagrammā visu sistēmu kopumā, tās darbības rezultāts tiek nodots tikai ārējai videi. Datu plūsmu diagrammas tāpat kā realitāšu-saišu diagrammas var zīmēt gan biznesa, gan informācijas sistēmas līmenī. Datu plūsmu diagrammām var izšķirt loģisko un fizisko apakšlīmeni. Loģiskajā apakšlīmenī netiek ņemts vērā, kā tieši informācijas sistēma, piemēram, tiks programmēta, bet fiziskajā apakšlīmenī uzmanība tiek pievērsta arī izstrādāšanas aspektiem.

5.4. DP diagrammu elementi un to apzīmējumi

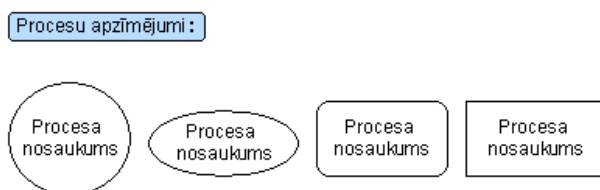
Visbiežāk datu plūsmu diagrammās izmanto šādus četrus pamatelementus:

Process – parāda, ko sistēma dara. Parasti tas pārveido ieejas datus par kādiem citiem izejas datiem, līdz ar to procesā var gan ieiet, gan iziet viena vai vairākas datu plūsmas.

Procesa sinonīmi:

- funkcija
- transformācija
- burbulis

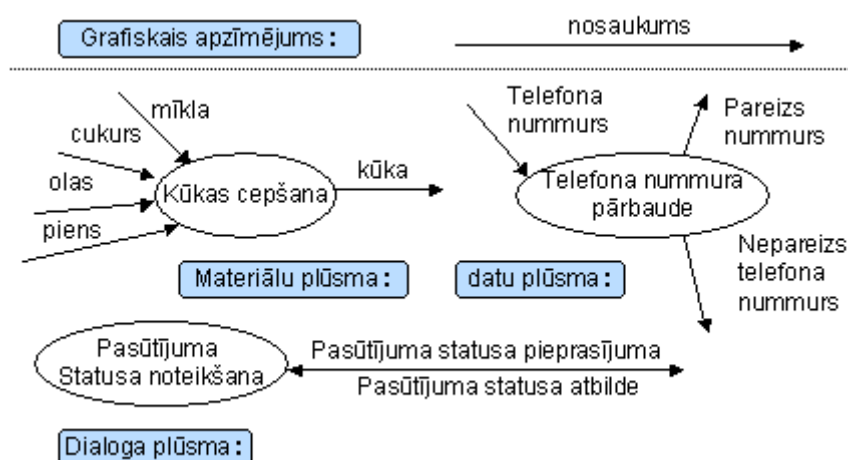
Procesu apzīmējumi:



5.6. att. Procesu apzīmējumi.

Datu plūsma – attēlo datu (vai materiālu) plūsmu sistēmā, tas ir līdzīgi telefona vadiem, pa kuriem notiek informācijas apmaiņa starp pilsētām.

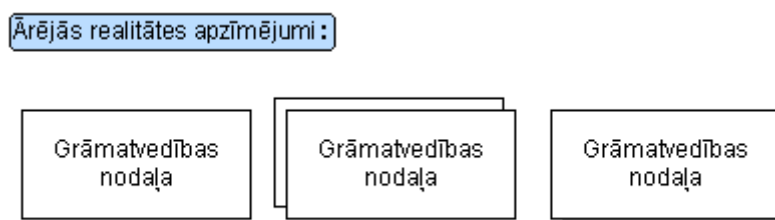
Datu plūsmas apzīmējumi:



5.7.att. Datu plūsmas apzīmējumi

Ārējā realitāte - vienmēr atrodas ārpus sistēmas. Ārējās realitātes ir objekti vai procesi, kas vai nu sniedz sistēmas darbībai nepieciešamos datus (materiālus) vai arī tos izmanto. Tātad, tās atrodas sistēmas ieejas un izejas plūsmu galos. Dažkārt realitātes, kas atrodas ieejas plūsmu galā sauc par "avotiem", bet realitātes, kas atrodas izejas plūsmu galā – par "izlietnēm". Svarīgi atcerēties, ka datu plūsmu diagrammā tiešas saites starp ārējām realitātēm nav jāuzrāda.

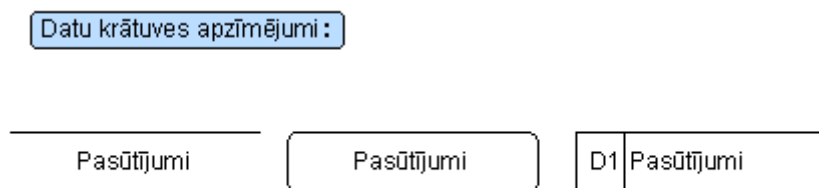
Ārējās realitātes apzīmējumi:



5.8.att. "Ārējās realitātes apzīmējumi"

Datu krātuve – glabā datus, kurus izmanto sistēma. To izmanto process, kurš vai nu "ņem" datus no datu krātuves vai "liek" datus datu krātuvē.

Datu krātuves apzīmējumi:



5.9.att. Datu krātuves apzīmējumi

5.5. Kā izveidot DP diagrammu?

Likumi, kas jāievēro, zīmējot datu plūsmu diagrammas:

1. Katram procesam var būt viena vai vairākas ieejošās datu plūsmas, kā arī viena vai vairākas izejošās datu plūsmas. Tātad nevar būt process, kuram nav nevienas ieejošās vai izejošās plūsmas.
2. Datu plūsmām ir savs nosaukums un unikālais numurs. Vismaz viens datu plūsmu gals ir saistīts ar procesu. Tās var zīmēt starp:
 - diviem procesiem,
 - no datu krātuves uz procesu,
 - no procesa uz datu krātuvi,
 - no ārējās realitātes uz procesu,
 - no procesa uz ārējo realitāti.
3. Visas datu plūsmas, kuras ieiet konkrētā procesā vai iziet no konkrētā procesa augstākā hierarhijas līmenī, ir obligāti jāatspoguļo šī procesa detalizētākajā līmenī.

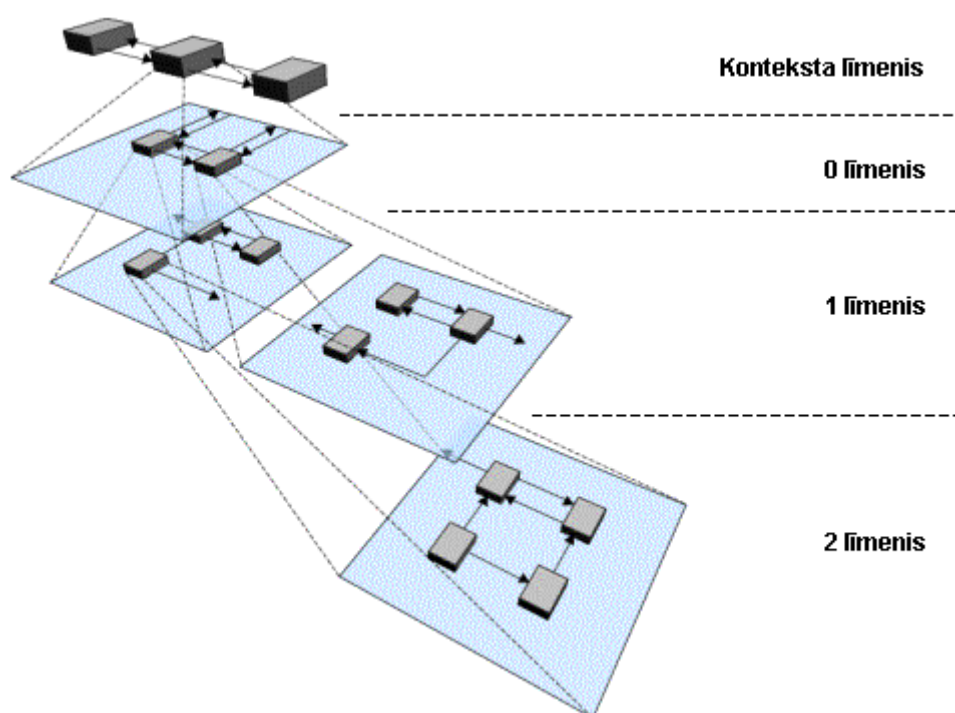
Bez pamatlikumiem ir arī likumi, kuri attiecas uz elementu nosaukumiem:

1. Procesiem nosaukums sastāv no vienkāršas frāzes un to ir ieteicams aprakstīt vienā teikumā. Nosaukums ir darbības atspoguļojums, tāpēc visbiežāk tas ir darbības vārds ar paskaidrojošo lietvārdu, piemēram, "preces saņemšana".
2. Datu krātuvju nosaukumos lieto lietvārdus, pamatterminus un specifiskus terminus, tādus kā PERSONAS_DATI, PAGĀJUŠĀ_GADĀ_IENĒMUMI, bet netiek lietoti kombinētie termini, jo katram šim terminam ir sava struktūra. Vēl svarīgi ievērot, ka katrai datu krātuvei ir jābūt ar unikālu nosaukumu, kas atspoguļo datu krātuves saturu. Bieži par datu krātuves nosaukumu izvēlas tajā ieejošās datu plūsmas nosaukumu daudzskaitlī.

3. Datu plūsmu nosaukumiem vislabāk būtu lietot tikai vienu vārdu, bet ja šis vārds atkārtojas vairākām plūsmām, tad nevar zināt, ko tieši konkrētā datu plūsma nozīmē. Tāpēc vārda nozīme būtu jāpaskaidro ar apzīmētāja palīdzību. Piemēram, bieži tiek lietots vārds "dokumenti". Lai labāk varētu saprast, kādi konkrēti dokumenti kurā situācijā tiek lietoti, lieto paskaidrojošo lietvārdu: "personas dokumenti", "piegādātāju dokumenti", "administratīvie dokumenti". Tas galvenokārt attiecas uz plūsmām starp procesiem. Ja plūsma saista procesu ar datu krātuvi, tad atkarībā no situācijas var rīkoties divējādi, - vai nu vispār nelietot plūsmu nosaukumu, vai arī plūsmu nosaukumā attēlot, kādus datus no datu krātuves izmanto vai tajā ievieto, piemēram, PERSONAS_NR+ALGA, PIEGĀDĀTĀJA_NR + DAUDZUMS.
4. Ārējo realitāšu nosaukumiem izmanto lietvārdu, kurš apzīmē sistēmas informācijas avotus (personu, vietu, lietu, procesu).

Kā izveidot datu plūsmu diagrammu?

Parasti tiek zīmētas vairāklīmeņu datu plūsmu diagrammas, kur katrs pakārtotais līmenis detalizētāks apraksta kāda konkrētā līmeņa tekošo procesu. Shematiskai vairāklīmeņu datu plūsmu diagrammas attēlošanai ir izmantots W. Wooiinii.m piemērs (Woodman. 1988) piemērs:



5.10.att. Datu plūsmu diagramma

E. Yordon (Yordon, 1989) piedāvā šādas četras vadlīnijas vienkāršas un pareizas datu plūsmu diagrammas uzzīmēšanai:

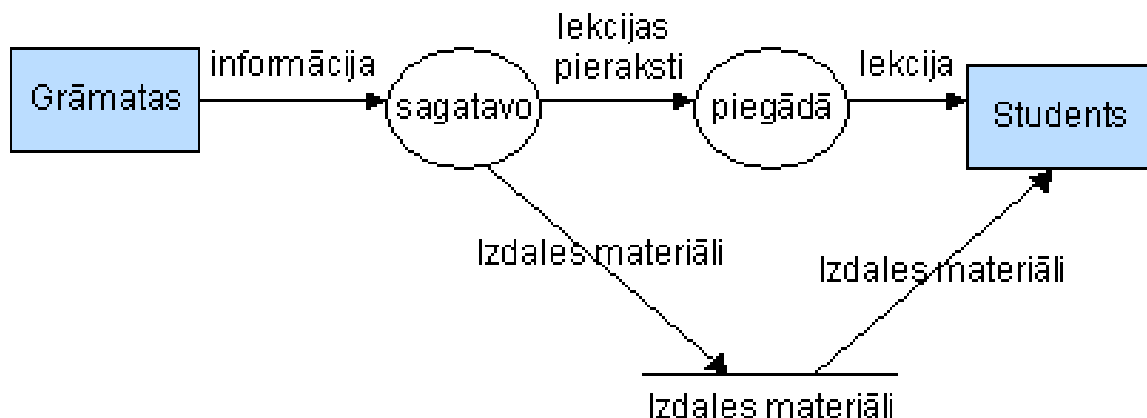
1. Jāizvēlas precīzi procesu, datu plūsmu, datu krātuvju un ārējo realitāšu nosaukumi. Tas atvieglos gatavā modeļa saprašanu sistēmas lietotājam, pasūtītājam, sistēmanalītiķim un programmētājam. Lietotājam bieži patīk lietot specifiskus saīsinājumus un vietējos terminus, bet tos ir grūti saprast citiem (ārpus sistēmas) lietotājiem, kuri vēlāk strādās ar šo pašu sistēmu, tāpēc ir ieteicams lietot nosaukumus, kurus sapratīs visi. Savukārt, programmētājiem ir tendence likt nosaukumus atbilstoši programmēšanas valodām, piemēram, PROCEDURE, SUBSYSTEM un tml. Konstruējot datu plūsmu diagrammas, arī no šādiem nosaukumiem ieteicams izvairīties.

2. Procesu numurēšana. Ja procesi ir numurēti, ar tiem vieglāk darboties, kad notiek sistēmas modificēšana vai modelēšana. Pēc procesa numura var noteikt tekošā procesa atrašanās vietu un svarīgumu.

3. Izvairieties no sarežģītām datu plūsmu diagrammām. Diagrammai ir jābūt pārskatāmai ne tikai sistēmas izstrādātājam, bet arī sistēmas lietotājam. Nevajag izveidot daudz procesu, ārējo realitāšu, datu krātuvju un datu plūsmu, jo tas sarežģīs diagrammas saprašanu. Ir dažādi viedokļi par to, cik daudz procesu, ārējo realitāšu, datu krātuvju un datu plūsmu ir jāattēlo vienā līmenī. Vieni uzskata, ka ir jāattēlo tik daudz, lai viss būtu pārskatāms uz vienas lapas, citi uzskata, ka ne vairāk kā 50 procesi kopā pa visiem diagrammas līmeņiem. Domas dalās arī attiecībā uz līmeņu skaitu. Parasti lietotājam pietiekot ar 2 līmeņiem – konteksta un 1. līmeni.

4. Diagrammas izstrādāšanai nepieciešamais laiks. Datu plūsmu diagrammu izstrādāšanas procesa laikā to parasti ir nepieciešams vairākas reizes pārzīmēt līdz diagramma ir tehniski pareiza, atbilst lietotāja prasībām un apstiprināta ar parakstu. Proti, tas nozīmē, ka datu plūsmu diagrammas izstrādāšana prasa daudz laika, tāpēc ar to ir jāreķinās uzreiz - visi darbi ir jāplāno un jāveic regulāri.

Datu plūsmu diagrammas piemērs:



5.11.att. Datu plūsmu diagrammas piemērs

Citi procesu piemēri

1. **Apdrošināšanas polise:** "aģents" pēc klienta pieteikuma izveido polises sagatavi, "eksperts" novērtē risku un autorizē apdrošināšanas veida un likmes izvēli, "aģents" pabeidz polisi, ko izdrukā un izsniedz klientam
2. **Fotoalbūma veidošana:** "vadītājs" izveido tukšu albūmu, viens vai vairāki "dalībnieki" paralēli viens otram pievieno katrs pa kādai bilžu kolekcijai (piemēram, arhivētu JPG attēlu direktorijas) un "vadītājs" izveido satura rādītāju, kas dod iespēju apskatīt šīs kolekcijas noteiktā secībā.
3. **Java praktiskā darba taisīšana:** "pasniedzējs" pievieno "darba aprakstu", "students" pievieno "risinājumu", pasniedzējs pievieno "testēšanas rezultātus" un "atzīmi".

!!! Tikai tad, kad ir izstrādāts detalizēts un standartiem atbilstošs DB projekts var ķerties klāt tā realizācijai. Bet, kas attiecas uz realizāciju, viens no projektēšanas punktiem ir realizācijas vide, jo katrai videi (DBVS) ir savas īpatnības, kas ir jāparedz jau projektējot sistēmu.

6. Datu bāzes struktūras projektēšanas grafiskie līdzekļi

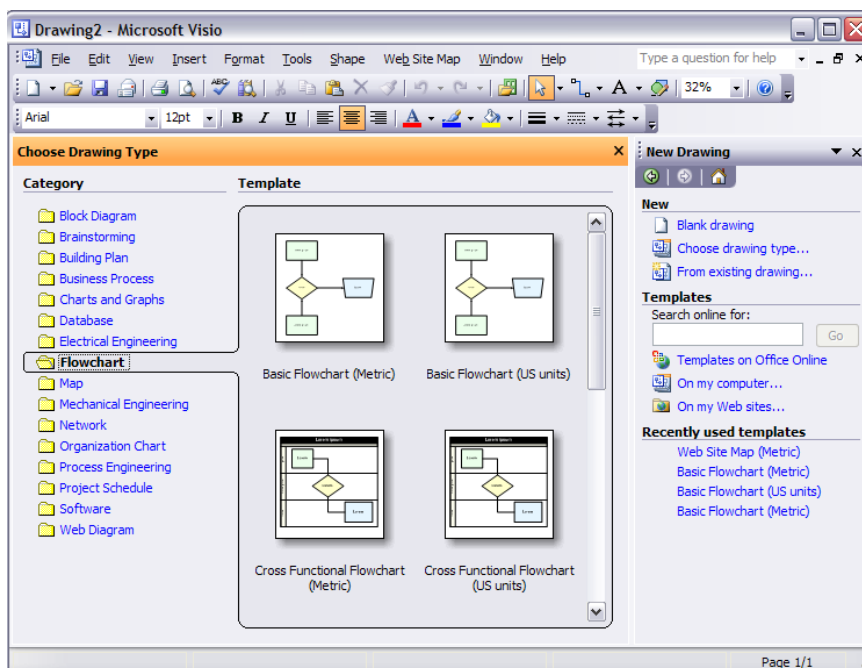
Viens no līderiem shēmu un biznesa diagrammu zīmēšanā programmas produktiem ir Microsoft ražots produkts Visio.

Visio ir shēmu izveides programma, kas palīdz izveidot biznesa un tehniskās shēmas, lai dokumentētu un organizētu sarežģītas idejas, procesus un sistēmas. Programmā Visio izveidotās shēmas ļauj skaidri, kodolīgi un efektīvi vizualizēt un pasniegt informāciju tādā veidā, kā to nevar izdarīt tikai ar teksta un skaitļu palīdzību. Turklāt Visio automatizē datu vizualizēšanu, sinhronizējot tieši ar datu avotiem, lai tiktu izmantotas jaunākās shēmas. Šo programmu var pielāgot, lai tā atbilstu katra uzņēmuma vajadzībām.

Piezīme. Turpmāk tekstā tiks aprakstītas pamatdarbības Visio programmā, balstoties uz MS Visio 2003 versiju.

6.1. MS Visio vide

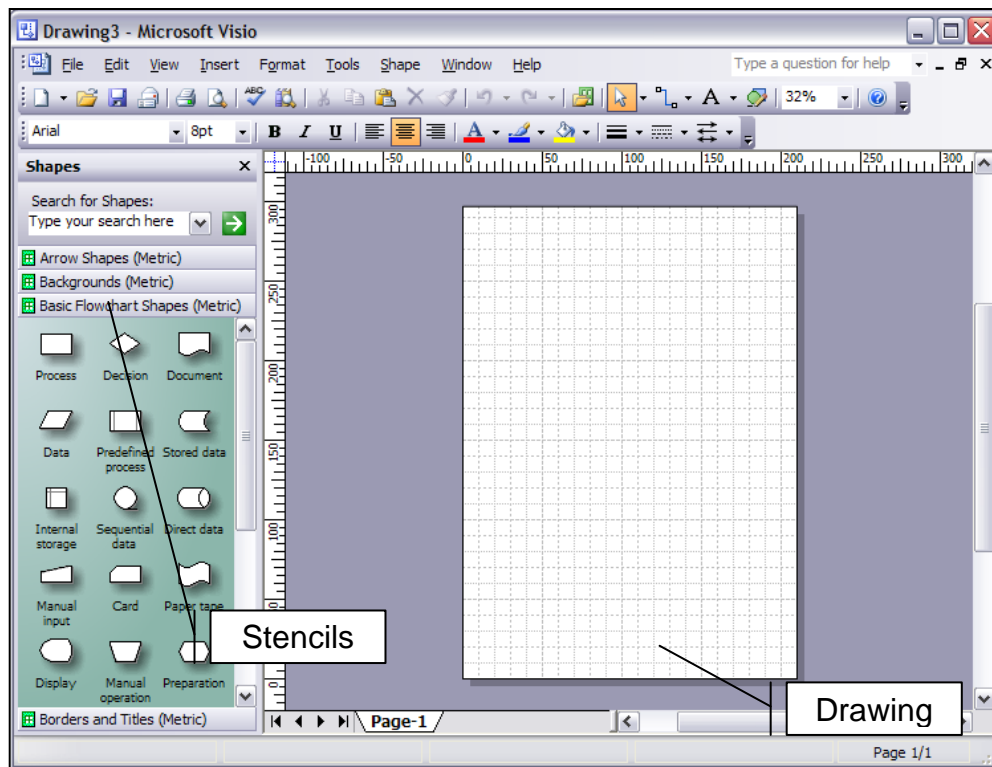
MS Visio aktivizēšana – Start/Programs/Microsoft Office/Microsoft Office Visio 2003:



6.1. att. MS Visio veidnes izvēles logs.

MS Visio interfeisa kreisajā pusē ir trafareta logs (*Stencils*), labajā - shēmu zīmēšanas logs *Drawing*.

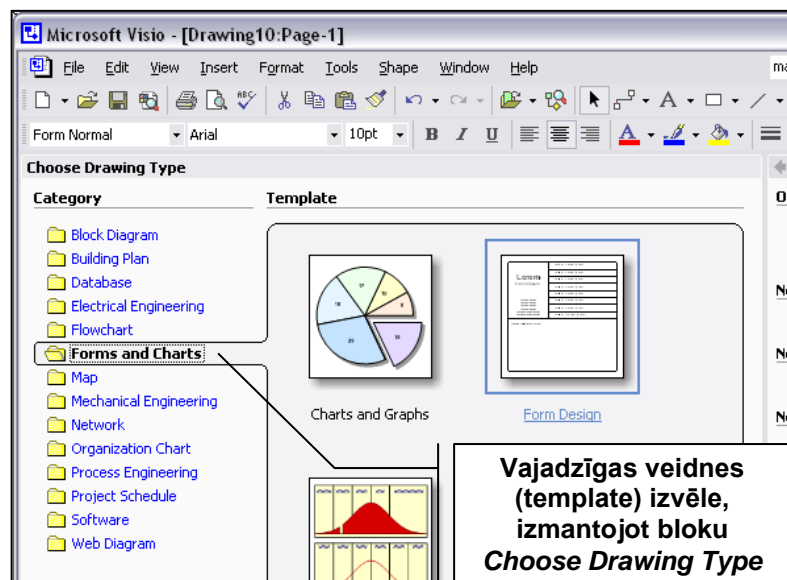
Interfeisa izskata maiņai - izvēlne *Tools->Customize*.



6.2. att. MS Visio interfeiss.

Visi diagrammu veidi ir sadalīti vairākās objektorientētās grupās (tiek piedāvāti ***Choose Drawing Type*** blokā)

Aktivizē ar komandu ***File->New->Choose Drawing Type...***

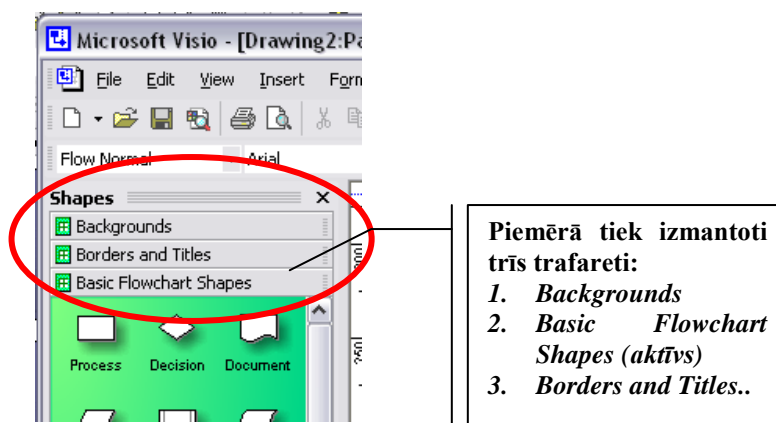


6.3. att. MS Visio veidnes izvēle.

6.2. MS Visio failu tipi.

Trafarets *.VSS (*Stencil*) – grafisko elementu virkne, izmantojot kuru var zīmēt dažāda veida grafiskus attēlus. Ir iespējams veidot arī savus trafaretus.

Attēlu formēšanai programmā var ielādēt vienu vai vairākus trafaretus (*Stencil*), kas tiks attēloti atbilstošajā logā *Shapes*:



6.4. att. MS Visio trafareti.

Darbavieta *.VSW (*Workspace*) – fiksē ielādētus trafaretus un loga fizisku izvietojumu uz ekrāna.

Lai atceltu darbavietas saglabāšanu – File/Save As/ un zem pogas Save (labajā pusē) sarakstā jāatķeksē Workspace, saglabājas tikai diagramma, trafareti netiek saglabāti). Lai atsevišķi saglabātu darbavietu, izmanto Alt+F2 komandu.

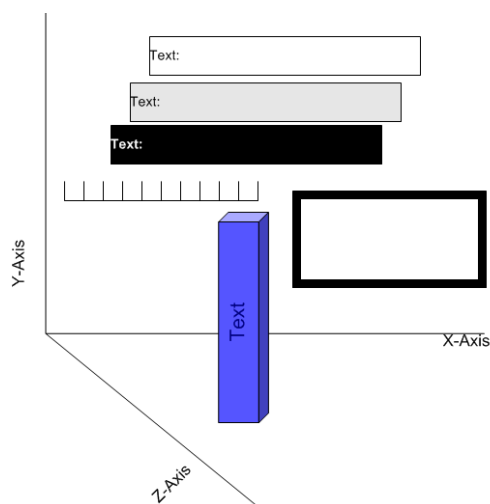
Attēls *.VSD (*Drawing*) – fails, kurā tiek saistīti grafiskie, attiecīgā veida un stila, objekti.

6.3. Darbs ar formām

Viens no grafisku objektu veidiem ir formas: viendimensiju (1D) un divdimensiju (2D)

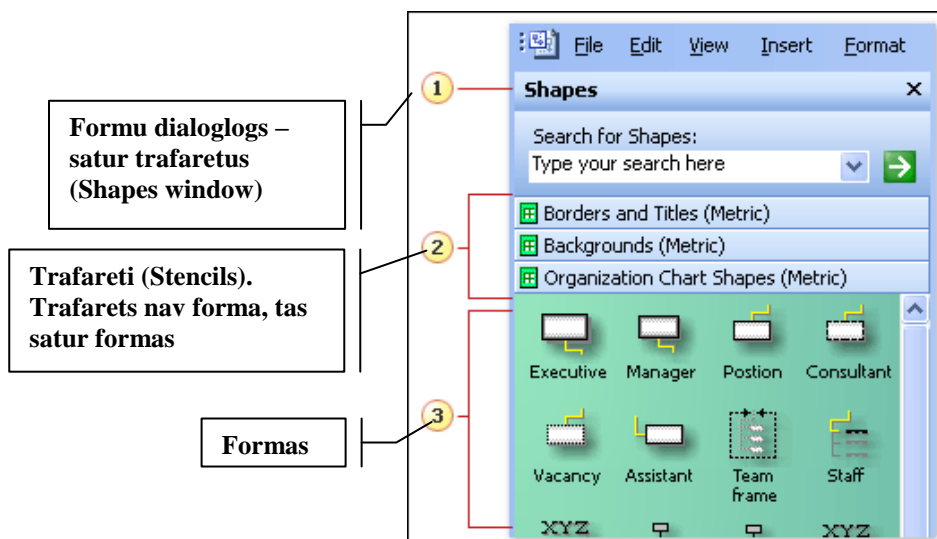
Darbības, ko var veikt ar formām:

- iezīmēšana (Select),
- kopēšana (Copy),
- izgriešana (Cut),
- savienošana (Connection).



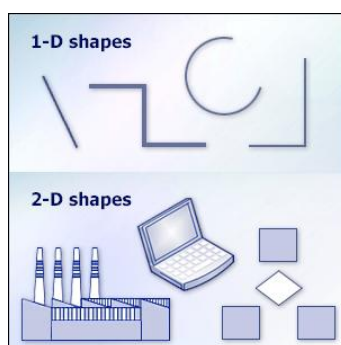
6.5.att. MS Visio grafiskie objekti.

Formas atrodas trafaretos, savukārt trafareti – formu dialoglogā:



6.6.att. MS Visio trafareti un formas.

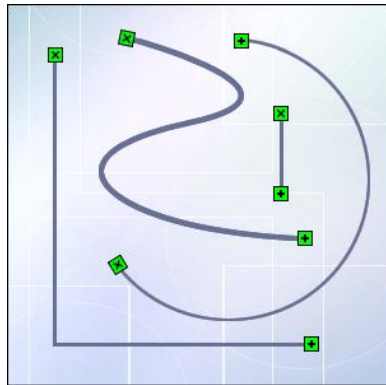
Pastāv divu veidu formas: viendimensiju 1-D un divdimensiju 2-D formas:



6.7.att. MS Visio 1D un 2D formas.

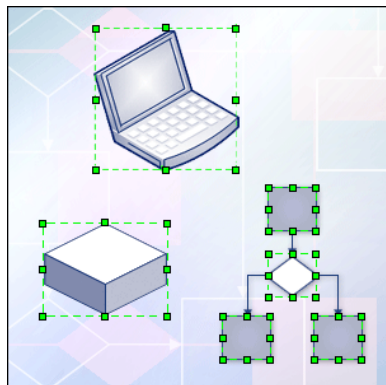
1-D un 2-D galvenās īpašības.

Ja iezīmēt 1-D formas, tad tās sākuma būs X zīmē un beigās plusa (+) zīme. 1-D formas pārsvarā ir savienotāji un līnijas.



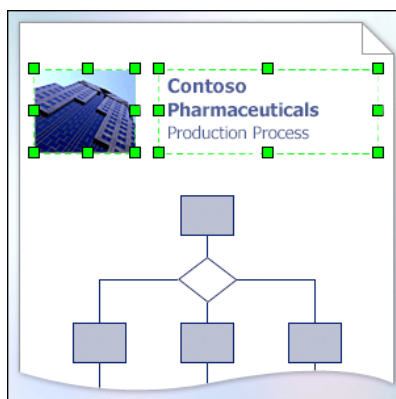
6.8.att. MS Visio 1D formas.

Iezīmējot 2-D formu, tai apkārt parādās 8 jūtīgie punkti.



6.9.att. MS Visio 2D formas.

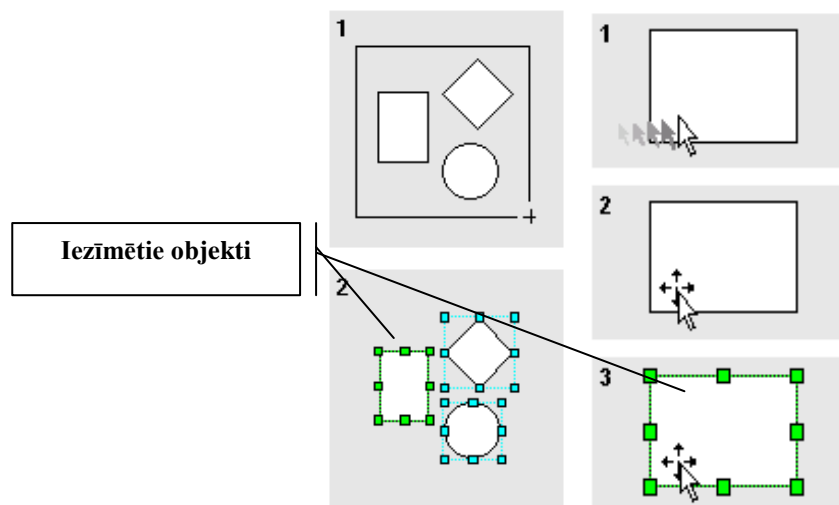
Visio programmā ikviens objekts ir forma (**shape**), ieskaitot attēlus un tekstu.



6.10.att. MS Visio objekti.

Formu iezīmēšana (Select).

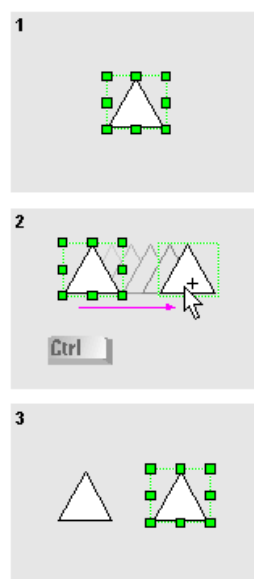
Lai iezīmētu vairākas formas vienlaicīgi, jāizmanto pogas SHIFT



6.11.att. Formu iezīmēšana.

Formu kopēšana (Copy).

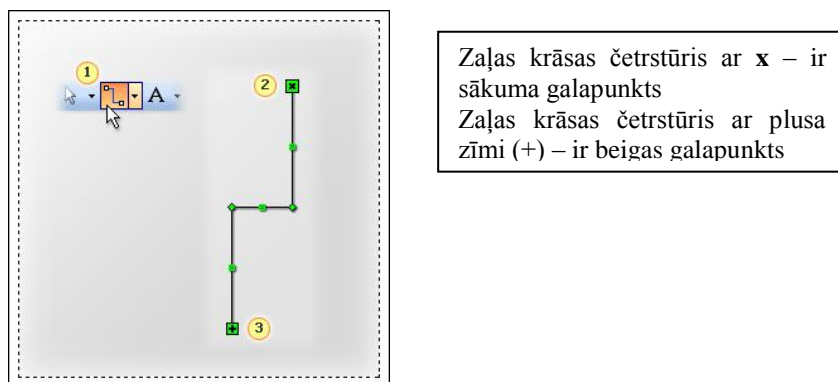
1. Ar izvēlnes palīdzību.
2. Ar CTRL pogas palīdzību.



6.12.att. Formu kopēšana.

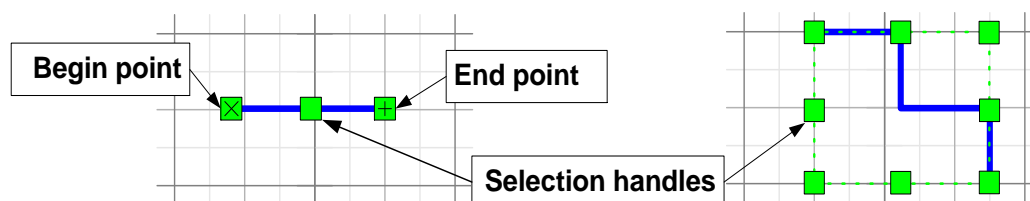
Formu savienošana (Connection).

Līnijas, ko izmanto formu savienošanai, sauc par savienotāju. Savienotājus var zīmēt, izmantojot **Connector** instrumentu no standarta rīku joslas.



6.13.att. Formu savienošana.

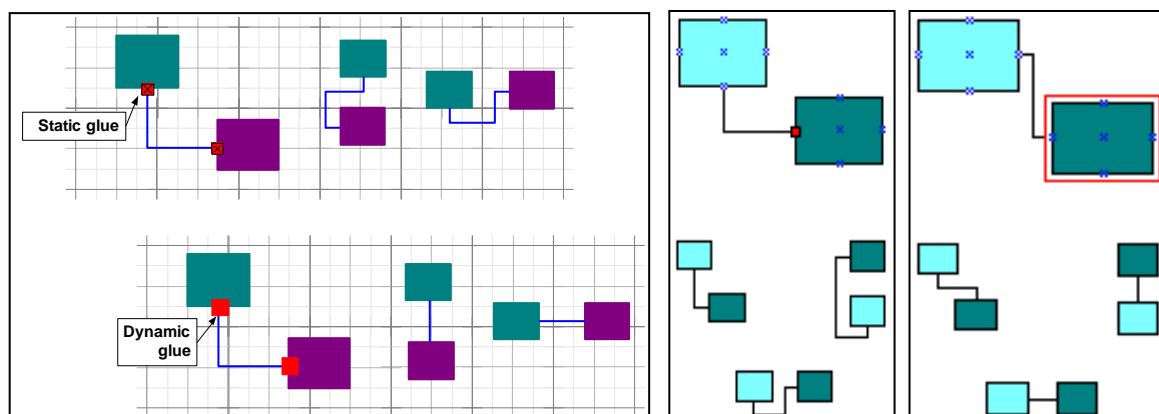
Savienotājs – ir 1D forma ar galapunktiem (*endpoints*). Savienotāji tiek izmantoti lai savienotu 2D formas.



6.14.att. Formu savienotāju galapunkti.

Pastāv divi savienošanas veidi:

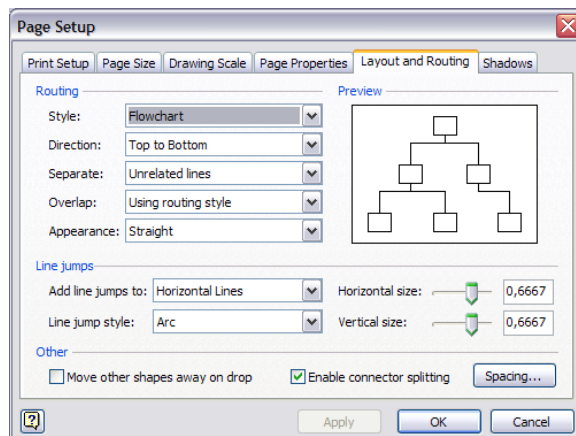
1. **Forma-forma** (*Shape-to-Shape*) – savieno formas un ļauj savienojuma bultai mainīt piesaistīšanas punktu apkārt formai. Tas ir dinamisks savienojums (Dynamic): notiek ar formu centriem.
2. **Punkts-punkts** (*Point-to-Point*) – savieno formas, fiksē savienojuma bultas pozīciju konkrētā punktā. Tas ir statisks savienojums (Static): notiek ar formu aktīviem punktiem.



6.15.att. Formu savienošana.

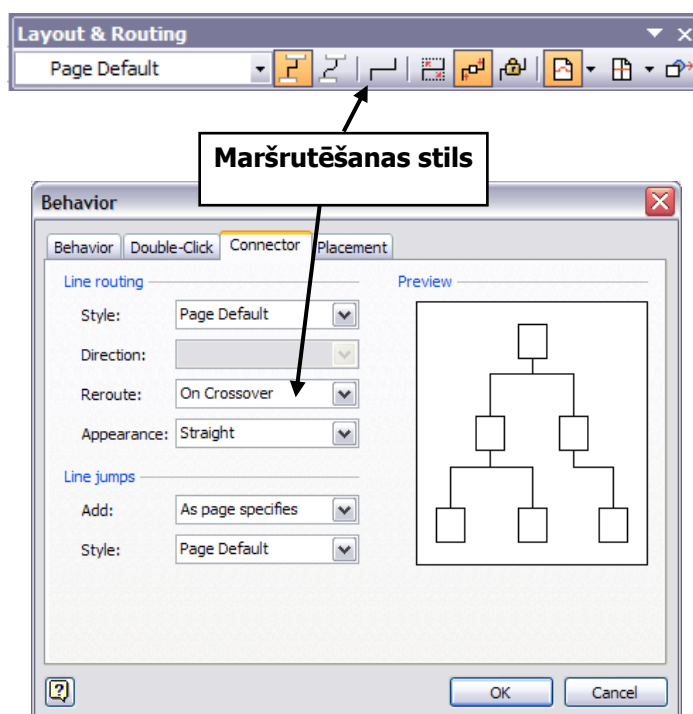
Savienotāju maršrutēšana (Routing), uzstādījumi.

File/Page Setup lapa *Layout and Routing*



6.16.att. Formu savienotāju maršrutēšanas uzstādījumi.

Rīku josla *Layout & Routing* vai izvēlne *Format/Behavior* lapa **Connector**



6.17.att. Formu savienotāju maršrutēšanas uzstādījumi.

Izvēlnes punkts *Format/Behavior* ir aktīvs tikai tad, kad ir iezīmēts kaut viens savienotājs.

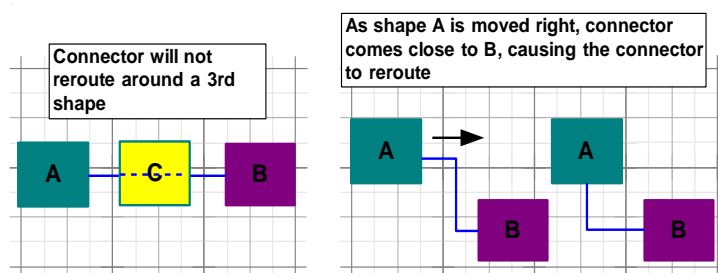
Maršrutēšanas uzstādījumi:

Reroute – pārveido līnijas (savienotāja) maršrutu,

Reset Connector – pārveido savienojuma maršrutu, pat arī ja bija uzstādīts parametrs **Never Reroute**,

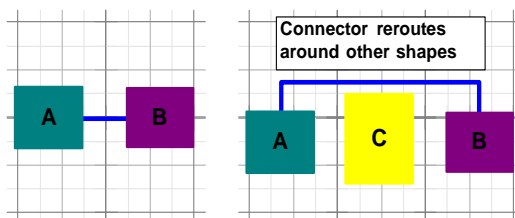
Reroute Freely – savienojuma veids, kurš automātiski izvēlās optimālāko bultas (savienojuma) maršrutu (starp formām),

Reroute on Crossover – savienojuma veids, kurš pārveido maršrutu, kad savienotāja galapunkti (bultas) krustojas.



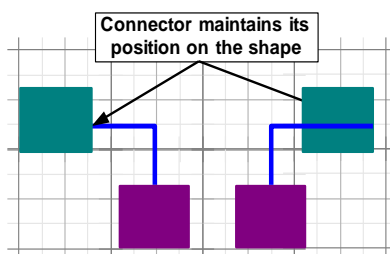
6.18.att. Reroute on Crossover.

Reroute as Needed – maršruta pārveidošana pēc vajadzības, piemēram, kad uz līnijas tiks pārvietota cita forma (**Shape**).



6.19.att. Reroute as Needed.

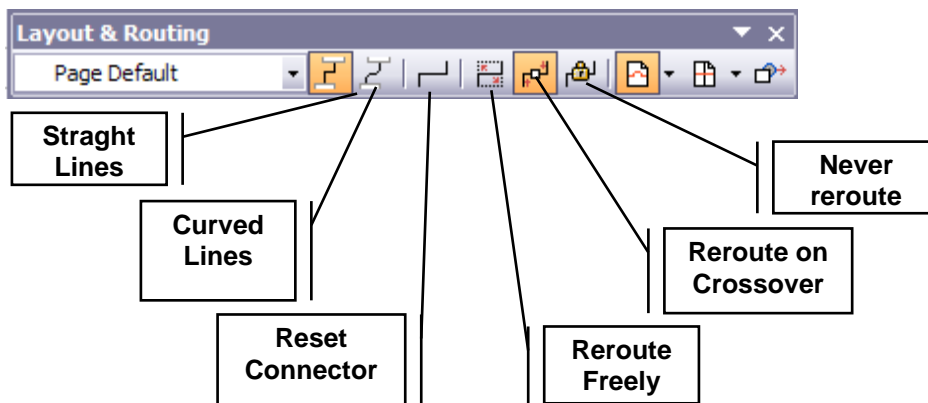
Never Reroute – maršruts netiks pārveidots



6.20.att. Never Reroute .

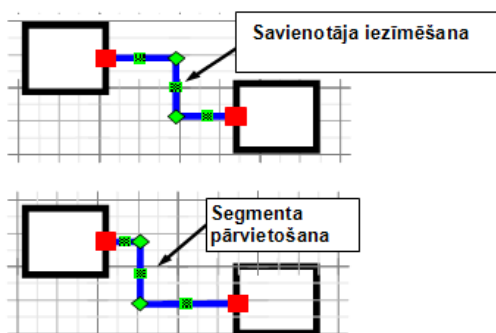
Kad tiek savienotas divas formas (*Shape*), **Routing** vērtība pēc noklusējuma ir **Reroute Freely**. Ja lietotājs pats redīgē savienotāju, tad **Routing** veids tiks automātiski pārslēgts **Reroute on Crossover** režīmā.

Virziena veids **Reroute on Crossover** var automātiski pārslēgties pret **Reroute Freely**, gadījumā, ja formas tika būtiski pārvietotas

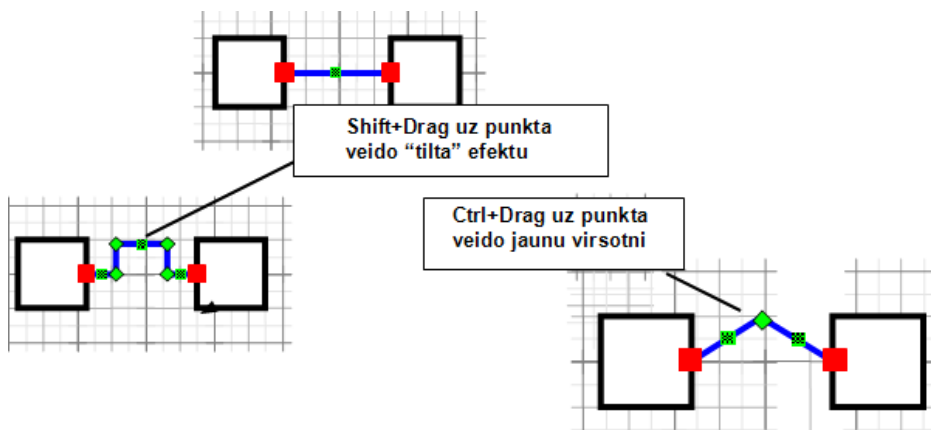


6.21.att. *Layout & Routing* rīku joslas iespējas.

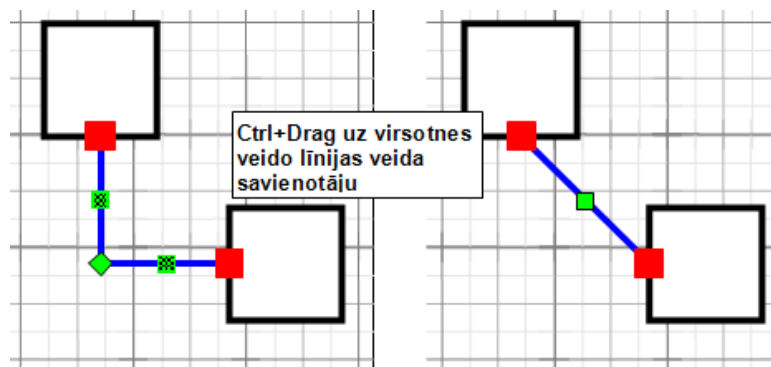
Manuāla savienošana



6.22.att. Formu manuāla savienošana.

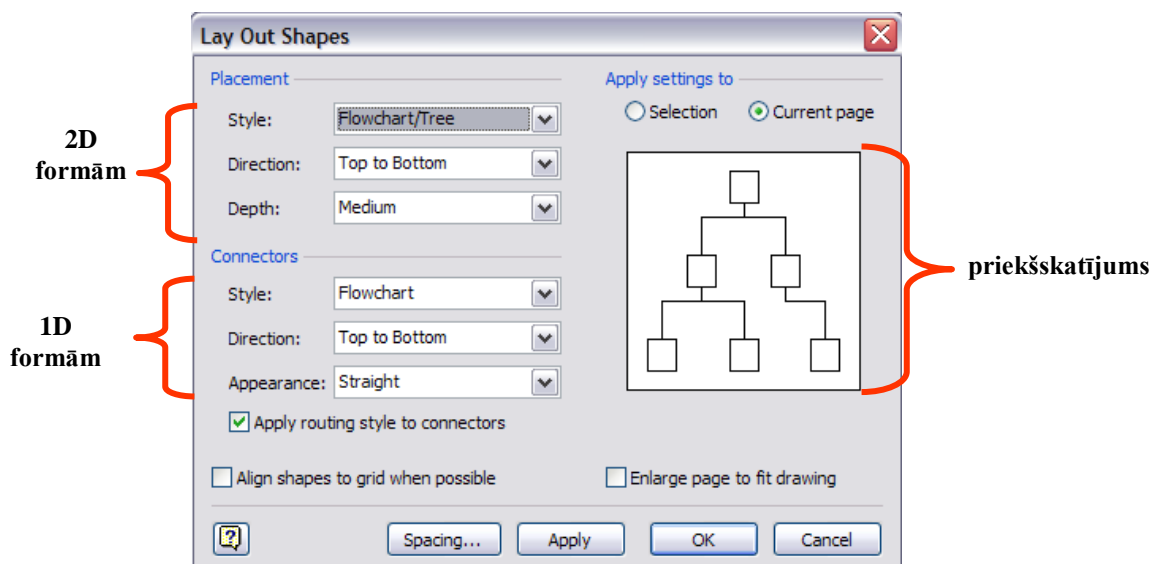


6.23.att. Formu manuāla savienošana.



6.24.att. Formu manuāla savienošana.

Lai nomainītu formu izkārtojumu uz lapas, izmanto *Shape/Lay Out Shapes* dialoglogu.



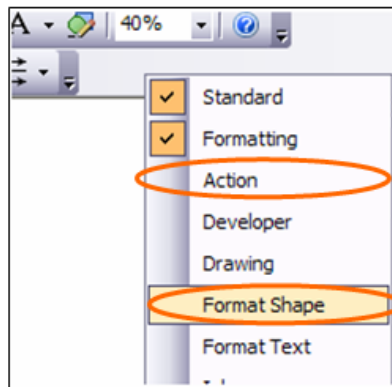
6.25.att. *Lay Out Shapes* dialoglogs.

Lai pievienotu tekstu savienotājam, ir jāiezīmē savienotais un jāsāk ievadīt tekstu no tastatūras.

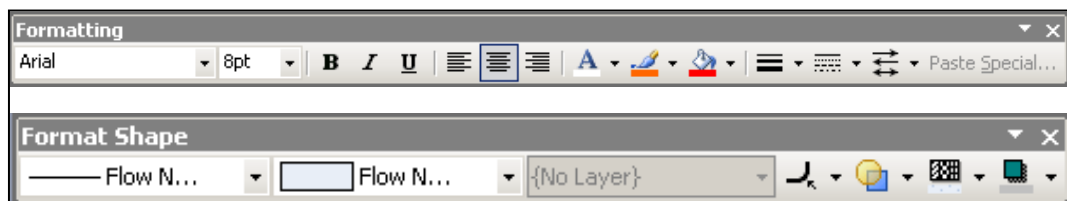
6.4. Formu un savienotāju noformēšana

Formas noformējumam paredzētas rīku joslas:

1. *Action*
2. *Format Shape*
3. *Formatting*



6.26.att. Rīku joslu izvēle.



6.27.att. Rīku joslas formu noformējumam –
Formatting un *Format Shape*.

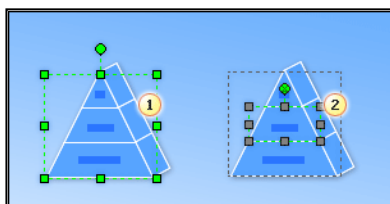
Rīku joslas *Action* ikonu apraksts no kreisās uz labo pusi:



6.28.att. Rīku josla *Action*

- Align*** – formu līdzināšana lapā
- Distribute shapes*** - formu izkārtojuma veidi
- Connect Shapes*** – formu automātiskā savienošana
- Layout Shapes*** – formu izvietojuma dialoga loga atvēršana
- Flip*** - formu spoguļattēla izveide
- Flip Horizontal*** – formas horizontālais spoguļattēls
- Flip Vertical*** – formas vertikālais spoguļattēls
- Rotate*** – formu pagriešana
- A*** – teksta pagriešana uz 90°
- BringToFront*** – izvietot priekšplānā
- SendToBack*** – izvietot aizmugurē
- Group*** – formu grupēšana
- Ungroup*** - atgrupēt

Kad objekti ir sagrupēti, visus noformēšanas parametrus tiek pielieto katrai formai grupā.



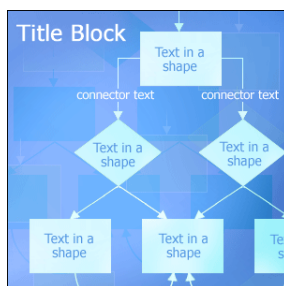
6.29.att. Objekta noformēšana grupā.

Lai noformētu kādu konkrētu grupas objektu, tas no sākuma jāiezīmē (iezīmējot no sākuma grupu, un pēc tam vajadzīgo objektu) un tikai pēc tam jāmaina noformējuma parametri (sk. 6.29. att.).

6.5. Darbs ar tekstu

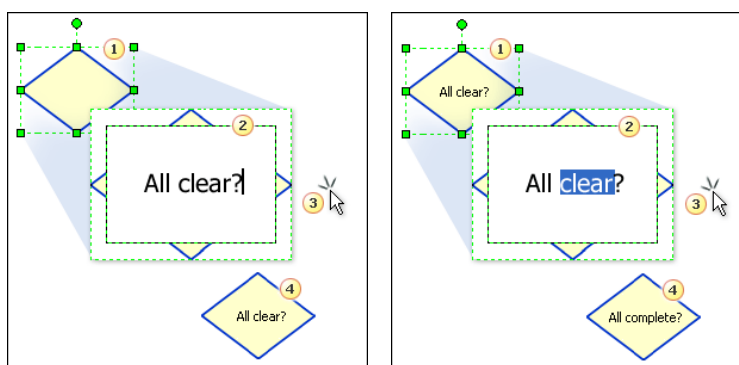
MS Visio nodrošina:

- iespēju pievienot tekstu jebkurai formai (ieskaitot bultas un savienotājus), kā arī pievienot neatkarīgu tekstu attēlam;
- teksta rediģēšanu un sintakses pārbaudi, kā arī meklēšanu, kā to ļauj veikt vairāki teksta redaktori.



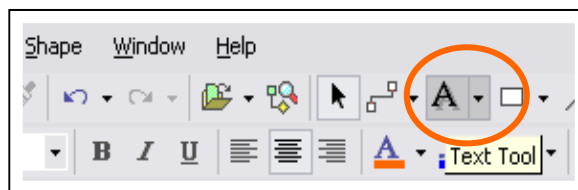
6.30.att. Darbs ar tekstu.

Lai pievienotu tekstu objektiem, ir jāiezīmē objekts un jāsāk ievadīt tekstu.



6.31.att. Teksta pievienošana objektiem

Teksta pievienošana formām un savienotājiem notiek, izmantojot instrumentu **Text Tool (A)**, kas atrodas uz **Standard** rīku joslas.

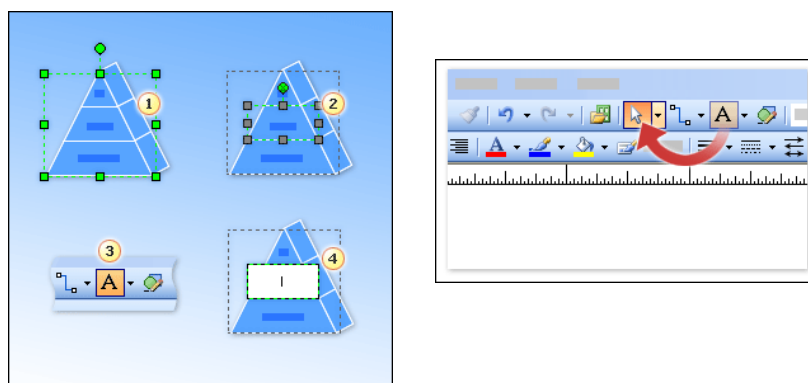


6.32.att. Instrumenta **Text Tool** aktivizēšana.

Šo režīmu var izmantot visiem objektiem, kam MS Visio ļauj pievienot tekstu, kā arī ievietojot tekstu patvaļīgajā vietā lapā.

Teksta pievienošana objektiem, kas ir sagrupēti:

1. Grupas iezīmēšana
2. Objekta iezīmēšana (grupā)
3. **Text tool** instrumenta izvēle
4. Teksta ievads



6.33.att. Teksta pievienošana objektam grupā.

Rīku joslas Format Text ikonu apraksts no kreisās uz labo pusi:



6.34.att. Rīku josla Format Text.

Style - teksta veids

Increase (Decrease) Font Size (A') – fonta izmēra palielinājums (samazinājums)

Strikethrough – pārsvītrots

Small Caps – mazie burti

Superscript – augšējais indekss

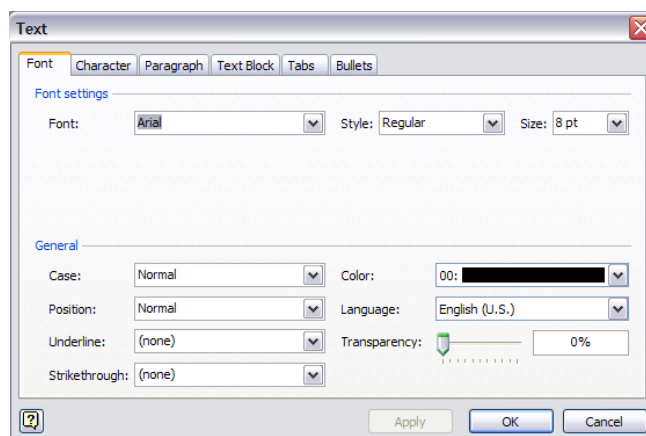
Subscript - apakšējais indekss

Align – līdzinājums (top, middle, bottom)

Decrease (Increase) Indent – palielināt (samazināt) atkāpi

Decrease (Increase) Paragraph Spacing – palielināt (samazināt) atkāpi

Papildus teksta noformēšanai izmanto ***Text*** dialoglogs ***Format/Text***



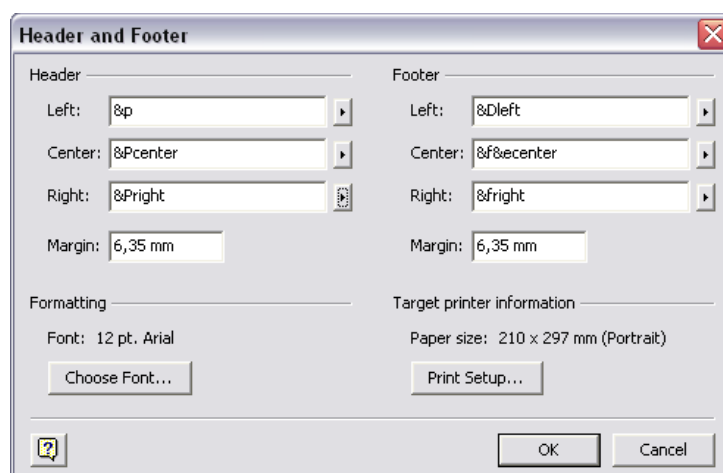
6.35.att. Dialoglogs Text.

Galvenes un kājenes (Header un Footer)

Galvene (***header***) – viena vai vairākas identificējošas rindiņas, ko drukā lappuses augšā. Galvenē parasti atrodas lappuses numurs, datums, autoru vārdi vai dokumenta virsraksts.

Kājene (***footer***) – informācija, kas teksta apstrādes vai lapušu izkārtojuma programmās atkārtojas un ko parasti drukā kāda izdevuma vai dokumenta lappuses beigās (*lappuses numurs, dokumenta vai izdevuma nosaukums u.c.*)

Lai pievienotu lapai galveni vai kājeni, izmanto ***View/Header and Footer...*** komandu.



6.36.att. Galvenes un kājenes dialoglogs.

Page number – lapas numurs

Page name – lapas nosaukums

Total printed pages – drukājamu lapu skaits

Curent time – tekošais laiks

Curent date – tekošais datums

File name – faila nosaukums

File extension – faila paplašinājums

File name and extensions – faila nosaukums ar paplašinājumu

6.6. Darbs ar slāņiem (Layer)

Slānis ir formu (*shapes*) kategorija. Slāņi (*layers*) tiek pielietoti, lai uz lapas organizētu saistītas formas. Piesaistot formas dažādiem slāņiem, var selektīvi pārskatīt, drukāt, rediģēt dažādas formu kategorijas.

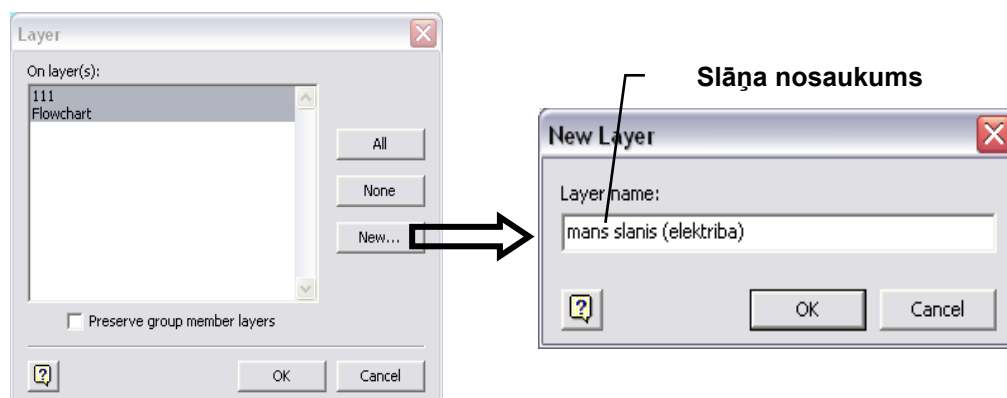
Forma var būt piesaistīta vairākiem slāņiem, vai arī nevienam. Katrai diagrammas lapai var būt dažāds slāņu kopums. Vairākas formas pēc noklusējuma ir jau piesaistītas konkrētiem slāņiem.

MS Visio diagrammas katra no lapām var saturēt vienu vai vairākus slāņus. Piemēram, zīmējot telpas izvietojumu, sienas, logi un durvis tiek piesaistīti vienam slānim, elektriskais tīkls – otram, bet mēbeles trešajam slānim.

Veidojot jaunu slāni, tas tiks pievienots tikai tekošai lapai, savukārt veidojot jaunu lapu, jānosaka tās slāņi, jo jauna lapa nesatur slāņus no iepriekšējām.

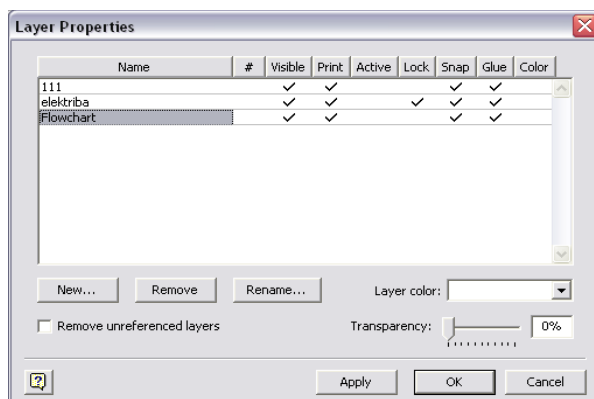
Kopējot formas no vienas lapas uz citu, slāņi tiks automātiski pievienoti sarakstam. Gadījumā, ja lapa jau satur slāņi ar to pašu nosaukumu, forma tiks pievienota eksistējošam slānim.

Lai izveidotu jaunu slāni un apskatītu kādiem slāņiem pieder iezīmētas formas, izmanto **Format/Layer**, un tad **New**



6.37.att. Jauna slāņa izveide.

Slāņu uzstādījumi atrodas **View/Layer Properties** logā.



6.38.att. Slāņa uzstādījumi.

Name – slāņa nosaukums

- pasaistītu formu skaits

Visible – slāņa paslēpšana

Print – vai slānis tiks drukāts

Active – ja atzīmēts, tad visas jaunas formas tiks automātiski pievienotas šim slānim

Lock – aizliegt rediģēt

Snap – norāda, vai ir iespējams pievienot formas

Glue – nosaka, vai ir iespējams savienot formas no cita slāņa, tekošajam

Color – nosaka slāņa formu krāsas

New – jauna slāņa pievienošana

Remove – nodzēst iezīmēto slāni

Rename – nomainīt iezīmētā slāņa nosaukumu

Remove Unreferenced Layers – nodzēst visus slāņus, kuri nesatur formas

Layer Color - slāņa krāsa

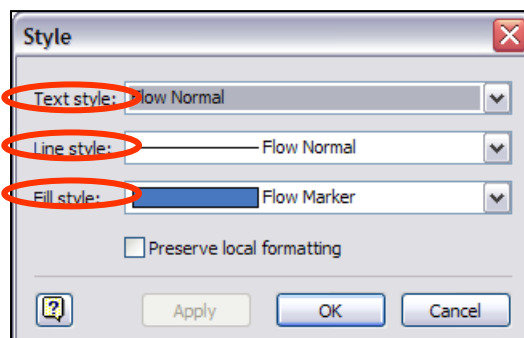
Transparency – slāņa krāsas caurspīdīgums (procentos)

6.7. Formas stili un krāsu shēmas

Stils (Style) – ir atribūtu kopums, kas glabājas veidnē (Template) vai diagrammas (Drawing) failā. Stils ir noformēšanas parametru kopums, kuru pielieto, lai nomainītu aizpildījuma krāsu, tekstu vai līniju krāsu.

MS Visio programmā stili sastāv no 3 sastāvdaļām:

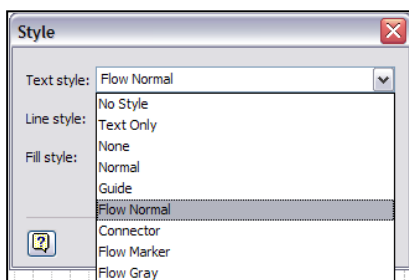
- Teksta formāts (**Text**)
- Līniju un bultu formāts (**Line**)
- Aizpildīšanas formāts (**Fill**)



6.39.att. MS Visio stili.

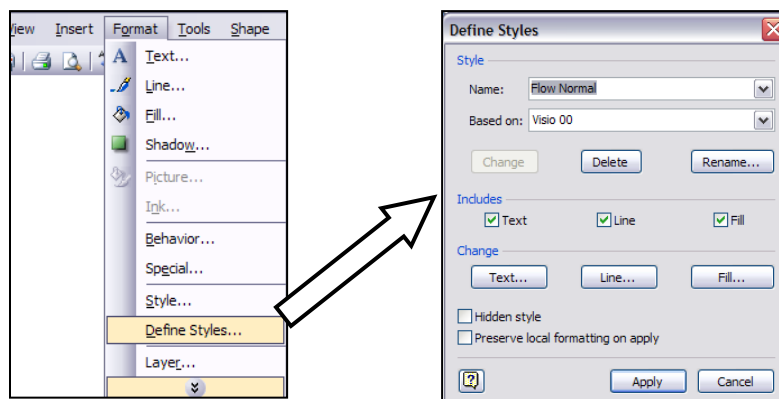
Katrs stils var saturēt visas sastāvdaļas, gan arī tikai atsevišķas. Katra forma, ko lietotājs ir izveidojis no jauna vai arī izvietojies no trafareta, jau atbilst stilam pēc noklusējuma.

Piemēram, pēc noklusējuma visas formas (Shape) un savienotāji (Connectors) atbilst **Flow Normal** stilam, teksts – **Text only** stilam.



6.40.att. Standarta stili.

Jebkura stila rediģēšana ietekmē visu formu izskatu, kuras atbilst tam stilam. Savukārt atsevišķas formas noformējuma parametru rediģēšana neietekmē citu formu izskatiem. Stila parametru rediģēšanai izmanto **Format/Define Styles...**

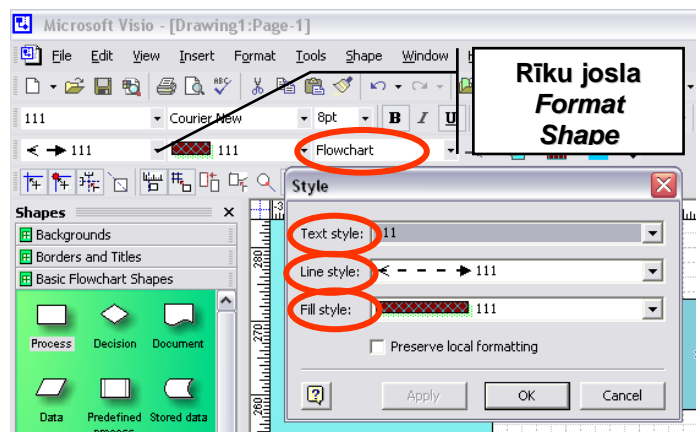


6.41.att. Stila parametru rediģēšana

MS Visio nodrošina jauktu stilu izmantošanu: atsevišķus, dažādus stilus līnijām, tekstam un aizpildījumam.

Stilu pielietošana.

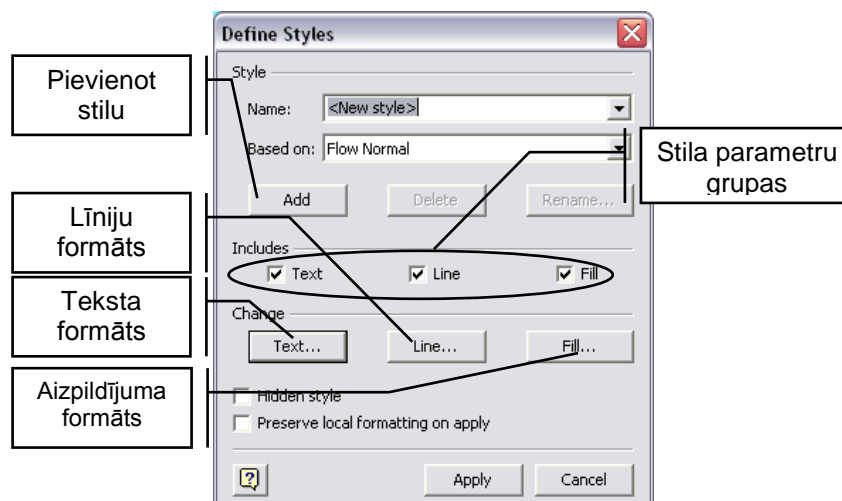
Stilu, kuram atbilst formas, var apskatīt aktivizējot komandu **Format/Style** vai izmantojot rīku joslu **Format Shape**, iepriekš iezīmējot formu.



6.42.att. Stilu pielietošana

Jauna stila izveide.

Lai izveidotu jaunu stilu izmanto **Format/Define Styles...**



6.43.att. Stila izveide.

Stila kopēšana.

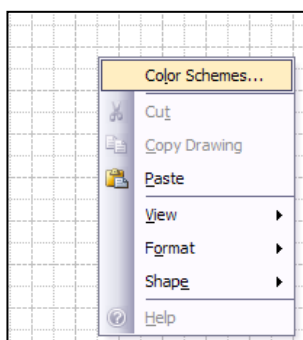
Visi MS Visio dokumenti tiek veidoti uz veidnes pamata, katrā no kurām ir jau definēts nepieciešamo stilu kopums. Pēc jauna stila izveides, tas tiek automātiski saglabāts tajā failā, kurā tika izveidots. Atšķirībā no pārējām MS Office programmām, MS Visio programmā nav tiešas stilu kopēšanas iespējas no viena dokumenta uz citu. Lai nokopētu stilu, izmanto šādu darbību secību:

- Jāiezmē forma, kura atbilst nepieciešamajam stilam un nokopē to - **Edit/Copy**;
- Dokumentā, kurā jāielīmē šis stils, ielīmēt šo formu – **Edit/Paste**;
- Ja šī forma nav vajadzīga, tad to var nodzēst, savukārt stils paliks pieejamo stilu sarakstā.

Krāsu shēmu lietojums.

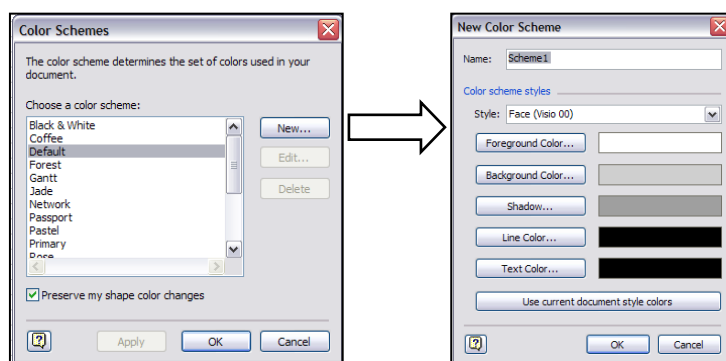
Krāsu shēma ir paredzēta visa dokumenta efektīvam un ātram noformējumam. Izmantojot šo iespēju var veidot vienota stila dokumentus. Krāsu shēmas ļauj manipulēt ar bultu krāsām, formu aizpildījumu krāsām utt.

Izmantojot krāsu shēmu, jebkura forma, neatkarīgi no tās izveidošanas procesa, tiks nokrāsota pēc shēmā noteiktiem parametriem. Pēc noklusējuma tiek pielietota **Default** krāsu shēma.



6.44.att. Krāsu shēmas izvēle.

Lai nomainītu krāsu shēmu izmanto labo klikšķi uz lapas vai arī izvēlni **Tools/Add-Ons/Visio Extras/Color Schemes**.



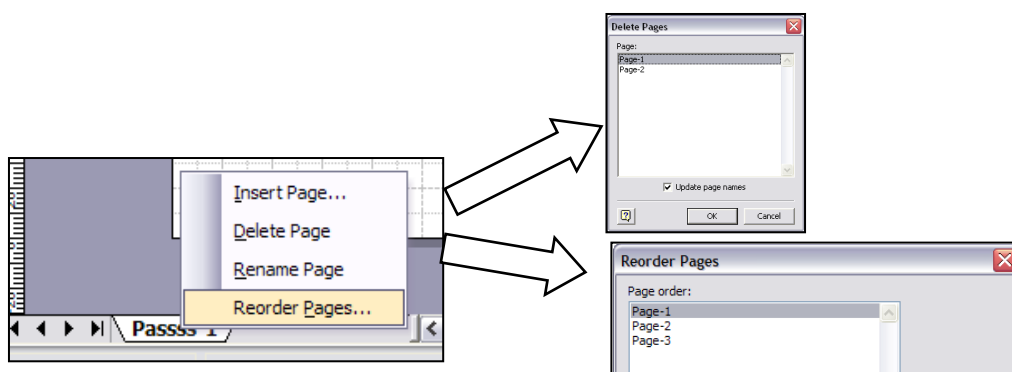
6.45.att. Krāsu shēmas izveide.

6.8. Darbs ar lapām, drukāšana.

Lai pievienotu jaunu lapu, izmanto **Inset/New Pages...**

Lai izdzēstu lapu, izmanto **Edit/Delete Pages...**

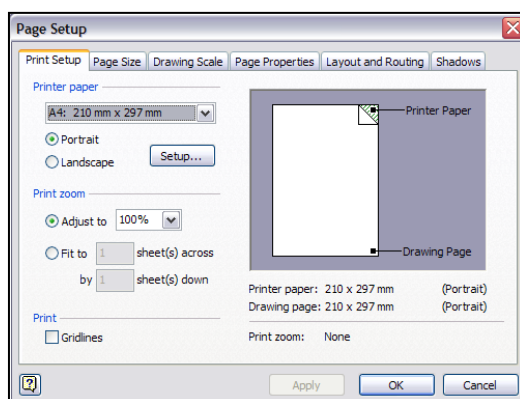
Var izmantot arī labo klikšķi uz lapas nosaukuma.



6.46.att. Jaunas lapas pievienošana un secības maiņa.

Dokumenta drukāšana ir viens no Visio diagrammu attēlošanas veidiem. Pats sagatavošanas process MS Visio programmā atšķiras no pārējām MS programmām, t.i., attēla lapas un drukājamās lapas izmēri var būt dažādi.

Vairākos gadījumos pirms drukāšanas jāsagatavo dokuments – tam ir paredzēts **File/Page Setup** dialoglogs:



6.47.att. Page Setup dialoglogs.

Diagrammas saspiešana.

Print Zoom – attēla saspiešanas koeficients, to izmanto lai samazinātu vai palielinātu attēlu. Pēc noklusējuma ir **None** vērtība. Atkarībā no pārējiem parametriem, saspiešanas koeficients darbojas **uz visām** lapām, tas nozīmē to, ka nevar izdrukāt vienu lapu ar saspiešanas koeficientu, bet pārējās bez tā.

Pastāv divi paņēmieni, kā var uzstādīt lapas attēla saspiešanu:

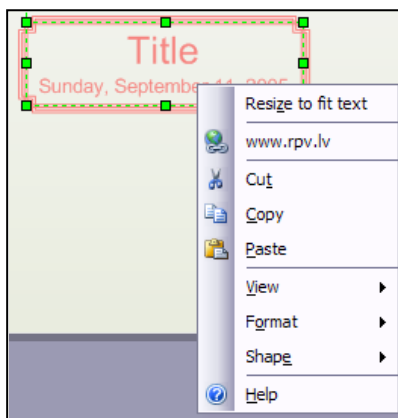
1. procentuālā saspiešana;
2. lapu skaita noteikšana (uz kurām ir jāattēlo diagramma). Visio automātiski sadala diagrammas lapu uz noteikto lapu skaitu

6.9. Diagrammu publicēšana Internetā, hipersaites. Sūtīšana pa e-pastu.

MS Visio nodrošina hipersaišu izmantošanu diagrammās, piesaistot visu veidu objektiem:

1. Jāiezīmē objekts;
2. **Insert/Hyperlinks** (**Ctrl+K**) un jāievada adrese vai ceļš.

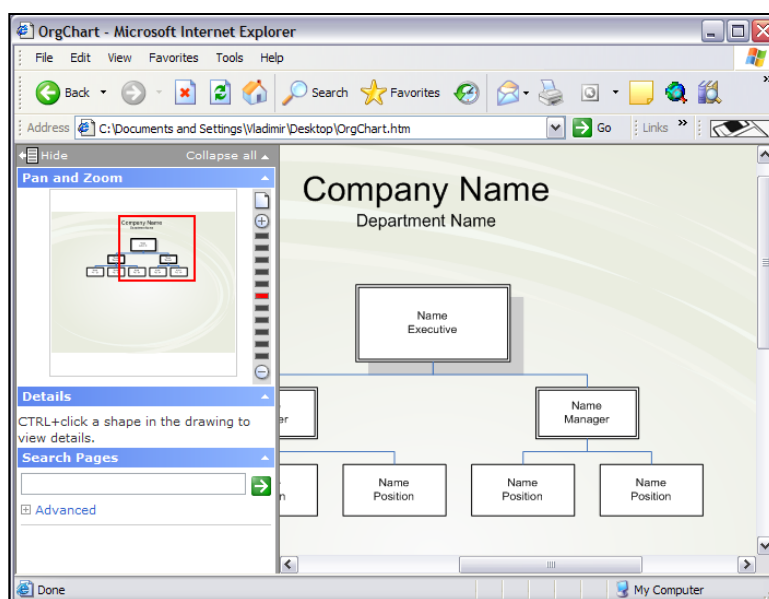
Hipesaiti aktivizē ar peles labo pogu:



6.48.att. Hipersaites aktivizēšana.

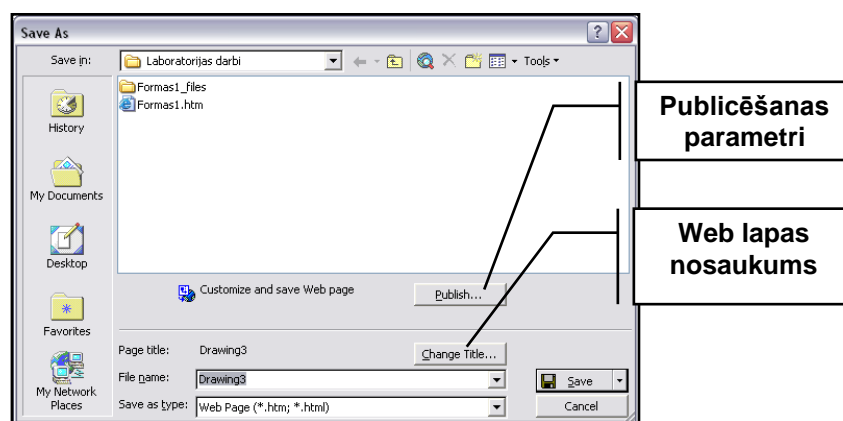
Diagrammas konvertēšana HTML formātā.

Jebkuru, izveidotu **MS Visio** programmā, attēlu var konvertēt HTML formātā. Tas nodrošina iespēju apskatīt jebkuru diagrammu kā mājas lapu.



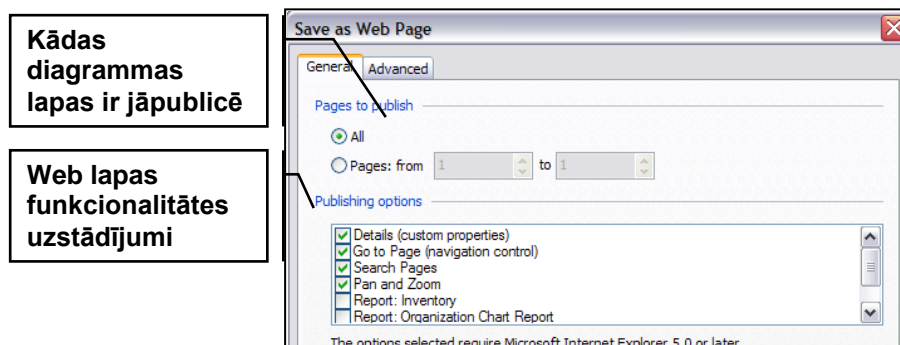
6.49.att. MS Visio diagramma HTML formātā.

Ja diagramma sastāv no vairākām lapām, tad katrai diagrammas lapai tiks izveidota atsevišķa mājas lapa, ar navigāciju pogām. Pēc konvertēšanas, izveidotas mājas lapas var apskatīt vairākās pārlūkprogrammās (MSIE, NETSCAPE, OPERA u.c.). Konvertēšanai izmanto **File/Save As WebPage...**

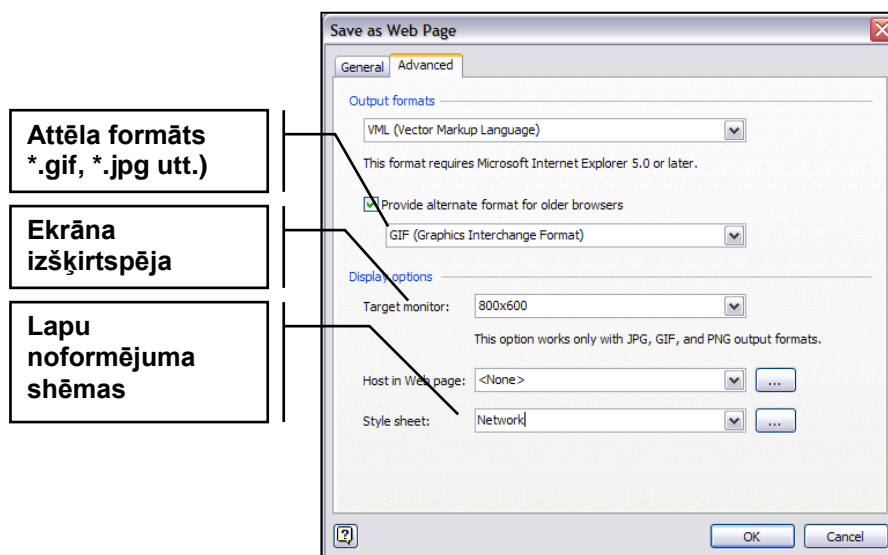


6.50.att. Diagrammas konvertēšana.

Publicēšanas parametru maiņa:

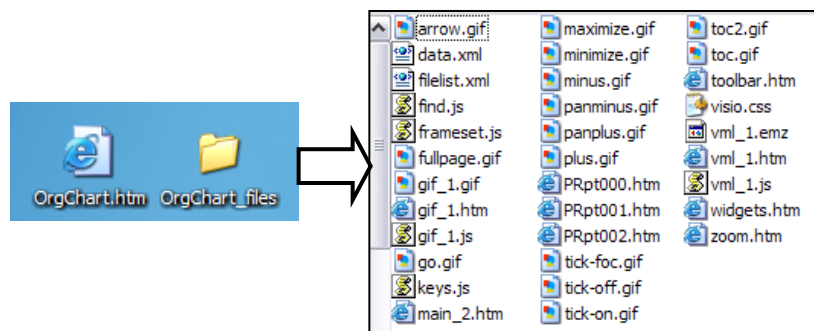


6.51.att. Publicēšanas parametri – lapa *General*.



6.52.att. Publicēšanas parametri – lapa *Advanced*.

Konvertēšanas rezultātā visas diagrammas tiek pārveidotas attēlos un glabājas atsevišķajā mapē:



6.53.att. Konvertēšanas rezultāts.

6.10. Jaunu formu un trafaretu izveide

Jaunu formu izveide.

MS Visio ir ļoti daudz standartu formu, kas atrodas trafaretos, tomēr dažreiz ir nepieciešams izveidot arī savas, specifiskas formas. Forma var būt izveidota:

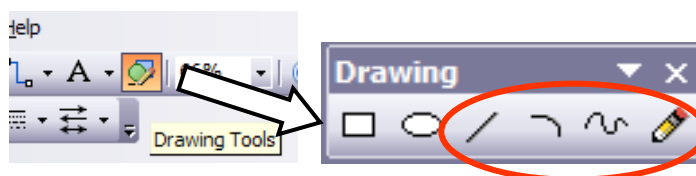
- “no nulles” – ar zīmēšanas līdzekļu palīdzību;
- ar standarta formu dažādām kombinācijām (piem. apvienošana);
- pārveidojot jau esošās formas, izmantojot rediģēšanas līdzekļus.

Formas, kas ir izveidotas ar zīmēšanas līdzekļu palīdzību sauc par **lietotāja formām**. MS Visio zīmēšanas līdzekļi:

1. **Line Tool** – taisnu līniju zīmēšanai
2. **Arc Tool** – loka zīmēšanai (elipses fragments)
3. **Freeform Tool** – splainu zīmēšanai (*spline*)
4. **Pencil Tool** – loka zīmēšanai (aploces fragments)

Zīmējot lietotāja formas, var izmantot **Rectangle Tool** un **Ellipse Tool** instrumentus, kas ir paredzēti taisnstūru un ovālu zīmēšanai.

Visi zīmēšanas instrumenti atrodas **Drawing** rīku joslā:



6.54.att. Zīmēšanas līdzekļi.

Līnija (Line) – vienkāršāka viendimensiju forma, var būt taisna līnija vai loks. Izmantojot līnijas un loku var veidot sarežģītākās formas: lauztas līnijas un *splainus* (*spline*).

Spline – ir lauza līnija, kas sastāv no loku kopas.

Lai uzzīmētu līniju ar slīpuma leņķi 45°, 90°, 135° utt., jāizmanto **Shift** taustiņš. Lauztas līnijas sastāv no vienkāršāku līniju vai loku (segmentu) kopas.

Vairāku neatkarīgu līniju savienošana veido vienu veselu formu, kuru nav iespējams sadalīt sastāvdaļās ar standarta instrumentiem.

Patvaļīgas formas var zīmēt, izmantojot **Freeform Tool** instrumentu. Taisnstūri un ovāli ir vairāku sarežģītu formu pamats. Lai vienkāršotu tās izveidi tiek piedāvāti **Rectangle** (taisnstūriem) un **Ellipse Tool** (elipsēm) instrumenti.

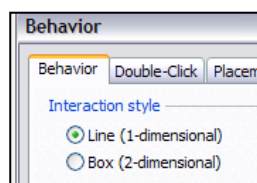
Formas „kvadrāts” un „aploce” arī var izveidot ar tiem pašiem līdzekļiem, izmantojot SHIFT pogu.

Slēgtas formas tiek izveidotas uz lauztas līnijas pamata, kad līnijas pirmais un pēdējais galapunkti sakrīt.

Segmentu sakrišanas pazīme – parādās aizpildījuma krāsa (**Fill**) (pēc noklusējuma gaiši pelēka).

Jebkuras izveidotas pēc šī principa lietojuma formas var pievienot trafaretam un vēlāk izmantot kā standarta formas.

Slēgtas formas ir divdimensiju objekti. Lai pārveidotu izmanto **Format/Behavior..., Interaction Style (line-1D vai box-2D)**.



6.55.att. Format/Behavior dialoglogs.

Darbības ar formām.

MS Visio ir trīs operācijas, kas nodrošina formu apvienošanu (**Shape/Operations/...**):

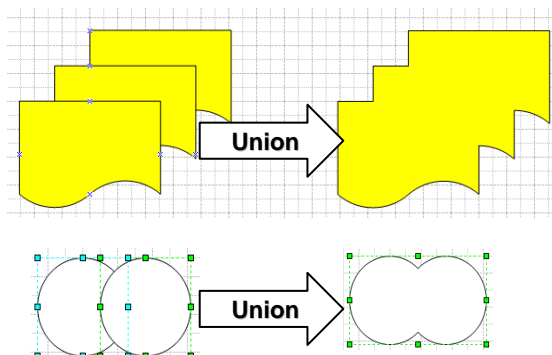
Union – apvienošana

Join – savienošana

Combine – kombinēšana

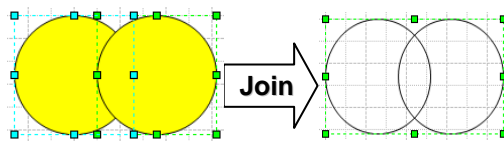
Apvienojot iezīmētās formas, tās paliek kā viens vesels objekts. Apvienošanas operācijas var pielietot arī vienai formai – šajā gadījumā kontūra līnija nemainīsies, bet visi savienošanas punkti tiks nodzēsti.

Shape/Operaton/Union - formu iekšējās daļas tiek nodzēstas, bet kontūra līnijas saglabājas.



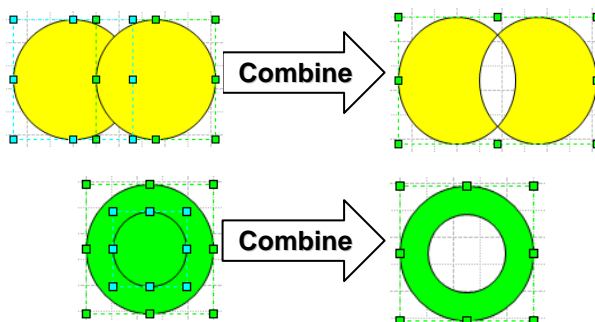
6.66.att. Apvienošanas piemēri **Union**.

Shape/Operaton/Join - centrālā (kopējā) formu daļa tiek nodzēsta, bet visas līnijas saglabājas.



6.67.att. Apvienošanas piemēri *Join*.

Shape/Operaton/Combine - jaunu formu izveide, apvienojot formu fragmentus, kas nekrustojas.



6.68.att. Apvienošanas piemēri *Combine*.

MS Visio sadalīšanas operācijas (**Shape/Operations/...**):

Subtract – atņemšana

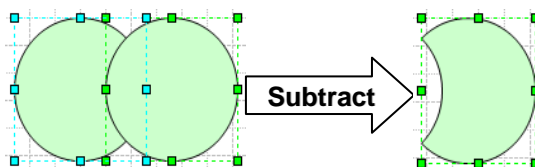
Fragment – sadalīšana

Trim – apgriešana

Intersect – šķelšana

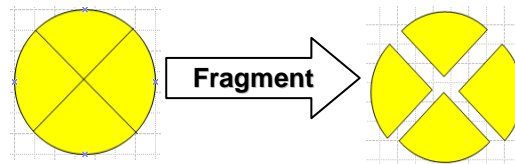
Sadalīšanas operāciju rezultātā paliek tikai formu segmenti vai vairāki fragmenti, kuru robežas nosaka kontūrlīnijas.

Shape/Operaton/Subtract - no formas kura tika pirmā iezīmēta, tiks atņemti visi formas (formu) fragmenti kas krustojas.



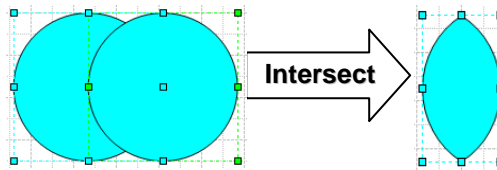
6.69.att. Sadalīšanas piemēri *Subtract*.

Shape/Operaton/Fragment - sadalīšana uz fragmentiem operācija ļauj izveidot vairākas formas, kas ir sākuma formas fragmenti.



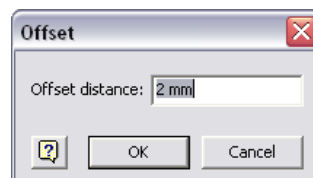
6.70.att. Sadalīšanas piemēri *Fragment*.

Shape/Operaton/Intersect - šķelšanas operācija ļauj izveidot formas, kas ir iezīmētu formu kopējā daļa.



6.71.att. Sadalīšanas piemēri *Intersect*.

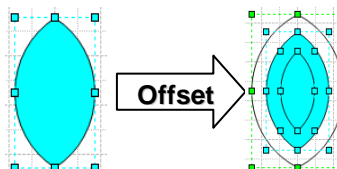
MS Visio paralēlo līniju pievienošana (formu vairošana) ***Offset***. ***Offset*** ļauj savairot izvēlētajā tipa formu skaitu. Pēc šīs operācijas pielietošanas parādās uzreiz divas kopijas – palielinātā un samazinātā. Samazināšana un palielināšana ir atkarīga no ievadītās vērtības – ***offset distance***.



6.72.att. *Offset distance*.

Ja ***Offset*** pielietot formu grupai, tad katrai formai atsevišķi tiks izveidotas kopijas.

Shape/Operaton/Offset - rezultātā tiks izveidotas divas formas, kurās ir līdzīgas kontūrlīnijas

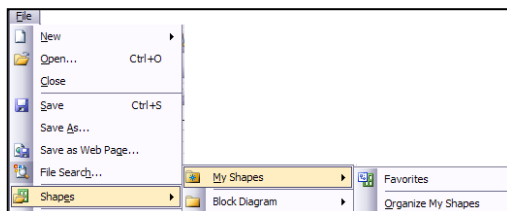


6.73.att. *Offset* operācijas piemērs.

Formu pievienošana trafaretam.

Visus zīmēšanas līdzekļus un operācijas var pielietot gan standarta formām, gan arī jaunizveidotām formām.

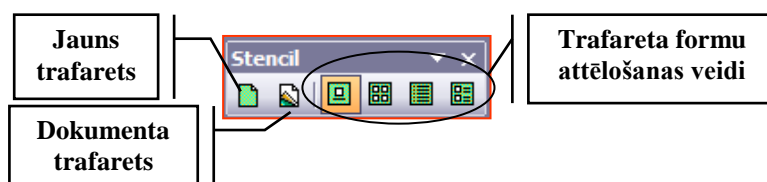
Jaunizveidotās formas var saglabāt speciālā – lietotāja trafaretā, aktivizējot **File/Shapes/My Shapes/Favorites**.



6.74.att. Lietotāja trafarets.

Katrs standarta trafarets ir tikai lasāms, tāpēc lietotāja formas var glabāt speciāli paredzētajā trafaretā – **Favorites**, vai arī izveidojot jaunu trafaretu.

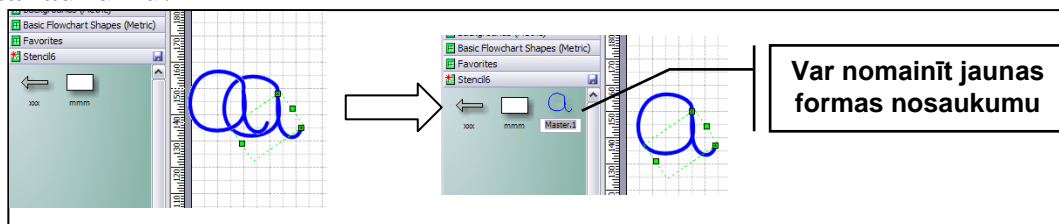
Lai izveidotu jaunu trafaretu izmanto **File/Shapes/New Stencil**.



6.75.att. Stencil rīku josla.

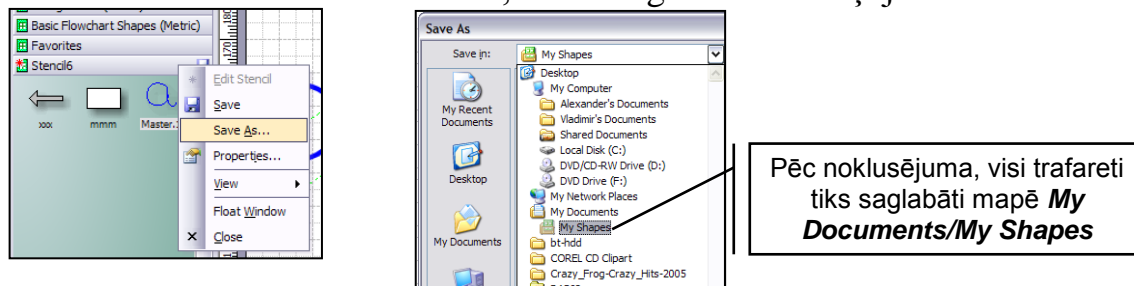
Jauna trafareta izveide. Stencil rīku josla.

Jaunizveidotajā trafaretā formas ievieto, pārnesot no diagrammas uz trafareta laukumu.



6.76.att. Formas pievienošana trafaretam.

Pēc formas ievietošanas trafaretā, to var saglabāt atsevišķajā failā.



6.77.att. Trafareta saglabāšana.

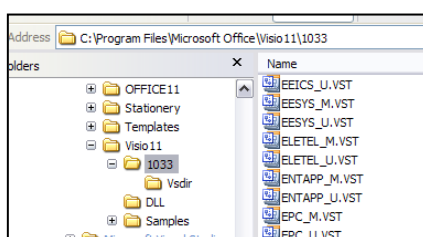
6.11. Darbs ar veidnēm, fona lapām.

Veidnes pielieto situācijās, kad jāveido daudz vienvēidīgas diagrammas. MS Visio veidne – ir fails (*.vst), kurā glabājas instrumentu kopas, trafareti, stili un uzstādījumi, kas nosaka attēla (diagrammas) īpatnības.

Lietotāja veidnes pielieto, kad, veidojot diagrammas, ir nepieciešami:

- vienvēidīgi, nestandarta izmēra vai mēroga uzstādījumi;
- rediģēšanas loga specifiskie parametru uzstādījumi;
- lietotāja trafaretu vai nestandarta trafaretu kopas pielietošana;
- lietotāja stilu, formu, teksta pielietošana.

Visas standarta veidnes glabājas mapē C:/Microsoft Office/Visio 11/1033/*.vst.



6.78.att. Standarta veidnes.

Veidnes veidošana.

Vienkāršākais variants, kā var izveidot jaunu veidni, t.i., izveidot jaunu dokumentu, noformēt un saglabāt to veidnes formātā (*.vst). (File/Save As...).

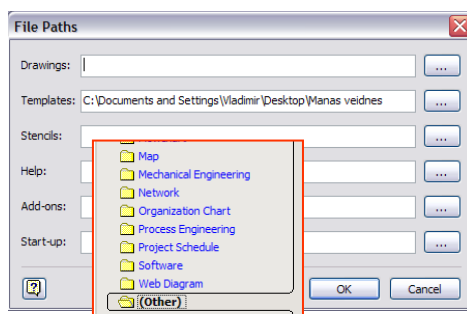
Saglabājot veidni, **Properties** dialoglogā var ievadīt nepieciešamās vērtības:

- Keywords – atslēgvārdi, pēc kuriem tiks meklēta šī veidne;
- Description – veidnes īss apraksts.

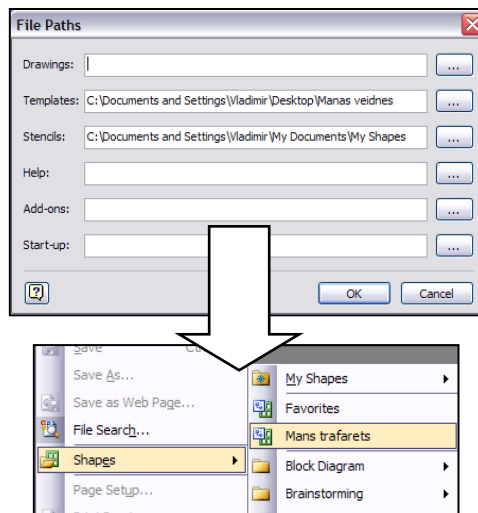
Veidnes un trafareta mapes reģistrēšana.

Veidnes reģistrēšana notiek pēc tās saglabāšanas noteiktā mapē un ceļa noteikšanas MS Visio programmā, aktivizējot **Tools/Options, Advanced, FilePaths** jāievada pilns ceļš.

Ja ir nepieciešams norādīt vairākas mapes, tad tās ceļi jāatdala ar “,” simbolu



6.79.att. Veidnes mapes reģistrēšana.



6.80.att. Trafareta mapes registrēšana.

Fona lapas.

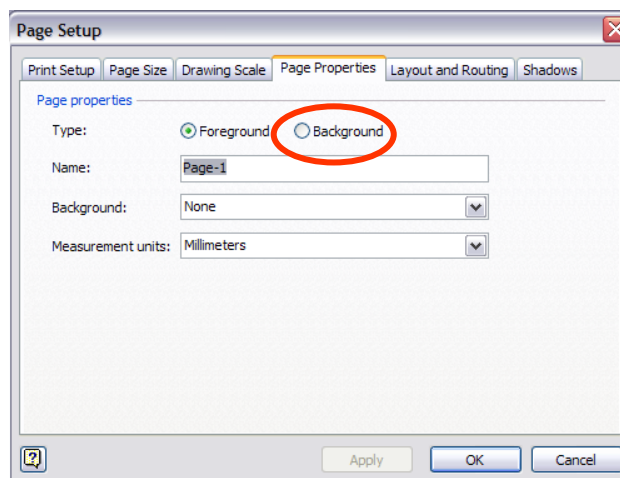
Uz lapas var izvietot ne tikai formas, bet arī fona attēlus. Par fona attēliem visbiežāk izmanto firmas logo, karogus utt. Katrs fona attēls tiek veidots uz atsevišķas lapas, neatkarīgi no pārējām lapām. Fonu lapu skaits failā nav ierobežots. Par fonu lapu var pārveidot jebkuru dokumenta lapu un otrādi. Šāda savstarpējā aizstājamība ļauj plaši izmantot fona attēlus. Atšķirībā no slāņiem, fona attēls ir viens vesels objekts.

Fona lapu var izveidot divos veidos:

1. no jauna, noformējot lapu kā fona attēlu;
2. izveidojot zīmējumu un pēc tam pārveidojot to par fona lapu.

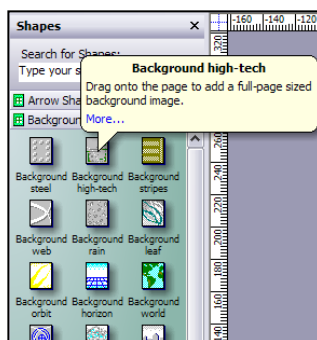
Lai izveidotu fona lapu no jauna:

1. **Insert/New Page**
2. **File/Page Setup/Page Properties** logā izvēlas **Background**



6.81.att. Fona lapas izveide no jauna.

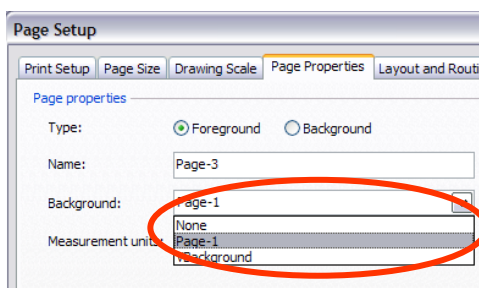
Lai izveidotu fona attēlu, aktivizē **File/Shapes/Visio Extras/Background**, <vajadzīgais fona attēls>



6.82.att. Fona izvēle.

Fona attēla pievienošana lapai.

Lai pievienotu lapai iepriekš izveidotu fonu, aktivizē **File/Page Setup/Page Properties**, **Background** un izvēlas vajadzīgo fona lapu.

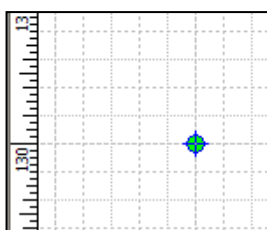


6.83.att. Fona attēla pievienošana lapai.

Veicot izmaiņas fona attēlā, tās automātiski tiek veiktas visās lapās, kuras izmanto šo fona attēlu.

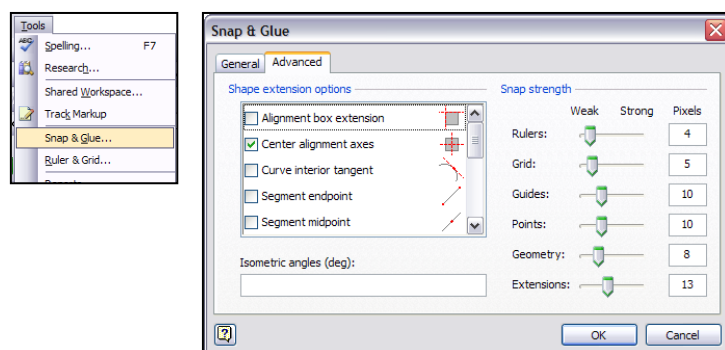
6.12. Piesaistīšanas punktu un līniju pielietošana. Perspektīvas zīmēšana.

Piesaistīšanas punkti ir paredzēti lai noteiktu tīkla (Grid) attiecīgus mezglus, piemēram, kad notiek formu savienošana. Pēc to noteikšanas, mezglam ir sekojoša īpašība – visi formas savienošanas punkti vai virsotnes, kas atrodas piesaistīšanas punkta aktīvajā zonā, automātiski tiks pārvietoti šajā mezglā.



6.84.Att. Piesaistīšanas punkts.

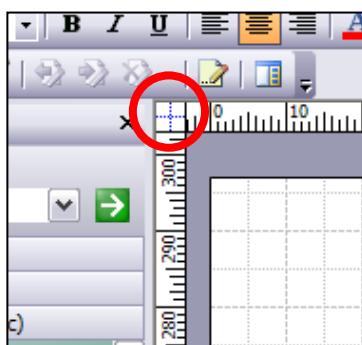
Automātiskas piesaistīšanas visi parametri atrodas ***Snap & Glue*** dialoglōgā, to aktivizē ar ***Tools/Snap & Glue***.



6.85.att. *Snap & Glue* dialoglōgs.

Piesaistīšanas punkta izveide.

Piesaistīšanas punkts tiek pievienots diagrammai, izmantojot Guide Point instrumentu, kas atrodas rediģēšanas loga augšējā kreisajā stūrī.

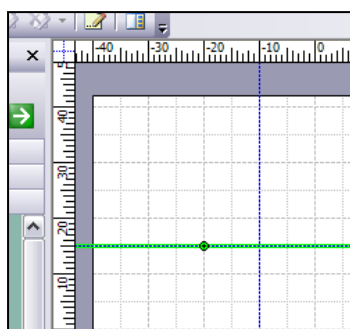


6.86.att. *Guide Point* instruments.

Piesaistīšanas punktu var paslēpt ar ***View/Guides***. Drukājot diagrammu, piesaistīšanas punkts netiek drukāts.

Piesaistīšanas līnijas.

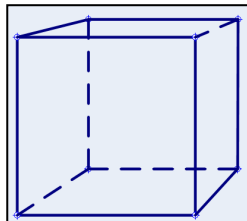
Piesaistīšanas līniju īpašības ir līdzīgas piesaistīšanas punktiem. Piesaistīšanas līnijas var būt vertikālās un horizontālās.



6.87.att. Piesaistīšanas līnijas.

Uzdevums.

Piemērā apskata piesaistīšanas punktu un līniju pielietošanu, zīmējot perspektīvas. Visio programmā nav tieša veida kā uzzīmēt perspektīvu, izņemot formu trafaretus ar ēnām. Par pamata formu izmanto kvadrātu, kā rezultātā iegūst šo formu:



6.88.att. Perspektīvas zīmēšana.

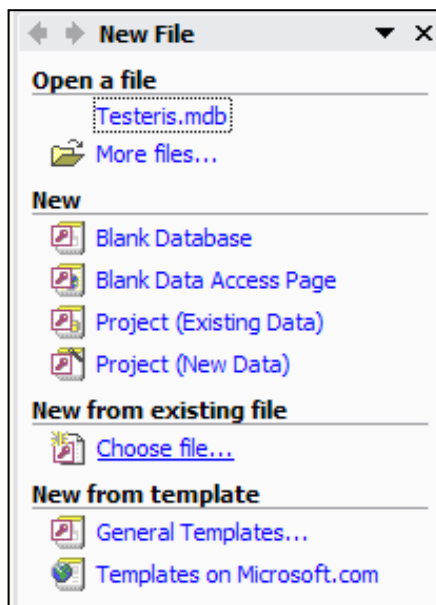
Darbību secība:

1. Izveidot jaunu diagrammu un pievienot horizontālu piesaistīšanas līniju.
2. Izmantojot piesaistīšanas horizontālo līniju, izveidot horizonta līniju, uz kuras izvieto perspektīvas saieta punktu. Līnijai jābūt novilkta caur punktu ar $Y=150$.
3. Pārveidot horizonta līniju par 2D objektu **Format/Behavior** logs.
4. Perspektīvas saieta punktā ($X=150$, $Y=150$) izvietot piesaistīšanas punktu.
5. Uzzīmēt uz lapas kvadrātu 50 mm x 50 mm.
6. Izvietot kvadrātu aizmugurē (Send to Back). (Nepieciešams, lai redzētu piesaistīšanas līnijas).
7. Pievienot kvadrāta virsotnēm piesaistīšanas punktus.
8. Pārvietot horizonta līnijas rotācijas centru piesaistīšanas punktā.
9. Pagriezt līniju tā, lai tā būtu novilkta caur kvadrāta kreiso augšējo virsotni.
10. Izveidot horizontālu piesaistīšanas līniju, kura ir novilkta caur $Y=125$.
11. Izveidot vertikālu piesaistīšanas līniju, caur horizontālas un slīpas līnijas krustpunktu.
12. Izvietot līniju krustpunktā piesaistīšanas punktu.
13. Pagriezt slīpu līniju tā, lai tā būtu novilkta caur augšējo labo virsotni.
14. Līdzīgas darbības izdarīt ar apakšējām virsotnēm.
15. Pēc figūras virsotnes izveides pabeigšanas, nodzēst visas liekās piesaistīšanas līnijas.
16. Ar *Line Tool* instrumentu palīdzību savienot virsotnes un atbilstoši noformēt.
17. Salīdzināt ar pareizo variantu (sk. 6.88. att.)

7. Lietojumprogrammas MS Access vide

Ja ir veikta standarta instalācija, tad MS Access ir atrodams *Start/Programs* izvēles kartē.

MS Access tiek aktivizēts. Vispirms uz ekrāna parādās MS Access titullapa un pēc īsa brīža – MS Access uzsākšanas logs. Atšķirībā no, piemēram, MS Word vai MS Excel, uzreiz nav iespējams sākt datu ievadīšanu un to šķirošanu. Vispirms ir jāatver jau iepriekš izveidotā datu bāze vai jāsāk jaunas datu bāzes veidošana. Tāpēc loga labajā pusē parādās uzsākšanas izvēles logs:

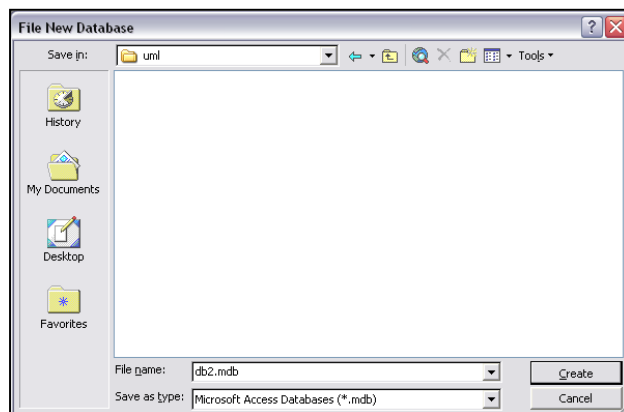


7.1.att. MS Access uzsākšanas izvēles logs.

Ja tiek uzsākta datu bāzes veidošana, jāizvēlas viena no izvēlēm:

- 1. gadījumā – *New, Blank Database* – MS Access izveidos tukšu datu bāzi, kur paša veidotāja ziņā būs visu objektu tālāka veidošana;
- 2. gadījumā – *New from Template, General Templates...* – MS Access piedāvās izvēlēties vienu no vairākām datu bāzēm. Starp tām ir gan vienkāršas vienas tabulas datu bāzes, kā, piemēram, Address Book, gan diezgan sarežģītas datu bāzes, piemēram, Service Call Management un Inventory Control, kur datu bāzes tabulu un formu skaits pārsniedz desmitu.

Kad ir izvēlēta kāda no parauga datu bāzēm vai arī tukšā datu bāze (piemēram, Blank Database), un ar klikšķi uz OK pogas apstiprināta izvēle, tiks atvērta File New Database logs:



7.2.att. File New Database logs.

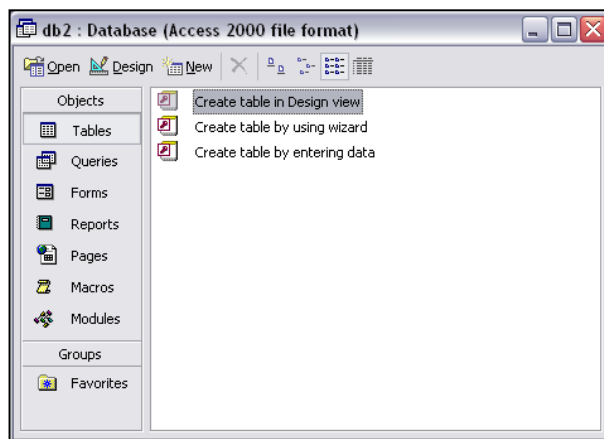
Jāpiebilst, ka uzsākšanas izvēles logs tiek atvērts, tikai uzsākot darbu ar MS Access. Ja MS Access jau ir aktīvs un darbs ar vienu datu bāzi tiek pabeigts, datu bāzi aizverot, tad uzsākšanas izvēles logs vairs neparādās.

Šajā gadījumā, lai izveidotu jaunu datu bāzi, jāizpilda File/New Database vai jāklikšķina uz attiecīgās ikonas. Arī tagad atveras File New Database logs.

Darbības šajā logā ir tādas pašas kā citos Windows failu atvēršanas vai saglabāšanas logos:

- logā *Save in* jānorāda jaunā faila atrašanās mape;
- logā *File Name* jāieraksta datu bāzes faila nosaukums. MS Access pats piedāvās nosaukumus, kas sākas ar “db” – vispirms “db1”, nākošajā reizē “db2” utt. Faila paplašinājumu .mdb (MS Access datu bāzes failiem ir paplašinājums *.mdb) var arī nenorādīt, MS Access to pievienos automātiski;
- jānoklikšķina poga OK vai tastatūras taustiņš Enter.

Tad uz ekrāna ir parādījies jauns logs – datu bāzes logs un tas ir galvenais objekts darbības procesā. Tas ir kā komandtiltiņš tāpat kā uz kuģa – vissvarīgākā vieta, no tā vada datu bāzi: rada jaunus objektus, dod tiem pavēles un pārbauda, lai viss norisināties saskaņoti un precīzi.



7.3.att. Datu bāzes logs.

Logam ir sešas nodaļas, kurās izvietojas Access sešas datu bāzes objektu grupas: tabulas (table), vaicājumi jeb vaicājumi (query), formas (form), pārskati (reports), makro (macro) un programmu kodi (modules):

Tabulas (table) – datu sakopojums par kādu noteiktu tematu, piemēram, par uzņēmuma klientiem, darbiniekiem, saražotiem produktiem, iepirktiem materiāliem, par izrakstītiem rēķiniem utt. Vizuāli tabulas ir līdzīgas MS Excel tabulām.

Tabulas ir datu bāzes pamats, praktiski visa informācija tiek glabāta tabulās. Pārējie datu bāzes elementi tiek izmantoti, lai ātri un efektīvi piekļūtu šiem datiem, veiktu to atlasī, parādītu datus tādā vai citādā veidā.

Starp tabulām var izveidot pastāvīgas saites, kas uztur MS Access un nodrošina datu integritāti.

Vaicājumi jeb vaicājumi (query) – līdzeklis datu atlasei no vienas vai vairākām tabulām, datu apskatīšanai dažādos griezumos, to kombinēšanai un summēšanai.

Formas (form) – atvieglo lietotāju darbu ar sistēmu, datu ievadi un apskati. Formā var nodrošināt visu vajadzīgo (un tikai vajadzīgo!) datu parādīšanos uz ekrāna, dažos laukos atļaut datu ievadi, dažos tikai apskati, nodrošināties pret kļūdainu datu ievadi. Pieredzējis lietotājs var izveidot formu, kuru pēc tam izmanto citi lietotāji, un šajā formā ir iespējams veikt tikai noteiktas darbības.

Pārskati (report) – līdzeklis ievadīto datu attēlošanai izdrukās veidā. Plašas iespējas izmantot skaitļu grafisku attēlojumu, zīmējumus, dažādus fontus.

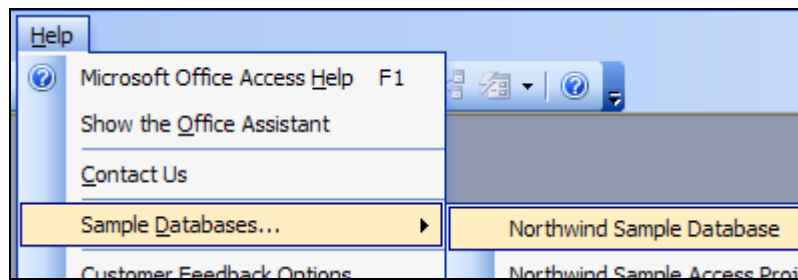
Makro (macro) – komandas formu atvēršanai, atskaišu izdrukai, kursora novietošanai formā, operācijām ar datiem u. c., tās var apvienot makrogrupās, veidojot noslēgtu darbību kopumu.

Programmu kodi (module) – programmēšanas vide Visual Basic valodā.

Lai pārtrauktu darbu ar datu bāzi un MS Access, jādod komanda File/Exit, tad MS Access tiks aizvērts, bet datu bāze tiks saglabāta automātiski. Ja nepieciešams aizvērt datu bāzi un sākt darbu ar jaunu, tad, neizejot no MS Access, jādod komanda File/Close – tiks aizvērta datu bāze un uz ekrāna būs MS Accesss uzsākšanas logs.

Uzsākt darbu ar jaunu datu bāzi var arī datu bāzes logā, aktivizējot File/Open Database – jau atvērtā datu bāze tiks aizvērta, bet tiks atvērta Open Database logā norādītā datu bāze. Atšķirībā no MS Word vai MS Excel, kuri vienlaikus var strādāt ar vairākiem dokumentiem vai darba burtnīcām, MS Access atver tikai vienu datu bāzi. Ja tomēr ir nepieciešams pārmaiņus strādāt ar vairākām datu bāzēm (vairākiem .mdb failiem), MS Access var atvērt vairākas reizes, cik vien to atļauj datora resursi.

Piemēros izmantosim parauga datu bāzi Northwind.mdb, kas pievienota MS Access instalācijai un ir atrodama mapē *Program Files/Microsoft Office/Office/Samples*.



7.4. att. Standarta datu bāze Northwind.mdb.

Lai to atvērtu, jāizpilda *Help/Samples Databases*.

8. MS Access datu bāzes tabulas atribūti

Tabulas ir datu bāzes pamats. Praktiski visa informācija tiek glabāta tabulās. Pārējos datu bāzes elementus izmanto, lai ātri un efektīvi piekļūtu šiem datiem, veiktu to atlasī, parādītu datus tādā vai citādā veidā. Starp tabulām var izveidot pastāvīgas saites, kuras uztur Access un nodrošina datu integritāti.

Katra tabula sastāv no ierakstiem un laukiem. Lauku skaitu ierakstos un to tipu nosaka tabulas konstruēšanas procesā. Lauku lielumu nosaka lietotājs. Jāseko, lai lauka lielums nebūtu pārāk liels, jo tādā gadījumā ir nevajadzīgi izmantota atmiņa.

Katras tabulas laukam ir savs nosaukums un atbilstošais datu tips. Lauka nosaukums (Field Name) – tas ir lietotāja dots lauka nosaukums, piemēram, ieraksta numurs, vārds, uzvārds, dzimšanas datums, amats utt. Nosaukums var būt 64 simbolus garš, tas nedrīkst sākties ar tukšumu “ ”, nedrīkst saturēt “.”, “!”, “””, “[“, “[”.

Tabulas lauka datu tips (Data Type) ir lauka vērtību tips. Projektējot datu bāzi, jānorāda katram laukam atbilstošais datu tips:

- **Text** – teksta lauks, lauka tips pēc noklusēšanas, garums līdz 255 zīmēm.
- **Memo** – teksta lauks apjomīgai informācijai, garums līdz 64 000 zīmēm. Datu bāzes ātrākai darbībai ieteicams, kur tas ir iespējams, lietot Text tipu teksta veida informācijai.
- **Number** – skaitļu lauks, kurš satur skaitlisku vērtību, kura diapazonu nosaka parametrs Field Size.
- **Date/Time** – datuma un laika lauks.
- **Currency** – naudas formāta lauks, precizitāte – 15 skaitļi pa kreisi no punkta un 4 skaitļi pa labi no punkta.
- **Counter** – lauks, kurā katru jaunu ierakstu Access pats numurē un lietotājs izmainīt nevar.
- **Yes/No** – loģiskais lauks, tajā iespējama viena vērtība no divām; neaizpildīts lauks nav atļauts.
- **OLE Object** – objekti, kas izveidoti ar citām programmām izmantojot OLE protokolu.
- **Hyperlink** – šī tipa laukā glabājas hipersaites, kuras nodrošina ceļu pie cietā diska vai adresi internet tīklā.
- **LookupWizard** – ļauj parādīt lauku nozīmi parasta vai kombinēta saraksta izskatā.

9. Datu bāzes tabulas izveide

9.1. Tabulas izveide, lauku īpašības

Kad ir izveidota jauna tukša datu bāze, parādās datu bāzes logs (Database), kurā aktīva ir lapa *Tables*. Sākumā tā ir tukša.

Veidojot jaunas tabulas, to nosaukumi parādīsies šajā lapā. Datu bāzes loga augšējā pusē ir trīs pogas:

Open – atvērt esošu tabulu;

Design – veikt izmaiņas esošā tabulā, pārejot konstruktorā;

New – izveidot jaunu tabulu.

Lai izveidotu jaunu tabulu, datu bāzes loga *Tables* lapā jānoklikšķina *New* poga. Atveras logs *New Tables*:



9.1.att. *New Tables* logs.

Lietotājam jāizvēlas, kādā veidā veidot jaunu tabulu. Piedāvā jaunu tabulu veidošanas 5 veidus:

Datasheet View - uzreiz var sākt datu ievadi datulapā, kas ļoti līdzīga MS Excel darba lapai. Tabulas definīciju (lauku tipus un pārējos parametrus) Access nosaka pats pēc ievadītajiem datiem. Tas ir viegls paņēmieni, kā izveidot jaunu tabulu, bet bieži vien Access kļūdās parametru izvēlē un tāpat pēc tam jāveic korekcijas tabulas definīcijā. Tāpēc labāk iepriekš izplānot visus vajadzīgos laukus un pašiem nodefinēt tabulu.

Table1 : Table				
	Field1	Field2	Field3	Field4
▶				

9.2.att. *Datasheet* skats.

Šo iespēju, kā veidot jaunu tabulu, ir lietderīgi izmēģināt pirmo reizi veidojot tabulas, lai redzētu kāda izskatās izveidotā tabula, bet vēlāk, veidojot jaunas tabulas, labāk izvēlēties *Design View* (pašam nodefinēt tabulas definīciju).

Table Wizard - Access atver *Table Wizardu* (“palīgu”), kas ir palīgs tabulu veidošanā. Lietotājs var izvēlēties no parauga tabulu saraksta vispiemērotāko tabulu, pēc tam izvēlēties vajadzīgos laukus - un tas ir viss, kas nepieciešams, lai izveidotu jaunu tabulu ar Wizarda palīdzību. Tāpat kā iepriekšējā gadījumā, arī izveidojot tabulu ar Wizarda palīdzību, lietotājam bieži vien jāveic ļoti daudzas korekcijas tabulas definīcijā, jo standarta lauki, ko piedāvā Access neatbilst lietotāja vēlmēm. Tāpat kā iepriekšējais variants, arī šis noder tikai sākumā, kad apgūst tabulu veidošanu, bet vēlāk daudz efektīvāk un ātrāk ir pašam izveidot tabulas definīciju.

Import Table - jaunas tabulas izveidošana, ieimportējot to no citas datu bāzes (to var darīt gan no MS Access datu bāzēm, gan no citurienes (FoxPro, MS Excel, SQL tabulas, dBase u.c.). Izvēloties šo iespēju, lietotājam jānorāda, no kurienes viņš grib ieimportēt tabulu un kuru tieši tabulu. Lietotājs var izvēlēties vai tiks importēta tabula ar datiem, vai tikai tabulas definīcija.

Link Table - jaunas tabulas izveidošana, piesaistot (link) jau esošu tabulu no citas datu bāzes (tā var būt gan Access, gan cita datu bāze). Tāpat kā importējot tabulu, lietotājam jānorāda no kurienes un kuru tieši tabulu viņš grib pievienot. Tādā veidā lietotājs var pieslēgties citu datu bāzu datiem, izmantot tos un arī mainīt tos tieši tajās datu bāzēs (ja ir atļauja to darīt). Pievienotās tabulas tabulu logā ir atzīmētas ar bultiņu – pievienotās tabulas simbolu. Pievienotai tabulai nav iespējams mainīt struktūru (izņemot atsevišķas īpašības). Pievienotās tabulas var atvienot, izdzēšot tabulu no loga (tabula tiek atvienota, nevis izdzēsta fiziski).

Design View - jaunas tabulas izveidošana, pašam izveidojot tabulas definīciju. Tieši šo metodi izmantos veidojot mums vajadzīgās tabulas. Šajā gadījumā lietotājam pašam jānodefinē tabulas īpašības, lauki un to īpašības. Tātad, pirms sākt veidot tabulu, būtu jau iepriekš jāizdomā, kādu tabulu grib veidot.

10. Tabulas lauku definēšana

10.1. Lauku definēšana

Uzklīkšķinot *New Table* un nospiežot *OK* pogu, atveras tabulas logs veidošanas (konstruktora) skatā (design view). Sākotnēji loga augšējā daļa, kas paredzēta lauku sarakstam, ir tukša. Aizpilda šādi:

10.1.att. *Design View* logs.

Pirmā kolonna *Field Name* – lauka vārds – ir lietotāja dots lauka nosaukums. Kā visi *MS Access* nosaukumi tas var būt 64 zīmes garš un saturēt dažādu zīmju kombināciju. Jāievēro, ka nosaukums nedrīkst sākties ar tukšumu “ ”, un nedrīkst saturēt punktu “.”, izsaukuma zīmi “!”, apostrofu “'” un kvadrātiekavas “[”, “]”.

Otrajā kolonnā norāda lauka datu tipu (sk. 8.nodaļā).

Pēdējā kolonna – *Description* – paredzēta paša lietotāja komentāriem, un *MS Access* darbībā tai nav nozīmes.

10.2. Lauku secības maiņa

Jau nodefinēto lauku secību var mainīt šādi:

Iezīmēt lauku, novietojot kursoru loga kreisajā malā uz lauku izvēles taisnstūriem tā, lai kursors kļūst par horizontālu tumšu bultiņu, tad noklikšķināt peli. Lauka rinda tiek iekrāsota tumša. Lauku pārvilkt vertikāli uz vajadzīgo pozīciju, lauka novietošanās vietu parāda tumša vertikāla līnija, un atlaist peli.

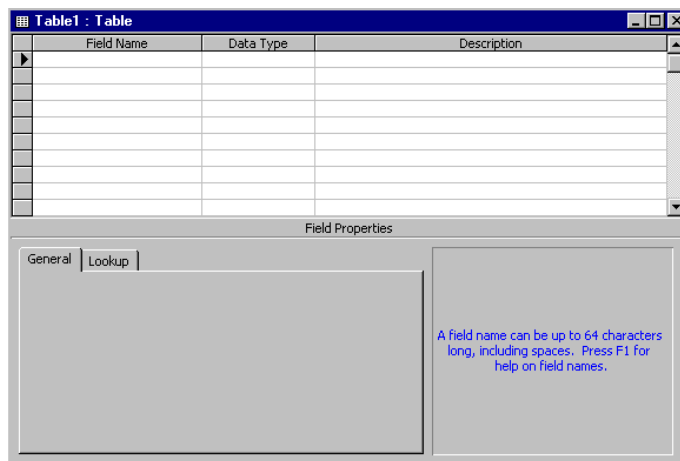
Vienlaicīgi var pārvietot vairākus blakus esošus laukus, tos iezīmējot un pēc tam pārvelkot. Lai iezīmētu vairākus blakus esošus laukus:

- peles taustiņu, klikšķinot uz viena lauka izvēles taisnstūra, pietur nospiestu un pārvelk pāri attiecīgajiem blakus esošajiem laukiem;
- kad viens lauks iekrāsots, pārvieto kursoru uz blakus esošā lauka izvēles taisnstūri tā, lai kursors kļūst par horizontālu tumšu bultiņu. Tad, turot nospiestu *Shift* taustiņu, noklikšķina vēlreiz. Visi blakus esošie lauki tiek tumši iekrāsoti.

Nevajadzīgos laukus var izdzēst (pirms tam noteikti jāpārliedzinās, vai šajos laukos netiek glabāti vajadzīgi dati!), iezīmējot lauku un nospiežot taustiņu *Delete*.

11. Lauku īpašības *Field Properties*

Logā *New Table* izvēla izvēlni *Design View* (paši veidosim tabulu). Atveras tabulas logs veidošanas skatā. Šis logs paredzēts visu lauku un to īpašību nodefinēšanai.



11.1.att. *Design View* logs.

Tabulas logs veidošanas skatā sastāv no divām daļām – augšējā daļā ir visu lauku saraksts, bet apakšējā daļā ir lauka īpašību logs.

Loga apakšējā daļā tiek noteiktas lauku īpašības. Jāievēro, ka šajā daļā parādās īpašības tam laukam, uz kura ir apstājies kursora veidošanas loga augšējā daļā. Atkarībā no lauka tipa īpašību saraksts var mainīties. Iespējamās šādas īpašības:

Field Size – lauka izmērs. Text laukiem pēc noklusēšanas no 0 līdz 255. Number lauku iespējamās vērtības:

Byte – naturāliem skaitļiem no 0 līdz 255.

Integer – veseliem skaitļiem no –32 768 līdz 32 767.

Long Integer – veseliem skaitļiem no –2 147 483 648 līdz 2 147 483 647.

Single – skaitļiem ar sešu zīmju precizitāti no –3.402823E38 līdz 3.402823E38.

Double – skaitļiem ar 10 zīmju precizitāti no –1.79769313486232E308 līdz 1.7976931486232E308.

Replication ID – laukam, kurā ir replikācijas indekss. Aizņem 16 baitus. Replikācija ir datu sinhronizēšana, ja vienu un to pašu datu bāzi lieto vairākās vietās neatkarīgi, t. i., strādā ar datu bāzes kopijām, regulāri apmainoties ar datiem.

Svarīgi: Parasti vajadzētu izmantot mazāko iespējamo lauku izmēru, jo mazāka izmēra dati tiek apstrādāti ātrāk un aizņem mazāk atmiņas.

Ja rodas nepieciešamība samazināt lauka izmēru, jāuzmanās no iespējamās datu zaudēšanas. Ja teksta laukam izmēru samazina no 50 uz 20, zīmes, sākot ar 21. (ja tādas ir), tiks zaudētas.

Ieteicams noteikt Currency tipu laukiem, kuriem noteikti būs ne vairāk kā 4 zīmes aiz komata. Aprēķinus ar Currency laukiem veic ar fiksēta decimālkomata metodi tie ir ātrāki par Single un Double laukiem, kuriem lieto peldošā decimālkomata metodi.

Format – lauka formāts. Text un Memo tipa laukiem ar speciālu simbolu palīdzību var norādīt, vai visus burtus uzrādīt kā lielos, vai mazos burtus un gadījumā, ja lauks ir tukšs, parādīt kādu noteiktu tekstu.

Decimal Places – decimālzīmju skaits aiz komata. Īpašība iespējama Number un Currency tipa laukiem.

Input Mask – ievada maska jeb ievada šablons.

Caption – uzraksts, kas parādīsies lauka nosaukuma vietā datulapās, formās un atskaitēs.

Default Value – lauka noklusētā vērtība.

Validation Rule – likumības nosacījums, kam jābūt spēkā laukā ievadītajai vērtībai.

Validation Text – teksts, kas parādīsies uz ekrāna, ja laukā ievadīs vērtību, kura neatbilda likumības nosacījumam.

Required – obligāts datu ievads šajā laukā.

Allow Zero Length – vai vērtība “ ” – a zero – length string – tiek atļauta šajā laukā.

Indexed – vai ieraksti tiks indeksēti (t. i. sakārtoti) pēc šī lauka. Indeksēšana palēlina datu ievadu, toties datu meklēšana pēc šī lauka un datu kārtošana vaicājumos notiek ievērojami ātrāk. Izvēloties Yes(No Duplicates) tiek aizliegta vērtības atkārtošana šajā laukā.

Field Name	Data Type	Description
Vārds	Text	
Uzvards	Text	
Dzimums	Text	
Pers_kods	Text	
Adrese	Text	
Telefons	Text	
Bernudarzs	Yes/No	
Piezīmes	Text	

Field Properties	
General	Lookup
Field Size	50
Format	
Input Mask	
Caption	Vārds
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Allow Zero Length	No
Indexed	No

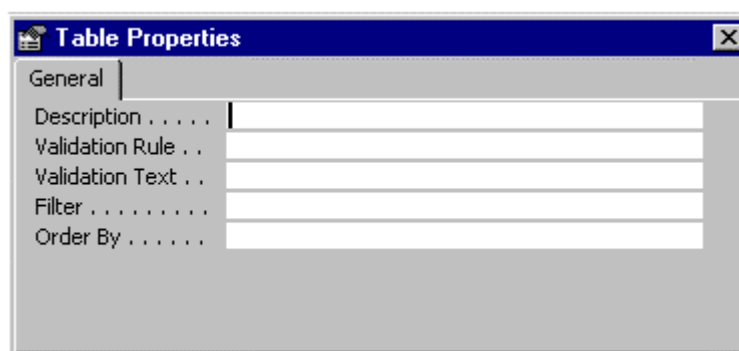
A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

11.2.att. Īpašību noteikšana.

Lookup nodaļu ir iespējams lietot tikai *Text*, *Number* un *Yes/No* tipa laukiem. *Lookup* nodaļā var norādīt, no kuras tabulas vai no kāda saraksta tiek ņemtas vērtības ievietošanai šajā laukā, ar kādu kontrolelementu šis lauks tiek parādīts formās. *Text* un *Number* tipa laukiem pēc noklusēšanas īpašība *Display Control* ir *Text Box*. *Lookup* nodaļu var atstāt arī pilnīgi neaizpildītu.

Ja, veidojot lauku, datu tipa norādīšanai tika izmantots *Lookup Wizard*, tad lauka *Lookup* nodaļa jau ir aizpildīta.

Īpašības var norādīt ne tikai laukiem bet, visai tabulai. Izpildot *View/Properties* vai klikšķinot uz attiecīgās ikonas, atvērsies tabulas īpašību logs:



11.3.att. Tabulas īpašību logs.

Pirmajām trīs tabulas īpašībām ir tāda pat nozīme kā lauku īpašībām. Jāpiebilst, ka tabulas *Validation Rule* likumības nosacījumu var veidot, saistot vairāku lauku vērtības, un tas tiek pārbaudīts, kad ieraksts tiek saglabāts, piemēram, pārejot uz jaunu ierakstu. Ieraksta ievades laikā kontrole nenotiek.

Īpašības *Filter* un *Order By* uzrāda pēdējos datu kārtēšanas un atlasē nosacījumus, kas ir attiecināmi šai tabulai. Parasti datu kārtēšana un atlase tiek izpildīta tabulas datulapā, šeit nosacījumus *MS Access* ieraksta automātiski pēc tabulas saglabāšanas – *Save* komandas izpildīšanas

11.1. Ievada maska

Vienkāršākais veids, kā laukam izveidot ievada masku, ir izmantot *MS Access* masku veidotāju – *Input Mask Wizard*:

- Tabulas veidošanas loga augšējā daļā jāatzīmē vajadzīgais lauks, ieklikšķinot tajā.
- Lauka īpašību daļā jāieklikšķina *Input Mask* rindā, lai rindas labajā pusē parādītos poga ar trīs punktiem(⋮).

Šādas pogas parādīšanās kādā rindā nozīmē, ka, noklikšķinot šo pogu, tiks izsaukts logs, kurā varēs izveidot rindas saturu.

- Jāuzklikšķina uz šīs pogas un ar *Yes* jāatbild uz *MS Access* paziņojumu, ka tabula ir jā saglabā. Līdzīgi tiek atvērts masku veidotāja *Input Mask Wizard* logs, kurā dialoga veidā jāatbild uz veidotāja jautājumiem. Beidzot dialogu, *Input Mask Wizard* logs tiek aizvērts.
- *Input Mask* rindā ir parādījies maskas kods, ko tagad var arī labot.

Maskas kods sastāv no trīs daļām, kuras atdala semikols.

Maskas koda pirmajā daļā lietojamie simboli aprakstīti zemāk tabulā.

Maskas otrajā daļā jānorāda:

0 – lai datu bāzē uzglabātu datus kopā ar maskas zīmēm;

1 – vai tukšums – lai uzglabātu tikai ievadītos datus.

Pēdējā daļā tiek norādīta zīme, kas tiek parādīta datu ievades vietās.

Ievada maskas simboli.

11.1.tabula

Maskas simbols	Datulapā vai formā jāievada
0	Skaitlis (tiek pieprasīta ievade)
9	Skaitlis (ievade netiek pieprasīta)
#	Skaitlis, + vai - zīme, vai atstarpe (ievade netiek pieprasīta)
L	Burts (tiek pieprasīta ievade)
?	Burts (ievade netiek pieprasīta)
A	Burts vai skaitlis (tiek pieprasīta ievade)
a	Burts vai skaitlis (ievade netiek pieprasīta)
&	Jebkura rakstzīme vai atstarpe (tiek pieprasīta ievade)
C	Jebkura rakstzīme vai atstarpe (ievade netiek pieprasīta)
. , : - /	Decimālpunktu, tūkstošu, datumu un laika atdalīšanai (atkarīgs no <i>Windows Control Panel</i> parametriem)
<	Pārvērš sekojošās rakstzīmju par mazajiem burtiem
>	Pārvērš sekojošās rakstzīmju par lielajiem burtiem
\	Nosaka, ka sekojošās zīmes maskā būs redzamas kā rakstzīmes. Lieto, lai simbolus, kas pieminēti šajā tabulā, varētu arī maskā parādīt kā rakstzīmes. Piemēram, \A nozīmē, ka maskā parādīsies burts A.
Password	Paroles ievade laukā. Jebkura ievadītā rakstzīme uz ekrāna parādīsies kā zvaigznīte (*).

Nākošā tabula parāda dažus noderīgu masku piemērus:

Ievada maskas piemēri.

11.2.tabula

Ievada maska	Paraugvērtības
(000) 000-0000	(206) 555-0248
(999) 000-0000	(206) 555-0248 vai () 555-0248
(000) AAA-aaaa	(206) 555-TEL
#999	-20 vai 2000
>L??L?000	GREGR339 vai MA R 993
>L<?????????????	Maria
ISBN 0-&&&-0	ISBN 1-5-7-7
>LL00-9	DB51-

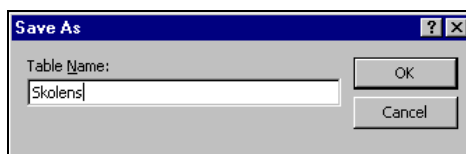
11.2. Tabulu saglabāšana

Lai apskatītu to, ko mēs esam uzkonstruējuši, jāapskatās tabula datulapas skatā; lai to izdarītu, rīku joslā jāuzklikšķina pirmajai ikoniņai no kreisās puses.



11.4.att. Tabulas skata izvēle.

Sākumā Access izvadīs paziņojumu, ka sākumā ir jāsavienā tabula – jānospiež Yes poga un nākošajā logā Save As... jāievada tabulas nosaukums.



11.5.att. File/Save As logs.

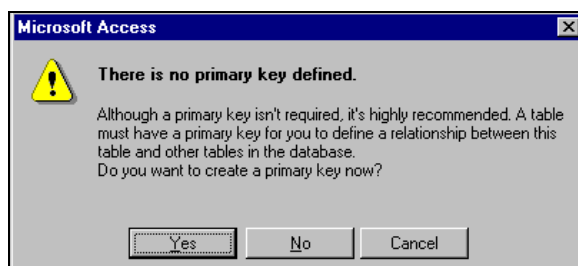
Tabulu varēja saglabāt jau iepriekš, tas novērstu iespēju pazaudēt uztaisīto (tas attiecas uz visiem citiem objektiem). Lai to izdarītu bija jānospiež uz ikoniņas Save.

Ar to varētu arī beigt tabulas lauku izveidi, bet tabulai ir jāpiešķir primārās atslēgas lauks – **Primary Key**. Tas ir lauks, kas nodrošina katra ieraksta unikalitāti – pēc šī lauka var viennozīmīgi identificēt ierakstu.

11.3. Primārās atslēgas noteikšana

Primārās atslēgas lauks pieprasa vērtības ievadu katrā ierakstā un nepieļauj dubultas vērtības. Ierakstu kārtību datulapā, atverot tabulu, nosaka primārās atslēgas lauks. To izdara apstājoties uz attiecīgā ieraksta un nospiež Primary Key uz instrumentu paneļa.

Tālāk Access izdot paziņojumu “*There is no primary key defined*”:

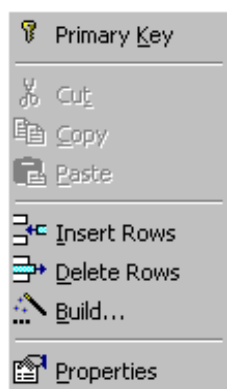


11.6.att. Paziņojums par primārās atslēgas izveidi.

Nospiežot Yes, Access automātiski izveidos primārās atslēgas lauku. Access izveido lauku ID ar datu tipu AutoNumber, kurš ir indeksēts – Yes (No Duplicates). Tālāk atver tabulu datu lapas skatā.

Kā primārā atslēga var būt arī vairāki lauki, tad šo lauku kombinācijai ir jābūt unikālai un pēc šiem laukiem viennozīmīgi var identificēt ierakstu.

Ja iepriekšējā dialoglogā atbild ar No, tad primārās atslēgas lauks netiek izveidots, bet to var izdarīt lietotājs pats tabulas veidošanas skatā.



11.7.att. Primārās atslēgas manuāla noteikšana.

Tabulas veidošanas skatā nedefinējot lauku un uzklikšķinot ar peles labo taustiņu uz šī lauka, parādās uznirstošā izvēlne, kur jāizvēlas ikoniņu *Primary Key* (atslēga). Tādā veidā var nedefinēt šo lauku par primārās atslēgas lauku.

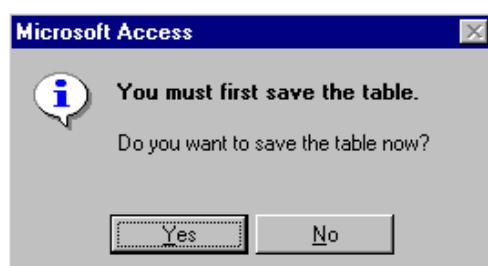
Parasti primārās atslēgas lauks ir ar datu tipu *AutoNumber*, bet var būt arī cits datu tips, vienīgais, uzstādot primārās atslēgas lauku automātiski, tiek izmainīta tā īpašība Indexed: *Yes (No Duplicates)* un šo īpašību nevar mainīt.

Tabulas primārās atslēgas lauks nav obligāti jādefinē, bet tas paātrina datu meklēšanu un atļauj saišu definīciju starp tabulām.

Ja nospiež uz *Yes*, kad Access piedāvā izveidot primāro atslēgu automātiski, tad primārās atslēgas lauks tiks nodefinēts ar nosaukumu *ID*. Šim laukam datu tips ir **Auto Number**.

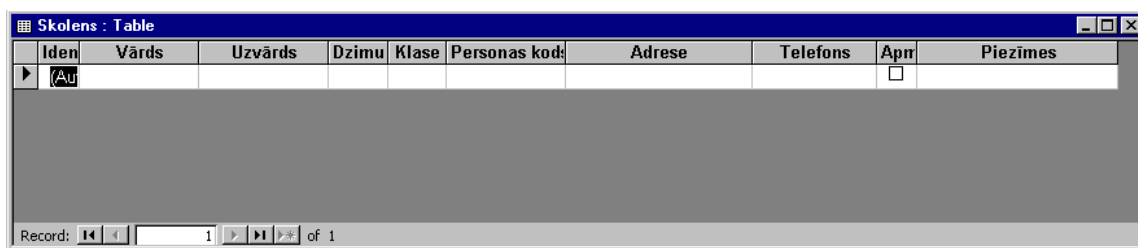
Pēc visu lauku izveides un visu īpašību noteikšanas var sākt ievadīt datus tabulā. Lai to darītu, ir nepieciešams pāriet no tabulas konstruktora uz datu lapas, nospiežot pogu Datasheet View (dotie tabulas skatā) uz instrumentu paneļa. To izdarot pogas izskats mainīsies, bet nospiežot to atkārtoti nokļūsim atkal tabulas konstruktora logā. Šai pogai atbilst arī komanda Datasheet View no izvēlnes View.

Pabeidzot darbu pie tabulas, veidošanas skatā var pāriet pie datu ievades un apstrādes. Nospiežot ikonu Datasheet View (verot tabulu datulapas skatā pirmoreiz Access prasīts tabulas nosaukumu (kā aprakstīts iepriekš), nākošā reizē Access vienkārši paziņos, ka ir jāsavaglabā tabulu.



11.8.att. Paziņojums par tabulas saglabāšanu.

Ja ir veiktas kādas izmaiņas veidošanas skatā), atveras tabula datulapas skatā:



11.9.att. Tabula datu lapas skatā.

Tagad var ķerties pie tabulas aizpildīšanas, tā pašlaik sastāv no viena tukša ieraksta.

11.4. Objektu nosaukumi

Access objektu nosaukumos pieļauj visu latīņu alfabēta simbolu lietošanu (nešķiro lielos un mazos simbolus).

Ir arī zināmi ierobežojumi un tie ir šādi:

1. **nosaukums nedrīkst sākties ar tukšumu;**
2. **nedrīkst lietot simbolus ”.” (punkts), “!” (izsaukuma zīme), “[“ un “]” (kvadrātiskās iekavas);**
3. **nosaukuma garums nedrīkst pārsniegt 64 simbolus.**

Nebūtu ieteicams lietot latviešu alfabēta burtus. Tāda iespēja ir pieļauta, bet labāk no mīkstinājuma zīmju izmantošanas izvairīties, jo tas var radīt zināms problēmas palīga (Wizard) lietošanā, reizēm tas ”saprot”, reizēm rodas Access iekšējās nesaskaņas šo mīkstinājuma zīmju lietošanas gadījumos. Labāk tos nelietot, jo šis ir tikai objekta iekšējais nosaukums un tiek izmantots tikai Access iekšējām vajadzībām, bet nosaukumos, kas parādās lietotājam tomēr ieteicams izmantot šo lielisko iespēju.

Access nepieļauj divus vienādus nosaukumus (viena objektu tipa ietvaros). Ir situācijas, kad vienādu nosaukumu gadījumā viens no objektiem tiek pazaudēts (lietotāj brīdina, bet labāk līdz šādām situācijām nenonākt), ir arī situācijas, kad vienādu nosaukumu gadījumā jaunākais tiek papildināts ar skaitli.

Access pieļauj divus vienādus nosaukumus objektiem, ja nesakrīt to tips (piemēram, tabula “Darbinieki” un tāda paša nosaukuma vaicājums). Arī no šīs iespējas vajadzētu atteikties, jo var rasties diezgan muļķīgas situācijas, kad nevar identificēt atbilstošo objektu, ja nav zināms tā tips (tā parasti notiek ar tabulām un vaicājumiem).

Nosaukumus vajadzētu izvēlēties atbilstoši funkcijām, kādas veic konkrētais objekts. Izvairieties no Access piedāvātajiem nosaukumiem – *Table1*, *Table2*, *Query1*, *Query2*, utt., ieteicams kaut mazliet nosaukumā ielikt kādu pazīmi, kas varētu palīdzēt identificēt objektu.





12. Darbs ar tabulas datiem

12.1. Datu ievads

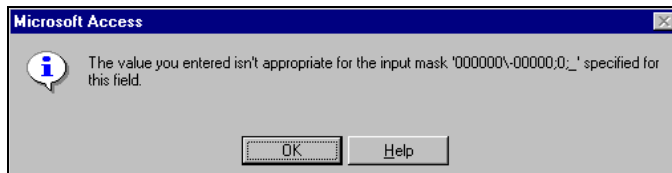
Atverot pirmoreiz, tabula ir tukša, ir tikai jaunā ieraksta rinda. Sākumā šai rindai vēl nav numura (laukā *Identifikators* ir uzraksts *AutoNumber* un šo lauku nevar izmainīt). Tikko ievada pirmo ierakstu, tām automātiski parādās numurs (par vienu lielāks nekā iepriekšējais) un parādās jauna tukša ieraksta rinda. Lai ievadītu uzvārdu, jānospiež taustiņš *Tab* vai ar peli jānorāda, kurā laukā vēlas ievadīt datus.

Tabulās dati ir izvietoti kolonnās – tās sauc par laukiem, un rindās – tās sauc par ierakstiem.

Datulapas augšējā rindā ir lauku nosaukumi. Datulapas kreisajā pusē ir ierakstu izvēles taisnstūri (record selectors). Uz tiem var būt šādi apzīmējumi:

-  • esošais ieraksts – kursors ir novietots šajā ierakstā;
-  • jauns ieraksts – nākošais aiz pēdējā ieraksta;
-  • slēgts ieraksts – daudzlietotāju vidē to pašlaik labo kāds cits lietotājs,
-  • ieraksts tiek labots, izmaiņas vēl nav apstiprinātas.

Ja ievadītais neatbilst ievada maskai, tad Access izdot paziņojumu un tālāk netiek atļauta datu ievade, kamēr nav izlabots šis lauks.



12.1.att. Paziņojums, ja ievadītais neatbilst ievada maskai.

Ieraksts tiek saglabāts brīdī, kad lietotājs pāriet uz citu ierakstu. Ieraksts tiek saglabāts tikai gadījumā, ja izpildās visi nosacījumi, kas aprakstīti tabulas definīcijā. Katram ierakstam tiek iedots savs identifikators, kas pieaug par vienu ar katru nākošo ierakstu.

Ierakstu var izdzēst, iezīmējot to un nospiežot *Del*. Izdzēšot ierakstu, nākamajiem laukiem netiek piešķirts tāds pats identifikators – vienkārši ieraksta ar tādu identifikatoru nav.

Tāpat kā Excel, arī Access datulapas izskatu var mainīt: var nomainīt katras kolonas platumu; rindas augstums tiek uzstādīts visām rindām kopā. Kolonnu izvietojumu var mainīt – iezīmējot (uzspiežot uz kolonas virsraksta) un pēc tam pārvelkot vajadzīgajā vietā (līdz ar to mainās arī lauku izvietojums tabulas veidošanas skatā).

Datulapas izskatu var mainīt arī izvēlnes (Menu) palīdzību, šim nolūkam var izpildīt:

- **Format/Font...** – fonu uzstādīšana (visai datulapai);
- **Format/Cells...** – šūnu izskata uzstādīšana;
- **Format/RowHeight...** – rindas augstuma uzstādīšana;
- **Format/ColumnWidth...** – kolonnu platuma uzstādīšana.

Access datulapā var izmantot pierastās taustiņu kombinācijas kopēšanai: *Ctrl+C*, *Ctrl+V*, *Ctrl+X*, bet ieteicams iemācīties vēl dažas citas kombinācijas, lai varētu ērtāk ievadīt un izlabot datus:

- *Tab/Shift+Tab* – pāriet uz nākošo/ iepriekšējo lauku;
- *Home/End* – pāriet uz ieraksta pirmo/pēdējo ierakstu;
- *<Down Arrow>/<Up Arrow>* (kursortaustiņi) – pāriet uz nākošo/iepriekšējo ierakstu;
- *Ctrl+Home/Ctrl+End* – pāriet uz pirmā ieraksta pirmo lauku/pēdējā ieraksta pēdējo lauku;
- *PageUp/PageDown* – pāriet uz augšu/apakšu par vienu ekrāna lapu;
- *F2* – iezīmēt aktīvo lauku;
- *Esc* – nesaglabāt izmaiņas laukā;
- *Ctrl+Enter* – izveidot laukā jaunu rindu;
- *Ctrl+''* – aizvietot lauka vērtības ar tā paša lauka iepriekšējo vērtību;
- *F9* – atsvaidzināt ierakstus;
- *Alt+<Down Arrow>* - atvērt komboboksi.

Šīs visas komandas izpildās arī vaicājumu datulapas skatā un formās. Lai atvieglotu darbu ar ierakstiem datulapā, katrai no tām ir navigācijas pogas (*Navigation Buttons*):



12.2.att. Navigāciju pogas.

Klikšķinot uz tām, var nokļūt uz pirmo, iepriekšējo, nākošo, pēdējo vai jauno ierakstu. Vēl tiek rādīts arī kopējais ierakstu skaits un aktīvā ieraksta numurs. Ja vēlas nokļūt uz konkrētu ierakstu – aktīvā ieraksta numura laukā jāievada vajadzīgais numurs un jānospiež taustiņu *Enter*.

12.2. Hipersaišu izmantošana tabulā

Kā jau minēts, lauki ar hipersaites datu tipu ir piemēroti tāda veida informācijas kā elektroniskā pasta, Interneta adrešu glabāšanai. Šādā laukā ierakstīto MS Access uztver kā hipersaites adresi, un tabulas datulapā vai formā, kas balstās uz šo tabulu, noklikšķinot tiek aktivizēta hipersaite. Atbilstoši hipersaitei tiek atvērta WWW tīmekļa pārlūkprogramma vai elektroniskā pasta programma.

Piemēram, tabulā Darbinieki ir nepieciešams atzīmēt arī katra darbinieka elektroniskā pasta adresi un Interneta mājas lapas adresi.

Vispirms tabulā Darbinieki ir jāizveido šādi lauki. Atveram tabulu Darbinieki veidošanas skatā un lauku saraksta beigās izveidojam vēl divus laukus:

Darbinieki : Table			
	Field Name	Data Type	
	Piezīmes	Memo	Pa
	E-pasts	Hyperlink	
	Internet	Hyperlink	

12.3.att. Hipersaites lauka izveide.

Pārejam uz datulapu, tabulu saglabājot. Pirmajā jaunajā laukā E-pasts ierakstām kādu elektroniskā pasta adresi, to sākot ar tekstu "**mailto:**", piemēram, **mailto:janis@cl.lv**. Ierakstītā adrese tiek pasvītrotā un iekrāsota zila.

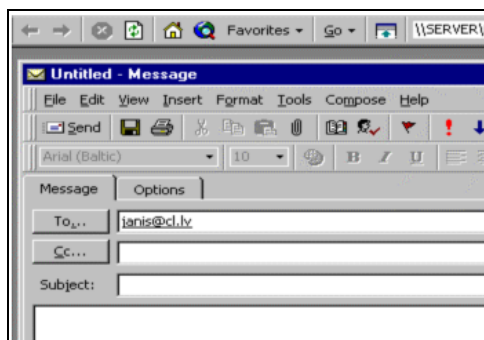
Otrajā laukā Internet ierakstām Interneta adresi standarta pieraksta veidā, piemēram, www.cl.lv.

Darbinieki : Table					
	Studs	Telefons	Piezīmes	E-pasts	Internet
	No	(371)-7-28	Beidzis LU.	mailto:janis@cl.lv	www.cl.lv
	No	(358)-0-54	Dzīvo Helsinki.		

12.4.att. Interneta adreses ievads.

Pēc hipersaites lauka apstiprināšanas, kursoru novietojot uz šī lauka, tas kļūst par baltu plaukstu ar rādītājpirkstu. Noklikšķinot uz šī lauka,

- ikonu joslā parādās tīmekļa rīku josla (Web toolbar), ja vien tās tur vēl nav;
- atkarībā no ierakstītās adreses tiek atvērta programma. Šajā gadījumā pēc klikšķa uz E-mail lauka tiek atvērta Microsoft Outlook e-pasta programma, kas ir šī datora elektroniskā pasta programma:



12.5.att. E-pasta programmas startēšana.

Pēc ziņojuma nosūtīšanas e-pasta programmu var aizvērt un turpināt darbu ar datu bāzi.

Pēc klikšķa Internet laukā atveras attiecīgā WWW tīmekļa lappuse. Pabeidzot darbu ar tīmekļa pārlūkprogrammu, to var aizvērt un turpināt darbu ar datu bāzi.

Hipersaite var būt *UNC* (standarta formāts, lai norādītu ceļu uz lokālā tīkla serveri) ceļš vai *URL* norāde (adrese uz objektu, dokumentu, lappusi utt. Internetā vai intranetā).

Kad noteiktam protokolam atbilstošais teksts tiek drukāts hipersaites laukā, MS Access mēģina to atpazīt un pārveidot pareizas hipersaites adreses formā. Piemēram, tekstu "<http://www.microsoft.com>", MS Access pārveido par [#http://www.microsoft.com#](http://www.microsoft.com#).

Hipersaites adrese var sastāvēt no 3 daļām, kas atdalītas ar zīmi (#):

ekrāna teksts#adreses galvenā daļa#apakšadrese

Hipersaites adreses daļas

12.1.tabula

Adreses daļa	Apraksts
Ekrāna teksts	Teksts, kas ir redzams laukā vai ar lauku saistītajā formas kontrolelementā. Tas var būt vienkārši aprakstošs teksts, tā lietošana nav obligāta.
Adreses galvenā daļa	Absolūtais vai relatīvais ceļš uz mērķa dokumentu. Absolūtais ceļš ir pilns <i>URL</i> vai <i>UNC</i> ceļš. Relatīvais ceļš var būt uzrādīts attiecībā pret bāzes ceļu, kas norādīts <i>Hyperlink base</i> lodziņā datu bāzes <i>Properties</i> logā (pieejams, izpildot <i>File/Database Properties</i>) vai attiecībā pret ceļu, kas norāda uz esošo datu bāzi. Adreses galvenā daļa jāuzrāda obligāti, izņemot vienīgi tos gadījumus, ja apakšadreses daļa norāda uz objektu esošajā datu bāzē.
Apakšadrese	Atrašanās vieta failā vai lappusē. Piemēram, var norādīt uz kādu formu vai pārskatu <i>MS Access</i> datu bāzē, vai kādu grāmatzīmi <i>MS Word</i> dokumentā. Lietošana nav obligāta.

Visas trīs hipersaites daļas uz datora ekrāna netiek parādītas. Ja vajag redzēt pilnu hipersaites adresi, kursors jānovieto uz hipersaites lauka (neklikšķinot, lai neaktivizētu saiti) un jānospiež F2 tastatūras taustiņš.

- Ja ir ievadīta adreses ekrāna teksta daļa, MS Access neparāda atlikušās adreses daļas.
- Ja nav ievadīta adreses ekrāna teksta daļa, MS Access parāda tikai adreses galveno daļu.
- Apakšadresi parāda tikai tad, ja pirmo divu daļu nav un ja apakšadrese norāda uz objektu esošajā datu bāzē.

Ievadot hipersaites adresi, atļauta atsevišķu adreses daļu izlaišana, piemēram:

- ja tiek norādīts uz kādu objektu esošajā datu bāzē, pietiek norādīt tikai objekta nosaukumu. Ja datu bāzē ir vairāki objekti ar to pašu nosaukumu, MS Access meklē šādā secībā: formas, pārskati, tabulas/vaicājumi, makro, moduļi. Objekta tipu var norādīt pirms objekta nosaukuma, piemēram, Table Orders, lai norādītu uz tabulu Orders;
- ja ievadītais teksts nesatur zīmi (#), protokolu (piemēram, http:), vai atsauci uz objektu esošajā datu bāzē, MS Access šo tekstu uztver kā ekrāna tekstu un adreses galvenajā daļā šo tekstu pievieno pie "http://".

Kad URL atbilstošs teksts tiek drukāts hipersaites laukā, MS Access var atpazīt šādus protokolus (nākamajā tabulā tiek pieminēti biežāk lietotie).

Protokoli un to apraksts.

12.2.tabula

Protokols	Apraksts
http	Atver WWW tīmekļa lappusi.
ftp	Pārsūta failus no datora uz datoru ar Internet starpniecību.
mailto	Atver jūsu elektroniskā pasta programmu, lai nosūtītu ziņojumu norādītajai Interneta e-pasta adresei. Sintakse: <i>mailto:lietotāja vārds@domēns</i>
nntp	Uzsāk jaunumlasītāju ² (<i>newsreader</i>) un atver norādīto tīkla <i>Usenet</i> intereškopu ³ (<i>newsgroup</i>). Sintakse: <i>news:intereskopas nosaukums</i>
mid	Atskaņo <i>MIDI</i> failus, ja lietotāja datorā ir skaņas karte.
telnet	Uzsāk <i>telnet</i> termināla emulācijas programmu.

Instalējot Microsoft Office, Setup programma instalē failus, kas nepieciešami, lai savienotu visus MS Office dokumentu tipus un failus uz http vai ftp serveriem. Citiem protokoliem ir nepieciešama attiecīgo failu instalācija.

² jaunumlasītājs (**newsreader**) - programma, ko tīklā Usenet izmanto e-rakstu lasīšanai, saglabāšanai un atbilžu sagatavošanai, kā arī jaunu e-rakstu nosūtīšanai.

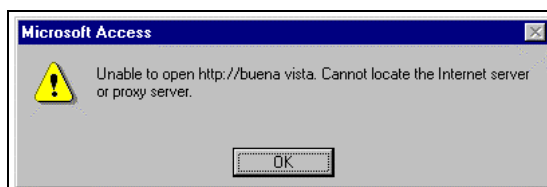
³ Intereškopa (**newsgroup**) - ziņojumdeļa sistēmas lietotāju grupa, kuru apvieno kopīgas intereses, piemēram, sports, mūzika, vēsture vai politika.

Hipersaites adreses pieraksta piemēri.

12.3.tabula

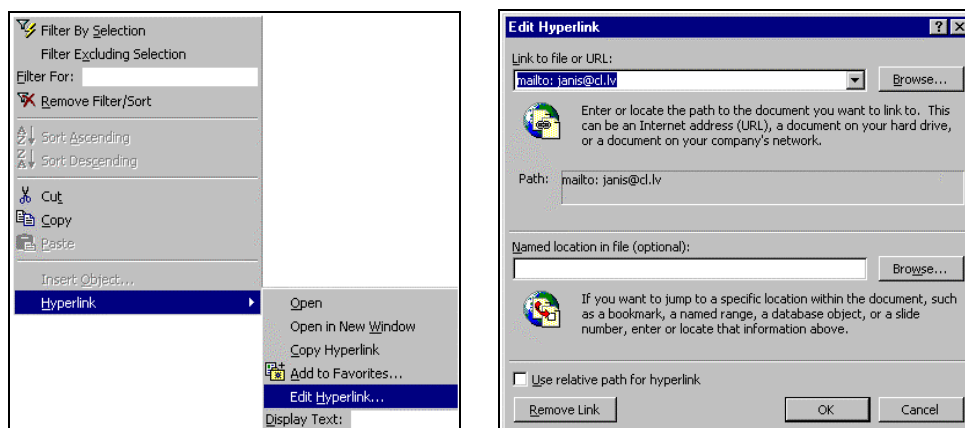
Hipersaites adreses piemērs	Apraksts
Kaķu pasaule#http:// www.cats.com#	Lappuse WWW tīmeklī. Adrese uz ekrāna parādās kā teksts <i>Kaķu pasaule</i> .
#c:\program files\ msoffice\ office\ samples\ northwind.mdb#Form Suppliers	<i>Suppliers</i> forma <i>Northwind</i> datu bāzē, kas atrodas c:\program files\ msoffice\office\samples
Piegādātāji##Suppliers	<i>Suppliers</i> forma esošajā datu bāzē. Adrese uz ekrāna parādās kā teksts <i>Piegādātāji</i>
#c:\mani dokumenti\ 1998 G.ppt#13	Slīds 13 <i>PowerPoint</i> dokumentā 1998 G, kas atrodas c:\mani dokumenti
#c:\mani dokumenti\ nauda.xls#Sheet1!A2	A2 šūna Sheet1 lapā nauda.xls failā, kas atrodas c:\mani dokumenti
#c:\mani dokumenti\ pārskats .doc#specifikācija	Sekcija failā pārskats .doc ar grāmatzīmi specifikācija, kas atrodas c:\mani dokumenti

Ja ierakstītā adrese ir nepareiza, tad pēc klikšķa uz tās parādīsies attiecīgs paziņojums:



12.6.att. Paziņojums, ja nevar atvērt saiti.

Lai rediģētu hipersaites lauku vai arī lai izdzēstu hipersaiti, vispirms uz lauka jānoklikšķina peles labais taustiņš un komandkartē, kas tiek atvērta, jāizvēlas *Hyperlink/Edit Hyperlink*.



12.7.att. Saites rediģēšana.

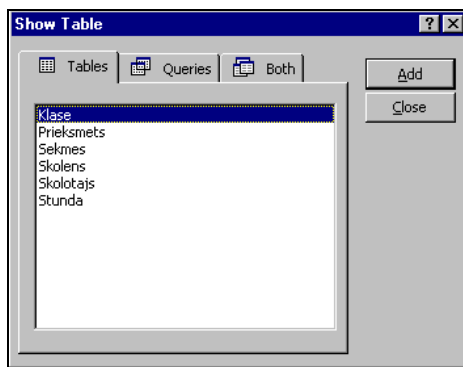
Edit Hyperlink logā var labot hipersaites adreses galveno daļu un apakšadresi. Lai labotu hipersaites ekrāna tekstu, pēc peles labā kursora klikšķa uz lauka jāizpilda Hyperlink un lodziņā Display Text jāieraksta vajadzīgais teksts.

13. Relāciju (saišu) izveide

Tabulu saišu definēšanai Access izmanto logu Relationship. Lai iegūtu šo logu uz datubāzes loga Design View, kaut kur ir jānospiež peles labais taustiņš un jāizvēlas izvēlne Relationships (šo logu var arī iegūt ar rīku joslas palīdzību), un atveras Relationships logs. Pirmo reizi atverot šo logu un veidojot jaunu relāciju modeli datu bāzei, tas ir tukšs, bet gadījumā, ja tiek atvērts jau esošais modelis (atver tāpat kā jaunu), tas satur jau iepriekš definēto modeli un to var palabot. Jālabo ļoti uzmanīgi, jo tādejādi var izjaukt jau darbojošos datu bāzi!

Bet, atverot jaunu, tas ir tukšs, relāciju modelis jādefinē pilnīgi no sākuma.

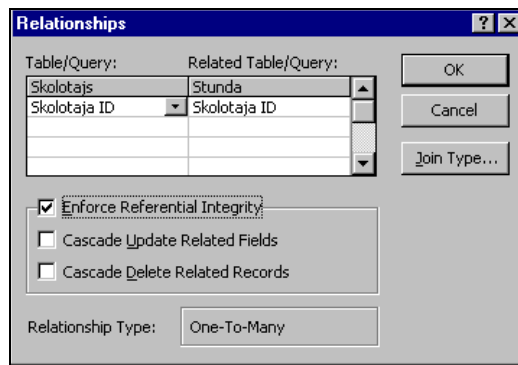
Kā to dara? Pirmais, kas ir jāizdara, jāpievieno visas datu bāzes tabulas (ir gadījumi, kad ne visas, bet no tā ieteicams atturēties). Lai pievienotu jaunas tabulas modelim kaut kur uz Relationships loga jānoklikšķina peles labais taustiņš un jāizvēlas izvēle Show Table (pirmo reizi tas nav jādara, jo logs Show Table atveras automātiski pēc tam, kad ir atvēries logs Relationships) vai arī Relationships logā, izpildot *Relationships/Show Table*.



13.1.att. Show Table logs.

Šinī logā izvēlas tabulas vai vaicājumus, kādus vēlas pievienot relāciju modelim (šeit parādās pirmā atšķirība starp vienkāršu ER modeli un Access relāciju modeli – Relationships var saturēt ne tikai tabulas, bet arī vaicājumus).

Pēc tam, kad ir pievienotas visas tabulas, jānedefinē saites starp šīm tabulām. Jāpiezīmē, ka laukiem, kas tiek savienoti, var būt atšķirīgi nosaukumi, taču tiem noteikti ir jābūt tā paša tipa laukiem. Ja lauki ir Number tipa, tiem jābūt ar vienādu FieldSize īpašību. Jāievēro, ka AutoNumber tipa lauku var saistīt ar Number tipa lauku, kuram FieldSize īpašība ir LongInteger. To dara šādi: aktivizē kādu tabulas lauku, ar kuru vēlas saistīt citu tabulu, un, neatlaižot peles kreiso taustiņu, pārvelk uz otras tabulas lauku, kā rezultātā parādās saites definēšanas logs Relationships.



13.2.att. Relationships logs.

Formas augšā tabulā parāda, kuru tabulu un kādi lauki tiek saistīti (jāskatās, lai velkot saiti, tās būtu novilkas pareizi). Apakšā parādīts saites veids: One-To-Many, One-To-One utt. Tas uzstādītās automātiski – Access pats nosaka, kāda saite var pastāvēt starp šīm tabulām.

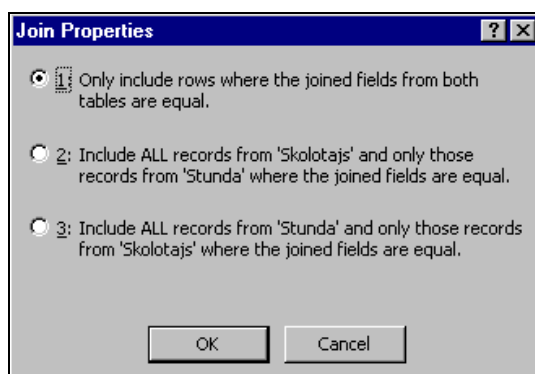
Enforce Referential Integrity nozīmē vai starp tabulām ir definēta referenciālā integritāte.

Referenciālā integritāte nozīmē, ka pievienojot vai izdzēšot vienas tabulas ierakstus, tiks ietekmēti saistītās tabulas ieraksti.

Ar īpašībām *Cascade Update Related Fields* un *Cascade Delete Related Records* palīdzību var mazliet mainīt referenciālo integritāti:

Cascade Update Related Fields – atzīmējot šo īpašību, būs atļauts tabulā (saites galā – viens (one)) mainīt primārās atslēgas lauku, un Access automātiski izmaina šo vērtību otrajā tabulā (saites galā – daudz (many)). Ja tabulā primārās atslēgas lauks ir ar datu tipu AutoNumber, tad šī īpašība neko nedod.

Cascade Delete Related Records – atzīmējot šo īpašību, būs atļauts tabulā (saites galā – viens (one)) dzēst ierakstus, Access automātiski izdzēš visus atbilstošos ierakstus otrajā tabulā (saites galā – daudz (many)). Apskatot tabulas Klase un Skolens, ja tiktu atzīmēta šī iespēja, tad varētu izdzēst jebkuru klasi, bet tad automātiski tiktu izdzēsti visi skolēni šajā klasē. Tāpēc vajadzētu uzmanīties ar šīs īpašības izmantošanu.



13.3.att. Join Properties logs.

Vēl var modificēt saiti, nospiežot pogu **Join Type...** Atveras logs Join Properties, kurā var izvēlēties, vai saite būs *inner joins* (1) – tā ir noklusētā vērtība, *left outer joins* (2) vai *right outer joins* (3).

Izvēloties šos veidus, mainās tikai veids, kā izpildīsies vaicājumi, kas balstīsies uz šīm tabulām, bet nemainās pašu tabulu saistība.

Inner joins – tiek atlasītas tikai tās rindiņas, kurām saistītie lauki ir vienādi;

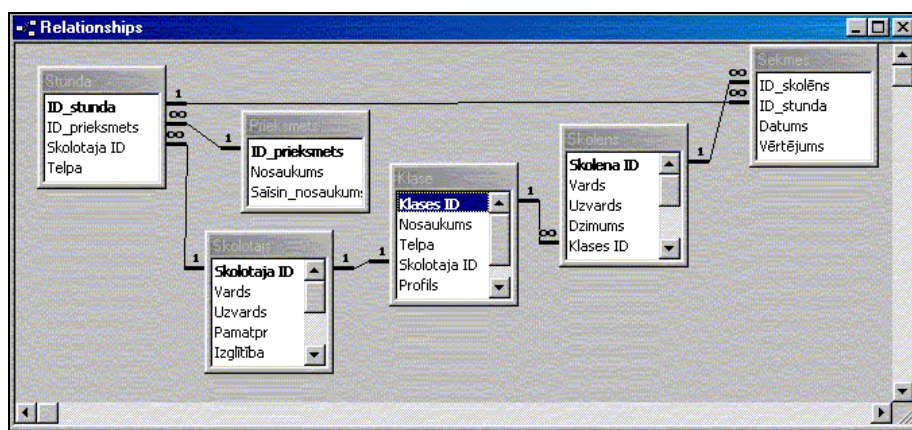
Left outer joins un **Right outer joins** – tiek atlasīti visi ieraksti no vienas tabulas un tikai tie ieraksti no otras, kur saistītie lauki ir vienādi.

Aizverot Relationships logu, MS Access jautā, vai saglabāt saišu definīcijas izskatu, t. i., jautājums ir tikai par loga izskatu. Saišu definīcijas tiek saglabātas neatkarīgi.

Lai mainītu jau esošo saiti, jāizpilda dubultklikšķis uz šīs saites, atvērsies saišu īpašību logs, kurā var izmainīt saites definīciju.

Lai dzēstu esošo saiti, jāuzklikšķina uz šīs saites un jānospiež Delete taustiņš.

Pievienotai tabulai (linked table) var definēt saiti, taču nevar definēt referenciālo integritāti.



13.4.att. Tabulu savstarpējā saistība .

Iespējams izveidot saites katrā vaicājumā atsevišķi, taču referenciālo integritāti var definēt tikai šeit.

14. Datu kārtošana, meklēšana un aizvietošana

14.1. Datu kārtošana

Bieži vien noder iespēja sakārtot ierakstus pēc kaut kāda lauka. Lai to izdarītu, uz kolonnas, pēc kuras grib kārtot, jānospiež peles labais taustiņš – parādās izvēlne, no kuras jāizvēlas:

- *Sort Ascending* – sakārtot augošā secībā;
- *Sort Descending* – sakārtot dilstošā secībā.

To var izdarīt arī ar rīku joslas ikoniņu palīdzību.



14.1.att. Datu kārtošana .

Piemēram sakārtojot *Skolenu* tabulu pēc lauka *Uzvārds*, iegūst šādu datulapu:

Skolens : Table							
Iden	Vārds	Uzvārds	Dzimu	Personas kod	Adrese	Telefons	Apr
246	Ilmārs	Ābelīte	Vīr	051286-10300	Rīgā, Lomonosova N 7-3		<input type="checkbox"/>
264	Sandija	Āboliņa	Siev	190687-11004	Ogrē, Mālikalna N 9-24	7253923	<input checked="" type="checkbox"/>
93	Bronislava	Andersone	Siev	240790-10122	Rīgā, Vaidavas N 2-28 k. 1	7319332	<input type="checkbox"/>
213	Māriete	Antonova	Siev	160488-11822	Rīgā, Dainas N 1-2		<input checked="" type="checkbox"/>
205	Valentīns	Arbidans	Vīr	130488-11844	Rīgā, Brīvības N 100-16a		<input checked="" type="checkbox"/>
169	Ilze	Auslande	Siev	220389-12711	Rīgā, J. Sudrabkalna N 64-99	7318632	<input type="checkbox"/>
148	Zenta	Auslande	Siev	190189-10318	Rīgā, Krāslovas N 34-4		<input checked="" type="checkbox"/>
33	Žanna	Averjanova	Siev	110491-10468	Rīgā, M. Matīsa N 3-9	2689732	<input type="checkbox"/>
92	Māris	Avotnieks	Vīr	051289-11044	Ogrē, Turkalnes N 1a-5	7319344	<input checked="" type="checkbox"/>
233	Genādijs	Baikovs	Vīr	180587-10301	Rīgā, Zvaigžņu N 22-23	7524532	<input checked="" type="checkbox"/>
69	Edvīns	Baldevāns	Vīr	040190-10505	Rīgā, Gobas N 25-10	2644742	<input checked="" type="checkbox"/>
110	Bruno	Balgālis	Vīr	280990-13071	Rīgā, Blaumaņa N 9-81	7916867	<input checked="" type="checkbox"/>
258	Edīte	Bērziņa	Siev	231087-10321	Rīgā, Nākotnes N 3-5		<input checked="" type="checkbox"/>
243	Andrejs	Bebrišs	Vīr	050787-10412	Rīgā, Maskavas N 250-11. k.-81	2357512	<input type="checkbox"/>
132	Ģirts	Bedrītis	Vīr	051089-11221	Rīgā, Praulienas N 14-57		<input checked="" type="checkbox"/>
54	Haralds	Beikmanis	Vīr	110291-10507	Rīgā, Meža N 12-2	7565472	<input checked="" type="checkbox"/>
151	Rudīte	Bendzule	Siev	050289-12078	Bauskas r. Iecavas p. Ozolu 1-11	6498984	<input checked="" type="checkbox"/>
230	Aleksandrs	Bernāns	Vīr	190488-11234	Rīgā, Rītupe N 28-16		<input type="checkbox"/>
51	Arnis	Birzgalis	Vīr	250891-12789	Rīgā, Klusā N 11-12	7319002	<input checked="" type="checkbox"/>
207	Semions	Brūveris	Vīr	280588-11013	Ogres r. Ikšķilē, Peldu N 3	7318742	<input checked="" type="checkbox"/>
73	Edvīns	B...	Vīr	050289-12078	Bauskas r. Iecavas p. Ozolu 1-11	6498984	<input type="checkbox"/>

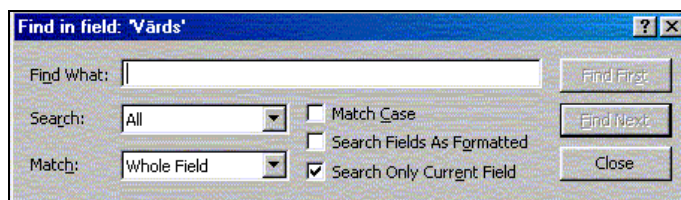
14.2.att. Sakārtota datu lapa .

Sakārtot var arī pēc vairākiem laukiem (blakusesošiem) – jāieezīmē vajadzīgās kolonas un jāizpilda iepriekš aprakstītais.

14.2. Datu meklēšana un aizvietošana

Lai atrastu ierakstu pēc kāda lauka vērtības, jāieklikšķina vajadzīgajā laukā (vienalga kurā ierakstā) un jānospiež ikoniņa *Find* vai arī izpilda *Edit/Find...*

Izvēloties kolonu *Vārds* un nospiežot ikoniņu – atveras logs *Find in field: 'Vārds'*:



14.3.att. Logs Find in field.

Laukā *Find What* ieraksta meklējamo izteiksmi.

Search laukā jāizvēlas meklēšanas virzienu (*Up* – uz augšu, *Down* – uz apakšu, *All* – pa visu tabulu).

Match laukā jāizvēlas meklēšanas veids (*Any Part of Field* – lauka vērtībai jāsaturs meklējamā izteiksme, *Whole Field* – meklējamai izteiksmei jābūt identiskai ar lauka vērtību, *Start of Field* – lauka vērtībai jā sākas ar meklējamo izteiksmi).

Čeksbokšos jāizvēlas:

Match Case – ja vēlas, lai tiktu atrasti precīzi tādi vārdi kā norādīts (izšķirtu lielos un mazos burtus);

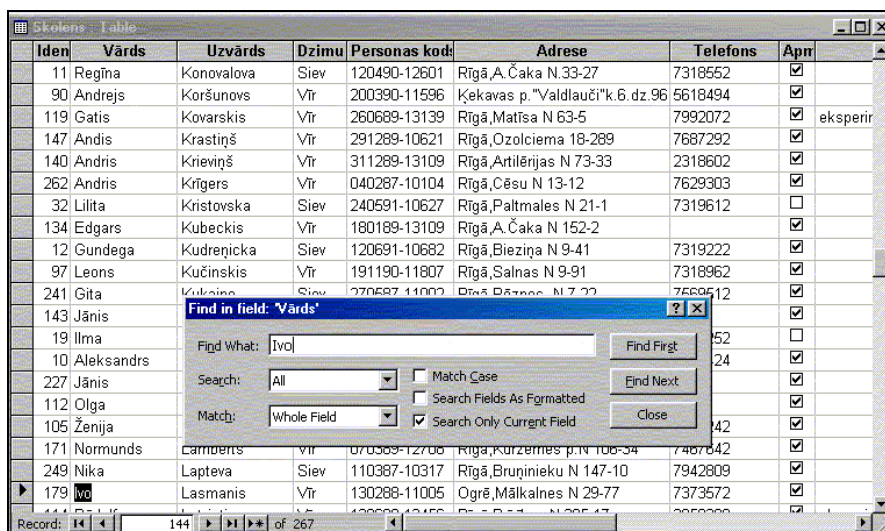
Search Field as Formatted – ja vēlas meklēt izteiksmes, tādas, kā tās ir uz ekrāna;

Search Only Current Field – meklēt tikai norādītajā laukā (ja nav atķeksēts, tad meklē visā tabulā);

Ar pogas *Find First* atrod pirmo lauku, kas atbilst nosacījumiem, poga *Find Next* palīdz atrast nākošos laukus, kas atbilst nosacījumam.

Laukā *Find What* ievadam, piemēram, *Ivo*.

Nospiežot pogu *Find First* – pirmais ieraksts, kas atbilst nosacījumiem kļūst aktīvs.



14.4.att. Meklēšana.

Nospiežot pogu *Find Next* (lai atrastu vēl kādu lauku, kas atbilst nosacījumiem), Access izdod paziņojumu, ka vairāk tādu lauku nav – tas

nozīmē, ka tabulā ir tikai viens skolēns ar vārdu *Ivo*. Ja tāda skolēna nav, tad nospiežot pogu *Find Next*, tiek paziņots, ka šāda lauka nav.

Ja vēlas lauka vienu vērtību aizstāt ar citu, tad ar ***Edit/Replace...*** – atveras logs *Replace*. Šajā logā visi lauki ir tādi paši kā logā *Find*, vienīgais klāt nāk lauks *Replace With*, kurā jānorāda vērtība, ar kuru aizvietot meklējamo vērtību.

15. Filtri

Ar meklēšanas un kārtošanas palīdzību var atrast vajadzīgos ierakstus, bet tas nav diezgan ērts paņēmieni, ja ierakstu ir ļoti daudz. Daudz ērtāk būtu, ja varētu atrast vajadzīgos ierakstus un atstāt tikai tos, pārējos nerādīt. Šādam nolūkam lieto filtrus. Tas var lietot:

Filtrēšana pēc iezīmētiem datiem.

Iezīmē informāciju, pēc kuras vēlas filtrēt un nospiež uz ikoniņu *Filter By Selection* vai izpilda **Records/Filter/Filter By Selection**, vai ar labo peles taustiņu iegūst iznirstošo izvēlni un izvēlas *Filter By Selection*. Izpildot to, tabulas datus nofiltrē un tiek parādīti tikai atbilstošie ieraksti.



Idens	Vārds	Uzvārds	Dzimu	Personas kod	Adrese	Telefons	Apr	
9	Oļegs	Gordijenko	Vīr	250891-10642	Rīgā, Bieziņa ielā N 9-41	7319211	<input checked="" type="checkbox"/>	
106	Oļegs	Moisejevs	Vīr	070890-10422	Rīgā, Zilupes N 31-42	7319322	<input checked="" type="checkbox"/>	
162	Oļegs	Panarins	Vīr	051289-10119	Rīgā, Vaidavas 2-16	7957092	<input checked="" type="checkbox"/>	Grūti audz

Record: 1 of 3 (Filtered)

15.5.att. Filtrēšana pēc iezīmētiem datiem.

Tika filtrēts pēc iezīmētiem datiem *Oļegs* laukā *Vārds*. Tika atrasti trīs ieraksti, kas atbilst nosacījumiem (lauka *Vārds* vērtība ir *Oļegs*). Tabulas loga apakšējā daļā ir redzams, ka ir pielietots filtrs – uzraksts *of 3 (Filtered)*. (sk. attēls)

Lai atsauktu filtru un atkal parādītu visus ierakstus, jāuzklikšķina uz ikonas *Remove Filter* vai izpilda **Records/Remove Filter Sort**, vai no uznirstošās izvēlnes *Remove Filter/Sort*:

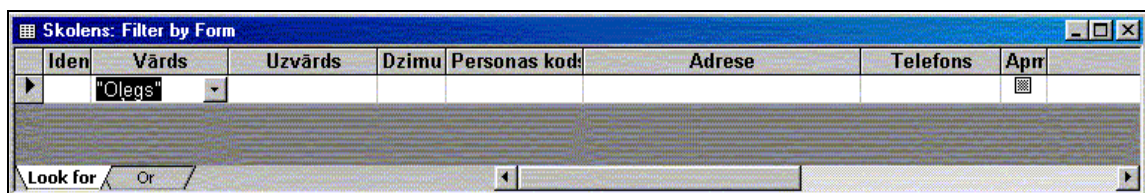
- Ja ir iezīmēts viss lauks (tā kā iepriekšējā piemērā), tad tiek atlasīti visi ieraksti, kas ir identiski;
- ja ir iezīmēts lauka sākums (piemēram, tikai *O*), tad tiek atlasīti visi ieraksti, kas sākas ar *O*;
- ja ir iezīmētas lauka beigas (piemēram, tikai *gs*), tad – visi, kas beidzas ar *gs*;
- ja ir iezīmēts lauka vidus (piemēram, tikai *eg*), tad – visi, kas satur izteiksmi *eg*.

Ja ir nepieciešams atlasīt pretējos datus (datus, kas neatbilst nosacījumiem), tad izpilda **Records/Filter/Filter Excluding Selection** vai no uznirstošās izvēlnes *Filter Excluding Selection*.

Vēl līdzīgs filtra veids ir no uznirstošās izvēlnes *Filter For*., kur jāievada vajadzīgā izteiksme un tiek atlasīti ieraksti, kas atbilst šai izteiksmei.

Filtrēšana pēc formas.

Tas ir filtru veids, kad var filtrēt pēc vairākiem nosacījumiem un vairākiem laukiem. Lai to izdarītu, jāuzklikšķina uz ikoniņas *Filter By Form* vai jāizpilda komanda **Records/Filter/Filter By Form** – atveras filtra veidošanas logs.



15.6.att. Filtra veidošanas logs.

Šajā logā katram laukam var ievadīt nosacījumu. Nosacījums var būt (atbilstoši iepriekšējam piemēram): *Oļegs*, *O**, **gs*, **eg**. Access automātiski tos pārveido *par Like "Oļegs"*, *Like "O*"*, utt. Tādā veidā var uzrakstīt nosacījumu katram laukam (ja nevēlas kādu lauku filtrēt – vienkārši atstāj tukšu). Nosacījumi tiek pielietoti ar loģisko operāciju *Un* (visiem nosacījumiem jāizpildās), ja vēlas, lai darbotos *Vai*, jāraksta nodaļā *Or*. Lai pielietotu šo filtru jāuzklikšķina uz ikoniņas *Apply Filter* vai izpilda **Filter/Apply Filter Sort**.

AND un OR operatoru salīdzinājums

15.1.tabula

OR

AND

X	Y	Rezultāts
0	0	Nav
1	0	Ir
0	1	Ir
1	1	Ir

X	Y	Rezultāts
0	0	Nav
1	0	Nav
0	1	Nav
1	1	Ir

0 – kritērijs neizpildās;

1 – kritērijs izpildās;

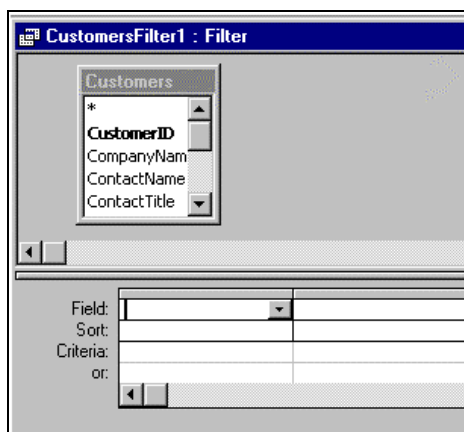
X, Y – kritēriji.

Ja kritēriji tiek sastādīti dažādiem laukiem vienā nodaļā, atlasot datus, starp tiem darbojas loģiskais UN operators, šie kritēriji tiek izpildīti vienlaicīgi. Ja kritēriji atrodas dažādās nodaļās (Or nodaļas var būt vairākas), starp tām darbojas loģiskais VAI operators, tiek izpildīts vismaz vienas nodaļas kritērijs.

Netriviāli filtri (Advanced).

Ja ar šīm iespējām tomēr ir maz un ir nepieciešams sastādīt sarežģītākus datu atlasas nosacījumus, ir jālieto "lietpratēju" filtrs – advanced filter **Records/Filter/Advanced Filter Sort...** - . Klikšķinot uz attiecīgās ikonas vai

izpildot komandu Records/Filter /Advanced Fiter/ Sort..., tiek atvērts filtra veidošanas logs ar tabulas attēlu loga augšējā daļā.



15.7.att. *Advanced* filtra veidošanas logs.

Vispirms izdzēs iepriekšējās atlasē kritērijus, ja tādi ir. Filtra veidošanas loga apakšējā daļā atrodas QBE režģis (Query by Example grid), kurā jānorāda lauki, kuriem tiks norādīta kārtošanas secība vai atlasē nosacījumi. To dara šādi:

- QBE režģī ieklikšķina Field rindā, lai lauka labajā pusē parādās izvēles saraksta bultiņa. Tad no saraksta izvēlas vajadzīgo lauku;
- lauku no tabulas attēla loga augšējā daļā ar peli pārvelk uz QBE režģi;
- loga augšējā daļā uz vajadzīgā tabulas lauka ar peli izpilda dubultklikšķi.

Nākošajā solī jānorāda, kā tiks kārtoti atlasītie ieraksti. QBE režģa rindā Sort laukā, pēc kura ieraksti tiks kārtoti, jāklikšķina uz izvēles saraksta bultiņas, no saraksta jāizvēlas Ascending vai Descending (vai not sorted, lai atceltu kārtošanu).

Field:	Uzvards/Nosaukums:	Cena
Sort:	Ascending	
Criteria:	"K. Kalniņš"	70000 Or 1
or:	"A. Uzuliņš"	

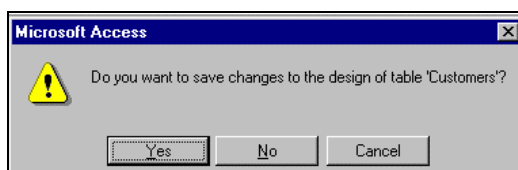
15.8.att. *Advanced* filtra kritēriji.

QBE režģa rindā Criteria jānorāda datu atlasē kritēriji. Starp **kritēriju rindām darbojas loģiskais OR operators**. Sīkāk par kritēriju sastādīšanu runāsim turpmāk *Simbolisko zīmju lietošanā*, nodaļā **13.Vaicājumi**.

Kad filtra nosacījumi pabeigti, filtru var aktivizēt, klikšķinot uz attiecīgās ikonas vai izpildot Filter/Apply Filter/Sort.

Filtra veidošanas logs paliek fonā atvērts. Ieklikšķinot filtra veidošanas logā (vai izpildot *Records/Filter/Advanced Filter/Sort...*), tur būs iepriekš definētie nosacījumi, kurus var papildināt.

Aizverot tabulu, pēc paveiktās datu atlasē MS Access izvada paziņojumu.

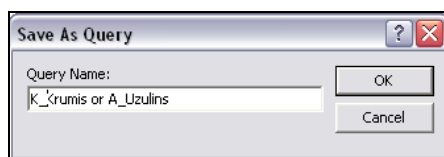


15.9.att. Paziņojums pēc datu atlasē.

Changes to the design – izmaiņas tabulas veidošanas definīcijā šajā gadījumā nozīmē jautājumu, vai jā saglabā pēdējais pielietotais datu filtrs. Atbildot *No*, filtrs netiks saglabāts. Atbildot *Yes*, filtra nosacījumi tiks ierakstīti tabulas īpašībai *Filter*, bet, nākošreiz atverot tabulu, datulapā būs redzami visi ieraksti, ne tikai izfiltrētie.

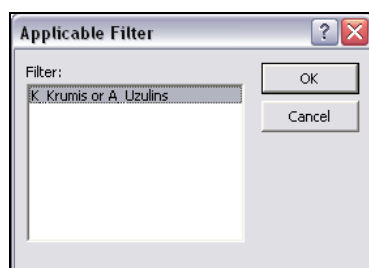
Lai atkārtoti aktivizētu filtru, jāizpilda *Filter/Apply Filter/Sort* vai jāklikšķina uz attiecīgās ikonas.

Lai saglabātu filtru, tad veidošanas skatā jāizpilda komanda *File/Save As Query*.



15.10.att. Filtra saglabāšana.

Šo filtru varēs izsaukt filtra veidošanas skatā, izpildot *File/Load From Query* vai klikšķinot uz ikonas, un no esošo filtru saraksta logā izvēloties vajadzīgo.



15.11.att. Filtra aktivizēšana.

Simboliskās zīmes, veidojot filtrus.

Veidojot filtra nosacījumus, jāzina simbolisko zīmju lietošanas noteikumi.

Simboliskās zīmes (wildcard characters)

15.2.tabula

Simbols	Apraksts	Piemērs
*	Jebkurš simbols (simbolu skaits nav ierobežots)	ja* atrod jauns, jaunrade, jauks u.t.t
?	Viens jebkurš alfabēta simbols	B?ll atrod ball, bell, un bill
[]	Jebkurš viens simbols, no simboliem kas ir iekavās	b[ae]ll atrod ball un bell, bet ne bill
!	Jebkurš viens simbols, kas nav iekavās	b[!ae]ll atrod bill un bull, bet ne ball vai bell
-	Jebkurš viens simbols no simbolu virknes. Var piešķirt (A-Z, bet ne Z-A)	b[a-c]d atrod bad, bbd, un bcd
#	Jebkurš viens skaitliskais simbols	1#3 atrod 103, 113, 123

Izteiksmes elementi var būt saistīti, izmantojot operatorus:

- aritmētiskie: *, +, -, /, ^
- salīdzināšanas: <, <=, >, >=, =, <>
- loģiskie: And (un), Not (nav), Or (vai)

Like – lai izmatotu aizvietošanas loģiku izteiksmēs:

Like "E*"

In – lai noteiktu, vai atbilst datu elements vērtību sarakstam:

In ("Exotic Liquids"; "Mayumi's"; "Tokyo Traders")

Between...And – lai izfiltrētu, noteiktam intervālam atbilstošas vērtības:

Between 10 And 30

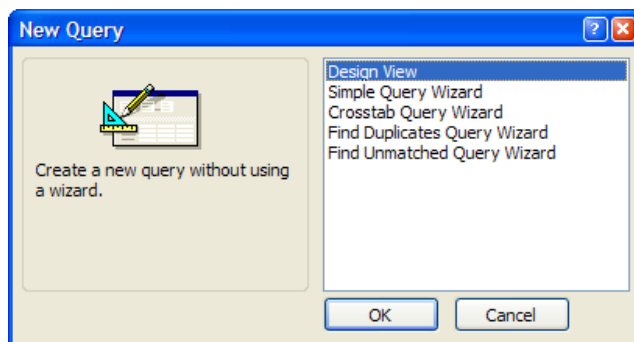
Par filtru izteiksmēm var būt konkrētas vērtības, kuras Access izmanto salīdzināšanai tādā veidā, kā tās tika ievadītas:

- Skaitli tiek ievadīti bez ierobežojumiem, piem., 22.
- Tekstam jābūt ievadītam pēdīgās, piem., "Kalniņš".
- Datumus ierobežo ar # simboliem, piem., #10/10/05#.

16. Vaicājumi

Vaicājumi (Query) ir otra svarīgākā objektu klase pēc tabulām, tas ir līdzeklis datu atlasei no vienas vai vairākām tabulām, datu apskatīšanai dažādos griezumos, to kombinēšanai un summēšanai.

Visa darbība ar vaicājumiem ir līdzīga kā ar tabulām, vismaz tanī aspektā, kas attiecas par pogām – tāpat trīs pogas. Kamēr nav neviena objekta, pieejama ir tikai viena poga *New*.



16.1.att. New Query logs.

Pēc šīs pogas nospiešanas atveras logs New Query, tas pilda tās pašas funkcijas, kādas bija jaunas tabulas veidošanas gaitā, tas ir, jāizvēlas, kādā veidā tiks veidots jaunais vaicājums. Piedāvātie veidi nedaudz atšķiras no tabulas veidošanas režīmiem, bet lielos vilcienos ir saglabātas tās pašas tendences; ir gan lietotājam pilnīgi neatkarīgi veidošanas režīmi un ir arī dažādas sarežģītības pakāpes palīgi (Wizard). Piedāvātās iespējas ir šādas:

- **Design View** – tiek veidots vaicājums bez jebkāda Wizarda palīdzības;
- **Simple Query Wizard** – šis Wizards palīdz izveidot vaicājumu no laukiem, kurus lietotājs ir iezīmējis,
- **Crosstab Query Wizard** – šis Wizards palīdz izveidot vaicājumu (Crosstab Query), kas attēlo datus kompaktā datulapas (kā Excel tabula) formātā,
- **Find Duplicates Query Wizard** – šis Wizards palīdz izveidot vaicājumu, kas atrod ierakstus ar identiskām vērtībām tabulās vai vaicājumos,
- **Find Unmached Query Wizard** - šis Wizards palīdz izveidot vaicājumu, kas atrod ierakstus (rindiņas), kuras nesatur saistītās tabulas.

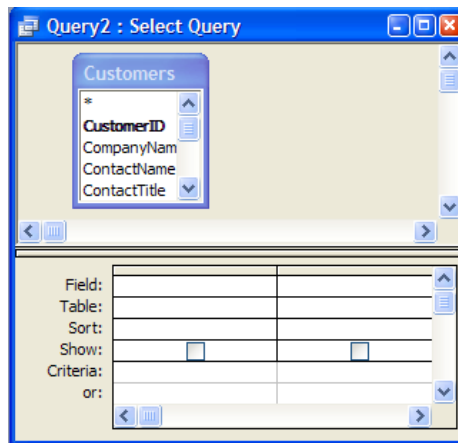
Dažādās situācijās visi šie palīgu veidi var būt noderīgi, tos var izmēģināt, bet tie ir diezgan vienveidīgi.

16.1. Vaicājuma veidošana

Pastāv vairāku veidu vaicājumi. Vaicājumu apskatu jāsāk ar datu atlasē (**select**) tipa vaicājumiem, kas atlasa, šķiro, kārtē, summē un veic citas operācijas ar datiem no vienas vai vairākām tabulām, neko nemainot pašā datu avotā – tabulā.

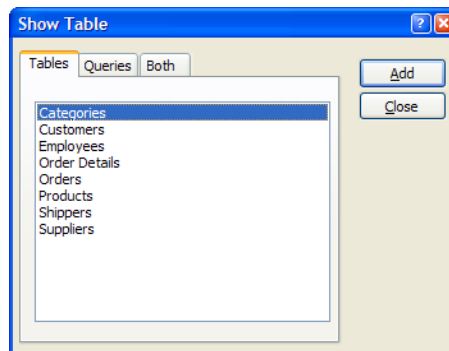
Vaicājumu veidošana ir līdzīga tabulas filtru veidošanai. Jaunu vaicājumu var sākt veidot šādi (piemērā izmantota standarta datu bāze *Northwind.mdb*):

- Datu bāzes loga Table nodaļā iezīmē tabulu *Customers*, uz kuras pamata vaicājums tiks veidots, tad izpilda **Insert/Query** vai noklikšķina attiecīgo ikonu, un atveras *New Query* logs, kurā var izvēlēties vaicājuma veidošanu ar veidotāja *Wizard* palīdzību vai veidot vaicājumu patstāvīgi. Šajā gadījumā jāatstāj iezīmēta izvēle *Design View* un jānospiež OK poga. Atveras vaicājuma veidošanas skata *Design View* logs:



16.2.att. *Design View* logs.

- Datu bāzes loga *Query* nodaļā, noklikšķinot pogu *New*, atveras *New Query* logs. *New Query* logā jāatstāj iezīmēta izvēle *Design View* un jānospiež *OK* poga. Atveras vaicājuma veidošanas skata *Design View* logs, kam virspusē atrodas *Show Table* logs:

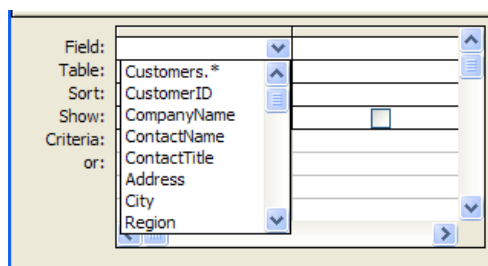


16.3.att. *Show Table* logs.

Show Table logā iezīmē tabulu *Customers*, tad nospiež *Add* pogu, tabulas attēlu novieto vaicājuma veidošanas loga augšējā daļā. Līdzīgi var pievienot arī citas tabulas un vaicājumus.

16.2. QBE režģis

Vaicājuma veidošanas loga apakšējā daļā atrodas QBE režģis (*Query by Example grid*), kurā jānorāda lauki, kādi tiks atlasīti vaicājumā. To veic šādi: QBE režģī ieklikšķina *Field* rindu, lai lauka labajā pusē parādītos izvēles saraksta bultiņa un no saraksta izvēlas vajadzīgo lauku. Zvaigznīte “*” vai “*Customers.**” nozīmē “tabulas *Customers* visi lauki”;



16.4. att. Lauka izvēles logs.

Vaicājumā noteikti jāiekļauj lauki:

- kuru datiem ir jāparādās datulapā;
- kuriem pielietos datu atlases kritērijus.

Vaicājumā var izveidot aprēķināto lauku (*calculated field*), kura vērtība tiek aprēķināta, veicot matemātiskās darbības ar vienu vai vairākiem tabulas laukiem. Aprēķinātais lauks var parādīt arī kāda tabulas lauka daļu vai kombināciju ar citiem laukiem, piemēram, apvienot laukus *Vārds* un *Uzvārds* vienā laukā.

Sākotnēji vaicājumam izvēlas laukus *CustomerID*, *CompanyName*, *City*, *Country*; viena no vaicājuma priekšrocībām ir iespēja izvēlēties, kurus laukus parādīt.

Nākamajā solī jānorāda, pēc kuriem laukiem kārtos atlasītos ierakstus: QBE režģa rindā *Sort* attiecīgajiem laukiem jānoklikšķina uz izvēles saraksta bultiņas, no saraksta jāizvēlas *Ascending* vai *Descending* (vai *not sorted*, lai atceltu kārtotāšanu).

Ir iespējams, ka kāda lauka ierakstu atlasei ir jānorāda kritēriji vai kārtotāšana, taču datulapā šim laukam nav jāparādās. QBE režģa rindā *Show* tiek norādīts, vai lauka datiem ir jābūt redzamiem datulapā. Ieklikšķinot *Show* rindā, rāmītī, parādās vai pazūd ķeksītis, ar to norāda, vai lauka datiem jābūt redzamiem datulapā.

Sort:		
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		

16.5.att. Rinda *Show*.

Tālāk QBE režģa rindā Criteria var norādīt kritērijus, pēc kuriem atlasīs datus.

Field:	Country	City	CompanyName	CustomerID
Table:	Customers	Customers	Customers	Customers
Sort:	Ascending	Ascending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	"Germany"			
or:	"France"			

16.6.att. Vaicājums veidošanas skatā.


Piemērā gan “*Germany*”, gan “*France*” ir teksta vērtības, tāpēc tās jāliek pēdiņās. Ja rakstot pēdiņas netiks ieliktas, *MS Access* pats interpretēs šīs vērtības kā tekstu un ieliks pēdiņas.

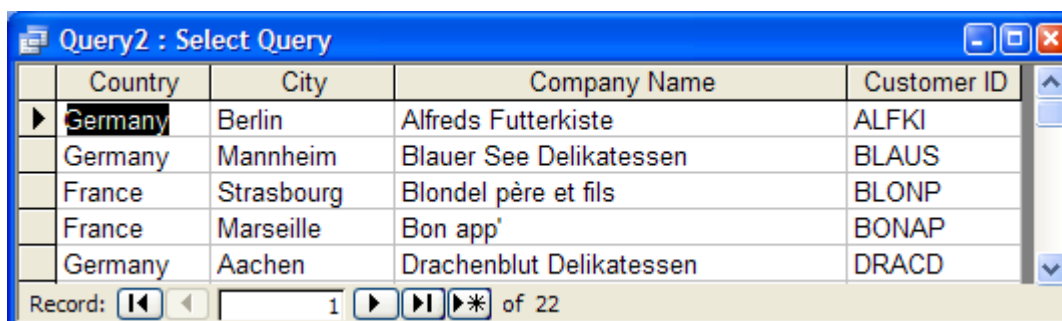
Laukam *Country* kritēriju pieraksta šādi: pirmajā kritēriju rindā – “*Germany*”, otrajā kritēriju rindā – “*France*”, jo starp kritēriju rindām darbojas loģiskais *OR* operators. Taču tikpat labi varēja rakstīt “*Germany*” un “*France*” vienā rindā, pa vidu liekot *OR* operatoru.

Field:	Country	City
Table:	Customers	CU
Sort:	Ascending	As
Show:	<input checked="" type="checkbox"/>	
Criteria:	"Germany" Or "France"	
or:		

16.7.att. Loģiskā operatora *OR* lietojums.


16.3. Vaicājuma skati

Lai apskatītu vaicājuma atlasītos datus, izpilda *View/Datasheet View* vai klikšķina uz attiecīgās ikonas .

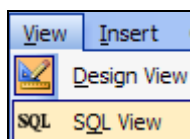


Country	City	Company Name	Customer ID
Germany	Berlin	Alfreds Futterkiste	ALFKI
Germany	Mannheim	Blauer See Delikatessen	BLAUS
France	Strasbourg	Blondel père et fils	BLONP
France	Marseille	Bon app'	BONAP
Germany	Aachen	Drachenblut Delikatessen	DRACD

16.8.att. Atlases logs.

Lai atgrieztos veidošanas skatā, izpilda *View/Design View* vai klikšķina uz attiecīgās ikonas .

Vaicājumiem skatu (*View*) ikonu grupā ir arī trešā ikona *SQL View*. Nospiežot to vai izpildot *View/SQL View*, iegūst logu, kur vaicājuma definīcija ir rakstīta *SQL* valodā.



16.9.att. SQL skata aktivizēšana.

Zinot šo valodu, vaicājuma definīciju varētu rakstīt šeit. Tomēr, tieši tā ir *MS Access* priekšrocība, ātrāk un ērtāk ir sagatavot vaicājumu veidošanas skatā *Design View*.

16.4. Kritēriju pielietojuma piemēri

Lai iepazītos, kā notiek datu atlase atkarībā no kritērijiem, apskatīsim vairākus kritēriju veidošanas piemērus.

1. Atgriežoties veidošanas skatā un papildinot vaicājuma nosacījumus: laukā *City* atlasīsim vērtības, kas sākas ar burtu *L*. Ierakstus tiks atlasīti, ievērojot abus kritērijus, jo starp dažādu lauku kritērijiem, kas rakstīti vienā rindā, darbojas loģiskais *AND* operators. Lai norādītu, ka pēc burta *L* var sekot jebkuras citas zīmes, pēc *L* jāraksta simbols *** – *L**. *MS Access* pats to pārveidos “savā valodā”,

pievienos operatoru **Like** un ieliks pēdiņas, tikko kursorš pārvietosies uz citu rindu vai lauku.

Field:	Country	City
Table:	Customers	Customers
Sort:		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:	"Germany" Or "France"	Like "L**"
or:		

16.10.att. SQL skata aktivizēšana.

Pārejot uz datulapu, *MS Access* parāda atlasītos ierakstus:

Query2 : Select Query				
	Country	City	Company Name	Customer ID
►	France	Lille	Folies gourmandes	FOLIG
	Germany	Leipzig	Morgenstern Gesundkost	MORGK
	France	Lyon	Victuailles en stock	VICTE
*				

Record: 1 of 3

16.11.att. Atlasītie ieraksti.

2. Atgriezīsimies veidošanas skatā un izdzēsīsim kritēriju laukam *City*. Atlasīsim ierakstus, kuriem firmu nosaukumi sākas ar burtiem A – D: lauka *CompanyName* kritēriju uzrakstīsim <E. *MS Access* atlasīs ierakstus:

Query2 : Select Query				
	Country	City	Company Name	Customer ID
►	Germany	Berlin	Alfreds Futterkiste	ALFKI
	Germany	Mannheim	Blauer See Delikatessen	BLAUS
	France	Strasbourg	Blondel père et fils	BLONP
	France	Marseille	Bon app'	BONAP
	Germany	Aachen	Drachenblut Delikatessen	DRACD
	France	Nantes	Du monde entier	DUMON
	Germany	Stuttgart	Die Wandernde Kuh	WANDK
*				

Record: 1 of 7

16.12.att. Atlasītie ieraksti, kuriem firmu nosaukumi sākas ar burtiem A – D .

3. Atgriezīsimies veidošanas skatā un izdzēsīsim visus kritērijus. Noskaidrosim, kuriem klientiem nav atzīmēts faksa numurs: lauka *CustomerID* vietā režģī ievietosim lauku Fax un tam noteiksim kritēriju *Null*, *MS Access* apzīmējumu tukšam laukam. Pārvietojot kursoru uz citu rindu vai lauku, *MS Access* kritēriju pārveidos *IsNull*. Pārejot uz datulapas skatu, varam aplūkot atlasītos ierakstus:

Country	City	Company Name	Fax
Mexico	México D.F.	Antonio Moreno Taquería	
UK	London	B's Beverages	
Switzerland	Bern	Chop-suey Chinese	
Brazil	São Paulo	Comércio Mineiro	
Brazil	São Paulo	Familia Arquibaldo	
Sweden	Bräcke	Folk och få HB	
Spain	Sevilla	Godos Cocina Típica	

16.13.att. Atlasītie ieraksti, kuriem nav atzīmēts faksa numurs.

4. Atgriezīsimies veidošanas skatā un pievienosim vaicājumam tabulu *Orders*. Izpildām *Query/Show Table* vai klikšķinām uz attiecīgās ikonas, atveras *Show Table logs*, kurā izpildām dubultklikšķi uz tabulas nosaukuma *Orders*. Tabulas *Orders* attēls novietojas veidošanas loga augšējā daļā blakus *Customers* tabulai, un starp tabulām *MS Access* automātiski novelk saiti. *QBE* režģī aizpildām šādi:

Field:	Country	City	OrderDate
Table:	Customers	Customers	Orders
Sort:			
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			>22/12/96 AND <1/1/97
or:			

16.14.att. Tabulas pievienošana vaicājumam.

Tātad, atlasīsim ierakstus par pasūtījumiem, kuri ir veikti 1996. gadā, laikā starp Ziemassvētkiem un Jauno gadu. Pārvietojot kursoru uz citu rindu vai lauku, *MS Access* lauka *OrderDate* kritērija datumus ievietos datumu simbolu # vidū:

OrderDate
Orders
<input checked="" type="checkbox"/>
>#1996.12.22.# And <#1997.01.01.#

16.15.att. Datumu lietošana vaicājumos.

Pēc šī kritērija atlasītie ieraksti:

Company Name	Country	City	Order Date
Ernst Handel	Austria	Graz	23-Dec-1996
Drachenblut Delikatessen	Germany	Aachen	23-Dec-1996
Piccolo und mehr	Austria	Salzburg	24-Dec-1996
Save-a-lot Markets	USA	Boise	25-Dec-1996
Hungry Coyote Import Store	USA	Elgin	25-Dec-1996
HILARIÓN-Abastos	Venezuela	San Cristóbal	26-Dec-1996
Frankenversand	Germany	München	27-Dec-1996
Princesa Isabel Vinhos	Portugal	Lisboa	27-Dec-1996
Save-a-lot Markets	USA	Boise	30-Dec-1996
Vaffeljernet	Denmark	Århus	31-Dec-1996

16.16.att. Atlases rezultāts datumu lietošanas vaicājumos.

- Atgriezīsimies veidošanas skatā un pārveidosim lauka *OrderDate* kritēriju. Noskaidrosim, kuri klienti ir veikuši pasūtījumus dienu pirms Ziemassvētkiem. Lauka *OrderDate* kritēriju rakstām *.12.23. , jo mums derēs jebkurš gads. *MS Access* kritēriju pārveidos *Like* "*.12.23.". Pārejot uz datulapas skatu, atlasīsies ieraksti:

Company Name	Country	City	Order Date
Ernst Handel	Austria	Graz	23-Dec-1996
Drachenblut Delikatessen	Germany	Aachen	23-Dec-1996
Frankenversand	Germany	München	23-Dec-1997
Wolski Zajazd	Poland	Warszawa	23-Dec-1997

16.17.att. Datumu lietošana vaicājumos, izmantojot *Like*.

Piezīme: Darbā ar datumiem ir jābūt uzmanīgiem, izmantojot datuma atdalītāju (simbolu) – punktu (.), slīpsvītru (/) vai restīti (#). Atdalītājs ir atkarīgs no reģionālajiem uzstādījumiem jūsu datorā. (*Control Panel/Regional and Language Options*).

16.5. Aprēķinātie lauki – *calculated field*

Izveidosim arī aprēķināto lauku – *calculated field*. Aprēķinātā lauka vērtību iegūst, veicot matemātiskas darbības vai kombinējot vienas vai vairāku tabulas lauku vērtības.

Šoreiz aprēķinātā lauka vērtību iegūsim, lauka *Country* vērtībai pievienojot lauka *City* vērtību un starp tām ievietojot komatu.

Lai atsauktos uz lauka vērtību, lauka nosaukums ir jāliek kvadrātiekvāds. Ja vaicājumā vairākās tabulās ir vienāda nosaukuma lauki, tad noteikti ir jānorāda arī tabulas nosaukums, piemēram, [*Customers*].[*City*].

Laukus un tekstu “, “ saista apvienojuma zīme &:

Field:	Company Name	[Country] & ", " & [City]	OrderDate
Table:	Customers		Orders
Sort:		Ascending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			Like "23/12/"
or:			

16.18.att. Apvienojuma zīmes & lietošana.

Kombinētajam laukam *MS Access* dod nosaukumu *Expr1*, un tas parādās datulapā:

Query1 : Select Query			
	Company Name	Expr1	Order Date
▶	Blondel père et fils	France, Strasbourg	23-Dec-94
	QUICK-Stop	Germany, Cunewalde	23-Dec-94
*			
Record: 1 of 2			

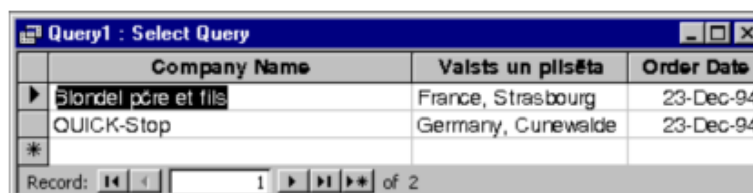
16.19.att. Kombinētais lauks.

Ja jūs gribat citu nosaukumu, piemēram, *Valsts* un *pilsēta*, šis nosaukums vienkārši jāieraksta *Expr1* vietā:

Field:	Firmas nosaukums: CompanyName	Valsts un pilsēta: [Country] & ", " & [City]	Datums: OrderDate
Table:	Customers		Orders
Sort:		Ascending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			Like "23/12/"
or:			

16.20.att. Kombinēta lauka nosaukuma maiņa.

Citu lauku, kas ir kādā no pieprasījuma tabulām, kolonnu nosaukumus datulapā var mainīt tikai tad, ja tabulās šo lauku *Caption* īpašība ir atstāta tukša. Šajā piemērā *QBE* režģī datulapas kolonnu nosaukums ir piešķirts gan laukam *CompanyName*, gan laukam *OrderDate*.



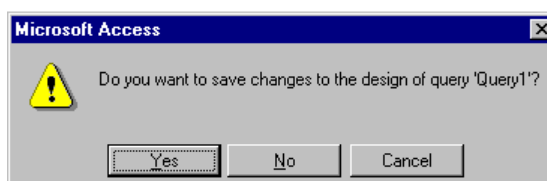
Company Name	Vaists un plisēta	Order Date
Blondel père et fils	France, Strasbourg	23-Dec-94
QUICK-Stop	Germany, Cunevalde	23-Dec-94

16.21.att. Kolonnas nosaukuma maiņa, *Caption* īpašība.

Tomēr datulapā tie neparādās, jo tabulās šiem laukiem jau ir definēta *Caption* īpašība.

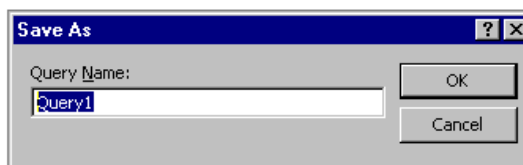
16.6. Vaicājuma saglabāšana

Izveidoto vaicājumu saglabā izpildot *File/Save*, vai vienkārši aizverot vaicājuma logu. *MS Access* pats piedāvā to izdarīt:



16.22.att. Vaicājuma saglabāšanas logs.

Atbildot *No*, vaicājums saglabāts netiks. Atbildot *Yes*, tiks atvērts *Save As* logs:



16.23.att. *Save As* logs.

Ievadot nosaukumu (*Query Name*) un nospiežot *OK*, vaicājums tiks saglabāts.

16.7. Parametru vaicājumi – *parameter queries*

Lai jau izveidotam vaicājumam izmainītu kritērijus, vaicājums jāatver veidošanas skatā. Pēc tam, kad kritēriji ir izmainīti, var pāriet uz datulapas skatu un apskatīt datus.

Vaicājumiem, kas bieži tiek izmantoti datu caurskatīšanai, var vienkāršot kritēriju izmainīšanu, izveidojot parametru vaicājumu. Atverot parametru vaicājumu kā datulapu, *MS Access* dialoga logā pieprasa ievadīt kritērijus. Tādējādi nav nepieciešams katru reizi pāriet uz *Design* skatu un mainīt nosacījumus.

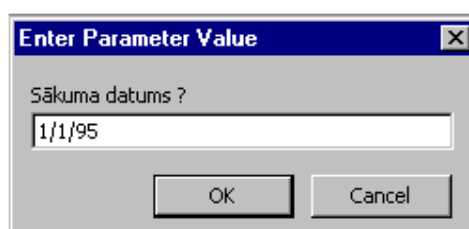
Nākamajā piemērā izveidots vaicājums, kuram norādot sākuma un beigu datumu, *MS Access* atlasa šī intervāla pasūtījumus.

Lauka *Datums* kritēriju vietā kvadrātiekvāds jāieraksta teksts, kuram kā jautājumam jāparādās, aktivizējot vaicājumu. Šajā piemērā izveidoti divi parametri:

Datums: OrderDate
Orders
<input checked="" type="checkbox"/>
>=[Sākuma datums ?] And <=[Beigu datums ?]

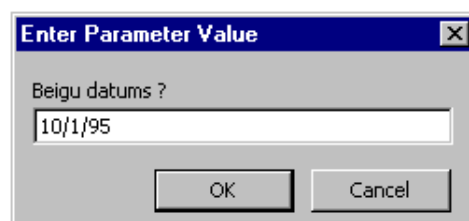
16.24.att. Parametru definēšana.

Parasti ar to pietiek, un *MS Access* spēj parametru datu tipu noteikt pats. Atvērot vaicājumu kā datulapu, *MS Access* pieprasa pirmo parametru:



16.25.att. Pirmā parametra ievads.

Kad sākuma datums ir ievadīts un ar *OK* apstiprināts, tad tiek pieprasīts otrs parametrs:



16.26.att. Otrā parametra ievads.

Kad ievadīts un apstiprināts arī beigu datums, *MS Access* atver datulapu:

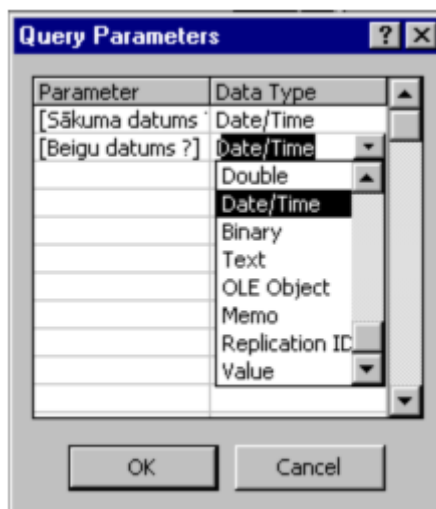


Firmas nosaukums	Valsts un pilsēta	Order Date
Queen Cozinha	Brazil, São Paulo	04-Jan-95
Mère Paillarde	Canada, Montréal	09-Jan-95
La maison d'Asie	France, Toulouse	03-Jan-95
Hungry Owl All-Night Grocers	Ireland, Cork	05-Jan-95
Wolski Zajazd	Poland, Warszawa	05-Jan-95
Folk och få HB	Sweden, Bräcke	10-Jan-95
Chop-suey Chinese	Switzerland, Bern	03-Jan-95
Seven Seas Imports	UK, London	09-Jan-95
Hungry Coyote Import Store	USA, Elgin	06-Jan-95
Split Rail Beer & Ale	USA, Lander	02-Jan-95

16.27.att. Parametru vaicājuma rezultāts.

Gadījumā, ja *MS Access* pats nespēj noteikt parametru datu tipu un ziņo par kļūdu, jārikojas šādi:

- izpildot *Query/Parameters*, jāatver *Query Parameters* logs;
- parametru logā jāieraksta precīzi tāds pats teksts kā kritēriju rindā, ieteicams lietot kopēšanu;
- jānorāda katra parametra datu tips:

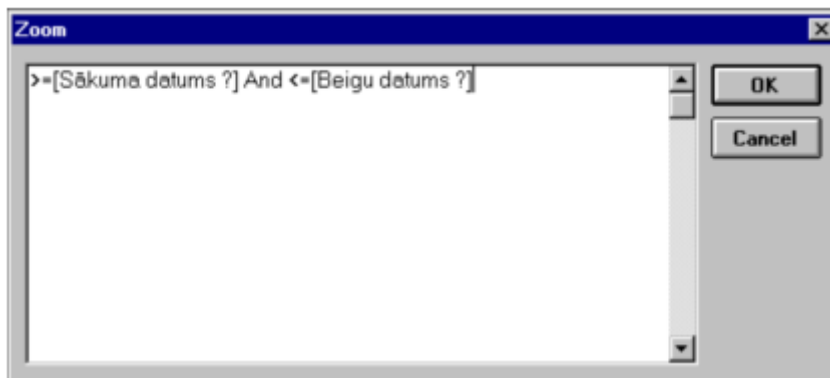


16.28.att. Vaicājuma parametru logs.

Pēc tam var aizvērt parametru logu un atvērt vaicājumu kā datulapu, lai to saglabātu.

16.8. Zoom logs

Ja, piemēram, kritērijs ir tik garš, ka kritēriju rindā to ir grūti labot, ar taustiņu kombināciju *Shift + F2* var atvērt palielinājuma (*Zoom*) logu, kas ir paredzēts darbam ar garām izteiksmēm, un to var izmantot *MS Access* vidē:



16.29.att. Zoom logs.

Kad izteiksme ir izlabota, *Zoom* logu atkal var aizvērt.

16.9. Saites starp vaicājuma tabulām

Izveidojot jaunu vaicājumu: vaicājuma veidošanas loga augšējā daļā novieto tabulas *Customers* un *Orders*. Saites starp šīm tabulām *MS Access* novelk automātiski.

Ja vaicājuma veidošanas loga augšējā daļā *MS Access* pats novelk līniju starp tabulām, tad tās jau ir saistītas. Pretējā gadījumā saites jāizveido pašam lietotājam. Tabulas ir jāsaista, lai datulapa saturētu savstarpēji saistītus ierakstus.

Ja tabulas netiek saistītas, *MS Access* izveido datulapu, kas satur visu iespējamo ierakstu kombināciju. Ja, piemēram, ir 2 tabulas, vienā ir 5 ieraksti, otrā ir 20 ieraksti, tad tiks izveidota datulapa ar $5 \times 20 = 100$ ierakstiem.

MS Access pats novelk saiti:

- ja starp tabulām ir definēta saite jeb relācija;
- starp tabulu laukiem, kuriem ir tas pats lauku nosaukums un datu tips.

Saišu līniju izskats norāda, vai savienojums nodrošina referenciālo integritāti (to nodrošina saite starp šīm tabulām) un savienojuma veidu.

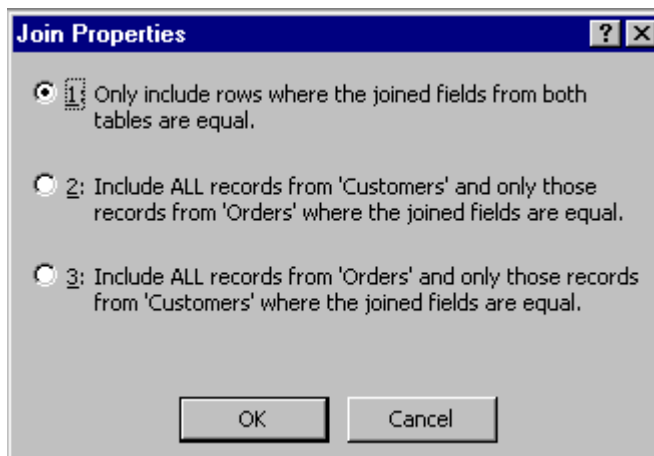
16.10. Vaicājums ar viena virziena saiti – *outer join*

Iepriekšējos vaicājumos darbojās līdzvērtīgas saites – *equ-joins*, tagad apskatīsim viena virziena saiti – *outer join*.

Vaicājuma uzdevums būs noteikt, vai tabulā *Customers* ir kāds klients, kurš nav izdarījis nevienu pasūtījumu.

Vaicājumā būs vajadzīgs tabulas *Customers* lauks *CompanyName* un tabulas *Orders* lauks *OrderDate*, šos laukus novieto *QBE* režģī. Laukam *CompanyName* norāda kārtotāšanu *Ascending*.

Izpilda dubultklikšķi uz saites starp tabulām *Customers* un *Orders*, lai tiek atvērts saišu īpašību – *Join Properties* – logs:

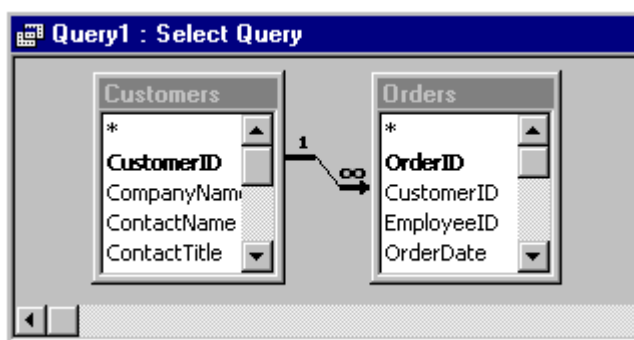


16.30.att. Saišu uzstādījumu logs.

Izvēle 1 saišu īpašību logā vienmēr ir līdzvērtīgā saite: vaicājumā tiek iekļauti ieraksti, kur vienas tabulas ierakstam atbilst kāds ieraksts otrā tabulā.

Izvēle 2 un 3 nozīmē, ka noteikti tiek parādīti visi vienas vai otras tabulas ieraksti neatkarīgi no tā, vai pretējā tabulā ir atbilstoši ieraksti.

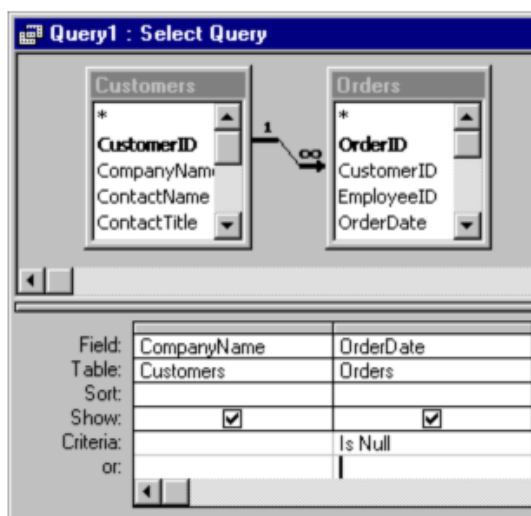
Lai tiktu parādīta 2. izvēle, jāatzīmē ir visi tabulas *Customers* ieraksti. Ar *OK* aizverot saišu īpašību logu, saites līnija ir ieguvusi uz tabulu *Orders* vērstu bultiņu:



16.31.att. Saites izveide.

Uzreiz pārejot uz datulapas skatu un pārbaudot atlasītos ierakstus, būs grūti atrast vajadzīgos ierakstus, jo datulapa saturēs visu klientu pasūtījumus. Šajā gadījumā tie būs 832 ieraksti, kuri būs jāpāršķirsta, kamēr tiks atrasts, ka 198. ierakstā parādās klients bez neviena pasūtījuma – lauks *OrderDate* ir tukšs.

Tāpēc laukam *OrderDate* jāuzraksta kritēriju *Null*. Pārvietojot kursoru uz citu rindu vai lauku, *MS Access* kritēriju pārveidos par *IsNull*.



16.32.att. Vaicājuma izveide, izmantojot saites.

Pārejot uz datulapas skatu, *MS Access* parādīs tikai vajadzīgos ierakstus:



16.33.att. Vaicājuma rezultāts, izmantojot saites.

Pēc visām veiktajām izmaiņām, vaicājumu var saglabāt.

16.11. Summēšanas vaicājumi – *totals queries*

Nākošā iespēja, ko var veikt ar vaicājumu, ir atlasīto datu summēšana, vidējās vērtības noteikšana, maksimālās vai minimālās vērtības atlasīšana.

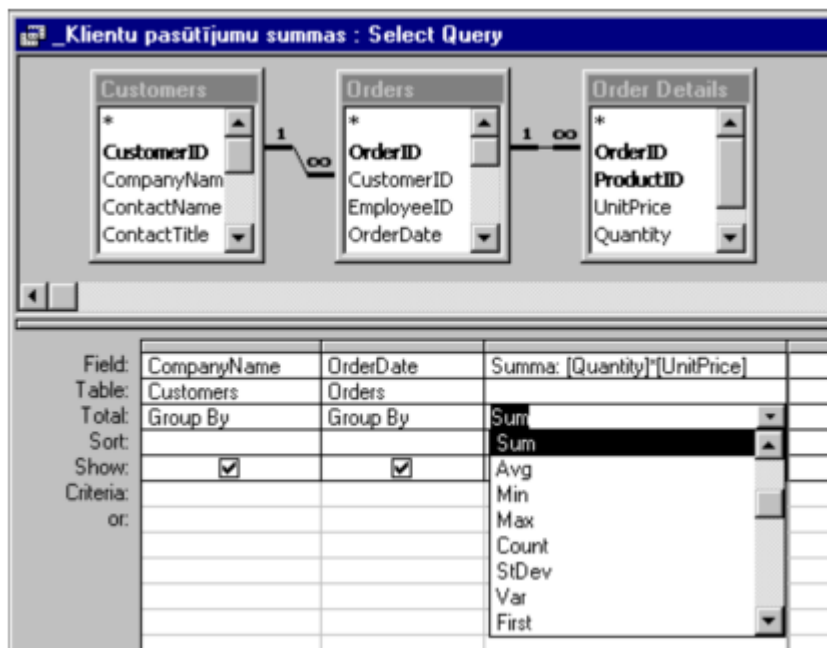
Lai izveidotu, vaicājuma veidošanas loga augšējā daļā novieto tabulas, piem., *Customers*, *Orders* un *Order Details*. Saites starp šīm tabulām *MS Access* novelk automātiski.

Vaicājuma uzdevums būs noteikt katra pasūtījuma summu, kas sastāv no atsevišķu preču cenas un to daudzuma.

Loga apakšējā daļā – *QBE* režģī – novieto lauku *CompanyName* no tabulas *Customers* un lauku *OrderDate* no tabulas *Orders*. Šiem laukiem nosaka arī kārtotāņu alfabēta secībā – *Ascending*. Tad izveido aprēķina lauku no tabulas *Order Details* laukiem *UnitPrice* un *Quantity*, un nosauc šo aprēķina lauku *Summa*.

Lai piešķirtu vaicājumam summēšanas iespējas, izpilda *View/Totals* vai klikšķina uz attiecīgās ikonas.

QBE režīmā parādās jauna rinda ar virsrakstu *Total*. Šajā rindā jānorāda, pēc kuriem laukiem dati tiks grupēti un kuri lauki tiks izmantoti aritmētiskajās operācijās. Rindai noteikti jābūt aizpildītai, nedrīkst būt lauki, kuriem *Total* rindā ir tukšums. Laukiem *CompanyName* un *OrderDate* jāatstāj *MS Access* noklusēto vērtību *Group By*. Laukam *Summa* rindā *Total* norāda *Sum*.



16.34.att. *Total* lauka aizpildīšana.

***Total* izvēles.**

16.1.tabula

Group By	Grupē pēc lauka.
Sum	Summē lauka datus. Ir spēkā šādiem lauku tipiem: <i>Number</i> , <i>Date/Time</i> , <i>Currency</i> , <i>AutoNumber</i> , <i>Yes/No</i> .
Avg	Atrrod lauka vidējo vērtību. Ir spēkā šādiem lauku tipiem: <i>Number</i> , <i>Date/Time</i> , <i>Currency</i> , <i>AutoNumber</i> , <i>Yes/No</i> .
Min	Atrrod lauka minimālo vērtību.
Max	Atrrod lauka maksimālo vērtību.
Count	Saskaita, cik ierakstiem ir lauka vērtības. Tukšie ieraksti ar <i>Null</i> vērtību netiek skaitīti.
StDev	Atrrod lauka vērtību standarta novirzi. Ir spēkā šādiem lauku tipiem: <i>Number</i> , <i>Date/Time</i> , <i>Currency</i> , <i>AutoNumber</i> , <i>Yes/No</i> .
Var	Atrrod lauka vērtību dispersiju. Ir spēkā šādiem lauku tipiem: <i>Number</i> , <i>Date/Time</i> , <i>Currency</i> , <i>AutoNumber</i> , <i>Yes/No</i> .

First	Atrod pirmo lauka vērtību.
Last	Atrod pēdējo lauka vērtību.
Expression	Izveido aprēķinātu lauku, kur tiek izmantotas vairākas izteiksmes vai funkcijas.
Where	Norāda tā lauka kritērijus, kas netiek izmantots grupēšanā. Šis lauks datulapā būs neredzams – <i>MS Access</i> rindā <i>Show</i> krustiņš būs izdzēsts.

Atverot vaicājumu kā datulapu:

Company Name	Order Date	Summa
Alfreds Futterkiste	25-Sep-95	\$ 1 086.00
Alfreds Futterkiste	03-Nov-95	\$ 878.00
Alfreds Futterkiste	13-Nov-95	\$ 330.00
Alfreds Futterkiste	15-Feb-96	\$ 851.00
Alfreds Futterkiste	15-Apr-96	\$ 491.20
Alfreds Futterkiste	09-Mai-96	\$ 980.00
Ana Trujillo Emparedados y helados	19-Okt-94	\$ 88.80
Ana Trujillo Emparedados y helados	08-Sep-95	\$ 479.75
Ana Trujillo Emparedados y helados	29-Dec-95	\$ 320.00

16.35.att. *Total* lauka aizpildīšana.

Aizverot vaicājumu (saglabājot to) un tad atkal atverot veidošanas skatā, var konstatēt, ka *QBE* režģī *MS Access* ir izmainījis lauka *Summa* definīciju: rindā *Total* tagad ir *Expression*, bet *Sum* kā funkcija ir novietota *Field* rindā.

Dažreiz ir nepieciešams pārveidot vaicājumu tā, lai summēšana notiktu nevis katram pasūtījumam, bet tiktu summēti pasūtījumi gada laikā. Tabulā tiek glabāts pasūtījuma datums. Lai no datuma iegūtu pasūtījuma gadu, jāizmanto funkcija *Year()*.

Lai noteiktu ne tikai pasūtījumu kopējo summu gada laikā, bet arī maksimālo viena pasūtījuma summu, *QBE* režģis ir jāaizpilda šādi:

Field:	CompanyName	Gads: Year([OrderDate])	Summa: Sum([Quantity]*[UnitPrice])	Max: [Quantity]*[UnitPrice]
Table:	Customers			
Total:	Group By	Group By	Expression	Max
Sort:	Ascending	Ascending		
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:				

16.36.att. *Total* lauka aizpildīšana.

Atverot vaicājumu kā datulapu:

Company Name	Gads	Summa	Max
Alfreds Futterkiste	1995	\$ 2 294.00	\$ 878.00
Alfreds Futterkiste	1996	\$ 2 302.20	\$ 825.00
Ana Trujillo Emparedados y helados	1994	\$ 88.80	\$ 60.00
Ana Trujillo Emparedados y helados	1995	\$ 799.75	\$ 340.00
Ana Trujillo Emparedados y helados	1996	\$ 514.40	\$ 348.00
Antonio Moreno Taquería	1994	\$ 403.20	\$ 403.20
Antonio Moreno Taquería	1995	\$ 6 452.15	\$ 1 050.00
Antonio Moreno Taquería	1996	\$ 660.00	\$ 380.00
Around the Horn	1994	\$ 480.00	\$ 390.00

16.37.att. Summēšanas vaicājuma piemērs.

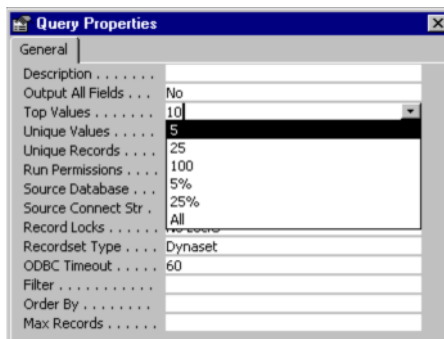
Var saglabāt jau izveidoto vaicājumu ar nosaukumu piem., *_Klientu pasūtījumu summas pa gadiem*.

16.12. Vaicājuma īpašība *Top Values*

Ir iespējams definēt, cik ierakstiem ar noteikta lauka lielākajām vai mazākajām vērtībām ir jāparādās vaicājuma datulapā.

Vaicājumu var pārveidot tā, lai tas sniegtu informāciju par desmit firmas labākajiem klientiem, piemēram, 1995. gadā.

- Lauka *Gads* rindā *Criteria* ieraksta 1995.
- Tagad kārtošana būs nepieciešama nevis pēc laukiem *CompanyName* un *Gads*, bet pēc lauka *Summa*. Šeit jānorāda *Descending* kārtošana.
- Izpildot *View/Properties* vai klikšķinot uz attiecīgās ikonas, atver vaicājuma īpašību – *Query Properties* – logu:



16.38.att. Vaicājuma īpašību *Query Properties* logs.

Ja tiek atvērts nevis *Query Properties*, bet *Field Properties* vai *Field List Properties* logs, tas nozīmē, ka kursora pirms loga atvēršanas atradās *QBE* režģī vai uz tabulu attēliem loga augšējā daļā. Tad vajag noklikšķināt kaut kur vaicājuma logā ārpus pieminētajiem apgabaliem.

Query Properties logā rindā *Top Values* jānorāda vajadzīgais atlasīto ierakstu skaits – 10. Ieklikšķinot šajā rindā un izvēršot *MS Access* piedāvāto vērtību sarakstu, redzams, ka papildus konkrētam ierakstu skaitam ir iespējams norādīt arī ierakstu skaitu no kopējā ierakstu skaita procentos.

Tagad var pāriet uz datulapas skatu. 10 labākie klienti ir:

Klientu pasūtījumu summas pa gadiem - Select Query				
	Company Name	Gads	Summa	Max
▶	Save-a-lot Markets	1995	\$ 66 711.24	\$ 4 456.44
	QUICK-Stop	1995	\$ 61 784.00	\$ 7 905.00
	Ernst Handel	1995	\$ 44 058.58	\$ 3 944.00
	Hungry Owl All-Night Grocers	1995	\$ 27 086.85	\$ 4 850.00
	Mère Paillard	1995	\$ 26 507.10	\$ 10 329.20
	Queen Cozinha	1995	\$ 21 304.60	\$ 8 432.00
	Rattlesnake Canyon Grocery	1995	\$ 19 029.20	\$ 6 324.00
	Berglunds snabbköp	1995	\$ 16 859.10	\$ 3 952.50
	Simons bistro	1995	\$ 13 558.40	\$ 10 540.00
	HILARIÓN-Abastos	1995	\$ 13 481.30	\$ 2 640.00

16.39.att. *Query Properties* īpašības lietošana.

Vaicājumu var saglabāt ar nosaukumu, piem., *10 labākie klienti 1995*.

16.13. Šķērstabulas vaicājumi – *crosstab queries*

Atvērot saglabāto vaicājumu *_Klientu pasūtījumu summas pa gadiem.* (sk. p.12.11.), tas nav ērti pārskatāms. Vaicājumu vajadzētu pārveidot tā, lai visa informācija par firmu aizņemtu vienu rindu, gadu nosaukumi būtu kolonnu virsraksti, bet gadu summas izvietotas šajās kolonnās.

Šķērstabulas vaicājumu – *crosstab query* – izmanto, ja nepieciešams summēt datus rindu un kolonnu formātā. Kāda lauka vai izteiksmes dati šķērstabulas vaicājumā tiek izmantoti kā datulapas kolonnu virsraksti.

Lai *select* tipa vaicājumu pārveidotu par *crosstab* tipa vaicājumu, jāizpilda *Query/Crosstab Query* vai jāklikšķina uz izvērses bultiņas *Query Type* ikonu grupā, kas sastāv no 6 ikonām un norāda vaicājuma tipu.

Kamēr vaicājums ir *select* tipa, nospiešot ikonu ir stāvoklī *Select Query*. Noklikšķinot uz *Crosstab query* ikonas, tiks mainīts vaicājuma tips un, attiecīgi, vaicājuma veidošanas logs.

Vaicājumu pārveidojot par *crosstab*, *QBE* režģī tiek pievieno *Crosstab* rindu, kurā jāizvēlas:

- **Row Heading** – lauka dati tiks izkārtoti kā parasti summēšanas vaicājumos. Atļauts definēt vairākiem laukam.
- **Column Heading** – lauka datu vērtības tiks izmantotas kā datulapas kolonnu virsraksti. Atļauts definēt tikai vienam laukam.
- **Value** – lauka dati tiks izkārtoti kolonnā atbilstoši Column Heading lauka vērtībai. Atļauts definēt tikai vienam laukam.

Vaicājumu veidošanas skatā pasūtījuma maksimālās vērtības vietā noderīgāka būs pasūtījumu kopējā summa pa visiem gadiem:

Field:	CompanyName	Gads: Year([OrderDate])	Summa: Sum([Quantity]*[UnitPrice])	Kopā: [Quantity]*[UnitPrice]
Table:	Customers			
Total:	Group By	Group By	Expression	Sum
Crosstab:	Row Heading	Column Heading	Value	Row Heading
Sort:	Ascending			
Criteria:				

16.40.att. Crosstab query.

Atverot vaicājumu kā datulapu:

Company Name	Kopā	1994	1995	1996
Alfreds Futterkiste	\$ 4 596.20	\$ 2 294.00	\$ 2 302.20	
Ana Trujillo Emparedados y helados	\$ 1 402.95	\$ 68.80	\$ 799.75	\$ 514.40
Antonio Moreno Taquería	\$ 7 515.35	\$ 403.20	\$ 6 452.15	\$ 660.00
Around the Horn	\$ 13 806.50	\$ 480.00	\$ 5 819.90	\$ 7 506.60
B's Beverages	\$ 6 089.90	\$ 479.40	\$ 3 179.50	\$ 2 431.00
Berglunds snabbköp	\$ 26 988.15	\$ 2 102.00	\$ 16 659.10	\$ 8 207.05
Blauer See Delikatessen	\$ 3 239.80		\$ 1 079.80	\$ 2 160.00
Blondel père et fils	\$ 19 088.00	\$ 9 986.20	\$ 8 371.80	\$ 730.00
Bólido Comidas preparadas	\$ 5 297.80	\$ 982.00		\$ 4 315.80
Bon app'	\$ 23 850.95	\$ 4 202.50	\$ 12 462.55	\$ 7 185.90

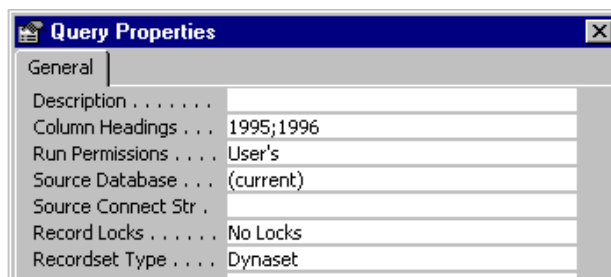
16.41.att. Kopējā summa pa visiem gadiem.

16.14. Kolonnu virsrakstu kārtošana šķērstabulas vaicājumiem

Ja, piemēram, svarīgi ir nevis visu, bet noteiktu gadu dati, var ierobežot datulapas kolonnu skaitu. *MS Access* nodrošina iespēju sakārtot vai ierobežot kolonnu virsrakstu vērtības.

Vaicājumam veidošanas skatā pievieno vēl lauku *Country*, pēc kura grupē vaicājumu, un tam norāda kārtošanu *Ascending*.

Jāizpilda *View/Properties* un *Query Properties* logā *Column Headings* rindā vajadzīgajā secībā jāieraksta kolonnu virsraksti, atdalot tos ar atdalīšanas zīmi, kas noteikta *Windows Control Panel* par *list separator*. Rakstot kolonnu virsrakstus, jāatceras par datu tipu. Ja, piemēram, tie ir *text* tipa, virsraksti jāliek pēdīnās.



16.42.att. *Query Properties* logs.

Atverot vaicājumu kā datulapu:



The image shows a Crosstab Query result titled '_Klientu pasūtījumu summas pa gadiem : Crosstab Query'. The table has columns for Country, Company Name, Kopā, 1995, and 1996. The data is as follows:

Country	Company Name	Kopā	1995	1996
Argentina	Cactus Comidas para llevar	\$ 1 814.80	\$ 225.50	\$ 1 589.30
Argentina	Océano Atlántico Ltda.	\$ 3 460.20	\$ 429.20	\$ 3 031.00
Argentina	Rancho grande	\$ 2 844.10	\$ 1 149.40	\$ 1 694.70
Austria	Ernst Handel	\$ 113 236.68	\$ 44 058.58	\$ 57 182.90
Austria	Piccolo und mehr	\$ 26 259.95	\$ 11 124.60	\$ 4 393.75
Belgium	Maison Dewey	\$ 10 430.58	\$ 2 380.00	\$ 8 050.58
Belgium	Suprêmes délices	\$ 24 704.40	\$ 6 375.10	\$ 11 890.50
Brazil	Comércio Mineiro	\$ 3 810.75	\$ 1 128.00	\$ 513.75
Brazil	Familia Arquibaldo	\$ 4 438.90	\$ 3 510.90	

16.42.att. *Query Properties* lietošana.

Datulapā redzams, ka virsrakstu ierobežojumi nav ņemti vērā laukam *Kopā*, tur joprojām tiek rēķināta visu gadu kopsumma. Ja ir nepieciešams, lai arī kopsumma tiktu rēķināta tikai šiem diviem gadiem, tas jānorāda kā lauka *Kopā* kritērijs.

Izpildot *File/Save As*, var saglabāt vaicājumu, piem., ar nosaukumu *_Klientu pasūtījumu summu šķērstabula*.

16.15. Biežāk lietotās funkcijas un operatori

Vaicājumus veidojot, apskatījām, ka, lai no datuma iegūtu pasūtījuma gadu, jāizmanto funkcija Year(). Vaicājumos, makrokomandās, formās un atskaitēs biežāk lietoto funkciju saraksts:

Biežāk lietotās funkcijas un operatori.

16.2.tabula

Apraksts	Funkcija vai operators
Atrod esošo datumu vai laiku	Date(), Now(), Time()
Nosaka argumenta – datuma/ laika mainīgā:	
gadu	Year (arguments)
mēnesi	Month (arguments)
nedēļas dienu	Weekday (arguments)
dienu	Day (arguments)
stundu	Hour (arguments)
minūti	Minute (arguments)
sekundi	Second (arguments)
Veic matemātiskās darbības ar datumiem	DateAdd (burts, kas apzīmē intervālu, intervāla skaitlis, sākuma datums);
	DateDiff (burts, kas apzīmē intervālu, sākuma datums, beigu datums)
Nosaka, vai arguments ir Null lielums	IsNull(arguments)
Aritmētiskie operatori	*, +, -, /, ^ (kāpināšana), \ (izdala divus integer tipa skaitļus), Mod (izdala divus skaitļus un rāda atlikumu)
Salīdzināšanas operatori	<, <=, >, >=, =, <> (nav vienāds)
Savienošanas operatori	&, +
Loģiskie operatori	And, Eqv, Not, Or, Xor
Atrod paraugam līdzīgos datus	Like
Atrod datus, kuru vērtība atrodas norādītajā intervālā starp diviem argumentiem	Between arguments1 And arguments2
Nosaka, vai dotā vērtība atrodas norādīto argumentu sarakstā	vērtība In (arguments1, ..., arguments n)

Burts, kas apzīmē intervālu, datumu funkcijās.

16.3.tabula

gads	yyyy
kvartāls	q
mēnesis	m
diena	d
nedēļa	ww
nedēļas diena	w
stunda	h
minūte	n
sekunde	s

16.16. Datu rediģēšana vaicājumos

Kad vaicājums ir izveidots, tas tiek aktivizēts un rezultāts apskatāms datulapas veidā. Parasti vaicājuma datulapā iespējams labot un pievienot ierakstus, tad attiecīgi mainās dati tabulās, no kurām vaicājums ir veidots. Dažu vaicājuma tipu datus datulapā nav iespējams mainīt. Tad *MS Access* statusa joslā parāda paziņojumu, parasti: "*This Recordset is not updatable.*"

Nav iespējams mainīt datulapā šādu vaicājumu datus:

- *Totals* tipa vaicājumiem;
- *Crosstab* tipa vaicājumiem;
- dažiem vaicājumiem ar viena virziena saiti (*outer join*) vai saiti, kur ierakstam vienā tabulā atbilst vairāki ieraksti citā tabulā (*one-to-many query*).

16.17. Darbības vaicājumi – *action queries*

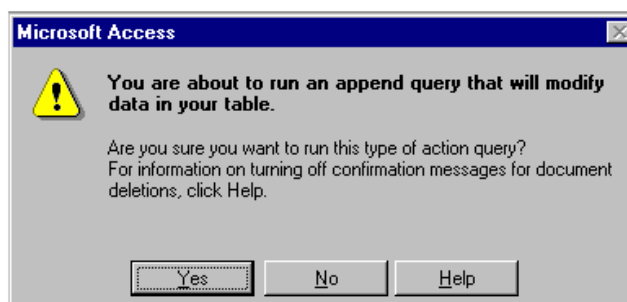
Līdz šim apskatītie vaicājumi atlasīja un sakārtoja tabulu datus, nemainot pašus datus. Turpmāk aprakstītie vaicājumu tipi ir darbības vaicājumi (*action queries*), kas tieši maina tabulu datus.

Šāda tipa vaicājumi vispirms tiek izveidoti līdzīgi *select* vaicājumiem. Arī datus, ar kādiem šie vaicājumi darbosies, ir iespējams apskatīt datulapā. Tad vaicājums tiek izpildīts, un tas izmaina tabulas datus.

Izpildīt darbības (*action*) vaicājumu var:

- veidošanas (*Design*) skatā izpildot *Query/Run* vai noklikšķinot *Run* ikonu;
- datu bāzes logā nospiežot *Open* pogu. Atšķirībā no *select* tipa vaicājumiem *Open* poga nevis atver datulapu, bet izpilda vaicājumu!

Izpildot vaicājumu no datu bāzes loga ar *Open* pogu, *MS Access* ziņo, ka darbības vaicājums (*append query, update query* u. c.) izmainīs datus:



16.43.att. Paziņojums par darbības vaicājuma aktivizēšanu.

Nospiežot *No* pogu, vaicājuma izpildi var atcelt. Nospiežot *OK* pogu, vaicājuma tiks izpildīts.

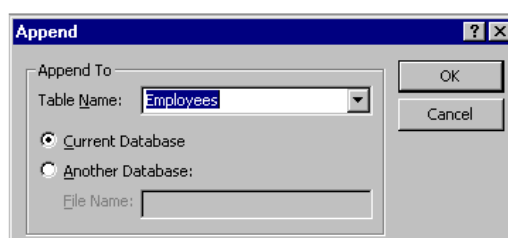
Datu bāzes loga nodaļā *Query* visiem action vaicājumiem priekšā ir ikona ar izsaukuma zīmi.

16.18. Datu pievienošanas vaicājums – *append query*

Append tipa vaicājumus izmanto, lai pievienotu esošajai tabulai atvērtās tabulas vai vaicājuma ierakstus, vai citas datu bāzes.

Ja tabula, kurai pievienos datus, satur primārās atslēgas lauku, pievienojamajiem ierakstiem arī jāsatur tāda paša tipa lauks. *MS Access* nepievienos ierakstus, ja pievienošanas rezultātā primārās atslēgas laukā varētu rasties tukša vai dubulta vērtība.

Piemēram, tabulai *Employees* var pievienot ierakstus no tabulas *Darbinieki*, t.i., atver veidošanas skatā jaunu vaicājumu, vaicājuma avots ir tabula *Darbinieki* un izpilda *Query/Append Query* vai klikšķina uz attiecīgās ikonas. Atveras *Append* logs:

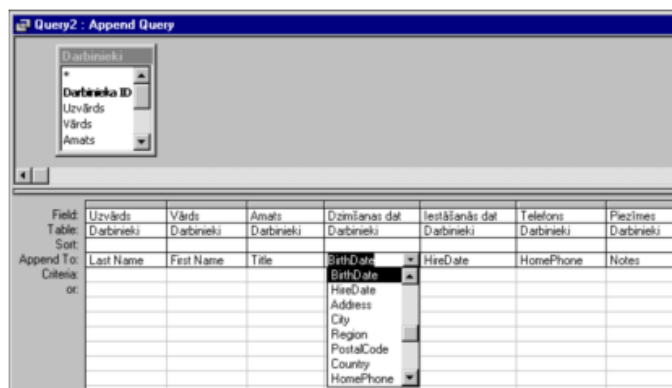


16.44.att. *Append* logs.

Lodziņā *Table Name* no saraksta jāizvēlas tabulas nosaukums, kurai pievienos ierakstus, tā būs tabula *Employees*. Ja ieraksti tiks pievienoti citai datu bāzei, tad jāieklikšķina *Another Database* un lodziņā *File Name* jāieraksta ceļš uz šo tabulu. Pēc tam logu *Query Properties* var aizvērt.

QBE režģī ir parādījusies jauna rinda *Append To*.

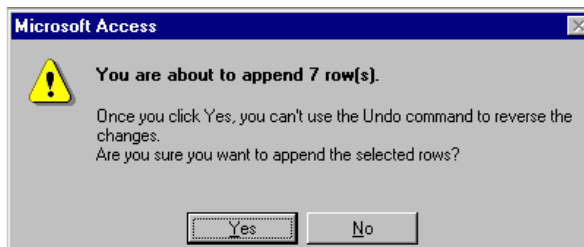
QBE režģi aizpilda šādi: katram laukam no tabulas *Darbinieki* rindā *Append To* jānorāda tabulas *Employees* lauks, kurā dati tiks pievienoti. Lauku *Darbinieka ID* vaicājumā neiekļauj, jo tabulā *Employees* darbinieka identifikatora lauks *EmployeeID* ir *AutoNumber* lauks, turklāt, primārās atslēgas lauks, un *MS Access* pats piešķirs pievienotajiem darbiniekiem identifikatoru, t. i., turpinās numurēšanu.



16.45.att. *QBE* režģis datu pievienošanas vaicājumā.

Pārejot uz datulapas skatu, varam pārbaudīt datus, kurus pievienos norādītajai tabulai.

Ja ar *Tools/Options* izsaucamajā *Options* logā *Edit/Find* nodaļas *Confirm* sekcijā ir atzīmēta izvēle *Action Queries*, tad izpildot vaicājumu, *MS Access* ziņos, ka tabulai *Employees* tiks pievienotas 7 rindas:



16.46.att. Paziņojums par ierakstu pievienošanu.

Lai pārliecinātos, ka tabulā *Employees* tiešām ir pievienoti ieraksti, no datu bāzes ekrāna ir jāatver tabula *Employees*.

16.19. Datu maiņas vaicājums – *update query*

Update tipa vaicājumus izmanto, lai izmainītu tās ierakstu grupas datus, kas tiek norādīta vaicājuma kritērijos. Protams, var mainīt uzreiz visu tabulas lauku datus.

Piemēram, tabulas *Employees* laukā *Title* lielākajai daļai ierakstu, kas pievienoti no tabulas *Darbinieki*, ir vērtība *Tirdzn. aģents*. To vajadzētu nomainīt ar *Sales Representative*.

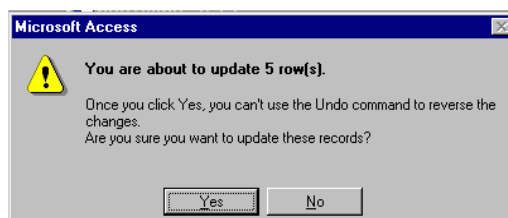
Atver veidošanas skatā jaunu vaicājumu, vaicājuma avots ir tabula *Employees* un izpilda *Query/Update Query* vai klikšķina uz attiecīgās ikonas.

QBE režģī parādās jauna rinda *Update To*, kurā jānorāda vērtība, kāda laukam jāpiešķir. *QBE* režģi aizpilda šādi:



16.47.att. QBE režģis maiņas vaicājumam.

Pārejot uz datulapas skatu, var pārbaudīt datus, kurus izmainīs. Izpildot vaicājumu, *MS Access* ziņo, ka tiks izmainīti dati 5 rindās:



16.48.att. Paziņojums pēc maiņas vaicājuma izpildes.

Lai pārliecinātos, ka tabulā *Employees* tiešām ir izmainīti ieraksti, no datu bāzes ekrāna ir jāatver tabula *Employees*.

Var pārbaudīt, vēlreiz izpildot vaicājumu. Tad *MS Access* ziņos, ka tiks izmainīti dati 0 rindās, jo vairs nav palicis ierakstu, kuru laukā *Title* ir ieraksts *Tirdzn. Aģents*.

16.20. Datu dzēšanas vaicājums – *delete query*

Delete tipa vaicājumi tiek izmantoti, lai izdzēstu ierakstu grupu, kuru norāda vaicājuma kritērijos. Ierakstus var izdzēst no vienas tabulas vai no

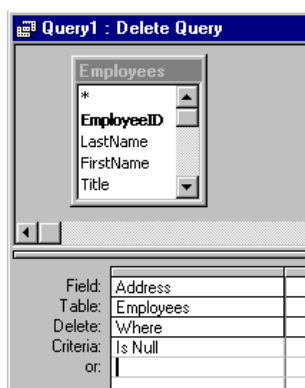
vairākām tabulām, ja starp tām ir *one-to-one* saite – ierakstam vienā tabulā atbilst viens ieraksts otrā tabulā.

Kad ieraksti ar *delete* vaicājumu ir izdzēsti, tos atjaunot vairs nav iespējams! Pirms vaicājuma izpildes noteikti jāpārlicinās, ka izdzēsti tiks tiešām vairāk nevajadzīgi ieraksti.

Piemēram, ir jāizdzēš no tabulas *Employees* tikko pievienotos ierakstus. Atpazīt šos ierakstus var pēc tā, ka tiem nav aizpildīti visi lauki, piemēram, lauks *Address*.

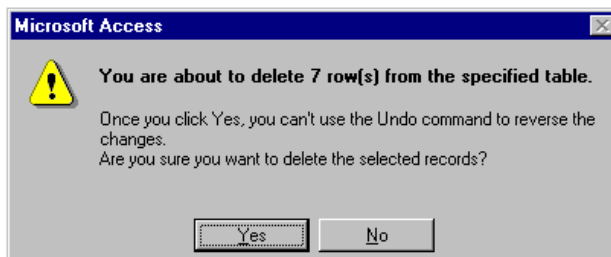
Atverot veidošanas skatā jaunu vaicājumu, vaicājuma avots ir tabula *Employees*, izpilda *Query/Delete Query* vai klikšķina uz attiecīgās ikonas.

QBE režģī parādās jauna rinda *Delete*, kurā jābūt *Where* (to parasti *MS Access* ieliek pats), ja zemāk laukam tiek norādīts ierakstu izdzēšanas kritērijs. *QBE* režģī aizpilda šādā veidā:



16.49.att. QBE režģis dzēšanas vaicājumam.

Pārejot uz datulapas skatu, var pārbaudīt datus, kuri tiks izdzēsti. Izpildot vaicājumu, *MS Access* ziņo, ka tiks izdzēstas 7 rindas.



16.50.att. Paziņojums pēc dzēšanas vaicājuma izpildes.

Lai pārlicinātos, ka tabulā *Employees* tiešām ir izdzēsti ieraksti, no datu bāzes ekrāna ir jāatver tabula *Employees*.

Pārbaudīt varētu, vēlreiz izpildot vaicājumu. Tad *MS Access* ziņos, ka tiks izdzēstas 0 rindas, jo vairs nav palicis ierakstu, kuriem lauks *Address* ir tukšs.

16.21. Tabulas izveidošanas vaicājums – *make-table query*

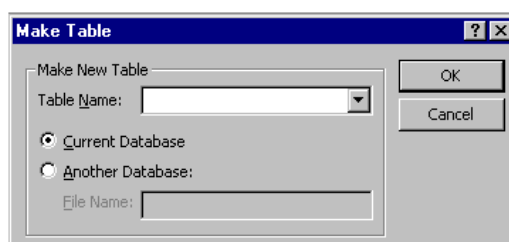
Make-table tipa vaicājumus izmanto, lai izveidotu tabulu no vaicājuma atlasītajiem datiem.

Ar *make-table* vaicājumiem arhivē datus, kopētas tabulas atvērtajā vai citā datu bāzē vai arī tiek veidotas starptabulas, kas ir pamats atskaitēm, piemēram, par noteiktu laika periodu.

Jaunās tabulas laukiem būs tas pats datu tips un īpašība *Field Size* kā ir izejas tabulu laukiem. Citas lauku vai tabulas īpašības netiek pārnestas.

Piemēram, ir jāizveido 1993. gada labāko klientu tabulu, izmantojot iepriekš izveidoto vaicājumu, kurš tos atlasa.

Atver veidošanas skatā vaicājumu un izpilda *Query/Make Table Query* vai klikšķina uz attiecīgās ikonas. Tiek atvērts *Query Properties* logs:



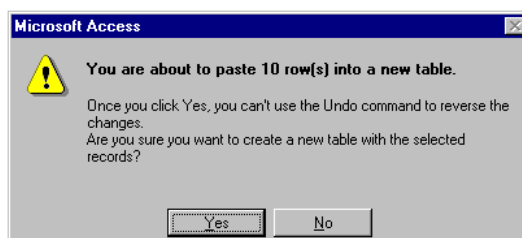
16.51.att. Make Table logs.

Lodziņā *Table Name* jāieraksta jaunās tabulas nosaukums. Ja no saraksta tiek atzīmēts jau esošās tabulas nosaukums, šīs tabulas struktūra un dati tiks aizvietoti ar vaicājuma struktūru un datiem. Nosaukt to var, piemēram, *labākie klienti 1996*.

Ja ieraksti tiks pievienoti citai datu bāzei, tad jāieklikšķina *Another Database* un lodziņā *File Name* jāieraksta ceļš uz šo tabulu, pēc tam logu *Query Properties* var aizvērt.

QBE režģī rindu skaits un izvietojums nemainās, un pašā vaicājumā nekas nav jāizmaina.

Pārejot uz datulapas skatu, var pārbaudīt datus, no kuriem izveidos jaunu tabulu.



16.52.att. Paziņojums pēc tabulas izveidošanas vaicājuma izpildes.

Izpildot vaicājumu, *MS Access* ziņo, ka jaunā tabulā tiks iekopētas 10 rindas.