```sql
--VISTA_A that, for each of the games of the current season of the national football
--championship of Liga Nos, shows the date of the game, the visited and visiting
team,
--the number of goals of each team, and the result information in the format (1, x,
2),
--when, respectively, there is a victory for the visited team (1), a tie (x), or a
victory
--for the visiting team (2). Sort the results by the date of the game.

--Jornada DataJogo equipaVisitada equipaVisitante resultado
--------- -------- --------------- --------------- ---------

  create or replace view view_a AS
  select hg.phase_id as Round,
         hg.match_date as MatchDate,
         hg.A_team_id as Hosts,
         hg.B_team_id as Visitors,

--        case inserting 1,2,x depending on the match result
        case
        when hg.A_goals > hg.B_goals then '1'
        when hg.A_goals < hg.B_goals then '2'
        when hg.A_goals = hg.B_goals then 'x'
        else ''
        end as match_result

  from history_games hg
  join phases p
  on hg.phase_id = p.phase_id
  where p.competition_id = 'LN'; --this is our Liga Nos season 2019/2020

```

```sql
--VISTA_B that, for each of the games of the last matchday of the current season of
--the national football championship of Liga Nos, show the date of the game,
--the visited and visiting team, the total amount bet and the total amount of
--prizes paid to the bettors who won the respective bets. Sort the results by the
total stake.

--DataJogo equipaVisitada equipaVisitante TotalApostado totalPremios
---------- -------------- --------------- ------------- ------------

create or replace view view_b as

select match_date,
       A_team_id,
       B_team_id,
       odds_sums.sum_bets,
       payouts_sums.sum_payouts


from games g,
  (
--    get games_id with the total money bet on that game
    select o.game_id as o_game_id,
           round(sum(b.money_placed),2) as sum_bets
    from odds o
    left outer join bets b on o.odd_id = b.odd_id
    group by o.game_id
    order by o.game_id desc
  ) odds_sums,

  (
--    get games_id with the total prizes paid for that game
    select o.game_id as o_game_id,
           round(sum(p.money),2) as sum_payouts
    from odds o
    left outer join bets b on o.odd_id = b.odd_id
    left outer join payouts p on b.bet_id = p.bet_id
    group by o.game_id
  ) payouts_sums

--      connect odds and prizes by the same game_id
where g.game_id = odds_sums.o_game_id and
      g.game_id = payouts_sums.o_game_id and

--      there are some games the money is not bet on
      odds_sums.sum_bets is not null and
      payouts_sums.sum_payouts is not null and

--      select only those matches with the latest match_date
      match_date = (
                    select max(match_date)
                    from games);
```

```sql
--VISTA_C that considering all the games played so far in the current season of
--the national football championship of Liga Nos, show for each team, their
--classification, the number of games played, the number of wins, the number of
--draws, the number of losses and the number of points obtained.
--Consider that 3 points are awarded for each victory and 1 point for each tie.
--The ranking criteria are: the number of points, the lowest number of games played,
--and the highest number of victories.

--Classif nomeEquipa nJogos NVitorias NEmpates NDerrotas NPontos
--------- ----------- ------ --------- --------- --------- -------

create or replace view view_c as

select row_number() over
--        number rows orderer by points scored
       (order by (s.won*3) + s.draw desc, s.played, s.won desc) as place,
       t.name,
       s.played,
       s.won,
       s.draw,
       s.lost,
       (s.won*3) + s.draw as points

from teams t, team_statistics s
where t.team_id = s.team_id and s.competition_id = (
        select competition_id
        from competitions
        where name = 'Liga NOS' and season = '2019/20');
```

```sql
--VISTA_D that considering all the bets registered in the last 2 years show which
--are the 10 games with the highest volume ($$) of bets. Also show the number of
players.
--Sort descending by the total stake.

--DataJogo equipaVisitada equipaVisitante TotalApostado NumApostadores
---------- ------------- --------------- ------------- ----------------

create or replace view view_d as

select * from(
  select g.match_date,
         g.A_team_id as visited,
         B_team_id as visiting,
         sum(b.money_placed) as bets,
         count(o.game_id) as bettors

  from bets b, history_odds o, games g

--  connect subqueries
  where o.odd_id = b.odd_id and o.game_id = g.game_id and

--  form recent two years
       g.match_date > add_months(sysdate, -2 * 12)
  group by g.match_date, g.A_team_id, B_team_id
  order by bets desc)

--  take top 10. Rownum is in outer query in order to take into account 'order by'
clause.
where rownum <= 10;
```

```
--VISTA_E that considering all the bets of the games registered in the last
--2 years show which players have abnormal bets, that is, whose total value
--bet by the player is greater than 10% of the total bet in that game by all the
bettors.
--Sort the result in descending order by percentage.

--DataJogo equipaVisitada equipaVisitante NomeJogador TotalApostado Percent
---------- -------------- --------------- ----------- ------------- -------

create or replace view view_e as

select g.match_date,
       g.A_team_id,
       g.B_team_id,
       c.name,
       b.money_placed,
       round(b.money_placed / game_total_sum.total_money * 100,2) as percentage

--       connect games with odds and bets with bettors
--       games -> odds -> bets -> clinets
from history_games g
left outer join history_odds o on g.game_id = o.game_id
left outer join bets b on o.odd_id = b.odd_id
join clients c on b.client_id = c.client_id,
(
--      game_id with the total monet placed on that game
      select ga.game_id as ga_game_id,
--      if there is no money placed (sum = null) place 0
            coalesce(sum(b.money_placed),0) as total_money

      from history_games ga
      left outer join history_odds o on ga.game_id = o.game_id
      left outer join bets b on o.odd_id = b.odd_id
      left outer join clients c on b.client_id = c.client_id
      group by ga.game_id

      ) game_total_sum

--      take only matches from past two years
where g.match_date > add_months(sysdate, -2 * 12) and

--      bet by the player is greater than 10% of the total bet in that game by all
the bettors.
      b.money_placed > game_total_sum.total_money * 0.1 and

--      connect game_id from total sum with current game listed
      game_total_sum.ga_game_id = g.game_id
order by percentage asc;
```

```
--VISTA_F that for the game with the highest volume of bets, show how the odd value of
--each of the possible results has evolved over time. As in an hourly interval
--(eg from 10 am to 11 am) the value of an odd can change more than once,
--for each of the possible results, show the maximum value of the odd in each hour.
--Order the result temporally.

--DataJogo equipa1 equipa2 dataHoraOdd oddVitoriaEq1 OddEmpate oddVitoriaEq2
---------- ------- ------- ----------- ------------- --------- -------------
create or replace view view_f as

select g.match_date, g.A_team_id, g.B_team_id, o.odd_date,
  round((select value from odds where game_id = g.game_id and odd_type_id = 1),2) as TeamAWin,
  round((select value from odds where game_id = g.game_id and odd_type_id = 2),2) as Draw,
  round((select value from odds where game_id = g.game_id and odd_type_id = 3),2) as TeamBWin
from games g, odds o
--odd_type_id = 1 prevents the same record from appearing 3 times
where g.game_id = o.game_id and o.odd_type_id = 1 and

--      take records from certain period of time (from 10 to 11)
      to_char(o.odd_date, 'HH24') > '10' and
      to_char(o.odd_date, 'HH24') < '11' and
--
--      take only odds from game with the highest volume of bets
      g.game_id = (
            select g_game_id
            from (
                  select g.game_id as g_game_id,
                         count(b.bet_id)
                  from history_games g
                  join history_odds o on g.game_id = o.game_id
                  join bets b on o.odd_id = b.odd_id
                  group by g.game_id
                  order by count(b.bet_id) desc
                  )
--              take top 1
              where rownum = 1);
```

```sql
--VISTA_G that shows the next 10 games scheduled, and not yet played, and for one
--of them shows the current value of the odd of each of the possible results.
--Sort the result by the date of the game (closest to first).

--DataJogo equipaVisitada equipaVisitante oddVitoriaEq1 OddEmpate oddVitoriaEq2
---------- --------------- --------------- ------------- --------- -------------

create or replace view view_g as

select *
from(
  select g.match_date,
         g.A_team_id,
         g.B_team_id,
         (
--         get odd for the win
           select round(value,2)
           from odds
           where game_id = g.game_id and odd_type_id = 1
         ) as odd_1,
         (
--         get odd for the draw
           select round(value,2)
           from odds
           where game_id = g.game_id and odd_type_id = 2
         ) as odd_x,
         (
--         get odd for the lost
           select round(value,2)
           from odds
           where game_id = g.game_id and odd_type_id = 3
         ) as odd_2

  from games g
--  get following games
  where g.match_date > sysdate
  order by g.match_date
  )
--  get ten matches
where rownum <= 10;
```

```
--VISTA_H that for each team, show which game was assigned a higher win result value.
--Consider only the games played in the previous month. Sort descending by the
highest odd.

--DataJogo equipaVisitada equipaVisitante oddVitoriaEq
---------- --------------- --------------- ------------

create or replace view view_h as

select g1.match_date,
       g1.A_team_id,
       g1.B_team_id,
       max_odds_games.max_odd

from history_games g1,
     (
--          get game_id and max odd value for that game
--          subquery is used to avoid disconnection from other join tables
          select g.game_id as g_game_id,
                 round(max(o.value),2) as max_odd
          from history_games g
          join history_odds o on g.game_id = o.game_id

--          take only odd beting winning teamA or teamB
          where o.odd_type_id = 1 or o.odd_type_id = 3
          group by g.game_id
          order by round(max(o.value),2) desc
      ) max_odds_games

where g1.game_id = max_odds_games.g_game_id and
--          take only matches from previous month
--          greater then first day of previous month:
      g1.match_date > last_day(add_months(sysdate,-2))+1 and
--          smaller then last day of previous month:
      g1.match_date < last_day(add_months(sysdate,-1));
```

```sql
--VISTA_I showing the top 10 players / punters who have been paid the highest total
--prize amounts. Consider the total amount of prizes paid since the beginning of
--the current year and whose total amount received in prizes is greater than 100%
--of the total amount spent on these bets. Consider only those players who have bet
--on at least 10% of the games that occurred during that period. Sort descendingly
--by the total amount paid in premiums.

--NomeJogador AnoMes(?) MontanteTotalApostado MontanteTotalPremios
------------- ------ -------------------- --------------------

create or replace view view_i as

select *
from (
          select c.name,
               round(sum(b.money_placed),2) as money_placed,
               round(sum(p.money),2) as money_won

        --join clients with payouts
        --  client -> bets -> odds -> payouts
        from clients c
        join bets b on c.client_id = b.client_id
        join history_odds o on b.odd_id = o.odd_id
        join payouts p on b.bet_id = p.bet_id

        --join odds with games
        join history_games g on o.game_id = g.game_id

        --take only matches from last year
        where g.match_date > add_months(sysdate, -1 * 12)
        group by c.client_id, c.name

        --players who have bet on at least 10% of the games that occurred during
        that period
        having count(g.game_id) < (
                        select count(game_id) * 0.1 as tenPercent
                        from history_games
                        --take only matches from last year
                        where match_date > add_months(sysdate, -1 * 12)
                      ) and

        --            total amount received in prizes is greater than 100% of the
        total amount spent on these bets
                    sum(p.money) > sum(b.money_placed)

        --Sort descendingly by the total amount paid in premiums.
        order by money_won desc
)
--showing the top 10 players
where rownum <= 10;
```

```sql
--Players' percentage contribution in Company's pure income. In other words,
--in which precentage company earn from clients looses
--contribution = total_placed - total_payout
--company_pure_income = contribution of all cients
--(How much money in percentage bettor left in company loosing)

create or replace view view_j_a2019156557 as

select c.client_id,
      c.name,
      c.surname,
      round(
--      calculate client contribution
      (client_sum_bet.total - client_sum_payout.total) /
--      divide by
      (
--        calculate all company pure income
--        total money placed by clients - total money paid out
        (select sum(b.money_placed) from bets b) -
        (select round(sum(p.money),2) from payouts p)
      ) * 100,2) as precetnage_contribution

from clients c,
    (
--    get client_id with all money placed by this client
      select b.client_id as b_client_id,
      sum(b.money_placed) as total
      from bets b
      group by b.client_id
    ) client_sum_bet,
    (
--    get client_id with all monet won(paid by company) by this client
      select p.client_id as p_client_id,
      round(sum(p.money),2) as total
      from payouts p
      group by p.client_id
    ) client_sum_payout

--    connect total sum and payouts from clients
where c.client_id = client_sum_bet.b_client_id and
      c.client_id = client_sum_payout.p_client_id
order by precetnage_contribution desc;
```

```sql
--List of clients with their win ratio of all

create or replace view view_j_a2019156734 as

select c.client_id,
       c.name,
       c.surname,
       round(p.total_wins / b.total_bets * 100,2) win_ratio,
       b.total_bets

from clients c,
    (
--    get client_id with client's counted all bets
      select b.client_id as b_client_id, count(b.bet_id) as total_bets
      from bets b
      group by b.client_id
    ) b,
    (
--    get client_id with client's counted all payouts
--    (if there is payout with bet_id, this bet is won)
      select p.client_id as p_client_id, count(p.payout_id) as total_wins
      from payouts p
      group by p.client_id
    ) p
--    connect all subqueries
where c.client_id = b.b_client_id and c.client_id = p.p_client_id
--order results by the winratio
order by round(p.total_wins / b.total_bets * 100,2) desc;
```

```sql
--list of top 10 underdogs(team which had big odd - not supposed to win) which
actually won the game
--with the money won and name of that client (true_fan).

create or replace view view_k_a2019156557 as

select c.client_id,
       c.name,
       round(o.value,2) as odd_value,

--       take the team name depending on with team was the bet
       case when o.odd_type_id = 1 then
           (
             Select t.name
             from teams t
             where t.team_id = g.A_team_id
           )
           else
           (
             Select t.name
             from teams t
             where t.team_id = g.B_team_id
           )
           end as underdog


--    history_games -> odds -> bets -> payouts
--    I needed to find only games of odds of bets attached to payouts
--    because bet on certain odd is winning when there is and payout with the odd_id

from history_games g
join history_odds o on o.game_id = g.game_id
join bets b on o.odd_id = b.odd_id
join payouts p on b.bet_id = p.bet_id,
     clients c

where (o.odd_type_id = 1 or o.odd_type_id = 3) and
      p.client_id = c.client_id

order by o.value desc;
```

```sql
--list of most aggresive teams - counted numer of fauls

create or replace view view_k_a2019156734 as

select t.name,
      event_count.total
from (
--         get team_id and counted total fouls
        select e.team_id as e_team_id,
        count(et.event_type_id) as total

        from history_games g
        join events e on g.game_id = e.game_id
        join event_type et on e.event_type_id = et.event_type_id

--         foul is an event of type 11
        where et.event_type_id = 11
        group by e.team_id

      ) event_count
--         join subquery with teams table to obtain the name of the team
      join teams t on event_count.e_team_id = t.team_id

order by event_count.total desc;
```