

Práctica 05

Se deberán entregar los archivos .sql y .txt correspondientes vía Moodle **antes** de las 23:59 del 3 de noviembre de 2022. No se aceptarán entregas extemporáneas o por otro medio.

Objetivo

Generar un script tipo SQL llamado Practica05_EqNN (donde N indica el número de Equipo). Por ejemplo, para el equipo 06 el nombre del script será Practica05_Eq06.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 o en las que se hayan usado `joins`.

Parte 0 – Preliminares

1. Cambiar el prompt al siguiente formato:

```
[06:50 pm] - [mundo] ->
```

La hora en sistema 12 hr en formato hh:mm entre corchetes, luego un guión y a un lado, sin espacios, el nombre de la base de datos en uso entre corchetes. Al final, ->

¡Ojo! Los elementos que componen el `prompt` son variables.

Parte 1 – Base de datos ‘mundo’

2. ¿Cuáles son los países de Europa tales que su gdp supera el promedio de gdp de los países en América (Norte, Sur y Caribe) y su área cumple ser menor que tres veces la de Cuba? Ordenar el resultado alfabéticamente.

15/100

3. Enlistar el nombre de los países y continente al que pertenecen; siempre y cuando sean países que se encuentren en el mismo continente que Argentina o Australia; y cuyo nombre tenga más de dos palabras. Ordenar el resultado alfabéticamente por continente y país.

15/100

Parte 2 – Base de datos ‘sakila’

4. Encontrar el id y el nombre de las categorías que tengan entre 65 y 75 películas registradas en ellas.

15/100

5. Listar el año y nombre de los meses en los que un pago realizado por una renta superó el mínimo pago, pero quedó por debajo del pago promedio en ese año, para los años 2005 y 2006. La siguiente es la tabla objetivo:

```
+-----+-----+
| anio | meses pago |
+-----+-----+
| 2005 | May        |
| 2005 | June       |
| 2005 | July       |
| 2005 | August     |
| 2006 | February   |
+-----+-----+
5 rows in set (0.06 sec)
```

15/100

6. Seleccionar el código postal, distrito y id de ciudad de los códigos postales de longitud 5 para los que haya registrados más de una dirección, y que cumplan que el primer dígito del id de su ciudad sea par. Este es el resultado esperado:

```
+-----+-----+-----+
| postal_code | district      | city_id |
+-----+-----+-----+
| 22474       | Uttar Pradesh | 448     |
| 52137       | Hiroshima     | 205     |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

15/100

7. Mostrar el nombre completo (nombre + apellido) y el correo de los clientes que han rentado una película antes del 2006 y no la hayan regresado aún.

15/100

8. Hallar el id de los clientes tales que el máximo monto de renta que le han pagado a la tienda esté por encima de los 10 dólares y que su última renta se haya hecho el día de San Valentín del año 2006.

cliente	max pago	ultima renta
21	11.0	14-February-2006
29	11.0	14-February-2006
33	11.0	14-February-2006
87	11.0	14-February-2006
114	11.0	14-February-2006
163	11.0	14-February-2006
168	11.0	14-February-2006
178	11.0	14-February-2006
200	11.0	14-February-2006
245	11.0	14-February-2006
300	11.0	14-February-2006
336	11.0	14-February-2006
388	11.0	14-February-2006
410	11.0	14-February-2006
441	11.0	14-February-2006
448	11.0	14-February-2006
457	11.0	14-February-2006
516	11.0	14-February-2006
550	11.0	14-February-2006
561	11.0	14-February-2006
592	12.0	14-February-2006

21 rows in set (0.06 sec)

Ejercicios extra

1. Usando la base de datos 'pixup' disponible en 16_pixupDatos.sql, diseñar una sola consulta que, al ejecutarse, genere una tabla de cuatro columnas:

- `disquera` es el nombre de la disqueras cuya longitud supera al promedio de longitud de nombre más una desviación estándar
- `num_caracteres` es la longitud de bytes del nombre de la disquera
- `num_random` es un número aleatorio seleccionado del conjunto $\{1, 2, \dots, \text{num_caracteres}\}$
- `resultado` es el nombre de la disquera modificado tal que en la posición correspondiente al `num_random`, se encuentre un signo de admiración de cierre (!)

Fijar semilla $N = 3$ en todas las generaciones aleatorias.

Por ejemplo, si el nombre de una disquera que cumpliera los requisitos de filtrado fuera 'Sofia Pejcovich' y el número (pseudo)aleatorio generado hubiese sido 14, la tupla correspondiente a ese registro se vería así:

```
+-----+-----+-----+-----+
| disquera      | num_caracteres | num_random | resultado      |
+-----+-----+-----+-----+
| Sofia Pejcovich |          15    |          14 | Sofia Pejcovi!h |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

10/100

2. Usando la base de datos 'world_x' disponible en base_world_x.sql diseñar una sola consulta que, al ejecutarse, genere una tabla de tres columnas:

- `pais` contendrá los nombres de los países tales que la segunda letra de su código sea la J o que su `bandera` sea 1
- `bandera` es una columna que podrá llevar los valores 0 o 1. Será 0 si el país tiene registrada su capital y 1 si ese registro es nulo
- `num_idiomas` es un número de idiomas que se hablan en los países que cumplan las condiciones buscadas

Ordenar por `num_idiomas` de manera descendente.

(🚫) ¡No pueden usar `case-when` en su query!

10/100