Please answer the following questions:

1. Write a query that displays InvoiceId, CustomerId and total dollar amount for each invoice, sorted first by CustomerId (in ascending order), and then by total dollar amount (in descending order).

﻿SELECT InvoiceId, CustomerId, Total FROM invoices ORDER BY CustomerId ASC, total DESC;

1. Write a query that displays InvoiceId, CustomerId and total dollar amount for each invoice, but this time sorted first by total dollar amount (in descending order), and then by CustomerId (in ascending order).

﻿SELECT InvoiceId, CustomerId, Total FROM invoices ORDER BY Total DESC, CustomerId ASC;

1. Compare the results of these two queries above. How are the results different when you switch the column you sort on first? (Explain it in your own words.)

The difference is, in first case table arranged ascending order by customer ID from small to largest, so if a customer ID corresponds to multiples transactions, total of all is in largest to smallest order per customer (customerID). Whereas in second case, transaction amounts are listed in descending order and customerId in ascending order.

1. Write a query to pull the first 10 rows and all columns from the invoices table that have a dollar amount of total greater than or equal to 10.

﻿SELECT \* FROM invoices WHERE Total >= 10 LIMIT 10;

1. Write a query to pull the first 5 rows and all columns from the invoices table that have a dollar amount of total less than 10.

﻿SELECT \* FROM invoices WHERE total < 10 limit 5;

1. Find all track names that start with 'B' and end with 's'.

﻿SELECT name FROM tracks WHERE name like "B%" or name like "%s"

1. Use the invoices table to find all information regarding invoices whose billing address is USA or Germany or Norway or Canada and invoice date is at any point in 2010, sorted from newest to oldest.

﻿ SELECT \* FROM invoices WHERE BillingCountry IN ("USA", "Germany", "Norway", "Canada") AND InvoiceDate BETWEEN '2010-01-01' AND '2010-12-31' ORDER BY InvoiceDate DESC;

1. How many tracks does each album have? Your solution should include Album id and its number of tracks sorted from highest to lowest.

SELECT t.AlbumId, count(t.TrackId) AS TrackCount from tracks t GROUP BY t.AlbumId ORDER BY count(t.TrackId) DESC;

1. Find the album title of the tracks. Your solution should include track name and its album title.

Select t.name, a.title from tracks t join albums a on a.AlbumId=t.AlbumId;

1. Find the minimum duration of the track in an album. Your solution should include track name, album id, album title and duration of the track sorted from highest to lowest.

﻿ SELECT t.Name, t.AlbumId, a.Title, min(Milliseconds) AS Duration FROM tracks t INNER JOIN albums a on t.AlbumId = a.AlbumId GROUP BY t.AlbumId ORDER BY Duration DESC;

1. Find the total duration of each album. Your solution should include track name, album id, album title and its total duration sorted from highest to lowest.

﻿ SELECT t.name, t.AlbumId, a.Title, SUM(Milliseconds) AS Duration FROM tracks t join albums a on t.AlbumId = a.AlbumId GROUP BY t.AlbumId ORDER BY Duration DESC;

1. Based on the previous question, find the albums whose total duration is higher than 70 minutes. Your solution should include album title and total duration.

﻿ SELECT a.Title, SUM(Milliseconds) AS Duration FROM tracks t join albums a on t.AlbumId = a.AlbumId GROUP BY t.AlbumId HAVING Duration > 4200000 ORDER BY Duration DESC;