Question 4:

|  |  |
| --- | --- |
| Nested Query | Solution |
| SELECT DISTINCT P. playerId  FROM Player P  WHERE (  SELECT COUNT (G.id)  FROM Game G  WHERE G. playerId = P. playerId  ) >10 | SELECT DISTINCT P.name, COUNT(G.playerId)  FROM Player P INNER JOIN Game G  ON P.playerId = G.playerId  GROUP BY G.playerId  HAVING COUNT(G.playerId) > 10 |
| SELECT DISTINCT P.name , ( SELECT  count (\*)  FROM Game G  WHERE P. playerId = G. playerId )  FROM Player P | SELECT DISTINCT P.name, COUNT(G.playerId)  FROM Player P LEFT OUTER JOIN Game G  ON P.playerId = G.playerId  GROUP BY G.playerId |

In Query 1, for each tuple in relation Player P, the nested count query will execute. This is an example of Type JA (dependent and aggregation) nested query.

Also, in both cases, sub queries are dependent on outer query.

To un-nest first query, we used INNER JOIN, because we want to list all the distinct players and their #played games, who has played more than 10 games. Here LEFT JOIN would not be necessary because players who have not played any game would be truncated by the HAVING clause.

To un-nest second query, we used LEFT OUTER JOIN, because we want to list all the distinct players (all from left table) and their #played games (only matching right table, null values will be replaced as 0 by aggregate function), even those who have not played any game.