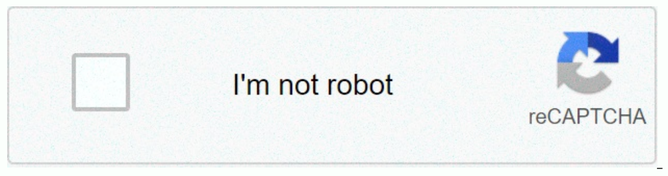

Continue

Vb pdf viewer

ライブカメラをロードしています。 Hello, this time we'll report the best-selling products by date range, for example, the last 7 days, the current month, or custom dates. We will use layered architecture and object-oriented programming, with Visual Basic, Windows Form, Report Viewer control, local RDLC report designer, and SQL Server. Tutorial Well, let's start creating the database and ☺ database we have 4 related sales order tables and thousands of thousands of data additions. You can download the full script from the database from the following window so that you can practice. create a relative database (BDR) script diagram create a BikeStore database create table products (product_id identity int (1.1) primary key, product_name varchar (200) not NULL, model_year smallint not NULL, decimal price (10, 2) not NULL); Create table clients (customer_id INT identity (1, 1) primary key, first_name VARCHAR (255) not NULL, last_name VARCHAR (255) not NULL, Phone VARCHAR (25), e-mail VARCHAR (255) not NULL, VARCHAR Street (255), VARCHAR City (50), state VARCHAR (25), zip_code CHAR VARCHAR (5) create table orders (order_id identity int (1.1) primary key, customer_id int not null, order_date date not null, Foreign FK_Customer constraint (customer_id) refers to clients(customer_id); order_items table (order_item_id identity int (1.1), order_id int not null, product_id INT NOT NULL, INT NOT NULL Quantity, Decimal Price (10, 2) Not NULL, Decimal Discount (4, 2) No NULL Default 0, Foreign Key Fk_Order Constraint (order_id) Order References(order_id) , forcing fk_Product foreign key (product_id) product references(product_id); Download Full Database + Add SQL Records Query/Procedure – Report best-selling products by date range in the following query, we'll see product ID, product name, product unit base price, quantity sold, and product revenue amount. If you want, you can create a stored procedure to view the query (see tab 2) and run the procedure from the app's data access group. Select 10 p.product_id, p.product_name, p.price as unitBasePrice, Amount (oi.quantity) as price quantities, Total amount (oi.price*oi.quantity)-oi.discount) total from orders or internal combination order_items Oi on internal join products oi.order_id-o.order_id p in p.product_id-oi.product_id where o.order_date oi.product_id between the hours 23:59:59 2019 23:59:59 2019 23:59:59 2019 23:59:59 by p.product_id, p.product_name, p.price order by quantitiesMassed desc create proc GetBestSellingProducts @startDate date, date @endDate as top choice 10 p.product_id, p.product_name, p.price as a unitBasePrice, amount(oi.quantity) as Total amount ((oi.price*oi.quantity)-oi.discount) total orders or inner join order_items oi by oi.order_id-o.order_id internal join products p on p.product_id-oi.product_id where o.order_date between @startDate and @endDate group by p.product_id, p.product_name, p.price order by quantitSold desc go application after creating the database, create the 4 related tables, add the sales order records, check the previous query to get the sales in detail or created the stored procedure, let's move to build the application. Create a layered project – Visual Studio We create a blank Visual Studio solution (New-> > > Project-> Templates-> Other types of projects -> Visual Studio Solutions-> Blank Solution) and name it as desired. We added a new Windows Form Application project for Layer.We added a new department library project for the domain or business layer. Adding cross-layer references (dependencises) We add cross-layer references based on architecture in traditional layers, with a layer above only familiar with the layer below it. In a presentation layer: We refer to a Layer.In layer: we refer to the data access layer. After you create and add cross-layer references, we'll switch to layer encoding. Connection Class Data Access Layer (ConnectionSQL.vb Let's add an abstract class (MustInherit) to connect to SQL Server.We import the system. Data.SqlClient.We have created a protected SqlConnection method to get the connection. Using the generated method, we simply return a connection instance to sql, as a parameter, send the connection string, to the local server, specify the database, and connect by using Windows credentials. Importing system.data.SqlClient public MustInherit class connectionsSQL protected getConnection function() as SqlConnection return new SqlConnection(server(on-site); database bike shop; true security class integrated object order) Function End Class Data Access Object Order (OrderDao.vb) We add another public class for the entity's data access object sales orders. We note that the class inherits from the SQL connection wait. We import System.DataSqlClient.We create a Method of DataTable type or list to get the best-selling products by <Object> Date Range, for this we create 2 parameters, one for the start date and the other for the query end date. We added the following codes to execute the query. System Import.data.SqlClient Public Class OrderDao Inherits ConnectionsQL Public Function </Object>as date and time, to describe date and time) as a data table using connection = getConnection() connection. Open() using the = New SqlCommand() command. Connection = Connection command. CommandText = GetBestSellingProducts command. Parameters.Add with value (@startDate, from) command. Parameters.AddWithValue (@endDate, toDate) commandType=CommandType.StoredProcedure Dim reader = command. Dim Data Reader Table = New Data Table Table Table Table Table. Charging reader(reader). Dispose() of the end return table by using the End Class Import System of the Edge function.data.SqlClient Public Class OrderDao inherits ConnectionSQL public function GetBestSellingProducts (from date and time, to date and time) as a data table by using connection=GetConnection() connection. Open() using the = New SqlCommand() command. connection = command.CommandText = select up(10) oi.product_id, p.product_name, p.price as unitBasePrice, Amount (quantity) quantity, amount(oi.price*oi.quantity)-oi.discount)beanies from order_items oi inner join orders o in o.order_id =oi.order_id internal join products p on p.product_id = oi.product_id where o.order_date between @startDate and @endDate group by oi.product_id, p.product_name, p.price order by desc command quantity. Parameters.add(@startDate, SqlDbType.Date). Value = from the Date command. Parameters.add(@endDate, SqlDbType.Date). Value = toDate command. CommandType = CommandType.Text Dim reader = Command. Dim Data Reader Table = New Data Table Table Table Table Table. Charging reader(reader). Get rid of() the end return table by ending the capa de Dominio class end function – Negocio Ahora codificaremos la capa de dominio o negocio, agregamos las siguientes clases/Objetos para guardar el listado de los productos mas vendidos: Clase Meres productos (BestProductjos.vb) BestProducts Public Asset ProductId class as an intrinsic number of public asset productname as a public asset price string as the amount of public property doubles as the public property amount and in whole number as the double end class Clase Reporte de Ventas (SalesReport.vb) Agregamos una clase para reporte de ventas. Imports of Amos La Capa de Exo Detos Detosdcalarmos Los Atributos y Profiades Naxarius Para El Doh. Declaramos una lista de objetos de tipo mejores productos (BestProducts.vb) para almacenar la consulta de la base de datos. A final cream of a chilly Matodo Farah to the E-Sisiser Los Obajtos report. Import data Access public class sales report 'privacy features _reportDate like date and time private _startDate like date and time private _endDAte as date and time private _listBestProducts as list (of best products) private _total as double 'propiertes public property report Date Get Return _reportDate End Get Private Group (Value as Date) _reportDate = End Set Value Public Property StartDate as Return Date _startDate End Get Private Set (Value as Date) _startDate = End Value Setting Public End Property End of date get return _endDAte end to get private group (value as date) _endDAte ? End value set end of public asset list (of best products) get refunded _listBestProducts end get private kit (value as list(of BestProduct)) _listBestProducts ? End value set public asset end amount as duplicate get return _total end get private set (value like double _total) 'builders' public practices sub GetBestSellingProductsReport(dated date and time, to schedule as date and time) 'implement ReportDate dates' date.Now() StartDate ? From EndDate Date (to Date 'Creating List of Best Selling Products List Best Products ? new list (of the best products)() DM orderDao ? OrderDao() dim resultTable .todate) for each row as DataRow in the resultTable.RowproductProductProductproductproduct() with . ProductId - Convert. ToInt32 (row(0)), . Product Name - Convert.ToString(Row(1)), . Price - Convert. ToDouble (row(2)), . Quantity - Convert. ToInt32 (row(3)), . Total - Convert.ToDouble(Row(4)) - ListBestProducts.Add(bestProductModel) Grand Total +- Convert.ToDouble(Row(4)) Next end Sub End Class Presentation Layer After the domain layer encoding is complete, we will now add the data sets for the report, create the report file, and add a form to initialize the report viewer and create the buttons with a specific or custom date range. Report Builder You can use any report builder, whether Crystal Reports, Microsoft RDLC Report Designer, One Component, DevExpress, Jasper Reports, or anyone else, as we simply use it as the report designer and viewer, and the domain layer already has all the data ready to be viewed. In this guide, we use Local Processing Mode Reporting Services (RDLC) and view the report by using the Report Viewer control. Starting with Visual Studio 2017, this control and on-site reporting services are no longer integrated into the default Installation of Visual Studio, and you must install it from nuget or Visual Studio extensions. You can view this installation tutorial. Sales Reports File (SalesReport.rdlc) We add the local RDLC File/Report Designer blank if you want to use the Report Wizard. We build the presentation layer to load the references. We adjust the page size of the report (in this case A4->21 cm x 29.7 cm) and the report body (21 cm x 15 cm, body height should be adjusted for report elements). We add the SalesReport and BestProducts (.vb) .vb objects to the data value of the local report. Finally we designed the report as follows and thus, we added the fields to of our reporting objects). Design a form After you finish designing the local report, we added buttons with dates specified for the report and two datetimePicker controls for the custom date report. We add the Report Viewer control and dock to the entire container. In Report Viewer, we select the report file to display, in this case, the best-selling products report (BestProductsReport.rdlc)After you select the report, two data link sources for 2 objects in the report dataset (SalesReportDt and BestProductsDT) will be created at the bottom. Form code We import the domain or business layer. We created the LoadBestProductsReport method, with 2 parameters for the report's start and end date. Using the method, we prevent the sales report class of the domain layer and run the get report from the best-selling product method. Format the Report Viewer data source (SalesReportBindingSource, BestProductsBindingSource) with its report object and property (salesOrderModel, salesOrderModel.ListBestProducts). Finally, we run the above method from the buttons with the appropriate date range. Import Domain Public Class Form1 Sub-Private Form1_Load (Sender as Object, e as EventArgs) Handles MyBase.Load Edge Subload Product Report (startDate as Date and Time, End date and time) SalesReport() SalesReport() SalesReport.GetBestSellingProductsReport (startDate, endDate) SalesReportBindingSource.DataSource ? salesOrder The best products in the source player. DataSource ? salesSOrderModel.ListBastProducts me. ReportViewer1.RefreshReport() Sub-Private Finish btnToday_Click (Sends as Object, E as EventArgs) handles btnLium.Click Muffle from Page . , to schedule) end sub-private btnLast7Days_Click (sender as object, e as EventArgs) handles btnLast7Days.Click Dim from DateTime.Today.Today.AddDays(-7) DIM to Date ? DateTime.Now LoadBestProductsReport(From Date, todate) Has finished sub-private btnThisMonth_Click (sends as object, e as EventArgs) handles btn Month.Click D from Date (New Date and Time. DateTime.Now.Year,Datetime.Now.Month.Month,1) Dim todate ? DateTime.Now LoadBestProductsReport (adjusted, date) Sub-private termination btnLast30Days_Click (sends as object) e as EventArgs) handles btnL-30Days.Click D from DateTime.Today.AddDays(-30) D to Date ? DateTime.Now LoadBestProductsReport(from Date, todate) has finished sub-private btnThisYear_Click (sends as object, e as EventArgs) handles btnth this year.Click D from Date (New Date and Time . DateTime.Now.Year, 1, 1) Dim todate ? DateTime.Now LoadBestProductsReport(from Date, todate) Sub-season finale btnApply_Click (sender object, e As EventArgs) handles btnApply.Click DMD ? DateTimePickerFrom.Value Dim todate ? DateTimePickerToDate.Value LoadBestProductsReport(from date, date and time new (toDate.Year, toDate.Month, toDate.Day, 23, 59, 59)) Sub-end class finish and it's all ☺ Watch video guide downloads download database sales order + 7K records