homework 4

GunGyeom James Kim

October 1, 2022

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

   Since there is no one field containing full names, I am *selecting firstName* and *lastName*

```
SELECT firstName, lastName, customerid, country
FROM customer
WHERE 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 customer.firstname, customer.lastname,
        invoice.invoiceid, invoice.billingcountry
    FROM customer
INNER JOIN invoice
    ON customer.customerid = invoice.customerid
WHERE customer.country = 'Brazil';
```

4. **Provide a query showing only the Employees who are Sales Agents.**

   Using *LIKE-String Pattern Matching* to find *Sales Agents.*

```
SELECT *
    FROM employee
WHERE title LIKE 'sales%agent';
```

5. **Provide a query showing a unique list of billing countries from the Invoice table.**

```
SELECT DISTINCT billingcountry
    FROM invoice;
```

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

   Dropping queries that does not have associated *sales agent* using *INNER JOIN*

```
    SELECT employee.firstname, employee.lastname, invoice.*
        FROM invoice
    INNER JOIN customer
        ON invoice.customerid = customer.customerid
    INNER JOIN employee
        ON employee.employeeid = customer.supportrepid
    WHERE employee.title LIKE 'sales%agent';
```

7. **Provide a query that shows the Invoice Total, Customer name, Country and Sale Agent name for all invoices and customers.**

   I wanted to *FULL OUTER JOIN invoice* and *customer* as I wanted to keep all the row, but *MySQL* does not support *FULL OUTER JOIN*, so I used *LEFT JOIN, RIGHT JOIN* and *UNION* to achieve identical output.

```
    SELECT invoice.total, customer.firstname, customer.lastname,
            customer.country, employee.firstname, employee.lastname
        FROM invoice
    LEFT JOIN customer
        ON invoice.customerid = customer.customerid
    INNER JOIN employee
        ON employee.employeeid = customer.supportrepid
    WHERE employee.title LIKE 'sales%agent'

    UNION

    SELECT invoice.total, customer.firstname, customer.lastname,
            customer.country, employee.firstname, employee.lastname
        FROM invoice
    RIGHT JOIN customer
        ON invoice.customerid = customer.customerid
    INNER JOIN employee
        ON employee.employeeid = customer.supportrepid
    WHERE employee.title LIKE 'sales%agent';
```

8. **How many Invoices were there in 2009 and 2011? What are the respective total sales for each of those years?**

   There were 83 invoices in 2009 and 83 invoices in 2011 as well. Total sales in 2009 was $449.46 and total sales in 2011 was $469.58.

```
    SELECT YEAR(invoicedate), COUNT(invoicedate), SUM(total)
        FROM invoice
    WHERE YEAR(invoicedate) = 2009 OR YEAR(invoicedate) = 2011
    GROUP BY YEAR(invoicedate);
```

9. **Looking at the InvoiceLine table, provide a query that COUNTs the number of line items for Invoice ID 37.**

```
    SELECT COUNT(*)
        FROM invoiceline
    WHERE invoiceid = 37;
```

10. **Looking at the InvoiceLine table, provide a query that COUNTs the number of line items for each Invoice. HINT: GROUP BY**

```
SELECT invoiceid, COUNT(*)
    FROM invoiceline
GROUP BY invoiceid;
```

11. **Provide a query that includes the track name with each invoice line item.**

```
SELECT track.name, invoiceline.*
    FROM invoiceline
LEFT JOIN track
    ON invoiceline.trackid = track.trackid;
```

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

```
SELECT track.name, artist.name, invoiceline.*
    FROM invoiceline
LEFT JOIN track
    ON invoiceline.trackid = track.trackid
LEFT JOIN album
    ON track.trackid = album.albumid
LEFT JOIN artist
    ON album.artistid = artist.artistid;
```

13. **Provide a query that shows the # of invoices per country. HINT: GROUP BY**

```
SELECT billingcountry, count(*)
    FROM invoice
GROUP BY billingcountry;
```

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

```
SELECT playlisttrack.playlistid, playlist.name, count(*)
    FROM playlisttrack
LEFT JOIN playlist
    ON playlist.playlistid = playlisttrack.playlistid
GROUP BY playlisttrack.playlistid;
```

15. **Provide a query that shows all the Tracks but displays no IDs. The resultant table should include the Album name, Media type and Genre.**

    I don't get the first part of the question. Do you want me to display every field in the track except IDs?

```
SELECT track.name as trackName, album.title as albumTitle,
        mediatype.name as mediaType, genre.name as genre
    FROM track
LEFT JOIN album
    ON album.albumid = track.albumid
LEFT JOIN mediatype
    ON mediatype.mediatypeid = track.mediatypeid
LEFT JOIN genre
    ON genre.genreid = track.genreid;
```

16. **Provide a query that shows all Invoices but includes the # of invoice line items.**

```
    SELECT invoice.*, COUNT(invoiceline.invoicelineid)
        FROM invoiceline
    LEFT JOIN invoice
        ON invoice.invoiceid = invoiceline.invoiceid
    GROUP BY invoiceline.invoiceid;
```

17. **Provide a query that shows total sales made by each sales agent.**

```
    SELECT employee.firstname, employee.lastname, SUM(invoice.total)
        FROM employee
    LEFT JOIN customer
        ON employee.employeeid = customer.supportrepid
    LEFT JOIN invoice
        ON invoice.customerid = customer.customerid
    WHERE employee.title LIKE "sale%agent"
    GROUP BY employee.employeeid;
```

18. **Which sales agent made the most in sales in 2009?**

    Steve Johnson made most sales, $164.34, in 2009 among all sales agents.

```
    SELECT employee.firstname, employee.lastname, SUM(invoice.total)
        FROM employee
    LEFT JOIN customer
        ON employee.employeeid = customer.supportrepid
    LEFT JOIN invoice
        ON invoice.customerid = customer.customerid
    WHERE employee.title LIKE "sale%agent"
        AND YEAR(invoice.invoicedate) = 2009
    GROUP BY employee.employeeid;
```

19. **Which sales agent made the most in sales in 2010?**

    Jane Peacock made most sales, $221.92, in 2010 among all sales agents.

```
    SELECT employee.firstname, employee.lastname, SUM(invoice.total)
        FROM employee
    LEFT JOIN customer
        ON employee.employeeid = customer.supportrepid
    LEFT JOIN invoice
        ON invoice.customerid = customer.customerid
    WHERE employee.title LIKE "sale%agent"
        AND YEAR(invoice.invoicedate) = 2010
    GROUP BY employee.employeeid;
```

20. **Which sales agent made the most in sales over all?**

    Jane Peacock made most sales, $833.04, in over all among all sales agents.

```
    SELECT employee.firstname, employee.lastname, SUM(invoice.total)
        FROM employee
    LEFT JOIN customer
        ON employee.employeeid = customer.supportrepid
    LEFT JOIN invoice
        ON invoice.customerid = customer.customerid
    WHERE employee.title LIKE "sale%agent"
    GROUP BY employee.employeeid;
```

21. **Provide a query that shows the # of customers assigned to each sales agent.**

```
SELECT employee.firstname, employee.lastname,
        SUM(customer.supportrepid)
    FROM employee
LEFT JOIN customer
    ON employee.employeeid = customer.supportrepid
WHERE employee.title LIKE "sale%agent"
GROUP BY employee.employeeid;
```

22. **Provide a query that shows the total sales per country. Which country's customers spent the most?**

    Customers in USA spent the most with $523.06.

```
SELECT billingcountry, SUM(total)
    FROM invoice
GROUP BY billingcountry
ORDER BY SUM(total) desc;
```

23. **Provide a query that shows the most purchased track of 2013.**

```
SELECT invoiceline.trackid, SUM(invoiceline.quantity)
    FROM invoiceline
LEFT JOIN invoice
    ON invoice.invoiceid = invoiceline.invoiceid
WHERE YEAR(invoice.invoicedate) = 2013
GROUP BY invoiceline.trackid
ORDER BY SUM(invoiceline.quantity) desc
LIMIT 1;
```

24. **Provide a query that shows the top 5 most purchased tracks overall.**

```
SELECT invoiceline.trackid, SUM(invoiceline.quantity)
    FROM invoiceline
LEFT JOIN invoice
    ON invoice.invoiceid = invoiceline.invoiceid
GROUP BY invoiceline.trackid
ORDER BY SUM(invoiceline.quantity) desc
LIMIT 5;
```

25. **Provide a query that shows the top 3 bestselling artists.**

    best selling in track selling wise

```
SELECT artist.name, SUM(invoiceline.quantity)
    FROM artist
LEFT JOIN album
    ON album.artistid = artist.artistid
LEFT JOIN track
    ON track.albumid = album.albumid
LEFT JOIN invoiceline
    ON invoiceline.trackid = track.trackid
GROUP BY artist.name
ORDER BY SUM(invoiceline.quantity) desc
LIMIT 3;
```

26. **Provide a query that shows the most purchased Media Type.**

```
    SELECT mediatype.name, SUM(invoiceline.quantity)
        FROM mediatype
    LEFT JOIN track
        ON track.mediatypeid = mediatype.mediatypeid
    LEFT JOIN invoiceline
        ON invoiceline.trackid = track.trackid
    GROUP BY mediatype.name
    ORDER BY SUM(invoiceline.quantity) desc
    LIMIT 1;
```

27. **Provide a query that shows the number tracks purchased in all invoices that contain more than one genre.**

    I don't get what the question means by *"shows the number tracks"*.

```
    SELECT invoice.*, COUNT(genre.genreid)
        FROM invoice
    LEFT JOIN invoiceline
        ON invoice.invoiceid = invoiceline.invoiceid
    LEFT JOIN track
        ON invoiceline.trackid = track.trackid
    LEFT JOIN genre
        ON genre.genreid = track.genreid
    GROUP BY invoice.invoiceid
    HAVING COUNT(genre.genreid) > 1;
```