ФГБОУ ВО «Московский технологический университет»

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-1 «Защита информации»

**Дисциплина:** «Безопасность систем баз данных»

**Отчет по лабораторной работе № 6**

**Тема:** «Создание SQL-запросов»

**Вариант задания № 14**

**Выполнил:**

Студент группы БББО-05-20

Кутьин Захар Сергеевич

**Проверил:**

Войтенков Д.В.

**Учебная цель занятия:** Углубить теоретические знания и выработать практические умения в области настройки безопасности компонента Database Engine с помощью среды управления Management Studio и инструкций языка Transact-SQL.

**Место проведения занятия:** компьютерная аудитория.

**Учебно-материальное обеспечение:**

1. Методическая разработка.
2. Компьютерный класс с ПЭВМ.
3. SQL Server 2022 Developer или Enterprise.

**Задание № 1**

Создать и заполнить базу данных РПЛ.

**Ход выполнения задания**

Процесс создания базы данных РПЛ с использованием инструкции языка Transact-SQL показан на рисунке 1.

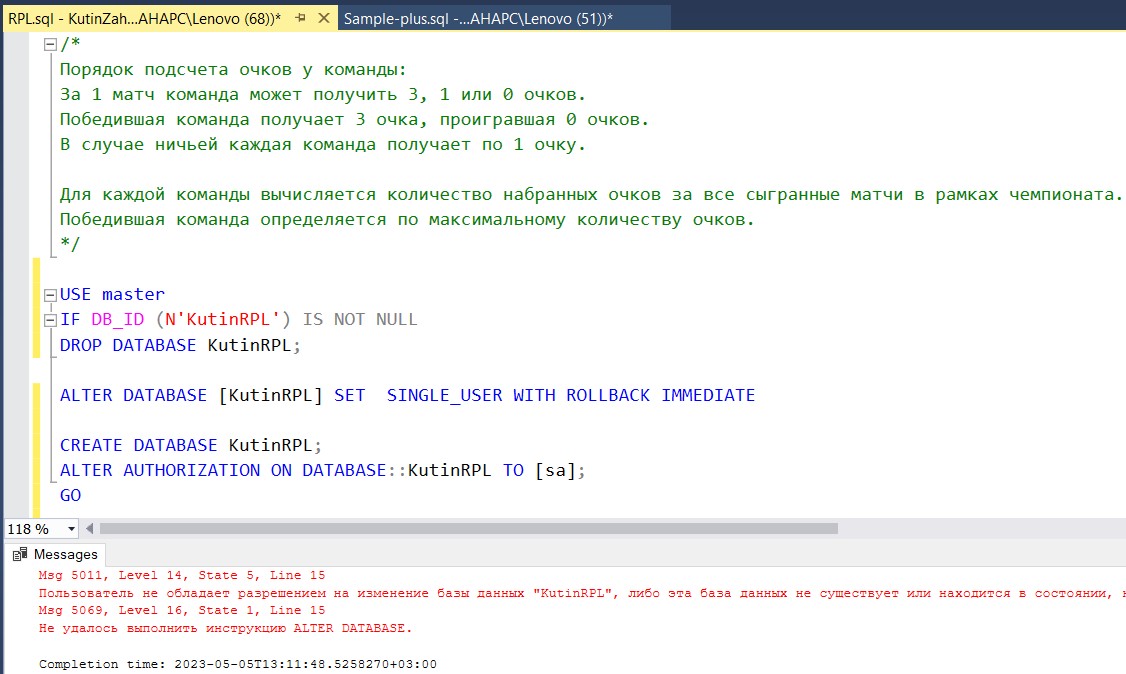


Рисунок 1 - Создание базы данных РПЛ.

Процесс заполнения базы данных РПЛ с использованием инструкции языка Transact-SQL показан на рисунках 2-6.

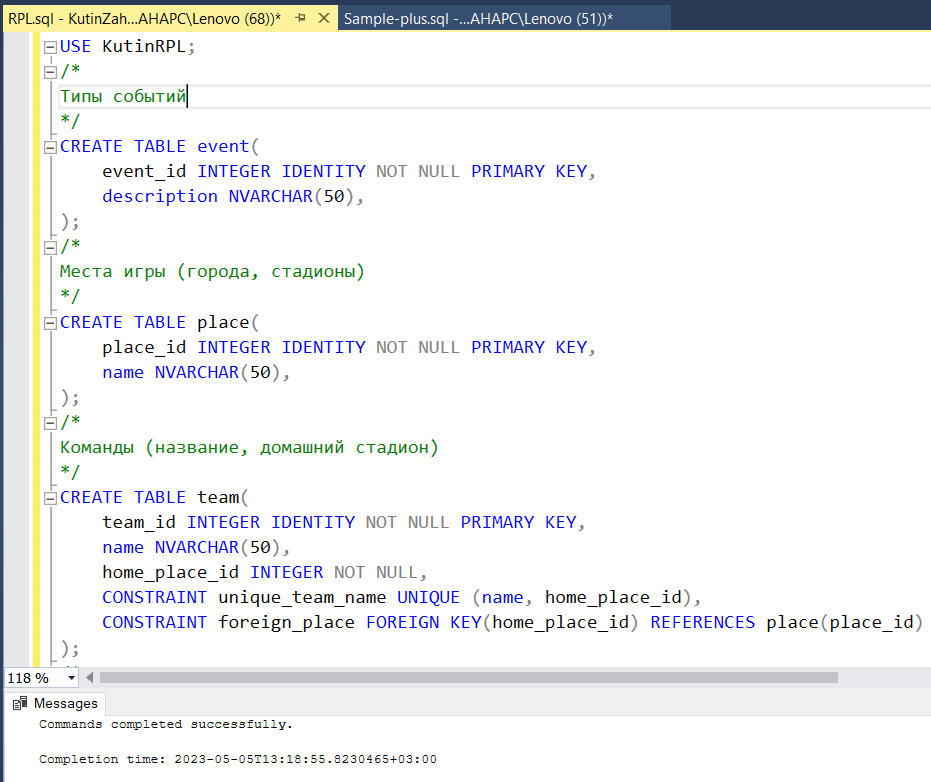


Рисунок 2 - Создание таблиц в базе данных РПЛ (часть 1).

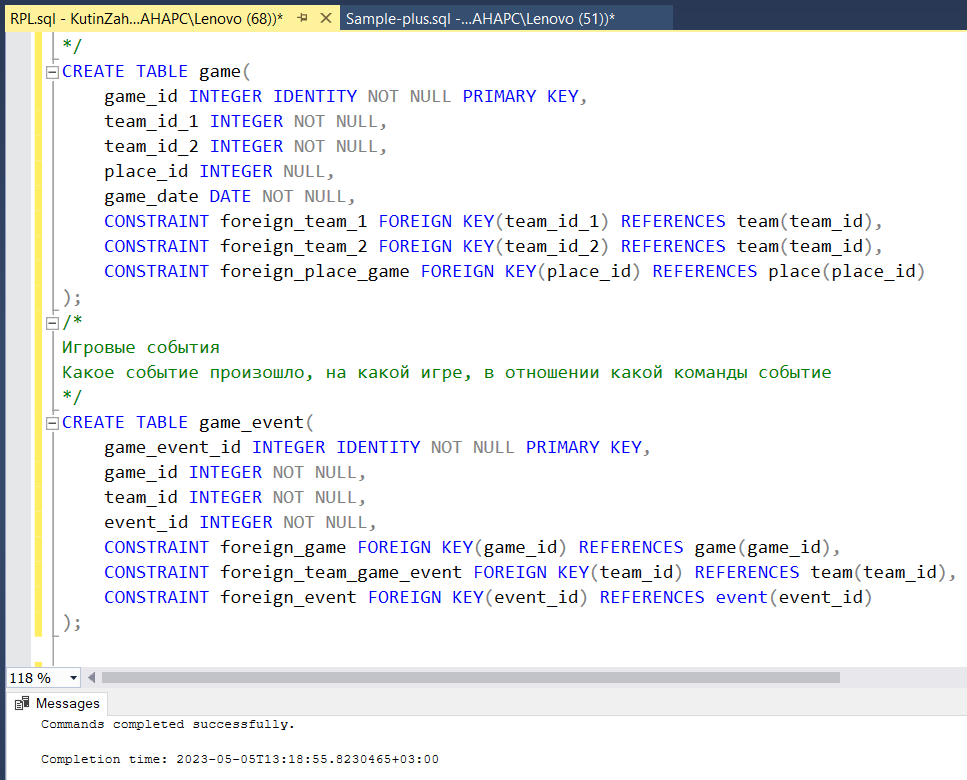


Рисунок 3 - Создание таблиц в базе данных РПЛ (часть 2).

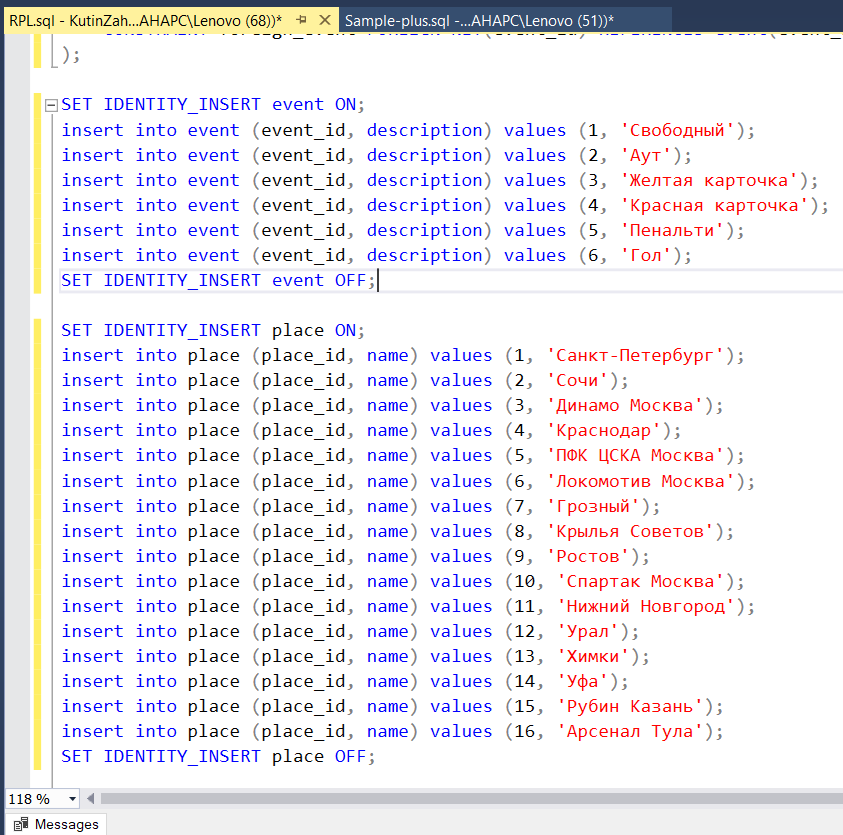


Рисунок 4 - Заполнение таблиц event и place.



Рисунок 5 - Заполнение таблиц team и game.

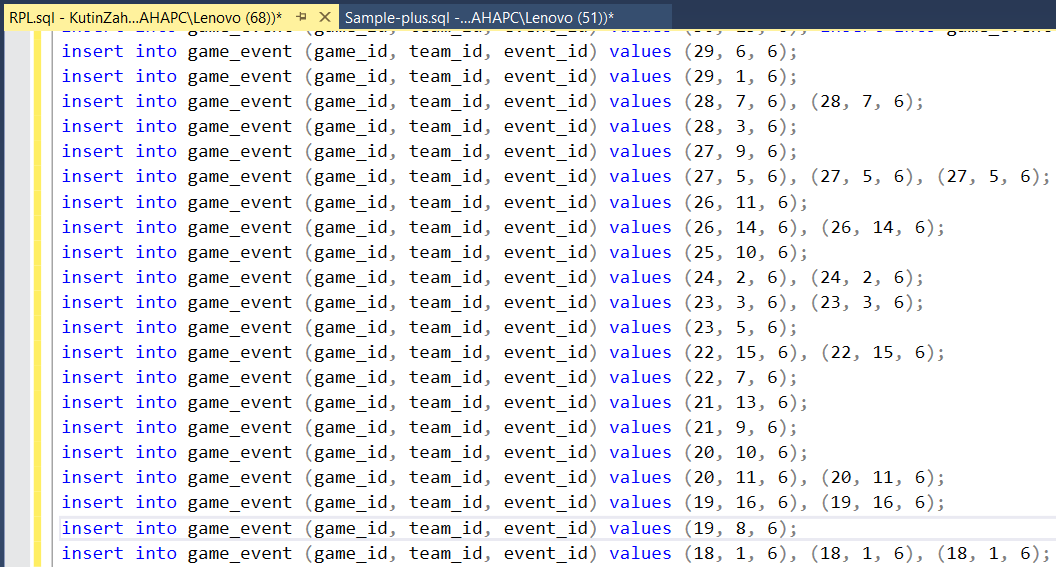


Рисунок 6 - Заполнение таблицы game\_event.

**Задание № 2**

Создать запрос, при выполнении которого будет определен победитель

**Ход выполнения задания**

Процесс создания запроса с использованием инструкции языка Transact-SQL показан на рисунке 7.

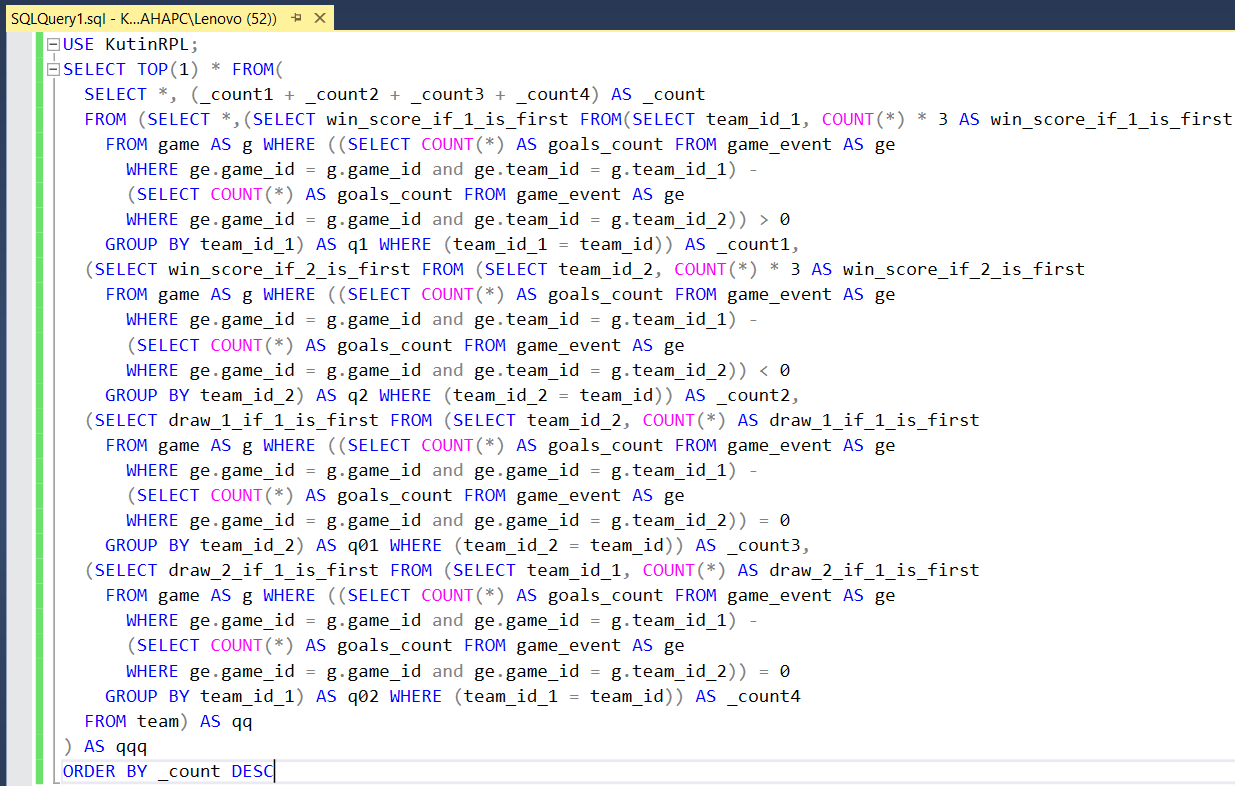


Рисунок 7 – Создание запроса для выявления победителя.

Результатом выполнения запроса будет верхняя строчка отсортированной по возрастанию (ORDER BY \_count DESC) турнирной таблицы.

Первый подзапрос подсчитывает количество победных очков при победе команды, если она принимает «дома»: подсчитывается разница голов, если разница > 0, то команде с team\_id\_1 начисляются 3 очка (COUNT(\*) \* 3).

Второй подзапрос подсчитывает количество победных очков при победе команды, если она играет «в гостях»: подсчитывается разница голов, если разница < 0, то команде с team\_id\_2 начисляются 3 очка (COUNT(\*) \* 3).

Третий подзапрос подсчитывает количество победных очков при ничье команды, если она принимает «дома»: подсчитывается разница голов, если разница = 0, то команде с team\_id\_1 начисляются 1 очко (COUNT(\*)) .

Четвертый подзапрос подсчитывает количество победных очков при ничье команды, если она играет «в гостях»: подсчитывается разница голов, если разница = 0, то команде с team\_id\_2 начисляются 1 очко (COUNT(\*)) .

При выполнении всех подзапросов получается «турнирная» таблица, в которой столбцы \_count1 и \_count2 значат количество очков за победы команды, когда она играла «дома» и «в гостях» соответственно, а \_count3 и \_count4 значат количество ничей команды, когда она играла «дома» и «в гостях» соответственно. Столбец \_count содержит сумму \_count1, \_count2, \_count3 и \_count4. После этого табличка сортируется по убыванию значений столбца \_count. Таким образом первой строкой в таблице окажется команда набравшая наибольшее количество очков.

Результат выполнения запроса показан на рисунке 8.

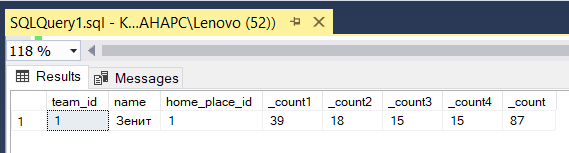


Рисунок 8 - Результат выполнения запроса.

Листинг запроса:

USE KutinRPL;

SELECT TOP(1) \* FROM(

SELECT \*, (\_count1 + \_count2 + \_count3 + \_count4) AS \_count

FROM (SELECT \*,(SELECT win\_score\_if\_1\_is\_first FROM(SELECT team\_id\_1, COUNT(\*) \* 3 AS win\_score\_if\_1\_is\_first

FROM game AS g WHERE ((SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.team\_id = g.team\_id\_1) -

(SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.team\_id = g.team\_id\_2)) > 0

GROUP BY team\_id\_1) AS q1 WHERE (team\_id\_1 = team\_id)) AS \_count1,

(SELECT win\_score\_if\_2\_is\_first FROM (SELECT team\_id\_2, COUNT(\*) \* 3 AS win\_score\_if\_2\_is\_first

FROM game AS g WHERE ((SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.team\_id = g.team\_id\_1) -

(SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.team\_id = g.team\_id\_2)) < 0

GROUP BY team\_id\_2) AS q2 WHERE (team\_id\_2 = team\_id)) AS \_count2,

(SELECT draw\_1\_if\_1\_is\_first FROM (SELECT team\_id\_2, COUNT(\*) AS draw\_1\_if\_1\_is\_first

FROM game AS g WHERE ((SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.game\_id = g.team\_id\_1) -

(SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.game\_id = g.team\_id\_2)) = 0

GROUP BY team\_id\_2) AS q01 WHERE (team\_id\_2 = team\_id)) AS \_count3,

(SELECT draw\_2\_if\_1\_is\_first FROM (SELECT team\_id\_1, COUNT(\*) AS draw\_2\_if\_1\_is\_first

FROM game AS g WHERE ((SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.game\_id = g.team\_id\_1) -

(SELECT COUNT(\*) AS goals\_count FROM game\_event AS ge

WHERE ge.game\_id = g.game\_id and ge.game\_id = g.team\_id\_2)) = 0

GROUP BY team\_id\_1) AS q02 WHERE (team\_id\_1 = team\_id)) AS \_count4

FROM team) AS qq

) AS qqq

ORDER BY \_count DESC

**Выводы по проделанной работе**

В результате выполнения лабораторной работы были решены следующие задачи:

- углублены теоретические знания в области настройки безопасности компонента Database Engine;

- освоены приемы настройки безопасности компонента Database Engine с помощью среды управления Management Studio;

- выработаны практические умения настройки безопасности компонента Database Engine с помощью инструкций языка Transact-SQL.