

-- 1.WHO IS THE SENIOR MOST EMPLOYEES BASED ON THE LEVEL

```
select first_name,levels    from employee
order by levels desc
limit 1 ;
```

-- 2.WHICH COUNTRY HAVE THE MOST NUMBER OF INVOICES

```
select count(invoice_id) as no_of_invoices, billing_country
from invoice
group by 2
order by no_of_invoices desc
limit 1 ;
```

-- 3.TOP 3 VALUES OF TOTAL INVOICES

```
select sum(total) as amount ,billing_country
from invoice
group by 2
order by amount desc
;
```

-- 4.WHICH CITY HAS THE BEST CUSTOMERS

```
select sum(total) as invoice_total,billing_city
from invoice
group by billing_city
order by invoice_total desc ;
```

-- 5.WHO IS THE BEST CUSTOMER , THE CUSTOMER WHO HAS SPENT THE MOST MONEY
WILL BE DECLARED AS THE BEST CUSTOMER .WIRTE A

-- QUERY THAT RETURNS THE PERSON WHO HAS SPENT THE MOST MONEY

```
select customer.first_name,customer.last_name,sum(invoice.total) as
amount_spent
from customer
inner join invoice on
customer.customer_id = invoice.customer_id
group by 1,2
order by amount_spent desc;
```

-- 6.write a query to return email ,first_name,last_name and genre of the
rock music listeners .

-- Return the list ordered alphabetically by email starting with 'A'

```
select distinct email,first_name,last_name
from customer
join invoice on customer.customer_id = invoice.customer_id
join invoice_line on invoice.invoice_id = invoice_line.invoice_id
where track_id in
```

(

```
select track_id from track
join genre on track.genre_id = genre.genre_id
where genre.name like 'ROCK'
)
```

```
order by email ;
```

```
-- 7.LETUS INVITE THE ARTIST WHO HAVE WRITTEN THE MOST ROCK MUSIC IN OUR
DATASET.
```

```
-- WRITE AQUERY THAT RETURN THE ARTIST NAME AND THE TOTAL TRACK COUNT OF
THE TOP 10 ROCK BANDS
```

```
select artist.artist_id,artist.name,count(artist.artist_id) AS
number_of_songs
from track
join album2 on album2.album_id = track.album_id
join artist on artist.artist_id = album2.artist_id
join genre on genre.genre_id = track.genre_id
where genre.name like 'ROCK'
group by 1 ,2
order by number_of_songs desc
limit 10;
```

```
-- 8.RETURN ALL THE SONG NAMES THAT HAVE SONG LENGTH LONGER THAN THE
AVERAGE SONG LENGTH.
```

```
-- RETURN THE NAME AND THE MILLISECONDS FOR EACH TRACK ,ORDER BY THE SONG
LENGTH
```

```
select name , milliseconds
from
track
where milliseconds > (
select avg(milliseconds) as avg_track_length
from track
)
```

```
order by milliseconds desc;
```