**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