Rīgas Tehniska universitāte

Datorzinātnes un Informācijas tehnoloģijas fakultāte

Trešais praktiskais darbs mācību priekšmetā

„Datu bāzu vadības sistēmas”

MS Access formas un pārskati

Izstrādāja: Sergejs Terentjevs

2. kurss, 9.grupa

Apl.nr. 061RDB140

Pārbaudīja: lekt. V. Vinogradova

2007./2008. māc. g.

# ANOTĀCIJA

Laboratorijas darbā gaitā tika izstrādāta *Kancelejas Preču* informācijas sistēma, pamatojoties uz MS Access formām, pārskatiem, makro komandām un izmantojot MS Access notikumu programmēšanas valodas *Visual Basic for Application* palīdzību. Praktiskajā darba par pamatu tika ņemta iepriekšējos darbos izstrādāta datu bāze, tādējādi tika turpināta šīs informācijas sistēmas izstrādāšana un nelielas datu bāzes konstruēšana, kurā glabā informāciju (vienkāršota veidā) par kādas kompānijas personālu, šī personāla vajadzīgas pēc kādas preces un veiktos pasūtījumus, informāciju par piegādātājiem, kategorijām un precēm.

Laboratorijas darbā pārskatā ir aprakstītas konceptuālas prasības datu bāzei, paradīta un aprakstīta lietojuma kopēja shēma, aprakstītas katras atsevišķas formas un pārskata būtiskas īpašības un to veidošanas paskaidrojums. Laboratorijas darba pārskats satur pielikumu – kompaktdisku , kurā atrodas izstrādāta MS Access datu bāze.

Praktiskais darbs tika izstrādāts MS Access 2007 uz *Pentium III*  tipā procesorā *Windows XP Proffesional* vidē. Tajā tika izstrādātas 29 makrokomandas, 15 SQL vaicājumi, 2 pārskati, 1 galvenais modulis, kurš glabā publiski pieejamas procedūras un globālos mainīgos, vairākas notikumu procedūras, 11 formas un 10 apakšformas un 3 funkcijas.

Laboratorijas darba pārskata apjoms: 37 lappuses izklāsta tekstā, tajā ietilpst 19 nodaļas, literatūras saraksts, 46. attēli.

# SATURS

[ANOTĀCIJA 2](#_Toc196097210)

[SATURS 3](#_Toc196097211)

[Uzdevuma nostādne 4](#_Toc196097212)

[Izmantotas tabulas 5](#_Toc196097213)

[Lietojuma kopēja shēma 9](#_Toc196097214)

[Formu veidošana 10](#_Toc196097215)

[Pieteikšanas forma 12](#_Toc196097216)

[Galvenā forma 16](#_Toc196097217)

[Preču forma 18](#_Toc196097218)

[Preces statusa maiņas forma 22](#_Toc196097219)

[Preces pasūtījumu forma 22](#_Toc196097220)

[Darbinieku forma 26](#_Toc196097221)

[Forma par preces saņēmušiem darbiniekiem 27](#_Toc196097222)

[Darbinieku administrēšanas forma 28](#_Toc196097223)

[Forma par preces nesaņēmušiem darbiniekiem 31](#_Toc196097224)

[Kategoriju forma 32](#_Toc196097225)

[Piegādātāju forma 33](#_Toc196097226)

[Pārskati 34](#_Toc196097227)

[SECINĀJUMI 36](#_Toc196097228)

[IZMANTOTA LITERATŪRA 37](#_Toc196097229)

# Uzdevuma nostādne

Izveidot datu bāzes lietojumu, izmantojot vismaz 5 formas, vismaz 1 pārskatu, makrokomandas un funkcijas.

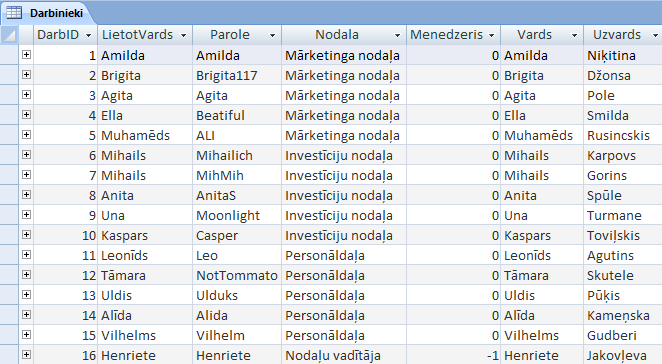
Formām jāattēlo dati gan no vienas tabulas, gan arī no vairākām tabulām. Visām formām jābūt saistītam un viena forma ir galvenā. Formas nodrošina pilnu darbību klāstu ar izvēlētām tabulām ( datu aplūkošana nepieciešamajos skatos, datu meklēšana un datu rediģēšana – pievienošana, labošana, dzēšana ) un lietotāja kļūdu novēršanu.

Darba aprakstā iekļaut lietojuma kopējo shēmu un atsevišķo elementu veidošanas paskaidrojums, norādot to būtiskās īpašības.

# Izmantotas tabulas

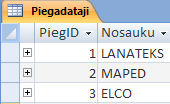
Iepriekšējos laboratorijas darbos tika risināta kādas kompānijas problēma, kura izpaudās, ka kompānijas finanses līdzekļu pārtērēšana kancelejas preču pasūtījumiem visas kompānijas nodaļas, tāda veida nevarēja noteikt kopējas izmaksas, ierobežot šīs izmaksas un atrast izdevīgos piegādātājus. Lai atrisinātu šo problēmu pirmajā praktiskā darbā tika izveido vienota centralizēta kancelejas preču pasūtījumu informācijas sistēmas datu bāze visām kompānijas nodaļām, otrajā uzdevuma tika izveidoti vairāki vaicājumi, kuri tapa par pamatu šī trešā praktiska darba izstrādātājai informācijas sistēmai pēdējam projektēšanas posmam. Datu bāzes pamatā ir septiņas tabulas.

Tabula *Darbinieki* glāba informāciju par kompānijas personālu, šī informācija galvenokārt ir nepieciešama informācijas sistēmas vajadzībām, tādam ka pieejas tiesību ierobežošana nodaļas darbiniekam tādas formas, kuras nodaļu vadītāja veic darbinieku administrēšanu, kategoriju, preču un piegādātāju ievadīšanu un dzēšanu. Par pamatu šai tabulai kalpo *Admin\_emp* (darbinieku administrēšanas) forma.



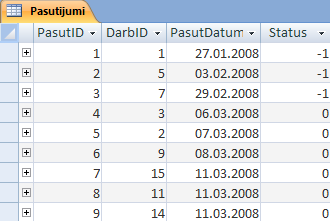
1. att. *Darbinieki* tabula.

Tabula *Piegādātāji* glabā kompāniju nosaukumus, kuri veic preču piegādājumus un šo kompāniju unikālos identifikatorus (kodus). Par pamatu dotai tabulai tika izstrādāta *Suppliers* (piegādātāji) forma.



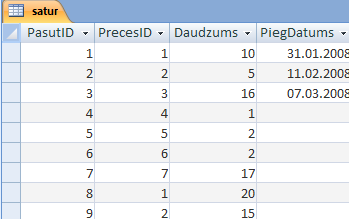
1. att. *Piegādātāji* tabula.

Tabula *Pasūtījumi* satur pasūtījuma unikālo identifikatoru, darbinieka unikālo identifikatoru, kurš ir veicis pasūtījumu, pasūtīšanas datumu un statusu ( vai pasūtījums piegādāts vai nē ). Informācijas sistēmas izstrādes posmā tika izveido vairāki SQL vaicājumi, pamatojoties uz šo tabulu.



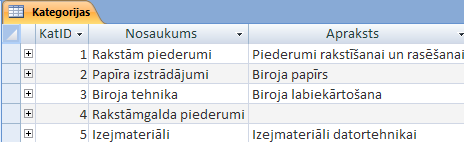
1. att. *Pasūtījumi* tabula.

Tabula *Satur* pilda starp tabulas lomu starp tabulām *Pasūtījumi* un *Preces*. Tā ļauj noteikt kopējo preces daudzumu pasūtījumos, kuros darbinieks ir to pasūtījis (izmaksas). Balstoties uz šo un *Pasūtījumi* tabulām tika izveidoti vairāki vaicājumi un apakšformas, ka, piemēram, *STATUS\_subform* apakšforma, tāda veido lietotājs var mainīt pasūtījuma statusu, kad tas ir saņemts.



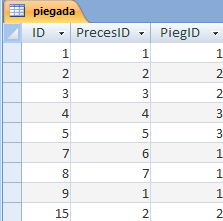
1. att. *Satur* tabula.

Tabula *Kategorijas* ir informatīva tabula, kura satur kategorijas unikālo identifikatoru, nosaukumu un aprakstu. Dotai tabulai ir veltīta *Categories* (kategorijas) forma.



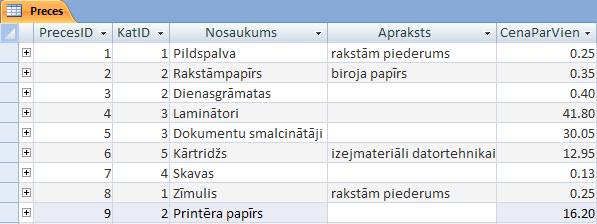
1. att. *Kategorijas* tabula.

Tabula *Piegādā* pilda starp tabulas lomu starp tabulām *Piegādātāji* un *Preces*. Tā ļauj noteikt kādu preci pasūtījumā ir jāpiegādā kādām piegādātājām. Balstoties uz tabulu tika izveidoti vairāki vaicājumi un apakšformas.



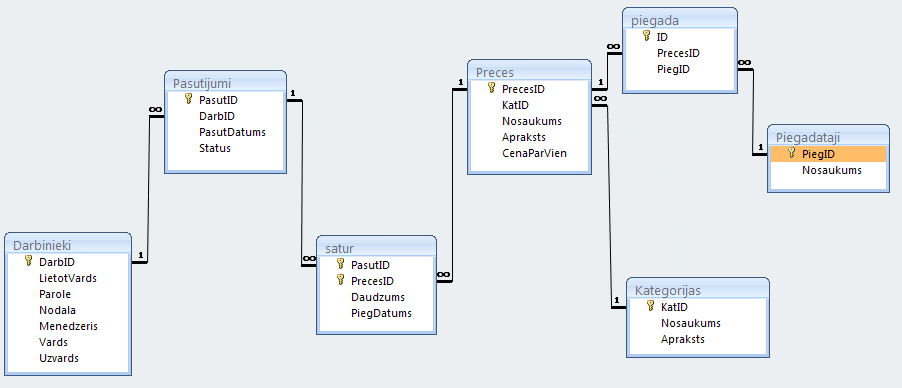
1. att. *Piegādā* tabula.

Tabula *Preces* satur katras preces unikālo identifikatoru, kurš tiek izmantots vairākas tabulas, preces nosaukumus, aprakstus, cenas par vienu vienību un norādi uz kategoriju, kādai pieder konkrēta prece. Dotai tabulai ir veltīta forma *Preces*, vairāki vaicājumi un apakšformas.



1. att. *Preces* tabula.

Saišu starp tabulām labākai uzskatāmībai, paradīsim to attēla 8, izmantojot MS Access 2007 *Relationships* veidni.



1. att. Relāciju saites starp tabulām.

# Lietojuma kopēja shēma

Pirmais formu un pārskatu (informācijas sistēmas) projektēšanas posms ir izstrādāt kopējo vienotu lietojuma shēmu. Šāda shēma ir parādīta attēlā 9, pēc shēmas ir redzams, ka lietotājs sāk darbu autorizējoties sistēmā, turpmāk tiek atvērta galvenā forma, no kuras lietotājs var izvēlēties vairākas darbības. Formas *Articles*, *Employers*, *Categories*, *Suppliers* satur tiesību ierobežojumus, t.i., ja lietotājs nav nodaļu vadītājs, tad viņš nevarēs izpildīt dažās komandas, ko piedāvā forma un atvērt formas, kurās parastam lietotājam pieeja liegta ( sarkana krāsa bultiņa ). Lietotājs var atvērt pārskatus ( zila krāsa bultiņa ). Plašākā informācija par formām būs pieejama zemākas nodaļas.

*Authentication* (pieteikšanas forma)

*MainForm* (galvenā forma)

*Articles* (preču forma)

*Orders* (pasūtīt preci)

*Status* (mainīt statusu)

*Employers* ( darbinieki)

*Admin\_emp* ( administrēšana)

*Employers\_Taked*

*Employers*\_Waiting

*Categories* ( kategorijas)

*Suppliers*( piegādātāji)

*Izmaksu-atskaite*

*Pasūtījumu-atskaite*

1. att. Kopēja lietojuma shēma.

( ) – vienreizēja pārēja (atvērta forma nevar atvērt formu, no kuras veikta pārēja);

( ) – vairākkārtēja pāreja;

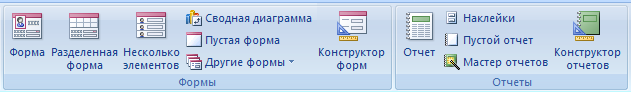
( ) – šādu formu spēj atvērt tikai nodaļu vadītajā;

( ) – pārēja uz pārskatu.

# Formu veidošana

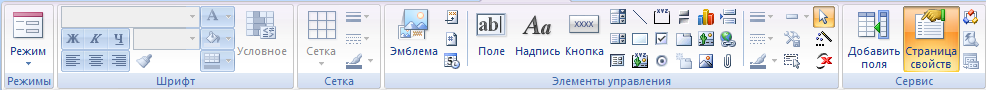
MS Access formas un pārskati ir ļoti labs risinājums, lai atvieglotu lietotāju no rutīnu darbību veikšanas. Jā ar datu bāzi ir jāstrādā ļoti bieži, tad izdevīgāk ir vadīt, koriģēt, dzēst datus izmantojot formas, nevis tabulas, bez tām formas var tikt izstrādātas uz vaicājumu pamatā. Savukārt, pārskati ļauj apkopot datus, tādā veida iegūstot pilna mēra atskaiti, kuru turpmāk var izdrukāt.

Formu un atskaišu veidošana MS Access 2007 vidē sākas ar rīkjoslu *Create*, kurā ir iedalīta visiem datu bāzes objektiem, tajā skaitā formām un pārskatiem (skat. att. 10).

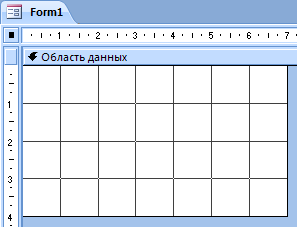


1. att. *Create* rīkjoslas mazais fragments.

Jaunas formas veidošanai bez MS Access piedāvātājām ir jāizvēlas formu konstruktors, rezultātā tiks atvērta rīkjosla *Konstruktors* ( skat. att. 11) un zem tās tukša forma (skat. att. 12).



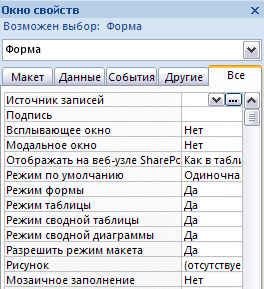
1. att. *Konstruktors* rīkjosla.



1. att. Tukša forma.

*Konstruktors* rīkjosla nodrošina pogas, laukus, diagrammas, zīmējumus, *ActiveX objektus*, pārslēgu grupas, lappuses pārrāvumus u.c. elementus formas arēja izskata veidošanai (skat. 11. att. zem šiem elementiem nosaukums - vadības elementi) un citus elementus, kuri ir atdalīti un saucas serviss. Šeit formai var pievienot tabulas vai vaicājuma atribūtus, atvērt *Visual Basic for Application* (turpmāk vba ) vidi un formu, ka arī elementu īpašību veidni (att. 11 šī komandpoga ir iezīmēta).

Īpašību veidnī ir iespējams noradīt objektu īpašības, piemēram, uzraksts, augstums, garums, datu avots u.t.t. Tajā ir četri ieliktņi: Makets *(Format)*, Dati *(Data),* Notikums *(Event)* , Citi (*Other*) un Viss *(All)* ( skat. att. 13). Savā darbā es pārsvarā izmantoju trīs ieliktņus.



1. att. Īpašību veidnis.

Ieliktnī Notikums (*Event*) var norādīt kādu darbību izpildīs objekti, kad notiksies kāds notikums. Šādu darbību var realizēt izmantojot procedūru, kuras pieraksts jāveic VBA, vai arī makrokomandas un funkcijas.

Ieliktnis Dati (*Data*) paredz datu manipulēšanu, piemēram, datus nevar mainīt, dzēst, pievienot u.t.t. no dotas formas. Šeit var arī norādīt datu avotu.

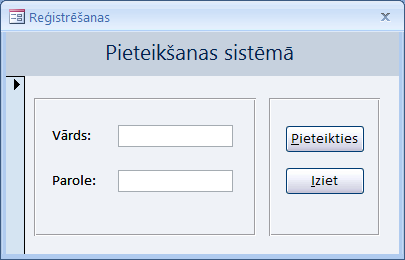
Ieliktnis Makets(*Format*)satur īpašību sarakstu, kuras nodrošina formas vai cita elementa ārēja izskata veidošanu.

# Pieteikšanas forma

Ielādējoties Access uzreiz ielādē pieteikšanas formu (*Authentication*), tādejādi lietotājam vispirms ir nepieciešams identificēt sevi. Atkarība no lietotāja, kurš ir pieteicies sistēmā, turpmāk vairākas formas tiek ierobežotas piekļuves tiesības pie darbībām, kuras piedāvā šīs formas.

Pieteikšanas formai īpašību veidnī tika noradīts uzraksts (*Caption*) „Reģistrēšanas”, atslēgtas pārvietošanas pogas ( *Navigation Buttons* ) un loga izmēra pogas ( *MinMaxButtons* ), atstājot vienīgi aizvēršanas pogu ( *CloseButton* ). Access 2007 ir divas svarīgas īpašības, kuras es biju izmantojis visām formām un pat pārskatiem, *peldošais logs* un *modālais logs*, noradot formas izskatu, to atvērot (skat. att. 13). Šīs īpašības būs kopīgas visām formām.

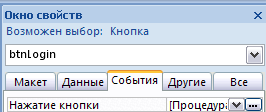
Formas ārējo izskatu veido uzraksts ( *Label* ) „ Pieteikšanas sistēmā”, kurš ir izvietots formas augšēja virsrakstā (*Form Header*), divi teksta lauki (*TextBox*), kuros lietotājs ievada lietotājvārdu un paroli, divas pogas (*Command* *Button*) „Pieteikties” un „Iziet”. Šīs kontroles sava starpā ir atdalītas ar elementu taisnstūris ( *Rectangle*).



1. att. Pieteikšanas forma.

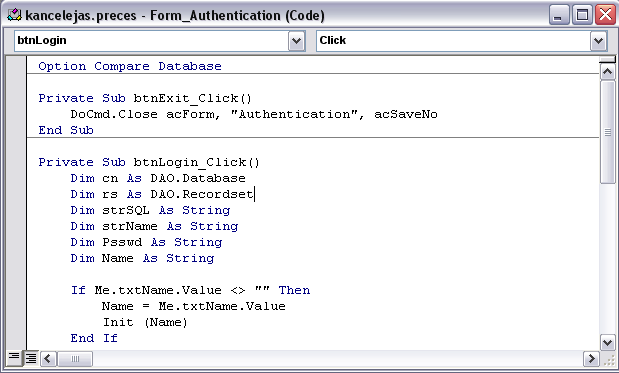
Formai ir jāidentificē lietotāju, lai panāktu šādu darbību klāstu, labākais variants ir uzrakstīt procedūru, kura atlasītu no tabulas *Darbinieki* lietotājvārdu un paroli, kurus ir noradījis lietotājs. Gadījumā jā lietotājs neievada informāciju vai arī ievada nekorekto informāciju, jāizvada attiecīgs paziņojums, citādi jāpariet uz nākamo formu.

Izvēlēsimies pogu ar nosaukumu „Pieteikties” un notikumu *OnClick* ieliktnī Notikums (*Event*) (skat. att. 15),tāda veida mūsu izstrādāta procedūra izpildīsies uzreiz pēc pogas nospiešanas.



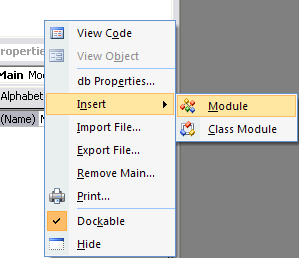
1. att. Notikums *OnClick*.

Izpildot šo punktu atvērsies logs *Veidotājs (Builder)*, kurā var izvēlēties veidot makrokomandas (*Macro Builder*), funkcijas (*Expression Builder*) vai arī procedūras - VBA *Sub* (*Code Builder*). Mums ir jāizvēlas trešais punkts, ka rezultātā tiks atvērts *Visual Basic for Application* programmēšanas vide (skat. att. 16). Pievēršot uzmanību, var redzēt, ka šajā loga tiek piedāvātas divas izvēlnes, no kurām viena var noradīt objektu un otrā notikumu, piemēram, *btnLogin* un *Click*.



1. att. VBA programmēšanas vide.

Atgriežoties pie sākotnēja uzdevuma, rodas vēl viena problēma – ka saglabāt lietotājvārdu (mainīgo), ko ievada lietotājs pieteikšanas formā, tā lai tas būtu pieejams visām Access formām sesijas laikā, jo lokālie mainīgi ( definēti procedūrā ) tiek atbrīvoti no atmiņas uzreiz pēc procedūras izpildes. Access piedāvā veidot moduli, kurā ir iespējams glabāt globālos mainīgos un publiski pieejamas funkcijas. Lai izveidotu tādu moduli, ar peles labo taustiņu jānospiež uz izvēlnes, kurā attēlojas viss objektu saraksts, un atvērtajā izvēlne jāizvēlas *Insert -> Module* (skat. att. 17).



1. att. Galvenā moduļa veidošana.

Modulī veicām šādu pierakstu:

Option Compare Database

Global gblNickName As String

Public Sub Init(ByVal Value As String)

gblNickName = Value

End Sub

Tādejādi mēs definējām globālo mainīgo *gblNickName* un publiski pieejamo procedūru *Init*, kurai tiek padota mainīga *Value* vērtība un tā tiek piešķirta globālajam mainīgajām.

Nākamais punkts ir uzrakstīt instrukciju pogai „Pieteikties”, pieraksts izskatās šādi:

Private Sub btnLogin\_Click()

Dim cn As DAO.Database

Dim rs As DAO.Recordset

Dim strSQL As String

Dim Psswd As String

Dim Name As String

If Me.txtName.Value <> "" Then

Name = Me.txtName.Value

Init (Name)

End If

If Me.txtPsswd.Value <> "" Then

Psswd = Me.txtPsswd.Value

End If

If gblNickName <> "" And Psswd <> "" Then

Set cn = CurrentDb

strSQL = "SELECT DarbID, LietotVards, Parole FROM Darbinieki" & \_

" WHERE LietotVards='" & gblNickName & "' AND Parole='" & Psswd & "';"

Set rs = cn.OpenRecordset(strSQL, dbOpenDynaset)

If rs.EOF Then

MsgBox "Autorizesanas kluda!", vbCritical, "Error"

Else

DoCmd.OpenForm "MainForm"

DoCmd.Close acForm, "Authentication", acSaveNo

End If

Else

MsgBox "Ludzu ievadiet informaciju!", vbCritical, "Warning"

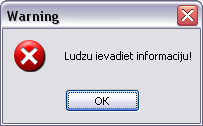
End If

Set rs = Nothing

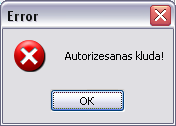
Set cn = Nothing

End Sub

Vispirms definētajam mainīgam *Name* tiek piešķirta teksta lauku *txtName* vērtība, kurā lietotājs ievada lietotājvārdu un izsaukta publiska procedūra *Init*, kurā globālajam mainīgajam galvenajā modulī tiek piešķirta mainīga *Name* vērtība. Līdzīgi notiek ar otru mainīgo *Psswd* , kuram tiek piešķirta paroles vērtības. Turpmāk seko pārbaude - ja lietotājs nav neko ievadījis un ir piespiedis pogu tiks atgriezts attiecīgs paziņojums (skat. att. 18), citādi izmantojot *DAO* datu piekļuves objektu izpildām vaicājumu, kura atlasam darbinieku pēc ievadīta lietotājvārda un paroles. Ja netiek atgriezts neviens šāds raksts, tad tiek izvadīts paziņojums par autorizēšanas kļūdu (skat. att. 19), citādi tiks atvērta nākama forma un aizvērta patreizēja forma.



1. att. Brīdinājuma paziņojums.



1. att. Brīdinājuma paziņojums.

Līdzīgās darbības veiksim ar formas otru pogu „Iziet” (*btnExit*), kurai pierakstam šādu instrukciju:

Private Sub btnExit\_Click()

DoCmd.Close acForm, "Authentication", acSaveNo

End Sub

Šīs pieraksts pasaka galvenajām komandu izpildes objektam *DoCmd* aizvērt pieteikšanas formu, to nesalabojot ( šāds pieraksts izskatās makrokomandām ).

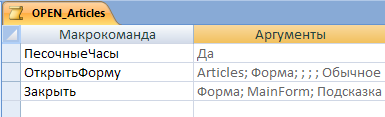
# Galvenā forma

Pēc lietotāju identificēšanas tiek atvērta galvenā forma (*MainForm*), kurā lietotājs var izvēlēties atvērt preču, darbinieku, kategoriju, piegādātāju formas vai pasūtījumu un izmaksu pārskatus, vai arī iziet no Access.



1. att. Galvenā forma.

Formas ārējo izskatu veido vairākas kontroles - pogas, teksta lauki, pārslēgu grupas (*Option Group*) un zīmējums (*Image*). Var pamanīt, ka vairākas kontroles ir sagrupētas pārslēgu grupas un šīm grupām ir dots attiecīgs nosaukums, tādējādi lietotājam ir vieglāk orientēties šinī formā. Grupas „Formas” pogām „Preces”, „Darbinieki”, „Kategorijas” un „Piegādātāji” ir izstrādātas makrokomandas *OPEN\_Articles*, *OPEN\_Employers, OPEN\_categories, OPEN\_Suppliers* notikumā *OnClick.* Apskatīsim makrokomandu *OPEN\_Articles*:



1. att. *OPEN\_Articles* makrokomanda.

Makrokomanda izpildīsies, kad ir nospiesta poga „Preces”. Vispirms tā izvadīs uz ekrānā smilšu pulksteni parasta peles kursora vietā, tad atvērs preču formu (*Articles*) un aizvērs patreizējo formu.

Līdzīgi ir ar parējām makrokomandām, tikai katra atvērs savu formu:

* poga „Darbinieki” atvērs darbinieku formu;
* poga „Kategorijas” atvērs kategoriju formu;
* poga „Piegādātāji” atvērs piegādātāju formu.

Gandrīz līdzīgas pēc darbības makrokomandas ir izstrādātas grupas „Pārskati” pogām - makrokomanda *OPEN\_Orders\_report* atvērs pārskatu *Pasūtījumu\_atskaite* un makrokomanda *OPEN\_expenses* atvērspārskatu *Izmaksu\_atskaite*, vienīgi abos gadījumos netiks aizvērta iepriekš atvērta forma.

Galvenajā forma tika izmantotas divas funkcijas:

* *date* - šodien dienas datuma izvadīšanai;
* *time* - laika izvadīšanai.

Lai panāktu laika patstāvīgo atjaunošanos, formas īpašībai taimerim (*Timer*) tika noradīta izveidota makrokomanda *Updater*, kura ik pēc noradīta taimera intervālā atjauno teksta lauku, kurš izvada laiku (skat. att. 22).

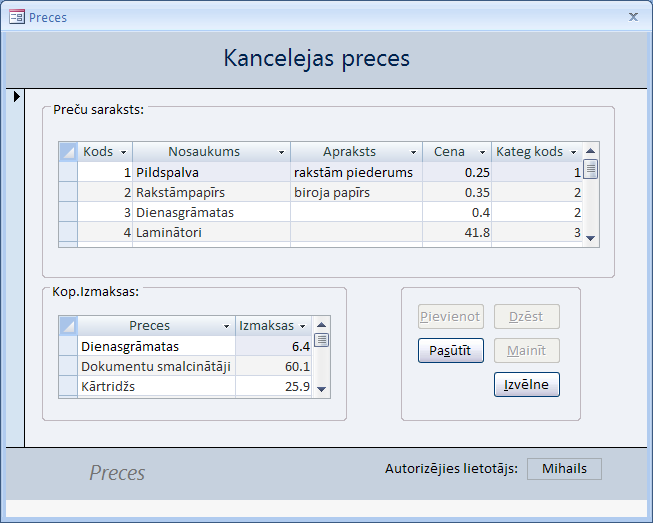
Forma ļauj iziet no MS Access, tas tiek panākts ar makrokomandu *Exit,* nospiežot uz pogas, uz kuras ir attēlotas durvis (skat. att. 20). Pirms aizvēršanas, tiks izvadīts paziņojums, piedāvājot saglabāt veiktas izmaiņas, ja tādas ir veiktas.



1. att. *Updater*  makrokomanda.

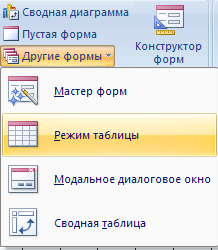
# Preču forma

Preču forma (*Articles*) ir preču informatīva forma, kurā autorizējies lietotājs var apskatīt preču sarakstu un izmaksas par katru pasūtītu preci. Neatkarīgi no lietotāja, kas ir autorizējies sistēmā, viņš var atvērt formu, lai veiktu preces pasūtījumu, bet pievienot un dzēst preci var tikai nodaļu vadītāja. Viņai arī ir pieejas tiesības atvērt formu, kurā var mainīt pasūtījuma statusu – atzīmēt to par saņemtu, ja prece ir atnākusi.



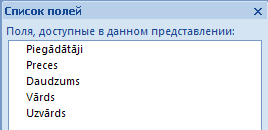
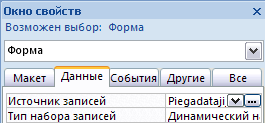
1. att. Preču forma.

Tabulas pamatā ir divas apakšformas *Articles\_Subform* un *Expenses\_Subform*, kuras ir izveidotas uz vaicājumu pamatā. Šāda apakšforma ir ļoti labs risinājums, lai aizliegtu lietotājam mainīt datus tabulā. Lai izveidotu šādu apakšformu, iepriekš pieminētajā *Create* rīkjosla ir jāizvēlas veidot formu tabulas režīmā (skat. att. 24).



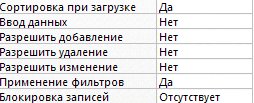
1. att. Tabulas režīms.

Atvērsies tukša forma. Nākamais solis ir noradīt datu avotu un pārvilkt uz formu laukus, kurus mēs vēlāmies redzēt. Tātad vispirms īpašību veidni ieliktnī *Dati* norādām datu avotu, par kuru var kalpot vai nu vaicājums, vai arī tabula (skat. att. 25) un rīkjoslā *Konstruktors* izvēlāmies komandu *Pievienot laukus* (skat. att. 26). Rezultāta īpašību veidņa vietā tiks atvērts lauku (tabulas vai vaicājuma atribūtu) saraksta veidnis (skat. att. 27) no kurā *Design View* jeb konstruktorā režīmā ir jāpārvelk šos laukus uz formu.



1. att. Datu avots. 26. att. Komandpoga. 27. att. Lauku saraksts.

Izpildot šos punktus, tiks iegūta tabulas veida forma, tāda formas ir redzamas attēlā 23 pārslēgu grupās „preču saraksts” un „kop. izmaksas”. Lai neļautu lietotājam modificēt apakšformas datus, īpašību veidnī *Dati* ieliktni var norādīt pēc vajadzības attiecīgas īpašības (skat. att. 28).



1. att. Īpašības.

Tāda veida es esmu panācis, lai lietotājs nevarētu izdarīt to, ko es esmu viņam aizliedzis. Lai paradītu apakšformu galvenajā formā, ir jāizmanto saliktās formas (*SubForm*) elementu.

Formas apakšā ir paradīts lietotājs, kurš ir autorizējies sistēmā, mūsu gadījuma tās ir Mihails, sistēma spēj atpazīt, ka viņš nav nodaļu vadītājs un tāpēc pogas „Pievienot”, „Dzēst” un „Mainīt” (statusu) paliek neaktīvas. Tas tika panākts formas notikumam *OnLoad* pierakstot procedūru, kura no tabulas *Darbinieki* izgūst nodaļu vadītāja lietotājvārdu un salīdzina to ar pat reiz pieteikto lietotāja vārdu, kura vērtība ir saglabājusies globālajā mainīgajā. Tāda pārbaude ir veikta formai ielādējoties.

Šādas procedūras pieraksts izskatās šāds:

Private Sub Form\_Load()

Dim cn As DAO.Database

Dim rs As DAO.Recordset

Dim strSQL As String

Dim Admin As String

Me.txtNickName.Value = gblNickName

cn = CurrentDb

strSQL = "SELECT LietotVards FROM Darbinieki WHERE Menedzeris = true;"

rs = cn.OpenRecordset(strSQL, dbOpenDynaset)

If rs.RecordCount > 0 Then

rs.MoveFirst()

Admin = rs.Fields(0)

End If

If Admin <> "" Then

If Admin <> Me.txtNickName.Value Then

Me.btnAdd.Enabled = False

Me.btnDELETE.Enabled = False

Me.btnChange.Enabled = False

End If

End If

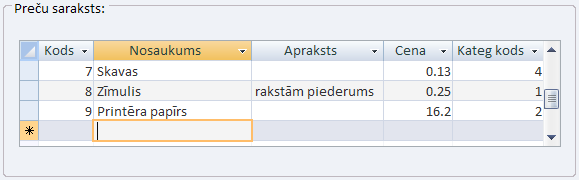
rs = Nothing

cn = Nothing

End Sub

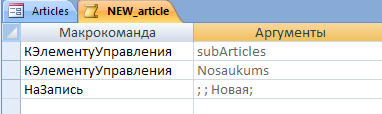
Ielādējoties formai, tās teksta laukam (*txtNickName*) tiek piešķirta globāla mainīga vērtība, kurš satur lietotājvārdu kādu ir noradījis lietotājs autorizējoties sistēmā. Tad izmantojot ADO datu piekļuves objektu no tabulas darbinieki tiek atlasīts nodaļu vadītājas lietotājvārds un tās tiek piešķirts *Admin* mainīgajām. Turpmāk seko pārbaude, kuras rezultāta pogas „Pievienot”, „Dzēst” un „Mainīt” paliek neaktīvas, ja autorizējies lietotājs nav nodaļu vadītāja Henriete.

Pieņemot, ka autorizējies lietotājs ir nodaļu vadītājs, tas var nospiest pogu „Pievienot”. Izpildes rezultātā kursors pārleks uz preču formas *SubForm* elementā esošo apakšformu *Articles\_Subform* (atrodas grupā ar nosaukumu „Preču saraksts”) un piedāvās veikt jauno ierakstu, kurš saglabāsies *Preces* tabulā (skat. 29. att.).



1. att. Datu ievadīšana.

Šādu darbību klāstu nodrošina makrokomanda *NEW\_article*, kurā ir izstrādāta pogas „Pievienot” notikumā *OnClick*. Tātad nospiežot pogu, kursors pārlec uz *SubForm* elementu *SubArticles* un tieši pie apakšformas lauka *Nosaukums*, vēl makrokomandai ir noradīts jauna ieraksta režīms, tāpēc kursors pārlec uz jaunu rakstu pēc pēdēja esoša. Makrokomanda ir paradīta attēlā 30.



1. att. makrokomanda *NEW\_article*.

Bez jaunas informācijas pievienošanas forma paredz arī datu dzēšanu, kuru var veikt vienīgi nodaļas vadītājs. Pogai „Dzēst” notikuma *OnClick* ir uzrakstīts šāds kods:

Private Sub btnDELETE\_Click()

'Dim rec As dao.Recordset

'Dim lngID As Long

'lngID = Forms!Articles!subArticles(PrecesID)

'Set rec = CurrentDb.OpenRecordset("Preces")

'rec.Seek "=", lngID

'rec.Delete

'Set rec = Nothing

DoCmd.SetWarnings(False)

DoCmd.RunSQL("DELETE \* FROM Preces WHERE PrecesID = " & \_

"Forms!Articles!subArticles!PrecesID.Value")

DoCmd.SetWarnings(True)

Forms!Articles!subArticles.Requery()

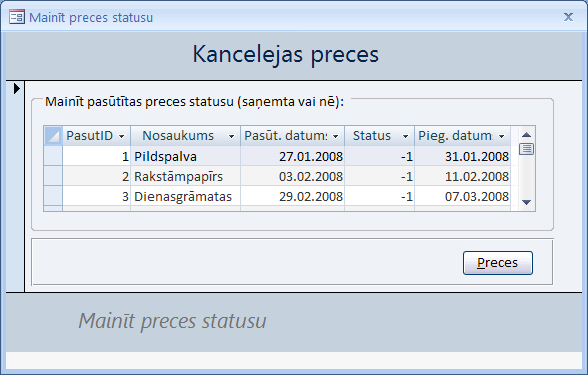
End Sub

Vispirms atslēdzam visus brīdinājuma paziņojumus par dzēšanu un tad izpildām SQL vaicājumu, kā rezultātā tiks dzēsts tas raksts, kas ir iezīmēts vai ieradīts *SubForm* elementā apakšforma *Articles\_Subform* (*PrecesID* vērtība tiek iegūta automātiski norādot konkrēta raksta pat citu lauku)*.* Lai atjaunotu apakšformu pēc dzēšanas, izpildu atjaunošanu, kuru realizē *Requery* metode.

Pārējas pogas atver citas formas ar makrokomandu palīdzību, kuru darbība līdzinās makrokomandām, kuras tika apskatītas 16. lpp. Poga „Mainīt” atver pasūtījuma vai preces statusa maiņas formu (*Status*) (parasts lietotājs nevar to atvērt), savukārt, poga „Pasūtīt” atver pasūtījumu veikšanas formu (*Orders*) un poga „Izvēlne” atgriež lietotāju galvenajā forma (*MainForm*).

# Preces statusa maiņas forma

Preces statusa maiņas forma (Status) ļauj nodaļu vadītājam vai vadītājai mainīt preces statusu, ja esoša prece ir pienākusi un darbinieks to ir saņēmis. Lai to izdarītu, vadītajai tabulā ir jāsameklē atbilstošs pasūtījuma kods, var arī orientēties pēc preces un pasūtīšanas datuma un kolonnā status „0” vietā jāieraksta „1”, ka arī obligāti jānorada piegādes datumu.



1. att. Preces statusa maiņas forma.

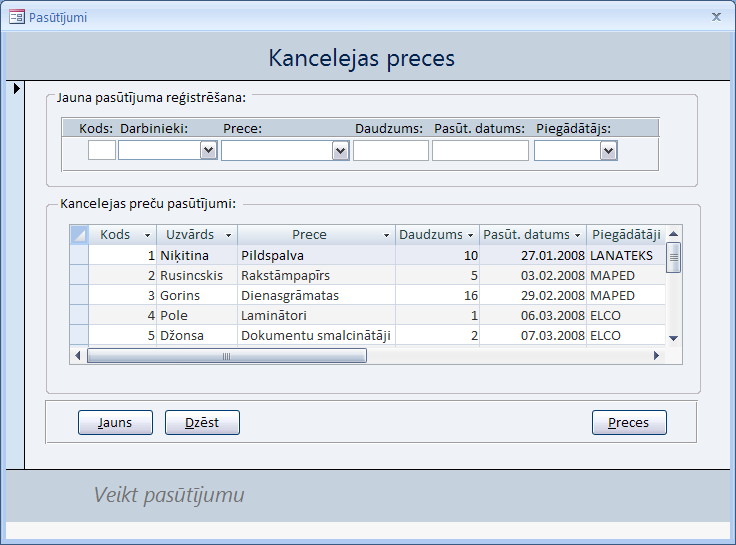
Formas pamatā ir apakšformā *Status\_subform*, kuras pamats ir vaicājums, kas atlasa datus no tabulām *Preces*, *Satur* un *Pasūtījumi.* Forma satur pogu „Preces”, kas atgriež lietotāju preču forma.

# Preces pasūtījumu forma

Šī forma ir paredzēta preču pasūtījumu pieteikšanai. Ērtībās labām forma sastāv no divām grupām. Pirmā grupa ar nosaukumu „Jauna pasūtījuma reģistrēšana” satur vairākas kontroles, kuras lietotājām ir jānorāda kārtējas pasūtījuma kods, izmantojot izvēlni (elements *ComboBox*) jāizvēlas sevī no saraksta pēc uzvārda, nākamā izvēlne ar nosaukumu „Prece” ir jāizvēlas preces nosaukums, kuru lietotājs vēlas pasūtīt, tad ir jānorāda daudzums un pasūtījuma veikšanas datums, ka pēdējo ir jāpiemin izvēlni „Piegādātājs”, kurā lietotājam no saraksta ir jāizvēlas piegādātāju, kurš piegādās doto preci. Šiem laukiem ir jābūt obligāti aizpildītiem, pirms spiest pogu „Jauns”, citādi jauns pasūtījums netiks pieteikts un lietotājām tiks paziņots par nekorektas informācijas ievadīšanu.

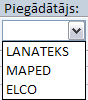
Otrajā grupa ar nosaukumu „Kancelejas preču pasūtījumi” ietver sevī detalizētu sarakstu tabulās veidā par visiem lietotāju veiktiem pasūtījumiem. Kad lietotājs veic jaunu pasūtījumu, jauns pasūtījums uzreiz tiek paradīts šajā saraksta. Lai izdzēstu kādu pasūtījumi, to var izdarīt visi lietotāji bez ierobežojumiem, lietotājam šajā sarakstā jāizvēlas pasūtījums un jāiezīmē jebkuru šī pasūtījuma lauku vai arī uzreiz visu rindu un jānospiež pogu „Dzēst”, rezultāta pasūtījums tiks izdzēsts.

Preču pasūtījumu forma ir paradīta 32 attēlā.



1. att. Preču pasūtījumu forma.

Tagad tiksim skaidrība, ka tas viss strādā un sāksim ar pirmo grupu. Pirmās grupas izvēlņu (*ComboBox*) pamatā ir vaicājumi (kontroļu datu avots īpašība), kuri atlasa darbinieku uzvārdus, preču un piegādātāju nosaukumus no attiecīgajām tabulām *Darbinieki*, *Preces* un *Piegādātāji*, tāda veida lietotājam tika atvieglota preču pasūtījuma sistēmā. Tāda izvēle ir paradīta 33 attēlā vienai no trim izvēlnēm „Piegādātājs”.



1. att. Izvēlne.

Nākamais posms ir uzrakstīt procedūru, kura saglabā lietotāja ievadīto informāciju vairākas tabulas, precīzāk, tabulas *Pasūtījumi*, *Satur* un *Piegādā*. Šāda procedūra tika izstrādāta pogai „Jauns” notikumā *OnClick*, tas pieraksts izskatās šāds:

Private Sub btnADD\_Click()

Dim cn As DAO.Database

Dim rs As DAO.Recordset

Dim strSQL As String

Dim Employee As String

Dim Article As String

Dim Supplier As String

If (Me.txtCode.Value <> "") And (Me.txtQuantity.Value <> "") \_

And (Me.cmbArticles.Value <> "") \_

And (Me.cmbEmployee.Value <> "") And (Me.txtDate.Value <> "") \_

And (Me.cmbSupplier.Value <> "") Then

cn = CurrentDb

strSQL = "SELECT DarbID FROM Darbinieki WHERE Uzvards = '" &

Me.cmbEmployee.Value & "';"

rs = cn.OpenRecordset(strSQL, dbOpenDynaset)

If rs.RecordCount > 0 Then

rs.MoveFirst()

Employee = rs.Fields(0)

End If

strSQL = "SELECT PrecesID FROM Preces WHERE Nosaukums = '" &

Me.cmbArticles.Value & "';"

rs = cn.OpenRecordset(strSQL, dbOpenDynaset)

If rs.RecordCount > 0 Then

rs.MoveFirst()

Article = rs.Fields(0)

End If

strSQL = "SELECT PiegID FROM Piegadataji WHERE Nosaukums = '" &

Me.cmbSupplier.Value & "';"

rs = cn.OpenRecordset(strSQL, dbOpenDynaset)

If rs.RecordCount > 0 Then

rs.MoveFirst()

Supplier = rs.Fields(0)

End If

rs = Nothing

cn = Nothing

DoCmd.SetWarnings(False)

DoCmd.RunSQL("INSERT INTO Pasutijumi(PasutID,DarbID,PasutDatums)" & \_

" VALUES (" & Me.txtCode.Value & "," & \_

" " & Employee & ", '" & Me.txtDate.Value & "');")

DoCmd.RunSQL("INSERT INTO satur(PasutID, PrecesID, Daudzums ) " & \_

"VALUES (" & Me.txtCode.Value & "," & \_

" " & Article & ", " & Me.txtQuantity.Value & ");")

DoCmd.RunSQL("INSERT INTO piegada( PrecesID, PiegID)" & \_

" VALUES (" & Article & "," & \_

" " & Supplier & ");")

DoCmd.SetWarnings(True)

Me.subOrders.Requery()

Else

MsgBox("Nav noradita informacija")

End If

End Sub

Analizējot procedūru, ir redzams, ka vispirms seko pārbaude, kas pārbauda vai lietotājs ir aizpildījis visus grupas „Jauna pasūtījuma reģistrēšana” lauku vērtības. Jā tas netika izdarīts, procedūra atgriezis paziņojumu „nav noradīta informācija”, citādi vispirms no trijām izvēlnēm „Darbinieki”, „Prece” un „Piegādātājs” ir jāiegūst lietotāja norādīto vērtību unikālas identifikatoru vērtības, lai mēs varētu turpmāk šo informāciju saglabāt tabulas *Piegādātāji*, *Satur* un *Piegādā*, jo šajās tabulas glabājas nevis nosaukumi, bet unikālie preču, piegādātāju un darbinieku identifikatori (skat. 5., 6., 7. lpp.).

Tātad, izmantojot DAO datu piekļuves objektu, vispirms izgūstam darbinieka unikālo identifikatoru un piešķīrām to mainīgajām *Employee*, tad preces unikālo identifikatoru piešķiram mainīgajām *Article* un piegādātāja unikālo identifikatoru piešķirām mainīgajām *Supplier*.

Tad atslēdzam brīdinājuma paziņojumus par datu pievienošanu tabulām un izmantojot *DoCmd* izpildes objektu izpildām pirmo vaicājumu, kurš tabulā *Pasūtījumi* saglabā lietotāja kontroles ievadītas vērtības, t.i., pasūtījuma kodu, darbinieka unikālo identifikatoru, pasūtīšanas datumu. Tad tiek izpildīts otrais vaicājums, kurš tabulā *Satur* saglabā pasūtījuma kodu, preces unikālo identifikatoru un daudzumu, ko ir noradījis lietotājs. Pēdējais vaicājums veic preču un piegādātāju unikālo identifikatoru saglabāšanu tabulā *Piegādā.* Viss tagad pasūtījums ir saglabāts un darbiniekam atliek vienīgi gaidīt, kad tas atnāks.

Gadījumā ja lietotājs vēlas izdzēst kādu pasūtījumu un viņš to ir atzīmējis sarakstā (saraksts ir izstrādāts uz apakšformas *Pasutijumu\_saraksts* pamatā), viņam ir tikai jānospiež pogu „Dzēst”, tad tiks izpildīta šāda procedūra:

Private Sub btnDELETE\_Click()

Dim cn As DAO.Database

Dim rs As DAO.Recordset

Dim strSQL As String

Dim Supplies As String

cn = CurrentDb

strSQL = "SELECT PrecesID FROM Preces WHERE Nosaukums = '" &

Forms!Orders!subOrders!Preces.Value & "';"

rs = cn.OpenRecordset(strSQL, dbOpenDynaset)

If rs.RecordCount > 0 Then

rs.MoveFirst()

Supplies = rs.Fields(0)

End If

rs = Nothing

cn = Nothing

DoCmd.SetWarnings(False)

DoCmd.RunSQL("DELETE \* FROM satur WHERE PasutID = " & \_

"Forms!Orders!subOrders!PasutID.Value")

DoCmd.RunSQL("DELETE \* FROM Pasutijumi WHERE PasutID = " & \_

"Forms!Orders!subOrders!PasutID.Value")

DoCmd.RunSQL("DELETE \* FROM piegada WHERE PrecesID = " & Supplies & ";")

DoCmd.SetWarnings(True)

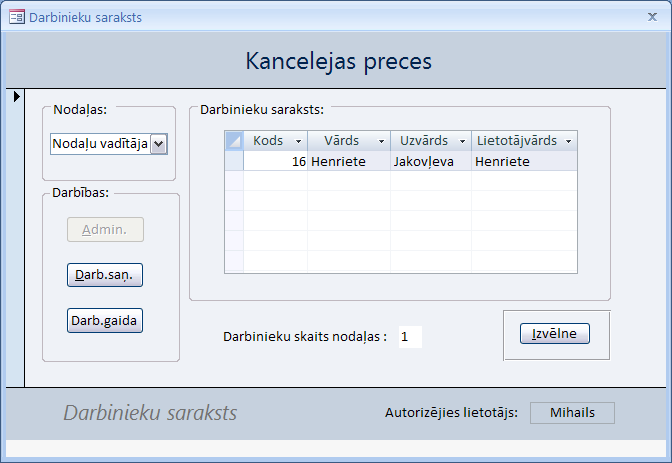
Me.subOrders.Requery()

End Sub

Dota procedūra ir līdzīga iepriekšējai, tikai šoreiz tiek dzēsti raksti, kurus lietotājs ir atzīmējis pasūtījumu sarakstā. *Supplies* mainīgais ir nepieciešams, lai izgūtu preces identifikatoru.

# Darbinieku forma

Atgriežoties pie galvenās formas un nospiežot pogu „Darbinieki”, tiks atvērta darbinieku forma (*Employers*). Šī forma ir informatīva forma, kas sniedz lietotājam pilno informācijas apkopojumu par darbiniekiem nodaļas, darbiniekiem, kuri gaida un ir saņēmuši preces. Darbinieku forma līdzīgi iepriekšējai formai Preces (*Articles*) ielādes laikā arī veic lietotāja pārbaudi (skat. 19, 20 lpp.). Formas apakšēja nostūrī ir redzams pašreiz sistēma pieteikta lietotāja lietotājvārds; pat reiz ir autorizējies Mihails, kas nav nodaļu vadītājs, tāpēc poga „Admin” ir neaktīva. Šī poga ļauj nodaļu vadītājai atvērt darbinieku administrēšanas formu, tāda veida nekāds cits darbinieks nevares izmainīt cita darbinieka paroles u.tml.



1. att. Darbinieku forma.

Formas pārslēgu grupa ar nosaukumu „Nodaļas” satur izvēlni, kuras datu avots ir *Nodaļas* vaicājums, kas atlasa visas nodaļas no *Darbinieku* tabulas. Lietotājs no šīs izvēlnes var izvēlēties kādu nodaļu, ka rezultātā pārslēgu grupas „Darbinieku saraksts” tabulā paradīsies informācija tikai par tiem darbiniekiem, kuri pieder šai nodaļai, piemēram, attēlā 34 lietotājs Mihails ir izvēlējies apskatīties informāciju par nodaļas vadītāju.

Šāds darbību klāsts tika panākts izveidojot vaicājumu *Darb\_saraksts*, kurš atlasa tikai tos darbinieku unikālos identifikatorus, lietotājvārdus, vārdus un uzvārdus, kuri ir izvēlēti izvēlne:

SELECT DarbID AS Kods, Vards AS Vards, Uzvards AS Uzvards, LietotVards AS Lietotajvards

FROM Darbinieki

WHERE Nodala=Forms!Employers!cmbDepartments.Value;

Uz vaicājuma pamata tika izstrādāta apakšforma *Darb\_Subform*, kuras īpašības tika aizliegts modificēt datus un šī apakšforma tika iekļauta darbinieku forma, izmantojot *SubForm* elementu.

Lai tabula tiktu vienmēr atjaunota, kad lietotājs izvēlas nodaļas, izvēlnes elementam *cmbDepartments* notikumā *AfterChange* tika izstrādāta atbilstoša makrokomanda (skat. att. 35).



1. att. Makrokomanda.

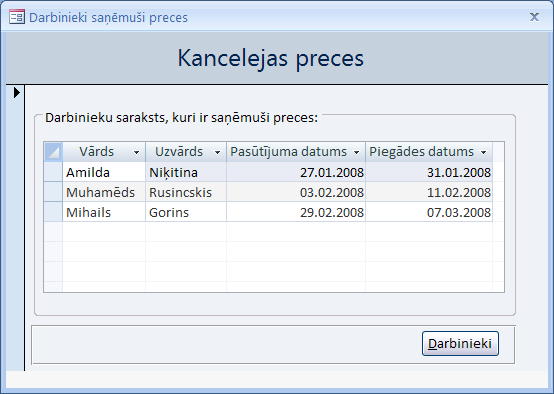
Forma satur *DCount* funkciju. Šī funkcija skaita darbinieku skaitu nodaļas, kuras izvēlnē izvēlas lietotājs. Funkcija pildās katru reizi, kad lietotājs maina nodaļu.

=DCount("DarbID";"Darbinieki";"Nodala = '" & [Forms]![Employers]![cmbDepartments].[Value] & "'")

Forma paredz divas darbības vienkāršajām lietotājam – atvērt formu par darbiniekiem, kuri ir saņēmuši preces un formu par darbiniekiem, kuri gaida preces, bez minētam darbībām tikai nodaļu vadītāja var atvērt darbinieku administrēšanas formu. Forma ļauj lietotājam atgriezties galvenajā forma. Šo darbību pamatā ir makrokomandas, kuras tika apskatītas jau iepriekš 16. lpp.

# Forma par preces saņēmušiem darbiniekiem

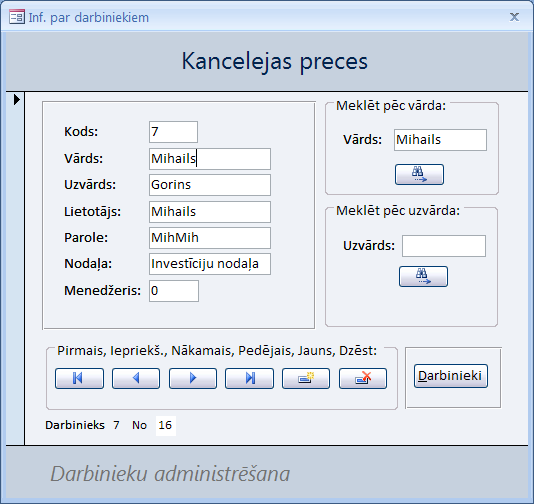
Forma izvada darbinieku sarakstu, kuri ir saņēmuši preces. Šāda informācija ir lietderīga, ja rodas kādas domstarpības starp darbinieku un nodaļas vadītāju. Saraksta paradīta informācija no tabulām *Darbinieki*, *Pasūtījumi* un *Satur*. Forma ļauj atgriezties uz darbinieku formu.



1. att. Forma par preces saņēmušiem darbiniekiem.

# Darbinieku administrēšanas forma

Darbinieku administrēšanas formas (*Admin\_emp*) galvenais uzdevums ir palīdzēt nodaļu vadītājai darbinieku administrēšanas darbos; tādi ir sameklēt visu informāciju par kādu darbinieku, mainīt datus par viņu, dzēst kādu darbinieku, ja viņš ir aizgājis no darba un pievienot jaunu darbinieku.



1. att. Darbinieku administrēšanas forma.

Formas datu avots ir tabula *Darbinieki*, tā piedāvā visus ierakstus (rakstus), kuri glabājas šajā tabulā. Lai apskatītu rakstus vai arī pievienotu un dzēstu tos, lietotājs var izmantot komandas, kuras atrodas grupā ar nosaukumu, kas raksturo katru pogu tās atrašanas secība, t.i., pirmais, iepriekšējais, nākamais, pēdējais, jauns un dzēst. Katrai pogai ir izveidota notikuma procedūra.

'pogas "pirmais" notikuma pirmteksts

Private Sub btnFIRST\_Click()

On Error GoTo Err\_btnFIRST\_Click

DoCmd.GoToRecord, , acFirst

Exit\_btnFIRST\_Click:

Exit Sub

Err\_btnFIRST\_Click:

MsgBox(Err.Description)

Resume Exit\_btnFIRST\_Click

End Sub

'pogas "ieprieksejais" notikuma pirmteksts

Private Sub btnPREVIOUS\_Click()

On Error GoTo Err\_btnPREVIOUS\_Click

DoCmd.GoToRecord, , acPrevious

Exit\_btnPREVIOUS\_Click:

Exit Sub

Err\_btnPREVIOUS\_Click:

MsgBox("Ir sasniegts pirmais ieraksts", vbOKOnly, "Darbinieku saraksts")

Resume Exit\_btnPREVIOUS\_Click

End Sub

'pogas "nakamais" notikuma pirmteksts

Private Sub btnNEXT\_Click()

On Error GoTo Err\_btnNEXT\_Click

DoCmd.GoToRecord, , acNext

Exit\_btnNEXT\_Click:

Exit Sub

Err\_btnNEXT\_Click:

MsgBox("Ir sasniegts pedejais ieraksts", vbOKOnly, "Darbinieku saraksts")

Resume Exit\_btnNEXT\_Click

End Sub

'pogas "pedejais" notikuma pirmteksts

Private Sub btnLAST\_Click()

On Error GoTo Err\_btnLAST\_Click

DoCmd.GoToRecord, , acLast

Exit\_btnLAST\_Click:

Exit Sub

Err\_btnLAST\_Click:

MsgBox(Err.Description)

Resume Exit\_btnLAST\_Click

End Sub

'pogas "dzest" notikuma pirmteksts

Private Sub btnDELETE\_Click()

On Error GoTo Err\_btnDELETE\_Click

DoCmd.DoMenuItem(acFormBar, acEditMenu, 8, , acMenuVer70)

DoCmd.DoMenuItem(acFormBar, acEditMenu, 6, , acMenuVer70)

Exit\_btnDELETE\_Click:

Exit Sub

Err\_btnDELETE\_Click:

MsgBox(Err.Description)

Resume Exit\_btnDELETE\_Click

End Sub

'pogas "jauns" notikuma pirmteksts

Private Sub btnNEW\_Click()

On Error GoTo Err\_btnNEW\_Click

DoCmd.GoToRecord, , acNewRec

Exit\_btnNEW\_Click:

Exit Sub

Err\_btnNEW\_Click:

MsgBox(Err.Description)

Resume Exit\_btnNEW\_Click

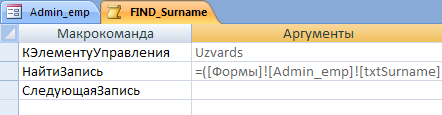
End Sub

Nav grūti pamanīt, ka visu procedūru vai arī metožu pirmteksti ir samēra vienādi, tikai katrā no tām izpildes objektam ir likts izpildīt dažādas darbības: pāriet pie iepriekšēja, pie nākama, pēdējā, pirmā, jaunā un dzēst izvēlēto rakstu. Pogas „nākamais” un „iepriekšējais” izvada paziņojumu, ja ir sasniegts pirmais vai pēdējais raksts (skat. att. 38).



1. att. Paziņojums

Lai atbrīvotu nodaļas vadītāju no rutīnu operāciju veikšanas, forma paredz divas meklēšanas operācijas: meklēt pēc vārda, vai nu pēc uzvārda. Abos gadījumos meklēšanas operācijas izpilda makrokomandas, kuru darbību klāsts ir vienāds, tikai viena makrokomanda ielasa darbinieka uzvārda vērtību no teksta lauka *txtSurname* un meklē to tabulas kolonnā *Uzvārds*, bet cita ielasa darbinieka vārdu un meklē to tabulas kolonnā *Vārds*. Zemāk redzamā attēla ir paradīta viena no šādām makrokomandām.



1. att. Makrokomanda *FIND\_Surname*.

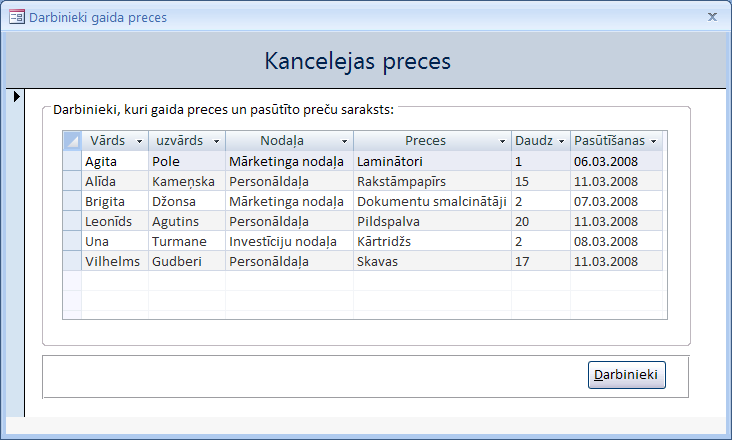
Šajā formā ir izmantota funkcija *DCount*, kas pirmajā gadījumā parāda darbinieka kārtas numuru ( piemēram 37. attēlā, Mihailam ir 7. kārtas numurs no 16 darbiniekiem), bet otrajā gadījumā tā parāda darbinieku kopēju skaitu.

=DCount("DarbID";"Darbinieki";"DarbID<=" & [Forms]![Admin\_emp]![DarbID])

=DCount("DarbID";"Darbinieki";"")

# Forma par preces nesaņēmušiem darbiniekiem

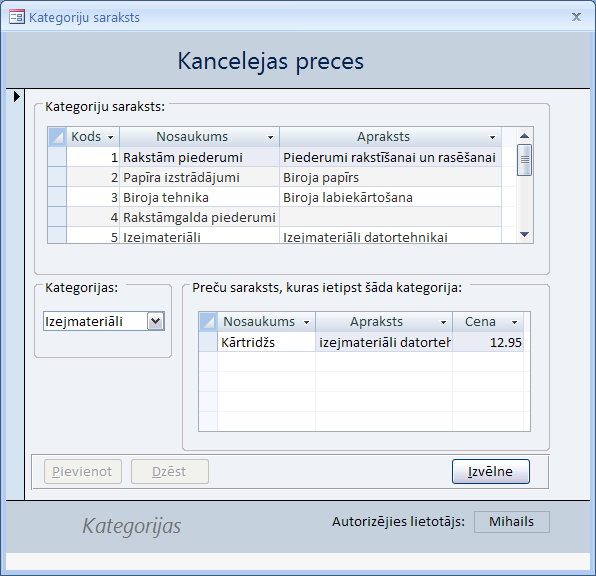
Forma izvada darbinieku sarakstu, kuri ir nav saņēmuši un joprojām gaida preces. Šāda maza forma ir noderīga informācijas ātrai pārlūkošanai, piemēram, vai šodien ir pienācis jauns pasūtījums un kāds darbinieks to ir veicis. Sarakstā pamatā ir vaicājums, kas atlasa datus no vairākām tabulām detalizētās informācijas iegūšanai. Forma ļauj atgriezties uz darbinieku formu.



1. att. Forma par preces saņēmušiem darbiniekiem.

# Kategoriju forma

Kategoriju formas (*Categories*) galvenās uzdevums ir informēt lietotāju par kategorijām, kurās var ietilpt viena vai vairākas preces. Formu veido kategoriju saraksts un preču saraksts, kurš mainās, lietotājam grupas „Kategorijas” izvēlnē izvēloties kategoriju, tāda veida lietotājs var atlasīt informāciju tikai par tām precēm, kuras viņam interese un kuras atrodas viņa noradīja kategorijā. Dota forma stipri līdzinās iepriekš apskatītajām; formas ielādes laikā tiek veikta pārbaude vai autorizējies lietotājs nav nodaļu vadītājs un jā tā nav, tad mūsu gadījumā pogas „Pievienot” un „Dzēst” kļūst neaktīvas. Formu var atvērt galvenā forma izvēloties pogu „Kategorijas”.



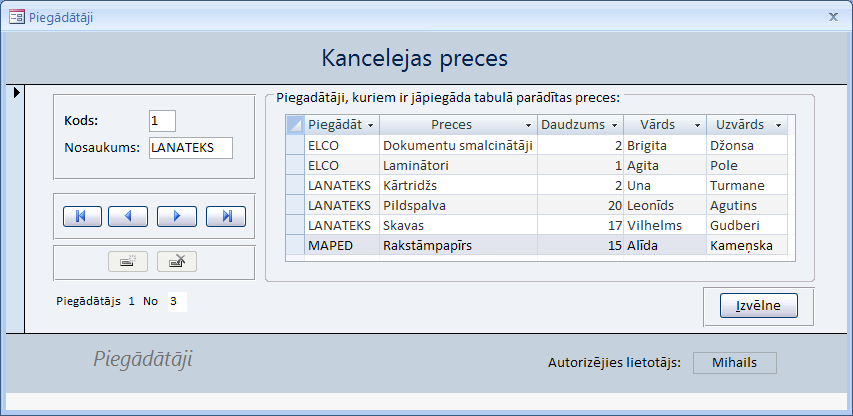
1. att. Kategorijas forma.

Preču saraksta atjaunošanos, pēc jaunas kategorijas izvēles izvēlnē, nodrošina pēc būtības līdzīga makrokomanda, kura tika apskatīta iepriekš 27. lpp, tāpat arī ar vaicājumu un apakšformu, kuri ir pamats šai preču saraksta tabulai.

Pēc pogas „Pievienot” nospiešanas, kursors pārleks uz jaunu rakstu kategoriju saraksta tabulā. Šādu darbību nodrošina makrokomanda, kura ir līdzīga 20. – 21. lpp. apskatītai makrokomandai *NEW\_Article*. Poga „Dzēst” satur notikuma procedūru, kas ar ne ar ko neatšķiras no 21. lpp aplūkotas, vienīgi šeit raksts tiek dzēsts no kategoriju saraksta tabulas.

# Piegādātāju forma

Piegādātāju forma ir domāta informācijas par piegādātājiem apskaitīšanai un labošanai, jauna piegādātāja ievadīšanai un dzēšanai. Jāpiemin, ka forma satur sarakstu, kurā ērti var redzēt piegādātājus, kuriem ir jāpiegādā blakus parādītas preces. Forma neļauj nodaļas darbiniekiem ievadīt un dzēst piegādātājus, to var izdarīt vienīgi nodaļu vadītajā.



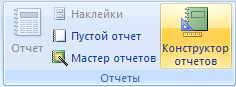
1. att. Piegādātāju forma.

Formas datu avots ir *Piegādātāji* tabula. Šī forma paredz tādas pašas darbības un funkcijas *DCount* pielietojumu, kuras tika apskatītas darbinieku administrēšanas formas nodaļā, tieši 28, 29, 30. lpp.

# Pārskati

Pārskati ir labs risinājums tabulu un vaicājumu datu apkopošanai, ļaujot iegūt pārskatāmu atskaiti, kuru nepieciešamības gadījuma var izdrukāt uz papīra.

Lai izveidotu pārskatu MS Access 2007 nepieciešams rīkjosla *Create* izvēlēties veidot pārskatu, izmantojot vai nu *Report Wizard*, vai arī *Design view* režīmu (skat. att. 43).

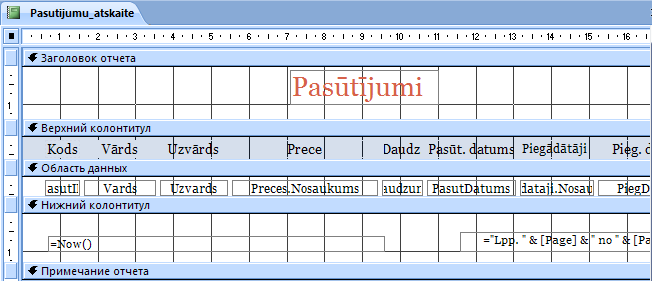


1. att. Konstruktors (*Design view*).

Izvēloties *Design view* režīmu, tiks atvērts tukšs pārskats, savukārt, *Report Wizard* piedāvās vairākas izvēles iespējas, ka rezultātā var iegūt jau gatavu atskaiti, kuru nepieciešamības gadījuma var koriģēt *Design* *view* režīmā.

Veidojot pasūtījumu un izmaksu pārskatus, es izvēlējos veidot pārskatu, izmantojot *Report Wizard* un labojumus veicu *Design view* režīmā. Pirmajā solī man tika piedāvāts norādīt datu avotu; pasūtījumu atskaitei es izvelējos *Pasut\_Report* vaicājumu, kas ietver sevī visu nepieciešamo informāciju par darbinieku veiktajiem pasūtījumiem, savukārt, izmaksu pārskatām izvēlējos *Izmaksas\_Report* vaicājumu, kurš satur informāciju par izmaksām.

Turpmākajos soļos izvēlējos laukus pēc kuriem ir jāveic grupēšana un izskatu, kādu iegūs pārskati, kad darbs ar *Report Wizard* būs beigts. Iegūtais pārskats ir paradīts attēlā 44.



1. att. pārskats *Design view* režīmā.

Pēc dažiem labojumiem, kas tika veikti *Design view* režīmā, piesaistīju izveidotos pārskatus galvenajai formai ar makrokomandu palīdzību (skat. 17. lpp.).



1. att. Izmaksu pārskats.



1. att. Pasūtījumu pārskats.

# SECINĀJUMI

Laboratorijas darba galvenais uzdevums bija iepazīties ar formu un pārskatu veidošanu MS Access vidē. Izstrādājot darbu, izpratu formu būtību, to pielietošanas principus, formu un tās vadības elementu īpašības un notikumus, plašāk iepazinu makrokomandas un *Visual Basic for Application* programmēšanas valodu un funkcijas. Noformējot pārskatus un formas, ieguvu vairākas jaunas iemaņas praktisko darbu noformēšana

Darba gaitā nācās sastapties ar vairākām nelielām grūtībām, kuras daļēji tika rādītas manas iniciatīvas dēļ, jo vēlējos iegūt pilnvērtīgu informācijas sistēmu, kura spētu apmierināt kādas nelielas kompānijas vajadzības. Pie tādam grūtībām var pieskaitīt globālo mainīgo un publisku metožu glabāšana moduļos, datu dzēšana no apakšformām, ka arī rakstot pirmtekstu VBA nelielas grūtības sagādāja dažas atšķirības no *Visual Basic* valodas, ka, piemēram, Access vidē vadības elementam laukam vērtības iegūšanas īpašība ir *value*, savukārt, *Visual Basic* šī īpašība saucas *text*.

Par lielu ieguvu uzskatu, ka esmu iemācījies praktiski pielietot vairākas darba izpildes gaitā uzzinātas zināšanas, pie tādam zināšanām jāpiemin DAO datu piekļuves objektu, kas ir vecāks par ADO un parādījās Office 97 versijā un praktiskas iemaņas šādā darba izstrādei Office 2007 vidē, ko biju darījis pirmo reizi.

Darba rezultātā tika iegūta mazā informācijas sistēma, kuru uzskatu ka esmu realizējis veiksmīgi, bet zinu, ka ir arī dažs trūkums, t.i., procedūra, kura veic autorizēta lietotāja lietotājvārda pārbaudi attiecība uz nodaļu vadītāju, varētu izgūt nodaļas darbinieka nevis lietotājvārdu, bet unikālo identifikatoru. Nožēlu, ka esmu mazliet nokavējies ar darba nodošanu, jo darbs izrādījās diezgan apjomīgs.

# IZMANTOTA LITERATŪRA

1. Viktorija Vinogradova. Lekciju konspekti studiju priekšmetā „Datu bāzes vadības sistēma” – Rīga, RTU, 2007.
2. Internets: <http://msdn2.microsoft.com/en-us/library/bb979619.aspx>