|  |  |  |
| --- | --- | --- |
| Лабораторная работа №6  Индексы | Ф.И.О. | Курляк Д.В. |
| Группа | ПрИн - 367 |
| Преподаватель | Соколов А. А. |
| Дата сдачи | 08.12.2020 |

**Индексы:**

1. SELECT

demo\_cup.cities.Name, demo\_cup.stadions.Name

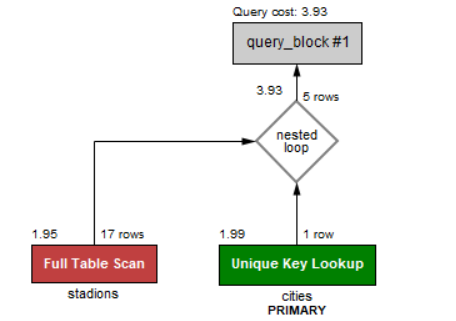
FROM

demo\_cup.cities

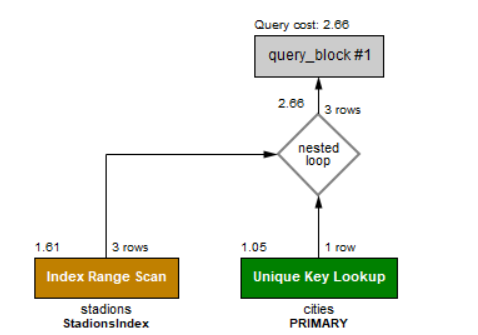
JOIN

demo\_cup.stadions on demo\_cup.stadions.CityId = demo\_cup.cities.id

where stadions.Capacity < 45000;



Create index stadionCapacityIndex on stadions (Capacity);



1. SELECT

FIO, DateTimeFrom, DataTimeTo

FROM

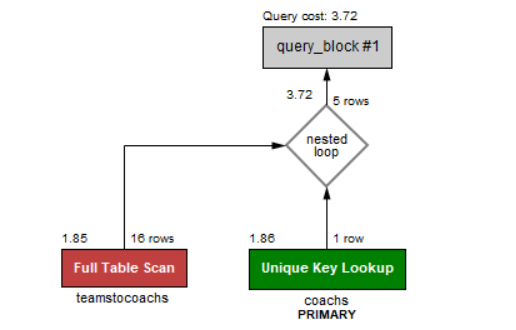
coachs

JOIN

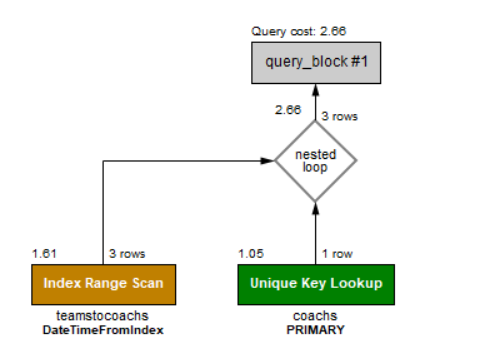
teamstocoachs ON teamstocoachs.CoachId = coachs.id

WHERE

teamstocoachs.DateTimeFrom > '2018-01-01';



Create index DateTimeFromIndex on teamstocoachs (DateTimeFrom);





SELECT

COUNT(\*) AS 'Количество стадионов Москвы', teams.Name

FROM

stadions

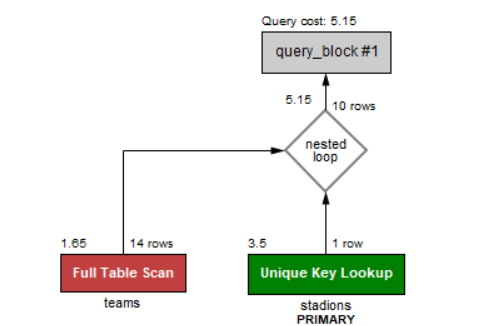
JOIN

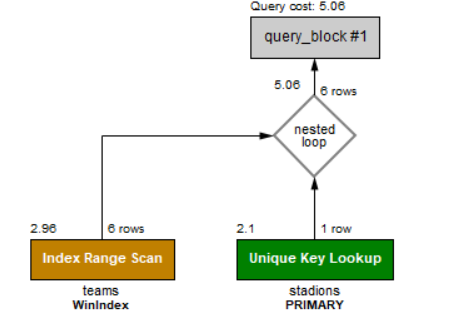
teams

on teams.StadionId = stadions.id

WHERE

Win between 2 and 5;



Create index WinIndex on teams (Win); 

1. SELECT

FIO, MAX(YellowCard) as "Максимальное количество количество карточек"

FROM

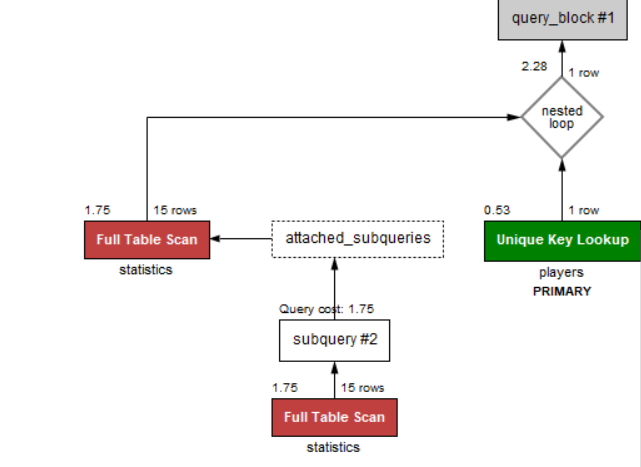
players

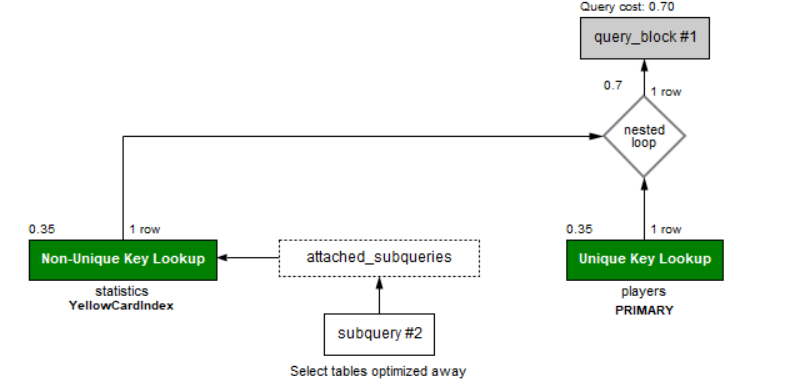
JOIN

statistics ON statistics.PlayerId = players.id

where YellowCard = (select MAX(YellowCard) from statistics)

Query cost: 2.7



Create index YellowCardIndex on statistics (YellowCard); 

1. SELECT

\*

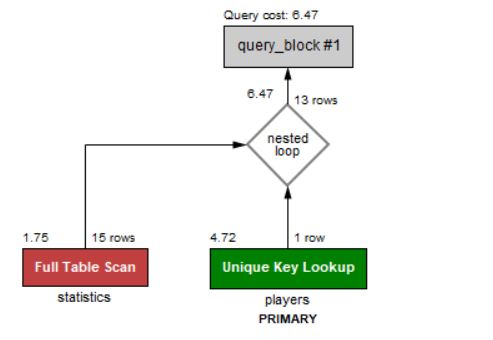
FROM

statistics

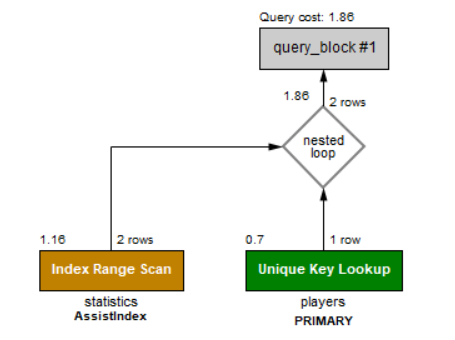
RIGHT JOIN

players ON statistics.PlayerId = players.id

where assist > 3;



Create index AssistIndex on statistics (assist);



**Процедуры**

1. **Показать игрока с максимальным количество карточек**

DELIMITER $$

CREATE PROCEDURE ShowPlayerWithMaxCountOfCards()

BEGIN

SELECT

FIO, MAX(YellowCard + RedCard) as "Количество карточек", YellowCard, RedCard

FROM

players

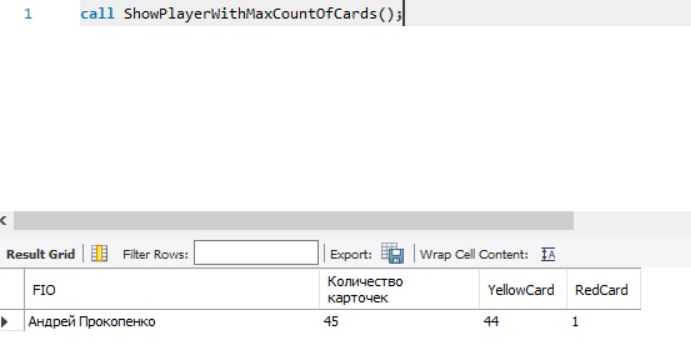
JOIN

statistics ON statistics.PlayerId = players.id

where YellowCard + RedCard = (select MAX(YellowCard + RedCard) from statistics);

END$$

DELIMITER ;



1. **Показать результат встречи с данным id**

DELIMITER $$

CREATE PROCEDURE ShowResultOfTheMatch(game\_id int)

BEGIN

SELECT

stadions.Name AS 'Стадион',

referees.Name AS 'Судья',

TourNumberId,

team1.name,

team2.name,

DateOfMatch,

statistics.goals,

teamstoplayers.TeamId,

players.FIO

FROM

gamesbetweentwoteams

JOIN

statistics

JOIN

teamstoplayers on statistics.PlayerId = teamstoplayers.PlayerId

JOIN

players on players.id = statistics.PlayerId

JOIN

referees ON referees.id = gamesbetweentwoteams.RefereeId

JOIN

stadions ON stadions.id = gamesbetweentwoteams.StadionId

JOIN

teams as team1 on team1.id = Team1Id

JOIN

teams as team2 on team2.id = Team2Id

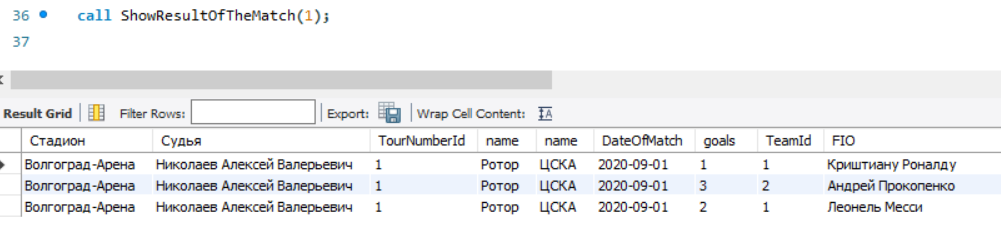
WHERE

statistics.GameId = game\_id

AND gamesbetweentwoteams.id = game\_id;

END$$

DELIMITER ;



1. **Показать состав команды с заданным id**

DELIMITER $$

CREATE PROCEDURE ShowsPlayersTeams(idOfTeam int)

BEGIN

SELECT

demo\_cup.players.FIO, demo\_cup.teamstoplayers.TeamId

FROM

demo\_cup.players

JOIN

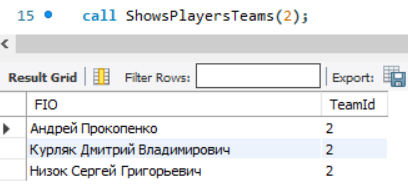
demo\_cup.teamstoplayers on demo\_cup.players.id = demo\_cup.teamstoplayers.PlayerId

WHERE

demo\_cup.teamstoplayers.TeamId = idOfTeam;

END$$

DELIMITER ;



**Функции**

1. DELIMITER $$

CREATE FUNCTION CountOfCard (yellow int, red int)

Returns int DETERMINISTIC

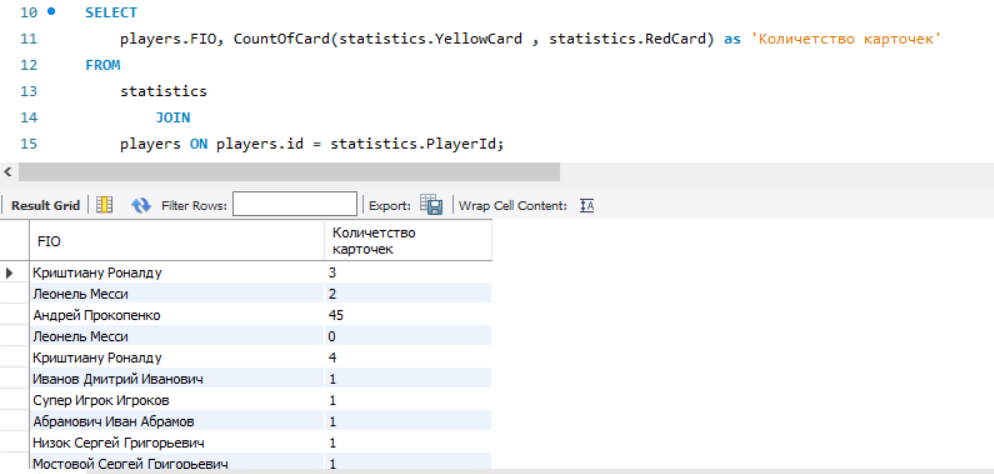
BEGIN

return yellow + red;

END;

$$

DELIMITER ;



1. DELIMITER $$

CREATE FUNCTION PlayersWithTeam (TeamId int)

Returns boolean DETERMINISTIC

BEGIN

declare a boolean;

if TeamId is null then

set a = false;

else

set a = true;

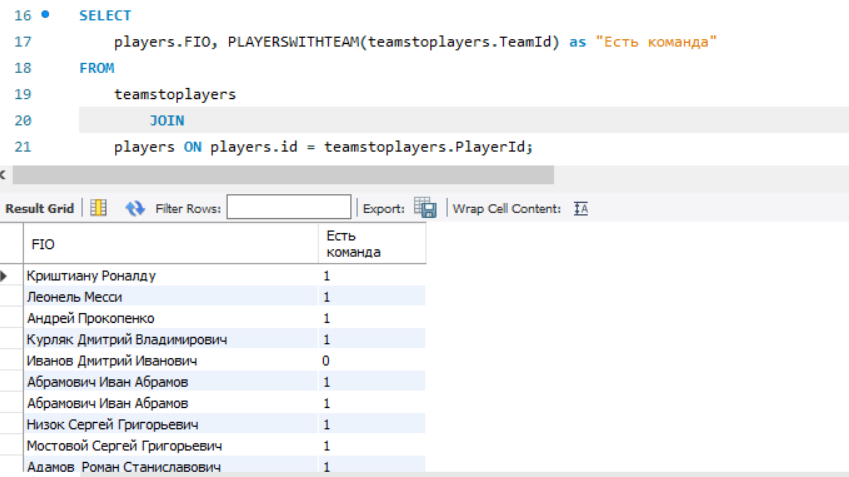
END IF;

return (a);

END;

$$

DELIMITER ;



DELIMITER $$

CREATE FUNCTION TourView (Number int)

RETURNS varchar(45)

DETERMINISTIC

BEGIN

declare a varchar(45);

case

when Number = 1 then set a = "Первый тур 04/09/2020 - 07/09/2020";

when Number = 2 then set a = "Второй тур 11/09/2020 - 14/09/2020";

when Number = 3 then set a = "Третий тур 17/09/2020 - 20/09/2020";

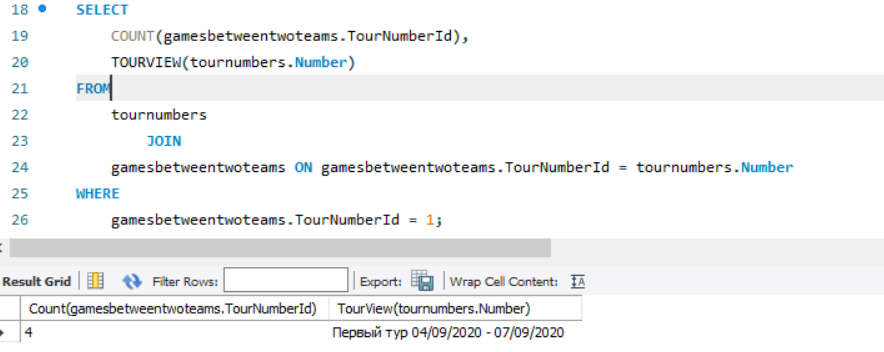
when Number = 4 then set a = "Четвертый тур 24/09/2020 - 27/09/2020";

end case;

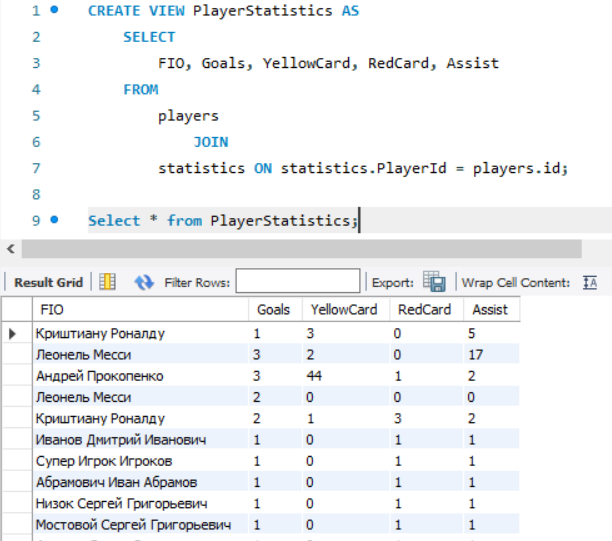
return (a);

END;

$$



**Представления**



2.

