```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,
	count(t.track_id) as pistas
from track as t
inner join genre as g on t.genre_id = g.genre_id
group by g.name;

-- Encuentra la longitud total (en milisegundos) de todas las pistas para
cada álbum.
select
	a.title,
	sum(milliseconds) as "duración(ms)"
from album as a
inner join track as t on a.album_id = t.album_id
group by a.title;

-- Lista los 10 álbumes con más pistas.
select
	a.title,
	count(t.track_id) as pistas
from album as a
inner join track as t on a.album_id = t.album_id
group by a.title
order by pistas desc
limit 10;

-- Encuentra la longitud promedio de la pista para cada género.
select
	g.name,
	avg(t.milliseconds) as "long promedio (ms)"
from genre as g
inner join track as t on t.genre_id = g.genre_id
group by g.genre_id, g.name;

-- Para cada cliente, encuentra la cantidad total que han gastado.
select
	c.customer_id,
	c.first_name,
	c.last_name,
	sum(i.total) as "total gastado"
```

```
from customer c
inner join invoice i on i.customer_id = c.customer_id
group by c.customer_id;

-- Para cada país, encuentra la cantidad total gastada por los clientes.
select
	i.billing_country,
	sum(i.total) as "gasto total"
from invoice i
group by billing_country
order by sum(i.total) desc;

-- Clasifica a los clientes en cada país por la cantidad total que han
gastado.
select
	c.first_name,
	c.last_name,
	i.billing_country,
	sum(i.total) as "gasto total"
from customer c
inner join invoice i on i.customer_id = c.customer_id
group by c.customer_id, i.billing_country
order by i.billing_country, sum(i.total) desc;

-- Para cada artista, encuentra el álbum con más pistas y clasifica a los
artistas por este número.
with pistas_album as (
	select
		a.name,
		a2.title,
		count(t.track_id) as "cantidad pistas",
		row_number() over (partition by a.artist_id order by
count(t.track_id) desc) as rn
	from artist a
	inner join album a2 on a2.artist_id = a.artist_id
	inner join track t on t.album_id = a2.album_id
	group by a.artist_id, a2.title
)

select
	name,
	title,
	"cantidad pistas"
from pistas_album
where rn = 1
order by "cantidad pistas" desc;

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

```
        name
from track
where lower(name) LIKE 'love %' or
      lower(name) LIKE '% love' or
      lower(name) like '% love %' or
      lower(name) = 'love';

-- Selecciona a todos los clientes cuyo primer nombre comienza con 'A'.
select
        first_name,
        last_name
from customer
where lower(first_name) like 'a%';

-- Calcula el porcentaje del total de la factura que representa cada
factura.
select
        invoice_id,
        total,
        round(total*100 /
                (select sum(total) from invoice), 2) as "procentaje (%)"
from invoice i
order by "procentaje (%)" desc;

-- Calcula el porcentaje de pistas que representa cada género.
select
        g.name AS genero,
        count(t.track_id ) as pistas,
        round(count(t.track_id) * 100.0 /
        (select count(track_id ) from track), 2) as "porcentaje por
genero (%)"
from track t
join genre g on t.genre_id = g.genre_id
group by g.name
order by "porcentaje por genero (%)" desc;

-- Para cada cliente, compara su gasto total con el del cliente que gastó
más.
WITH gasto_por_cliente AS (
    SELECT
        customer_id,
        SUM(total) AS gasto_total
    FROM invoice
    GROUP BY customer_id
),
cliente_top_gasto AS (
    SELECT
        MAX(gasto_total) AS gasto_maximo
    FROM gasto_por_cliente
```

```
)
select
        c.first_name as nombre,
        c.last_name as apellido,
        gpc.gasto_total,
        ROUND((gpc.gasto_total * 100.0 / ctg.gasto_maximo), 2) AS
"porcentaje respecto al mayor (%)"
from gasto_por_cliente gpc
join customer c on gpc.customer_id = c.customer_id
CROSS JOIN cliente_top_gasto ctg
ORDER BY "porcentaje respecto al mayor (%)" DESC;

-- Para cada factura, calcula la diferencia en el gasto total entre ella
y la factura anterior.
select
        invoice_id,
        total,
        lag(total) over (order by invoice_id) as total_anterior,
        total - lag(total) over (order by invoice_id) as diferencia
from invoice;

-- Para cada factura, calcula la diferencia en el gasto total entre ella
y la próxima factura.
select
        invoice_id,
        total,
        lead(total) over (order by invoice_id) as total_proxima,
        total - lead(total) over (order by invoice_id) as diferencia
from invoice;

-- Encuentra al artista con el mayor número de pistas para cada género.
with pistas_por_artista_genero as (
    select g.name as genero, ar.name as artista, count(t.track_id) as
total_pistas, row_number() over (partition by g.genre_id order by
count(t.track_id) desc) as pos
    from track t
    join genre g on t.genre_id = g.genre_id
    join album al on t.album_id = al.album_id
    join artist ar on al.artist_id = ar.artist_id
    group by g.name, g.genre_id, ar.name
)

select
    genero,
    artista,
    total_pistas
from pistas_por_artista_genero
where pos = 1;
```

```sql
-- Compara el total de la última factura de cada cliente con el total de
su factura anterior.
with facturas_ordenadas as (
    select
        customer_id,
        invoice_id,
        invoice_date,
        total,
        lag(total) over (partition by customer_id order by invoice_date)
as total_anterior
    from invoice
)

select
    customer_id,
    invoice_id,
    total as total_actual,
    total_anterior,
    total - total_anterior as diferencia
from facturas_ordenadas
where total_anterior is not null;

-- Encuentra cuántas pistas de más de 3 minutos tiene cada álbum.
select
    a.album_id,
    a.title,
    count(t.track_id) as pistas_mas_3_minutos
from album a
join track t on a.album_id = t.album_id
where t.milliseconds > 180000  -- 3 minutos en milisegundos
group by a.album_id, a.title
order by pistas_mas_3_minutos desc;
```