Team Foundation Server: ggaete-ntbk y lspgc

Valor Factura, es el costo total del bien. (No se le resta el pie)

SELECT o.provision\_material FROM t\_operacion o

El quivalente seria

SELECT SUM(valor\*cantidad)

FROM t\_materiales M

WHERE operacion = @operacion

El pie esta en la operación y es único, independientemente de la cantidad de bienes. No todas las operaciones tiene pie.

Condiciones para determinar si el contrato NO es automático:

Cuando no se pueda determinar la renta el contrato.

Cuando tenga bienes en monedas distintas.

Tenga mas de un bien y con moneda diferente l peso

~~T\_avales\_fianzas~~ y/o t\_repre\_legales están los representantes del banco (institución que da el lease)

Mientras que en t\_repre\_legales\_general esta el resto del universo, contiene a las personas y el respectivo rut de la empresa.

Aval = Fiador

Cualquier garantía: entiéndase: Bien, Empresa, Persona se guarda en t\_garantias\_paso.

En t\_avales\_fianzas se guarda el detalle de los Avales. Si tipo Persona o Empresa el cod\_tipo\_garan=93.

t\_rel\_operacion\_repre\_y\_notario aca se guardan los representantes seleccionados para la operación. Tipo: 1 representante del banco, tipo 2: representante cliente solo en caso que sea una empresa) y tipo 3: notaria.

La dirección de todos se guarda en t\_direccion

Contrato lo genera pa\_cls\_clausulas\_contrato\_micro5

Crear un nuevo Reporte en Crystal Report para Chile

1. Crear el registro en la tablas leasedoc..p\_versiones\_documentos y leasedoc..p\_formato\_documentos
2. Crear la arquitectura en la app que invoca el SP que retorna datos.
3. La primera vez se guarda la salida del SP en un xml, el sp debe retornar al menos un registro.

T\_actividades\_personas almacena la persona encatgada de ejecutar cierta actividad dentro del WorkFlow

T\_modelos\_produccion Almacena el WorfFlow

Pa\_lc\_habilita\_usuario activa el usuario con clave encriptada.

**Leaseback**: ocurre cuando

La persona es dueña del bien (es como un empeño)

Cuando la institución que da el leasing es dueña del bien.

**Garantía**: Persona, Bien, Aval (Fiador)

T\_hoja\_ruta están los pasos de una operación a través del WorkFlow

<http://localhost:24023/Paginas/AdmSeguros/SeguroCliente/CartasSegClientes.aspx>

6867462k

De t\_cotizacion\_cliente se registran inicialmente y luego pasan a t\_clientes y t\_clientes\_empresas

public void Update(t\_materiales\_detalle updatedObject)

{

try

{

using (var context = new leacomEntities())

{

object originalObject = null;

var key = context.CreateEntityKey("t\_materiales\_detalle", updatedObject);

if (!context.TryGetObjectByKey(key, out originalObject)) return;

context.t\_materiales\_detalle.ApplyCurrentValues(updatedObject);

context.SaveChanges();

}

}

catch

{

throw;

}

}

Almacena el Representante Legal BCI y Santander

t\_cotizacion\_adicional

select \* from t\_cotizacion\_prov\_legal => select \* from t\_operacion\_prov\_legal

select \* from t\_cotizacion\_prov\_modif => No existe

select \* from t\_cotizacion\_prov\_otros => select \* from t\_operacion\_prov\_otros

No subir el archivo licence.licx pendiente al crear un nuevo repositorio

public void Insert(int correlativo, byte origen, byte codProducto, DateTime?,

decimal? tasa)

{

Database db = DatabaseFactory.CreateDatabase("LeaseOper");

DbCommand dbCommand =

db.GetSqlStringCommand(

"INSERT INTO t\_solicitud\_modif values(@correlativo,@origen,@cod\_producto,@fecha\_solicitud,@fecha\_siniestro,@fecha\_indemnizacion,@fecha\_bloqueo,@rut\_novacion,@riesgo\_vigente,@desc\_solicitud,@desc\_especial,@usuario,@monto\_reprog,@cod\_clasif\_super,@nueva\_fecha\_venc,@estado, @tasa)");

db.AddInParameter(dbCommand, "@correlativo", DbType.Int32, correlativo);

db.AddInParameter(dbCommand, "@origen", DbType.Byte, origen);

db.AddInParameter(dbCommand, "@cod\_producto", DbType.Byte, codProducto);

db.AddInParameter(dbCommand, "@fecha\_solicitud", DbType.DateTime,

fechaSolicitud < new DateTime(1900, 1, 1) ? null : fechaSolicitud);

db.AddInParameter(dbCommand, "@fecha\_siniestro", DbType.DateTime,

fechaSiniestro < new DateTime(1900, 1, 1) ? null : fechaSiniestro);

db.AddInParameter(dbCommand, "@fecha\_indemnizacion", DbType.DateTime,

fechaIndemnizacion < new DateTime(1900, 1, 1) ? null : fechaIndemnizacion);

db.AddInParameter(dbCommand, "@fecha\_bloqueo", DbType.DateTime,

fechaBloqueo < new DateTime(1900, 1, 1) ? null : fechaBloqueo);

db.AddInParameter(dbCommand, "@rut\_novacion", DbType.Int32, rutNovacion);

db.AddInParameter(dbCommand, "@riesgo\_vigente", DbType.Double, riesgoVigente);

db.AddInParameter(dbCommand, "@desc\_solicitud", DbType.String, descSolicitud);

db.AddInParameter(dbCommand, "@desc\_especial", DbType.String, descEspecial);

db.AddInParameter(dbCommand, "@usuario", DbType.Int16, usuario);

db.AddInParameter(dbCommand, "@monto\_reprog", DbType.Double, monto);

db.AddInParameter(dbCommand, "@cod\_clasif\_super", DbType.Byte, codClasfSuper);

db.AddInParameter(dbCommand, "@nueva\_fecha\_venc", DbType.DateTime,

nuevaFechaVenc < new DateTime(1900, 1, 1) ? null : nuevaFechaVenc);

db.AddInParameter(dbCommand, "@estado", DbType.Byte, estado);

db.AddInParameter(dbCommand, "@tasa", DbType.Double, tasa);

var i = db.ExecuteNonQuery(dbCommand);

db = null;

}

cbxproducto.SelectedItem = cbxproducto.Items.FindByValue(cod\_producto.ToString());

var container =

chckexiste.NamingContainer as GridViewDataItemTemplateContainer;

En esta tabla: p\_bienes\_financiar se guardan solo vehículos Franck.

//AC. Comentado por requerimiento, ocultar "Se financia". 13-05-2015

Administrar los reportes vía web:

<http://localhost:7226/PresentationLayer/Edition/DocManEditor/MainPage.aspx>

Com Progreso

<http://192.168.1.11:123/>

Comercial Chile

<http://192.168.1.37:>

Comercial de Santander:

<http://192.168.1.35:6301>

Las plantillas se guardan en esta tabla:

SELECT \* FROM leasedoc.dbo.p\_versiones\_documentos

<dx:GridViewDataTextColumn Caption="Ruc visitado" FieldName="rut\_visita" VisibleIndex="3" HeaderStyle-HorizontalAlign="Center" >

<PropertiesTextEdit **DisplayFormatString**="N0" MaxLength="14">

<ClientSideEvents KeyDown="SoloNumeros" GotFocus="NoCopyPaste" LostFocus="SiCopyPaste" />

</PropertiesTextEdit>

</dx:GridViewDataTextColumn>

<dx:ASPxButton ID="btnNuevo" runat="server" AutoPostBack="False" CssClass="boton-icono"

Cursor="pointer" EnableDefaultAppearance="False" EnableTheming="False" Text="Nuevo2" Width="80px" Height="8px">

<ClientSideEvents Click="alert('jo');" />

<Image Url="~/img/icon/arrow\_up.png">

</Image>

</dx:ASPxButton>

<PropertiesTextEdit DisplayFormatInEditMode="true" DisplayFormatString="{0:0.00}" MaskSettings-Mask="<0..99999999999g>.<0..99>"

MaskSettings-IncludeLiterals="DecimalSymbol">

</PropertiesTextEdit>

<EditItemTemplate>

<dx:ASPxTextBox ID="txtMonto" runat="server" MaxLength="20" DisplayFormatInEditMode="true"

ClientInstanceName="txtMonto" Text='<%# Eval("valor\_unitario") %>' DisplayFormatString="{0:0.00}" MaskSettings-Mask="<0..99999999999g>.<0..99>">

<ClientSideEvents KeyDown="SoloNumeros" GotFocus="NoCopyPaste" LostFocus="SiCopyPaste" />

<ValidationSettings Display="Dynamic">

</ValidationSettings>

</dx:ASPxTextBox>

</EditItemTemplate>

Instancias de Base de Datos:

BD-DESARROLLO\LEASINGTOTAL Lease2013

BD-DESARROLLO\PROGRESO

BCO-CHILE2012\CHILEDESA Lease2014

BCO-CHILE2012\BANAGRO Lease2015

BD-DESARROLLO\TANNER Lease2013

BD-DESARROLLO\PROGRESO Lease2014

bd-preproduccion\bci Lease2012

SRV-2012\santander\_desa Lease2015

**Tanner**

<http://192.168.1.11:8700/PresentationLayer/Edition/DocManEditor/MainPage.aspx>

SELECT @cod\_tipo\_pago\_arriendo = tipo\_pago\_arriendo FROM leasecom.dbo.t\_operacion WHERE operacion = @operacion

Si (@cod\_tipo\_pago\_arriendo = tipo\_pago\_arriendo) es 1 tiene pie (inicial), y se le debe restar.

Database db = DatabaseFactory.CreateDatabase("LeaseCom");

try

{

var dt = new DataTable();

DbCommand com = db.GetSqlStringCommand("select tipo\_cliente as contar from t\_clientes where rut\_cliente = " + rutAval);

dt.Load(db.ExecuteReader(com));

db = null;

return Convert.ToInt32(dt.Rows[0][0]);

}

catch (Exception ex)

{

db = null;

return 0;

}

// Ejecutar procedimiento almacenado que crea el nuevo documento de cotizacion

string connectionString = ConfigurationManager.ConnectionStrings["LeaseCom"].ConnectionString;

var connection = new SqlConnection(connectionString);

connection.Open();

var command = new SqlCommand("pa\_cls\_crea\_documento\_simulacion", connection);

sqlCommand.SqlCommandType = SqlCommandType.StoredProcedure;

sqlCommand.Parameters.AddWithValue("@numero", codCotizacion);

sqlCommand.Parameters.AddWithValue("@alternativa", alternativa);

sqlCommand.Parameters.AddWithValue("@tipo", tipoDocumento);

sqlCommand.Parameters.AddWithValue("@retorno", 0);

sqlCommand.Parameters["@retorno"].Direction = ParameterDirection.Output;

sqlCommand.ExecuteNonQuery();

string nuevoCorrelativoDocumento = sqlCommand.Parameters["@retorno"].Value.ToString();

sqlCommand.Dispose();

connection.Close();

connection.Dispose();

//grvDocumentos.DataBind();

//this.Page = new PerfilesWebBc().OverridePage(this.Page, "btnAgregarNewDocument;ASPxButton");

return nuevoCorrelativoDocumento;

DECLARE @resultado INT

EXEC sp\_repre\_legal\_get 1, @resultado OUTPUT

PRINT @resultado

string sConexion = ConfigurationManager.ConnectionStrings["LeaseCom"].ToString();

SqlConnection sqlConnection = new SqlConnection(sConexion);

SqlCommand sqlCommand = new SqlCommand("pa\_lo\_riesgo\_vigente\_clie", sqlConnection);

sqlCommand.CommandText = nombreSP;

sqlCommand.CommandType = CommandType.StoredProcedure;

sqlCommand.Parameters.Add(

new SqlParameter("@rut\_cliente", 7096802));

sqlConnection.Open();

using (var da = new SqlDataAdapter(sqlCommand))

{

da.Fill(dt);

}

sqlCommand.Dispose();

sqlConnection.Close();

sqlConnection.Dispose();

<dx:ASPxGridView ID="grvRepre" runat="server" AutoGenerateColumns="False" DataSourceID="odsGetRepreLegal"

KeyFieldName="codCotizacion" EnableCallBacks="true" ClientInstanceName="grvRepre">

<Columns>

<dx:GridViewDataTextColumn FieldName="repre\_legal" VisibleIndex="0" ReadOnly="true">

<EditFormSettings Visible="true" />

</dx:GridViewDataTextColumn>

</Columns>

<SettingsPager Visible="False">

</SettingsPager>

</dx:ASPxGridView>

<asp:ObjectDataSource ID="odsGetRepreLegal" runat="server" TypeName="LeaseComWeb.Negocio.Captacion.ImprimirWebBc"

SelectMethod="ConsultarRepresentanteLegal1" OnSelecting="odsGetRepreLegal\_Selecting">

<SelectParameters>

<asp:Parameter Name="codCotizacion" Type="Int32" />

</SelectParameters>

</asp:ObjectDataSource>

SqlConnection sqlConnection = new SqlConnection(sConexion);

SqlCommand sqlCommand = new SqlCommand("pa\_lo\_riesgo\_vigente\_clie", sqlConnection);

sqlCommand.CommandText = nombreSP;

sqlCommand.CommandType = CommandType.StoredProcedure;

sqlCommand.Parameters.Add(

new SqlParameter("@rut\_cliente", 7096802));

sqlConnection.Open();

dt.Load(sqlCommand.ExecuteReader());

sqlCommand.Dispose();

sqlConnection.Close();

sqlConnection.Dispose();

string sConexion = ConfigurationManager.ConnectionStrings["LeaseCom"].ToString();

SqlConnection sqlConnection = new SqlConnection(sConexion);

SqlCommand sqlCommand = new SqlCommand("pa\_lo\_riesgo\_vigente\_clie", sqlConnection);

sqlCommand.CommandText = nombreSP;

sqlCommand.CommandType = CommandType.StoredProcedure;

sqlCommand.Parameters.Add(

new SqlParameter("@rut\_cliente", 7096802));

sqlConnection.Open();

using (var da = new SqlDataAdapter(sqlCommand))

{

da.Fill(dt);

}

sqlCommand.Dispose();

sqlConnection.Close();

sqlConnection.Dispose();

cbxCliente.SelectedItem = cbxCliente.Items.FindByValue(Convert.ToInt32(Session["cotizacionComb"].ToString()));

private DataTable ConvertLinqToDatatable<T>(IEnumerable<T> varlist)

{

var tbl = new DataTable();

PropertyInfo[] properties = null;

if (varlist == null) return tbl;

foreach (T rec in varlist)

{

if (properties == null)

{

properties = ((Type)rec.GetType()).GetProperties();

foreach (PropertyInfo property in properties)

{

Type colType = property.PropertyType;

if ((colType.IsGenericType) && (colType.GetGenericTypeDefinition() == typeof(Nullable<>)))

colType = colType.GetGenericArguments()[0];

tbl.Columns.Add(new DataColumn(property.Name, colType));

}

}

DataRow drow = tbl.NewRow();

foreach (PropertyInfo property in properties)

drow[property.Name] = property.GetValue(rec, null) ?? DBNull.Value;

tbl.Rows.Add(drow);

}

return tbl;

}

SELECT

@bienes = @bienes + lugar\_entrega + ','

FROM t\_materiales M INNER JOIN

t\_proveedores P ON P.rut = M.rut\_proveedor

WHERE operacion = 100276

Da como resultado la unión de las n filas en una sola fila:

Av bolivar,AV ANDRES bello,Alameda con Moneda,

Verificacion de tabla temporales (no sirve sys. Sysobjects)

IF OBJECT\_ID('tempdb..#t\_materiales\_temp') IS NOT NULL DROP TABLE #t\_materiales\_temp