```sql
--1. Who is the senior most employee based on job title?  ----
select top 1 e.employee_id,e.title ,e.levels,e.first_name,e.last_name
from employee e
order by e.employee_id desc

-- In above we need to sort by level to get


--2.-- Which countries have the most Invoices?
select * from invoice
select * from invoice_line

select top 10 count (I.invoice_id) as count_of_invoices ,I.billing_country
from invoice I
group by I.billing_country
order by I.billing_country Desc

-- This counting the number of invoices and then grouped by country

-- --3  What are top 3 values of total invoice?

select * from invoice

select top 3 Round (I.total,2 ) as Rounded_Total_value
from invoice I
order by I.total Desc
--- this gives rounded value to 2 digit
or

Select top 3 i.total
from invoice I
order by I.total Desc

/* 4. Which city has the best customers? We would like to throw a promotional Music
Festival in the city we made the most money. Write a query that returns one city that
has the highest sum of invoice totals. Return both the city name & sum of all invoice
totals **/

select * from invoice
select * from invoice_line


select top 1 i.billing_city ,Round (sum (I.total),2) as Sum_of_Invoices_by_city
from invoice I
group by I.billing_city
order by sum (I.total) Desc
-- this sum the invoices and then group by city
```

```
/* 5. Who is the best customer? The customer who has spent the most money will be
declared the best customer. Write a query that returns the person who has spent the
most money */

select * from customer
select * from invoice

select  i. customer_id,C.First_name,C.last_name, round(Sum(I.total),2) as
  Sum_Amount_by_customer
from customer c join invoice I on C.customer_id = I.customer_id
Group by I.customer_id,C.first_name,C.last_name
order by Sum(I.total) Desc

select top 5 i. customer_id,C.First_name,C.last_name, round(Sum(I.total),2) as
  Sum_Amount_by_customer
from customer c join invoice I on C.customer_id = I.customer_id
Group by I.customer_id,C.first_name,C.last_name
order by Sum(I.total) Desc


/* 6 Write query to return the email, first name, last name, & Genre of all Rock Music

listeners. Return your list ordered alphabetically by email starting with A */


select * from customer
-- > C.id = C ID
select * from invoice
---> Invoiceid =  invoice ID
select * from invoice_line
--- > TrackId = trackid
select * from track
---> genreID = Genre ID
select * from genre

Select distinct c.first_name,c.last_name,c.email
from customer c
join invoice I on c.customer_id = I.customer_id
join invoice_line ID on I.invoice_id = id.invoice_id
join track T on ID.track_id = t.track_id
join genre g on t.genre_id = g.genre_id
where g.genre_id = 1
order by email asc  --- use where condition


/**2. Let's invite the artists who have written the most rock music in our dataset.
  Write a
query that returns the Artist name and total track count of the top 10 rock bands */


select * from album
select * from artist
```

```sql
select * from genre
select * from track


select g.genre_id, g.name, count(t.track_id) as Count_of_Track
from track t join genre g on t.genre_id= g.genre_id
group by g.genre_id,g.name
order by count(t.track_id) Desc --- this gives count of songs in each genre

select * from album
select * from artist
select * from genre
select * from track


select top 10
ar.artist_id,ar.name,count(ar.artist_id) as number_of_songs
from Track T
Join  album al on al.album_id =t.album_id
join artist ar on ar.artist_id = al.album_id
join genre g on g.genre_id= t.genre_id
where g.name = 'rock'
group by ar.artist_id,ar.name
order by number_of_songs desc


/** Return all the *track names * that have a song length longer than the average song length.
Return the Name and Milliseconds for each track. Order by the song length with the
longest songs listed first
*/

select * from track


select t.name ,t.milliseconds
from track t
where t.milliseconds > (
select  avg(t.milliseconds) as Avg_length_Songs
from track t )
order by t.milliseconds desc

/*find how much amount spent by each customer on artists? Write a query to return
customer name, artist name and total spent */

with most_selling_singer as
(
select  i.customer_id , ar.name as Artist_name,
sum (l.unit_price * l.quantity) as total_amount_spending
from invoice_line l
join invoice I on I.invoice_id =l.invoice_id
join track t  on l.track_id = t.track_id
join album al on t.album_id = al.album_id
join artist ar on al.artist_id = ar.artist_id
group by i.customer_id,ar.name
)
select  c.customer_id ,C.first_name,c.last_name,CSW.Artist_name
from customer c join invoice I on c.customer_id= I.customer_id
join invoice_line L  on l.invoice_id= I.invoice_id
join most_selling_singer CSW on I.customer_id =csw.customer_id
group by c.customer_id ,C.first_name,c.last_name,CSW.Artist_name
order by total_amount_spending desc
-- my query above

-- just restructuring and  ( Mistake _ order clause i was using in within cte is taken out to external main query

WITH most_selling_singer AS (
    SELECT
        i.customer_id,
        ar.name AS Artist_name,
        SUM(l.unit_price * l.quantity) AS total_amount_spending
    FROM
        invoice_line l
        JOIN invoice I ON I.invoice_id = l.invoice_id
        JOIN track t ON l.track_id = t.track_id
        JOIN album al ON t.album_id = al.album_id
        JOIN artist ar ON al.artist_id = ar.artist_id
    GROUP BY
        i.customer_id, ar.name
)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    CSW.Artist_name,
    MAX(CSW.total_amount_spending) AS max_total_amount_spending
FROM
    customer c
    JOIN invoice I ON c.customer_id = I.customer_id
    JOIN invoice_line L ON L.invoice_id = I.invoice_id
    JOIN most_selling_singer CSW ON I.customer_id = CSW.customer_id
GROUP BY
    c.customer_id, c.first_name, c.last_name, CSW.Artist_name
ORDER BY
    max_total_amount_spending DESC;

*We want to find out the most popular music Genre for each country. We determine the
most popular genre as the genre with the highest amount of purchases. Write a query
that returns each country along with the top Genre. For countries where the maximum
number of purchases is shared return all Genres*/


select * from genre
select * from track
select * from invoice
select * from invoice_line

/*We want to find out the most popular music Genre for each country. We determine the
most popular genre as the genre with the highest amount of purchases. Write a query
that returns each country along with the top Genre. For countries where the maximum
number of purchases is shared return all Genres*/

WITH POPULAR_GENRE AS (
    SELECT
        COUNT(invoice_line.quantity) AS purchases,
        customer.country,
        genre.name AS genre_name,
        genre.genre_id,
        DENSE_RANK() OVER (PARTITION BY customer.country ORDER BY COUNT
          (invoice_line.quantity) DESC) AS Rnk --- instead of rownumber using dense
          clause to assign rank

    FROM
        invoice_line
        JOIN invoice ON invoice.invoice_id = invoice_line.invoice_id
        JOIN customer ON customer.customer_id = invoice.customer_id
        JOIN track ON track.track_id = invoice_line.track_id
        JOIN genre ON track.genre_id = genre.genre_id
    GROUP BY customer.country, genre.name, genre.genre_id
)
SELECT
    country,
    genre_name AS top_genre
FROM POPULAR_GENRE
WHERE Rnk = 1;
```