

Actividad | 3 | Consultas

Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez Vega

ALUMNO: Christopher Dávila Rojas

FECHA: 11/09/2024

Índice

- 1. Introducción**
- 2. Descripción**
- 3. Justificación**
- 4. Desarrollo**
 - 4.1 Contar los registros existentes de una tabla**
 - 4.2 Verificar si existen o no registros duplicados**
 - 4.3 Ordenar de mayor a menor una tabla**
 - 4.4 Seleccionar sólo el campo ID_cliente y el campo ID_producto**
 - 4.5 Ordenar por orden alfabético el campo Nombre_cliente**
 - 4.6 Mostrar sólo el campo Fecha_venta que tenga registros del año 2022**
 - 4.7 Ver los datos de una tabla**
 - 4.8 Crear un procedimiento almacenado que muestre los clientes que tengan por nombre “Raúl”**
 - 4.9 Crear un ciclo donde, si existe el ID_producto número “3”, deberá mostrar el nombre del mismo (Nombre_producto correspondiente)**
- 5. Conclusión**
- 6. Referencias**

Introducción

Las bases de datos relacionales son un recurso indispensable para el almacenamiento y la organización de grandes volúmenes de información. Para interactuar con estas bases de datos y obtener datos precisos de manera eficiente, se utiliza el lenguaje SQL (Structured Query Language).

Este lenguaje permite realizar consultas que extraen, filtran y manipulan la información contenida en las bases de datos de acuerdo con los criterios que se necesiten.

A lo largo de esta actividad, se explorarán distintas consultas aplicables en PostgreSQL o SQL Server, tales como el conteo de registros, la verificación de duplicados, la ordenación de datos y la creación de procedimientos almacenados.

Este ejercicio tiene como finalidad introducirme como estudiante en la formulación y ejecución de consultas SQL, proporcionando ejemplos prácticos que abordan situaciones comunes en la administración de bases de datos.

A medida que las empresas dependen cada vez más de los sistemas de información para su toma de decisiones, la capacidad de ejecutar consultas correctamente se vuelve una habilidad fundamental en el perfil de los profesionales en desarrollo de software, administración de sistemas y análisis de datos.

Descripción

La actividad se centra en el uso de SQL como herramienta principal para interactuar con bases de datos. SQL es un lenguaje estándar que permite realizar diversas operaciones sobre los datos, como consultar información, modificar registros y generar reportes.

En particular, las consultas solicitadas en esta actividad incluyen operaciones para contar los registros presentes en una tabla, identificar registros duplicados, ordenar los datos según diferentes criterios y seleccionar campos específicos de una tabla. Asimismo, se requerirá la creación de procedimientos almacenados y ciclos, los cuales son herramientas avanzadas que permiten automatizar procesos y realizar acciones complejas dentro de la base de datos.

La importancia de estas operaciones radica en que proporcionan un control total sobre la información almacenada. Por ejemplo, contar registros es una tarea esencial cuando se necesita verificar el tamaño de una tabla o validar que ciertos datos hayan sido correctamente ingresados.

La verificación de duplicados, por su parte, es clave para mantener la integridad de la base de datos, ya que la existencia de datos repetidos puede causar inconsistencias o errores en los reportes. Ordenar los datos, ya sea numéricamente o alfabéticamente, es útil cuando se necesita presentar la información de manera clara y estructurada.

Justificación

El uso de SQL como herramienta de consulta y manipulación de datos está ampliamente justificado en la actualidad, ya que las bases de datos relacionales son el sistema más comúnmente utilizado para la gestión de grandes volúmenes de información.

La capacidad de extraer, ordenar y procesar datos de manera eficiente permite optimizar los procesos internos de las organizaciones, asegurando que la información esté disponible cuando se necesite y en el formato adecuado.

SQL es un lenguaje robusto y flexible que permite realizar tanto consultas simples como operaciones complejas dentro de una base de datos. En este sentido, una de las mayores ventajas de emplear SQL es su capacidad para trabajar con datos estructurados de manera eficiente y segura.

Por ejemplo, la capacidad de identificar registros duplicados es fundamental para garantizar que los datos sean únicos y consistentes, lo que es clave en sectores como el financiero, donde la precisión de los datos es crítica. Asimismo, la creación de procedimientos almacenados permite automatizar tareas recurrentes, lo que reduce el margen de error humano y optimiza el tiempo de los administradores de bases de datos.

Desarrollo

Para el desarrollo de esta actividad, se solicitan los siguientes puntos a cubrir:

- Contar los registros existentes de una tabla
- Verificar si existen o no registros duplicados
- Ordenar de mayor a menor una tabla
- Seleccionar sólo el campo ID_cliente y el campo ID_producto
- Ordenar por orden alfabético el campo Nombre_cliente
- Mostrar sólo el campo Fecha_venta que tenga registros del año 2022
- Ver los datos de una tabla
- Crear un procedimiento almacenado que muestre los clientes que tengan por nombre “Raúl”
- Crear un ciclo donde sí existe el ID_producto número “3”, deberá mostrar el nombre del mismo (Nombre_producto correspondiente)

Para **contar los registros existentes de una tabla**, se utilizará la siguiente línea de código, donde utilizamos count y group by para realizar los registros:

```
SELECT pais,COUNT(*)  
AS cantidad_de_personas_por_pais  
FROM clientes GROUP BY pais
```

De esta manera, se le indica además por medio de un alias que a esta tabla de conteo se le asigne el nombre: “cantidad de personas por país”, ya que de otro modo solo aparecería como bigcount, siendo menos intuitivo. El resultado fue:

pais character varying (50) 🔒	cantidad_de_personas_por_pais bigint 🔒
España	3
México	3
Argentina	2

Lo siguiente será, **verificar si existen o no registros duplicados**. Para ello, se utilizó la siguiente línea de código:

```
SELECT nombre, COUNT(*)  
FROM personas  
GROUP BY nombre  
HAVING COUNT(*) > 1;
```

En esta línea de código, aún nos basamos en la sentencia count, pero adicionalmente le añadimos la condicional having count, de manera que si el valor es mayor que 1, nos muestre el resultado.

Nota: Para esta actividad se creó otra tabla adicional denominada personas, a fin de no repetir los procesos, ya que eran demasiados.

La tabla personas era la siguiente:

dni character varying (8)	nombre character varying (45)	apellidopatemo character varying (45)	apellidomaterno character varying (45)	fechanacimiento date	municipio character varying (45)	estad chara
100012A	Juan	Vazquez	Perez	2020-05-08	Guadalajara	Jalisco
1001B12	Juan	Vazquez	Perez	2021-02-08	Merida	Yucat
1002W12	Carlos Miguel	Lopez	Perez	2020-03-15	Morelia	Mich
1003Q43	Maria Carlota	SAnchez	Perez	2018-02-02	Guasave	Sinal
1004S23	Casandra	Gavilan	Gonzalez	2021-07-25	Monterrey	Nuev

Por lo que la consulta quedó de la siguiente forma, al haber dos personas con el mismo nombre:

	nombre character varying (45)	count bigint
1	Juan	2

La siguiente consulta nos permitirá **ordenar de mayor a menor una tabla**. Para ello usare de ejemplo la tabla productos.

id_producto integer	nombre_producto character varying (20)	precio_producto numeric (8,2)	marca_producto character varying (20)
4789	Auriculares	89.95	SoundPro
5190	Smartwatch	199.99	WristTech
3023	Tablet	499.99	TabMaster
2045	Laptop	1299.50	ComputeX
1250	Oppo	1500.00	Movistar

Por medio de la siguiente línea de código, le indicaremos a la consola que muestre de mayor a menor la información de los precios gracias a ORDER BY.

```
SELECT*From producto
ORDER BY precio_producto DESC
```


Una vez hecho esto, ejecutamos con F5 y nos mostrará la tabla con la información ordenada como lo requerimos.

id_producto integer	nombre_producto character varying (20)	precio_producto numeric (8,2)	marca_producto character varying (20)
1250	Oppo	1500.00	Movistar
2045	Laptop	1299.50	ComputeX
3023	Tablet	499.99	TabMaster
5190	Smartwatch	199.99	WristTech
4789	Auriculares	89.95	SoundPro

Para seleccionar sólo el campo el campo **ID_producto**, usaremos la siguiente línea de código:

```
SELECT id_producto
From venta;
```

Quedando del siguiente modo:

id_producto integer
1250
2045
3023
4789

Mientras que para **id_cliente** será la siguiente línea de código:

```
SELECT id
FROM clientes
```

Mostrando la siguiente tabla:

id [PK] integer
1
2
3
4
-

Para Ordenar por orden alfabético el campo Nombre_cliente, usaremos de igual manera ORDER BY, pero esta vez con la asignación ASC, quedando el código del siguiente modo:

```
SELECT * FROM clientes
ORDER BY nombre asc
```

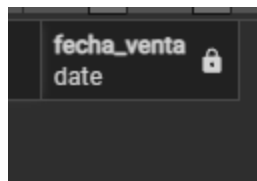
Mostrando la tabla del siguiente modo:

id [PK] integer	nombre character varying (50)	país character varying (50)
2	Ana	México
1	Carlos	México
5	Elena	México
3	Jorge	España
4	Laura	España

Para **mostrar sólo el campo Fecha_venta que tenga registros del año 2022**, vamos a utilizar la función EXTRACT, combinándola con YEAR FROM, quedando de la siguiente manera el código:

```
SELECT Fecha_venta  
FROM venta  
WHERE EXTRACT(YEAR FROM Fecha_venta) = 2022;
```

Al ejecutar la línea de código, no arroja resultados, debido a que en los registros, no hay ventas de ese año, sin embargo la consulta fue ejecutada exitosamente por lo que el código es funcional.



fecha_venta
date

Para simplemente **ver los datos de una tabla**, indicaremos con SELECT*FROM la tabla que deseamos visualizar. Quedando del siguiente modo el código y la tabla:

```
SELECT*FROM marca
```

id_marca	marca_producto
integer	character varying (20)
1	Movistar
2	ComputeX
3	TabMaster
4	SoundPro
5	WristTech

Para crear un procedimiento almacenado que muestre los clientes que tengan por nombre “Raúl”, utilizaremos un código más complejo.

```
CREATE OR REPLACE FUNCTION MostrarClientesRaulClientes()  
RETURNS TABLE(nombre VARCHAR)  
LANGUAGE plpgsql  
AS $$  
BEGIN  
    RETURN QUERY  
    SELECT c.nombre  
    FROM clientes AS c  
    WHERE c.nombre = 'Raul';  
END;  
$$;
```

PostgreSQL usa funciones en lugar de procedimientos, pero se pudo lograr el mismo efecto usando una función que devuelva los registros.

Nota: Al usar clientes AS c, estamos creando un alias (c) para la tabla clientes.

Esto nos permite referirnos a las columnas de la tabla usando este alias (por ejemplo, c.id, c.nombre, etc.), ya que PostgreSQL no identificaba si hacía referencia a la columna de la tabla o a una variable dentro de la función llamada id y mostraba error.

Una vez que se creó la función, se ejecutó por medio de la siguiente línea de código:

```
SELECT * FROM MostrarClientesRaulClientes();
```

Mostrando el siguiente resultado:

nombre
Raul

Por último, **para Crear un ciclo donde, si existe el ID_producto número “3”, deberá mostrar el nombre del mismo (Nombre_producto correspondiente),** vamos a utilizar la siguiente línea de código:

```
DO $$
DECLARE
    nombre_producto_var VARCHAR(20); -- Cambiamos el nombre de la variable
BEGIN
    -- Verificamos si el ID_producto número "3" existe
    FOR nombre_producto_var IN
        SELECT p.nombre_producto -- Usamos un alias para la tabla
        FROM producto p
        WHERE p.id_producto = 3023
    LOOP
        -- Si existe, mostramos el nombre del producto
        RAISE NOTICE 'Nombre del producto con ID 3: %', nombre_producto_var;
    END LOOP;
END $$;
```

Dada la complejidad de este último código, también procederé a explicarlo:

- **DO \$\$... \$\$:** Esta estructura permite ejecutar un bloque PL/pgSQL sin necesidad de crear una función almacenada.
- **DECLARE:** Aquí se declara la variable `nombre_producto`, que almacenará el nombre del producto correspondiente.
- **FOR ... IN:** Se utiliza un ciclo FOR que recorre cada registro que coincida con el `id_producto = 3023`. Como sólo esperamos un producto con ese id, el ciclo debería ejecutarse una vez (si el producto existe).
- **RAISE NOTICE:** Esta instrucción sirve para mostrar mensajes en la consola. *Nota: En este caso, muestra el nombre del producto si se encuentra el `id_producto = 3023`, ya que es el tercero en la columna `producto`. (derivado que en `id_producto` se ingresaron datos con más de un dígito, lo maneje así para que fuera funcional)*
- **Si el `id_producto` no existe,** no mostrará nada, pero no arrojará error.

Tras la ejecución con F5, el resultado en consola es el siguiente:

```
NOTICE: Nombre del producto con ID 3: Tablet
DO

Consulta retornó exitosamente en 205 msec.
```

Conclusión

En esta actividad, se ha abordado la importancia de las consultas SQL como una herramienta indispensable para acceder, filtrar y organizar la información almacenada. Las operaciones realizadas, desde la verificación de duplicados hasta la creación de procedimientos almacenados, demuestran cómo SQL permite optimizar el manejo de datos y automatizar procesos que, de otro modo, requerirían tiempo y esfuerzo significativos.

En el campo laboral, la capacidad de ejecutar consultas SQL de manera efectiva permite a los profesionales extraer información crítica de los sistemas en los que trabajan. Esto es esencial para generar reportes, analizar tendencias o incluso para el mantenimiento diario de la base de datos. Además, la automatización de tareas mediante procedimientos almacenados reduce la carga operativa, permitiendo que los administradores de bases de datos y desarrolladores se enfoquen en tareas más estratégicas.

Considero que el dominio de SQL es una habilidad fundamental en el desarrollo profesional de cualquier persona que trabaje con datos. La comprensión y ejecución de consultas, como las realizadas en esta actividad, son esenciales no solo para el correcto funcionamiento de los sistemas de información, sino también para la toma de decisiones basadas en datos sólidos y bien organizados.

Referencias

W3Schools.com. (n.d.). https://www.w3schools.com/sql/sql_datatypes.asp

TecnoBinaria. (2019, Julio 23). *Cómo EXPORTAR e IMPORTAR una BASE de DATOS en PGADMIN | Curso de PostgreSQL #65* [Video]. YouTube.

<https://www.youtube.com/watch?v=icEvkyIXqug>

Flortronic. (2021, Septiembre 28). *[5 horas y media] Curso completo de bases de datos en postgresQL desde cero hasta avanzado Parte 1* [Video]. YouTube.

<https://www.youtube.com/watch?v=HGfrzsGg3As>