

## Pautas

Lea **atentamente** las pautas para la resolución de los ejercicios planteados y la entrega del trabajo práctico.

- (a) Este TP **debe resolverse individualmente**.
- (b) Genere los archivos `apellido_nombre_batallaDeGallos_DDL.sql` y `apellido_nombre_batallaDeGallos_DML.sql`, reemplazando las palabras *apellido\_nombre* por los correspondientes al alumno.  
Ejemplo: `arevalo_gabriela_ferías_DDL.sql`.
- (c) Guarde las consultas pedidas en el archivo correspondiente.
- (d) Guarde las descripciones o aclaraciones en el archivo que corresponda utilizando comentarios de sql. Parte del proceso de corrección se realiza mediante parseo automático, es importante el uso de comentarios en todo lo que no sean sentencias sql.
- (e) Escriba todas las consultas de forma prolija. Es importante que entre cada consulta haya espacios de líneas suficientes. Termine cada consulta con ; (punto y coma).
- (f) Es importante que cada consulta comience en una línea aparte. Esto disminuye la cantidad de errores y facilita la corrección.
- (g) Esta es una instancia de evaluación, aclare todo lo que considere importante, aún lo que considere trivial.
- (h) Suba al aula virtual ambos archivos comprimidos en un archivo `apellido_nombre_ferías.zip` con la resolución del trabajo.
- (i) Puede enviar consultas al `tpi-doc-bd@listas.unq.edu.ar` escribiendo en el Asunto del email `[UNQ - BD] Consulta TP Ejercicio(s) XX` identificando XX con el/los número(s) de ejercicio(s) sobre el/los cuales quiere hacer preguntas.
- (j) La fecha límite de entrega es el **25 de Junio de 2021 a las 20:00** inclusive. Si termina el TP antes de la fecha límite, puede subirlo al aula antes.

# 1. Resumen de SQL

La sintaxis del DML de SQL puede resumirse de la siguiente manera:

```
SELECT [ALL | DISTINCT] <atributos>
FROM <tablas>
[WHERE <expresion condicional>]
[GROUP BY <atributo(s)>]
[HAVING <expresion condicional>]
[ORDER BY <columna(s)>]
```

donde:

- <atributos> es la información a obtener de la base de datos.
- FROM <tablas> especifica de qué tablas se obtiene la información buscada.
- WHERE <expresion condicional> expresa una condición que deben cumplir las filas de la consulta resultante.
- GROUP BY <atributo(s)> permite formar consultas agrupadas para extraer información global sobre los grupos formados.
- HAVING <expresion condicional> es condición sobre los grupos formados.
- ORDER BY <columna(s)> ordena por una o varias columnas.
- DISTINCT: No permite la aparición de filas idénticas.
- <expresion condicional>: formada por un conjunto de predicados combinados mediante los operadores lógicos AND, OR y NOT.
- Los predicados utilizados permiten comparar columnas:
  - predicados de comparación: =, <>, >, <, >=, <=.
  - predicado BETWEEN: permite comprobar si un escalar está en un rango.
  - predicado IN: permite comprobar si el valor está dentro de un conjunto.
  - predicado IS NULL: permite comprobar si el valor es nulo.

# 2. Estilo requerido para el código SQL

## 1. Uso de mayúsculas y minúsculas

- a) Palabras reservadas del lenguaje (select, on , where, etc.): MAYÚSCULAS
- b) Nombres de tablas: minúsculas y singular
- c) Nombres de atributos: minúsculas

## 2. Organización del código

- a) Un renglón para todo lo relativo al SELECT
- b) Un renglón para el FROM
- c) Un renglón para cada tabla joineada
- d) Un renglón para el WHERE
- e) Un renglón para cada <expresion condicional> del WHERE

## Ejercicio 1: DDL

Se nos solicitó desarrollar una base de datos para almacenar las rimas realizadas en competencias de batallas de gallos. Una batalla de gallos consiste en dos competidores enfrentados dentro de una competición diciéndose rimas en varias rondas.

Cada ronda está representada por una temática en particular dentro de esa competición. En este modelo solo queremos almacenar las rimas que fueron dichas por cada competidor. Un competidor puede decir varias rimas bajo la misma temática en competición, pero solo una por patrón. Por ejemplo, el competidor *Dani* hace rimas en la competición FMS JORNADA 1, bajo la temática RIMAS CON OBJETOS y bajo dicha temática realizó 6 rimas, es decir, hizo 6 patrones distintos. Estas rimas tienen un valor dado por un jurado, el cual también se almacena en la base de datos.

Cada competidor se registra en el sistema por la plaza que lo representa, que a su vez fue en la primera donde comenzó en el mundo del freestyle(rapear). Esto lo tomamos como su lugar de origen y cada competidor es de una plaza en particular, pero a una plaza pertenecen varios competidores.

La estructura de la base de datos se especifica a continuación:

```
plaza (
    id serial PK,
    nombre varchar(30) NOT NULL,
    ciudad varchar(50) NOT NULL,
    provincia varchar(50) NOT NULL
)

competicion (
    id serial PK,
    nombre varchar(50) NOT NULL,
    fecha timestamp NOT NULL,
    hora integer NOT NULL,
    ciudad varchar(50) NOT NULL,
    predio varchar(50) NOT NULL,
    provincia varchar(30) NOT NULL
)

competidor (
    id serial PK,
    sobrenombre varchar(30) NOT NULL,
    especialidad varchar(20) NOT NULL,
    es_mayor boolean NOT NULL,
    plaza_id integer NOT NULL
)

tematica (
    id serial PK,
    duracion_en_segundos integer NOT NULL,
    nombre varchar(50) NOT NULL,
    descripcion varchar(255) NOT NULL
)

tematica_en_competicion (
    beat_autor varchar(50) NOT NULL,
    beat_nombre varchar(50) NOT NULL,
    competicion_id integer NOT NULL,
    tematica_id integer NOT NULL,
)
```

```

rima (
    valoracion int,
    patron integer PK,
    competidor_id integer PK NOT NULL,
    competicion_id integer PK NOT NULL,
    tematica_id integer PK NOT NULL
)

```

- (a) Genere una base de datos en el motor PostgreSQL cuyo nombre sea `tp_su_apellido`. No desaprobe por literalidad. Describa los pasos que tuvo que llevar a cabo para lograrlo. Guarde las sentencias que usó para la creación de las tablas en el archivo `sql`.
- (b) Escriba las queries para crear las tablas y estructuras de acuerdo a lo descripto más arriba. El puntaje de las rimas por defecto debe ser 2 y la duración de la temática debe ser por default de 30 segundos.
- (c) Identifique todas las claves foráneas que correspondan y escriba las queries para crearlas.
- (d) Ejecute las queries de modo tal que todas estas estructuras sean creadas en la base de datos creada en el punto a.
- (e) Inserte en la base los datos brindados en el archivo `datos_batalla_de_gallos.sql`. Describa los pasos que tuvo que llevar a cabo para lograrlo, qué método usó.
- (f) Limite el atributo `patron` a un máximo de 4 y el atributo `valoracion` a un máximo de 10 en la tabla `rima`.
- (g) Realice la inserción de datos de forma tal que exista una rima de la temática 4x4 en una competición hecha en el predio del Luna Park por un competidor de sobrenombre `Wolf` oriundo de una plaza de Bernal, y que dicha rima esté valorada con 7. Dichas queries deben estar en el entregable.

## Ejercicio 2: Consultas

Genere las siguientes *queries* SQL:

1. Obtener el sobrenombre, id y el nombre de la plaza, de los competidores de plazas de la ciudad de Quilmes en Buenos Aires.
2. Obtener el sobrenombre y el nombre de su plaza de los competidores que solo asistieron a una competencia.
3. Obtener las competencias donde todas sus temáticas duran más de 2 minutos.
4. Obtener el nombre y el promedio de duración de las temáticas utilizadas en las competencias realizadas en los predios llamados Colonial de Buenos Aires y de Córdoba.
5. Obtener los competidores registrados pero que aún no hayan competido.
6. Obtener la mejor rima de cada competidor que no sea del mismo lugar en donde se realizó la competición, ordenadas por valoración.
7. Obtener la temática en competición de las rimas cuyos promedio en competición no superen el valor 6.4 .
8. Obtener el autor del beat, el nombre del beat y el nombre de la competicion de aquellas competencias en donde el competidor no haya hecho rimas con valoración menor a 2.
9. Obtener una lista que muestre la cantidad de plazas por zona, ordenados descendientemente por la cantidad pero ascendientemente por provincia y ciudad. Una zona se define por su ciudad y provincia. Se debe visualizar la zona en el resultado.
10. Obtener una lista que muestre la cantidad de competidores por zona, ordenados descendientemente por la cantidad y zona. Se debe contar a los competidores que hicieron al menos una rima. Se debe visualizar la zona en el resultado.

11. Obtener un listado que muestre, de cada plaza, el promedio de valoración de las rimas, la máxima valoración y la mínima de los competidores de esa plaza.
12. Obtener de las últimas 10 rimas registradas, el sobrenombre del competidor, la valoración de la rima, el nombre del beat, el nombre del autor del beat y el nombre de la plaza del competidor.
13. En la tabla de competidor conocemos su PK, pero es necesario impedir que pueda repetirse el sobrenombre entre distintos competidores. Explique cómo lo haría e impleméntelo.
14. Cree una vista(view) de los competidores cuyo promedio histórico de rimas en competencias de Buenos Aires es mayor a 7, participaron en más de 3 competencias, tienen al menos 4 rimas con puntaje perfecto(10) y compitieron antes del 2015 o después del 2020.
15. Cree un índice de las competencias por nombre y fecha para mejorar la velocidad de las consultas.
16. Obtener los nombres de los beats y su autor en donde el sobrenombre de un competidor que haya hecho rimas en ese beat y el nombre del autor del beat sean el mismo.
17. Obtener la competición y el competidor campeón de la misma. Un competidor es campeón al ser el que más puntos obtuvo en una competición.