Game

1.List of main players in a team in a match

```
πp.name,pig.id,g.team_id,g.match_id((((game⋈g.player_id=pig.idσis_main_lineup=1
(player_in_game))⋈g.team_id=t.team_idteam)⋈g.match_id=m.match_idmatches)⋈pig.id=p.idplayer)

select p.name, pig.id, g.team_id, g.match_id
from game g
join player_in_game pig on pig.id = g.player_id
join team t on t.team_id = g.team_id
join matches m on m.match_id = g.match_id
join player p on p.id = pig.id
where is_main_lineup = 1
```

2.Details about games between two teams

```
Join1=game⋈g.player_id=pig.idplayer_in_game
Join2=Join1⋈g.team_id=t.team_idteam
Join3=Join2⋈g.match_id=m.match_idmatches

γt.team_id,t.name,g.match_id;SUM(pig.goals),SUM(pig.fouls),SUM(pig.yellow_cards),SUM(pig.red_cards
),goals_received,lineup((πt.team_id,t.name,g.match_id,SUM(pig.goals)⟶goals_scored,SUM(pig.fouls)⟶
fouls,SUM(pig.yellow_cards)⟶yellow_cards,SUM(pig.red_cards)⟶
red_cards,SUBSTRING_INDEX(m.result,'−',−1)⟶goals_received,m.lineup_home⟶lineup
(σt.team_id=m.home_team_id(Join3)))∪(πt.team_id,t.name,g.match_id,SUM(pig.goals)⟶
goals_scored,SUM(pig.fouls)⟶fouls,SUM(pig.yellow_cards)⟶yellow_cards,SUM(pig.red_cards)⟶
red_cards,SUBSTRING_INDEX(m.result,'−',1)⟶goals_received,m.lineup_away⟶lineup(σt.team_id
=m.home_team_id(Join3))))

SELECT
    t.team_id,
    t.name,
    g.match_id,
    SUM(pig.goals) AS goals_scored,
    CASE
        WHEN t.team_id = m.home_team_id THEN SUBSTRING_INDEX(m.result, '-', -1)
        ELSE SUBSTRING_INDEX(m.result, '-', 1)
    END AS goals_received,
    SUM(pig.fouls) AS fouls,
    SUM(pig.yellow_cards) AS yellow_cards,
    SUM(pig.red_cards) AS red_cards,
    CASE
```

```
        WHEN t.team_id = m.home_team_id THEN m.lineup_home
        ELSE m.lineup_away
    END AS lineup
FROM
    game g
JOIN
    player_in_game pig ON g.player_id = pig.id
JOIN
    team t ON t.team_id = g.team_id
JOIN
    matches m ON m.match_id = g.match_id
GROUP BY
    g.team_id, g.player_id, g.match_id;
```

3.Name and time of goal scorers

πt.team_id,g.g_id,t.name,g.match_id,g.player_id,g.goal_time((game⋈g.player_id=pig.idσpig.goals>0 (player_in_game))⋈g.team_id=t.team_idteam)

```
select t.team_id, g.g_id,  t.name, g.match_id, g.player_id, g.goal_time
from game g
join team t on t.team_id = g.team_id
join player_in_game pig on pig.id = g.player_id
where pig.goals > 0
```

4.Name of players who got fouls

πg.g_id,g.match_id,t.team_id,p.id,t.name,p.name,g.foul_time,g.foul_type((game⋈g.player_id=pig.id σpig.fouls>0 ∧ g.foul_time =NULL(player_in_game×game))⋈g.team_id=t.team_idteam)⋈pig.id=p.id player

```
select g.g_id, g.match_id, t.team_id, p.id, t.name, p.name, g.foul_time,
g.foul_type
from game g
join team t on t.team_id = g.team_id
join player_in_game pig on pig.id = g.player_id
join player p on pig.id = p.id
where pig.fouls > 0 and g.foul_time IS NOT NULL
```

5.Subs in a match

πt.team_id,g.match_id,g.g_id,pig.id,p.name,t.name,g.sub_in,g.sub_out,g.sub_time((σg.sub_time!=NULL (game)⋈g.player_id=pig.idplayer_in_game)⋈g.team_id=t.team_idteam)⋈pig.id=p.idplayer

```
select t.team_id, g.match_id, g.g_id, pig.id, p.name, t.name, g.sub_in, g.sub_out,
g.sub_time
from game g
join team t on g.team_id = t.team_id
join player_in_game pig on pig.id = g.player_id
join player p on pig.id = p.id
where sub_time IS NOT NULL
```

Player performance

1.Player history in different teams

Join1=player⋈p.id=pf.player_ssnplays_for
Join2=Join1⋈pf.team_id=t.team_idteam
Join3=Join2 LEFT_OUTER_JOIN mp ON (p.id=mp.player_id)

πp.id,p.name→player_name,t.name→
team_name,pf.contract_draft_time,pf.contract_termination_date,pf.contract_amount,COALESCE(mp.matches_played,0),COALESCE(mp.goals_scored,0),COALESCE(mp.fouls,0),COALESCE(mp.yellow_cards,0),COALESCE(mp.red_cards,0),COALESCE(mp.player_score,0),COALESCE(mp.subbed_out_games,0)(Join3)

```
SELECT
    p.id,
    p.name AS player_name,
    t.name AS team_name,
    pf.contract_draft_time,
    pf.contract_termination_date,
    pf.contract_amount,
    COALESCE(mp.matches_played, 0) AS matches_played,
    COALESCE(mp.goals_scored, 0) AS goals_scored,
    COALESCE(mp.fouls, 0) AS fouls,
    COALESCE(mp.yellow_cards, 0) AS yellow_cards,
    COALESCE(mp.red_cards, 0) AS red_cards,
    COALESCE(mp.player_score, 0) AS plyer_score,
    COALESCE(mp.subbed_out_games, 0) AS subbed_out_games
FROM
    player p
JOIN
    plays_for pf ON p.id = pf.player_ssn
JOIN
    team t ON t.team_id = pf.team_id
LEFT JOIN (
```

```
    SELECT
        pig.id AS player_id,
        COUNT(g.match_id) AS matches_played,
        SUM(pig.goals) AS goals_scored,
        SUM(pig.fouls) AS fouls,
        SUM(pig.yellow_cards) AS yellow_cards,
        SUM(pig.red_cards) AS red_cards,
        AVG(pig.player_score) AS player_score,
        COUNT(CASE WHEN g.sub_out > 0 AND pig.is_main_lineup > 0 THEN 1 END) AS
subbed_out_games
    FROM
        player_in_game pig
    JOIN
        game g ON pig.id = g.player_id
    GROUP BY
        pig.id
) AS mp ON mp.player_id = p.id;
```

2.player history in different leagues

Join1=game⋈g.player_id=pig.idplayer_in_game
Join2=Join1⋈pig.id=p.idplayer
Join3=Join2⋈g.team_id=t.team_idteam

πpig.id,g.league_id,g.team_id,t.name,p.name,time_played,goals_scored,player_score
(γpig.id,g.league_id,g.team_id,t.name,p.name; ∑ (pig.time_played), ∑ (pig.goals),avg(pig.player_score)
(Join3))

```
select
    pig.id,
    g.league_id,
    g.team_id,
    t.name,
    p.name,
    sum(pig.time_played) as time_played,
    sum(pig.goals) as goals_scored,
    avg(pig.player_score) as player_score
from
    game g
join
    player_in_game pig on g.player_id = pig.id
join
    player p on p.id = pig.id
join
```

```
    team t on g.team_id = t.team_id
group by
    pig.id,
    g.league_id,
    g.team_id,
    t.name,
    p.name

-- SELECT
--     pig.id AS player_id,
--     SUM(pig.time_played) AS time_played,
--     SUM(pig.goals) AS goals_scored,
--     AVG(pig.player_score) AS player_score,
--     GROUP_CONCAT(DISTINCT g.league_id) AS leagues,
--     GROUP_CONCAT(DISTINCT g.team_id) AS teams,
--     GROUP_CONCAT(DISTINCT t.name) AS team_names,
--     p.name AS player_name
-- FROM
--     game g
-- JOIN
--     player_in_game pig ON g.player_id = pig.id
-- JOIN
--     player p ON p.id = pig.id
-- JOIN
--     team t ON g.team_id = t.team_id
-- GROUP BY
--     pig.id;
```

Team

1.Games played by a team in a league type

Join1=game⋈g.match_id=go.match_id ∧ g.team_id =go.team_idgame as go
Join2=Join1⋈g.team_id=t.team_idteam
Join3=Join2 LEFT_OUTER_JOIN (go⋈go.team_id=top.team_idteam as top)
Join4=Join3⋈g.player_id=pig.idplayer_in_game
Join5=Join4⋈g.match_id=m.match_idmatches
Join6=Join5⋈g.league_id=tl.league_idteam_league

πg.team_id,go.team_id⟶Opponent_team_id,t.name⟶team_name,top.name⟶
op_team_name,tl.week_of_league,m.result,goals_scored,goals_received,fouls,yellow_cards,red_cards,li
neup

(yg.team_id,go.team_id,t.name,top.name,m.result,g.match_id,m.home_team_id,m.away_team_id,m.lineup_home,m.lineup_away,tl.week_of_league; ∑ (pig.fouls), ∑ (pig.yellow_cards), ∑ (pig.red_cards)(Join6)

```
select
    g.team_id,
    go.team_id as Opponent_team_id,
    t.name as team_name,
    top.name as op_team_name,
    tl.week_of_league,
    m.result,
    case
        when g.team_id = m.home_team_id then substring_index(m.result, '-',1)
        else substring_index(m.result, '-', -1)
    end as goals_scored,
    case
        when g.team_id = m.home_team_id then substring_index(m.result, '-', -1)
        else substring_index(m.result, '-',1)
    end as goals_received,
    sum(pig.fouls) as fouls,
    sum(pig.yellow_cards) as yellow_cards,
    sum(pig.red_cards) as red_cards,
    case
        when g.team_id = m.home_team_id then m.lineup_home
        else m.lineup_away
    end as lineup
from
    game g
left join
    game go on g.match_id = go.match_id and g.team_id != go.team_id
left join
    team t on g.team_id = t.team_id
left join
    team top on go.team_id = top.team_id
join
    player_in_game pig on g.player_id = pig.id
join
    matches m on g.match_id = m.match_id
join
    team_league tl on tl.league_id = g.league_id
group by
    g.team_id,
    go.team_id,
    t.name,
    top.name,
    m.result,
```

```
    g.match_id,
    m.home_team_id,
    m.away_team_id,
    m.lineup_home,
    m.lineup_away,
    tl.week_of_league;
```

2.Team and technical info for each coach

Join1=staff⋈s.id=wf.staff_idworks_for
Join2=Join1⋈wf.team_id=t.team_idteam
Join3=Join2⋈t.team_id=g.team_idgame
Join4=Join3⋈g.match_id=m.match_idmatches
Join5=Join4⋈g.player_id=pig.idplayer_in_game

πs.id,s.name,wf.contract_draft_time⟶
contract_date,wf.contract_termination_date,wf.contract_period,wf.contract_amount,matches,fouls,goal
s
(γs.id,s.name,wf.contract_draft_time,wf.contract_termination_date,wf.contract_period,wf.contract_amo
unt;count(g.match_id), ∑ (pig.fouls), ∑ (goals)(σs.role='Coach'(Join5)))

```
select
    s.id,
    s.name,
    wf.contract_draft_time as contract_date,
    wf.contract_termination_date,
    wf.contract_period,
    wf.contract_amount,
    count(g.match_id) as matches,
    sum(pig.fouls) as fouls,
    sum(case
        when g.team_id = m.home_team_id then cast(substring_index(m.result, '-',1)
as unsigned)
        else cast(substring_index(m.result, '-', -1) as unsigned)
    end) as goals
from
    staff s
join
    works_for wf on s.id = wf.staff_id
join
    team t on t.team_id = wf.team_id
join
```

```
    game g on g.team_id = t.team_id
join
    matches m on g.match_id = m.match_id
join
    player_in_game pig ON g.player_id = pig.id
where
    s.role = 'Coach'
group by
    s.id,
    s.name,
    wf.contract_draft_time,
    wf.contract_termination_date,
    wf.contract_period,
    wf.contract_amount;
```

3.Player technical details that are in a team

```
Join1=player⋈p.id=pf.player_ssnplays_for
Join2=Join1⋈pf.team_id=t.team_idteam

π∗(σpf.contract_termination_date>current_date()(Join2))

select
    *
from
    player p
join
    plays_for pf on p.id = pf.player_ssn
join
    team t on t.team_id = pf.team_id
where pf.contract_termination_date > current_date()
```

4.Coach details that is in a team

```
π∗(Order(σs.role='Coach' /\ wf.contract_termination_date>current_date()(Join2),t.team_id))

select
    *
from
    staff s
join
    works_for wf on wf.staff_id = s.id
join
    team t on wf.team_id = t.team_id
where s.role = 'Coach' and wf.contract_termination_date > current_date()
```

```
order by
    t.team_id
```

5.TS details that are in a team

```
Join1=staff⋈wf.staff_id=s.idworks_for
Join2=Join1⋈wf.team_id=t.team_idteam

π∗(Order(σs.role='Coach' ∧ wf.contract_termination_date>current_date()(Join2),t.team_id))

select
    *
from
    staff s
join
    works_for wf on wf.staff_id = s.id
join
    team t on wf.team_id = t.team_id
where s.role != 'Coach' and wf.contract_termination_date > current_date()
order by
    t.team_id
```

Contracts

1.Players bought by teams

```
Join1=player LEFT OUTER JOINpf.player_ssn=p.id plays_for
Join2=Join1 LEFT OUTER JOINpt.player_ssn=p.id player_transfer
Join3=Join2 LEFT OUTER JOINpt.team_id=nt.team_id ∨ pf.team_id=nt.team_id team  as  nt
Join4=Join3 LEFT OUTER JOINpt.prev_team_id=ot.team_id team  as  ot

πp.id as player_id,p.name as
player_name,p.age,p.address,p.current_shirt_no,p.injury,p.player_overall_score,p.goals,p.fouls,p.most_
played_position,p.red_cards,p.yellow_cards,pf.contract_draft_time,pf.contract_termination_date,pf.cont
ract_period,pf.contract_amount,pt.contract_draft_date as new_contract,nt.team_id as
new_team_id,nt.name as new_team_name,ot.team_id as old_team_id,ot.name as old_team_name
(σnt.team_id=1(Join4))

SELECT
    p.id AS player_id,
    p.name AS player_name,
    p.age,
    p.address,
    p.current_shirt_no,
```

```
    p.injury,
    p.player_overall_score,
    p.goals,
    p.fouls,
    p.most_played_position,
    p.red_cards,
    p.yellow_cards,
    pf.contract_draft_time,
    pf.contract_termination_date,
    pf.contract_period,
    pf.contract_amount,
    pt.contract_draft_date as new_contract,
    nt.team_id AS new_team_id,
    nt.name AS new_team_name,
    ot.team_id AS old_team_id,
    ot.name AS old_team_name
FROM
    player p
LEFT JOIN
    plays_for pf ON pf.player_ssn = p.id
LEFT JOIN
    player_transfer pt ON pt.player_ssn = p.id
LEFT JOIN
    team nt ON (pt.team_id = nt.team_id OR pf.team_id = nt.team_id)
LEFT JOIN
    team ot ON pt.prev_team_id = ot.team_id
where nt.team_id = 1
```

2.Contracts of all players with teams

Join1=player LEFT OUTER JOINpf.player_ssn=p.id plays_for
Join2=Join1 LEFT OUTER JOINpt.player_ssn=p.id player_transfer
Join3=Join2 LEFT OUTER JOIN(pt.team_id=nt.team_id ∨ pf.team_id=nt.team_id) team as nt
Join4=Join3 LEFT OUTER JOINpt.prev_team_id=ot.team_id team as ot

πp.id as player_id,p.name as player_name,p.age,p.address,p.current_shirt_no,p.injury,p.player_overall_score,p.goals,p.fouls,p.most_played_position,p.red_cards,p.yellow_cards,pf.contract_draft_time,pf.contract_termination_date,pf.contract_period,pf.contract_amount,pt.contract_draft_date as new_contract,nt.team_id as new_team_id,nt.name as new_team_name,ot.team_id as old_team_id,ot.name as old_team_name
(σpf.contract_termination_date>'2020-01-01' ∧ pf.contract_termination_date<CURRENT_DATE()(Join4))

```
SELECT
    p.id AS player_id,
```

```
    p.name AS player_name,
    p.age,
    p.address,
    p.current_shirt_no,
    p.injury,
    p.player_overall_score,
    p.goals,
    p.fouls,
    p.most_played_position,
    p.red_cards,
    p.yellow_cards,
    pf.contract_draft_time,
    pf.contract_termination_date,
    pf.contract_period,
    pf.contract_amount,
    pt.contract_draft_date as new_contract,
    nt.team_id AS new_team_id,
    nt.name AS new_team_name,
    ot.team_id AS old_team_id,
    ot.name AS old_team_name
FROM
    player p
LEFT JOIN
    plays_for pf ON pf.player_ssn = p.id
LEFT JOIN
    player_transfer pt ON pt.player_ssn = p.id
LEFT JOIN
    team nt ON (pt.team_id = nt.team_id OR pf.team_id = nt.team_id)
LEFT JOIN
    team ot ON pt.prev_team_id = ot.team_id
where pf.contract_termination_date > '2020-01-01' AND
pf.contract_termination_date < CURRENT_DATE();
```

3.Contracts of TS

```
Join1=staff JOINs.id=wf.staff_id works_for
Join2=Join1 JOINwf.team_id=t.team_id team

πt.team_id,t.name,s.id,s.name,s.role,wf.contract_draft_time,wf.contract_termination_date,wf.contract_
period,wf.contract_amount(σwf.contract_termination_date>'2020−01−01' ∧
wf.contract_termination_date<'2023−01−01'(Join2))

select
    t.team_id,
    t.name,
```

```
    s.id,
    s.name,
    s.role,
    wf.contract_draft_time,
    wf.contract_termination_date,
    wf.contract_period,
    wf.contract_amount
from
    staff s
join
    works_for wf on s.id = wf.staff_id
join
    team t on t.team_id = wf.team_id
where contract_termination_date > '2020-01-01' and contract_termination_date <
'2023-01-01'
```

4.Overall expenses of a team buying players

Join1=team JOINt.team_id=pf.team_id plays_for

πt.team_id,t.name,overall_contracts(γt.team_id,t.name,SUM(pf.contract_amount)→overall_contracts (σpf.contract_termination_date>CURRENT_DATE ∧ pf.contract_draft_time<CURRENT_DATE(Join1)))

```
select
    t.team_id,
    t.name,
    sum(pf.contract_amount) as overall_contracts
from
    team t
join
    plays_for pf on t.team_id = pf.team_id
where
    pf.contract_termination_date > current_date() and pf.contract_draft_time <
current_date()
group by
    t.team_id,
    t.name
```

5.Overall expenses of TS in a team

Join1=team JOINt.team_id=wf.team_id works_for

πt.team_id,t.name,overall_contracts(γt.team_id,t.name,SUM(wf.contract_amount)→overall_contracts (σwf.contract_termination_date>CURRENT_DATE ∧ wf.contract_draft_time<CURRENT_DATE(Join1)))

```
select
    t.team_id,
    t.name,
    sum(wf.contract_amount) as overall_contracts
from
    team t
join
    works_for wf on t.team_id = wf.team_id
where
    wf.contract_termination_date > current_date() and wf.contract_draft_time <
current_date()
group by
    t.team_id,
    t.name
```

League

1.League details

Result = (team_league)

```
select * from team_league
```

2.Games in a league

R1 = (game ⋈team_id = team_id team) ⋈match_id = match_id (game ⋈team_id = team_id team)

R2 = (R1 ⋈ match) ⋈league_id = league_id league

R3 = ((((R2 ⋈ match_stadium) ⋈ stadium) ⋈ tickets_soldby_stadium) ⋈ ticket) ⋈ spectator

R4 = stadium_id, match_id F count spectator_id, sum ticket_price(R3)

R5 = (R4 ⋈ match_referee) ⋈ referee_observer

Result = R5

```
select
    l.league_id,
    l.name as league_name,
    t.name as home_team_name,
    top.name as away_team_name,
    m.result,
    m.match_date_time,
    sta.id as stadium_id,
    sta.name as stadium_name,
```

```
    sta.city,
    count(spec.id) as amount_of_spectators,
    -- ms.amount_of_spectators,
    sum(tic.price) as total_revenue,
    r.id as referee_observer_id,
    r.name as referee_observer_name,
    r.role
from
    game g
join
    league l on l.league_id = g.league_id
join
    team t on t.team_id = g.team_id
join
    matches m on m.match_id = g.match_id
join
    match_stadium ms on ms.match_id = m.match_id
join
    stadium sta on sta.id = ms.stadium_id
join
    ticket_soldby_stadium tss on tss.stadium_id = sta.id
join
    ticket tic on tic.ticket_number = tss.ticket_no
join
    spectator spec on spec.ticket_no = tic.ticket_number
join
    match_referee mr on mr.match_id = m.match_id
join
    referee_observer r on r.id = mr.ref_id
join
    team top on top.team_id = m.away_team_id
group by
    l.league_id,
    l.name,
    t.name,
    top.name,
    m.result,
    m.match_date_time,
    sta.id,
    sta.name,
    sta.city,
    r.id,
    r.name,
    r.role
    -- ms.amount_of_spectators;
```

3.Players and teams in a league

R1 = ((player_in_league ⋈ game) ⋈ league) ⋈ team

```
select
    p.name as player_name,
    p.id as player_id,
    g.league_id,
    t.name as team_name,
    count(g.match_id) as games_played,
    sum(pig.goals) as goals_scored,
    sum(pig.fouls) as fouls,
    sum(pig.yellow_cards) as yellow_cards,
    sum(pig.red_cards) as red_cards,
    avg(pig.player_score) as player_rating
from
    player p
join
    player_in_game pig on p.id = pig.id
join
    game g on g.player_id = pig.id
join
    team t on t.team_id = g.team_id
group by
    p.id,
    g.league_id,
    t.name
```

4.Suspended players

R1 = ((player_in_league ⋈ game) ⋈ league) ⋈ team

R2 = R1 ⋈ team_league ⋈ match

R3 = player_id, week_of_league F count yellow_cards, count red_cards

R4 = σmatch_date_time.week < week_of_league and (yellow_cards % 3 = 0 or red_cards = 1)(R3)

```
SELECT
    p.id,
    p.name AS player_name,
    t.name AS team_name,
    g.league_id
FROM
    player p
```

```
JOIN
    player_in_game pig ON p.id = pig.id
JOIN
    game g ON g.player_id = pig.id
JOIN
    team t ON t.team_id = g.team_id
JOIN
    matches m on g.match_id = m.match_id
WHERE
    pig.red_cards >= 1
GROUP BY
    p.id,
    p.name,
    t.name,
    g.league_id
HAVING
    SUM(pig.yellow_cards) % 3 = 0
    OR MAX(m.match_date_time) <= NOW();
```