

1. Find the total spending of each customer.

```
SELECT C.customer_id,  
       C.first_name,  
       C.last_name ,  
       SUM(total) Top_Invoice  
FROM Customer C  
JOIN Invoice I ON I.customer_id=C.customer_id  
GROUP BY  
       C.customer_id ,  
       C.first_name,  
       C.last_name
```

2. List the names of all artists who have released an album.

```
SELECT Art.name  
FROM Artist Art  
JOIN Album A ON A.artist_id=Art.artist_id  
GROUP BY Art.name  
HAVING COUNT(A.album_id)=1
```

3. List songs of a specific genre along with their album titles

```
SELECT DISTINCT G.name,  
               A.title  
FROM Genre G  
JOIN Track T ON G.genre_id=T.genre_id  
JOIN Album A ON T.album_id=A.album_id  
WHERE G.name LIKE '%Blues%'
```

4. List all playlists that contain a specific song.

```
SELECT PL.playlist_id,  
       T.name as Track_Name ,
```

```

        PL.name as List_Name
FROM Track T
JOIN playlist_track P ON T.track_id=P.track_id
JOIN playlist PL ON PL.playlist_id=P.playlist_id
WHERE T.track_id=222

```

5. Find the total revenue generated from each invoice by country.

```

SELECT billing_country,
        SUM(total) AS Total_Invoice
FROM Invoice I
GROUP BY billing_country
ORDER BY Total_Invoice DESC

```

6. Write a query that returns the person who has spent the most money.

```

SELECT TOP 1 I.customer_id ,C.first_name,C.last_name,
        SUM(I.Total)as Toplam
FROM Invoice I
JOIN Customer C
        ON C.customer_id=I.customer_id
GROUP BY I.customer_id,C.first_name,C.last_name
ORDER BY Toplam  DESC

```

5. List the email addresses, first names, last names, and genres of rock music listeners, sorted by email address in ascending order

```

SELECT DISTINCT G.name AS type,
        C.first_name,
        C.last_name,
        C.email
FROM Genre G
JOIN Track T ON G.genre_id=T.genre_id
JOIN Invoice_line I ON I.track_id= T.track_id

```

```
JOIN Invoice Inv ON Inv.invoice_id=l.invoice_id
JOIN Customer C ON C.customer_id=Inv.customer_id
WHERE G.name = 'Rock'
ORDER BY C.email ASC
```

6. Write a query that returns the artist name and total number of tracks for the top 10 rock bands

```
SELECT TOP 10 Art.name ,
           COUNT(T.track_id) AS Unity
FROM Genre G
```

```
JOIN Track T ON G.genre_id=T.genre_id
JOIN Album A ON A.album_id=T.album_id
JOIN Artist Art ON Art.artist_id= A.artist_id
```

```
WHERE G.name LIKE '%Rock%'
GROUP BY Art.name
ORDER BY Unity DESC
```

7. Write a query that returns the names and durations in milliseconds of songs that are longer than the average song duration, sorted from longest to shortest.

```
SELECT name as Track_Name,
       milliseconds
FROM Track
WHERE
       milliseconds > (SELECT AVG(milliseconds) FROM Track)
ORDER BY milliseconds DESC
```