

Práctica 08

Se deberán adjuntar los archivos .sql, y .txt vía Moodle **antes** de las 23:00 del 24 de mayo de 2023. No se aceptarán entregas por otro medio o extemporáneas.

Objetivo

Generar un script .sql llamado pr08_eqNN, donde NN indique el número de su equipo. Por ejemplo, para el equipo 02 el nombre del script deberá ser *pr08_eq02.sql*

Por ningún motivo el script deberá generar errores, sin importar el número de veces que se corra. Puede asumir que las bases de datos que se mencionan explícitamente en los encabezados de las secciones se encuentran cargadas en el manejador de quien calificará, por lo que no deberá incluir el comando *source*, ni el código de las bases.

No se calificarán consultas:

- con valores estáticos
- con la cláusula *limit*

Ejercicio 0

1. Cambie el prompt de manera que muestre la fecha y hora actuales, y el nombre de la base de datos en uso, seguido de '> '. El formato y otros accesorios quedan a su elección.

Ejercicio 1 - Base de datos 'replicas'

1. Defina una función que calcule la diferencia en semanas que se presenta entre dos fechas dadas. Podrá llamar a su función como prefiera, pero ésta deberá actuar como lo hace `semanas()`:

```
+-----+-----+-----+
| semanas('2023/01/30','2023/05/16') | semanas('2023/05/26','2023/05/17') | semanas('1999/01/01',NULL) |
+-----+-----+-----+
| 15.10 | 1.30 | -1.00 |
+-----+-----+-----+
1 row in set, 5 warnings (0.00 sec)
```

Escriba la consulta que genera la tabla mostrada usando su nueva función.

2. Use la función creada en 1. y almacene en la vista `mis_valores`: el promedio, el mínimo y el máximo número de semanas que le toma a la tienda enviar un pedido. Sólo se deben tomar en cuenta las órdenes que tengan el estatus de enviadas.

10/100

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

- el número de empleados trabajando
- el tiempo promedio que les toma enviar una orden en semanas
- una alerta si se supera el tiempo promedio calculado sobre todas las órdenes

La tabla resultado deberá visualizarse exactamente como sigue:

Ciudad	# empleados	# semanas promedio	Aviso
TOKYO	2	->1.17<-	ALERTA
BOSTON	2	->0.58<-	ALERTA
NYC	2	->0.52<-	OK
PARIS	5	->0.49<-	OK
LONDON	2	->0.49<-	OK
SAN FRANCISCO	6	->0.49<-	OK
SYDNEY	4	->0.48<-	OK

7 rows in set (0.02 sec)

Hint: Puede crear y hacer uso de tantas vistas como sean necesarias para resolver este ejercicio.

20/100

4. Diseñe una función llamada `fibonacci` tal que, al darle un número natural, regrese el n-ésimo término de la sucesión de Fibonacci para esa n. Considérese que:

- Tanto el parámetro como el resultado de la función serán números enteros.
- Para $n = 0$, la función regresará al valor 0.
- Devolverá ERROR en caso de que se ingrese un parámetro < 0 .
- No podrá usar la *Fórmula de Binet para la sucesión de Fibonacci* en su código.

15/100

5. Escriba una sola consulta que use `fibonacci()` para generar lo siguiente:

Número natural	Término Fibonacci
0	0
1	1
2	1
15	610
46	1,836,311,903

5 rows in set (0.00 sec)

10/100

6. Cree una función que cuente el número de vocales que hay en una palabra y póngala en uso en una consulta que genere la siguiente tabla.

PALABRA	VOCALES
Sofía	3
PARAGÜAS	4
murciélago	5
Luis Agustín Joaquín	9
gatito hace 'MiAUaëaA'	12

5 rows in set (0.00 sec)

Hint: Explore qué son las expresiones regulares y cómo pueden usarse en MySQL.

20/100

7. Realice una consulta a `information_schema` que muestre:

- Base de datos
- Nombre función
- Tipo de dato salida

de todas las funciones existentes en `replicas` y en `sakila`.

10/100

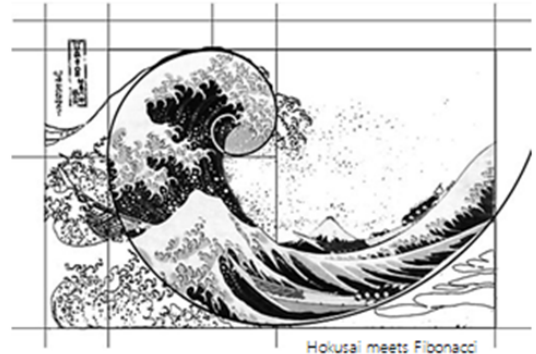
Ejercicio extra - Base de datos 'replicas'

1. 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).

Diseñe una función llamada `lim_fibonacci` tal que, al ingresarle un número `i`, devuelva un enunciado indicando para que número natural `n` se cumple que

$$\phi = \lim_{k \rightarrow \infty} \left(\frac{a_{k+1}}{a_k} \right) \approx \frac{a_{n+1}}{a_n}$$

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 $\phi = 1.6180339887$ en dos decimales.

Considere que:

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

Se espera el resultado a continuación:

```
SELECT
lim_fibonacci(0), lim_fibonacci(1), lim_fibonacci(2),
lim_fibonacci(3), lim_fibonacci(8), lim_fibonacci(9)\G
```

```
***** 1. row *****
lim_fibonacci(0): Para n = 1 la sucesión an+1 ÷ an = 1.0000000000 aproxima al
límite φ = 1.6180339887 con una precisión de 0 dígitos decimales.

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

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

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

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

lim_fibonacci(9): Para n = 23 la sucesión an+1 ÷ an = 1.6180339880 aproxima al
límite φ = 1.6180339887 con una precisión de 9 dígitos decimales.

1 row in set (0.01 sec)
```