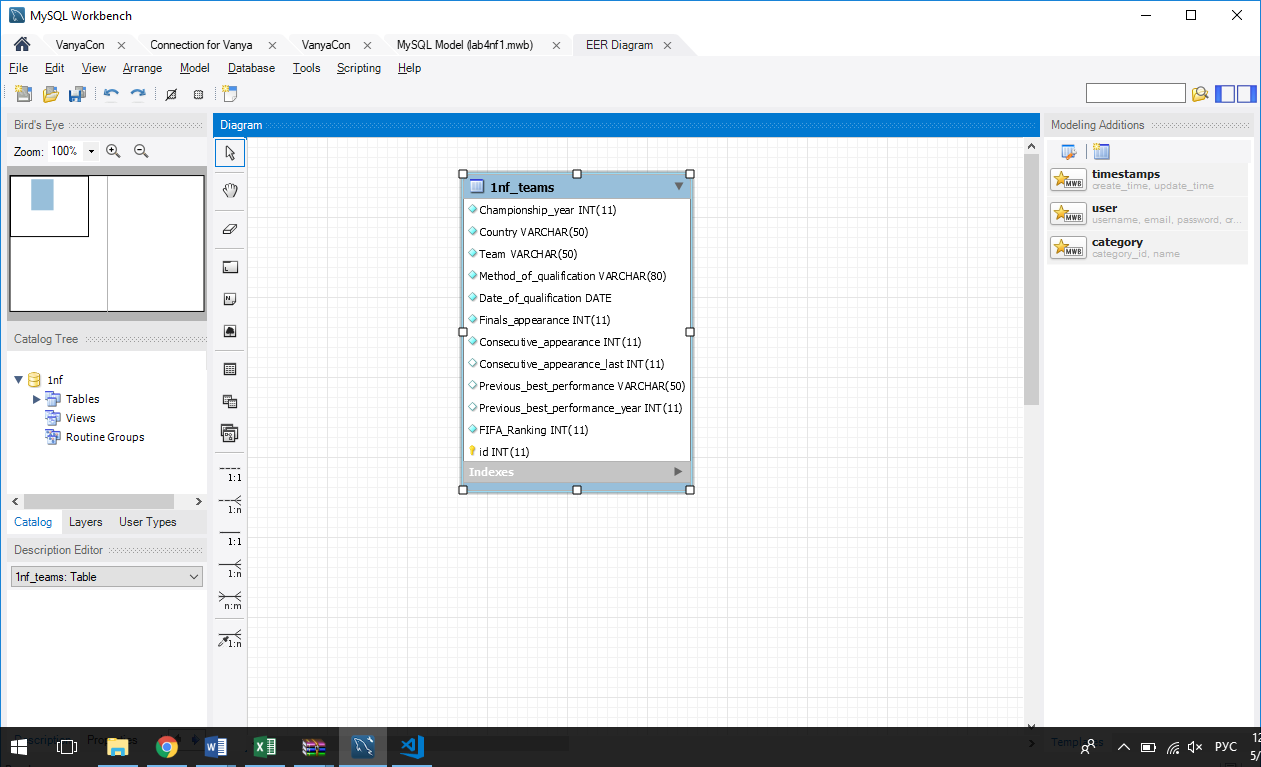
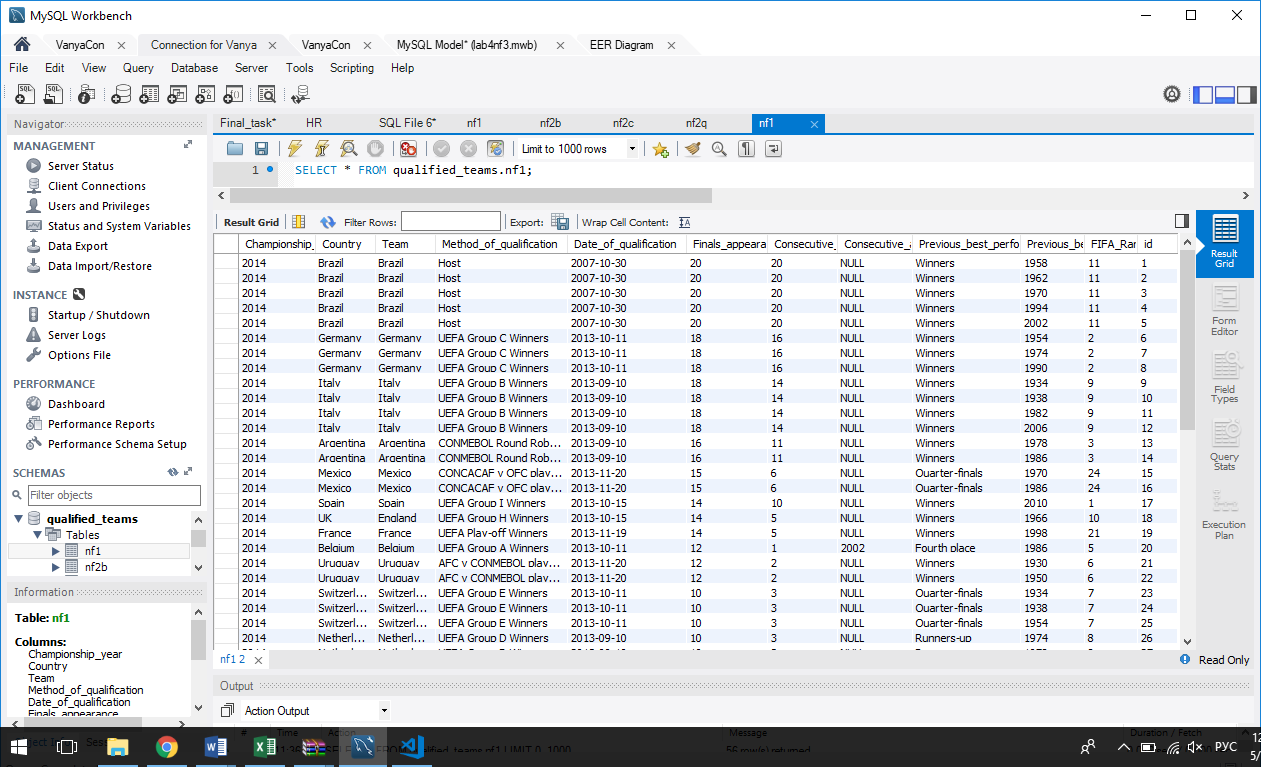
**Лабораторная работа №4.**    
«Сложные SQL запросы»

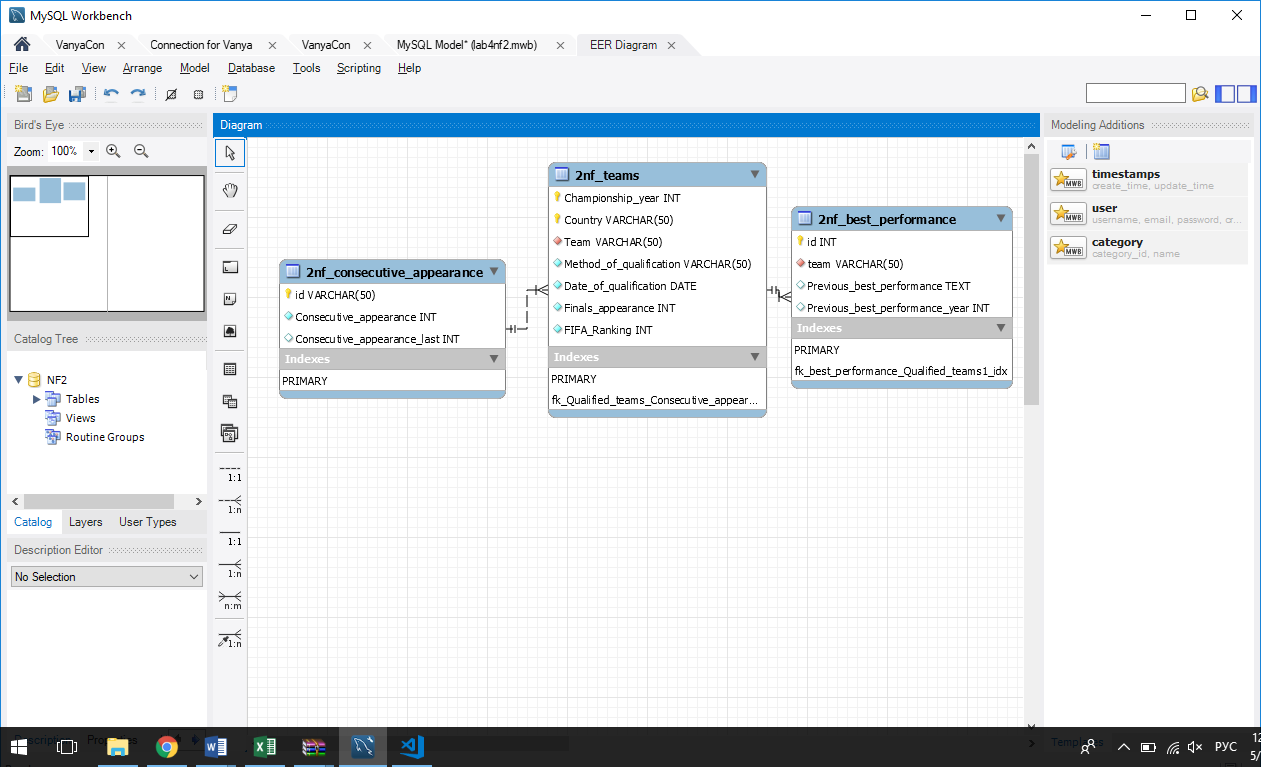
**Задание 1 (NF)**

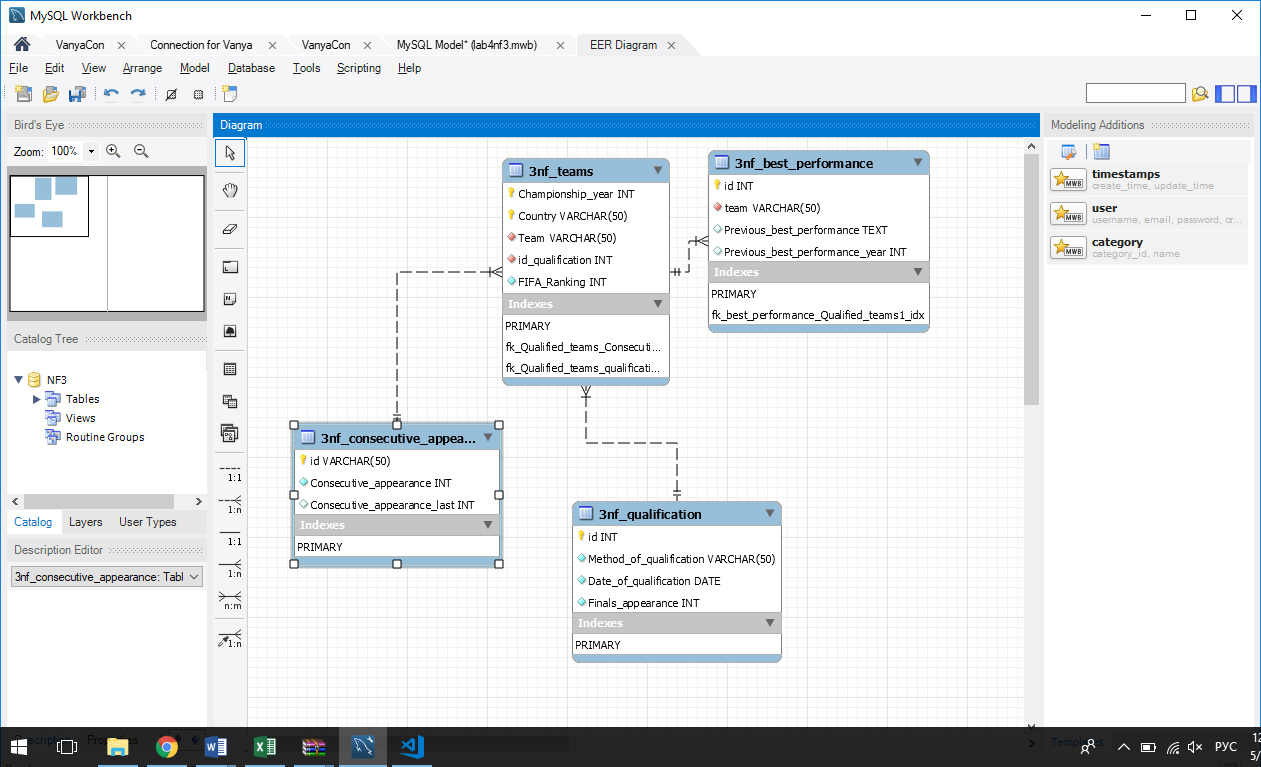
1. Привести ненормализованную таблицу из задания к 1NF и создать таблицу с данными для ее





1. Привести 1NF к 2NF, а 2NF к 3NF





**Задание 2 (views)**

1. На основании 1NF создать представления представляющие собой 2NF

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`2nf\_best\_performance` AS

SELECT

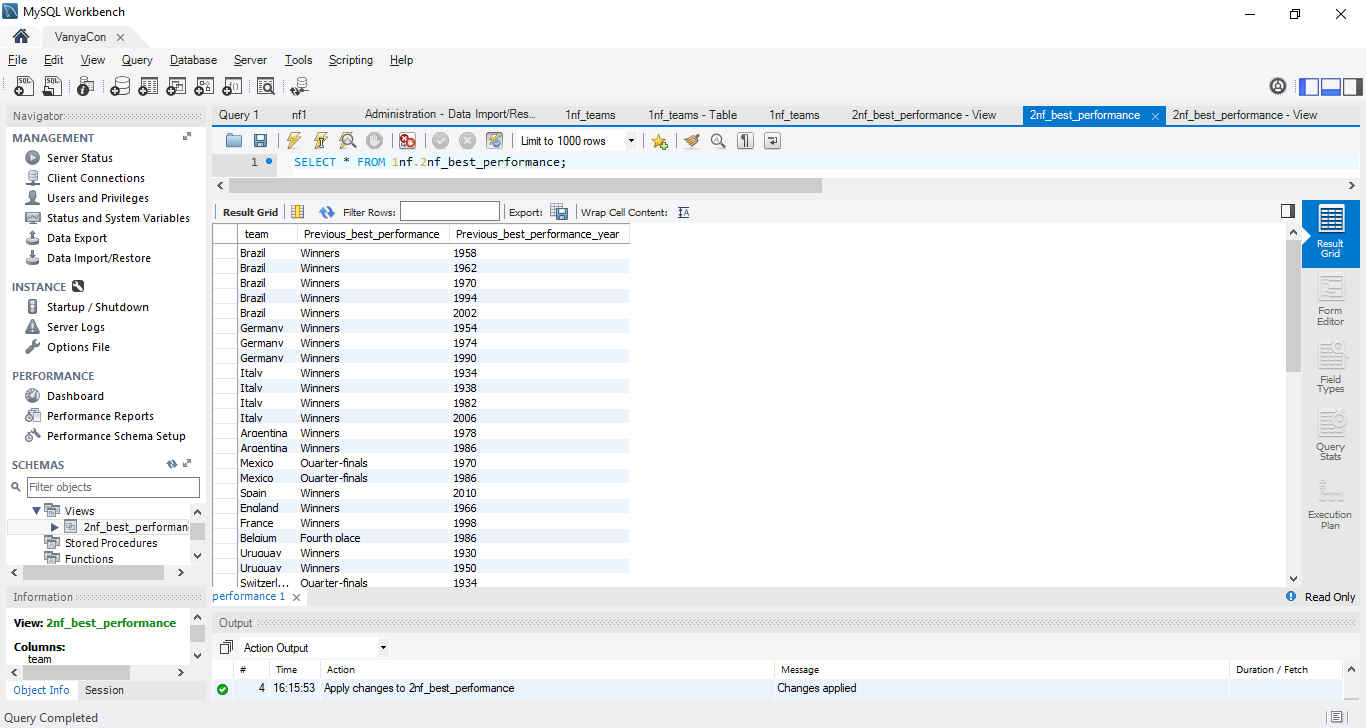
`1nf`.`1nf\_teams`.`Team` AS `team`,

`1nf`.`1nf\_teams`.`Previous\_best\_performance` AS `Previous\_best\_performance`,

`1nf`.`1nf\_teams`.`Previous\_best\_performance\_year` AS `Previous\_best\_performance\_year`

FROM

`1nf`.`1nf\_teams`



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`2nf\_consecutive\_appearance` AS

SELECT DISTINCT

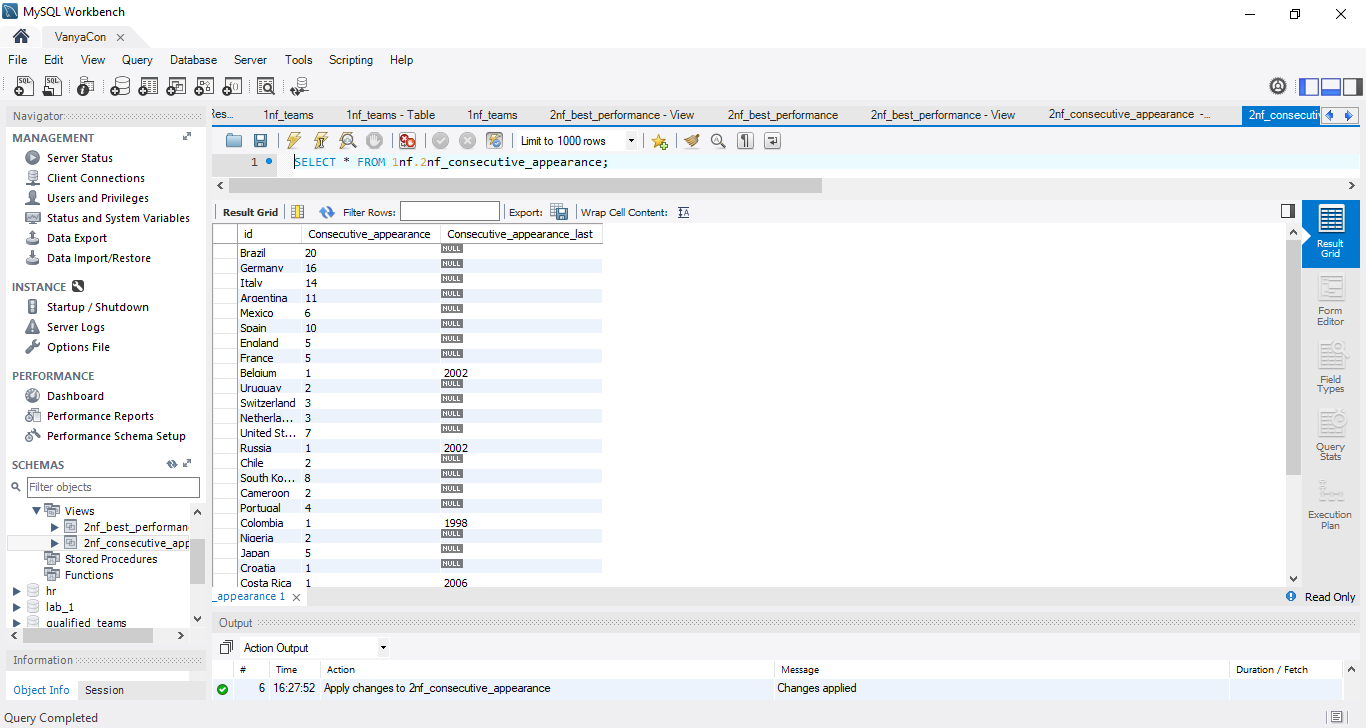
`1nf`.`1nf\_teams`.`Team` AS `id`,

`1nf`.`1nf\_teams`.`Consecutive\_appearance` AS `Consecutive\_appearance`,

`1nf`.`1nf\_teams`.`Consecutive\_appearance\_last` AS `Consecutive\_appearance\_last`

FROM

`1nf`.`1nf\_teams`



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`2nf\_teams` AS

SELECT DISTINCT

`1nf`.`1nf\_teams`.`Championship\_year` AS `Championship\_year`,

`1nf`.`1nf\_teams`.`Country` AS `Country`,

`1nf`.`1nf\_teams`.`Team` AS `Team`,

`1nf`.`1nf\_teams`.`Method\_of\_qualification` AS `Method\_of\_qualification`,

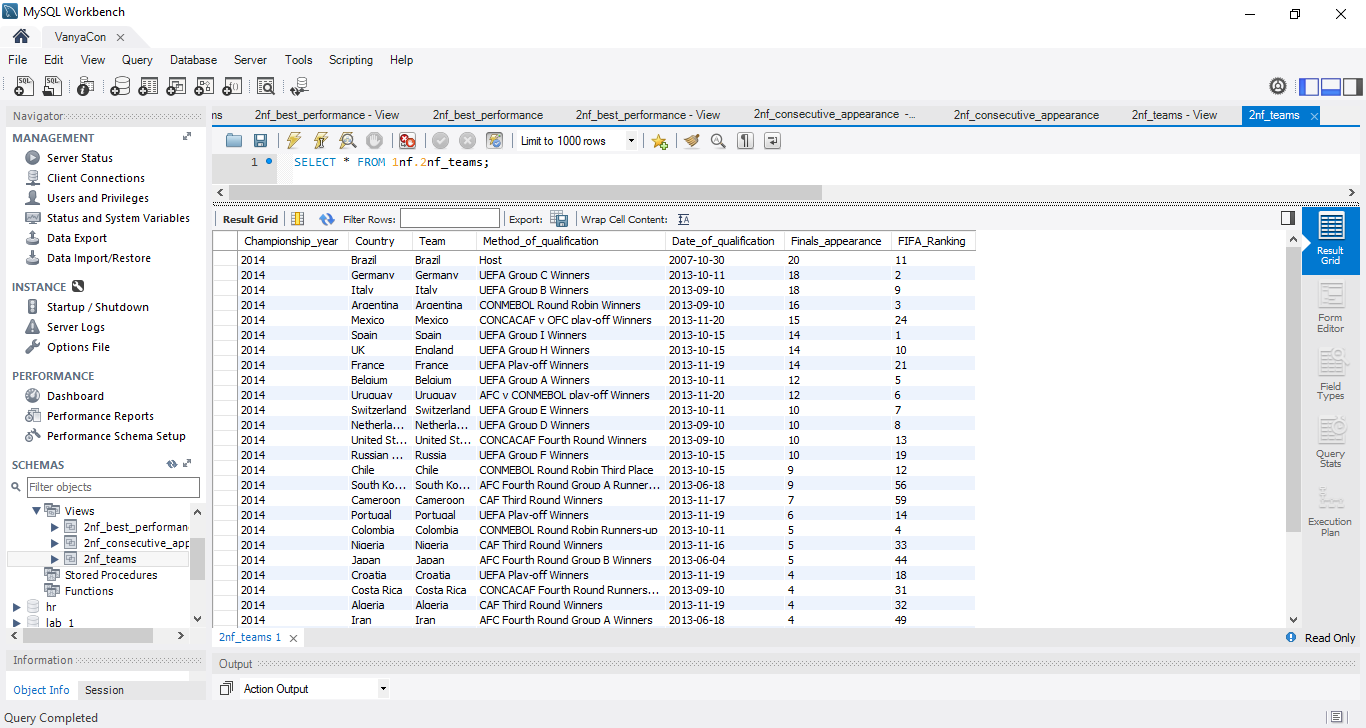
`1nf`.`1nf\_teams`.`Date\_of\_qualification` AS `Date\_of\_qualification`,

`1nf`.`1nf\_teams`.`Finals\_appearance` AS `Finals\_appearance`,

`1nf`.`1nf\_teams`.`FIFA\_Ranking` AS `FIFA\_Ranking`

FROM

`1nf`.`1nf\_teams`



1. На основании представлений 2NF создать представления представляющие собой 3NF

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`3nf\_best\_performance` AS

SELECT

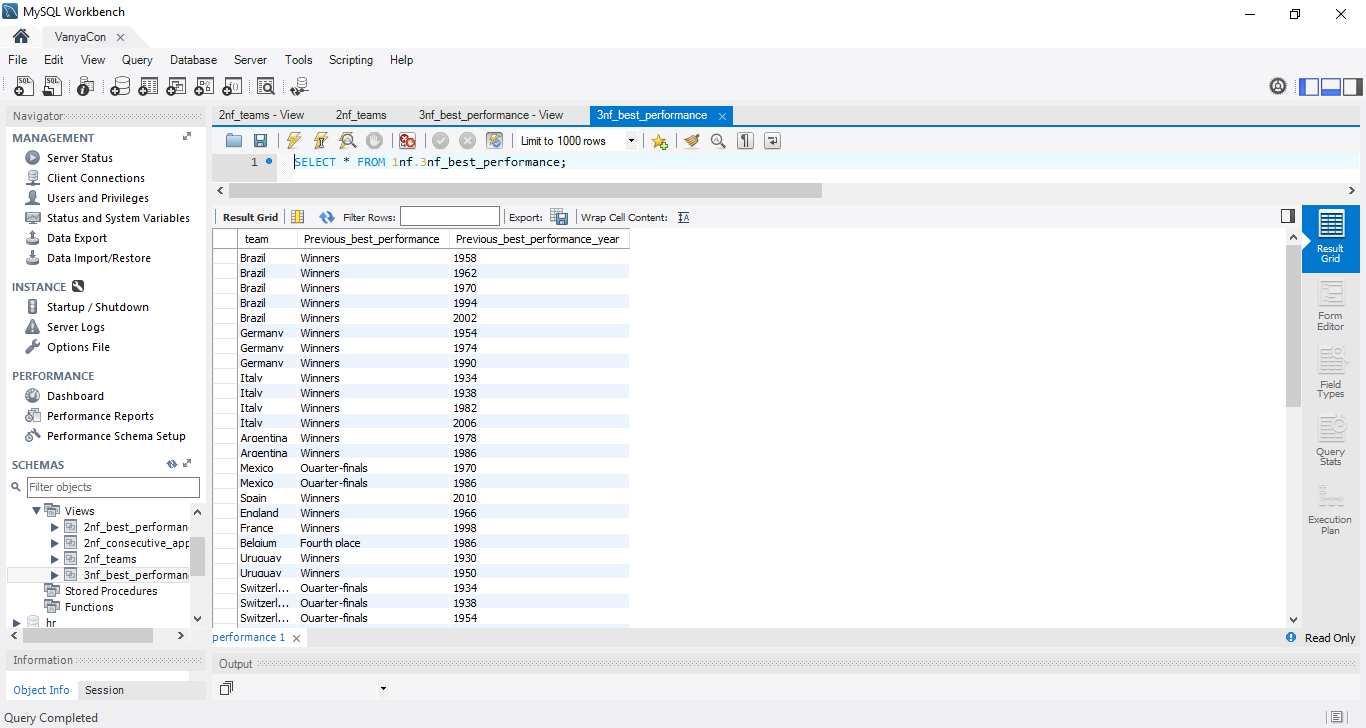
`2nf\_best\_performance`.`team` AS `team`,

`2nf\_best\_performance`.`Previous\_best\_performance` AS `Previous\_best\_performance`,

`2nf\_best\_performance`.`Previous\_best\_performance\_year` AS `Previous\_best\_performance\_year`

FROM

`1nf`.`2nf\_best\_performance`



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`3nf\_consecutive\_appearance` AS

SELECT

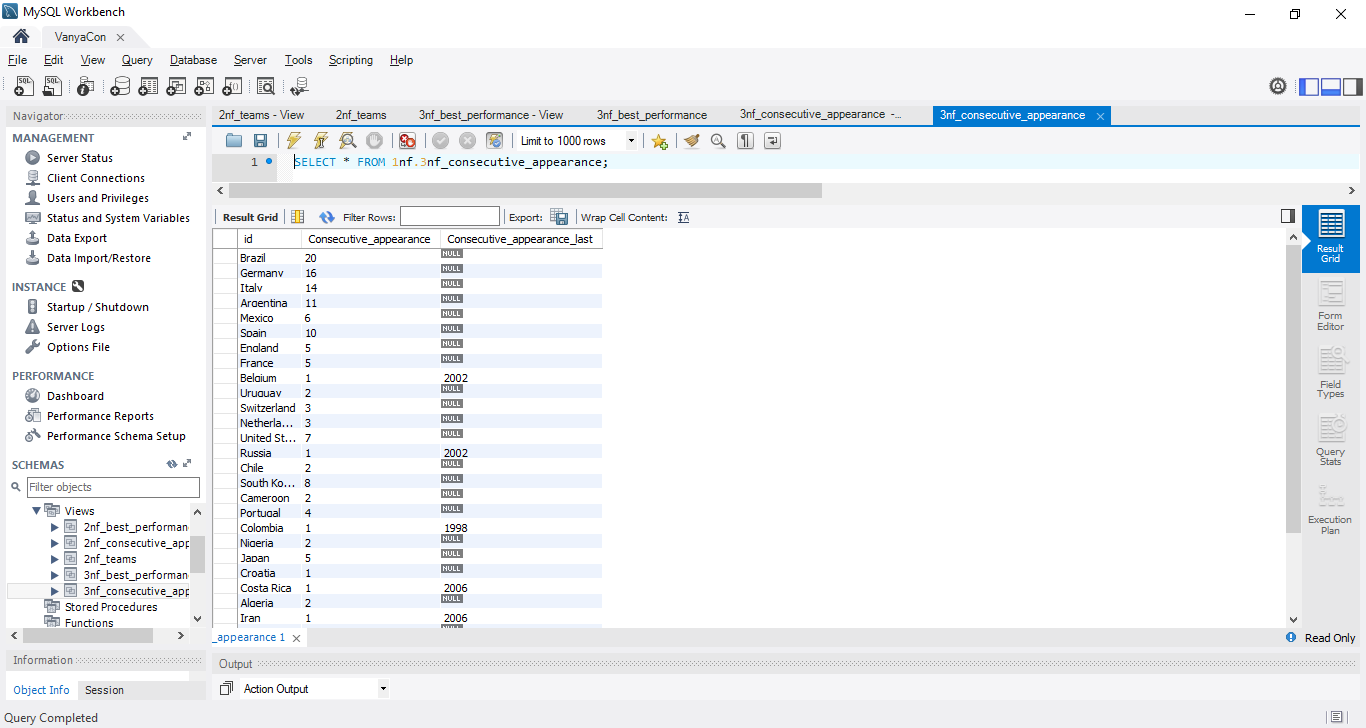
`2nf\_consecutive\_appearance`.`id` AS `id`,

`2nf\_consecutive\_appearance`.`Consecutive\_appearance` AS `Consecutive\_appearance`,

`2nf\_consecutive\_appearance`.`Consecutive\_appearance\_last` AS `Consecutive\_appearance\_last`

FROM

`1nf`.`2nf\_consecutive\_appearance`



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`3nf\_qualification` AS

SELECT

`2nf\_teams`.`Team` AS `Team`,

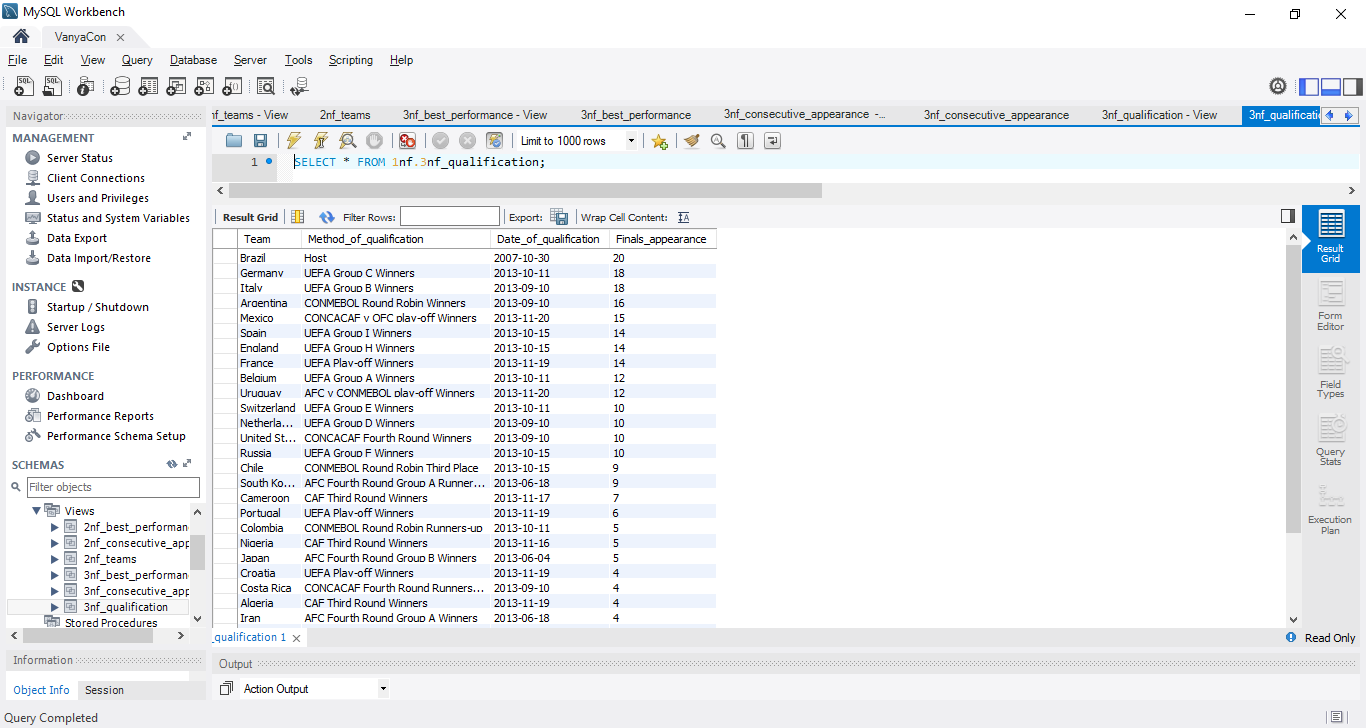
`2nf\_teams`.`Method\_of\_qualification` AS `Method\_of\_qualification`,

`2nf\_teams`.`Date\_of\_qualification` AS `Date\_of\_qualification`,

`2nf\_teams`.`Finals\_appearance` AS `Finals\_appearance`

FROM

`1nf`.`2nf\_teams`



CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf`.`3nf\_teams` AS

SELECT

`2nf\_teams`.`Championship\_year` AS `Championship\_year`,

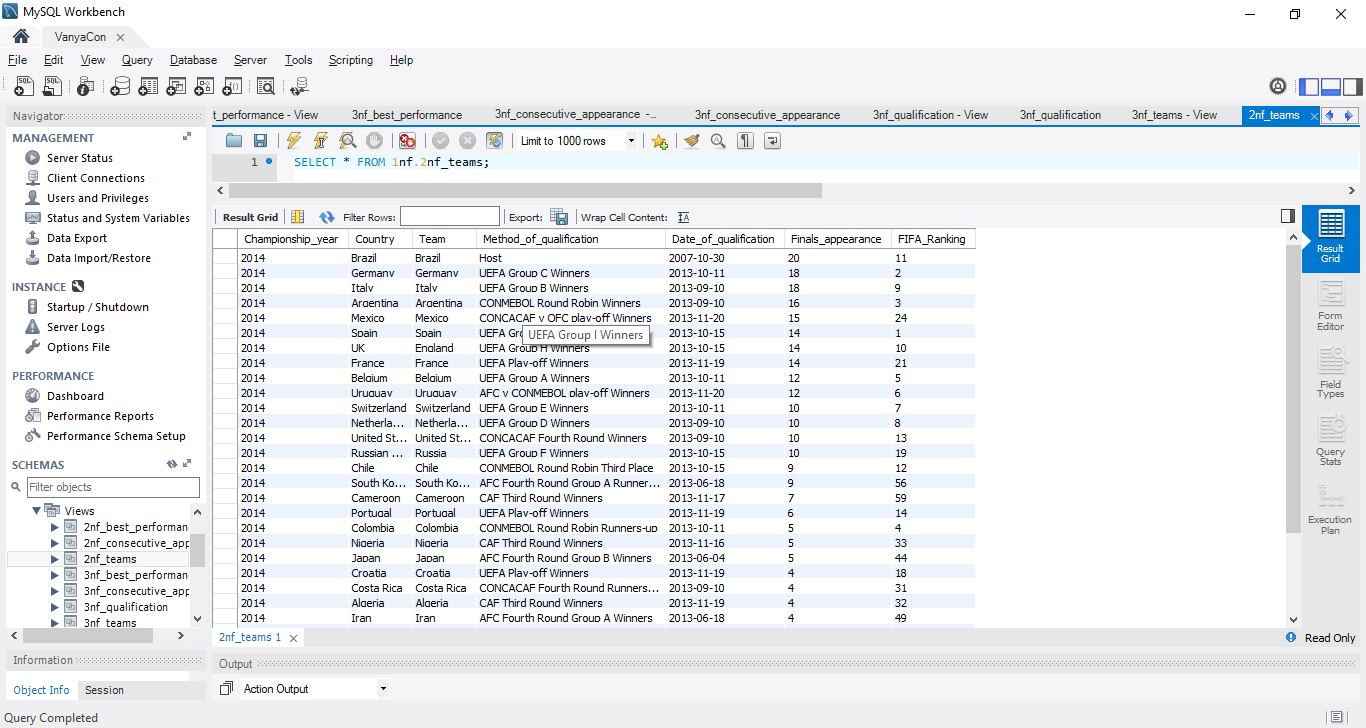
`2nf\_teams`.`Country` AS `Country`,

`2nf\_teams`.`Team` AS `Team`,

`2nf\_teams`.`FIFA\_Ranking` AS `FIFA\_Ranking`

FROM

`1nf`.`2nf\_teams`



**Задание 3 (подзапросы, добавление данных)**

1. На основании представлений из 3NF создать таблицы и вставить данные в таблицы для 3NF с необходимыми FK для этой нормальной формы.

CREATE TABLE `3nf\_consecutive\_appearance\_t` (

`id` varchar(50) NOT NULL,

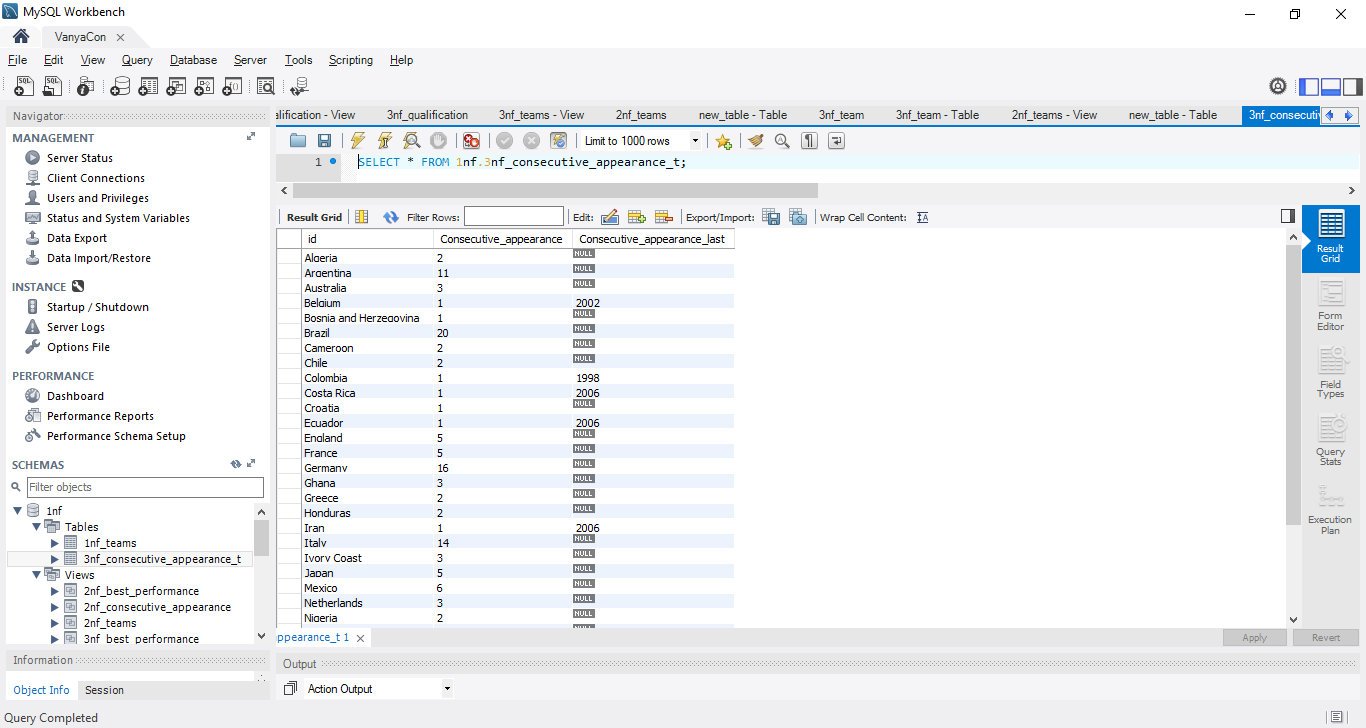
`Consecutive\_appearance` int(11) NOT NULL,

`Consecutive\_appearance\_last` int(11) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

SELECT \* FROM `3nf\_consecutive\_appearance`;



CREATE TABLE `3nf\_qualification\_t` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`Method\_of\_qualification` varchar(50) NOT NULL,

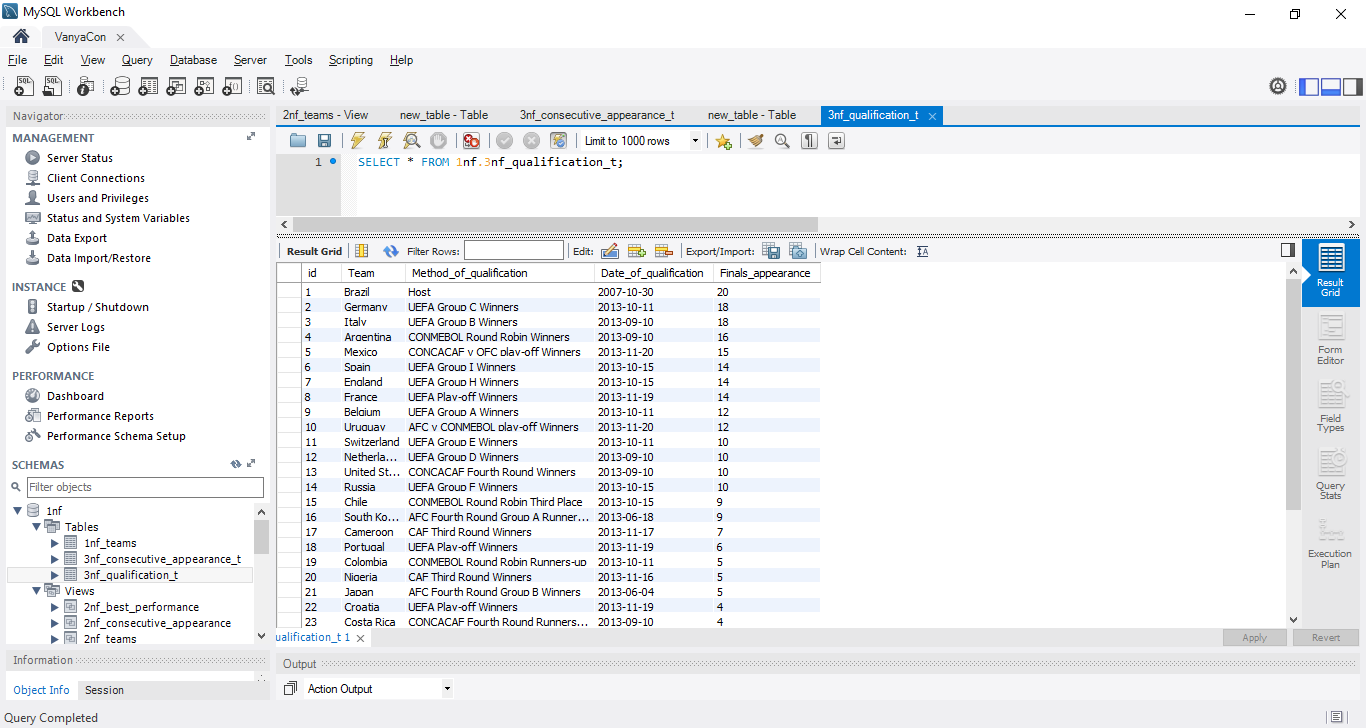
`Date\_of\_qualification` date NOT NULL,

`Finals\_appearance` int(11) NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

SELECT \* FROM `3nf\_qualification`;



CREATE TABLE `3nf\_teams\_t` (

`Championship\_year` int(11) NOT NULL,

`Country` varchar(50) NOT NULL,

`Team` varchar(50) NOT NULL,

`id\_qualification` int(11) NOT NULL AUTO\_INCREMENT,

`FIFA\_Ranking` int(11) NOT NULL,

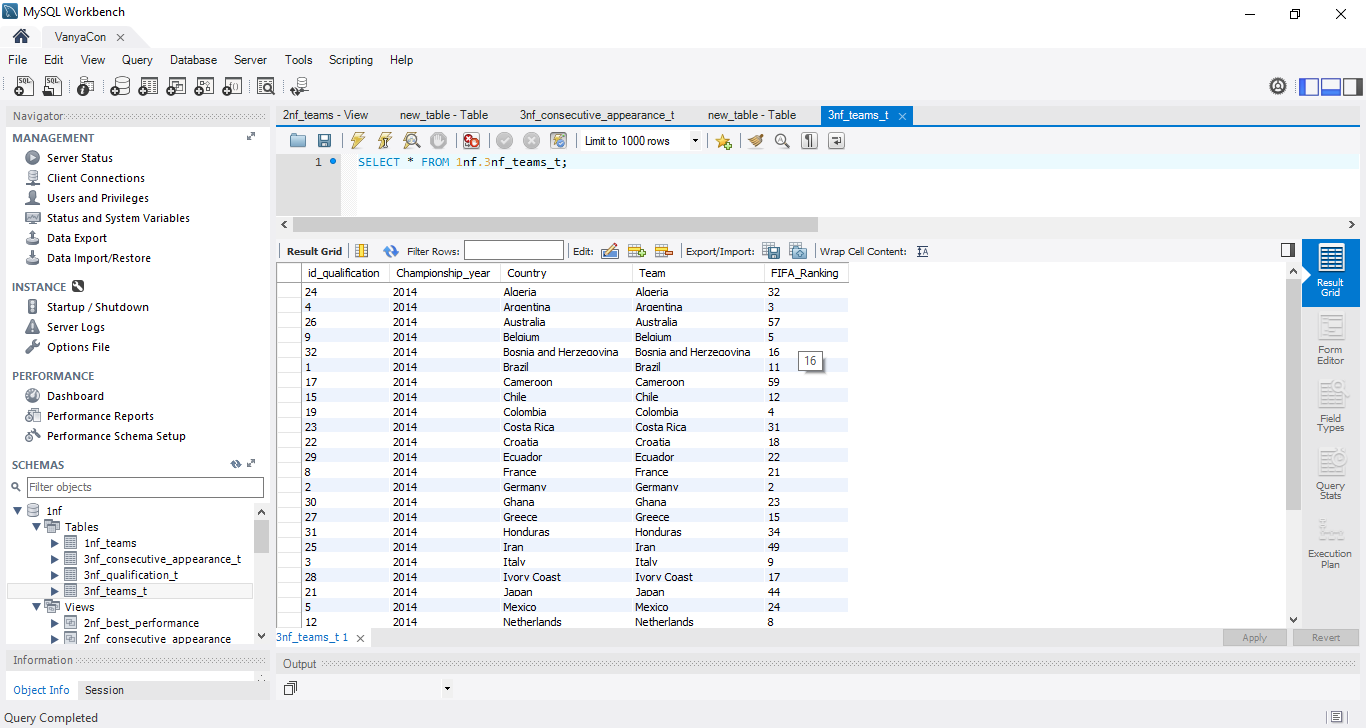
PRIMARY KEY (`Championship\_year`,`Country`),

FOREIGN KEY (`Team`) REFERENCES `3nf\_consecutive\_appearance\_t` (`id`),

FOREIGN KEY (`id\_qualification`) REFERENCES `3nf\_qualification\_t` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

SELECT \* FROM `3nf\_teams`;



CREATE TABLE `3nf\_best\_performance\_t` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`team` varchar(50) NOT NULL,

`Previous\_best\_performance` text,

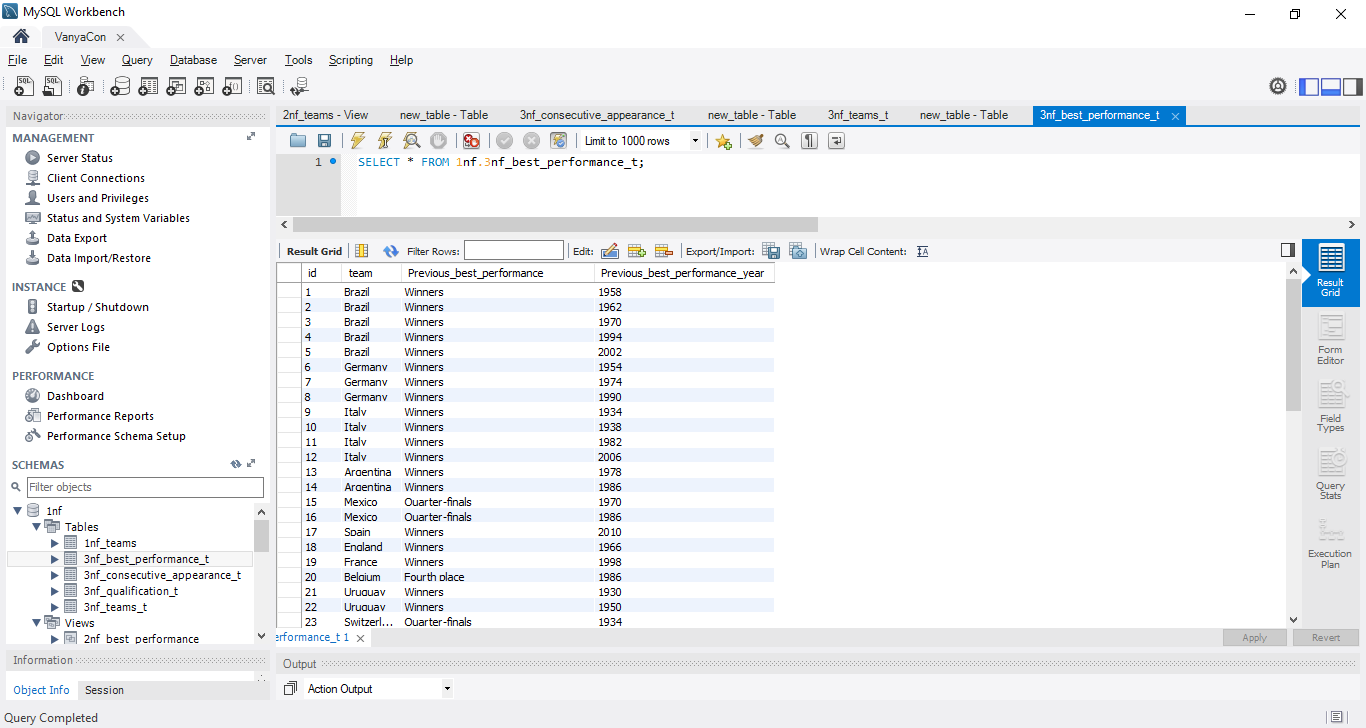
`Previous\_best\_performance\_year` int(11) DEFAULT NULL,

PRIMARY KEY (`id`),

FOREIGN KEY (`team`) REFERENCES `3nf\_teams\_t` (`Team`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8

SELECT \* FROM `3nf\_best\_performance`;



1. Создать представление выводящее базу данныx в 1NF на основании данных из таблиц для 3NF.

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `1nf\_view` AS

SELECT

`3t`.`Championship\_year` AS `Championship\_year`,

`3t`.`Country` AS `Country`,

`3t`.`Team` AS `Team`,

`3q`.`Method\_of\_qualification` AS `Method\_of\_qualification`,

`3q`.`Date\_of\_qualification` AS `Date\_of\_qualification`,

`3q`.`Finals\_appearance` AS `Finals\_appearance`,

`3c`.`Consecutive\_appearance` AS `Consecutive\_appearance`,

`3c`.`Consecutive\_appearance\_last` AS `Consecutive\_appearance\_last`,

`3b`.`Previous\_best\_performance` AS `Previous\_best\_performance`,

`3b`.`Previous\_best\_performance\_year` AS `Previous\_best\_performance\_year`,

`3t`.`FIFA\_Ranking` AS `FIFA\_Ranking`,

`3b`.`id` AS `id`

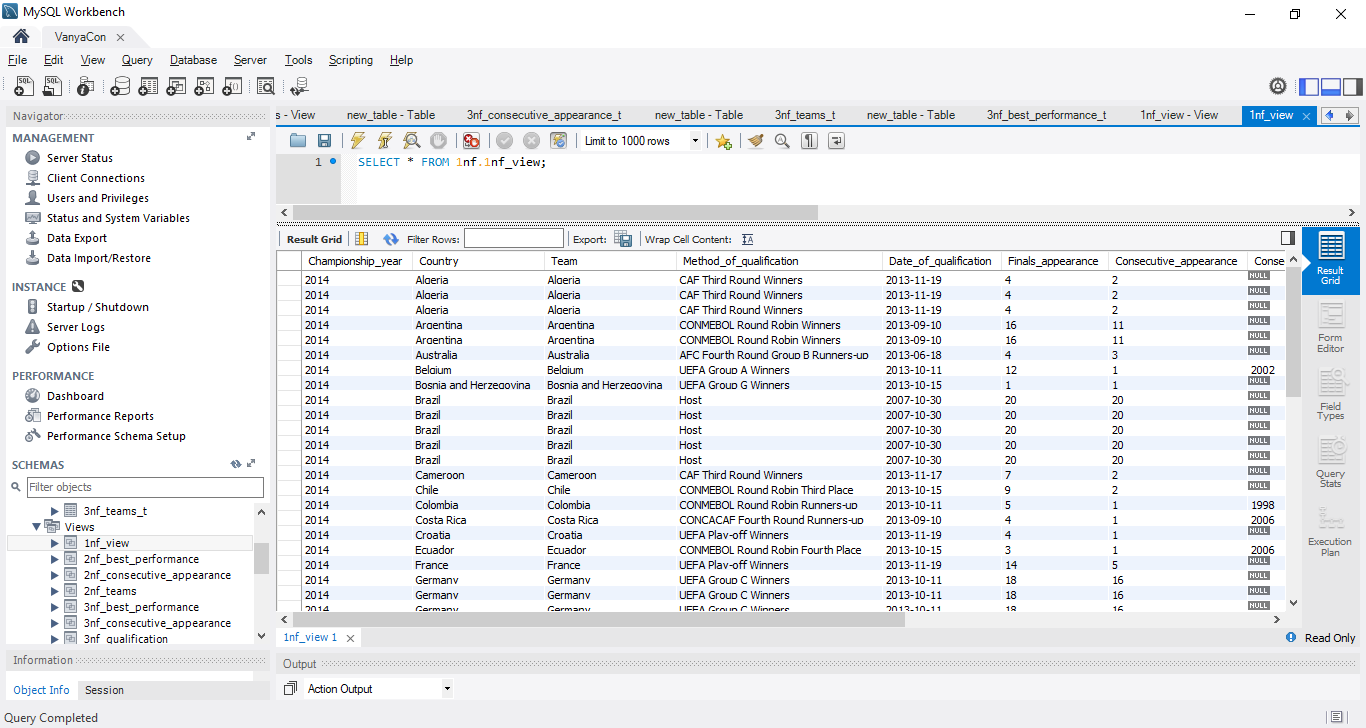
FROM

(((`3nf\_consecutive\_appearance\_t` `3c`

JOIN `3nf\_teams\_t` `3t` ON ((`3c`.`id` = `3t`.`Team`)))

JOIN `3nf\_best\_performance\_t` `3b` ON ((`3t`.`Team` = `3b`.`team`)))

JOIN `3nf\_qualification\_t` `3q` ON ((`3t`.`id\_qualification` = `3q`.`id`)));



1. Написать запрос сравнивающий данные из представления для 1NF (см. Зад 2.1) и данные в таблице для 1NF и разницу вывести на экран.

SELECT

\*

FROM

1nf\_view

LEFT JOIN

1nf\_teams ON 1nf\_view.id = 1nf\_teams.id

WHERE

1nf\_view.id IS NULL OR 1nf\_teams.id IS NULL;

