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.id​σis\_main\_lineup=1​(player\_in\_game))⋈g.team\_id=t.team\_id​team)⋈g.match\_id=m.match\_id​matches)⋈pig.id=p.id​player)

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.id​player\_in\_game

Join2=Join1⋈g.team\_id=t.team\_idteam

Join3=Join2⋈g.match\_id=m.match\_id​matches

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.id​σpig.goals>0​(player\_in\_game))⋈g.team\_id=t.team\_id​team)

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\_id​team)⋈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.id​player\_in\_game)⋈g.team\_id=t.team\_id​team)⋈pig.id=p.id​player

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\_ssn​plays\_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.id​player\_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\_id​game as go

Join2=Join1⋈g.team\_id=t.team\_idteam

Join3=Join2 LEFT\_OUTER\_JOIN (go⋈go.team\_id=top.team\_id​team as top)

Join4=Join3⋈g.player\_id=pig.idplayer\_in\_game

Join5=Join4⋈g.match\_id=m.match\_id​matches

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,lineup​(γg.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\_id​works\_for

Join2=Join1⋈wf.team\_id=t.team\_idteam

Join3=Join2⋈t.team\_id=g.team\_id​game

Join4=Join3⋈g.match\_id=m.match\_idmatches

Join5=Join4⋈g.player\_id=pig.id​player\_in\_game

πs.id,s.name,wf.contract\_draft\_time→contract\_date,wf.contract\_termination\_date,wf.contract\_period,wf.contract\_amount,matches,fouls,goals​(γs.id,s.name,wf.contract\_draft\_time,wf.contract\_termination\_date,wf.contract\_period,wf.contract\_amount;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\_ssn​plays\_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.id​works\_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.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​(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();