UDACITY

Return to "Data Foundations" in the classroom

Music SQL Database

REVIEW

CODE REVIEW 5

HISTORY

▼ ONYEKABA NZUBECHUKWU JUDE - SQL QUERY.txt 5

```
1  REFRENCES: N/A
2
3
4  1. /*TOP 10 ALBUMS IN 2010*/
```

AWESOME

Good usage of comments! 👏🏼
Comments markers are great to set headers or to explain something tricky in some logic.

```
5
6  SELECT
7    ar.Name Artist,
8    al.Title AlbumTitle,
9    al.AlbumId AlbumId,
10   inv.InvoiceDate DatePurchased,
11   SUM (inl.Quantity * inl.UnitPrice) AlbumSales
```

AWESOME

Well done! You have identified the proper relationship of the tables to perform the calculation of the t and do the sum only for unitPrice. You did very well!

```
12 FROM Artist ar
13 JOIN Album al
14   ON ar.ArtistId = al.ArtistId
15 JOIN Track tr
```

```
  ON al.AlbumId = tr.AlbumId
JOIN InvoiceLine inl
  ON tr.TrackId = inl.TrackId
JOIN Invoice inv
  ON inl.InvoiceId = inv.InvoiceId
WHERE inv.InvoiceDate BETWEEN '2010-01-01' AND '2011-01-01'
GROUP BY AlbumTitle
ORDER BY AlbumSales DESC
LIMIT 10;



2. /*Total Amount Spent based on country.*/

SELECT
SUM(total) TotalAmountSpent,
country Country
FROM customer cu,
```

AWESOME

Well done. Whenever you use more than one table you can give it aliases to make ease and faster the

```
SELECT fact.revenue, dim.name
FROM FACT_COST fact
JOIN CLIENT_DIMENSION dim
ON fact.id = dim.id
```

Note that we don't have to write the whole name of the tables in the SELECT and JOIN statements.
The usage of aliases is a very good practice.

```
Invoice inv
WHERE cu.customerid = inv.customerid
GROUP BY country
ORDER BY TotalAmountSpent DESC;


3./*Top 10 Genre based on highest Revenue.*/

SELECT
ge.Name GenreName,
SUM(Inl.UnitPrice) Revenue
FROM Genre ge
JOIN Track tr
ON tr.GenreId = ge.GenreId
JOIN InvoiceLine inl
ON inl.TrackId = tr.TrackId
GROUP BY ge.Name
```

AWESOME

You can also use numbers to point which column to perform the group.
It's very good for quick references, but if you want a good code for production, you can call the colum
It makes easier future maintenances.

```
ORDER BY Revenue DESC
LIMIT 10;

4
/*Popularity of Blues music by country.*/
```

```
56
57 SELECT
58   inv.BillingCountry Country,
59   COUNT(ge.Name) NumberOfTracks
60 FROM Invoice inv
61 JOIN InvoiceLine inl
62   ON inv.InvoiceId = inl.InvoiceId
63 JOIN Track tr
64   ON inl.TrackId = tr.TrackId
65 JOIN Genre ge
66   ON tr.GenreId = ge.GenreId
67 WHERE ge.Name = 'Blues'
```

AWESOME

Very professional work!
Joins, Filters, and Aggregations being used on a single query! 👏🏽

```
68 GROUP BY Country
69 ORDER BY NumberOfTracks DESC;
70
71
72
```

RETURN TO PATH