Tarea 2 - Consultas

Bases de Datos para Ingeniería - Laboratorio 2024

En este documento se presentan posibles soluciones de la Tarea 2.

1. Soluciones

1. *Devolver el conjunto de los géneros musicales de la artista "Amy Winehouse".*

```sql
SELECT DISTINCT (g.name)
FROM
    artist a, album al, track t, genre g
WHERE
    a.name = 'Amy Winehouse' AND
    a.artist_id = al.artist_id AND
    al.album_id = t.album_id AND
    t.genre_id = g.genre_id
```

2. *Devolver el nombre de los artistas con un único género musical.*

   Solución 1:

```sql
SELECT DISTINCT a.name
FROM
    artist a, album al, track t, genre g
WHERE
    a.artist_id = al.artist_id AND
    al.album_id = t.album_id AND
    t.genre_id = g.genre_id
GROUP BY a.artist_id
HAVING COUNT(DISTINCT g.genre_id) = 1
```

   Solución 2:

```sql
SELECT DISTINCT a.name
FROM
    artist a NATURAL JOIN
    album al JOIN
    track t ON al.album_id = t.album_id JOIN
    genre g on t.genre_id = g.genre_id
WHERE NOT EXISTS (
    SELECT 1
    FROM
        album al2 NATURAL JOIN
        track t2 JOIN
        genre g2 ON t2.genre_id = g2.genre_id
    WHERE a.artist_id = al2.artist_id AND g2.genre_id <> g.genre_id
)
```

3. *Devolver el nombre, la duración (en minutos) y el precio de la canción con mayor duración comprada por clientes de Canadá.*

```
SELECT t.name, (t.milliseconds/60000) AS duration, t.unit_price
FROM customer c, invoice i, invoice_line il, track t
WHERE c.customer_id = i.customer_id
    AND i.invoice_id = il.invoice_id
    AND t.track_id = il.track_id
    AND c.country = 'Canada'
    AND t.milliseconds = (
            SELECT MAX(milliseconds)
            FROM customer c, invoice i, invoice_line il, track t
            WHERE c.customer_id = i.customer_id
                AND i.invoice_id = il.invoice_id
                AND t.track_id = il.track_id
                AND c.country = 'Canada'
        )
```

4. *Devolver el nombre y apellido de los clientes que compraron al menos 2 canciones distintas del género musical "Rock".*

   Observación: *Al pedir canciones distintas, se debe considerar el invoice y no la columna "quantity"de invoice_line.*

```
SELECT c.first_name, c.last_name
FROM
    genre g JOIN
    track t on g.genre_id = t.genre_id JOIN
    invoice_line il ON t.track_id = il.track_id NATURAL JOIN
    invoice i NATURAL JOIN
    customer c
WHERE g.name = 'Rock'
GROUP BY g.genre_id, c.customer_id
HAVING COUNT(DISTINCT t.track_id) > 1
```

5. *Devolver el nombre y apellido de los clientes que compraron todas las canciones de la artista "Amy Winehouse".*

```
SELECT DISTINCT c1.first_name, c1.last_name
FROM
    artist a1 JOIN
    album al1 ON a1.artist_id = al1.artist_id JOIN
    track t1 ON al1.album_id = t1.album_id JOIN
    invoice_line il1 ON t1.track_id = il1.track_id JOIN
    invoice i1 ON i1.invoice_id = il1.invoice_id JOIN
    customer c1 ON i1.customer_id = c1.customer_id
WHERE a1.name = 'Amy Winehouse'
AND NOT EXISTS (
    SELECT 1
    FROM
        album al2 JOIN track t2 ON al2.album_id = t2.album_id
    WHERE al2.artist_id = a1.artist_id
        AND NOT EXISTS (
            SELECT 1
            FROM invoice_line il3 NATURAL JOIN
                invoice i3 NATURAL JOIN
                customer c3
            WHERE il3.track_id = t2.track_id AND
                c3.customer_id = c1.customer_id
    )
)
```

6. *Devolver el nombre, apellido y cantidad de compras de cada cliente asesorado por "Steve Johnson".*

```sql
SELECT c.first_name, c.last_name, COUNT(DISTINCT i.invoice_id) AS number_of_purchases
FROM employee e JOIN
    customer c ON c.support_rep_id = e.employee_id JOIN
    invoice i ON c.customer_id = i.customer_id JOIN
    invoice_line il on i.invoice_id = il.invoice_id
WHERE e.first_name = 'Steve' AND e.last_name = 'Johnson'
GROUP BY c.customer_id
```

7. *Devolver el nombre del género musical con más cantidad de playlists.*

```sql
SELECT g.name
FROM
    genre g
    JOIN track t ON g.genre_id = t.genre_id
    JOIN playlist_track pt ON t.track_id = pt.track_id
GROUP BY g.genre_id
HAVING COUNT(DISTINCT pt.playlist_id)
                >= ALL
        (
                        SELECT COUNT(DISTINCT pt.playlist_id)
                        FROM
                                genre g
                                JOIN track t ON g.genre_id = t.genre_id
                                JOIN playlist_track pt ON t.track_id = pt.track_id
                        GROUP BY g.genre_id
                )
```

8. *Devolver el nombre y la cantidad de ventas para los álbumes con las 3 mayores cantidades de ventas. La cantidad de ventas se debe devolver en una columna denominada "number of purchases". Ordenar el resultado según la cantidad de ventas de forma decreciente.*

```sql
SELECT al.title, COUNT(DISTINCT i.invoice_id) as number_of_purchases
FROM
    album al
    JOIN track t ON t.album_id = al.album_id
    JOIN invoice_line il ON il.track_id = t.track_id
    JOIN invoice i ON i.invoice_id = il.invoice_id
GROUP BY al.album_id
HAVING COUNT(DISTINCT i.invoice_id) IN
(
    -- El top 3 de cantidad de ventas
    SELECT DISTINCT COUNT(DISTINCT i2.invoice_id)
    FROM
        album al2
        JOIN track t2 ON t2.album_id = al2.album_id
        JOIN invoice_line il2 ON il2.track_id = t2.track_id
        JOIN invoice i2 ON i2.invoice_id = il2.invoice_id
    GROUP BY al2.album_id
    ORDER BY COUNT(DISTINCT i2.invoice_id) DESC
    LIMIT 3
)
ORDER BY COUNT(DISTINCT i.invoice_id) DESC
```

9. *Devolver el identificador, nombre y apellido del empleado que asesoró la mayor cantidad de ventas en el 2024*

```
SELECT e.employee_id, e.first_name, e.last_name
FROM employee e
    join customer c on e.employee_id = c.support_rep_id
    join invoice i on i.customer_id = c.customer_id
WHERE i.invoice_date BETWEEN '2024-01-01' and '2024-12-31'
GROUP BY e.employee_id
HAVING COUNT(DISTINCT i.invoice_id)
        >=
        ALL (
            SELECT COUNT(DISTINCT i2.invoice_id)
            FROM employee e2
                join customer c2 on e2.employee_id = c2.support_rep_id
                join invoice i2 on i2.customer_id = c2.customer_id
            WHERE i2.invoice_date BETWEEN '2024-01-01' and '2024-12-31'
            GROUP BY e2.employee_id
        )
```

10. *Describa lo que realiza la siguiente consulta. Justifique*

```
SELECT *, COUNT(t.track_id) as count
FROM invoice_line il
    JOIN invoice i ON i.invoice_id = il.invoice_id
    JOIN track t ON t.track_id = il.track_id
WHERE i.invoice_date BETWEEN '2013-01-01' and '2013-12-31'
GROUP BY t.track_id
ORDER BY count DESC
```

Esta consulta **no es sintácticamente correcta**; ya que en la cláusula SELECT se retornan columnas que no están referenciadas en la cláusula del GROUP BY.
En SQL cuando se utiliza una consulta con agrupación, las columnas que se retornan deben aparecer en la cláusula del GROUP BY o deben depender de las columnas referenciadas en las cláusulas del GROUP BY.