```sql
-- Q1: Find the top 10 ranked songs in 2010(spotify)
select
distinct year_rank,
group_name,
song_name
from billboard_top_100_year_end
where year = 2010
group by 1
limit 10

-- Q2: Top Ranked Songs(spotify)
select
trackname,
count(*)
from spotify_worldwide_daily_song_ranking
where position = 1
group by 1
order by 2 DESC

-- Q:3 Share of Active Users
with cte1 as(
select
*
from fb_active_users
where country = "USA" and status = "open"
),
cte2 as(
select
*
from fb_active_users
where country = "USA"
)
select (select count(*) from cte1)  /(select count(*)from cte2)
```

```sql
-- Q4: Users By Average Session Time(meta/Facebook)
select
user_id,
avg(timestampdiff(second, p_load_time, p_exit_time)) as avg1
from
(select
user_id,
date(timestamp),
max(case when action = "page_load" then timestamp else null end) as p_load_time ,
min(case when action = "page_exit" then timestamp else null end) as p_exit_time
from facebook_web_log
group by 1 , 2) as t
group by 1
having avg1 is not null

-- Q5: Risky Projects(linkdln)
with cte1 as
(select p.title, p.budget, e.salary,
(e.salary/365) as 1_day_of_sal,
datediff(p.end_date, p.start_date) as total_days
from linkedin_projects as p
join linkedin_emp_projects as ep
on p.id = ep.project_id
join linkedin_employees as e
on e.id = ep.emp_id)

select title,budget,prorated_employee_expense
from
(select *,
ceil(sum(1_day_of_sal * total_days)) as prorated_employee_expense
from cte1
group by 1) as t
where budget<prorated_employee_expense
```