**Гурьянов Артем Игоревич ОРПУД 1.1 №8**

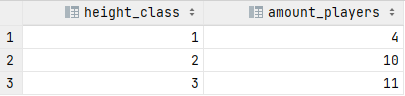
a. Напишите SQL запрос который возвращает количество игроков, сгруппированных по классу роста игрока. Результат отсортируйте по классу игрока. Формула распределения по классам представлена ниже как псевдокод.

ЕСЛИ рост < 190 ТОГДА 1

ЕСЛИ рост >= 190 И рост < 200 ТОГДА 2

В ОСТАЛЬНЫХ СЛУЧАЯХ 3

SELECT CASE WHEN height < 190 THEN 1  
 WHEN 190 <= height AND height < 200 THEN 2  
 ELSE 3 END  
 AS height\_class,  
 *count*(*\**) AS amount\_players  
FROM player  
GROUP BY height\_class  
ORDER BY height\_class;



b. 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 190 <= height AND height < 200 THEN 2  
 ELSE 3 END  
 AS height\_class,  
 *count*(*\**) AS amount\_players  
FROM player  
GROUP BY  
 GROUPING SETS (  
 (),  
 (height\_class)  
 )  
ORDER BY height\_class NULLS FIRST;

Изображение выглядит как стол

Автоматически созданное описание

c. 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.

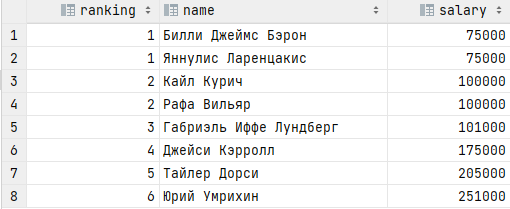
SELECT position,  
 team.name AS team\_name,  
 *count*(*\**) AS amount\_players,  
 *GROUPING*(position, team.name) AS "group"  
FROM player  
JOIN team ON team.id = player.team\_id  
GROUP BY  
 GROUPING SETS (  
 (position, team.name),  
 (position),  
 ()  
 )  
ORDER BY "group", position, team\_name;

Изображение выглядит как стол

Автоматически созданное описание

d. 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,  
 name, salary  
FROM player  
WHERE position = 'защитник'  
ORDER BY ranking, name;



e. (\* - опциональное) Выведите при помощи SQL запроса группировку по названию команды и позиции игрока в ней, если количество игроков на одной позиции в команде строго больше 1 и разница между максимальной и минимальной зарплатами всех игроков в команде строго больше 150 000 рублей

При выполнении задания в точном соответствии с его формулировкой получается такой результат:

WITH high\_salary\_difference\_teams AS (  
 SELECT team.id AS id, team.name AS name  
 FROM team  
 JOIN player ON team.id = player.team\_id  
 GROUP BY team.id, team.name  
 HAVING *max*(salary) - *min*(salary) > 150000  
)  
SELECT hsdt.name AS name,  
 position  
FROM high\_salary\_difference\_teams hsdt  
JOIN player ON hsdt.id = player.team\_id  
GROUP BY hsdt.name, position  
HAVING *count*(*\**) > 1;

Изображение выглядит как стол

Автоматически созданное описание

Но при этом не сходится ответ.

В этих командах действительно разница между максимальной и минимальной зарплатами **всех игроков в команде** (не тех, которые занимают повторяющиеся позиции) строго больше 150 000 рублей, и в то же время есть повторяющиеся позиции.

Сходится с ответом результат такого запроса:

SELECT team.name AS name,  
 position  
FROM team  
JOIN player ON team.id = player.team\_id  
GROUP BY team.name, position  
HAVING *count*(*\**) > 1 AND *max*(salary) - *min*(salary) > 150000;



В этом запросе возвращаются только команды и повторяющиеся позиции, где разница между максимальной и минимальной зарплатами **игроков, занимающих одну и ту же позицию** строго больше 150 000 рублей.