‹ Return to "Data Foundations" in the classroom

Music SQL Database

REVIEW

CODE REVIEW 4

ANNOTATIONS 1

HISTORY

SQL_Queries.txt 4

```
1 -- Insight #1 ---
2
3 SELECT i.BillingCountry, SUM(il.UnitPrice*il.Quantity) Total_Spent
4 FROM Invoice i
5 JOIN InvoiceLine il
```

```
  ON i.InvoiceId = il.InvoiceId
JOIN Track t
  ON il.TrackId = t.TrackId
JOIN Genre g
  ON t.GenreId = g.GenreId
WHERE g.Name = 'Rock'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10;
```

AWESOME

Solid logic here!

```


-- Insight #2 ---

SELECT g.Name, SUM(il.UnitPrice*il.Quantity) total_spent, i.InvoiceDate
FROM Invoice i
JOIN InvoiceLine il
  ON i.InvoiceId = il.InvoiceId
JOIN Track t
  ON il.TrackId = t.TrackId
JOIN Genre g
  ON t.GenreId = g.GenreId
WHERE i.InvoiceDate > '2012-12-31'
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;

```

AWESOME

Another flawless one here!

```

```

```sql
-- Insight #3 ---

SELECT g.name, SUM(il.UnitPrice*il.Quantity) total_spent,
     CASE WHEN sum(il.UnitPrice*il.Quantity) > 350 THEN 'Top Genre'
     WHEN sum(il.UnitPrice*il.Quantity) > 150 AND sum(il.UnitPrice*il.Quantity) < 350 THEN 'Middle Genre'
     ELSE 'Low Genre' END AS Genre_Category
FROM InvoiceLine il
JOIN Track t
  ON il.TrackId = t.TrackId
JOIN Genre g
  ON t.GenreId = g.GenreId
GROUP BY g.Name
ORDER BY 2 DESC
```

AWESOME

This is the kind of queries we were expecting to see. Fantastic us of conditionals to create segments!

```sql


-- Insight #4 ---

SELECT DISTINCT strftime('%Y', i.InvoiceDate) order_date, SUM(il.UnitPrice*il.Quantity) total_spent
FROM Invoice i
JOIN InvoiceLine il
  ON i.InvoiceId = il.InvoiceId
JOIN Track t
  ON il.TrackId = t.TrackId
JOIN Genre g
  ON t.GenreId = g.GenreId
WHERE i.BillingCountry = 'United Kingdom' AND g.name = 'Rock'
GROUP BY 1

```

AWESOME

Great use of strftime here!

```
61
62
63
```

RETURN TO PATH

Rate this review