```sql
-- Selecciona todos los registros de la tabla Albums.

SELECT * FROM album;

-- Selecciona todos los géneros únicos de la tabla Genres.

SELECT DISTINCT name FROM genre;

-- Cuenta el número de pistas por género.

SELECT g.name AS genre_name, COUNT(t.track_id) AS track_count
FROM genre g
JOIN track t ON g.genre_id = t.genre_id
GROUP BY g.name;

-- Encuentra la longitud total (en milisegundos) de todas las pistas para cada álbum.

SELECT a.title AS album_title, SUM(t.milliseconds) AS total_length
FROM album a
JOIN track t ON a.album_id = t.album_id
GROUP BY a.title;

-- Lista los 10 álbumes con más pistas.

SELECT a.title AS album_title, COUNT(t.track_id) AS track_count
FROM album a
JOIN track t ON a.album_id = t.album_id
GROUP BY a.title
ORDER BY track_count DESC
LIMIT 10;

-- Encuentra la longitud promedio de la pista para cada género.

SELECT g.name AS genre_name, AVG(t.milliseconds) AS average_length
FROM genre g
JOIN track t ON g.genre_id = t.genre_id
GROUP BY g.name;

-- Para cada cliente, encuentra la cantidad total que han gastado.

SELECT c.first_name, c.last_name, SUM(i.total) AS total_spent
FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
GROUP BY c.customer_id;

-- Para cada país, encuentra la cantidad total gastada por los clientes.

SELECT c.country, SUM(i.total) AS total_spent
```

```sql
FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
GROUP BY c.country;

-- Clasifica a los clientes en cada país por la cantidad total que han gastado.

SELECT c.country, c.first_name, c.last_name, SUM(i.total) AS total_spent
FROM customer c
JOIN invoice i ON c.customer_id = i.customer_id
GROUP BY c.country, c.customer_id
ORDER BY c.country, total_spent DESC;

-- Para cada artista, encuentra el álbum con más pistas y clasifica a los artistas por este
número.

SELECT ar.name AS artist_name, a.title AS album_title, COUNT(t.track_id) AS track_count
FROM artist ar
JOIN album a ON ar.artist_id = a.artist_id
JOIN track t ON a.album_id = t.album_id
GROUP BY ar.artist_id, a.album_id
ORDER BY track_count DESC;

-- Selecciona todas las pistas que tienen la palabra "love" en su título.

SELECT * FROM track WHERE lower(name) LIKE 'love %' or lower(name) LIKE '% love' or
lower(name) LIKE '% love %' or lower(name) LIKE 'love';

-- Selecciona a todos los clientes cuyo primer nombre comienza con 'A'.

SELECT * FROM customer WHERE first_name LIKE 'A%';

-- Calcula el porcentaje del total de la factura que representa cada factura.

SELECT i.invoice_id, round(i.total * 100 / (SELECT SUM(total) FROM invoice), 2) AS
invoice_percentage
FROM invoice i
ORDER BY invoice_percentage DESC;

-- Calcula el porcentaje de pistas que representa cada género.

SELECT g.name AS genre_name, COUNT(t.track_id) AS tracks, round(COUNT(t.track_id) *
100.0 / (SELECT COUNT(track_id) FROM track), 2) AS genre_percentage
FROM track t
JOIN genre g ON g.genre_id = t.genre_id
GROUP BY g.name
ORDER BY genre_percentage DESC;

-- Para cada cliente, compara su gasto total con el del cliente que gastó más.
```

```
WITH total_spent_per_customer AS (
   SELECT c.customer_id, c.first_name, c.last_name, SUM(i.total) AS total_spent
   FROM customer c
   JOIN invoice i ON c.customer_id = i.customer_id
   GROUP BY c.customer_id
)
SELECT t.first_name, t.last_name, t.total_spent,
      (SELECT MAX(total_spent) FROM total_spent_per_customer) AS max_spent
FROM total_spent_per_customer t;


-- Para cada factura, calcula la diferencia en el gasto total entre ella y la factura anterior.

SELECT i.invoice_id, i.total - LAG(i.total) OVER (ORDER BY i.invoice_date) AS
diff_from_previous
FROM invoice i;

-- Para cada factura, calcula la diferencia en el gasto total entre ella y la próxima factura.

SELECT i.invoice_id, LEAD(i.total) OVER (ORDER BY i.invoice_date) - i.total AS
diff_from_next
FROM invoice i;

-- Encuentra al artista con el mayor número de pistas para cada género.

WITH track_count_per_artist AS (
   SELECT g.name AS genre_name, ar.name AS artist_name, COUNT(t.track_id) AS
track_count
   FROM genre g
   JOIN track t ON g.genre_id = t.genre_id
   JOIN album a ON t.album_id = a.album_id
   JOIN artist ar ON a.artist_id = ar.artist_id
   GROUP BY g.name, ar.artist_id
)
SELECT genre_name, artist_name, track_count
FROM track_count_per_artist
WHERE (genre_name, track_count) IN (
   SELECT genre_name, MAX(track_count)
   FROM track_count_per_artist
   GROUP BY genre_name
)
ORDER BY genre_name;

-- Compara el total de la última factura de cada cliente con el total de su factura anterior.

WITH ranked_invoices AS (
   SELECT i.invoice_id, i.customer_id, i.total, i.invoice_date,
        ROW_NUMBER() OVER (PARTITION BY i.customer_id ORDER BY i.invoice_date
DESC) AS rn
   FROM invoice i
)
SELECT ri1.customer_id, ri1.invoice_id AS current_invoice_id, ri1.total AS
current_invoice_total,
      ri2.invoice_id AS previous_invoice_id, ri2.total AS previous_invoice_total
FROM ranked_invoices ri1
LEFT JOIN ranked_invoices ri2 ON ri1.customer_id = ri2.customer_id AND ri1.rn = 1 AND
ri2.rn = 2
WHERE ri1.rn = 1;

-- Encuentra cuántas pistas de más de 3 minutos tiene cada álbum.

SELECT a.title AS album_title, COUNT(t.track_id) AS tracks_over_3_minutes
FROM album a
JOIN track t ON a.album_id = t.album_id
WHERE t.milliseconds > 180000
GROUP BY a.album_id;
```