

REPÚBLICA BOLIVARIANA DE VENEZUELA
UNIVERSIDAD RAFAEL URDANETA
ASIGNATURA: MINERÍA DE DATOS
PROFESOR: GENYELBERT ACOSTA
ESCUELA DE INGENIERÍA EN COMPUTACIÓN

**ENTREGA DE PROYECTO FINAL
MINERÍA DE DATOS**

INTEGRANTES

- ÁLVAREZ, ANDRES
- GRATEROL, JULIO
- KNEIDER, VICTOR

MARACAIBO, 09 DE AGOSTO DE 2023

INTRODUCCIÓN

En la era digital en la que vivimos, la cantidad de datos generados diariamente es abrumadora. Cada interacción en línea, cada transacción comercial y cada dispositivo conectado contribuyen a la creación de un vasto océano de información. Sin embargo, la mera acumulación de datos no es suficiente por sí misma. Para que estos datos sean verdaderamente valiosos, es necesario convertirlos en información significativa y accionable.

Es en este contexto que entran en juego las bases de datos. Las bases de datos se han convertido en pilares fundamentales para cualquier organización que busque tomar decisiones informadas y desarrollar estrategias efectivas. Estas bases de datos son depósitos estructurados de información que permiten almacenar, organizar y recuperar datos de manera eficiente. Al aprovechar estos recursos, las empresas pueden obtener una visión profunda de sus operaciones, clientes, productos y servicios.

Una de las técnicas más poderosas y ampliamente utilizadas en el campo de la minería de datos es el algoritmo Apriori. Este algoritmo se utiliza para descubrir patrones o reglas de asociación en conjuntos de datos. Al aplicar el algoritmo Apriori a una base de datos, podemos identificar relaciones interesantes y significativas entre diferentes elementos. Estas reglas de asociación pueden revelar patrones de comportamiento, preferencias del cliente, correlaciones entre variables y mucho más.

Al comprender y utilizar los resultados obtenidos mediante el algoritmo Apriori, las organizaciones pueden generar estrategias efectivas para aumentar la incidencia de comportamientos o tendencias deseadas. Por ejemplo, si una empresa desea promocionar un producto en particular, el análisis de reglas de asociación puede revelar qué características o eventos están asociados con un mayor interés en ese producto. Esto permite a la empresa seleccionar el mejor momento para lanzar una campaña promocional y maximizar su impacto en el mercado.

Desarrollo

A través de consultas a la base de datos, las empresas pueden obtener información valiosa sobre su desempeño y tomar decisiones fundamentadas. Mediante la identificación de los productos más vendidos, las organizaciones pueden comprender qué productos generan mayores ingresos y ajustar sus estrategias de promoción en consecuencia. Además, al analizar los datos de ventas y aplicar consultas adecuadas, es posible identificar a los mayores y menores consumidores, lo que permite desarrollar estrategias para fomentar la lealtad de los clientes más frecuentes y atraer la atención de aquellos con menor participación en el mercado. Estas consultas a la base de datos proporcionan a las empresas una visión profunda de su rendimiento y les permiten tomar decisiones informadas para maximizar su competitividad y lograr un crecimiento sostenible.

- **Pais con Mayor Cantidad de Clientes Registrados**

```
SELECT p.nombre AS pais, COUNT(c.id) AS total_clientes
FROM paises p
INNER JOIN ciudades ci ON p.id = ci.id_pais
INNER JOIN clientes c ON ci.id = c.id_ciudad
GROUP BY p.id
ORDER BY total_clientes DESC
LIMIT 1;
```

- **Precio, costo y ganancia promedio de cada producto**

```
SELECT productos.nombre, AVG(productos.costo) as avg_cost,
AVG(productos.precio) as avg_price, AVG(productos.precio -
productos.costo) as avg_profit
FROM productos
GROUP BY productos.nombre;
```

- **TOP 10 productos más caros junto a su fabricante**

```
SELECT prod.nombre as nombre_producto, fab.nombre as nombre_fabricante,  
prod.precio FROM productos as prod  
INNER JOIN fabricantes as fab  
ON prod.id_fabricante = fab.id  
ORDER BY prod.precio DESC  
LIMIT 10;
```

- **TOP 10 productos más vendidos**

```
SELECT p.nombre AS producto, SUM(rp.cantidad) AS total_vendido  
FROM productos p  
INNER JOIN recibos_productos rp ON p.id = rp.id_producto  
GROUP BY p.id  
ORDER BY total_vendido DESC  
LIMIT 10;
```

- **TOP 10 productos menos vendidos**

```
SELECT p.nombre AS producto, SUM(rp.cantidad) AS total_vendido  
FROM productos p  
INNER JOIN recibos_productos rp ON p.id = rp.id_producto  
GROUP BY p.id  
ORDER BY total_vendido ASC  
LIMIT 10;
```

- **TOP 10 productos más vendidos para un país en específico**

```
SELECT p.nombre AS producto, pa.nombre AS pais, SUM(rp.cantidad) AS
total_vendido
FROM productos p
INNER JOIN recibos_productos rp ON p.id = rp.id_producto
INNER JOIN recibos r ON rp.id_recibo = r.id
INNER JOIN clientes c ON r.id_cliente = c.id
INNER JOIN ciudades ci ON c.id_ciudad = ci.id
INNER JOIN paises pa ON ci.id_pais = pa.id
WHERE pa.nombre = 'Venezuela'
GROUP BY p.id, pa.id
ORDER BY total_vendido DESC
LIMIT 10;
```

- **Nombre y Abreviatura de todos los paises registrados**

```
SELECT p.nombre, p.abreviatura, COUNT(c.id) AS total_ciudades
FROM paises p
INNER JOIN ciudades c ON p.id = c.id_pais
GROUP BY p.id;
```

- **Total de ventas por cliente, mostrando el nombre del cliente, el número de recibos y el monto total gastado por cada cliente**

```
SELECT c.nombres, c.apellidos, COUNT(r.id) AS numero_recibos,
SUM(pr.precio) AS monto_total
FROM clientes c
INNER JOIN recibos r ON c.id = r.id_cliente
INNER JOIN recibos_productos rp ON r.id = rp.id_recibo
INNER JOIN productos pr ON rp.id_producto = pr.id
GROUP BY c.id;
```

- **Nombre del fabricante y el Número total de productos fabricados por cada fabricante, ordenados de mayor a menor**

```
SELECT f.nombre AS fabricante, COUNT(p.id) AS total_productos
FROM fabricantes f
INNER JOIN productos p ON f.id = p.id_fabricante
GROUP BY f.nombre
ORDER BY total_productos DESC;
```

- **Productos más vendidos entre semana para un día de la semana específico (por ejemplo, miércoles)**

```
SELECT p.nombre AS producto, SUM(rp.cantidad) AS total_vendido
FROM recibos r
INNER JOIN recibos_productos rp ON r.id = rp.id_recibo
INNER JOIN productos p ON rp.id_producto = p.id
WHERE EXTRACT(DOW FROM r.fecha) = 3 -- 3 representa el miércoles (0 = domingo, 1 = lunes, ..., 6 = sábado)
GROUP BY p.nombre
ORDER BY total_vendido DESC;
```

- **Día de la semana con el mayor número de ventas**

```
SELECT EXTRACT(DOW FROM r.fecha) AS dia_semana, COUNT(r.id) AS
numero_ventas
FROM recibos r
GROUP BY EXTRACT(DOW FROM r.fecha)
ORDER BY numero_ventas DESC
LIMIT 1;
```

