Práctica 07

Se deberán entregar los archivos .sql y .txt correspondientes vía Moodle **antes** de las 23:59 del 22 de noviembre de 2022. No se aceptarán entregas extemporáneas o por otro medio.

Objetivo

Generar un script tipo SQL llamado Practica07_EqN (donde N indica el número de equipo). Por ejemplo, para el equipo 03 el nombre del script será Practica07_Eq03.sql

Por ningún motivo el script deberá generar errores, sin importar el número de veces que se corra. Deberán considerar que la calificadora cuenta con todas las bases de datos que se mencionan en los encabezados de las partes.

No se calificarán instrucciones en donde manipulaciones convenientes hayan intervenido para llegar al resultado esperado.

Parte 0 – Preliminares

Cambiar el prompt de manera que muestre la fecha actual completa y el nombre de la base de datos en uso, seguido de '->'

Parte 1 – Base de datos 'replicas_pr'

1. Crear una función que calcule la diferencia en semanas que se presenta entre dos fechas dadas. La función podrá llamarse como prefieran, pero deberá funcionar como lo hace `semanas_pedido`:

```
+-----------------------------------------+-----------------------------------+
| semanas_pedido('2022/11/15','2022/11/24') | semanas_pedido('2022/11,15',NULL) |
+-----------------------------------------+-----------------------------------+
|                                    1.30 |                             -1.00 |
+-----------------------------------------+-----------------------------------+
1 row in set, 3 warnings (0.00 sec)
```

15/100

2. Usar la función creada en 1 para almacenar en una vista el promedio, mínimo y máximo número de semanas que le toma a la tienda enviar un pedido. No se deben tomar en cuenta las órdenes que no han sido despachadas aún.

15/100

3. Por cada una de las ciudades en las que hay oficinas:

- contar el número de empleados,
- tiempo de procesamiento de órdenes promedio en semanas y
- asignar una etiqueta que alerte si en esa ciudad se supera el tiempo promedio calculado sobre todas las órdenes.

La tabla resultado tendrá que visualizarse exactamente como la siguiente:

```
+---------------+-------------+---------------+--------+
| Ciudad        | # empleados | TP por ciudad | Aviso  |
+---------------+-------------+---------------+--------+
| TOKYO         |           2 |    ->1.17<-   | ALERTA |
| BOSTON        |           2 |    ->0.48<-   | OK     |
| NYC           |           2 |    ->0.44<-   | OK     |
| PARIS         |           5 |    ->0.44<-   | OK     |
| SAN FRANCISCO |           6 |    ->0.43<-   | OK     |
| LONDON        |           2 |    ->0.42<-   | OK     |
| SYDNEY        |           4 |    ->0.39<-   | OK     |
+---------------+-------------+---------------+--------+
7 rows in set (0.04 sec)
```

20/100

Parte 2

4. Crear una base de datos con nombre `base_p07` y en ella, generar una función llamada `fibonacci` tal que, al ingresarle un número natural, regrese el n-ésimo término de la sucesión de Fibonacci para esa n. Consideraciones:

- ✓ El parámetro y resultado deberán ser números.
- ✓ Para n = 0, regresará el valor 0.
- ✓ Devolverá `ERROR` en caso de que se intente ingresar un parámetro < 0.
- ✓ No podrán usar la *Fórmula de Binet para la sucesión de Fibonacci* en su código.

20/100

5. Escribir una consulta usando la función `fibonacci` que genere la siguiente tabla:

```
+-----------------+--------------------+
| Número natural  | Término Fibonacci  |
+-----------------+--------------------+
|               0 |                  0 |
|               1 |                  1 |
|               2 |                  1 |
|              15 |                610 |
|              46 |      1,836,311,903 |
+-----------------+--------------------+
5 rows in set (0.00 sec)
```

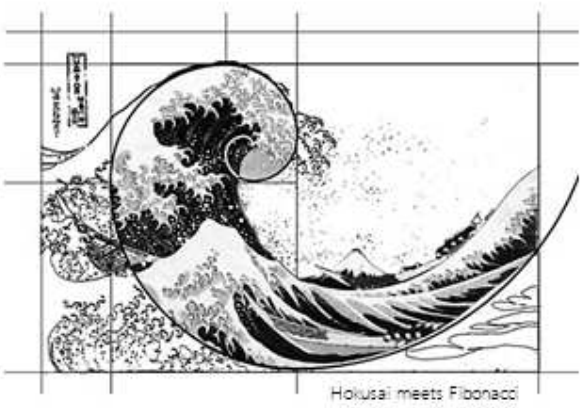
15/100

6. Realizar una consulta a `information_schema` que muestre las funciones existentes en la base de datos `base_p07` y los campos 'Base de datos', 'Nombre función' y 'Tipo de dato salida'.

15/100

Ejercicio extra

Dada la sucesión de Fibonacci, el límite de los cocientes de sus términos converge a un irracional conocido como el número áureo (usualmente denotado por φ).



En `base_p07`, diseñar una función llamada `fibonacci_lim` tal que, al ingresarle un número i, devuelva un enunciado indicando para que número natural n se cumple que $\varphi = \lim_{k\to\infty}\left(\frac{a_{k+1}}{a_k}\right) \approx \frac{a_{n+1}}{a_n}$ con una precisión de i decimales, acompañado de los valores de los cocientes.

Por ejemplo, para el parámetro i = 2, la función calcula que, en el natural n = 7, el cociente resulta 1.6152846150 que es exacto al límite φ = 1.6180339887 en dos decimales.

A considerar lo siguiente:

- ✓ El parámetro de ingreso tendrá que ser un número entero no negativo.
- ✓ El resultado de la función será texto.
- ✓ Devolverá `ERROR` en caso de que se intente ingresar un parámetro < 0.
- ✓ No podrán usar la *Fórmula de Binet para la sucesión de Fibonacci* en su código.

Al correr la siguiente línea, deberá obtenerse el resultado a continuación:

```
select
fibonacci_lim(0),
fibonacci_lim(1),
fibonacci_lim(2),
fibonacci_lim(3),
fibonacci_lim(8)\G
```

```
mysql> select fibonacci_lim(0),fibonacci_lim(1),fibonacci_lim(2),fibonacci_lim(3),fibonacci_lim(8)\G
*************************** 1. row ***************************
fibonacci_lim(0): Para n = 0 la sucesión an+1 ÷ an = 1.0000000000 aproxima al
 límite ϕ = 1.6180339887 con una precisión de 0 dígitos decimales.

fibonacci_lim(1): Para n = 4 la sucesión an+1 ÷ an = 1.6666666660 aproxima al
 límite ϕ = 1.6180339887 con una precisión de 1 dígito decimal.

fibonacci_lim(2): Para n = 7 la sucesión an+1 ÷ an = 1.6153846150 aproxima al
 límite ϕ = 1.6180339887 con una precisión de 2 dígitos decimales.

fibonacci_lim(3): Para n = 10 la sucesión an+1 ÷ an = 1.6181818180 aproxima al
 límite ϕ = 1.6180339887 con una precisión de 3 dígitos decimales.

fibonacci_lim(8): Para n = 21 la sucesión an+1 ÷ an = 1.6180339850 aproxima al
 límite ϕ = 1.6180339887 con una precisión de 8 dígitos decimales.

1 row in set (0.01 sec)
```