



[Return to "Data Foundations" in the classroom](#)

Music SQL Database

REVIEW

CODE REVIEW 4

HISTORY

▼ SQL Queries - Project 3 - Hari Sankaran Nair.txt 4

```
1 /* Query 1 - query used for first insight*/
2
3 SELECT
4     G.Name Genre,
5     G.GenreId ID,
6     SUM(IL.Unitprice * IL.Quantity) Sales_Value
7 FROM InvoiceLine IL
8 JOIN Track T
9     ON IL.TrackId = T.TrackId
10 JOIN Genre G
11     ON T.GenreId = G.GenreId
12 GROUP BY 1,
13          2
14 ORDER BY 3 DESC;
```



AWESOME

Good use of JOINS and aggregations

```
15
16 /* Query 2 - query used for second insight*/
17
18 SELECT
19     I.BillingCountry Country,
20     SUM(IL.UnitPrice * IL.Quantity) Rock_Sales,
21     G.Name Genre
```

```

21 FROM Invoice I
22 JOIN InvoiceLine IL
23   ON I.InvoiceId = IL.InvoiceId
24 JOIN Track T
25   ON IL.TrackId = T.TrackId
26 JOIN Genre G
27   ON T.GenreId = G.GenreId
28 WHERE G.Name = 'Rock'
29 GROUP BY 1,
30

```



AWESOME

Nice touch using aliases and grouping by column number instead of name. This makes the query run faster

```

31      3
32 ORDER BY 2 DESC;
33
34 /* Query 3 - query used for third insight*/
35
36 SELECT
37   C.FirstName,
38   C.LastName,
39

```



SUGGESTION

Just a little suggestion, you can use `||` to concatenate the Customer's first and last names, the code is be

```
SELECT c.FirstName || ' ' || c.LastName as "Customer Name"
```

```

39   C.Country,
40   COUNT(IL.InvoiceID) Num_songs,
41   G.Name Genre
42 FROM Customer C
43 JOIN Invoice I
44   ON C.CustomerId = I.CustomerId
45 JOIN InvoiceLine IL
46   ON I.InvoiceId = IL.InvoiceId
47 JOIN Track T
48   ON IL.TrackId = T.TrackId
49 JOIN Genre G
50   ON T.GenreId = G.GenreId
51 WHERE G.Name = 'Rock'
52 GROUP BY 1,
53          2,
54          3,
55          5
56 ORDER BY 4 DESC;
57
58
59 /* Query 4 - queries used for fourth insight*/
60
61 /* 2 Queries were used to determine the total sales per agent and the sales of rock s
62
63 /* Query 4a (Total Sales per Agent)*/
64

```

```
65 SELECT
66     E.FirstName,
67     E.LastName,
68     SUM(IL.UnitPrice * IL.Quantity) Total_Sales
69 FROM Employee E
70 JOIN Customer C
71     ON E.EmployeeId = C.SupportRepId
72 JOIN Invoice I
73     ON C.CustomerId = I.CustomerId
74 JOIN InvoiceLine IL
75     ON I.InvoiceId = IL.InvoiceId
76 JOIN Track T
77     ON IL.TrackId = T.TrackId
78 JOIN Genre G
79     ON T.GenreId = G.GenreId
80 GROUP BY 1,
81           2
82 ORDER BY 3 DESC;
83
84 /* Query 4b (Total Rock Songs Sales per Agent)*/
85
86 SELECT
87     E.FirstName,
88     E.LastName,
89     SUM(IL.UnitPrice * IL.Quantity) Total_Rock_Sales
90 FROM Employee E
91 JOIN Customer C
92     ON E.EmployeeId = C.SupportRepId
93 JOIN Invoice I
94     ON C.CustomerId = I.CustomerId
95 JOIN InvoiceLine IL
96     ON I.InvoiceId = IL.InvoiceId
97 JOIN Track T
98     ON IL.TrackId = T.TrackId
99 JOIN Genre G
100     ON T.GenreId = G.GenreId
101 WHERE G.Name = 'Rock'
102 GROUP BY 1,
```



AWESOME

Excellent query formatting

```
103         2
104 ORDER BY 3 DESC;
105
106
107
```

RETURN TO PATH

Rate this review
