

CHECK OUT THE LAST 2 PAGES FOR ANSWERS

Idk, just chuck the question and your answer and then debate about ig

Heads-up, this doc will be view-only for the duration of the test tonight, as will the study doc

Question 1:

Fetch all players that play for 'Washington State' University.

Expected Columns:

PLAYER_ID,PLAYER_SURNAME,PLAYER_NAME,PLAYER_DOB,UNIVERSITY

```
select PLAYER_ID, PLAYER_SURNAME, PLAYER_NAME, PLAYER_DOB, UNIVERSITY
from PLAYER
where UNIVERSITY = 'Washington State';
```

```
select *
from player
where university = 'Washington';
```

PLAYER_ID	PLAYER_SURNAME	PLAYER_NAME	PLAYER_D	UNIVERSITY
630	Goldson	Dashon	18/09/84	Washington
830	Huard	Damon	09/07/73	Washington
24	Alexis	Rich	06/03/81	Washington
95	Barnes	Khalif	21/04/82	Washington
228	Bruener	Mark	16/09/72	Washington
229	Brunell	Mark	17/09/70	Washington
350	Cooper	Marquis	11/03/82	Washington
912	Johnson	Tank	07/12/81	Washington
1001	Kreutz	Olin	09/06/77	Washington
1906	Williams	Reggie	17/05/83	Washington
1060	Looker	Dane	05/05/76	Washington

This is wrong by the way, the university is Washington State, not Washington.

```
SELECT PLAYER_ID,PLAYER_SURNAME,PLAYER_NAME,PLAYER_DOB,UNIVERSITY
FROM player
WHERE university='Washington State'
```

```
566,Frampton,Eric,1984-02-06,Washington State
714,Hanson,Jason,1970-06-17,Washington State
742,Harrison,Jerome,1983-02-26,Washington State
790,Hill,Jason,1985-02-20,Washington State
179,Brackenridge,Tyron,1984-06-30,Washington State
324,Coleman,Erik,1982-05-06,Washington State
403,Darling,Devard,1982-04-16,Washington State
405,David,Jason,1982-06-12,Washington State
1,Abdullah,Hamza,1983-08-20,Washington State
137,Bienemann,Troy,1983-02-19,Washington State
-Jdm
```

Question 2:

Fetch the name(s) of the newest coach(es)

Expected Columns:

COACH_NAME,COACH_SURNAME

Kind of confused about how we define 'newest' ngl, but i guess just from the most recent year?

---M

```
select coach_name, coach_surname, year_starting
```

```
from coach
```

```
join team_coach on coach.coach_id = team_coach.coach_id
```

```
where year_starting = (select Max(year_starting)
                        from team_coach)
```

--msa

```
SELECT COACH.COACH_NAME, COACH.SURNAME
FROM COACH, TEAM_COACH
WHERE TEAM_COACH.YEAR_STARTING = (SELECT MAX(YEAR_STARTING) FROM TEAM_COACH)
AND TEAM_COACH.COACH_ID = COACH.COACH_ID;
```

```
SELECT COACH_NAME, COACH_SURNAME
FROM COACH join TEAM_COACH TC on COACH.COACH_ID = TC.COACH_ID
WHERE YEAR_STARTING= (SELECT max(YEAR_STARTING) FROM team_coach)
```

Mike,Smith
Wade,Phillips
Tony,Sparano
-Jdm

Question 3:

Fetch the names of the players that have changed teams at least once. List the number of teams they have played for. List the player(s) with the most team changes first.

Expected Columns:

PLAYER_NAME, PLAYER_SURNAME, NUM_TEAMS

```
select coach.coach_name, coach.coach_surname, team_coach.year_starting
from coach
join team_coach
on coach.coach_id = team_coach.coach_id
order by team_coach.year_starting desc
FETCH FIRST 2 ROWS ONLY;
```

---M

```
select PLAYER_NAME, PLAYER_SURNAME, count(*)-1 as NUM_TEAMS
from PLAYER
join PLAYER_TEAM on PLAYER.player_id = PLAYER_TEAM.player_id
group by PLAYER_NAME, PLAYER_SURNAME
having count(*)-1 >= 1
```

---msa

```
SELECT PLAYER_NAME, PLAYER_SURNAME, COUNT(*) AS NUM_TEAMS
FROM PLAYER, PLAYER_TEAM
WHERE PLAYER.PLAYER_ID = PLAYER_TEAM.PLAYER_ID
GROUP BY PLAYER_NAME, PLAYER_SURNAME
HAVING COUNT(*) > 1
ORDER BY NUM_TEAMS DESC;
```

```
SELECT PLAYER_NAME, PLAYER_SURNAME, COUNT(P.PLAYER_ID) AS NUM_TEAMS
FROM PLAYER_TEAM, PLAYER P
WHERE P.PLAYER_ID=PLAYER_TEAM.PLAYER_ID
GROUP BY P.PLAYER_NAME, P.PLAYER_SURNAME
HAVING COUNT(P.PLAYER_ID) > 1
ORDER BY NUM_TEAMS DESC
```

Travis,Taylor,3
Michael,Bennett,2
Amon,Gordon,2
Otis,Grigsby,2
Curome,Cox,2
Lamont,Thompson,2
Matt,McCoy,2
Anthony,Mix,2
Shantee,Orr,2
Dave,Rayner,2
-Jdm

Question 4:

Generate a list of stadiums where at least one game has been played. For each stadium find the largest score differential between visiting and local teams for that stadium.

Expected Columns:

STADIUM_NAME,DELTA

---M

```
/*Generate a list stadiums where at least one game has been played. For each stadium
find the largest score differential between visiting and local teams for in that stadium.*/
select STADIUM_NAME,max(abs(local_score-visitor_score)) as DELTA
from game
join team_stadium on game.local_team_id = team_stadium.team_id
join stadium on team_stadium.stadium_id = stadium.stadium_id
```

group by STADIUM_NAME

---msa

```
SELECT STADIUM_NAME, MAX(ABS(VISITOR_SCORE-LOCAL_SCORE)) AS DELTA
FROM STADIUM, TEAM_STADIUM, GAME
WHERE STADIUM.STADIUM_ID = TEAM_STADIUM.STADIUM_ID AND GAME.LOCAL_TEAM_ID =
TEAM_STADIUM.TEAM_ID
GROUP BY STADIUM_NAME
```

```
SELECT STADIUM_NAME,MAX(ABS(VISITOR_SCORE-GAME.LOCAL_SCORE)) DELTA
FROM GAME
  JOIN TEAM T on T.TEAM_ID = GAME.LOCAL_TEAM_ID
  JOIN TEAM_STADIUM TS ON t.TEAM_ID=ts.TEAM_ID
  JOIN STADIUM S on TS.STADIUM_ID = S.STADIUM_ID
GROUP BY S.STADIUM_NAME
ORDER BY MAX(ABS(VISITOR_SCORE-GAME.LOCAL_SCORE)) DESC
Gillette Stadium,59
Qwest Field,58
Superdome,55
Edward Jones Dome,52
Arrowhead Stadium,49
Lambeau Field,49
Ralph Wilson,46
Ford Field,45
M&T Bank Stadium,45
Monster Park,45
-Jdm
```

Correct me if i was wrong, here is the output:

Reliant Stadium	35
Soldier Field II	33
Cleveland Browns Stadium	41
Gillette Stadium	59
Edward Jones Dome	52
University of Phoenix	40
Texas Stadium	37
Lincoln Financial Field	43
RCA Dome	35
Arrowhead Stadium	49

```

select s.STADIUM_NAME, abs(local_score - visitor_score) AS DELTA
from ((game g join team t on g.local_team_id = t.team_id)
join team_stadium ts on t.team_id = ts.team_id)
join stadium s on s.stadium_id = ts.stadium_id)
order by DELTA desc

```

STADIUM_NAME	DELTA
-----	-----
Gillette Stadium	59
Qwest Field	58
Superdome	55
Edward Jones Dome	52
Lambeau Field	49
Arrowhead Stadium	49

Question 5:

Find the average score of visiting and local teams for games played on 'Natural' turf. Round both averages to two decimal places.

Expected Columns:

VISITOR,LOCAL

Thoughts so far:

```

select avg(visitor_score) as visitor, avg(local_score) as local
from game
where (somehow get connection to turf)

```

No clue how to get that connection

```

SELECT AVG(VISITOR_SCORE), AVG(LOCAL_SCORE) FROM STADIUM
JOIN TEAM_CITY ON TEAM_CITY.CITY_ID=STADIUM.CITY_ID
JOIN GAME ON GAME.LOCAL_TEAM_ID=TEAM_CITY.TEAM_ID
WHERE TURF='Natural';

```

Resulting columns: 20.58, 22.77

I think this might be the answer? yes

```

select ROUND(AVG(VISITOR_SCORE),2) AS VISITOR, ROUND(AVG(LOCAL_SCORE),2) AS LOCAL
from game
where game.local_team_id in
(
select team_id
from stadium s join team_stadium ts on s.stadium_id = ts.stadium_id
where s.turf = 'Natural'
)

```

VISITOR	LOCAL
20.58	22.77

This was question 5 for me:

Fetch the names of the players that have changed teams at least once. List the number of teams they have played for. List the player(s) with the most team changes first.

Expected Columns:

PLAYER_NAME,PLAYER_SURNAME,NUM_TEAMS

Stuck with this one, not sure whether I'm right trying to count team id. SQL is complaining about the group by

```

select player_name, player_surname, count(*) as num_teams
from player
join player_team
on player.player_id = player_team.player_id
group by player_team.team_id
having count(*) >=1;

```

Not 100% sure but:

```

SELECT ROUND(AVG(VISITOR_SCORE), 2) AS VISITOR, ROUND(AVG(LOCAL_SCORE), 2)
AS LOCAL
FROM GAME
JOIN TEAM_STADIUM ON GAME.LOCAL_TEAM_ID = TEAM_STADIUM.TEAM_ID
JOIN STADIUM ON STADIUM.STADIUM_ID = TEAM_STADIUM.STADIUM_ID

```

```
WHERE TURF = 'Natural';
```

```
SELECT ROUND(AVG(VISITOR_SCORE), 2) VISITOR, ROUND(AVG(LOCAL_SCORE), 2)
LOCAL
FROM GAME
  JOIN TEAM T on T.TEAM_ID = GAME.LOCAL_TEAM_ID
  JOIN TEAM_STADIUM TS ON t.TEAM_ID=ts.TEAM_ID
  JOIN STADIUM S on TS.STADIUM_ID = S.STADIUM_ID
WHERE S.TURF='Natural'
20.58,22.77
```

-Jdm

Question 6:

How many cities are not associated with a team?

Expected Columns:

NO_TEAM_COUNT

```
select COUNT(*) as NO_TEAM_COUNT
from CITY
where city_id not in (select city_id
                      from team_city);
```

-- alternative:

```
select COUNT(*) as NO_TEAM_COUNT
from CITY left outer join TEAM_CITY on CITY.CITY_ID = TEAM_CITY.CITY_ID
where TEAM_ID is null;
```

The one above is the same way I did it :) I got 29 in the output

----M

```
select count(*) as NO_TEAM_COUNT
from (select city_name
from city
where not exists (select city_id from team_city where city.city_id = team_city.city_id))
```

```
SELECT COUNT(*) NO_TEAM_COUNT
FROM CITY LEFT JOIN TEAM_CITY TC on CITY.CITY_ID = TC.CITY_ID
WHERE TC.TEAM_ID IS NULL
29
-Jdm
```


<pre>select count(*) AS NO_TEAM_COUNT from city where city.city_id not in (select city_id from team_city)</pre>	<table><tr><th>NO_TEAM_COUNT</th></tr><tr><td>29</td></tr></table>	NO_TEAM_COUNT	29
NO_TEAM_COUNT			
29			

Wonder if the questions have been randomized, this is the Question 7 I got

This one should be 6:)

Generate a list stadiums where at least one game has been played. For each stadium find the largest score differential between visiting and local teams for in that stadium.

Expected Columns:

STADIUM_NAME, DELTA

Have no idea where to get the number of games that have been played in a stadium.

Question 7:

Get a list of cites that have multiple teams. Present the list in alphabetical order.

Expected Columns:

CITY_NAME

---M

*/*Get a list of cites that have multiple teams. Present the list in alphabetical order.*/*

*/*select team_name*

from team/*

select city_name

from city

join team_city on team_city.city_id = city.city_id

join team on team_city.team_id = team.team_id

group by city_name

having count() >= 2*

order by city_name asc

I did the same just without the second join?

select COUNT() as city_name*

from CITY

join TEAM_CITY

on CITY.CITY_ID = TEAM_CITY.CITY_ID

group by city_name

```
having count(*) >= 2
order by city_name asc;
```

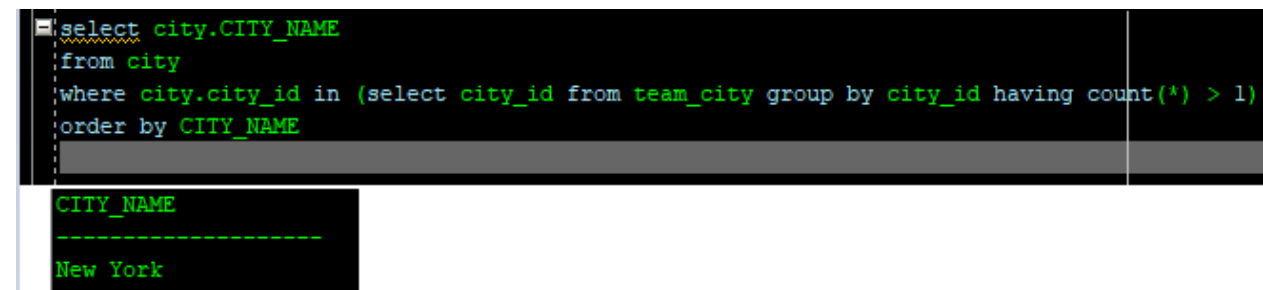
--msa

```
SELECT DISTINCT CITY.CITY_NAME
FROM TEAM_CITY, CITY
WHERE TEAM_CITY.CITY_ID = (SELECT TEAM_CITY.CITY_ID FROM TEAM_CITY GROUP BY
TEAM_CITY.CITY_ID HAVING COUNT(*) > 1) AND TEAM_CITY.CITY_ID = CITY.CITY_ID
ORDER BY CITY.CITY_NAME ASC;
```

```
SELECT CITY_NAME
FROM CITY
JOIN TEAM_CITY T on CITY.CITY_ID = T.CITY_ID
GROUP BY CITY_NAME
HAVING COUNT(CITY_NAME) > 1
```

New York

-Jdm



```
select city.CITY_NAME
from city
where city.city_id in (select city_id from team_city group by city_id having count(*) > 1)
order by CITY_NAME
```

CITY_NAME
New York

Question 8:

List coaches (if any) of teams with same number of wins as losses. Do not include any teams that are yet to play a game.

Expected Columns:

COACH_NAME,COACH_SURNAME

----M

/*List coaches (if any) of teams with same number of wins as losses. Do not include any teams that are yet to play a game.*/

```
select COACH_NAME,COACH_SURNAME
from coach
join team_coach on coach.coach_id = team_coach.coach_id
join team on team_coach.team_id = team.team_id
where wins = losses
```

```

select distinct coach_name, coach_surname
from coach
join team_coach on coach.coach_id = team_coach.coach_id
join team on team_coach.team_id = team.team_id
join player_team on team.team_id = player_team.team_id
where team.wins = team.losses and player_team.games_played > 0;

```

COACH_NAME	COACH_SURNAME
Brad	Childress
Ken	Whisenhunt
Andy	Reid
Gary	Kubiak

--msa

```

SELECT COACH_NAME, COACH_SURNAME
FROM COACH, TEAM, TEAM_COACH
WHERE TEAM.WINS = TEAM.LOSSES AND TEAM.WINS != 0 AND TEAM.LOSSES != 0 AND
COACH.COACH_ID = TEAM_COACH.COACH_ID
AND TEAM.TEAM_ID = TEAM_COACH.TEAM_ID;

```

```

SELECT COACH_NAME, COACH_SURNAME
FROM COACH
  JOIN TEAM_COACH TC on COACH.COACH_ID = TC.COACH_ID
  JOIN TEAM T on TC.TEAM_ID = T.TEAM_ID
WHERE WINS=LOSSES and WINS != 0
Ken,Whisenhunt
Gary,Kubiak
Brad,Childress
Andy,Reid
-Jdm

```

```
select COACH_NAME, COACH_SURNAME
from coach c natural join team_coach tc
where tc.team_id in (select team_id from team where team.wins = team.losses and team.wins > 0)
```

COACH_NAME	COACH_SURNAME
Ken	Whisenhunt
Gary	Kubiak
Brad	Childress
Andy	Reid

Question 9:

Find the offensive line player with highest PANCAKES for that DIVISION and CONFERENCE, order the player names from most to least pancakes.

Expected Columns: PLAYER_NAME, PLAYER_SURNAME

Some where towards the solution was where I got:

```
SELECT PLAYER_NAME, PLAYER_SURNAME
FROM OFFENSIVE_LINE
JOIN PLAYER ON PLAYER.PLAYER_ID=OFFENSIVE_LINE.PLAYER_ID
JOIN PLAYER_TEAM ON PLAYER.PLAYER_ID=PLAYER_TEAM.PLAYER_ID
JOIN TEAM ON PLAYER_TEAM.TEAM_ID=TEAM.TEAM_ID
GROUP BY DIVISION, CONFERENCE
HAVING COUNT(OFFENSIVE_LINE.PANCAKES);
ORDER BY PANCAKES;
```

Pls help

-- even liam thought this was was really hard lol, this is my answer

```
select PLAYER_NAME, PLAYER_SURNAME
from OFFENSIVE_LINE
    natural join PLAYER
    natural join PLAYER_TEAM
    natural join TEAM T1
where pancakes = (select MAX(pancakes)
    from OFFENSIVE_LINE
    natural join PLAYER
    natural join PLAYER_TEAM
    natural join TEAM T2
    where T2.DIVISION = T1.DIVISION
    and T2.CONFERENCE = T1.CONFERENCE)
order by PANCAKES desc;
```

This makes so much sense yesss

```
SELECT PLAYER_NAME, PLAYER_SURNAME
FROM (PLAYER
      JOIN OFFENSIVE_LINE OL ON PLAYER.PLAYER_ID = OL.PLAYER_ID
      JOIN PLAYER_TEAM PT on PLAYER.PLAYER_ID = PT.PLAYER_ID
      JOIN TEAM T on PT.TEAM_ID = T.TEAM_ID)
WHERE (PANCAKES, DIVISION, CONFERENCE) IN (
      SELECT MAX(PANCAKES), DIVISION, CONFERENCE
      FROM (PLAYER
            JOIN OFFENSIVE_LINE OL ON PLAYER.PLAYER_ID = OL.PLAYER_ID
            JOIN PLAYER_TEAM PT on PLAYER.PLAYER_ID = PT.PLAYER_ID
            JOIN TEAM T on PT.TEAM_ID = T.TEAM_ID)
      GROUP BY DIVISION, CONFERENCE
      )
ORDER BY PANCAKES DESC
```

Brian,Waters
Cullen,Loeffler
Ryan,Cook
Adam,Snyder
Pete,Kendall
Marshall,Yanda
Vincent,Manuwai
Rex,Hadnot
Dan,Buenning

-Jdm

HELLO TO ALL THE ANSWERS BELOW ARE CORRECT - CROSS CHECKED BY DANITA

```
-- Generate a list stadiums where at least one game has been played. For each stadium find the largest score differential between visiting and local teams for in that stadium.
```

```
SELECT S.STADIUM_NAME, max(abs(G.LOCAL_SCORE - G.VISITOR_SCORE)) AS delta
FROM GAME G JOIN TEAM LT ON G.LOCAL_TEAM_ID = LT.TEAM_ID JOIN TEAM STADIUM TS
ON TS.TEAM_ID = LT.TEAM_ID JOIN STADIUM S ON S.STADIUM_ID = TS.STADIUM_ID
GROUP BY S.STADIUM_NAME, S.STADIUM_ID;
```

```
-- Fetch all players that play for 'Washington State' University.
```

```
SELECT PLAYER_ID, PLAYER_SURNAME, PLAYER_NAME, PLAYER_DOB, UNIVERSITY
FROM PLAYER
WHERE UNIVERSITY LIKE 'Washington State';
```

```
-- Fetch the name(s) of the newest coach(es)
```

```
SELECT COACH_NAME, COACH_SURNAME
FROM COACH, TEAM_COACH
WHERE COACH.COACH_ID = TEAM_COACH.COACH_ID AND TEAM_COACH.YEAR_STARTING =
(SELECT MAX(YEAR_STARTING) FROM TEAM_COACH );
```

```
-- Fetch the names of the players that have changed teams at least once. List the number of teams they have played for. List the player(s) with the most team changes first.
```

```
SELECT P.PLAYER_NAME, P.PLAYER_SURNAME, count(PT.TEAM_ID)
FROM PLAYER P JOIN PLAYER_TEAM PT ON P.PLAYER_ID = PT.PLAYER_ID
```

```
WHERE P.PLAYER_ID = pt.PLAYER_ID
GROUP BY P.PLAYER_NAME, P.PLAYER_SURNAME
HAVING count(PT.TEAM_ID) > 1;
```

```
-- Find the average score of visiting and local teams for games played on
'Natural' turf. Round both averages to two decimal places.
```

```
SELECT ROUND(AVG(LOCAL_SCORE) , 2) AS LOCAL, ROUND(AVG(VISITOR_SCORE), 2) AS
VISITOR
FROM GAME G, team T, TEAM_STADIUM TS, STADIUM S
WHERE T.TEAM_ID = TS.TEAM_ID AND TS.STADIUM_ID = S.STADIUM_ID AND S.TURF LIKE
'Natural';
```

```
-- How many cities are not associated with a team?
SELECT count(*) AS NO_CITY_COUNT
FROM CITY
WHERE CITY_ID NOT IN (SELECT CT.City_ID FROM TEAM_CITY CT);
```

```
-- Get a list of cities that have multiple teams. Present the list in
alphabetical order.
SELECT C.CITY_NAME
FROM CITY C
WHERE 1 < (SELECT COUNT(TEAM_ID) FROM TEAM_CITY WHERE TEAM_CITY.CITY_ID =
C.CITY_ID)
GROUP BY C.CITY_NAME
ORDER BY c.CITY_NAME;
```

```
-- List coaches (if any) of teams with same number of wins as losses. Do not
include any teams that are yet to play a game.
```

```
SELECT C.COACH_NAME, C.COACH_SURNAME
FROM COACH C JOIN TEAM_COACH TC on C.COACH_ID = TC.COACH_ID JOIN TEAM T on
TC.TEAM_ID = T.TEAM_ID
WHERE T.WINS != 0 AND T.WINS = T.LOSSES
```

```
-- Find the offensive line player with highest PANCAKES for that DIVISION and
CONFERENCE, order the player names from most to least pancakes.
SELECT P.PLAYER_NAME, P.PLAYER_SURNAME, op.PANCAKES
FROM PLAYER P join OFFENSIVE_LINE OP ON P.PLAYER_ID = OP.PLAYER_ID JOIN
PLAYER_TEAM PT ON P.PLAYER_ID = PT.PLAYER_ID JOIN TEAM T on PT.TEAM_ID =
T.TEAM_ID
```

```
WHERE OP.PANCAKES = (SELECT MAX(PANCAKES)
FROM OFFENSIVE_LINE Join player team on
OFFENSIVE_LINE.PLAYER_ID = PLAYER_TEAM.PLAYER_ID JOIN team ON team.TEAM_ID =
PLAYER_TEAM.TEAM_ID
WHERE team.DIVISION = t.DIVISION AND team.CONFERENCE =
t.CONFERENCE)
order by PANCAKES desc;
```

GOOD LUCK - (Zahid khan)