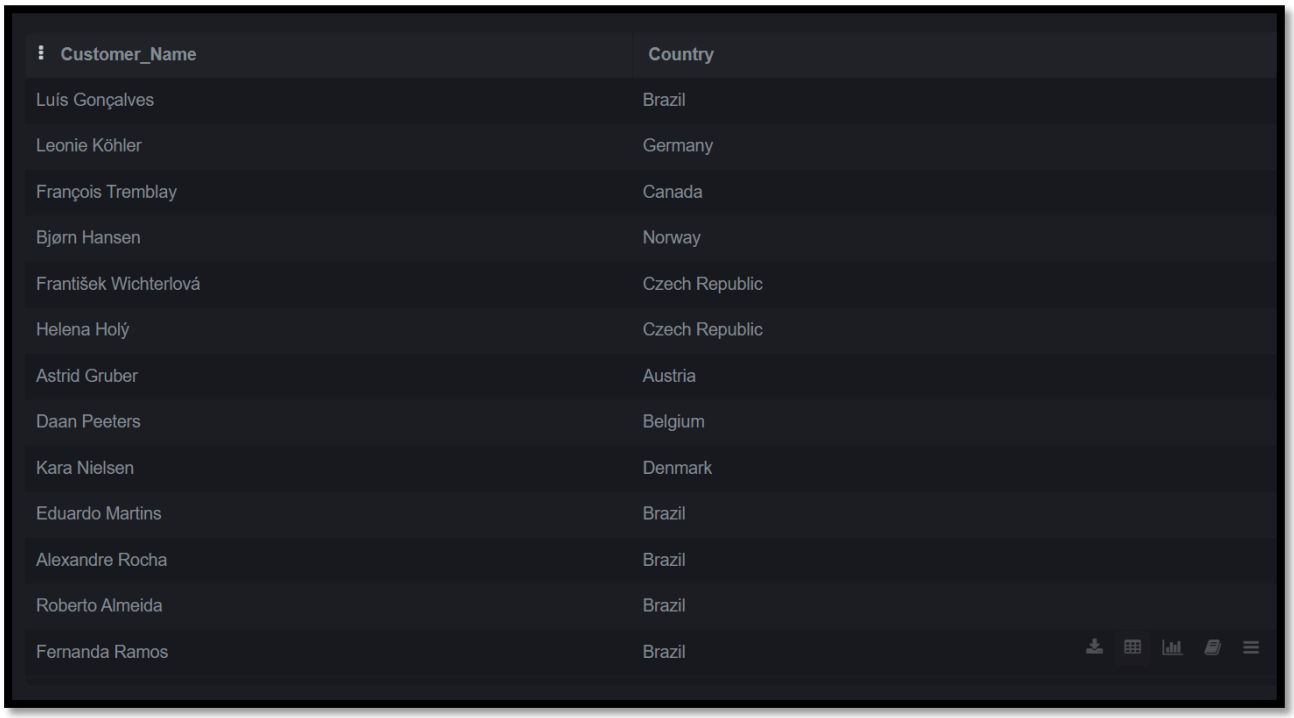
SQL Project: WSDA Music Store Dataset (Sample Outputs)


```
-- 1. List the names of all customers along with the country they belong
to.


SELECT
firstname ||' '|| lastname as Customer_Name,
country
FROM
Customer


-- Results:
```



Customer_Name	Country
Luís Gonçalves	Brazil
Leonie Köhler	Germany
François Tremblay	Canada
Bjørn Hansen	Norway
František Wichterlová	Czech Republic
Helena Holý	Czech Republic
Astrid Gruber	Austria
Daan Peeters	Belgium
Kara Nielsen	Denmark
Eduardo Martins	Brazil
Alexandre Rocha	Brazil
Roberto Almeida	Brazil
Fernanda Ramos	Brazil


```
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------


-- 2. List all playlists along with how many tracks each contains.


SELECT
p.Name,
COUNT(pt.TrackId) as Tracks
FROM
Playlist as p
JOIN
PlaylistTrack as pt
ON
p.PlaylistId = pt.PlaylistId
GROUP by
p.Name
ORDER by
Tracks desc


-- Results:
```

Name	Tracks
Music	6580
90's Music	1477
TV Shows	426
Classical	75
Brazilian Music	39
Heavy Metal Classic	26
Classical 101 - The Basics	25
Classical 101 - Next Steps	25
Classical 101 - Deep Cuts	25
Grunge	15
On-The-Go 1	1
Music Videos	1

```
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```


-- 3. Find the top 5 customers who spent the most money.

```
SELECT
c.firstname ||' '|| c.lastname as Customer_Name,
sum(I.total) as Total_Spent
FROM
Customer as C
JOIN
Invoice as I
ON
c.CustomerId = I.CustomerId
GROUP BY
c.CustomerId
ORDER BY
Total_Spent DESC
limit 5
```

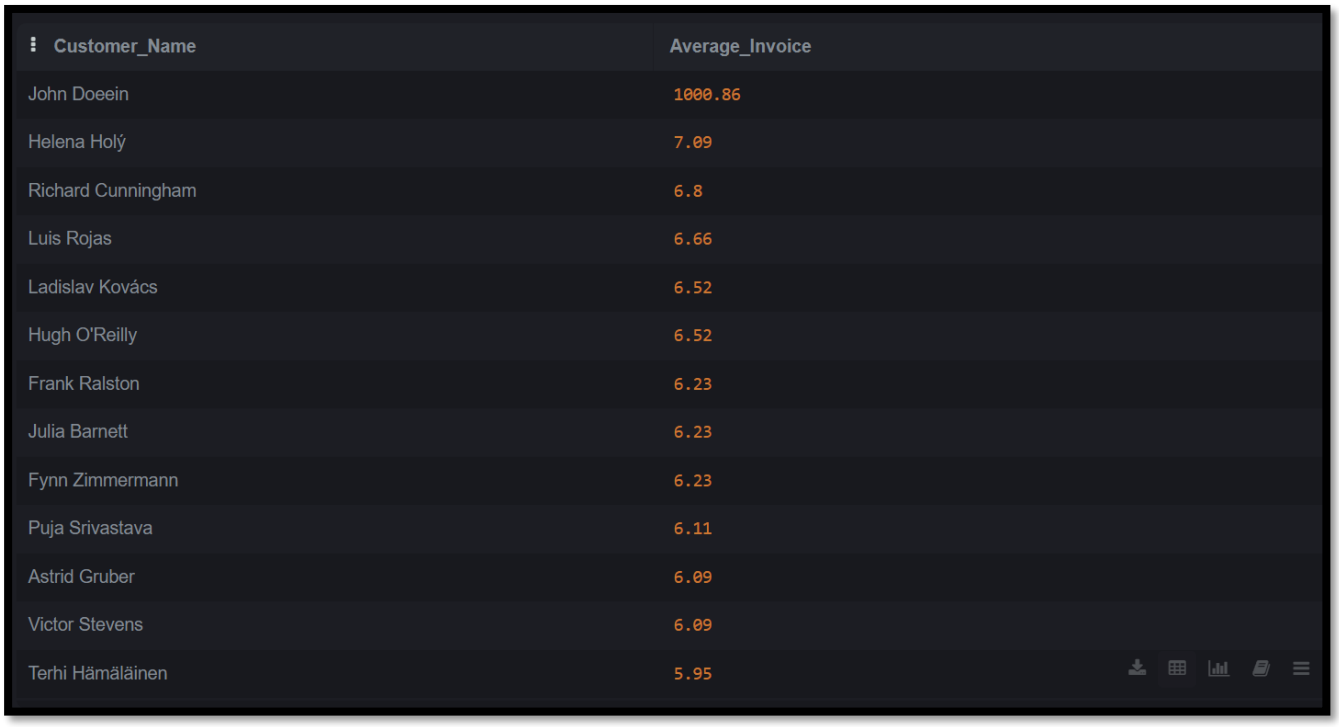
-- Results:

Customer_Name	Total_Spent
John Doeein	1000.86
Helena Holý	49.62
Richard Cunningham	47.62
Luis Rojas	46.62
Ladislav Kovács	45.62

-- 4. Find the average invoice total per customer.

```
SELECT
c.FirstName ||' '||c.LastName AS Customer_Name,
Round(Avg(i.Total),2) as Average_Invoice
FROM
Customer as c
JOIN
Invoice as I
on
c.CustomerId = I.CustomerId
GROUP BY
c.CustomerId
ORDER BY
Average_Invoice DESC
```

-- Results:



Customer_Name	Average_Invoice
John Doeein	1000.86
Helena Holý	7.09
Richard Cunningham	6.8
Luis Rojas	6.66
Ladislav Kovács	6.52
Hugh O'Reilly	6.52
Frank Ralston	6.23
Julia Barnett	6.23
Fynn Zimmermann	6.23
Puja Srivastava	6.11
Astrid Gruber	6.09
Victor Stevens	6.09
Terhi Hämäläinen	5.95


```
-------------------------------------------------------------------------
-------------------------------------------------------------------------

/* 5. Write a SQL query that selects track names, composers, and unit
prices, and categorizes each track based on its price.
Price Categories:
Budget: Tracks priced at $0.99 or less
Regular: Tracks priced between $1.00 and $1.49
Premium: Tracks priced between $1.50 and $1.99
Exclusive: Tracks priced above $1.99 */

SELECT
name as Track_Name,
composer,
unitprice as Price,
CASE
WHEN unitprice <=.99 THEN 'Budget'
WHEN unitprice BETWEEN 1 and 1.49 THEN 'Regular'
WHEN unitprice BETWEEN 1.50 and 1.99 THEN 'Premium'
Else 'Exclusive'
END as Price_Category
FROM
Track
ORDER BY
Price DESC

-- Results:
```

Track_Name	Composer	Price	Price_Category
Zaneta Ricci	Suricata suricatta	1.99	Premium
Enterprisewide	Bosco Sipes and Mosciski	1.99	Premium
Solarbreeze	Canis mesomelas	1.99	Premium
Germain Mergue	Canis mesomelas	1.99	Premium
process improvement	HarveSchoen and Keebler	1.99	Premium
Cookley	Tadorna tadorna	1.99	Premium
Harri Bedlington	Tadorna tadorna	1.99	Premium
Realigned	Marvin Kirlin and Thiel	1.99	Premium
Zamit	Laniarius ferrugineus	1.99	Premium
Garold Doumenc	Laniarius ferrugineus	1.99	Premium
Ronstring	Haematopus ater	1.99	Premium
Conny Maylam	Haematopus ater	1.99	Premium
radical	MacGyv Gleichner and Marks	1.99	Premium

```
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------


-- 6. Total revenue generated each month (Month-Year format).


SELECT
strftime('%Y-%m', invoicedate) as Year_Month,
sum(total) as Total_Revenue
FROM
Invoice
GROUP BY
strftime('%Y-%m', invoicedate)
ORDER BY
Year_Month


-- Results:
```

Year_Month	Total_Revenue
2009-01	35.64
2009-02	37.62
2009-03	37.62
2009-04	37.62
2009-05	37.62
2009-06	37.62
2009-07	37.62
2009-08	37.62
2009-09	37.62
2009-10	37.62
2009-11	37.62
2009-12	37.62
2010-01	52.62

```
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
```