Answering Business Questions using SQL

This project aims to analyze and answer key business questions about a fictional music record store called klook Music Players Inc (KMP)

Data set

Database :: music store database

Please download the .sql file from the link mentioned here and restore the database into your postgreSQL editor / pgAdmin

https://drive.google.com/drive/1J9XBNPqE2FmEjzm5iryQU-EJPQLDrroZ?export=download

This database includes tables on invoice information, track, album, artist and genre data, and employee and customer information related to the a music store's sales

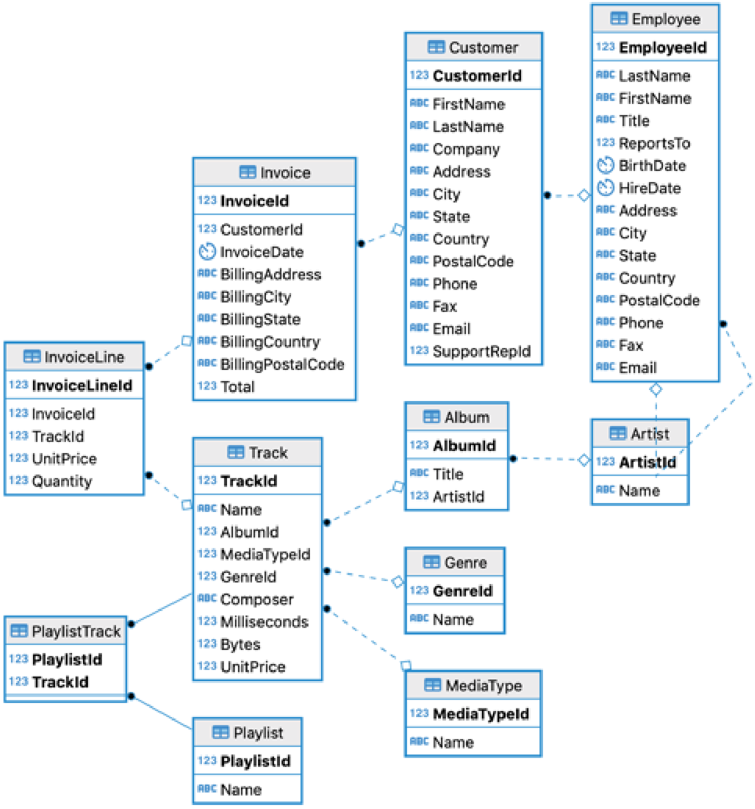
There are 11 tables in the music store database.

- **employee** table stores employees data such as employee id, last name, first name, etc. It also has a field named **ReportsTo** to specify who reports to whom.
- **customer** table has data related to customers, customers name company , email id, phone their service representative for any query resolution etc.
- **invoice & invoiceLine** tables: these two tables store invoice data.The invoices table stores invoice header data and the invoice_items table stores the invoice line items data. You can check on total sales or billing related analysis here
- **artist** table stores artists data. It is a simple table that contains only the artist id and name of the music artists.
- **album** table stores data about a list of tracks. Each album belongs to one artist. However, one artist may have multiple albums.
- **media_type** table stores media types such as MPEG audio and AAC audio files.
- **genre** table stores music types such as rock, jazz, metal,classical etc.
- **track** table stores the data of songs, each of the tracks belongs to one album.
- **playlist & playlist_track** tables: playlists table store data about playlists. Each playlist contains a list of tracks. Each track may belong to multiple playlists. The relationship

between the playlists table and tracks table is many-to-many. The playlist_track table is used to reflect this relationship.

Schema

You can refer the below schema to understand the various relationships and keys

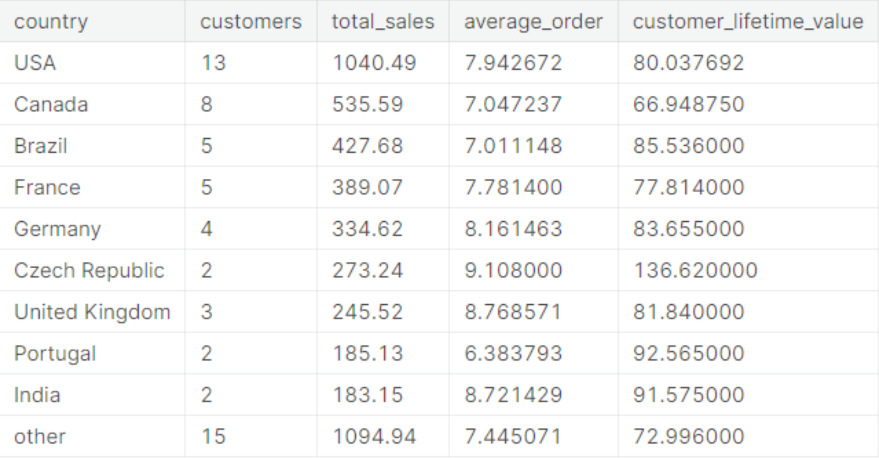


Concepts Covered

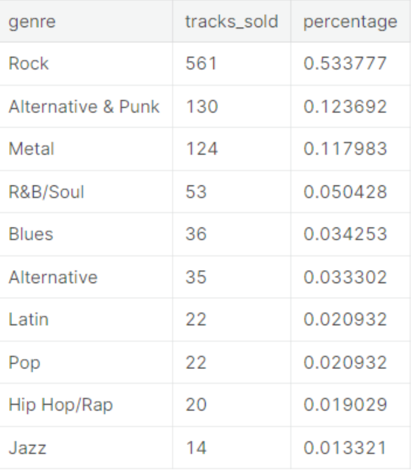
CONCAT, WITH,SUBQUERY, SELF JOIN, AVERAGE, AGGREGATION GROUP BY,CASE WHEN , CASTING , HAVING, INNER AND LEFT JOINS

Business Questions

1. Order of employees by date they joined? Name the person who joined recently **[Beginners]**
2. Who is the oldest employee in the firm? **[Beginners]**
3. What is the average duration of songs in minutes? **[Beginners]**

4. Who is the customer that has spent the most on music for each country? **[Intermediate]**
5. Which artist has the longest songs? **[Beginners]**
6. Find the total dollar amount of sales assigned to each sales support agent within the company **[Intermediate]**
7. Write a SQL query to prepare the following summary - (country,customers count, total sales amount, average order value and customer lifetime value **[Pro]**
   a. 

country	customers	total_sales	average_order	customer_lifetime_value
USA	13	1040.49	7.942672	80.037692
Canada	8	535.59	7.047237	66.948750
Brazil	5	427.68	7.011148	85.536000
France	5	389.07	7.781400	77.814000
Germany	4	334.62	8.161463	83.655000
Czech Republic	2	273.24	9.108000	136.620000
United Kingdom	3	245.52	8.768571	81.840000
Portugal	2	185.13	6.383793	92.565000
India	2	183.15	8.721429	91.575000
other	15	1094.94	7.445071	72.996000

   b. Customer lifetime value CLTV : Customer lifetime value is the total worth to a business of a customer over the whole period of their relationship, here you can use total sales per customers in their whole lifespan of buying tracks from the KLOOK store
8. Which genres sell the most tracks in the USA, return the following table- Top 10 genres, tracks sold and their percentage contribution in the market.**[Intermediate]**
   a. 

genre	tracks_sold	percentage
Rock	561	0.533777
Alternative & Punk	130	0.123692
Metal	124	0.117983
R&B/Soul	53	0.050428
Blues	36	0.034253
Alternative	35	0.033302
Latin	22	0.020932
Pop	22	0.020932
Hip Hop/Rap	20	0.019029
Jazz	14	0.013321

Evaluation Rubric

Rubric Description	Poor	Fair	Good
How Correct is the Output	Queries do not work correctly, output is wrong or there is no output	Query structured correctly - logically but there is few small error which were not handled	All queries work correctly
SQL Query Structure and logic building	Most queries are in long lines, new lines aren't used properly (e.g. SELECT queries are not broken by SELECT, FROM, WHERE, etc.)	Query structure needs work	Query structured logically aligned with Business Knowledge
How closely is the naming convention are followed	Names are too short or too cryptic, inconsistent	Names follow the conventions in most part, but need work	Alias / names are easy to understand, consistent
Correctness of Joins, Filters and Conditional Statements	Incorrect usage of concepts	Joins,Filters are correct but need some more work	All joins, filter and conditions are placed correctly
Using Advanced SQL Concepts (With, Sub query)	With, Sub query are not used and output is not coming correct	With, Sub query are not used but output correct Need to improve on this to make query more efficient	With, Sub query are used properly and correct output

Marks Matrix

For all correct responses maximum marks is as given below

Level	Max Marks	Poor Contr of Max Marks	Fair contr of Max Marks
Beeginers	10	20%	60%
Intermediate	15	20%	60%
Pro	25	20%	60%

Answers & Explanations ()

1. Order of employees by date they joined
   a.
   ```
   select (last_name || ' '|| first_name)as employee_name,
   date(hire_date) as hiredate
   from employee
   order by hire_date asc
   ```
2. Who is the oldest employee in the firm?
   a.
   ```
   select (last_name || ' '|| first_name)as
   employee_name,date(birthdate) as birthdate
   from employee
   order by birthdate asc
   ```

3. What is the average duration of songs in minutes?
    a.
    ```
    select round(avg(milliseconds)/60000,2) as Average_Duration
    from track limit 3
    ```

4. Who is the customer that has spent the most on music for each country?
    a.
    ```
    with t1 as
    (
    select cu.Country as Countr, cu.Customer_Id as _Id
    , cu.first_name as F_name, cu.Last_Name as L_name,
     sum(i.total) as tm
    from Invoice i
    JOIN customer cu
    on cu.Customer_Id = i.Customer_Id
    group by 1,2 , 3,4
    order by 5),
    t2 as
    (select countr  , max(tm) as mm from t1
    group by 1 )
    select t1.countr as Country , t1.tm as Totalexpense ,
    t1.f_name || ' ' || t1.L_name as Full_name  , t1._Id as
    Customer_Id
    from t1
    join t2
    on t1.countr = t2.countr and t1.tm = t2.mm
    order by 2 desc
    ```

5. Which artist has the longest songs
    a.
    ```
    select composer, round(avg(milliseconds)/60000,2) as
    Average_Duration
    from track
    where composer is not null
    group by composer
    having count(name)>1
    order by Average_Duration desc limit 3
    ```

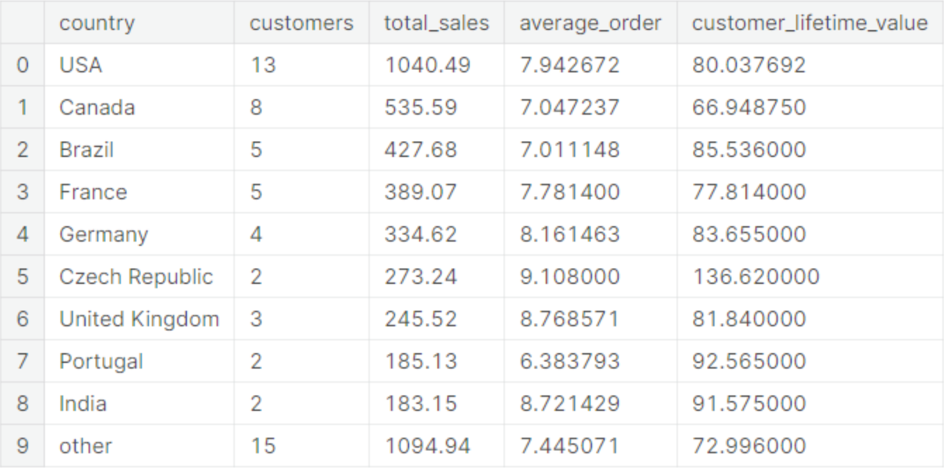
6. Analyzing Employee Sales Performance
    Each customer for the store gets assigned to a sales support agent within the company when they first make a purchase. analyze the purchases of customers belonging to each employee to see if any sales support agent is performing either better or worse than the others

find the total dollar amount of sales assigned to each sales support agent within the company

a.

```
Select e.first_name || '' || e.last_name employee,
e.hire_date,
SUM(inv.total) total_sales from employee e
left join customer c
ON c.support_rep_id = e.employee_id
Inner Join invoice inv
On inv.customer_id = c.customer_id::int
Group by 1,2
```

7. calculate data, for each country, on the:
   a. total number of customers
   b. total value of sales
   c. average value of sales per customer
   d. average order value
   e. Expected Output



	country	customers	total_sales	average_order	customer_lifetime_value
0	USA	13	1040.49	7.942672	80.037692
1	Canada	8	535.59	7.047237	66.948750
2	Brazil	5	427.68	7.011148	85.536000
3	France	5	389.07	7.781400	77.814000
4	Germany	4	334.62	8.161463	83.655000
5	Czech Republic	2	273.24	9.108000	136.620000
6	United Kingdom	3	245.52	8.768571	81.840000
7	Portugal	2	185.13	6.383793	92.565000
8	India	2	183.15	8.721429	91.575000
9	other	15	1094.94	7.445071	72.996000

   f.
   g. Answer
   h.

```
With purchase_data As (
       Select
       c.country,
       COUNT(Distinct(inv.customer_id)) customers,
       SUM(inv.total) total_sales,
      SUM(inv.total)/COUNT(inv.customer_id) average_order,
      SUM(inv.total)/COUNT(Distinct(inv.customer_id))
customer_lifetime_value
       From customer c
       Inner Join invoice inv
       ON c.customer_id = inv.customer_id
       Group BY 1
       Order By 3 Desc),

      purchases As (
       Select
      CASE
       when customers = 1 Then 'other'
       Else country
       End As country,
       customers,
       total_sales,
       average_order,
      customer_lifetime_value
      from purchase_data),

      updated As(
      Select country,Sum(customers) customers,
Sum(total_sales) total_sales,
       Sum(average_order)/Count(average_order) average_order,

Sum(customer_lifetime_value)/count(customer_lifetime_value)
customer_lifetime_value
       from purchases
       Group by 1
       Order BY 3 Desc),

      final as (Select *,
      Case When country = 'other' Then 1
       Else 0
       End as ord
       From updated order by ord)

       Select country,customers,total_sales,average_order,
       customer_lifetime_value
       From final
```

8. Which genres sell the most tracks in the USA, return the table given in the question- Top 10 genres, tracks sold and their percentage contribution in the market

```
WITH
        genre_track_sold AS
```

```
            (
            SELECT
                g.name genre,
                il.quantity,
                il.invoice_id
            FROM genre g
            INNER JOIN track t ON g.genre_id = t.genre_id
            INNER JOIN invoice_line il ON t.track_id =
il.track_id
            ),

        sold_USA AS
            (
            SELECT
                gts.genre,
                gts.quantity,
                c.country
            FROM genre_track_sold gts
            INNER JOIN invoice i ON i.invoice_id =
gts.invoice_id
            INNER JOIN customer c ON c.customer_id =
i.customer_id
            WHERE country = 'USA'
            )

    SELECT
        genre,
        SUM(quantity) tracks_sold,
        CAST(SUM(quantity) as float) / (SELECT COUNT(*) FROM
sold_USA) percentage
    FROM sold_USA
    GROUP BY 1
    ORDER BY 2 DESC
    LIMIT 10;
```

9.