

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_ferías_DDL.sql` y `apellido_nombre_ferías_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 **13 de Julio de 2020** 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 joinada
- 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 que permita registrar las ferias que se desarrollan en el país y poder hacer un seguimiento de los precios de los productos que se comercializan en ellas.

Para eso, diseñamos una base de datos que registre las ferias, los productos que se comercializan, los usuarios que declaran los precios que ven en las ferias.

Cada usuario, cuando visita la feria y decide registrar los precios que se ofertan, hace una declaración, y por cada producto que declara el precio, nos hace una declaración individual donde se registra el precio por bulto (en estas ferias, se venden los productos en diferentes presentaciones) y los kilos que tiene dicho bulto.

De esta forma podemos hacer un trabajo estadístico sin importar la forma de la presentación o su tamaño.

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

```
feria (  
  id int PK,  
  nombre varchar(255) NOT NULL,  
  cuit varchar(13) NOT NULL,  
  cantidad_puestos int(11) DEFAULT NULL,  
  localidad varchar(255) DEFAULT NULL,  
  domicilio varchar(255) COLLATE DEFAULT NULL,  
  zona varchar(255) )
```

```
producto (  
  id PK,  
  tipo_id integer NOT NULL,  
  especie varchar(255) NOT NULL,  
  variedad varchar(255) DEFAULT NULL,  
  activo bool NOT NULL  
)
```

```
producto_tipo (  
  id PK,  
  nombre varchar(255) NOT NULL,  
  descripcion varchar(255) DEFAULT NULL  
)
```

```
declaracion (  
  id PK,  
  fechageneracion datetime NOT NULL,  
  feria_id integer NOT NULL,  
  user_autor_id integer DEFAULT NULL  
)
```

```
declaracion_individual (  
  id PK,  
  producto_declarado_id integer NOT NULL,  
  declaracion_id integer NOT NULL,  
  fecha date NOT NULL,  
  precio_por_bulto decimal(12,2) DEFAULT NULL,  
  comercializado bool NOT NULL DEFAULT 1,  
  peso_por_bulto decimal(5,2) DEFAULT NULL  
)
```

```
user (  
  id PK,  
  email varchar(180) NOT NULL,  
  password varchar(25) NOT NULL,  
  nombre varchar(255) DEFAULT NULL,
```

```

    apellido varchar(255) DEFAULT NULL
)

user_feria (
    user_id integer NOT NULL,
    feria_id integer NOT NULL
)

```

- (a) Genere una base de datos en el motor PostgreSQL cuyo nombre sea `tp_su_apellido`. No desapruebe 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.
- (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_ferias.sql`. Describa los pasos que tuvo que llevar a cabo para lograrlo, qué método usó.
- (f) Agregue el usuario Ines Cosa, username `niñita`, mail `ines(at)cosa(dot)com`, *suerte* como palabra clave.

Ejercicio 2: Consultas

Genere las siguientes *queries SQL*:

1. Obtenga el nombre y apellido de los usuarios que no están relacionados con ninguna feria ordenados por nombre.
2. Obtenga el nombre y apellido de los usuarios que están relacionados con SOLAMENTE 1 feria ordenados por apellido.
3. Obtenga el nombre y apellido de los usuarios que no están relacionados con más de una feria.
4. Obtenga el precio por kilo promediado por mes de cada producto, ordenados por tipo de producto ascendente, por especie y variedad del mismo y por precio por kilo descendente.
5. Seleccione qué ferias están registradas pero no tienen ninguna declaración.
6. Seleccione el nombre, apellido y correo electrónico de los usuarios que hicieron declaraciones de ferias con las que no están relacionados.
7. Seleccione aquellas frutas cuyo precio promedio por kilo histórico no supere los 50 pesos.
8. Obtenga, ordenados alfabéticamente, el nombre y apellido de los usuarios que sólo frutas tienen en sus declaraciones (de acuerdo al tipo de producto).
9. Obtenga un listado que muestre la cantidad de ferias por zona, ordenados descendientemente por cantidad.
10. Obtenga un listado que muestre la cantidad de ferias por zona, ordenados descendientemente (el listado debe excluir a las ferias sin declaraciones).
11. Obtenga un listado que muestre la cantidad de ferias por zona, ordenados descendientemente (el listado debe incluir a las ferias sin declaraciones).
12. Obtenga un listado que muestre, de cada localidad donde haya usuarios registrados, el promedio de kilos por bulto, el máximo de kilos por bulto y el mínimo de kilos por bulto de naranjas ofrecidos en ferias de ese distrito.
13. En la tabla de productos conocemos su PK, pero es necesario impedir que pueda repetirse especie y variedad. Explique cómo lo haría e impleméntelo.

14. Cree una vista (view) con la información de correo del usuario, nombre, ubicación de todas las ferias con las que está relacionado. Dicho listado debe incluir a los usuarios que no tienen ferias asociadas.
15. Obtenga un listado con el precio promedio, precios máximos y mínimo por producto en la semana actual.
16. Obtenga el precio promedio por producto y por zona en la semana anterior a la actual.
17. Con el uso del sistema se identificaron muchísimas consultas buscando productos por su especie y variedad en la condición, cree un índice adecuado para dicha búsqueda.
18. Obtenga las 3 ferias con más usuarios que no hayan hecho declaraciones o que sólo las hayan hecho en ferias con menos de 50 puestos.