1. Для игроков вывести средний overal_rating. Пронумеровать игроков в порядке убывания среднего overal_rating

```
with cte as
(
select player_api_id, avg(overall_rating) as avg_overall
from player_attributes
group by player_api_id
)
select *, row_number() over(order by avg_overall desc) as avg_overall_rating
 from cte
```

2. Вывести рейтинг команд по показателю buildUpPlaySpeed в рамках каждого buildUpPlaySpeedClass (чем больше тем лучше) за 2010 год.

```
select team_api_id, buildUpPlaySpeedClass, buildUpPlaySpeed, row_number() over(partition by buildUpPlaySpeedClass order by buildUpPlaySpeed desc) as buildUpPlaySpeed_rating

from team_attributes

where date like '2010%'
```

3.Вывести для команд приращение параметра buildUpPlayPassing с течением времени

```
select team_api_id,`date`, buildUpPlayPassing, buildUpPlayPassing - lag(buildUpPlayPassing)
over (partition by team_api_id order by `date`) as diff
from team_attributes
```

4. Для каждой лиги и сезона вывести название лиги, сезон и список всех команд, которые играли в этой лиге в этом сезоне (в одном поле через запятую).

```
with cte as

(

select distinct season, league.name as league_name, home_team_api_id, away_team_api_id

from `match`, league

where league.id = league_id

),

 cte2 as

( select team_long_name, season, league_name

from team

join cte
on team_api_id = home_team_api_id or team_api_id = away_team_api_id
)
select season, league_name, group_concat(distinct team_long_name) as conca from cte2
group by season, league_name
```

5. Вывести для каждого сезона и лиги количество сыгравших команд в порядке убывания.

```
with cte as
(
select distinct season, league.name as league_name, home_team_api_id, away_team_api_id
from `match`, league
where league.id = league_id
),
 cte2 as
( select team_short_name, season, league_name
from team
join cte
on team_api_id = home_team_api_id or team_api_id = away_team_api_id
)
select season, league_name, count(distinct team_short_name) as team_count from cte2
group by season, league_name
order by team_count desc
```

6. Как менялось количество команд, сыгравших в каждой лиге от сезона к сезону?

```
with cte as
(
select distinct season, league.name as league_name, home_team_api_id, away_team_api_id
from `match`, league
where league.id = league_id
),
 cte2 as
( select team_short_name, season, league_name
from team
join cte

on team_api_id = home_team_api_id or team_api_id = away_team_api_id

)

select season, league_name, count(distinct team_short_name) as team_count, count(distinct team_short_name) - lag(count(distinct team_short_name)) over (partition by league_name order by season) as diff from cte2

group by season, league_name
```

7. Вывести рейтинг лиг по количеству команд, игравших в них по сезонам по убыванию. Лиги, в которых играли более 20 команд должны иметь значение новой переменной number_team_rating ‘high_number ‘, остальные ‘low_number ‘

```
select *, row_number() over(partition by number_team_rating order by cont desc) as rating

from

(

with cte as

(

select distinct season, league.name as league_name, home_team_api_id, away_team_api_id

from `match`, league

where league.id = league_id

),

 cte2 as

( select team_short_name, season, league_name

from team

join cte

on team_api_id = home_team_api_id or team_api_id = away_team_api_id

)

select season, league_name, count(distinct team_short_name)as cont,case when count(distinct team_short_name)>15 then 'high_number' else 'low_number' end as number_team_rating  from cte2

group by season, league_name

) as t
```