



EVALUACION PROCESUAL HITO 4

BASE DE DATOS I

PRESENTA:
RONALD URIEL CHOQUE PACO – SIS6972733

UNIFRANZ
Internacionalizate

INNOVACIÓN
EN EDUCACIÓN



EL ALTO, NOVIEMBRE DE 2021



01

DISEÑO DE BASE DE DATOS

Se presentará la elaboración de la base de datos ONG_H4, incluyendo el DIGRAMA LOGICO y el MODELO ENTIDAD RELACION E – R.

02

MANEJO DE VISTAS

Se presentará la elaboración de las diferentes vistas requeridas.

03

MANEJO DE FUNCIONES

Se presentara la elaboración de las diferentes funciones requeridas.

SECCIONES

BASE DE DATOS I



BASE DE DATOS I

SECCION: DISEÑO DE LA BASE DE DATOS



CONCEPTOS

RELEVANTES

dbo.

Es la especificacion inmediata del esquema a usarse en el Proyecto, siendo de eleccion el DBO (por defecto).

GO

Escribir GO luego de un proceso, simbolizada una indicacion para la ejecucion, pudiendo ser esta benefica para ejecutar todo el codigo (no parte por parte con se hacia de maneja convencional).

VISTAS

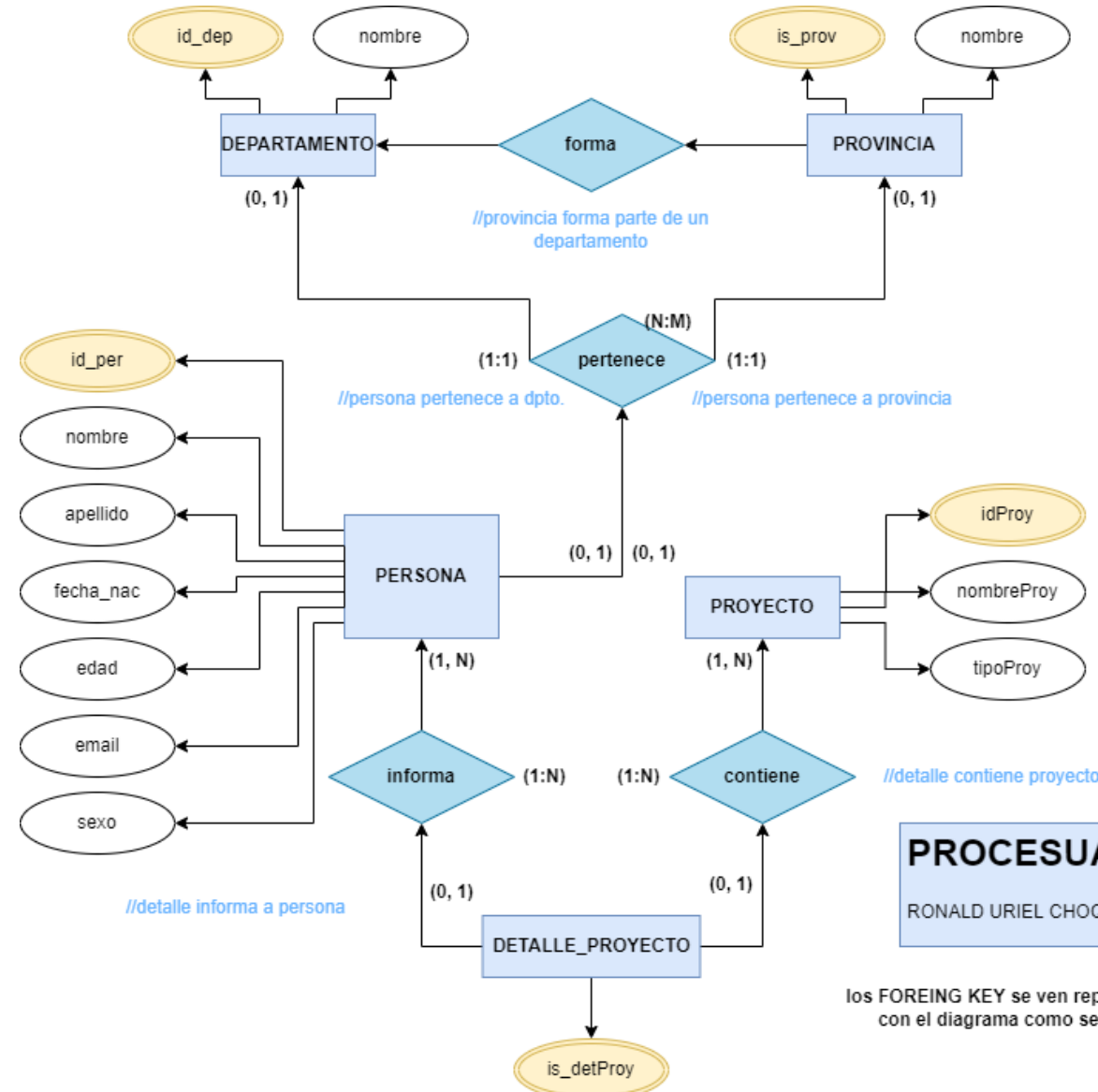
O tambien llamada tabla virtual, es un metodo de consulta comoda a partir deL SELECT.

FUNCIONES

Conjunto de instrucciones, para realizar una tarea especifica de manera automatica, existiendo las de AGREGACION y las CUSTOM.

01

MODELO ENTIDAD RELACION



PROCESUAL H4

RONALD URIEL CHOQUE PACO

los FOREIGN KEY se ven representados con el diagrama como se indico.

VISUALIZACION DETALLADA:

https://drive.google.com/drive/folders/1j1lp6-RuIXroSLcv4yrwya9_Uo1oK7SN?usp=sharing

02

A NOTAR: TIEMPO DE EJECUCION TOTAL
PARA LA CREACION Y DISEÑO DE LA
BASE DE DATOS

```
('Cercado', 2),  
( 'Ayopaya', 2),  
( 'Andres Ibanez', 3),  
( 'German Bush', 3),  
( 'Aniceto Arce', 4)  
[2021-11-19 01:36:50] 8 rows affected in 15 ms
```

Tiempo logrado gracias al uso del GO.

03

CREACION DE LA BASE DE DATOS

```
1 CREATE DATABASE ONG_H4;  
2 USE ONG_H4;
```

04


CREACION/LLENADO TABLA DEPARTAMENTO

```
4 CREATE TABLE dbo.departamento
5 (
6     id_dep INTEGER IDENTITY PRIMARY KEY NOT NULL,
7     nombre_dep VARCHAR(50) NOT NULL
8 );
9 GO
```


```
49 INSERT INTO departamento(nombre_dep) VALUES
50 ('La Paz'),
51 ('Cochabamba'),
52 ('Santa Cruz'),
53 ('Tarifa');
54 GO
```

NOTA: los **INSERT** fueron llenados de manera autónoma.



05

CREACION/LLENADO TABLA
PROVINCIA

```
10 CREATE TABLE dbo.provincia
11 (
12     id_prov INTEGER IDENTITY PRIMARY KEY NOT NULL,
13     nombre_prov VARCHAR(150) NOT NULL,
14     id_dep INTEGER NOT NULL,
15     FOREIGN KEY (id_dep) REFERENCES departamento(id_dep)
16 );
17 GO
```



```
55 INSERT INTO provincia(nombre_prov, id_dep) VALUES
56 ('Murillo', 1),
57 ('Ingavi', 1),
58 ('Sud Yungas', 1),
59 ('Cercado', 2),
60 ('Ayopaya', 2),
61 ('Andres Ibanez', 3),
62 ('German Bush', 3),
63 ('Aniceto Arce', 4);
64 GO
```



NOTA: los INSERT fueron llenados de manera autónoma.

06


CREACION/LLENADO TABLA
PERSONA

NOTA: los **INSERT** fueron llenados de manera autónoma.


```
18 CREATE TABLE dbo.persona
19 (
20     id_per INTEGER PRIMARY KEY NOT NULL,
21     nombre VARCHAR(20) NOT NULL,
22     apellido VARCHAR(50) NOT NULL,
23     fecha_nac DATE NOT NULL,
24     edad INTEGER NOT NULL,
25     email VARCHAR(100) NOT NULL,
26     sexo CHAR(1) NOT NULL,
27     id_dep INTEGER NOT NULL,
28     id_prov INTEGER NOT NULL,
29     FOREIGN KEY (id_dep) REFERENCES departamento(id_dep),
30     FOREIGN KEY (id_prov) REFERENCES provincia(id_prov)
31 );
32 GO
```

```
65 INSERT INTO persona(id_per, nombre, apellido, fecha_nac, edad, email, sexo, id_dep, id_prov) VALUES
66 (00111, 'nombre1', 'apellido1', '10/1/1995', 26, 'nombre1@gmail.com', 'm', 1, 1),
67 (00222, 'nombre2', 'apellido2', '5/11/1988', 32, 'nombre2@gmail.com', 'f', 1, 3),
68 (00333, 'nombre3', 'apellido3', '8/10/1991', 30, 'nombre3@gmail.com', 'm', 3, 6),
69 (00444, 'nombre4', 'apellido4', '11/1/1997', 24, 'nombre4@gmail.com', 'm', 4, 8),
70 (00555, 'nombre5', 'apellido5', '3/26/1995', 26, 'nombre5@gmail.com', 'm', 2, 3),
71 (00666, 'nombre6', 'apellido6', '7/7/1993', 28, 'nombre6@gmail.com', 'f', 1, 2),
72 (00777, 'nombre7', 'apellido7', '4/25/1990', 30, 'nombre7@gmail.com', 'f', 2, 3),
73 (00888, 'nombre8', 'apellido8', '2/13/1990', 30, 'nombre8@gmail.com', 'm', 2, 4),
74 (00999, 'nombre9', 'apellido9', '4/24/1997', 24, 'nombre9@gmail.com', 'm', 3, 7),
75 (01010, 'nombre10', 'apellido10', '4/12/1989', 31, 'nombre10@gmail.com', 'f', 3, 6),
76 (01111, 'nombre11', 'apellido11', '5/9/1993', 28, 'nombre11@gmail.com', 'f', 4, 8),
77 (01212, 'nombre12', 'apellido12', '8/3/1985', 35, 'nombre12@gmail.com', 'm', 1, 1),
78 (01313, 'nombre13', 'apellido13', '3/20/1991', 29, 'nombre13@gmail.com', 'f', 1, 2),
79 (01414, 'nombre14', 'apellido14', '10/24/1989', 31, 'nombre14@gmail.com', 'm', 2, 4),
80 (01515, 'nombre15', 'apellido15', '11/21/2002', 18, 'nombre15@gmail.com', 'm', 3, 7);
81 GO
```



07

CREACION/LLENADO TABLA
PROYECTO

```
33 CREATE TABLE dbo.proyecto
34 (
35     id_proy INTEGER PRIMARY KEY NOT NULL,
36     nombreProy VARCHAR(100) NOT NULL,
37     tipoProy VARCHAR(30) NOT NULL
38 );
39 GO
```



```
82 INSERT INTO proyecto(id_proy, nombreProy, tipoProy) VALUES
83     (3232, 'skynet internet', 'tipo_a'),
84     (3333, 'science lab', 'tipo_b'),
85     (3434, 'reforest', 'tipo_c'),
86     (3535, 'skynet internet', 'tipo_d');
87 GO
```





NOTA: los INSERT fueron llenados de manera autónoma.



08

CREACION/LLENADO TABLA
DETALLES_PROYECTO

```
40 CREATE TABLE dbo.detalle_proyecto
41 (
42     id_detProy INTEGER IDENTITY PRIMARY KEY NOT NULL,
43     id_per INTEGER NOT NULL,
44     id_proy INTEGER NOT NULL,
45     FOREIGN KEY (id_per) REFERENCES persona(id_per),
46     FOREIGN KEY (id_proy) REFERENCES proyecto(id_proy)
47 );
48 GO
```

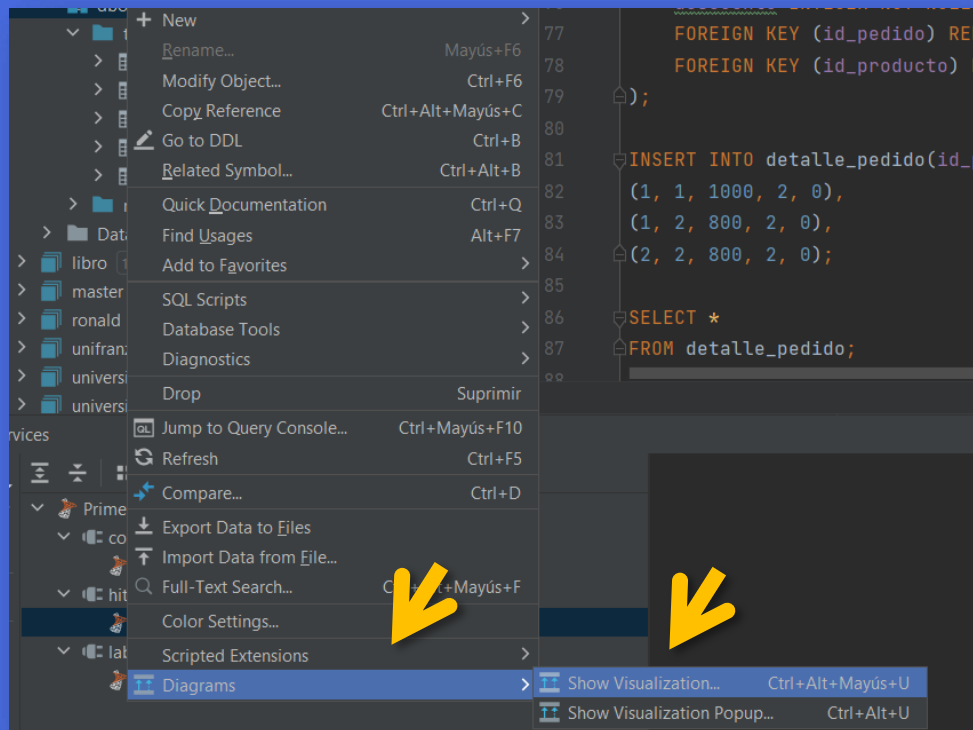


```
88 INSERT INTO detalle_proyecto(id_per, id_proy) VALUES
89     (00111, 3232),
90     (00222, 3434),
91     (00333, 3232),
92     (00444, 3333),
93     (00555, 3434),
94     (00666, 3434),
95     (00777, 3333),
96     (00888, 3232),
97     (00999, 3535),
98     (01010, 3333),
99     (01111, 3333),
100    (01212, 3232),
101    (01313, 3434),
102    (01414, 3535),
103    (01515, 3232);
104 GO
```

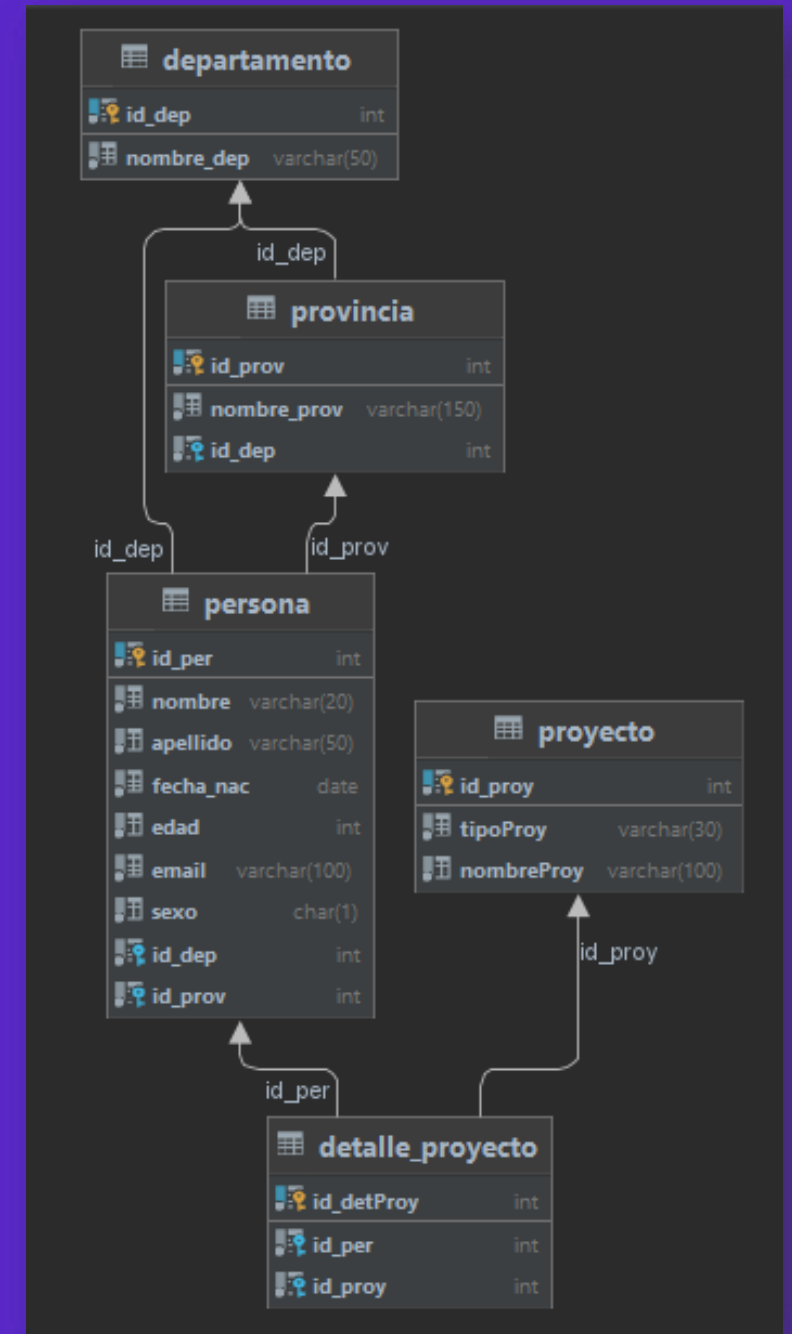


NOTA: los INSERT fueron llenados de manera autónoma.

DIAGRAMA DEL PROYECTO



Haciendo un clic derecho sobre la carpeta de tablas, vaya y seleccione **DIAGRAMS** y la opción **SHOW VISUALIZATION** para poder ver el diagrama.



SECCION: MANEJO DE VISTAS



2.1. MOSTRAR A TODAS LAS PERSONAS QUE VIVEN EN EL DEPARTAMENTO DE COCHABAMBA.

```
121 CREATE OR ALTER VIEW personas_cbba AS
122     SELECT pe.nombre, pe.apellido, pe.id_per, pe.email, dep.nombre_dep
123     FROM persona AS pe
124     INNER JOIN departamento AS dep ON pe.id_dep = dep.id_dep
125     WHERE dep.nombre_dep = 'Cochabamba'
126
127 SELECT pc.*
128 FROM personas_cbba AS pc;
```

personas_cbba	
nombre	varchar(20)
apellido	varchar(50)
id_per	int
email	varchar(100)
nombre_dep	varchar(50)

25%



2.2. MOSTRAR LA PERSONA (NOMBRES Y APELLIDOS) Y EL NOMBRE DEL PROYECTO EN DONDE TRABAJAN.

```
130 CREATE OR ALTER VIEW personas_proyecto AS
131     SELECT pe.nombre, pe.apellido, pro.nombreProy
132     FROM persona AS pe
133     INNER JOIN detalle_proyecto AS det ON pe.id_per = det.id_per
134     INNER JOIN proyecto AS pro ON det.id_proy = pro.id_proy
135     --WHERE pro.nombreProy = 'skynet internet'
136
137 SELECT pp.*
138 FROM personas_proyecto AS pp;
```

personas_proyecto	
nombre	varchar(20)
apellido	varchar(50)
nombreProy	varchar(100)

50%



2.3. ASUMIR QUE TIENE 3 TIPOS DE PROYECTOS (TIPO_A, TIPO_B, TIPO_C)

```
140 CREATE OR ALTER VIEW cam_proy_dep AS
141     SELECT pe.nombre, pe.apellido, pro.tipoProy AS tipo_poyecto, deparatemeto_a_aplicarse
142     CASE
143         WHEN pro.tipoProy = 'tipo_a' THEN 'CBBA'
144         WHEN pro.tipoProy = 'tipo_b' THEN 'LPZ'
145         WHEN pro.tipoProy = 'tipo_c' THEN 'SCZ'
146         ELSE 'se analiza a TJ'
147     END
148     FROM persona AS pe
149     INNER JOIN detalle_proyecto AS det ON pe.id_per = det.id_per
150     INNER JOIN proyecto AS pro ON det.id_proy = pro.id_proy
151
152 SELECT cpd.*
153 FROM cam_proy_dep AS cpd;
```

El objetivo es crear una Vista con los mismos datos de la tabla proyectos

- Sin embargo generar una nueva columna en la vista de nombre **departamento_aplicarse**
- Si el tipo de proyecto es de **TIPO_A** asignar **CBB**
- Si el tipo de proyecto es de **TIPO_B** asignar **LPZ**
- Si el tipo de proyecto es de **TIPO_C** asignar **SCZ**
- Si el tipo de proyecto es **otro tipo** asignar **'En proceso de análisis'**

cam_proy_dep	
nombre	varchar(20)
apellido	varchar(50)
tipo_poyecto	varchar(30)
deparatemeto_a_aplicarse	varchar(15)

75%



2.4. CREAR UNA VISTA CUALQUIERA QUE MUESTRE 5 COLUMNAS.

```
155 CREATE OR ALTER VIEW destino AS
156 SELECT pe.id_per, pe.nombre, pe.apellido, dep.nombre_dep AS departamento, destino_departamento =
157 CASE
158     WHEN pe.id_per > 100 AND pe.id_per <= 500 THEN 'TARIJA'
159     WHEN pe.id_per > 500 AND pe.id_per <= 900 THEN 'SANTA CRUZ'
160     WHEN pe.id_per > 900 AND pe.id_per <= 1300 THEN 'LA PAZ'
161     WHEN pe.id_per > 1300 THEN 'COCHABAMBA'
162 END
163 FROM persona AS pe
164 INNER JOIN detalle_proyecto AS det ON pe.id_per = det.id_per
165 INNER JOIN departamento AS dep ON pe.id_dep = dep.id_dep
166
167 SELECT des.*
168 FROM destino AS des;
```

.....→ Generar la vista en la que se muestra los datos de a persona, el departamento donde trabaja y el departamento al cual será destinado.

destino		
id_per		int
nombre		varchar(20)
apellido		varchar(50)
departamento		varchar(50)
destino_departamento		varchar(10)

100%



SECCION: MANEJO DE FUNCIONES

BDA



3.1. CREAR UNA FUNCIÓN QUE PERMITA SABER CUÁNTOS PROYECTOS DISTINTOS DEL TIPO_A, TIPO_B Y TIPO_C EXISTEN

.....→ La función no recibe ningún parámetro.

```
171  ✓ SELECT COUNT(pro.nombreProy) AS proyectos_distinto
172      FROM proyecto AS pro
173      WHERE pro.id_proy != 3232 AND pro.id_proy != 3333 AND pro.id_proy != 3434
```

	proyectos_distinto ▾
1	1

.....→ Funcion COUNT de agregacion, es propia del sistema.

25%



3.2. CREAR UNA FUNCIÓN QUE GENERE LOS PRIMEROS N NÚMEROS IMPARES.

```
172 CREATE OR ALTER FUNCTION numeros_naturales(@parametro1 INT)
173 RETURNS VARCHAR(100) AS
174 BEGIN
175     DECLARE @respuesta VARCHAR(100) = '';
176     DECLARE @contador INTEGER = 1;
177
178     WHILE @contador <= @parametro1
179     BEGIN
180         SET @respuesta = CONCAT(@respuesta, @contador, ', ');
181         SET @contador = @contador + 2;
182     END;
183
184     RETURN @respuesta;
185 END;
186 GO
187
188 SELECT dbo.numeros_naturales( @parametro1: 10) AS secuencia_de_impares;
```

	secuencia_de_impares
1	1, 3, 5, 7, 9,

50%



3.3. CREAR UNA FUNCIÓN QUE PERMITA INSERTAR UN REGISTRO A LA TABLA PERSONA.

```
196 CREATE OR ALTER PROCEDURE agregar_registro_persona
197 @id INT,
198 @nombre VARCHAR(20),
199 @apellido VARCHAR(50),
200 @fecha DATE,
201 @edad INT,
202 @email VARCHAR(100),
203 @sexo CHAR(1),
204 @id_dep INT,
205 @id_prov INT
206 AS
207 BEGIN
208     INSERT INTO persona VALUES
209     (@id, @nombre, @apellido, @fecha, @edad, @email, @sexo, @id_dep, @id_prov)
210 END;
211
212 EXEC agregar_registro_persona @id: 01717, @nombre: 'nombre17', @apellido: 'apellido17', @fecha: '6/2/1998'
213 --para verificar
214 SELECT *
215 FROM persona;
```

La función recibe los datos a insertarse a la tabla persona

- La función retorna un mensaje indicando que se insertó satisfactoriamente el registro.

Se dio el uso de un procedimiento a ser explicado en el video adjunto

75%



3.4. CREAR UNA FUNCIÓN CUALQUIERA: "SERIE FIBONACCI"

```
218 CREATE OR ALTER FUNCTION serie(@limite INT)
219 RETURNS VARCHAR(100) AS
220 BEGIN
221     DECLARE @resp VARCHAR(100) = '';
222     DECLARE @i INTEGER = 0;
223     DECLARE @siguiente INTEGER = 1;
224     DECLARE @uno INTEGER = 0;
225     DECLARE @dos INTEGER = 1;
226
227     WHILE @i <= @limite
228     BEGIN
229         SET @i = @i + 1;
230         IF @i <= 1
231         BEGIN
232             SET @siguiente = @i;
233         end
234         ELSE
235         BEGIN
236             SET @siguiente = @uno + @dos;
237             SET @uno=@dos;
238             SET @dos = @siguiente;
239
240         end
241
242         SET @resp = CONCAT(@resp, @siguiente, ', ');
243     END;
244     RETURN @resp;
245 end;
```



.....→ La lógica fue traspasada del lenguaje C# a SQL

```
static void ejercicio3(int fib)
{
    int siguiente, uno = 0, dos = 1;
    for (int i = 0; i <= fib; i++)
    {
        if (i <= 1)
        {
            siguiente = i;
        }
        else
        {
            siguiente = uno + dos;
            uno = dos;
            dos = siguiente;
        }
        Console.WriteLine("{0} - ", siguiente);
    }
}
```

```
246
247 SELECT dbo.serie( @limite: 17) AS serie_Fibonacci;
```

100%





¡GRACIAS POR SU ATENCION!

ronald.choque2111@gmail.com

Cel. +59165648933



UNIFRANZ
Internacionalizate

INNOVACIÓN
EN EDUCACIÓN



EL ALTO, NOVIEMBRE DE 2021