**Гурьянов Артем Игоревич ОРПУД 1.1 №7**

a. Напишите SQL запрос который возвращает средний, максимальный, минимальный рост всех игроков, а также суммарную их зарплату. Обратите внимание на ответ ниже с учетом именования выходных атрибутов вашего запроса.

SELECT *avg*(height) AS avg\_height,  
 *min*(height) AS min\_height,  
 *max*(height) AS max\_height,  
 *sum*(salary) AS total\_salary  
FROM player;



b. Перепишите SQL запрос из пункта a) в разрезе группировки данных по позиции игрока на игровом поле. Результат отсортируйте по позиции игрока. Округление по среднему росту сделайте до 2 знаков после запятой. Обратите внимание на часть ответа ниже с учетом именования выходных атрибутов вашего запроса.

SELECT position,  
 *round*(*avg*(height), 2) AS avg\_height,  
 *min*(height) AS min\_height,  
 *max*(height) AS max\_height,  
 *sum*(salary) AS total\_salary  
FROM player  
GROUP BY position  
ORDER BY position;

Изображение выглядит как текст

Автоматически созданное описание

c. Напишите SQL запрос который возвращает количество побед команд. Результат отсортируйте в убывающем списке по количеству побед. Обратите внимание на часть ответа ниже с учетом именования выходных атрибутов вашего запроса.

Если нужно возвращать только команды, имеющие победы:

SELECT name, *count*(*\**) AS count\_wins  
FROM team  
 JOIN game ON team.id = game.winner\_team\_id  
GROUP BY name  
ORDER BY count\_wins DESC;

Изображение выглядит как стол

Автоматически созданное описание

Если команды, не имеющие побед, тоже нужно возвращать:

SELECT name, *count*(winner\_team\_id) AS count\_wins  
FROM team  
 LEFT JOIN game ON team.id = game.winner\_team\_id  
GROUP BY name  
ORDER BY count\_wins DESC;

Изображение выглядит как стол

Автоматически созданное описание

d. Напишите SQL запрос который возвращает города, встречающиеся более чем один раз, используя конструкцию GROUP BY. Обратите внимание на ответ на задачу ниже с учетом именования выходных атрибутов вашего запроса.

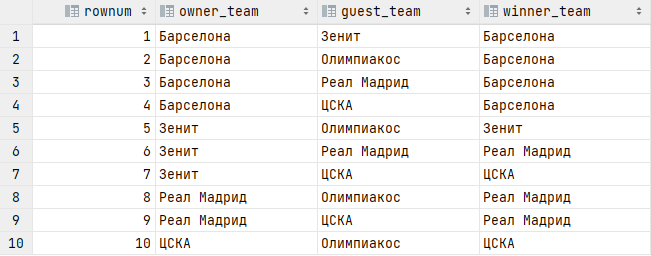
SELECT city FROM team  
GROUP BY city  
HAVING *count*(city) > 1;

Изображение выглядит как текст

Автоматически созданное описание

e. (\*) Напишите SQL запрос который возвращает информацию по играм между командами с указанием имени команды хозяина, имени гостевой команды , имени команды победителя и номера строки возвращаемого набора данных. Результат отсортируйте по имени команды хозяина и по имени гостевой команды. Обратите внимание на часть ответа ниже с учетом именования выходных атрибутов вашего запроса и на последовательный вывод номера строки!

WITH query AS (  
 SELECT owner\_team.name AS owner\_team,  
 guest\_team.name AS guest\_team,  
 CASE WHEN game.winner\_team\_id = game.owner\_team\_id  
 THEN owner\_team.name  
 ELSE guest\_team.name END AS winner\_team  
 FROM game  
 JOIN team owner\_team ON owner\_team.id = game.owner\_team\_id  
 JOIN team guest\_team ON guest\_team.id = game.guest\_team\_id  
 ORDER BY owner\_team, guest\_team  
)  
SELECT *row\_number*() OVER () AS rownum,  
 owner\_team, guest\_team, winner\_team  
FROM query;



f. (\*-задача опциональная) Подсчитайте общее количество очков забитых обеими командами и сгруппированными в разрезе команды победителя. Результат отсортируйте по общему количеству очков. Обратите внимание на часть ответа ниже с учетом именования выходных атрибутов вашего запроса.

SELECT *sum*(*split\_part*(score, ':', 1)::int  
 + *split\_part*(score, ':', 2)::int)  
 AS total\_points,  
 winner\_team.name AS name  
FROM game  
JOIN team winner\_team ON winner\_team.id = game.winner\_team\_id  
GROUP BY name  
ORDER BY total\_points;

Изображение выглядит как стол

Автоматически созданное описание