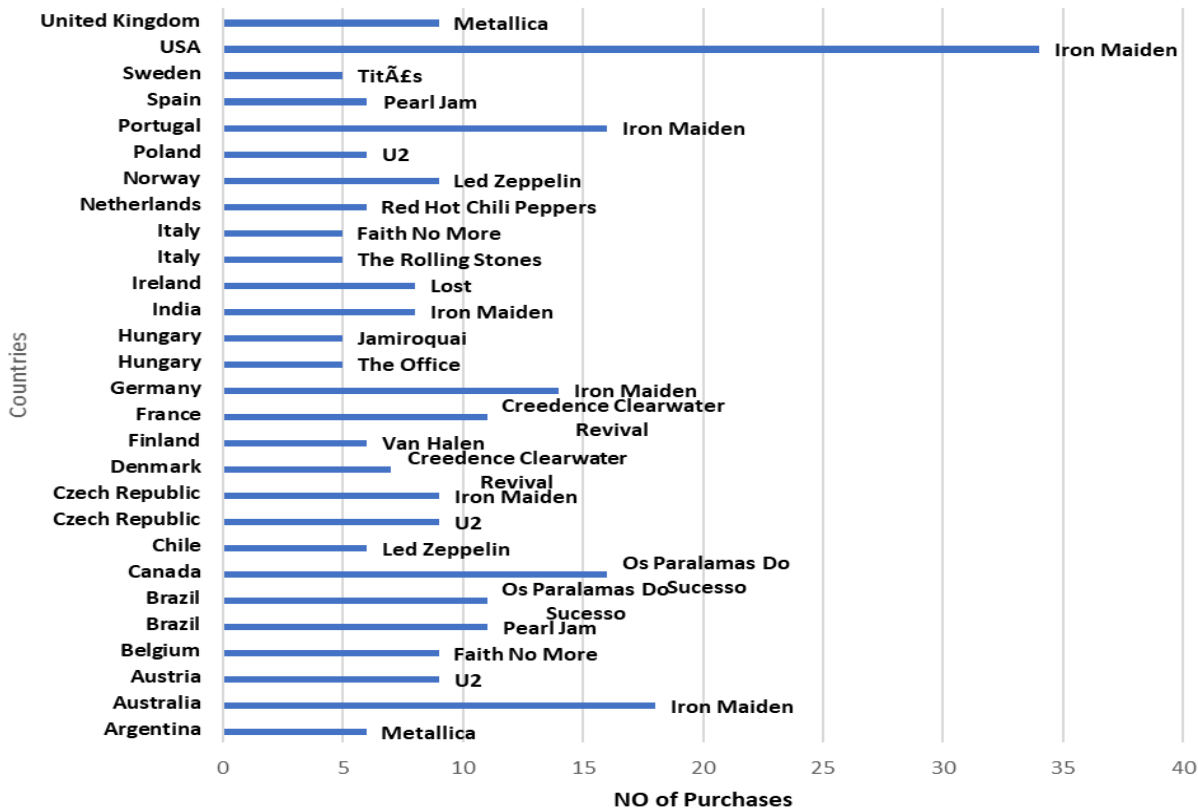


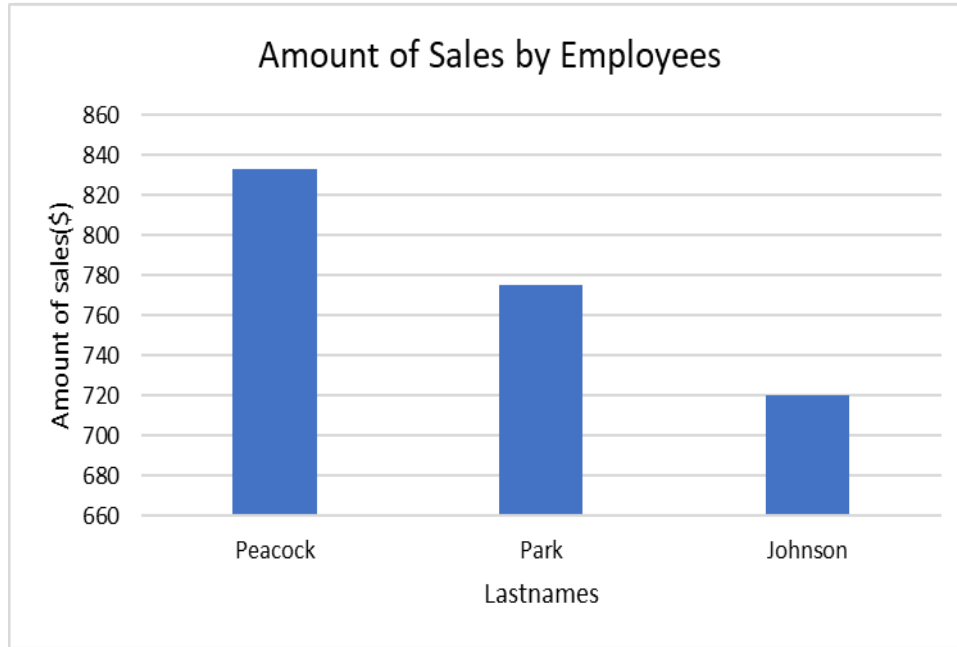
Most Popular Artist Across Countries

Most Popular Artist by Countries



- This graph shows the artist with the highest number of purchases across countries.
- From the graph we can see that the artist with the highest purchase is Iron Maiden in USA.
- While iron maiden is popular in Australia, Czech Republic, Germany and Portugal. It is a much bigger artist in the USA and a higher level of success compared to other countries.

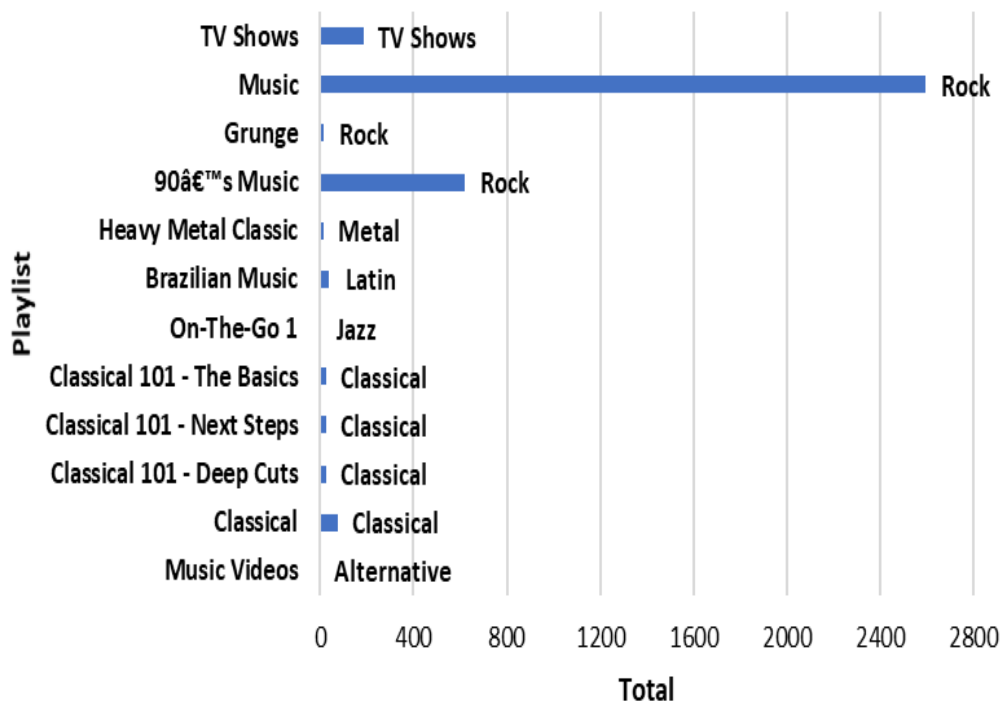
Amount of sales by employees



- This shows the amount of sales each employee made.
- Peacock has the highest amount of sales with \$833.04 and park having \$755.4 and Johnson \$720.16

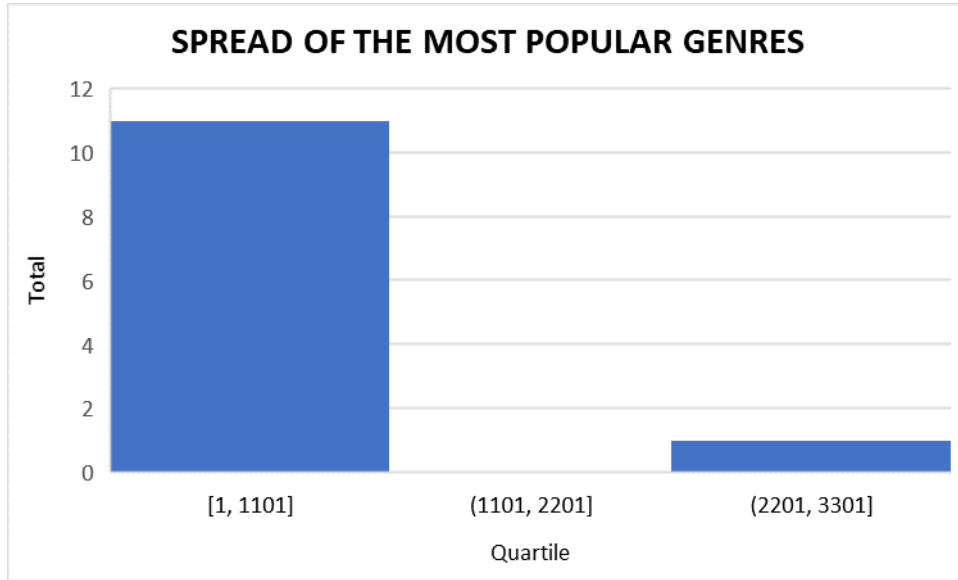
Most Popular Genre In Each Playlist

MOST POPULAR GENRE IN EACH PLAYLIST



- This graph shows the most popular genre in each playlist.
- The Music playlist has the highest count of Rock genre and also the largest count. While the Grunge and 90's music also have a high count of the rock in their respective playlist it is comparatively less than the music genre

Spread Of The Most Popular Artist In Each Genre



- This histogram show the spread of data across the most popular genres in the playlist.
- From the graph we can see that it is right skewed or positively skewed.

SQL code for popular artist across countries

```
WITH popular_Artist AS (  
  
  SELECT c.Country as countries, COUNT(ar.name) no_of_purchase, ar.Name as artists,  
         RANK() OVER(PARTITION BY c.Country ORDER BY COUNT(ar.Name) DESC) AS rank  
  FROM Genre as g  
  JOIN Track as t  
  ON g.GenreId = t.GenreId  
  JOIN InvoiceLine as il  
  ON t.TrackId = il.TrackId  
  JOIN Invoice as i  
  ON i.InvoiceId = il.InvoiceId  
  JOIN Customer as c  
  ON c.CustomerId = i.CustomerId  
  JOIN Album as a  
  ON a.AlbumId = t.AlbumId  
  JOIN Artist as ar  
  ON ar.ArtistId = a.ArtistId  
  GROUP BY c.Country, ar.Name  
  ORDER BY c.Country, no_of_purchase DESC )  
  
  SELECT Countries, no_of_purchase, artists,  
         CASE WHEN no_of_purchase >=30 THEN 'Big artist'  
         WHEN no_of_purchase <30 AND no_of_purchase >=10 THEN 'Popular'  
         ELSE 'Fairly popular'  
         END AS level  
  FROM popular_Artist  
  WHERE rank=1
```

SQL code for sales by employees

```
SELECT EmployeeId,e.LastName, sum(total) as totals  
FROM Employee as e  
JOIN Customer as c  
ON e.EmployeeId = c.SupportRepId  
JOIN Invoice as i  
ON c.CustomerId = i.CustomerId  
GROUP BY 1
```

SQL Code For Popular Genres In Each Playlist

```
WITH genre_playlist as (  
  
  SELECT p.Name as playlist_name,g.Name as genres,count(p.Name) as totals,  
  rank() OVER(PARTITION BY p.Name ORDER BY count(p.Name) DESC) as ranks  
  FROM Playlist AS p  
  JOIN PlaylistTrack as pt  
  ON p.PlaylistId = pt.PlaylistId  
  JOIN Track as t  
  ON pt.TrackId = t.TrackId  
  JOIN Genre as g  
  ON g.GenreId = t.GenreId  
  GROUP BY 1,2)  
  
  SELECT playlist_name,genres,totals,  
  ntile(4) OVER(PARTITION BY genres) as quartiles  
  FROM genre_playlist  
  WHERE ranks = 1
```

SQL code about the spread of popular artist in each genre

```
WITH genre_playlist as (  
  SELECT p.Name as playlist_name,g.Name as genres,count(p.Name) as totals,  
  rank() OVER(PARTITION BY p.Name ORDER BY count(p.Name) DESC) as ranks  
  FROM Playlist AS p  
  JOIN PlaylistTrack as pt  
  ON p.PlaylistId = pt.PlaylistId  
  JOIN Track as t  
  ON pt.TrackId = t.TrackId  
  JOIN Genre as g  
  ON g.GenreId = t.GenreId  
  GROUP BY 1,2)  
  
SELECT playlist_name,genres,totals  
FROM genre_playlist  
WHERE ranks = 1
```