Chinook Database

The Chinook data model represents a digital media store, including tables for artists, albums, media tracks, invoices and customers. Media related data was created using real data FROM an iTunes Library. customer and employee information was manually created using fictitious names, addresses that can be located on Google maps, and other well-formatted data (phone, fax, email, etc.). Sales information is auto-generated using random data for a four year period.

You can access the Chinook Database using pgadmin:

http://delivery-pgadmin.multiverse.io/
Username: pgadmin4@docker
Password: pgadmin4

-- 1. Provide a query showing customers (just their full names, customer ID and country) who are not in the US.

```
SELECT customer_id, first_name, last_name, country
FROM customer
WHERE not country = 'USA';
```

-- 2. Provide a query only showing the customers FROM Brazil.

```
SELECT * FROM customer
WHERE country = 'Brazil';
```

-- 3. Provide a query showing the invoices of customers who are FROM Brazil. The resultant table should show the customer's full name, invoice ID, Date of the invoice and billing country.

```
SELECT c.first_name, c.last_name, i.invoice_id, i.invoice_date, i.billing_country
FROM customer as c, invoice as i
WHERE c.country = 'Brazil' and
c.customer_id = i.customer_id;
```

-- 4. Provide a query showing only the employees who are Sales Agents.

```
SELECT * FROM employee
WHERE employee.title = 'Sales Support Agent';
```

-- 5. Provide a query showing a unique list of billing countries FROM the invoice table.

```
SELECT distinct billing_country FROM invoice;
```

-- 6. Provide a query showing the invoices of customers who are FROM Brazil.

```
SELECT *
FROM customer as c, invoice as i
WHERE c.country = 'Brazil' and
c.customer_id = i.customer_id;
```

-- 7. Provide a query that shows the invoices associated with each sales agent. The resultant table should include the Sales Agent's full name.

```
SELECT e.first_name, e.last_name, i.invoice_id, i.customer_id, i.invoice_date, i.billing_address, i.billing_country,
i.billing_postal_code, i.total
FROM customer as c
JOIN invoice as i
on c.customer_id = i.customer_id
JOIN employee as e
on e.employee_id = c.support_rep_id
ORDER BY e.employee_id;
```

-- 8. Provide a query that shows the invoice Total, customer name, Country and Sale Agent name for all invoices and customers.

```
SELECT e.first_name as employee_first, e.last_name as employee_last, c.first_name as customer_first,
c.last_name as customer_last, c.country, i.total
FROM employee as e
JOIN customer as c on e.employee_id = c.support_rep_id
JOIN invoice as i on c.customer_id = i.customer_id;
```

-- 9. How many invoices were there in 2009 and 2011? What are the respective total sales for each of those years?

```
SELECT COUNT(i.invoice_id), sum(i.total)
FROM invoice as i
WHERE i.invoice_date between date('2011-01-01 00:00:00') and date('2011-12-31 00:00:00');
```

-- 10. Looking at the invoice_line table, provide a query that COUNTs the number of line items for invoice ID 37.

```
SELECT COUNT(i.invoice_line_id)
FROM invoice_line as i
WHERE i.invoice_id = 37;
```

-- 11. Looking at the invoice_line table, provide a query that COUNTs the number of line items for each invoice. HINT: [GROUP BY]

```
SELECT invoice_id, COUNT(invoice_line_id)
FROM invoice_line
GROUP BY invoice_id;
```

-- 12. Provide a query that includes the track name with each invoice line item.

```
SELECT i.*, t.name
FROM invoice_line as i
JOIN track as t
on i.track_id = t.track_id;
```

-- 13. Provide a query that includes the purchased track name AND artist name with each invoice line item.

```
SELECT i.*, t.name as track, ar.name as artist
FROM invoice_line as i
        JOIN track as t on i.track_id = t.track_id
        JOIN album as al on al.album_id = t.album_id
        JOIN artist as ar on ar.artist_id = al.artist_id;
```

-- 14. Provide a query that shows the # of invoices per country. HINT: [GROUP BY]

```
SELECT billing_country, COUNT(billing_country) as number_of_invoices
```

```
FROM invoice
GROUP BY billing_country;
```

-- 15. Provide a query that shows the total number of tracks in each playlist. The Playlist name should be included on the resultant table.

```
SELECT COUNT(track_id) as number_of_tracks
FROM playlist_track
JOIN playlist
on playlist_track.playlist_id = playlist.playlist_id
GROUP BY playlist.playlist_id;
```

-- 16. Provide a query that shows all the Tracks, but displays no IDs. The result should include the Album name, Media type and Genre.

```
SELECT t.name, a.title as "Album Title", mt.name as "Media Type", g.name as "Genre"
FROM track t, album a, media_type mt, genre g
WHERE t.album_id = a.album_id
AND t.media_type_id = mt.media_type_id
AND t.genre_id = g.genre_id;
```

-- 17. Provide a query that shows all invoices but includes the # of invoice line items.

```
SELECT i.invoice_id, COUNT(il.invoice_line_id) as "# of invoice Line Items"
FROM invoice i
LEFT JOIN invoice_line il
ON i.invoice_id = il.invoice_id
GROUP BY i.invoice_id;
```

-- 18. Provide a query that shows the total sales made by each sales agent.

```
SELECT e.first_name || " " || e.last_name as "Sales Rep", SUM(i.Total) as "Total Sales"
FROM employee e, customer c, invoice i
WHERE e.employee_id = c.support_rep_id
AND c.customer_id = i.customer_id
GROUP BY e.first_name;

SELECT concat(e.first_name, ' ', e.last_name) as "Sales Rep", SUM(i.Total) as "Total Sales"
FROM employee e, customer c, invoice i
WHERE e.employee_id = c.support_rep_id
AND c.customer_id = i.customer_id
GROUP BY e.first_name, e.last_name;
```

--19. Which sales agent made the most in sales in 2009?

```
SELECT e.first_name || " " || e.last_name as "Sales Rep", SUM(i.Total) as "Total Sales"
FROM employee e, customer c, invoice i
WHERE e.employee_id = c.support_rep_id
AND c.customer_id = i.customer_id
AND i.invoice_date LIKE "2009%"
GROUP BY e.first_name || " " || e.last_name
ORDER BY SUM(i.Total) DESC
LIMIT 1;

SELECT concat(e.first_name, ' ', e.last_name) as "Sales Rep", SUM(i.Total) as "Total Sales"
FROM employee e, customer c, invoice i
WHERE e.employee_id = c.support_rep_id
```

```
AND c.customer_id = i.customer_id
AND EXTRACT('YEAR' FROM i.invoice_date) = '2009'
GROUP BY e.first_name, e.last_name
ORDER BY SUM(i.Total) DESC
LIMIT 1;


SELECT MAX(Total.Sales) as "Sales", Total.last_name
FROM(
        SELECT e.first_name, e.last_name, i.invoice_date, SUM(i.Total) as "Sales"
        FROM invoice i, employee e, customer c
        WHERE e.employee_id = c.support_rep_id
        AND i.invoice_date LIKE "2009%"
        AND c.customer_id = i.customer_id GROUP BY e.last_name) as Total;

SELECT MAX(Total.Sales) as Sales, Total.last_name
FROM(
        SELECT e.first_name, e.last_name, i.invoice_date, SUM(i.Total) as Sales
        FROM invoice i, employee e, customer c
        WHERE e.employee_id = c.support_rep_id
        AND EXTRACT('YEAR' FROM i.invoice_date) = 2009
        AND c.customer_id = i.customer_id
        GROUP BY e.first_name, e.last_name, i.invoice_date) as Total
GROUP BY Total.last_name;
```

--20. Which sales agent made the most in sales overall?
Hint: Use the MAX function on a subquery.

```
SELECT CONCAT(e.first_name, ' ', e.last_name) as "Sales Rep", SUM(i.Total) as "Total Sales"
FROM employee e, customer c, invoice i
WHERE e.employee_id = c.support_rep_id
AND c.customer_id = i.customer_id
GROUP BY 1
ORDER BY SUM(i.Total) DESC
LIMIT 1;
```

--21. Provide a query that shows the count of customers assigned to each sales agent.

```
SELECT CONCAT(e.first_name, ' ', e.last_name) as "Sales Rep", COUNT(c.support_rep_id) as "Num of customers"
FROM employee e, customer c
WHERE c.support_rep_id = e.employee_id
GROUP BY 1;
```

--22. Provide a query that shows the total sales per country.

```
SELECT distinct i.billing_country, COUNT(i.invoice_id) as "Number of Sales"
FROM invoice i
GROUP BY i.billing_country;
```

--23. Which country's customers spent the most?

```
SELECT distinct i.billing_country, SUM(i.Total) as "Total Sales"
FROM invoice i
GROUP BY i.billing_country
ORDER BY SUM(i.Total) DESC
```

LIMIT 1;

-- 24. Provide a query that shows the most purchased track of 2013?

```
SELECT t.Name, COUNT(il.invoice_line_id)
FROM Track t, invoice_line il, invoice i
WHERE t.Track_Id = il.Track_Id
AND il.invoice_id = i.invoice_id
AND EXTRACT('YEAR' FROM i.invoice_date) = 2013
GROUP BY t.Name
ORDER BY COUNT(il.invoice_line_id)
DESC Limit 1;
```

--25. Provide a query that shows the top 5 most purchased tracks over all?

```
SELECT t.Name, COUNT(il.invoice_line_id)
FROM Track t, invoice_line il
WHERE t.Track_Id = il.Track_Id
GROUP BY t.Name
ORDER BY COUNT(il.invoice_line_id) DESC
Limit 5;
```

--26. Provide a query that shows the top 3 best selling artists?

```
SELECT ar.Name, COUNT(il.Track_Id)
FROM Artist ar, Album al, Track t, invoice_line il
WHERE ar.Artist_Id = al.Artist_Id
AND al.album_id = t.album_id
AND t.Track_Id = il.Track_Id
GROUP BY ar.Name
ORDER BY COUNT(il.Track_Id) DESC
LIMIT 3;
```

--27. Provide a query that shows the most purchased Media Type?

```
SELECT mt.Name, COUNT(il.Track_Id)
FROM Media_Type mt, Track t, invoice_line il
WHERE mt.media_type_id = t.media_type_id
AND t.Track_Id = il.Track_Id
GROUP BY mt.Name
ORDER BY COUNT(il.Track_Id) DESC
LIMIT 1;
```