/\* Напишите SQL запрос который возвращает количество игроков, сгруппированных по классу роста игрока. Результат отсортируйте по классу игрока. Формула распределения по классам представлена ниже как псевдокод. ЕСЛИ рост < 190 ТОГДА 1 ЕСЛИ рост >= 190 И рост < 200 ТОГДА 2 В ОСТАЛЬНЫХ СЛУЧАЯХ 3 Пример результата ниже \*/

SELECT  
 CASE when height < 190 then 1  
 when (height >= 190 AND height < 200) then 2  
 else 3  
 END  
 AS height\_class, *count*(id) AS amount\_players  
FROM player  
GROUP BY height\_class  
ORDER BY height\_class;

A screenshot of a computer

Description automatically generated with medium confidence

/\* Видоизмените SQL запрос из задания a), получив дополнительно строку по общему количеству игроков из каждого класса. Отсортируйте результат по классу игрока и NULL значения должны быть в начале списка. Пример результата ниже. \*/

SELECT  
 CASE when height < 190 then 1  
 when (height >= 190 AND height < 200) then 2  
 else 3  
 END  
 AS height\_class, *count*(id) AS amount\_players FROM player  
union  
SELECT NULL AS height\_class, *count*(id) as amount\_players FROM player  
GROUP BY height\_class  
ORDER BY height\_class NULLS first;

Graphical user interface, text, application

Description automatically generated

/\*Выведите одним SQL запросом агрегированные данные по позиции игрока и его команде, получив информацию в следующих срезах - количество игроков играющих на определенной позиции в каждой из команд - количество игроков играющих на одной позиции на площадке - общее количество игроков - номер внутренней группы агрегации по игрокам (функция GROUPING … ) Результат отсортируйте по номеру внутренней группы, наименованию позиции игрока и имени команды\*/

SELECT p.position, t.name AS team\_name, *count*(p.id) AS amount\_players,  
 *dense\_rank*() OVER (ORDER BY t.name IS NOT NULL DESC, p.position IS NOT NULL DESC) AS "group"  
FROM team t JOIN player p ON t.id = p.team\_id GROUP BY GROUPING SETS((t.name, p.position), p.position, ())  
ORDER BY "group", p.position, t.name;

A picture containing graphical user interface

Description automatically generated

/\*Выведите при помощи SQL запроса ранжированные зарплаты всех защитников из всех команд. Результат отсортируйте по ранжированному значению и затем по имени игрока \*/

SELECT *dense\_rank*() OVER (ORDER BY salary) AS "ranking", p.name, p.salary FROM player p WHERE position = 'защитник'  
ORDER BY "ranking", p.name;

Graphical user interface, text

Description automatically generated