

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_spotify_DDL.sql` y `apellido_nombre_spotify_DML.sql`, reemplazando las palabras *apellido_nombre* por los correspondientes al alumno.
Ejemplo: `arevalo_gabriela_spotify_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) Envíe ambos archivos comprimidos en un archivo `apellido_nombre_spotify.zip` con la resolución del trabajo a `tpi-doc-bd@listas.unq.edu.ar` escribiendo en el Asunto del email `[UNQ - BD] <apellido_nombre> - Entrega TP SQL`, por ejemplo `[UNQ - BD] arevalo_gabriela - Entrega TP SQL`.
- (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 **4 de diciembre de 2019** inclusive. Si termina el TP antes de la fecha límite, puede enviarlo también.

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

artista < id , nombre, nombre_real, sigue_activo, pais_origen, anio_nacimiento >

Esta tabla registra a los artistas. Se los identifica por un id entero autoincremental, tanto el nombre como el nombre_real y el país son varchar de largo 50, sigue_activo es un booleano cuyo valor por defecto es true, y el anio_nacimiento es un entero. Ninguna de estas columnas puede registrar null. El nombre, si bien no es PK, no puede repetirse

usuario < nombre_completo, username , password, anio_inicio, pais_usuario >

Los usuarios se identifican por el username que no puede repetirse, nyap, username y pais_usuario son varchar(50) y el anio_inicio un entero.

cancion < nombre, artista , duracion, descargas_actuales, anio_creacion >

Las canciones se identifican por su nombre y el artista que las registró, dicho nombre tambien es varchar(50). Duración, descargas y anio_creacion son enteros.

en_lista < nombre_usuario , nombre_cancion, artista , anio_agregacion >

Un usuario arma listas de temas que le gusta escuchar.

sigue < artista , nombre_usuario , anio_agregacion >

En esta tabla registramos a los artistas favoritos, aquellos a los que el usuario decide seguir.

- (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) Cree las tablas y estructuras de acuerdo a lo descripto más arriba.
- (c) Inserte en la base los datos brindados en el archivo spotify-data.sql. Describa los pasos que tuvo que llevar a cabo para lograrlo, que metodo usó.
- (d) Agregue el usuario Ines Cosa, username niñaíta, muerte como palabra clave y sabemos que nació en Italia.

Ejercicio 2: Consultas

Genere las siguientes queries SQL:

1. Obtenga los nombres reales de los artistas junto con los nombres y apellido de los usuarios que agregaron alguna canción de ese artista a partir del año 2012.
2. Obtenga el nombre real y la edad de los artistas narcisistas que siguen activos. Se entiende por narcisista a aquellos artistas que poseen una canción nombrada igual que ellos la cual cuenta con más de diez mil descargas. Para calcular la edad puede generar una operación entre el año actual y el año de nacimiento.
3. Debido a un error en su creación, la tabla SIGUE podría poseer tuplas erróneas que indiquen que un usuario comenzó a seguir a un artista en un año anterior a que el usuario haya iniciado su actividad en el sistema. Genere una consulta que muestre el nombre del usuario, el año de inicio del usuario en el sistema y la cantidad de artistas que comenzó a seguir “años anteriores” a su inicio en el sistema. Ordene los resultados por cantidad de artistas, siendo que la primer fila tiene la mayor cantidad.
4. Seleccione el nombre de los artistas argentinos y la suma de descargas de sus canciones (ordenados por cantidad) los cuales sumen con todas sus canciones más descargas que el promedio de descargas de todas las canciones de todos los artistas.

5. Los administradores de la base quieren conocer la distribución de canciones según país de usuario y país de origen del artista de la canción. Obtenga la cantidad de canciones que tienen en sus listas los usuarios agrupando por país de origen del usuario y país de origen del artista de la canción. Ordene los resultados de forma descendente por cantidad de canciones y quédese con las 10 tuplas siguientes a las primeras 5. O sea, no deben mostrarse las tuplas 1 a 5.
6. Obtenga, ordenados alfabéticamente, el nombre y apellido de los usuarios que sólo tienen en sus listas canciones que fueron creadas a partir del año 2000.
7. Se quiere conocer la cantidad de artistas por país que no hayan creado canciones a partir del año 2015. Mostrar el nombre del país y la cantidad, ordenado por mayor cantidad.
8. Obtener el nombre de usuario, el año de inicio y el país de los usuarios que en sus listas no tienen ninguna canción de artistas argentinos.
9. Obtener el promedio de descargas de canciones, agrupadas por país de origen del artista. Los resultados deberán estar ordenados del país con mayor promedio al menor.
10. Obtenga los nombres reales de los artistas junto con los nombres y apellidos de los usuarios que comenzaron a seguir a ese artista antes del año 2008 (inclusive).
11. Obtenga el nombre real y la edad de los artistas inactivos que tengan por lo menos una canción que haya tenido menos de diez millones de descargas. Para calcular la edad puede generar una operación entre el año actual y el año de nacimiento.
12. Seleccione el nombre de los artistas de América del Norte y el promedio de descargas de sus canciones tal que ese promedio sea menor que el promedio de descarga de todos los artistas. Ordene los resultados por mayor promedio.
13. Obtenga, ordenados alfabéticamente, el nombre y apellido de los usuarios que tienen en sus listas canciones de "Calle 13" pero no "Alejandro Sanz".
14. Obtenga, ordenados alfabéticamente, el nombre y apellido de los usuarios que sólo siguen a artistas argentinos.
15. Se quiere conocer la cantidad de canciones por artistas que no hayan sido agregadas a listas por ningún usuario. Mostrar el nombre del artista y la cantidad, ordenado por mayor cantidad.
16. Obtener el nombre de usuario, el año de inicio y el país de los usuarios que no siguen a ningún artista inactivo.
17. Obtener el promedio de descargas de canciones, agrupadas por año de nacimiento del artista. Los resultados deberán estar ordenados del año de mayor promedio al menor.
18. Obtenga los nombres reales de los artistas junto con los nombres y apellido de los usuarios que siguen a artistas uruguayos o brasileños.
19. Obtenga el nombre real y la edad de los artistas activos que tengan por lo menos una canción que dure entre 100 y 300 segundos. Para calcular la edad puede generar una operación entre el año actual y el año de nacimiento. Quédese con las primeras 10 tuplas siguientes a las primeras 5. O sea, no deben mostrarse las tuplas 1 a 5.
20. Seleccione el nombre de los artistas alemanes y el promedio de duración de sus canciones tal que esa duración sea mayor que el promedio de duración de canciones de todos los artistas. Ordene los resultados por mayor promedio.
21. Los administradores de la base quieren conocer la distribución de artistas que son seguidos en cada país. Obtenga la cantidad de artistas activos que siguen los usuarios de los países fundadores del Mercosur, agrupando por país. Ordene los resultados de forma descendente por cantidad de artistas.
22. Obtenga, ordenados alfabéticamente, el nombre real de los artistas que son seguidos por usuarios japoneses o que sus canciones estén en listas de usuarios argentinos pero no hayan nacido en Argentina.
23. Obtenga, ordenados alfabéticamente, el nombre y apellido de los usuarios que sólo siguen artistas activos.
24. Se quiere conocer la cantidad de artistas por país que no hayan sido seguidos por ningún usuario. Mostrar el nombre del país y la cantidad, ordenado por mayor cantidad.

25. Obtener el nombre de usuario, el año de inicio y el país de los usuarios que no tienen en sus listas canciones creadas antes del año 2000.
26. Obtener el promedio de duración de las canciones, agrupadas por año de nacimiento del artista. Los resultados deberán estar ordenados de la duración promedio mayor a menor.

Fixes:

1. v01. se agrega atributo *anio_agregacion* a la tabla *sigue*.
2. v01. se modifica atributo *nyap* por *nombre_completo* de la tabla *usuario*