

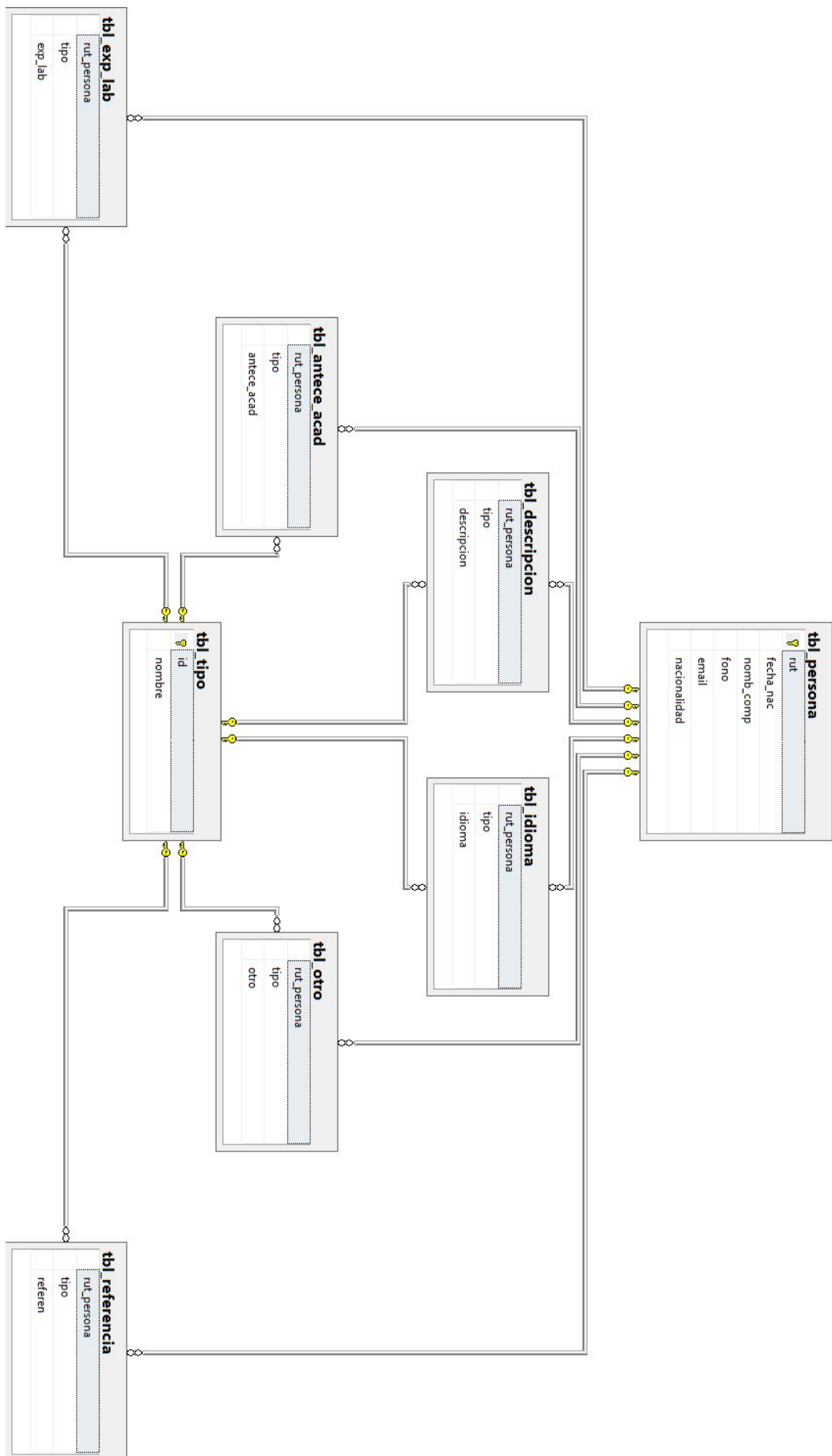


PROCEDIMIENTOS ALMACENADOS Y UTILIZACIÓN DE LENGUAJE
XML
EN BASE DE DATOS

Autor : Sony Oyarzun
Asignatura : Taller De Desarrollo De Bases De Datos

ÍNDICE

TABLAS	3
Tabla Persona	3
Tabla Tipo	3
Tabla Descripción	4
Tabla Antecedentes Académicos (antece_acad)	4
Tabla Experiencia Laboral (exp_lab)	5
Tabla Referencia	5
Tabla Idioma	6
Tabla Otro	6
ESTRUCTURA XML	7
Teléfonos	7
Descripción	7
Antecedentes Académicos	7
Experiencia Laboral	7
Referencias	8
Idiomas	8
Otros	8
INSERCIONES	9
Insertar Persona	9
Insertar Descripcion	10
Insertar Antecedentes Académicos	11
Insertar Experiencia Laboral	12
Insertar Referencias	13
Insertar Idioma	14
Insertar otros	15
Insertar Curriculum	16
MOSTRAR CURRÍCULUM	18
BIBLIOGRAFÍA	25



TABLAS

Tabla Persona



	Column Name	Data Type	Allow Nulls
	rut	varchar(12)	<input type="checkbox"/>
	fecha_nac	date	<input type="checkbox"/>
	nomb_comp	varchar(50)	<input type="checkbox"/>
	fono	xml	<input type="checkbox"/>
	email	varchar(50)	<input type="checkbox"/>
	nacionalidad	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla designada para el almacenamiento de personas y sus datos personales.

Tabla Tipo

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	nombre	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

1	Primer Trabajo
2	Recien Egresado
3	Cargo Gerencial

Tabla que permite almacenar los tipos de curriculum que se clasifican en 3 distintos:

- 1. Primer Trabajo :**
Clasificación para currículums sin Experiencia laboral ni Referencias.
- 2. Recien Egresado:**
Clasificación para currículums que buscan un trabajo relacionado a su carrera.
- 3. Cargo Gerencial:**
Clasificación para currículums que buscan cargos de fuerte posición en una empresa.

Tabla Descripción

tbl_descripcion			
	Column Name	Data Type	Allow Nulls
▶	rut_persona	varchar(12)	<input type="checkbox"/>
	tipo	int	<input type="checkbox"/>
	descripcion	xml	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla que permite almacenar una breve descripción de la persona, esta puede variar dependiendo del tipo de currículum que realicemos y a que labor o empresa estará dirigido.

Tabla Antecedentes Académicos (antece_acad)

tbl_antece_acad			
	Column Name	Data Type	Allow Nulls
▶	rut_persona	varchar(12)	<input type="checkbox"/>
	tipo	int	<input type="checkbox"/>
	antece_acad	xml	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla que permite almacenar uno o varios antecedentes académicos de una persona y que pueden variar según tipo de currículum.

Tabla Experiencia Laboral (exp_lab)

tbl_exp_lab			
	Column Name	Data Type	Allow Nulls
▶	rut_persona	varchar(12)	<input type="checkbox"/>
	tipo	int	<input type="checkbox"/>
	exp_lab	xml	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla que permite almacenar una o más empresas y datos correspondientes a esta, relacionado a su experiencia en ella.

Tabla Referencia

tbl_referencia			
	Column Name	Data Type	Allow Nulls
▶	rut_persona	varchar(12)	<input type="checkbox"/>
	tipo	int	<input type="checkbox"/>
	referen	xml	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla designada para almacenar las referencias que pueda tener una persona, con el fin de dar un aval a sus buenas labores dentro del ámbito laboral.

Tabla Idioma

	Column Name	Data Type	Allow Nulls
▶	rut_persona	varchar(12)	<input type="checkbox"/>
	tipo	int	<input type="checkbox"/>
	idioma	xml	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla que permite almacenar el o los idiomas de la persona, puede variar según tipo de currículum y empresa a la cual estará dirigido.

Tabla Otro

	Column Name	Data Type	Allow Nulls
▶	rut_persona	varchar(12)	<input type="checkbox"/>
	tipo	int	<input type="checkbox"/>
	otro	xml	<input type="checkbox"/>
			<input type="checkbox"/>

Tabla que permite almacenar datos variados, Hobbies u otro dato que se desea agregar según tipo de curriculum.

ESTRUCTURA XML

Teléfonos

```
<Fonos>
  <fono></fono>
</Fonos>
```

permite almacenar como valor de **contenido central** del **tag fono** el numero de movil o telefono de casa de la persona, permitiendo almacenar múltiples números, uno por cada tag.

Descripción

```
<Descripciones>
  <descripcion></descripcion>
</Descripciones>
```

Permite almacenar como valor de **contenido central** dentro del **tag descripción** una breve información que describa a la persona.

Antecedentes Académicos

```
<Antedecentes>
  <casa_estudio nombre="" carrera="" annioI="" annioT="" descripcion=""></casa_estudio>
</Antedecentes>
```

Permite almacenar como **atributos** dentro del **tag casa_estudio** el nombre de la institución (**nombre**), la carrera estudiada (**carrera**), año de ingreso (**annioI**) y término de la carrera (**annioT**), además de información adicional de la carrera o casa de estudio (**descripcion**).

Experiencia Laboral

```
<Experiencia_laboral>
  <empresa nombre="" descripcion="" annioI="" annioT="" cargo="" plabores=""></empresa>
</Experiencia_laboral>
```

Permite almacenar como **atributos** dentro del **tag empresa** el nombre del lugar de trabajo o empresa (**nombre**), información adicional de la empresa (**descripcion**), año de ingreso (**annioI**) y término de labores (**annioT**), cargo ejercido (**cargo**) y labores dentro de este.

Referencias

```
<Referencias>
  <referencia nombre="" email="" cargo=""></referencia>
</Referencias>
```

Permite almacenar como **atributos** dentro del **tag referencia**, el nombre de una persona ([nombre](#)), mail ([email](#)), cargo ([cargo](#)) de la persona que avala las labores del propietario del curriculum.

Idiomas

```
<Idiomas>
<idioma nombre="" nivel="" tipo=""></idioma>
</Idiomas>
```

Permite almacenar el como **atributos** dentro del **tag idioma**, el nombre del idioma([nombre](#)), nivel del idioma ([nivel](#)), tipo ([tipo](#)) el cual pudiese ser hablado o escrito.

Otros

```
<Otros>
<hobbie nombre="" annio=""></hobbie>
<otro></otro>
</Otros>
```

Permite almacenar Hobbies como **atributos** dentro del **tag hobbie**, ya sea el nombre de este([nombre](#)) y el año en que se realiza o realizó([annio](#)), además de ser posible agregar datos extras no clasificados estrictamente, para ello se almacenan como valor **del contenido central** dentro del **tag otro**, estos pudiesen ser: licencia de conducir, disponibilidad etc.

INSERCIONES

Insertar Persona

CREATE PROCEDURE INSERTAR_PERSONA

@rut varchar(12),@nomb varchar(50),@email varchar(50),@fecha date,@nacionalidad varchar(50),@rutaFono
Varchar(100)

AS

IF NOT EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN

DECLARE @xml xml
DECLARE @query nvarchar(max)

SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaFono + "", Single_Blob)
AS x)'

EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output
INSERT INTO tbl_persona (rut,nomb_comp,fecha_nac,email,nacionalidad,fono)
VALUES(@rut,@nomb,@fecha,@email,@nacionalidad,@xml)

END
ELSE
BEGIN
PRINT 'La persona ya esta registrada'
END

Este Procedimiento está encargado de insertar datos dentro de la tabla persona, para ello requiere como parámetros de entrada el rut, nombre completo, fecha de nacimiento en formato 'aaaa/mm/dd', nacionalidad y ruta donde se encuentra el documento xml que contiene el o los teléfonos del propietario del curriculum.

Ejemplo de Inserción:

EXECUTE INSERTAR_PERSONA

'1-1',
'Pablo Escobar',
'pablo@escobar',
'1940-07-30',
'Chileno',
D:\XML SOL BD\fono 1-1.xml'

Insertar Descripción

```
CREATE PROCEDURE INSERTAR_DESCRIPCION @rut varchar(12),@tipo int,@rutaDescripcion varchar(100)
AS
IF EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN
    IF NOT EXISTS(SELECT * FROM tbl_descripcion WHERE rut_persona=@rut and tipo=@tipo)
    BEGIN
        DECLARE @xml xml
        DECLARE @query nvarchar(max)

        SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaDescripcion +
        "", Single_Blob) AS x)'
        EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output

        INSERT INTO tbl_descripcion (rut_persona,tipo,descripcion) VALUES (@rut,@tipo,@xml)

    END
    ELSE
    BEGIN
        PRINT 'Descripcion ya esta registrada'
    END
END
ELSE
BEGIN
    PRINT 'Rut no registrado'
END
```

Procedimiento encargado de insertar una descripcion en la tabla descripcion, correspondiente al propietario del currículum, para ello requiere como parámetros de entrada el rut, tipo de currículum y ruta donde se encuentra el documento xml que contiene la descripcion.

Ejemplo de Inserción:

```
EXECUTE INSERTAR_DESCRIPCION
'1-1',
1,
'D:\XML SOL BD\descripcion 1-1 tipo 1.xml'
```

Insertar Antecedentes Académicos

```
CREATE PROCEDURE INSERTAR_ANTECEDENTES_ACAD @rut varchar(12),@tipo int,@rutaAnteAcad
varchar(100)
AS
IF EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN
    IF NOT EXISTS(SELECT * FROM tbl_antece_acad WHERE rut_persona=@rut and tipo=@tipo)
    BEGIN
        DECLARE @xml xml
        DECLARE @query nvarchar(max)

        SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaAnteAcad + "",
Single_Blob) AS x)'
        EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output

        INSERT INTO tbl_antece_acad (rut_persona,tipo,antece_acad) VALUES (@rut,@tipo,@xml)

    END
    ELSE
    BEGIN
        PRINT 'Antecedente ya está registrado'
    END
END
ELSE
BEGIN
    PRINT 'Rut no registrado'
END
```

Procedimiento encargado de insertar uno o más antecedentes académicos en la tabla antece_acad correspondiente al propietario del currículum, para ello requiere como parámetros de entrada el rut, tipo de currículum y ruta donde se encuentra el documento xml que contiene el los antecedentes.

Ejemplo de Inserción:

```
EXECUTE INSERTAR_ANTECEDENTES_ACAD
'1-1',
1,
'D:\XML SOL BD\ante_acad 1-1 tipo 1.xml'
```

Insertar Experiencia Laboral

```
CREATE PROCEDURE INSERTAR_EXP_LABORAL @rut varchar(12),@tipo int,@rutaExp varchar(100)
AS
IF EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN
    IF(@tipo!=1)
    BEGIN

        IF NOT EXISTS(SELECT * FROM tbl_exp_lab WHERE rut_persona=@rut and tipo=@tipo)
        BEGIN
            DECLARE @xml xml
            DECLARE @query nvarchar(max)

            SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaExp + "",
Single_Blob) AS x)'
            EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output

            INSERT INTO tbl_exp_lab (rut_persona,tipo,exp_lab) VALUES (@rut,@tipo,@xml)

        END
    ELSE
    BEGIN
        PRINT 'Experiencia laboral ya está registrada'
    END
    ELSE
    BEGIN
        PRINT 'CURRICULUM TIPO 1 NO REQUIERE EXPERIENCIA LABORAL'
    END
END
ELSE
BEGIN
    PRINT 'Rut no registrado'
END
```

Procedimiento encargado de insertar una o más experiencias laborales en la tabla exp_lab correspondiente al propietario del currículum, para ello requiere como parámetros de entrada el rut, tipo de currículum y ruta donde se encuentra el documento xml que contiene la experiencia.

Ejemplo de Inserción:

```
EXECUTE INSERTAR_EXP_LABORAL
'1-1',
2,
'D:\XML SOL BD\exp_lab 1-1 tipo 2.xml'
```

Insertar Referencias

```
CREATE PROCEDURE INSERTAR_REFERENCIA @rut varchar(12),@tipo int,@rutaReferencia varchar(100)
AS
IF EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN
    IF(@tipo!=1)
    BEGIN
        IF NOT EXISTS(SELECT * FROM tbl_referencia WHERE rut_persona=@rut and tipo=@tipo)
        BEGIN
            DECLARE @xml xml
            DECLARE @query nvarchar(max)

            SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaReferencia + "",
Single_Blob) AS x)'
            EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output

            INSERT INTO tbl_referencia (rut_persona,tipo,referen) VALUES (@rut,@tipo,@xml)

        END
    ELSE
        BEGIN
            PRINT 'Referencia ya esta registrada'
        END
    END
ELSE
    BEGIN
        PRINT 'CURRICULUM TIPO 1 NO REQUIERE REFERENCIAS'
    END
END
ELSE
BEGIN
    PRINT 'Rut no registrado'
END
```

Procedimiento encargado de insertar referencias que avalan al propietario del currículum en la tabla referencia, para ello requiere como parámetros de entrada el rut de la persona, el tipo de currículum al cual pertenece y la ruta en la cual se encuentra el documento xml que contiene las referencias.

Ejemplo de Inserción:

```
EXECUTE INSERTAR_REFERENCIA
'1-1',
2,
'D:\XML SOL BD\referencia 1-1 tipo 2'
```

Insertar Idioma

```
CREATE PROCEDURE INSERTAR_IDIOMA @rut varchar(12),@tipo int,@rutaldioma varchar(100)
AS
IF EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN
    IF NOT EXISTS(SELECT * FROM tbl_idioma WHERE rut_persona=@rut and tipo=@tipo)
    BEGIN
        DECLARE @xml xml
        DECLARE @query nvarchar(max)

        SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaldioma + "",
Single_Blob) AS x)'
        EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output

        INSERT INTO tbl_idioma (rut_persona,tipo,idioma) VALUES (@rut,@tipo,@xml)

    END
    ELSE
    BEGIN
        PRINT 'Idioma ya está registrado'
    END
END
ELSE
BEGIN
    PRINT 'Rut no registrado'
END
```

Procedimiento encargado de insertar uno o más idiomas en la tabla idioma correspondiente al propietario del currículum, para ello requiere como parámetros de entrada el rut, tipo de currículum y ruta donde se encuentra el documento xml que contiene el idioma.

Ejemplo de Inserción:

```
EXECUTE INSERTAR_IDIOMA
'1-1',
1,
'D:\XML SOL BD\referencia 1-1 tipo 1.xml'
```

Insertar otros

```
CREATE PROCEDURE INSERTAR_OTROS @rut varchar(12),@tipo int,@rutaOtro varchar(100)
AS
IF EXISTS(SELECT * FROM tbl_persona WHERE rut=@rut)
BEGIN
    IF NOT EXISTS(SELECT * FROM tbl_otro WHERE rut_persona=@rut and tipo=@tipo)
    BEGIN
        DECLARE @xml xml
        DECLARE @query nvarchar(max)

        SELECT @query ='(SELECT @varOut = BulkColumn from Openrowset(Bulk "" + @rutaOtro + "",
Single_Blob) AS x)'
        EXECUTE sp_executesql @Query, N'@varOut xml out',@varOut=@xml output

        INSERT INTO tbl_otro (rut_persona,tipo,otro) VALUES (@rut,@tipo,@xml)

    END
    ELSE
    BEGIN
        PRINT 'informacion adicional ya esta registrada'
    END
END
ELSE
BEGIN
    PRINT 'Rut no registrado'
END
```

Procedimiento encargado de insertar uno o más datos extras no clasificados estrictamente en la tabla otro correspondiente al propietario del currículum, para ello requiere como parámetros de entrada el rut, tipo de currículum y ruta donde se encuentra el documento xml que contiene el los datos extras.

Ejemplo de Inserción:

```
EXECUTE INSERTAR_OTROS
'1-1'
,1,
'D:\XML SOL BD\otro 1-1 tipo 1.xml'
```


Insertar Curriculum

```
CREATE PROCEDURE INSERTAR_CURRICULUM
  @Rut                varchar(12),
  @DescripcionRuta    varchar(1000),
  @AntecedentesRuta   varchar(1000),
  @ExperienciaRuta     varchar(1000),
  @ReferenciaRuta     varchar(1000),
  @Idiomaruta         varchar(1000),
  @OtroRuta           varchar(1000),
  @Tipo int
AS
EXECUTE INSERTAR_DESCRIPCION @Rut,@Tipo,@DescripcionRuta
EXECUTE INSERTAR_ANTECEDENTES_ACAD @Rut,@Tipo,@AntecedentesRuta
IF(@Tipo=1)
  BEGIN
    PRINT 'Curriculum Tipo 1 No requiere Experiencia Laboral ni Referencias'
  END
ELSE
  BEGIN
    EXECUTE INSERTAR_EXP_LABORAL @Rut,@Tipo,@ExperienciaRuta
    EXECUTE INSERTAR_REFERENCIA @Rut,@Tipo,@ReferenciaRuta
  END
EXECUTE INSERTAR_IDIOMA @Rut,@Tipo,@Idiomaruta
EXECUTE INSERTAR_OTROS @Rut,@Tipo,@OtroRuta
```

Procedimiento "Principal" que contiene a los demás procedimientos para insertar (excepto insertar persona) requiere como parámetros de entrada el rut de la persona, rutas de la descripcion, antecedentes, experiencia, referencias, idiomas, y otros, además del número del tipo de currículum que estamos insertando. cabe reclamar que los curriculum de tipo 1 son para la primera experiencia laboral, por lo cual no requieren que se ingrese experiencias laborales ni referencias, si no se ingresan se deben dejar NULL los parámetros de entrada.

Ejemplos en la siguiente Página ----->

Ejemplos de Inserción:

EXECUTE INSERTAR_CURRICULUM

```
/*rut*/          '1-1',
/*descripcion*/  'D:\XML SOL BD\descripcion 1-1 tipo 1.xml',
/*antecedentes*/ 'D:\XML SOL BD\ante_acad 1-1 tipo 1.xml',
/*experiencia*/  null,
/*referencias*/  null,
/*idioma*/       'D:\XML SOL BD\idioma 1-1 tipo 1.xml',
/*otro*/         'D:\XML SOL BD\otro 1-1 tipo 1.xml',
/*tipo*/         1
```

EXECUTE INSERTAR_CURRICULUM

```
/*rut*/          '1-1',
/*descripcion*/  'D:\XML SOL BD\descripcion 1-1 tipo 2.xml',
/*antecedentes*/ 'D:\XML SOL BD\ante_acad 1-1 tipo 2.xml',
/*experiencia*/  'D:\XML SOL BD\exp_lab 1-1 tipo 2.xml',
/*referencias*/  'D:\XML SOL BD\referencia 1-1 tipo 2.xml',
/*idioma*/       'D:\XML SOL BD\idioma 1-1 tipo 2.xml',
/*otro*/         'D:\XML SOL BD\otro 1-1 tipo 2.xml',
/*tipo*/         2
```

EXECUTE INSERTAR_CURRICULUM

```
/*rut*/          '1-1',
/*descripcion*/  'D:\XML SOL BD\descripcion 1-1 tipo 3.xml',
/*antecedentes*/ 'D:\XML SOL BD\ante_acad 1-1 tipo 3.xml',
/*experiencia*/  'D:\XML SOL BD\exp_lab 1-1 tipo 3.xml',
/*referencias*/  'D:\XML SOL BD\referencia 1-1 tipo 3.xml',
/*idioma*/       'D:\XML SOL BD\idioma 1-1 tipo 3.xml',
/*otro*/         'D:\XML SOL BD\otro 1-1 tipo 3.xml',
/*tipo*/         3
```

MOSTRAR CURRÍCULUM

```
CREATE PROCEDURE MOSTRAR_CURRICULUM @rut varchar(12),@tipo int
```

```
AS
```

```
IF EXISTS(SELECT nomb_comp FROM tbl_persona WHERE rut=@rut)
```

```
BEGIN
```

```
PRINT 'CURRICULUM VITAE'
```

```
DECLARE @xml xml
```

```
--PERSONA
```

```
DECLARE C1 CURSOR FOR
```

```
SELECT nomb_comp,fecha_nac,email,nacionalidad FROM tbl_persona WHERE rut = @rut
```

```
OPEN C1
```

```
DECLARE @nombre          varchar(max)
```

```
DECLARE @fecha           varchar(max)
```

```
DECLARE @mail            varchar(max)
```

```
DECLARE @nacionalidad varchar(max)
```

```
FETCH C1 INTO @nombre,@fecha,@mail,@nacionalidad
```

```
While(@@FETCH_STATUS=0)
```

```
BEGIN
```

```
PRINT 'NOMBRE'           :'+@nombre
```

```
PRINT 'FECHA'            :'+@fecha
```

```
PRINT 'MAIL'             :'+@mail
```

```
PRINT 'NACIONALIDAD'    :'+@nacionalidad
```

```
FETCH C1 INTO @nombre,@fecha,@mail,@nacionalidad
```

```
END
```

```
CLOSE C1
```

```
DEALLOCATE C1
```

Script continua en la siguiente página --->

--TELÉFONOS

```
SET @xml = (SELECT fono FROM tbl_persona WHERE rut=@rut)
DECLARE C1 CURSOR FOR
SELECT x.Rec.query('.').value('.', 'nvarchar(2000)') AS 'Telefonos'
FROM   @xml.nodes('/Fonos/fono') as x(Rec)
OPEN C1
DECLARE @fono          varchar(50)
DECLARE @cont          int
SET @cont=1;
FETCH C1 INTO @fono
PRINT 'TELEFONOS'
While(@@FETCH_STATUS=0)
BEGIN
PRINT 'Fono '+CAST(@cont as varchar(50))+          ':'+CAST(@fono as varchar(50))
FETCH C1 INTO @fono
SET @cont=@cont+1;
END
CLOSE C1
DEALLOCATE C1
```

--DESCRIPCION

```
SET @xml = (SELECT descripcion FROM tbl_descripcion WHERE rut_persona=@rut AND tipo=@tipo)
DECLARE C1 CURSOR FOR
SELECT x.Rec.query('.').value('.', 'nvarchar(2000)') AS 'Descripcion'
FROM   @xml.nodes('/Descripciones/descripcion') as x(Rec)
OPEN C1
DECLARE @descripcion          varchar(max)
FETCH C1 INTO @descripcion
While(@@FETCH_STATUS=0)
BEGIN
PRINT 'DESCRIPCION          :'
PRINT @descripcion
FETCH C1 INTO @descripcion
END
CLOSE C1
DEALLOCATE C1
```

Script continua en la siguiente página --->

```
--ANTECEDENTES ACADEMICOS
SET @xml = (SELECT antece_acad FROM tbl_antece_acad WHERE rut_persona=@rut AND tipo=@tipo)
DECLARE C1 CURSOR FOR
SELECT
x.Rec.query('data(/@nombre[1])').value('.', 'nvarchar(2000)') AS 'Nombre',
x.Rec.query('data(/@carrera[1])').value('.', 'nvarchar(2000)') AS 'Descripcion',
x.Rec.query('data(/@anniol[1])').value('.', 'nvarchar(2000)') AS 'Año Inicio',
x.Rec.query('data(/@annioT[1])').value('.', 'nvarchar(2000)') AS 'Año Termino',
x.Rec.query('data(/@descripcion[1])').value('.', 'nvarchar(2000)') AS 'Cargo'
FROM @xml.nodes('/Antedecentes/casa_estudio') as x(Rec)
OPEN C1
DECLARE @nombreAnt          varchar(max)
DECLARE @carreraAnt         varchar(max)
DECLARE @anniol_Ant         varchar(max)
DECLARE @annioT_Ant         varchar(max)
DECLARE @descripcionAnt     varchar(max)

SET @cont=1;
FETCH C1 INTO @nombreAnt,@carreraAnt,@anniol_Ant,@annioT_Ant,@descripcionAnt
IF(@nombreAnt!="")
BEGIN
PRINT "
PRINT 'ANTECEDENTES ACADEMICOS    :'
While(@@FETCH_STATUS=0)
BEGIN
PRINT "
PRINT 'CASA DE ESTUDIO '+CAST(@cont as varchar(50))
PRINT 'Nombre           :'+@nombreAnt
PRINT 'Carrera          :'+@carreraAnt
PRINT 'Inicio            :'+@anniol_Ant
PRINT 'Termino           :'+@annioT_Ant
PRINT 'Descripcion        :'+@descripcionAnt

SET @cont=@cont+1;
FETCH C1 INTO @nombreAnt,@carreraAnt,@anniol_Ant,@annioT_Ant,@descripcionAnt
END
END
CLOSE C1
DEALLOCATE C1
```

Script continua en la siguiente página --->

--EXPERIENCIA LABORAL

```
SET @xml = (SELECT exp_lab FROM tbl_exp_lab WHERE rut_persona=@rut AND tipo=@tipo)
DECLARE C1 CURSOR FOR
SELECT
x.Rec.query('data(/@nombre[1])').value('.', 'nvarchar(2000)') AS 'Nombre',
x.Rec.query('data(/@descripcion[1])').value('.', 'nvarchar(2000)') AS 'Descripcion',
x.Rec.query('data(/@anniol[1])').value('.', 'nvarchar(2000)') AS 'Año Inicio',
x.Rec.query('data(/@annioT[1])').value('.', 'nvarchar(2000)') AS 'Año Termino',
x.Rec.query('data(/@cargo[1])').value('.', 'nvarchar(2000)') AS 'Cargo',
x.Rec.query('data(/@plabores[1])').value('.', 'nvarchar(2000)') AS 'Principales Labores'
FROM @xml.nodes('/Experiencia_laboral/empresa') as x(Rec)
OPEN C1
DECLARE @nombreEmp      varchar(max)
DECLARE @descripEmp     varchar(max)
DECLARE @anniol_Emp     varchar(max)
DECLARE @annioT_Emp     varchar(max)
DECLARE @cargoEmp       varchar(max)
DECLARE @laborEmp       varchar(max)
SET @cont=1;
FETCH C1 INTO @nombreEmp,@descripEmp,@anniol_Emp,@annioT_Emp,@cargoEmp,@laborEmp
IF(@nombreEmp!="")
BEGIN
PRINT "
PRINT 'EXPERIENCIA LABORAL  : '
While(@@FETCH_STATUS=0)
BEGIN
PRINT "
PRINT 'EMPRESA '+CAST(@cont as varchar(50))
PRINT 'Nombre      :'+@nombreEmp
PRINT 'Descripcion  :'+@descripEmp
PRINT 'Inicio      :'+@anniol_Emp
PRINT 'Termino     :'+@annioT_Emp
PRINT 'Cargo       :'+@cargoEmp
PRINT 'Labores     :'+@laborEmp
SET @cont=@cont+1;
FETCH C1 INTO @nombreEmp,@descripEmp,@anniol_Emp,@annioT_Emp,@cargoEmp,@laborEmp
END
END
CLOSE C1
DEALLOCATE C1
```

Script continua en la siguiente página --->

--REFERENCIAS

```
SET @xml = (SELECT referen FROM tbl_referencia WHERE rut_persona=@rut AND tipo=@tipo)
DECLARE C1 CURSOR FOR
SELECT
x.Rec.query('data(/@nombre[1])').value('.', 'nvarchar(2000)') AS 'Nombre',
x.Rec.query('data(/@email[1])').value('.', 'nvarchar(2000)') AS 'Mail',
x.Rec.query('data(/@cargo[1])').value('.', 'nvarchar(2000)') AS 'Cargo'
FROM @xml.nodes('/Referencias/referencia') as x(Rec)
OPEN C1
DECLARE @nombreReferencia          varchar(max)
DECLARE @mailReferencia            varchar(max)
DECLARE @cargoReferencia          varchar(max)
SET @cont=1;
FETCH C1 INTO @nombreReferencia,@mailReferencia,@cargoReferencia
IF(@nombreReferencia!="")
BEGIN
PRINT "
PRINT 'REFERENCIAS  : '
While(@@FETCH_STATUS=0)
BEGIN
PRINT "
PRINT 'REFERENCIA '+CAST(@cont as varchar(50))
PRINT 'Nombre      :'+@nombreReferencia
PRINT 'Mail        :'+@mailReferencia
PRINT 'Cargo       :'+@cargoReferencia
SET @cont=@cont+1;
FETCH C1 INTO @nombreReferencia,@mailReferencia,@cargoReferencia
END
END
CLOSE C1
DEALLOCATE C1
```

Script continua en la siguiente página --->

--IDIOMA

```
PRINT "  
PRINT 'IDIOMAS :'  
SET @xml = (SELECT idioma FROM tbl_idioma WHERE rut_persona=@rut AND tipo=@tipo)  
DECLARE C1 CURSOR FOR  
SELECT  
x.Rec.query('data(./@nombre[1])').value('.', 'nvarchar(2000)') AS 'Nombre',  
x.Rec.query('data(./@nivel[1])').value('.', 'nvarchar(2000)') AS 'Nivel',  
x.Rec.query('data(./@tipo[1])').value('.', 'nvarchar(2000)') AS 'Tipo'  
  
FROM @xml.nodes('/Idiomas/idioma') as x(Rec)  
  
OPEN C1  
DECLARE @nombredidioma          varchar(max)  
DECLARE @niveldidioma  varchar(max)  
DECLARE @tipoldioma          varchar(max)  
  
SET @cont=1;  
FETCH C1 INTO @nombredidioma,@niveldidioma,@tipoldioma  
While(@@FETCH_STATUS=0)  
BEGIN  
PRINT "  
PRINT 'IDIOMA '+CAST(@cont as varchar(50))  
PRINT 'Nombre      :'+@nombredidioma  
PRINT 'Nivel      :'+@niveldidioma  
PRINT 'Tipo      :'+@tipoldioma  
  
SET @cont=@cont+1;  
FETCH C1 INTO @nombredidioma,@niveldidioma,@tipoldioma  
END  
CLOSE C1  
DEALLOCATE C1
```

Script continua en la siguiente página --->

--OTROS

```
PRINT "  
PRINT 'OTROS  ':
```

```
SET @xml = (SELECT otro FROM tbl_otro WHERE rut_persona=@rut AND tipo=@tipo)
```

--CURSOR HOBBIE

```
DECLARE C1 CURSOR FOR  
SELECT  
x.Rec.value('./@nombre[1]', 'nvarchar(2000)') AS 'Nombre',  
x.Rec.value('./@annio[1]', 'nvarchar(2000)') AS 'Año'  
  
FROM  @xml.nodes('/Otros/hobbie') as x(Rec)  
  
OPEN C1  
DECLARE @nombreHobbie      varchar(max)  
DECLARE @annioHobbie  varchar(max)  
DECLARE @otro              varchar(max)  
  
SET @cont=1;  
FETCH C1 INTO @nombreHobbie,@annioHobbie  
    While(@@FETCH_STATUS=0)  
    BEGIN  
        PRINT "  
        IF(@nombreHobbie!=")  
        BEGIN  
            PRINT 'HOBBIE '+CAST(@cont as varchar(50))  
            PRINT 'Nombre      ':'+@nombreHobbie  
            PRINT 'AÑO        ':'+@annioHobbie  
        END  
        SET @cont=@cont+1;  
        FETCH C1 INTO @nombreHobbie,@annioHobbie  
    END  
CLOSE C1  
DEALLOCATE C1
```

Script continua en la siguiente página --->

--CURSOR OTRO

```
DECLARE C1 CURSOR FOR
SELECT
x.Rec.value('.', 'nvarchar(2000)') AS 'Otro'

FROM  @xml.nodes('/Otros/otro') as x(Rec)
OPEN C1
FETCH C1 INTO @otro
SET @cont=1;
While(@@FETCH_STATUS=0)
BEGIN
    PRINT "
    IF(@otro!=")
    BEGIN
        PRINT 'OTROS DATOS '+CAST(@cont as varchar(50))+ '      :'+@otro
    END
    SET @cont=@cont+1;
    FETCH C1 INTO @otro
END
CLOSE C1
DEALLOCATE C1
END
ELSE
BEGIN
PRINT 'RUT SIN CURRÍCULUM REGISTRADO'
END
```

Procedimiento que consiste en realizar consultas a los campos xml de cada tabla, almacenando el resultado en una variable xml , se crea un cursor para poder recorrer y se realiza una nueva consulta, esta vez en **XQUERY** a los campos xml extraídos. de esta manera se visualiza el curriculum, el cual puede ser en 3 tipos dependiendo del parámetro que ingresemos, los parámetros de entrada requeridos son rut de la persona y tipo de curriculum.

BIBLIOGRAFÍA

SITIOS WEB:

Recuperación de valores a través x.value()

<https://docs.microsoft.com/es-es/sql/t-sql/xml/value-method-xml-data-type?view=sql-server-2017>

Consultas x.query()

<https://docs.microsoft.com/en-us/sql/t-sql/xml/nodes-method-xml-data-type?view=sql-server-2017>

Introducción a XML

<https://es.scribd.com/document/25168109/SQL-con-XML>

Posibles soluciones a error en la extracción de atributos de un xml

<https://stackoverflow.com/questions/15846990/extracting-attributes-from-xml-fields-in-sql-server-2008-table>

IMÁGENES:

Logo Portada

<https://blogredalyc.files.wordpress.com/2016/07/unknown.jpg>