

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_shopping_DDL.sql` y `apellido_nombre_shopping_DML.sql`, reemplazando las palabras *apellido_nombre* por los correspondientes al alumno.
Ejemplo: `arevalo_gabriela_shopping_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_shopping.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 **23 de Noviembre 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 joineada
- d) Un renglón para el WHERE
- e) Un renglón para cada <expresion condicional> del WHERE

Ejercicio 1: DDL

Estamos transcurriendo el último trimestre del 2020, y Argentina se encuentra normalizando sus actividades comerciales. El shopping Alto UNQ nos pide diseñar una base de datos que les permita organizar su funcionamiento. En nuestra base de datos, es necesario registrar las ventas en el shopping, clientes, empleados y su información médica, las tiendas (o comercios) y por supuesto los productos que se comercializan.

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

```
cliente (
    id SERIAL PK
    nombre VARCHAR(20)
    fecha_nac DATE
    telefono INT
    dni INT
)

empleado (
    id SERIAL PK
    nombre VARCHAR(20)
    telefono INT
    direccion VARCHAR(20)
    id_tienda (FOREIGN KEY TIENDA.ID)
    rol VARCHAR(20)
    inf_medica (FOREIGN KEY INF_MEDICA.ID)
)

tienda (
    id SERIAL PK
    nombre VARCHAR(20)
    local INT
    encargado (FOREIGN KEY EMPLEADO.ID)
)

informacion_medica (
    id SERIAL PK
    grupo_sanguineo VARCHAR(5)
    alergias VARCHAR(200)
    obra_social VARCHAR(20)
)

producto (
    id SERIAL PK
    tienda (FOREIGN KEY TIENDA.ID)
    tipo VARCHAR(20)
    nombre VARCHAR(20)
    precio INT
)

ventas (
    cliente (FOREIGN KEY cliente.ID) PK
    producto (FOREIGN KEY PRODUCTO.ID) PK
    tienda (FOREIGN KEY TIENDA.ID) PK
    fecha DATETIME
    descuentos INT
)
```

- (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.
- (c) Identificamos que en la tabla informacion_medica es necesario agregar un contacto de emergencia que es un texto que permita registrar un correo electronico a quien informar en caso de algun inconveniente con un empleado. Agregue el siguiente campo a la tabla mencionada:

```
contacto_emergencia VARCHAR(20)
```

- (d) Es importante agregar más informacion de cada local del shopping, debe crear la tabla:

```
locales (  
    id SERIAL  
    sector varchar  
    tamaño decimal  
)
```

- (e) Inserte de 5 a 10 tuplas en cada tabla.

Ejercicio 2: Consultas

Genere las siguientes *queries* SQL:

1. Por la necesidad de abrir el shopping no se cargaron los datos de contacto de algunos empleados de forma adecuada, y en las tablas los campos se cargaron con NULL. Se quiere remediar este problema lo mas rápido posible. Busque los id, nombre y teléfono de los empleados cuya contacto no está cargado o tiene el correo cargar@despues.com para así poder contactarlos.
2. Con el objetivo de ofrecerles ofertas y descuentos, se quiere conseguir la información de contacto de aquellos clientes que son mayores de 50 o compraron productos de tipo 'Tercera edad'.
3. Se quiere ver la cantidad de ventas hechas este año de producto de tipo 'Perfume' y el precio promedio que se dio entre todas las tiendas
4. Se requieren obtener aquellos empleados que tengan como grupo sanguíneo O- ordenados por apellido y por tienda de forma ascendente
5. Seleccione aquellas camisas (productos de tipo camisa) cuyo precio superen los \$ 800 que no hayan sido vendidas en el mes de octubre 2020.
6. Seleccione aquellas tiendas que hayan realizado un descuento de 50 % sobre sandalias Él pie frescoó 'La separados' y que ademas haya hecho descuentos en cualquier zapatilla
7. Seleccione aquellas tiendas que vendan trajes, pero no corbatas.
8. Obtener las 3 tiendas que tengan los productos más caros.
9. Obtener el promedio de precio del perfume Áromas del arrabal' de acuerdo al tamaño del local
10. Crear una vista que muestre las 5 tiendas que hayan realizado la mayor cantidad de ventas y sumatoria de pesos más alta en el mes de septiembre de 2020 ordenada por ambas
11. Identificar a los clientes que hayan gastado mas de \$ 100.000 mensuales (nombre, dni y teléfono y la sumatoria de compras aplicados los descuentos)
12. Obtener los empleados que trabajan en alguna de las 5 tiendas más exitosas
13. Es necesario identificar a los empleados convivientes, es decir que viven en la misma direccion
14. A partir del DNI de un cliente se busca obtener todos los puntos ha acumulado. Los puntos se calculan de la siguiente forma: se suma un punto por cada venta si ésta supera un límite de \$ 500 el puntaje el válido solamente en el año calendario.
15. Por cuestiones CoVID-19, debemos identificar las tiendas cuyo número de local sea par para que pueda abrir los días miércoles, viernes y domingo, y las impares martes, jueves y sábado. Los días lunes el shopping no abrirá. La consulta debe calcular los días de apertura que se asignan como texto, y ordenarla por fecha de apertura. La leyenda debe ser 'miercoles, viernes y domingo', y 'martes, jueves y sábado' y la columna se llamará dias_de_actividad.