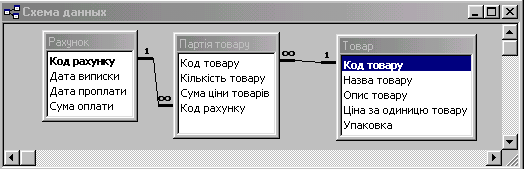
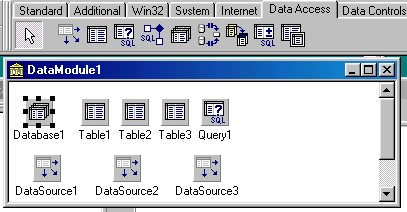
**Шевчук Андрій. ПМП-42, Львів – 2000**

**Створення клієнтської програми для користування базою данних MS ACCESS в Delphi 4.0**

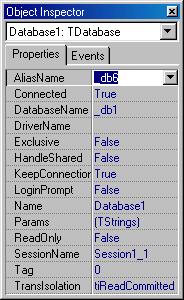
Створимо програму, яка буде прикладом роботи з базами даних в Delphi 4.0. Для цього використаємо базу даних Ms Access db1.mdb, в якій знаходяться наступні таблиці: “Товар» (таблиця, що містить дані про товар, його опис і ціну), «Партія товару» і “Рахунок”, які зв'язані наступним чином:



Для початку зареструємо псевдонім для бази даних в ODBC. Для цього перейдемо в папку “Панель управления” Пуск Настройки Панель управления (Start  Settings  Control Panel) і двічі натиснемо на ярличку “Источники данных ODBC 32 разряда” (“ODBC 32-bit”), в вкладці «Пользовательский DSN» (User DSN), натиснемо на кнопку “Добавить” (Add), виберемо зі списку “Microsoft Access Driver (\*.mdb)” і натиснемо “Готово” (Finish). У вікні, що відкриється введемо “Имя источника базы данных” (Description) текст “\_db6” і натиснувши на кнопку “Выбрать” (Select) вкажемо шлях до нашої бази даних: “С:\DB\db1.mdb”. Тиснемо “ОК” і переходимо безпосередньо в Delphi 4.0. Для створення програми виберемо в меню File  New Application. Створимо DataModule необхідний для розміщення на ньому компонент для роботи з базами даних, для цього виберемо в меню File  New, в вікні що з'явилось вибираємо піктограмку “DataModule”. Збережемо проект як db.dpr, DataModule1 data.pas, unit1 dp1.pas. Додамо з палітри компонентів з вкладки “Data Access” компоненти необхідні для роботи з нашою базою даних Database, три компоненти DataSource, три компоненти Table i Query помістивши їх на DataModule1:

Для компоненти Database1 в вікні “Object Inspector” надамо наступні значення для властивостей: AliasName = “\_db6” (вказуємо псевдонім нашої бази даних), DatabaseName = “\_db1” (ім'я бази даних яке будуть використовувати решта компоненти DataAccess), LoginPromt = “False” (не запрошувати ім'я користувача і пароль при відкритті бази даних). Для компонент Table1, Table2, Table3 встановимо DatabasName = “\_db1”,TableName відповідно “Товар”, “Рахунок”, “Партія товару”, і після цього для кожної компоненти Active присвоїмо значення True. Для компонент Datasource1, DataSource2, Datasource3 присвоїмо властивості DataSet відповідно значень “Table1”, “Table2”, “Table3”.

З вкладки Win32 на форму From1 помістимо компоненту PageControl, в вікні “Object Inspector” якої вкажемо Align = “AllClient”. Викликавши правою кнопкою миші контекстне меню виберемо пункт “New Page” і для нової закладки TabSheet1 властивість Caption присвоїмо значення “Товари”, яке буде назвою нашої закладки. На закладку помістимо компоненту DBGrid1 з закладки DataControls палітри компонентів. Надамо їй наступних властивостей: Align = “allClient”, DataSource = “DataModule1.DataSource1”. Тепер в вікні закладки “Товари” ми можемо переглядати дані з таблиці товар. Для можливості оперування з цими даними додамо на форму елемент управління DBNavigator. Для цього створимо для нашої форми Form1 панель інструментів. Помістимо на форму CoolBar1 з закладки Win32, на CoolBar1 ControlBar1 з закладки Additional, властивості AutoSize яких встановимо “True”. Після цього на ControlBar1 помістимо DBNavigator1. Створимо ще одну закладку PageControl1, для перегляду і редагування всіх трьох таблиць. Присвоїмо властивості Caption = «Виписка рахунку», для TabSheet2, і помістимо на закладку три компоненти GroupBox з палітри “Standart”, присвоївши властивості Caption для яких, відповідно, “Рахунок”, “Партія товару” і “Товар”. Помістимо на GroupBox1 (“Рахунок”) чотири компоненти DBEdit, з закладки “Data Controls”. Властивостям цих елементів DataSource присвоїмо значення DataModule1.DataSource3, а DataField, відповідно, імена полів з таблиці “Рахунок”: “Код рахунку”, “Дата виписки”, “Дата проплати”, “Сума оплати”. Над елементами DBEdit помістимо компоненти Label з палітри “Satndart”, присвоївши властивості Caption кожного з них імена відповідних полів. Аналогічно поступимо з GroupBox2 (“Товар”), помістивши на нього відповідну кількість DBEdit i Label, створивши перегляд таблиці “Рахунок”. На GroupBox3 (“Партія товару”) помістимо DBGrid, присвоїмо його властивостям такі значення: Align = AllClient, DataSource = DataModule1.DataSource2. Для того щоб перехід по всіх полях різних таблиць відбувався синхронно, а в DBGrid2 ми могли бачити список партій товару які виписані на вибраний рахунок вкажемо зв'язки між таблицями. Але щоб в першій закладці таблиця товар відображалась правильно зробимо це для таблиць “Рахунок” і “Партія товару” в «Object Inspector», а для таблиць “Партія товару” і “Товар” – програмно. Виберемо Table2 на DataModule1 і для властивості MasterSource виберемо значення DataSource3, для властивості MasterFields – “Код рахунку”. Для компоненти PageControl1 в колонці “Object Inspector” “Events” виберемо подію «OnChange», двічі натиснемо ліву клавішу миші і введемо наступне:



procedure TForm1.PageControl1Change(Sender: TObject);

begin

// якщо здійснено перехід на закладку “Товар”

if PageControl1.ActivePage=TabSheet1 then

begin

DBNavigator1.DataSource:=DataModule1.DataSource1; // панель DBNavigator керує таблицею “Товар”

DataModule1.Table1.MasterSource:=nil; // забираємо зв'язок з таблицею “Партія товару”

DataModule1.Table1.MasterFields:='';

end;

// якщо здіймнено перехід на закладку “Партія товару”

if PageControl1.ActivePage=TabSheet2 then

begin

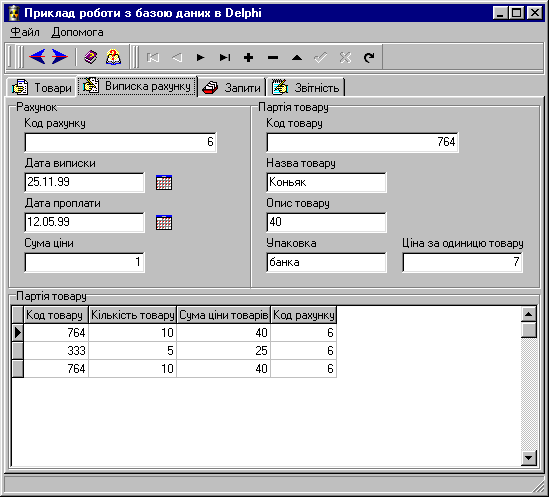
DBNavigator1.DataSource:=DataModule1.DataSource3; // панель DBNavigator керує таблицею “Рахунок”

DataModule1.Table1.MasterSource:=DataModule1.DataSource2; // встановлюємо зв'язок між «Товар» і

DataModule1.Table1.MasterFields:='Код товару'; // “Партія товару” по полю “Код товару”

end;

end;



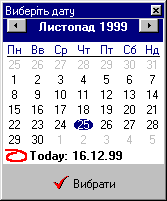
Біля полів “Дата виписки» та «Дата проплати» помістимо кнопки для переходу на нову форму – вибору дати. Створимо її у вигляді dll. Виберемо в меню меню File New, і вікні “New Items” виберемо піктограму “Dll”. Після появи самого тексту dll збережемо проект як «slave.dpr», і виберемо в меню File New Form. Властивостям нової форми надамо такі значення: Name = CalendarForm, Caption = «Виберіть дату», BorderStyle = bsToolWindow, Position = poDesktopCenter. Помістимо на форму компоненту MonthCalendar з закладки Win32, і SpeedButton з закладки Additional. Додамо в опис змінних для модуля :

var

selDate: TDate;

для SpeedButton1 створимо таку процедуру обробки події OnClick:

procedure TCalendarForm.SpeedButton1Click(Sender: TObject);



begin

selDate:=MonthCalendar1.Date; // запам'ятовуємо вибрану дату

CalendarForm.Close; // закриваємо форму

end;

В модуль slave.dpr введемо наступний текст:

uses

Forms,

SysUtils,

Classes,

cal in 'cal.pas' {CalendarForm};

function GetDate: TDateTime; stdcall;

begin

Application.CreateForm(TCalendarForm, CalendarForm); // динамічно створюємо форму

CalendarForm.ShowModal; // виводимо форму на екран

GetDate:=selDate;// значенню що повертає функція присвоюємо значення вибраної дати

end;

exports

GetDate; // функція для зовнішнього використання

begin

end.

Так як для більшої швидкості виконання програми, форми створюються динамічно, то відповідно в властивостях проекта (Projects  Properties) потрібно забирати форми з поля “AutoCreate Forms”.

Для SpeedButton1 i SpeedButton2 в основній програмі, в процедуру обробки події OnClick вставимо наступний текст:

procedure TForm1.SpeedButton1Click(Sender: TObject);

begin

DBEdit1.Text:=DateToStr(GetDate);

end;

procedure TForm1.SpeedButton2Click(Sender: TObject);

begin

DBEdit2.Text:=DateToStr(GetDate);

end;

Відповідно перед секцією Implementation вставимо наступне:

{export from slave.dll}

function GetDate: TDateTime; stdcall; external 'slave.dll';

Тепер при натисненні на кнопку біля відповідних полів користувач зможе вибрати дату з календаря.

Створимо нову закладку PageControl1, назвемо її “Запити”. Створимо для цієї закладки нову панель інструментів, помістивши на ControlBar1 ToolBar1. Додамо декілька кнопок викликаючи контектне меню на ToolBar1 і вибираючи опцію New Button. Помістимо на закладку DBGrid для перегляду результатів запитів. Помістимо на закладку GroupBox, на якому розташуємо для початку 6 компонент Edit, над якими розмістимо по Label з назвами відповідних полів з таблиці “Товар”. Розташуємо між двома останніми полями ComboBox1. Будемо використовувати відповідні поля для вводу критеріїв для запитів. Перш за все зробимо вікно в якому ми зможемо контролювати правильність створення запиту SQL. Виберемо в меню команду File New Form, надамо її властивостям наступних значень: Name = SQLText, BorderStyle = bsToolWindow, Position = poDesktopCenter. Помістимо на форму Memo1, змінимо властивість Align = AllClient. Для Query1 з DataModule1 введемо такий текст для обробки події AfterOpen:

var

SQLtxt:TStrings;

…

procedure TDataModule1.Query1AfterOpen(DataSet: TDataSet);

begin

SQLtxt:=DataModule1.Query1.SQL;

end;

a для події кнопки OnClick, якій ми присвоїмо Hint “Показати текст виконаного запиту”, цей напис буде з'являтись при наведенні курсору миші на кнопку (властивість ShowHints для відображення вспливаючих підказок повинна мати значення True, для присвоєння всім елементам цього значення, достатньо присвоїти ShowHints = True батьківському елементу, наприклад Form1):

procedure TForm1.ToolButton10Click(Sender: TObject);

begin

Application.CreateForm(TSQLText, SQLText); // динамічно створюємо форму

SQLText.Memo1.Lines:=SQLtxt; // виводимо текст запиту

SQLText.ShowModal; // показуємо форму

SQLText.Close;

end;

Для DBGrid3 властивості DataSource присвоїмо значення DataModule1.DataSource4, властивості DataSource4 DataSet = Query1. Щоб при завантаженні програми DBGrid не був пустим відобразимо в ньому таблицю «Товар» наступним чином – властивості DatabaseName Query1 надамо значення “\_db1” в полі SQL викликавши вікно введемо текст: “select \* from Товар;” (без лапок).

Для першої кнопки з ToolBar1 – ToolButton1 в полі властивості Hint напишемо настпуне – “Шукати записи по введених критеріях”, створимо для цієї кнопки виконання запиту з використанням інструкції SQL select. Перед цим ComboBox1 надамо таких значень для вибору в властивості «Items»: “=”,”>”,”<”,”..”. Для події OnChange ComboBox1 введемо наступне:

procedure TForm1.ComboBox1Change(Sender: TObject);

begin

if ComboBox1.ItemIndex=3 then Edit12.Visible:=True // при виборі значення “..” з'являється поле для //вказання інтервалу відбору значень

else Edit12.Visible:=False;

end;

Для обробки події OnClick ToolButton1 введемо наступний текст:

const

t1:string='[Код товару]'; // назви полів

t2:string='[Назва товару]';

t3:string='[Опис товару]';

t5:string='[Ціна за одиницю товару]';

t4:string='[Упаковка]';

…

procedure TForm1.ToolButton3Click(Sender: TObject);

var sqlTable,sqlWhere: String;

begin

DataModule1.Query1.SQL.Clear;

sqlTable:='Select \* ';

sqlWhere:='where ';

sqlTable:=sqlTable + 'from Товар ';

sqlWhere:=sqlWhere+'('+t1+' like '''+'%'+Edit1.Text+''') and ('+t2+' like '''+Edit2.Text+''') and ('+

t3+' like '''+Edit3.Text+''') and ('+t4+' like '''+Edit4.Text+''') and (';

case ComboBox1.ItemIndex of

0: sqlWhere:=sqlWhere+t5+' = '+Edit5.Text+');'; // рівне вказаному значенню

1: sqlWhere:=sqlWhere+t5+' > '+Edit5.Text+');'; // більше

2: sqlWhere:=sqlWhere+t5+' < '+Edit5.Text+');'; // менше

3: sqlWhere:=sqlWhere+t5+' between '+Edit5.Text+' and '+Edit12.Text+');'; // в інтервалі

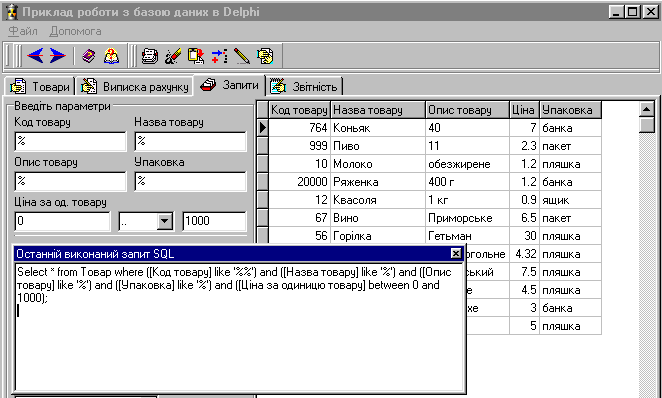
end;

DataModule1.Query1.SQL.Add(sqlTable+sqlWhere);

DataModule1.Query1.Active:=True;

end;

Символ процента використовується в мові SQL драйвера MS ACCESS для вибору всіх значень з предикатом Like (наприклад like’a%’ поверне всі значення що починаються з букви а), тому по замовчуванні для Edit полів таблиці товар задамо значення Text = %, для поля «Ціна за одиницю товару” Text = 0, для Edit12, поля в якому вказуэться верхня межа по замовчуванны введемо Text = 1000. Запустимо програму і перейшовши на закладку “Запити” натиснемо кнопку з підказкою “Шукати записи по введених критеріях”, після чого натиснемо на кнопку для перегляду тексту запиту з підказкою “Показати текст виконаного запиту”, отримаємо наступне:



Аналогічно створимо на панелі інструментів кнопки, яким присвоїмо властивість Hint відповідно «Знищити записи, що задовільняють критеріям», «Створити нову таблицю на основі записів, що задовільняють критеріям», «Додати новий запис» і «Ввести запит самому», для яких введемо наступний текст обробки подій OnClick:

procedure TForm1.ToolButton4Click(Sender: TObject); // знищення записів

var sqlTable,sqlWhere: String;

begin

if MessageDlg('Ви впевненні що хочете знищити записи, що задовільняють введеним даним?',

mtConfirmation, [mbYes, mbNo], 0) = mrYes then

begin

DataModule1.Query1.SQL.Clear;

sqlTable:='Delete \* ';

sqlWhere:='where ';

sqlTable:=sqlTable + 'from Товар ';

sqlWhere:=sqlWhere+'('+t1+' like '''+'%'+Edit1.Text+''') and ('+t2+' like '''+Edit2.Text+''') and ('+

t3+' like '''+Edit3.Text+''') and ('+t4+' like '''+Edit4.Text+''') and (';

case ComboBox1.ItemIndex of

0: sqlWhere:=sqlWhere+t5+' = '+Edit5.Text+');';

1: sqlWhere:=sqlWhere+t5+' > '+Edit5.Text+');';

2: sqlWhere:=sqlWhere+t5+' < '+Edit5.Text+');';

3: sqlWhere:=sqlWhere+t5+' between '+Edit5.Text+' and '+Edit12.Text+');';

end;

DataModule1.Query1.SQL.Add(sqlTable+sqlWhere);

DataModule1.Query1.ExecSQL;

end;

end;

procedure TForm1.ToolButton9Click(Sender: TObject); // додання записів

var sqlInsert: String;

begin

DataModule1.Query1.SQL.Clear;

sqlInsert:='Insert into Товар Values ('''+'%'+Edit1.Text+''','''+Edit2.Text+''','''+

Edit3.Text+''','''+Edit5.Text+''','''+Edit4.Text+''');';

DataModule1.Query1.SQL.Add(sqlInsert);

DataModule1.Query1.ExecSQL;

end;

procedure TForm1.ToolButton7Click(Sender: TObject); // створення нової таблиці

var TableName,sqlCreate:string;

begin

DataModule1.Query1.SQL.Clear;

TableName:=InputBox('Введіть ім''я нової таблиці','','');

sqlCreate:='Select \* into '+TableName+' from Товар where ('+t1+' like '''+'%'+Edit1.Text+''') and ('+t2+' like '''+Edit2.Text+''') and ('+

t3+' like '''+Edit3.Text+''') and ('+t4+' like '''+Edit4.Text+''') and (';

case ComboBox1.ItemIndex of

0: sqlCreate:=sqlCreate+t5+' = '+Edit5.Text+');';

1: sqlCreate:=sqlCreate+t5+' > '+Edit5.Text+');';

2: sqlCreate:=sqlCreate+t5+' < '+Edit5.Text+');';

3: sqlCreate:=sqlCreate+t5+' between '+Edit5.Text+' and '+Edit12.Text+');';

end;

DataModule1.Query1.SQL.Add(sqlCreate);

DataModule1.Query1.ExecSQL;

DataModule1.Query1.SQL.Clear;

DataModule1.Query1.SQL.Add('Select \* from '+TableName);

DataModule1.Query1.Active:=True;

end;

procedure TForm1.ToolButton11Click(Sender: TObject); // самостійне введення запиту

begin

with DataModule1.Query1 do

begin

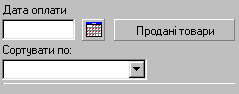
SQL.Clear;

SQL.Add(InputBox('Введіть запит','',''));

Active:=True;

end;

end;

Наведемо приклад створення запиту з двох таблиць. Помістимо на GroupBox Edit9, ComboBox3 i дві компоненти SpeedButton. Одна з них буде використовувати нашу функцію з dll GetDate. Запит повинен вибирати за введеною датою товари з таблиці «Товар» на основі дати виписки рахунку з таблиці «Рахунок». В ComboBox введемо такі елементи для вибору: Рахунок.[Сума оплати], Товар.[Назва товару], Рахунок.[Дата проплати], які користувач може вибрати для вибору сортування результатів. Введемо наступний текст для обробки події OnClick для SpeedButton4 (вибір дати) і SpeedButton6 (вивід результатів запиту):

procedure TForm1.SpeedButton4Click(Sender: TObject);

begin

Edit9.Text:=DateToStr(GetDate);

end;

procedure TForm1.SpeedButton6Click(Sender: TObject);

var sqlStr: String; sDate: string; fRep:TReplaceFlags;

begin

if Edit9.Text='' then

begin

ShowMessage('Потрібно вибрати дату');

exit;

end;

sDate:=StringReplace(Edit9.Text,'.','/',fRep);

sDate:=StringReplace(sDate,'.','/',fRep);

DataModule1.Query1.SQL.Clear;

sqlStr:='Select Товар.[Назва товару],Товар.[Опис товару],[Партія товару].[Код рахунку],Рахунок.[Дата проплати],Рахунок.[Дата виписки],Рахунок.[Сума оплати] from Товар inner join ([Партія товару] inner join '+

'Рахунок on Рахунок.[Код рахунку]=[Партія товару].[Код рахунку]) on '+

'Товар.[Код товару]=[Партія товару].[Код товару] where Рахунок.[Дата виписки]=#'+sDate+'#';

if ComboBox3.Items[ComboBox3.ItemIndex]<>'' then sqlStr:=sqlStr+' Order By '+ComboBox3.Items[ComboBox3.ItemIndex];

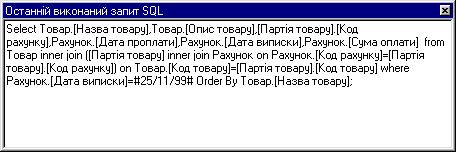
sqlStr:=sqlStr+';';

DataModule1.Query1.SQL.Add(sqlStr);

DataModule1.Query1.Active:=True;

end;

Текст нашого запиту при введеній даті і сортуванні по назві товару буде мати вигляд:



Перейдемо до створення останньої закладки програми – “Звітність”, в якій проілюструємо роботу з QuickReports. Виберемо в меню опцію File New, в вікні вибору натиснемо на піктограмці “Report”. Збережемо новий модуль, і заберемо його з опцій проекту, як форму що створюється автоматично. На нашу закладку звітність помістимо GroupBox, Сaption якого присвоїмо “Прайс лист товарів”. Помістимо на GroupBox кнопку «Перегляд і друк», події OnClick якої присвоїмо наступне:

procedure TForm1.SpeedButton7Click(Sender: TObject);

begin

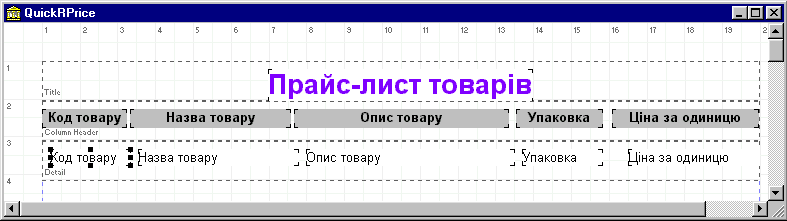
Application.CreateForm(TQuickRPrice, QuickRPrice);

QuickRPrice.Preview;

QuickRPrice.Destroy;

end;

Наш QuickReport1 переіменуємо в QuickRPrice, його властивості DataSet надамо значення DataModule1.Table1. Відкривши в властивостях групу “Bands” надамо значення True властивостям HasTitle (назва звіту), HasColumnHeader (загаловки колонок), HasDetail (самі дані, значення полів таблиці). З закладки QReports на Title помістимо QRLabel, з Caption = «Прайс-лист товарів», на ColumnHeader п'ять компонент QRLabel з назвами полів нашої таблиці, а на частину Detail п'ять QRDBText, властивостям яких DataSet надамо значення DataModule1.Table1, а DataField відповідно поля нашої таблиці:



На цьому закінчимо розробку першого звіту. Помістимо на закладку звітність ще один GroupBox, який назвемо відомості про продаж товарів, і організуємо як складний звіт з деталізацією інформації з різних таблиць. Для кнопки «Перегляд і друк» звіту введемо такий текст програми:

procedure TForm1.SpeedButton9Click(Sender: TObject);

begin

DataModule1.Table1.MasterSource:=DataModule1.DataSource2;

DataModule1.Table1.MasterFields:='Код товару';

Application.CreateForm(TQuickRSum, QuickRSum);

QuickRSum.Preview;

QuickRSum.Destroy;

DataModule1.Table1.MasterSource:=nil;

DataModule1.Table1.MasterFields:='';

end;

Створимо новий Report, назвемо його QuickRSum, властивість DataSet присвоїмо DataModule1.Table1, з групи Bands присвоїмо значення True властивостям HasTitle, HasColumnHeader, HasDetail. Помістимо на Report QRBand i QRSubDetail. На розділ Title помістимо QRLabel “Відомості про продаж товарів”, на ColumnHeader QRLabel з назвами полів таблиці «Товар», на частину Detail QRDBText відповідно з властивостю DataSet DataModule1.Table1, a DataField з назвами відповідних полів таблиці. На частину GroupHeader помістимо назви полів таблиці “Рахунок”, відповідно на частину SubDetail QRDBText з DataSet = DataModule1.Table3, i DataFiled яких надамо назви полів таблиці “Рахунок”:

Викликавши контекстне меню правою кнопкою миші можна подивитись попередій перегляд звіту, але, наприклад, в нашому випадку, звіт буде відображатись неправильно в режимі розробки програми, тому що зв'язок між таблицями «Товар» і «Партія товару» ми встановлюємо під час роботи програми. Скомпілювавши і запустивши програму ми зможемо переглянути звіт в нормальному вигляді натиснувши кнопку «Перегляд і друк».

