

Evaluación 3 - Base de Datos II

Puntos totales 18.5/27 ?

11 Febrero 2024, 7:00 AM a 9:00AM

Estimado estudiante,

La evaluación comprende manejo funciones escalares, procedimientos almacenados, triggers, ETL (SSIS) and Power BI

0 de 0 puntos

Apellidos y Nombres *

Caruajulca Tiglla Alex Eli

Código de Alumno *

2022110044

Pregunta 1 - Funciones

1 de 2 puntos

Lea atentamente y conteste la siguiente pregunta



Crear una función **fn_gen_num_xxx**, donde xxx son las iniciales de cada alumno, La función debe permitir ingresar 2 parámetros de tipo entero (variables desde y hasta) y debe retornar un valor entero.

Emplear la siguiente porción de código o instrucción como base de construcción de la función:

ABS(round(rndResult * @ParamHasta - @ParamDesde,0)) from rndView;

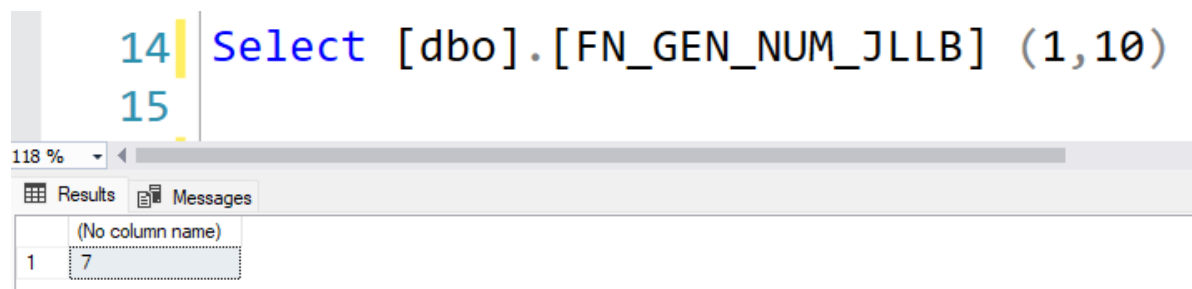
Esta porción de código va a generar un valor aleatorio a partir de los parámetros @ParamHasta y @ParamDesde.

El resultado generado debe guardarse en una variable de retorno del mismo tipo de datos.

Si como resultado del uso de la función se retorna 0 colocar un valor por defecto.

Probar la función pasándole parámetros como en la imagen.

Subir o pegar el código aquí.



```
CREATE FUNCTION [dbo].[FN_GEN_NUM_AECT]
(
    @ParamDesde INT,
    @ParamHasta INT
)
RETURNS INT
AS
BEGIN
    DECLARE @RandomNumber INT, @RandomValue FLOAT;

    SET @RandomValue = CONVERT(FLOAT, CONVERT(BIGINT, (SELECT @@DBTS)));
    SET @RandomValue = @RandomValue / (SELECT COALESCE(MAX(database_id), 1)
FROM sys.databases);
    SET @RandomValue = @RandomValue - FLOOR(@RandomValue);
    SET @RandomValue = @RandomValue * (SELECT COALESCE(MAX(database_id), 1)
FROM sys.databases);

    IF (@ParamHasta - @ParamDesde) = 0
    BEGIN
```

```
        SET @RandomNumber = 1;
    END
    ELSE
    BEGIN
        SET @RandomNumber = ABS(round(@RandomValue * (@ParamHasta - @ParamDesde)
+ @ParamDesde, 0));
    END

    RETURN @RandomNumber;
END

SELECT [dbo].[FN_GEN_NUM_AECT](1, 10);
.....
```

Comentarios

Al ejecutar su funcion solo muestra 190 como resultado las veces que se ejecuta su fn

Pregunta 2 - QUERY CREATE TABLE AND INSERT DATA

2.5 de 3 puntos

Lea atentamente y conteste la siguiente pregunta



Crear la tabla MM_Scorecred_xxx y poblarla (Cambiar xxx por las iniciales *2.5/3 de sus apellidos y nombres)

Para crear la tabla con 3 campos se debe leer el código de cliente de la tabla MM_cliente y **usando la función fn_gen_num_xxx** creada anteriormente para generar un numero aleatorio, poblar la columna de score crediticio. Los parámetros de rango del score crediticio deben oscilar entre 1 y 10. La fecha de carga debe ser la fecha del sistema.

Aplicar INSERT INTO SELECT, al QUERY para poblar la tabla.

Subir el código SQL

TABLA MM_SCORECRED		
CAMPO	TIPO DATO	VALOR DEFECTO
CODCLI	CADENA	"."
SCORECRED	ENTERO	
FECHACARGA	FECHAHORA	FECHASISTEMA

```
CREATE TABLE MM_Scorecred_AECT (  
    CODCLI VARCHAR(50) NOT NULL,  
    SCORECRED INT NOT NULL,  
    FECHACARGA DATETIME NOT NULL DEFAULT GETDATE()  
);
```

```
INSERT INTO MM_Scorecred_AECT (CODCLI, SCORECRED)  
SELECT CODCLI, DBO.FN_GEN_NUM_AECT(1, 10)  
FROM MM_cliente;
```

```
SELECT * FROM MM_Scorecred_AECT
```

Pregunta 3 - Funciones

3 de 6 puntos

Lea atentamente y conteste la siguiente pregunta



Crear una función llamada FN_CREDITO_XXX, donde xxx son las iniciales de cada alumno, que permita determinar el crédito que se puede otorgar a los clientes de acuerdo a las ventas del último periodo y además, de acuerdo al ranking crediticio de cada cliente. Es decir, lo guardado en la tabla MM_scorecred. El objetivo de la función es calcular crédito aprobado redondeado a enteros de un cliente. *3/6

Para ello la función recibirá el código de cliente y retornará el crédito aprobado.

En la lógica de la función se deben de sumar las ventas del último periodo registrado en la tabla HD_PEDIDO y agruparlo por nombre de cliente y score crediticio (tablas MM_cliente y MM_scorecred).

La información de ventas y el score crediticio del cliente deben ser guardadas en variables (una para ventas y otra para el score crediticio)

Con esas variables debemos crear una lógica usando IF para aplicar las reglas mencionadas en la tabla adjunta.

Es decir, si ventas estan en el rango de 1000 y 5000 la lineacredito base será de 10% del tope de 5000.

Si el scorecrediticio está en el rango de 1 a 3

el crédito aprobado será el 50% de la líneaadecredito base,

La función debe de retornar el importe del crédito aprobado en base a la lógica de la tabla adjunta.

Subir el código aquí

Logica de Crédito Base de Acuerdo a el ultimo periodo registrado							
Desde	Hasta	Regla desde	Regla hasta	Ejemplo Si Ventas :	Porcentaje Línea Crédito	Línea Base	Crédito
1000	5000	> 1000	<= 5000	4500	10%	500	
5001	10000	> 5000	<= 10000	8000	25%	2500	
10001	25000	> 10001	<= 25000	20000	35%	8750	
25001	45000	> 25001	<= 45000	45000	45%	20250	
45001	100000	> 45001	<= 100000	50000	50%	50000	
Logica de Disponibilización Crédito				Crédito Aprobado			
Rango Score	Porcentaje Disponible	500	2500	8750	20250	50000	
1 a 3	50%	250	1250	4375	10125	25000	
4 a 8	75%	375	1875	6562.5	15187.5	37500	
9 a 10	100%	500	2500	8750	20250	50000	

CREATE FUNCTION FN_CREDITO_AECT

(

@CODCLI VARCHAR(50)



```

)
RETURNS INT
AS
BEGIN
    DECLARE @VENTAS DECIMAL(18, 2), @SCORECRED INT, @LINEACREDITOBASE
    DECIMAL(18, 2), @CREDITOAPROBADO DECIMAL(18, 2);

    SELECT @VENTAS = SUM(MTOVALUNIT)
    FROM HD_PEDIDO h
    JOIN MM_CLIENTE c ON h.CODCLI = c.CODCLI
    WHERE h.CODCLI = @CODCLI
    AND h.FECPEDID = (SELECT MAX(FECPEDID) FROM HD_PEDIDO WHERE CODCLI =
    @CODCLI);

    SELECT @SCORECRED = SCORECRED
    FROM MM_Scorecred_AECT
    WHERE CODCLI = @CODCLI;

    --COMPARACIÓN VENTAS
    IF @VENTAS >= 1000 AND @VENTAS < 5000
    BEGIN
        SET @LINEACREDITOBASE = @VENTAS * 0.10;
    END
    ELSE
    BEGIN
        SET @LINEACREDITOBASE = @VENTAS * 0.05;
    END

    IF @SCORECRED >= 1 AND @SCORECRED <= 3
    BEGIN
        SET @CREDITOAPROBADO = @LINEACREDITOBASE * 0.50;
    END
    ELSE IF @SCORECRED >= 4 AND @SCORECRED <= 8
    BEGIN
        SET @CREDITOAPROBADO = @LINEACREDITOBASE * 0.75;
    END
    ELSE IF @SCORECRED >= 9 AND @SCORECRED <= 10
    BEGIN
        SET @CREDITOAPROBADO = @LINEACREDITOBASE;
    END
    ELSE
    BEGIN
        SET @CREDITOAPROBADO = 0;
    END

    RETURN ROUND(@CREDITOAPROBADO, 0);
END

```



Al ejecutar su funcion devuel credito otrogado al cliente 0 dado que su funcion indicaba 190 como valor aleatorio y dicho valor se cargo en la tabla anterior

Asimismo al correo con una funcion que si genera numeros aleatorios el valor que arroja su funcion es 9

Pregunta 4 - Store Procedure

2 de 4 puntos

Lea atentamente y conteste la siguientes pregunta



Elaborar un procedimiento almacenado SP_ADD_CARGOS_xxx que permita darle mantenimiento a la tabla CL_CARGOS, agregando 2 nuevos cargos

Ingeniero en Soluciones AWS/AZURE/GOOGLECLOUD y Científico de Datos

Asignar un código, salario mínimo, máximo y fecha de registro

Asimismo, debe actualizar el salario del Ingeniero de Datos (de 4mil a 7500 y de 10000 a 15000), también debe eliminar un cargo (vendedor 2 o vendedor 3)

Subir el código aquí.

```
CREATE PROCEDURE SP_ADD_CARGOS_AECT
AS
BEGIN
```

```
    INSERT INTO CL_CARGOS (ID_CARGOS, NOMBRE_CARGO, SALARIO_MINIMO,
SALARIO_MAXIMO)
    VALUES ('AWS001', 'Ingeniero en Soluciones AWS/AZURE/GOOGLECLOUD', 0, 0),
           ('DATA001', 'Científico de Datos', 0, 0);
```

```
    UPDATE CL_CARGOS
    SET SALARIO_MINIMO = 7500,
        SALARIO_MAXIMO = 15000
    WHERE NOMBRE_CARGO = 'Ingeniero de Datos';
```

```
    DELETE FROM CL_CARGOS
    WHERE NOMBRE_CARGO = 'Vendedor 2' OR NOMBRE_CARGO = 'Vendedor 3';
END;
```

Comentarios

Su procedimiento almacenado al ejecutar insertar los mismo valores al momento de ejecutar las veces que se ejecuta su SP, por tanto, debe de pasar parametros a su SP

Pregunta 5 - Triggers

4 de 4 puntos

Lea atentamente y conteste la siguientes pregunta

Crear un Trigger TG_EST_TAB_CARGOS_xxx que muestre un mensaje cuando se agrega un nuevo cargo, cuando se actualiza un cargo y cuando se elimina un cargo

Subir el código aquí.

```
CREATE TRIGGER TG_EST_TAB_CARGOS_AECT
ON CL_CARGOS
AFTER INSERT, UPDATE, DELETE
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @Mensaje NVARCHAR(1000);

    IF EXISTS(SELECT * FROM inserted) AND NOT EXISTS(SELECT * FROM deleted)
    BEGIN
        SET @Mensaje = 'Se ha agregado un nuevo cargo.';
    END
    ELSE IF EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT * FROM deleted)
    BEGIN
        SET @Mensaje = 'Se ha actualizado un cargo.';
    END
    ELSE IF NOT EXISTS(SELECT * FROM inserted) AND EXISTS(SELECT * FROM deleted)
    BEGIN
        SET @Mensaje = 'Se ha eliminado un cargo.';
    END

    PRINT @Mensaje;
END;
```

Pregunta 6 - Power BI

3 de 4 puntos

Lea atentamente y conteste la siguiente pregunta



Elaborar un reporte donde se filtre por vendedor, y se muestre el % de ventas segun vendedor y al mismo tiempo se muestre el % de ganancias obtenidas.

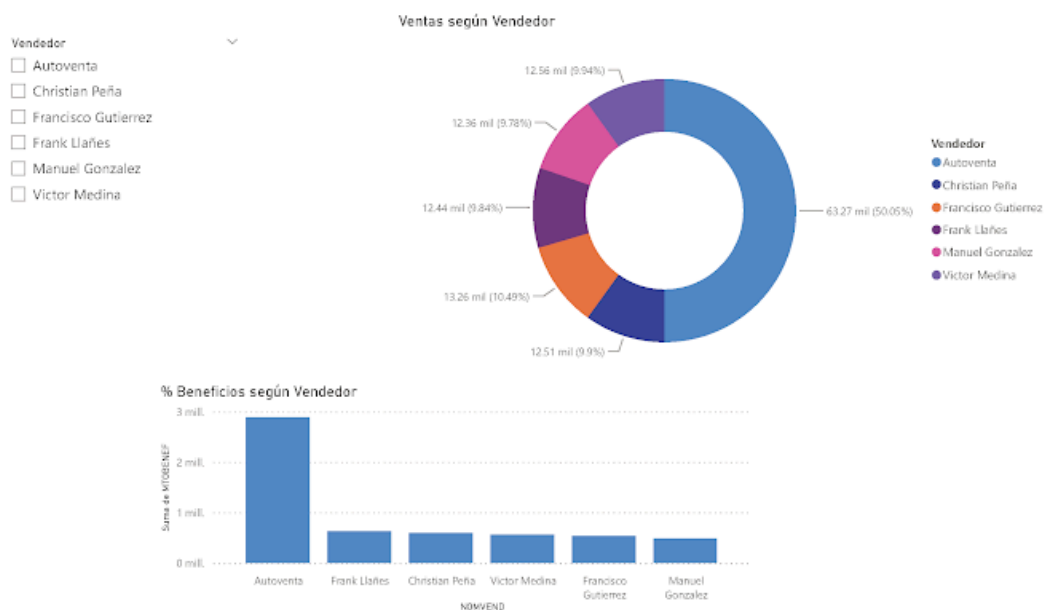
3/4

Debe agregar los titulos y leyendas respectivas, a fin de que no se muestre el nombre de la columna de la tabla

Debe hacer uso de las tablas HD_PEDIDO y MM_Vendedor

Luego Exportar en PDF y guardar con la siguiente secuencia ReportePBI_xxx

Finalmente adjuntar el archivo



PDF ReportePBI_AEC...

Añadir archivo

Comentarios

Falta el filtro del vendedor

Pregunta 7 - ETL

3 de 4 puntos



Lea atentamente y conteste la siguiente pregunta

Crear el ETL con Integration Services de Carga de la información de clientes. *3/4

Cree la base de datos **TALLERDWH_XXX**

Ejecute el siguiente código

```
/*          SCHEMAS          */

IF NOT EXISTS (SELECT NAME FROM SYS.SCHEMAS WHERE NAME ='UNC')

EXEC('CREATE SCHEMA [UNC]')

GO

/*          TABLES          */

-- CREACION DE LA CAPA STAGING DEL CICLO DE VIDA DE LOS DATOS

CREATE TABLE [UNC].[T_CLIENTE] (

    [IDCLI]      INT NOT NULL IDENTITY(1,1),

    [CODCLI]     NVARCHAR(40) COLLATE SQL_Latin1_General_CP1_CI_AS,

    [NOMCLI]     NVARCHAR(200) COLLATE SQL_Latin1_General_CP1_CI_AS,

    [CODUBIGEO]  INT,

    [CODSECTECON] INT,

    [CODMRCADO]  INT,

    [FECCARGA]   DATETIME,

    [FECACTUALIZA] DATETIME DEFAULT GETDATE()

)
```

Luego generar una consulta usando la tabla CLIENTES de la base de datos **tallersql** que permita extraer la información y poblar la tabla UNC.T_Cliente

Realizar las conexiones al servidor respectivamente

En Flujo de control

La tarea SQL es para limpiar datos con truncate a la tabla y para el proceso se usa Data Flow Task

En Flujo de datos



Se usar Origen OLE DB y Destino OLE DB

Pegar su consulta aqui y la cantidad de registro poblados.



USE TALLERDWH_XXX;

CREATE SCHEMA UNC;

```
CREATE TABLE UNC.T_CLIENTE (  
    [IDCLI] INT NOT NULL IDENTITY(1,1),  
    [CODCLI] NVARCHAR(40) COLLATE SQL_Latin1_General_CP1_CI_AS,  
    [NOMCLI] NVARCHAR(200) COLLATE SQL_Latin1_General_CP1_CI_AS,  
    [CODUBIGEO] INT,  
    [CODSECTECON] INT,  
    [CODMRCADO] INT,  
    [FECCARGA] DATETIME,  
    [FECACTUALIZA] DATETIME DEFAULT GETDATE()  
);
```

```
INSERT INTO UNC.T_CLIENTE (  
    CODCLI,  
    NOMCLI,  
    CODUBIGEO,  
    CODSECTECON,  
    CODMRCADO,  
    FECCARGA  
)  
SELECT  
    CAST(ID_CLIENTE AS NVARCHAR(40)) AS CODCLI,  
    CLI_NOMBRES + ' ' + CLI_APELLIDOS AS NOMCLI,  
    CUST_CIUADAD_ID AS CODUBIGEO,  
    NULL AS CODSECTECON,  
    NULL AS CODMRCADO,  
    GETDATE() AS FECCARGA  
FROM  
    tallersql.dbo.CL_CLIENTE;
```

TRUNCATE TABLE UNC.T_CLIENTE;

Este formulario se creó en Universidad Nacional de Cajamarca.
Does this form look suspicious? [Informe](#)

Google Formularios



