



# UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

# FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

MAESTRÍA EN CIENCAS DE DATOS

BASES DE DATOS RELACIONALES

MAESTRO: JOSÉ ALBERTO BENAVIDES VÁZQUEZ

# TAREA#9

ALUMNO: EDWIN MARTÍN ROMERO SILVA

MATRÍCULA: 1731276

# Tarea 9

- 1.-Usa lo aprendido para crear al menos 2 funciones o procedimientos almacenados que calculen alguno de los siguientes resultados:
- a) Correlación entre 2 conjuntos de datos.
- b) Regresión lineal entre 2 variables.
- c)Distancia de Levensthein entre cadenas de caracteres.
- d)Cantidad de elementos de un arreglo.
- e)Seasonal naive para series de tiempo.

#### 1.1.-Función Correlación

/\*Función que calcula la correlación entre las variables height y weigth de la tabla master\*/

```
CREATE FUNCTION correlacion()
RETURNS decimal(10, 2)
DETERMINISTIC
BEGIN
    DECLARE mean1 decimal(10, 2);
    DECLARE mean2 decimal(10, 2);
    DECLARE sum1 decimal(20, 2);
    DECLARE sum2 decimal(20, 2);
    DECLARE count1 int;
    DECLARE count2 int;
    DECLARE covariance decimal(20, 2);
    DECLARE std1 decimal(10, 2);
    DECLARE std2 decimal(10, 2);
    DECLARE correlation decimal(10, 2);
    #Promedios
    SELECT AVG (height) INTO mean1 FROM master;
    SELECT AVG(weight) INTO mean2 FROM master;
    #Suma
    SELECT SUM (height) INTO sum1 FROM master;
    SELECT SUM (weight) INTO sum2 FROM master;
    #Conteo
    SELECT COUNT (height) INTO count1 FROM master;
    #Covarianza
    SELECT SUM((height - mean1) * (weight - mean2))
    INTO covariance
    FROM master;
    SET covariance = covariance / count1;
    #Desviación Estándar
    SELECT STD(height) INTO std1 FROM master;
    SELECT STD (weight) INTO std2 FROM master;
   SET correlation = covariance / (std1 * std2);
    RETURN correlation;
END;
SELECT correlacion() AS correlacion limit 1;
      123 correlacion
                 0,68
```

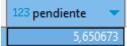
### 1.2.-Regresión Lineal

Con este código calculo los elementos m y b de la fórmula de regresión lineal: Y = mx + b

Para hacerlo utilizo en realidad 2 procedimientos, por lo que podríamos decir que este es un procedimiento anidado. Explicación más adelante.

# 1.2.1.-Regresión Lineal: Procedimiento 'pendiente'

```
/*Procedimiento que, en una regresión lineal entre las variables height y
weigth de la tabla master, calcula la pendiente */
drop procedure if exists pendiente;
CREATE PROCEDURE pendiente (OUT pendiente decimal(30,6))
BEGIN
    DECLARE n int;
    DECLARE arriba decimal (30,6);
    DECLARE suma_height decimal(10,2);
    DECLARE suma weight decimal(10,2);
    DECLARE suma hw decimal (30, 6);
    DECLARE abajo decimal(30,6);
    DECLARE sum height pow2 decimal(30,6);
    DECLARE pow2 sum height decimal(30,6);
    SELECT COUNT (height) INTO n
    FROM master;
    SELECT SUM (height) INTO suma height
    FROM master;
    SELECT SUM (weight) INTO suma weight
    FROM master;
    SELECT SUM (height*weight) INTO suma hw
    FROM master;
    SET arriba = n * suma hw - suma height*suma weight;
    SELECT SUM(POW(height,2)) INTO sum_height_pow2
    FROM master;
    SELECT POW (SUM (height), 2) INTO pow2 sum height
    FROM master;
    SET abajo = n * sum height pow2 - pow2 sum height;
    SET pendiente = arriba/abajo;
END;
SET @pendiente = 0;
CALL pendiente (@pendiente);
SELECT @pendiente AS pendiente;
```



# 1.2.2.-Regresión Lineal: Procedimiento 'constante'

```
/*Procedimiento que, en una regresión lineal entre las variables height y
weight de la tabla master, calcula la constante*/
drop procedure if exists constante;
CREATE PROCEDURE constante (OUT constante decimal (30,6))
BEGIN
    DECLARE suma height decimal(10,2);
    DECLARE suma weight decimal(10,2);
    DECLARE n int;
    SET @pendiente = 0;
    CALL pendiente (@pendiente);
    SELECT COUNT(height) INTO n FROM master;
    SELECT SUM (height) INTO suma height FROM master;
    SELECT SUM (weight) INTO suma weight FROM master;
    SET constante = (suma weight - (@pendiente*suma height))/n;
END;
SET @constante = 0;
CALL constante(@constante);
SELECT @constante AS constante;
      123 constante
```

# 2.-Guarda tu código en un archivo SQL y publicalo en tu repositorio.

Guardé el código, pero lo incluiré en este mismo PDF para subir solo 1 archivo.

■ TAREA\_9\_EDWIN\_ROMERO 02/07/2023 11:11 a. m. Archivo de origen ... 4 KB

#### 3.-Genera un reporte donde expliques como funciona tu código:

En esta parte voy a explicar que hace cada sección del código, tanto para la función correlación, como para los 2 procedimientos que en conjunto calculan los elementos para una regresión lineal.

#### 3.1.-Función Correlación

Hice una función llamada 'Correlacion' que calcula la correlación entre las variables height y weight de la tabla Master, la cual contiene información de jugadores.

Dentro de la función utilicé estas 2 fórmulas para calcular la correlación:

$$Cov(height, weight) = \frac{\sum_{i=1}^{N} (height_i - MeanHeight)(weight_i - MeanWeight)}{n}$$

$$Correlaci\'on = \frac{Cov(height, weight)}{STD(height) * STD(weight)}$$

### Esta es la explicación del código:

123 correlacion

```
CREATE FUNCTION correlacion()
RETURNS decimal(10, 2)
DETERMINISTIC
BEGIN
    DECLARE mean1 decimal(10, 2);
    DECLARE mean2 decimal(10, 2);
    DECLARE sum1 decimal(20, 2);
                                                     En esta parte estoy declarando todas las
    DECLARE sum2 decimal(20, 2);
                                                     variables necesarias en la función.
    DECLARE count1 int;
    DECLARE count2 int;
    DECLARE covariance decimal(20, 2);
    DECLARE stdl decimal(10, 2);
    DECLARE std2 decimal(10, 2);
    DECLARE correlation decimal(10, 2);
    #Promedios
    SELECT AVG (height) INTO mean1 FROM master;
    SELECT AVG (weight) INTO mean2 FROM master;
    #Suma
    SELECT SUM (height) INTO sum1 FROM master;
                                                               En este parte calculo todos los elementos
    SELECT SUM (weight) INTO sum2 FROM master;
                                                               necesarios en la fórmula de correlación y
    #Conteo
                                                               covarianza (necesaria para la fórmula de
    SELECT COUNT (height) INTO count1 FROM master;
                                                               correlación) y guardo todos esos cálculos en
    #Covarianza
                                                               las variables anteriormente declaradas.
    SELECT SUM((height - mean1) * (weight - mean2))
    INTO covariance
    FROM master;
    SET covariance = covariance / count1;
    #Desviación Estándar
    SELECT STD(height) INTO std1 FROM master;
    SELECT STD(weight) INTO std2 FROM master;
                                                                      Aquí calculo el valor final.
   SET correlation = covariance / (std1 * std2);
    RETURN correlation;
END;
                                                      Para mandar llamar la función utilizamos un
SELECT sorrelation() AS sorrelation limit 1;
                                                      select y el nombre de la función.
```

Limitamos la búsqueda a 1 registro, ya que si

no lo hacemos, nos trae una tabla completa

rellena del mismo valor '0.68'.

#### 3.2.-Regresión Lineal

Quise hacer un procedimiento que calculara una regresión lineal entre 2 variables: Y = mx + bPero lo hice en partes, utilicé 2 procedimientos:

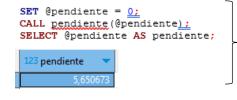
- \*Procedimiento 'pendiente': calcula la pendiente 'm'.
- \*Procedimiento 'constante': calcula la constante 'b'. Utiliza el procedimiento anterior, por lo que podríamos decir que es un procedimiento anidado.

Para el procedimiento 'pendiente' utilizo esta fórmula:

$$a = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

Esta es la explicación del código:

```
CREATE PROCEDURE pendiente (OUT pendiente decimal (30,6))
BEGIN
    DECLARE n int;
    DECLARE arriba decimal (30,6);
    DECLARE suma height decimal (10,2);
    DECLARE suma weight decimal(10,2);
                                                   Declaro las variables necesarias
    DECLARE suma bw decimal (30,6);
                                                   en el procedimiento.
    DECLARE abajo decimal(30,6);
    DECLARE sum_height_pow2 decimal(30,6);
    DECLARE pow2 sum height decimal (30,6);
    SELECT COUNT (height) INTO n
    FROM master:
    SELECT SUM(height) INTO suma height
    FROM master;
                                                                Calculo el numerador de la
    SELECT SUM (weight) INTO suma weight
                                                                fórmula de la pendiente.
    FROM master;
    SELECT SUM (height*weight) INTO suma hw
    FROM master;
    SET arriba = n * suma hw - suma height*suma weight:
    SELECT SUM(POW(height, 2)) INTO sum height pow2
    FROM master:
                                                                Calculo el denominador de
    SELECT POW(SUM(height),2) INTO pow2 sum height
                                                                la fórmula de la pendiente.
    FROM master:
    SET abajo = n * sum_height_pow2 - pow2_sum_height;
    SET pendiente = arriba/abajo;
                                                               Hago numerador/denominador para
END;
                                                               calcular el OUT del procedimiento.
```



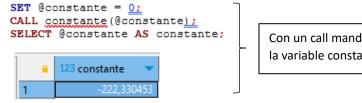
Con un call mando llamar el procedimiento y lo guardo en la variable pendiente, la cual mando llamar con un select.

Para el procedimiento 'constante utilizo esta fórmula:

$$b = \frac{\sum y - a \sum x}{n}$$

Esta es la explicación del código:

```
CREATE PROCEDURE constante(OUT constante decimal(30,6))
BEGIN
                                                Declaro las variables necesarias
    DECLARE suma height decimal(10,2);
    DECLARE suma weight decimal(10,2);
                                                para el procedimiento.
    DECLARE n int;
    SET @pendiente = 0;
                                        Mando llamar el procedimiento 'pendiente' y lo guardo en una variable
    CALL pendiente (@pendiente)
    SELECT COUNT (height) INTO n FROM master;
                                                                 Calculo los elementos necesarios en la
    SELECT SUM(height) INTO suma height FROM master;
                                                                 fórmula y guardo esos cálculos en las
    SELECT SUM(weight) INTO suma weight FROM master;
                                                                 variables anteriormente declaradas.
    SET constante = (suma_weight - (@pendiente*suma_height))/n;
END;
                                                                     Calculo el OUT del procedimiento.
```



Con un call mando llamar el procedimiento y lo guardo en la variable constante, la cual mando llamar con un select.

Una vez que tenemos ambos procedimientos listos, tenemos los m y b de la expresión: Y = mx + b y podemos estimar el valor de Weight con la variable Height.

SELECT height, weight, @pendiente\*height + @constante AS WEIGHT\_ESTIMADA
FROM master;

123 height	•	123 weight	•	123 WEIGHT_ESTIMADA 🔻
	75		220	201,470022
	72		180	184,518003
	75		190	201,470022
	75		190	201,470022
	73		184	190,168676
	73		220	190,168676
	72		192	184,518003
	71		170	178,86733
	71		175	178,86733
	68		169	161,915311
	71		190	178 86733

Apliqué la regresión en Python para verla gráficamente.

```
def regresion (x):
    y = (5.65*x) - 222.33
    return y

df['weight_estimada'] = df['height'].apply(regresion)

x = df['height']
y = df['weight']
z = df['weight_estimada']

plt.scatter(x, y, c ='blue')
plt.plot(x, z, c ='red')

plt.title('Y = 5.65x - 222.33')
plt.xlabel('Height')
plt.ylabel('Weight')
```

