**Гурьянов Артем Игоревич ОРПУД 1.1 №4**

a. Напишите SQL запрос, который возвращает имена стадионов и имена команд в одном списке. Результат отсортируйте в убывающем порядке.

SELECT name FROM team  
UNION  
SELECT name FROM arena  
ORDER BY name DESC;

Изображение выглядит как текст

Автоматически созданное описание

b. Напишите SQL запрос, который возвращает имена стадионов и имена команд в одном списке, но с типом, что запись является или “стадион” или “команда”. Результат отсортируйте в убывающем порядке по типу записи и потом по имени

SELECT name, 'команда' AS object\_type FROM team  
UNION  
SELECT name, 'стадион' FROM arena  
ORDER BY object\_type DESC, name;

Изображение выглядит как текст

Автоматически созданное описание

c. (\*) Напишите SQL запрос, который возвращает имена и зарплаты игроков в отсортированном списке по зарплатам в возрастающем порядке, НО зарплата со значением 475 000 должна быть на первом месте. Ограничьте вывод данных до 5 строк

SELECT name, salary FROM player  
ORDER BY  
 CASE WHEN salary = 475000 THEN -1  
 ELSE salary END  
LIMIT 5;

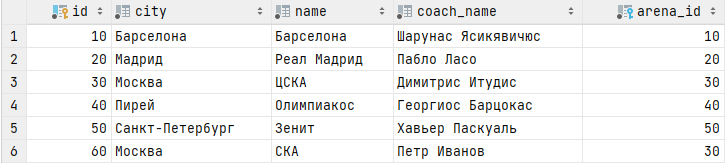
Изображение выглядит как стол

Автоматически созданное описание

d. Напишите DML запрос, который создает новую команду со следующими характеристиками:

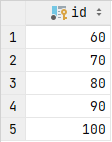
* id = 60
* город команды - Москва
* название команды - СКА
* тренер команды - Петр Иванов
* играет команда на том же стадионе что и команда ЦСКА

INSERT INTO team  
 SELECT 60, 'Москва', 'СКА', 'Петр Иванов', arena\_id  
 FROM team  
 WHERE name = 'ЦСКА';



e. Напишите SQL скрипт, который подсчитывает симметрическую разницу множеств A и B.

SELECT id FROM arena  
EXCEPT  
SELECT id FROM game  
UNION  
(SELECT id FROM game  
EXCEPT  
SELECT id FROM arena)  
ORDER BY id;



f. (\*\* - задача опциональная) Напишите SQL запрос, который подсчитывает коэффициент Жаккара (близости множеств) между двумя таблицами A (arena) и B (game) по атрибуту id на основании формулы

WITH intersect\_table AS (  
 SELECT *count*(*\**) AS intersect\_count FROM (  
 SELECT id FROM arena  
 INTERSECT  
 SELECT id FROM game  
 ) intersect\_rows  
), union\_table AS (  
 SELECT *count*(*\**) AS union\_count FROM (  
 SELECT id FROM arena  
 UNION  
 SELECT id FROM game  
 ) union\_rows  
)  
SELECT intersect\_count::float / union\_count AS val  
FROM intersect\_table, union\_table;

