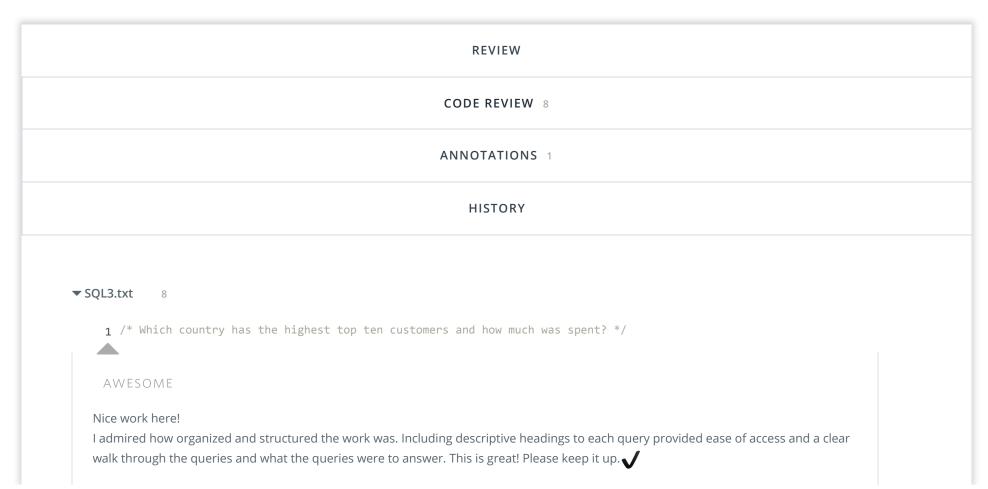
12/23/2018 Udacity Reviews



< Back to Data Foundations

Music SQL Database



12/23/2018 Udacity Reviews

Why we should comment code.

3 SELECT Customer.Country, SUM(Invoice.Total) AS Total_amount_spent



AWESOME

Bravo!

There is a great consistency I noticed in the provided queries. The good practice of selecting only columns that are relevant to the analysis to be made. This is excellent work due to query run times.

SQL Tutorial: How To Write Better Queries

- 4 FROM customer
 5 JOIN Invoice
 6 ON Customer.CustomerId = Invoice.CustomerId
 7 JOIN InvoiceLine
 8 ON InvoiceLine.InvoiceId = Invoice.InvoiceId
 9 GROUP BY country
 10 ORDER BY Total_amount_spent DESC
 11 LIMIT 10;

AWESOME

Nice work!

Limiting the records to 10 rows, this made the visualizations very clear and visually appealing. This is awesome!

```
12
13
14
15 /* What is the album name that sold most in USA, in which month and how many? */
16
17 SELECT Strftime('%m', Invoice.InvoiceDate) as Month, Album.Title as Album_name, COUNT(*) Number_sold, SUM(Invoice.Total)
18 FROM Invoice
19 JOIN InvoiceLine
```

AWESOME

JOINS and aggregations are properly included in all the queries as required. Nicely done!

```
20 ON InvoiceLine.InvoiceId = Invoice.InvoiceId
21 JOIN Album
22 On Album.AlbumId = Track.AlbumId
23 JOIN Track
24 ON Track.TrackId = InvoiceLine.TrackId
25 WHERE BillingCountry = "USA"
26 GROUP BY Month
27 ORDER BY Number_sold DESC;
```



12/23/2018

SUGGESTION

This is good way of using group by and order by with column names. Also, I think another great way of doing so is by using just the column numbers. This way seems to be more efficient as it reduces query run times. Speed is a great factor of queries. 🛴

```
28
29
30
31 /*Which top 5 tracks is the highest sold in the album of song in the United Kingdom? */
32 SELECT Track.Name as Track name, Album.Title as Album title, (count(Quantity) * sum(InvoiceLine.UnitPrice)) as Unit sold,
33 FROM Track
34 JOIN Album
35 ON Album.AlbumId = Track.AlbumId
36 JOIN InvoiceLine
37 On InvoiceLine.TrackId = Track.TrackId
38 Join Invoice
39 On Invoice.InvoiceId = InvoiceLine.InvoiceId
40 WHERE BillingCountry = 'United Kingdom'
41 GROUP BY Track_name
42 ORDER BY Total DESC
43 LIMIT 5;
```

AWESOME

Good job always using; to terminate a query. This is a good practice. It makes your work very professional. You did great with the queries.

The Most Effective Way to Write Effective SQL: Change Your Thinking Style

44

45

12/23/2018

```
Udacity Reviews
 46 /*In how many states does each support agent have the highest sales?*/
 47
 48 WITH t1 as (SELECT e.EmployeeId, e.LastName lastname, e.FirstName firstname, c.state as State, SUM(i.Total) totalsales
                FROM Employee e
 49
                JOIN Customer c
 50
                ON e.EmployeeId = c.SupportRepId
 51
                JOIN Invoice i
 52
                ON c.CustomerId = i.CustomerId
 53
                GROUP BY EmployeeId, 4),
 54
 55
         t2 as (SELECT State, MAX(totalsales) totalsales
 56
                FROM t1
 57
                GROUP BY State)
 58
 59
 60 SELECT (t1.firstname || " "|| t1.lastname) as SalesSupportRep, COUNT(t1.State) NumberOfStates, SUM(t1.totalsales) TotalSal
 AWESOME
Good job using II to concatenate first and last name into salesuportRep. This is good skill. There is another great way of doing this
```

using the concat function. Consider giving it a look at your free time.

```
61 FROM t1
62 JOIN t2
63 ON t1.State = t2.State AND t1.totalsales = t2.totalsales
64 GROUP BY EmployeeId
65 ORDER BY TotalSales DESC;
```



AWESOME

Brilliant work with the provided queries in this submission!

They are all well structured, neatly presented, simple to understand yet very insightful. Well done!



12/23/2018 Udacity Reviews

RETURN TO PATH