\*\*Teamwork Document for SQL Fundamentals:\*\*

SQL CC-11

Learning Outcomes:

- At the end of this coding challenge you will be correctly able to;

- Use keywords and statements in SQL

- Use aggregate functions

- Use GROUP BY and HAVING keywords

- Use JOINS

\*\*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, total DESC;

2. 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;

3. 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.)

In the first query, the sorting was done well, as desired, both in terms of customer ID and total. But in the second, only the total is sorted.

4. 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;

5. 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

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

SELECT \* FROM tracks

WHERE name LIKE "B%s"

7. 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 \*, strftime('%Y', InvoiceDate) as Year FROM invoices

WHERE BillingCountry IN ("USA", "Germany", "Norway", "Canada") AND Year = "2010"

ORDER BY InvoiceDate DESC;

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

SELECT tr.AlbumId, ii.Quantity

FROM tracks tr

INNER JOIN invoice\_items ii ON ii.TrackId = tr.TrackId

ORDER BY ii.Quantity DESC

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

SELECT tr.name, al.Title

FROM tracks tr

INNER JOIN albums al ON tr.AlbumId = al.AlbumId

10. 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 tr.name, tr.AlbumId, al.Title, tr.Milliseconds

FROM tracks tr

INNER JOIN albums al ON tr.AlbumId = al.AlbumId

ORDER BY Milliseconds DESC

11. 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 tr.name, tr.AlbumId, al.Title, SUM(tr.Milliseconds) AS TotalDuration

FROM tracks tr

INNER JOIN albums al ON tr.AlbumId = al.AlbumId

GROUP BY Title

ORDER BY TotalDuration DESC

12. 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 al.Title, SUM(tr.Milliseconds) AS TotalDuration

FROM tracks tr

INNER JOIN albums al ON tr.AlbumId = al.AlbumId

GROUP BY Title

HAVING TotalDuration > 4200000

ORDER BY TotalDuration DESC