МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

И ПРОГРАМНОЙ ИНЖИНЕРИИ

№ 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старш. преподаватель |  |  |  | Путилова Н.В. |
| должность, звание |  | подпись, дата |  | фамилия, инициалы |

|  |
| --- |
| ОТЧЕТ К ЛАБОРАТОРНОЙ РАБОТЕ №6 |
| «Хранимые процедуры» |
| по дисциплине: Проектирование баз данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТГР. № | 4936 |  |  |  | Судак.А.А. |
|  |  |  | подпись, дата |  | фамилия, инициалы |

**Санкт-Петербург 2021**

**Задание:**

По аналогии с примерами, приведенными в п. 1 , создать в БД ХП, реализующие:

— вставку с пополнением справочников (вставляется информация о студенте, если указанный номер группы отсутствует в БД, запись добавляется в таблицу с перечнем групп) (получаем ссылку на внешний ключ по значению данного из родительской таблицы);

— удаление с очисткой справочников

– удаление всех зависимых данных(удаляется информация о студенте, если в его группе нет больше студентов, запись удаляется из таблицы с перечнем групп);

— каскадное удаление (перед удалением записи о группе удаляются записи обо всех студентах этой группы);

— вычисление и возврат значения агрегатной функции (т.к. агрегатная функция дает единственный результат);

— формирование статистики во временной таблице

**Вариант 17:**

Школьные экскурсии: тип (развлекательная/образовательная), дисциплины к которым имеет отношение образовательная экскурсия, стоимость с человека, список участников, ответственный за проведение учитель

а. экскурсии на заводы (слово «завод» в любом месте названия)

б. экскурсии, относящиеся к биологии и истории

в. учащиеся, которые не ездили за экскурсии

г. учащиеся, которые не ездили в музей истории религии, но ездил в Кунсткамеру

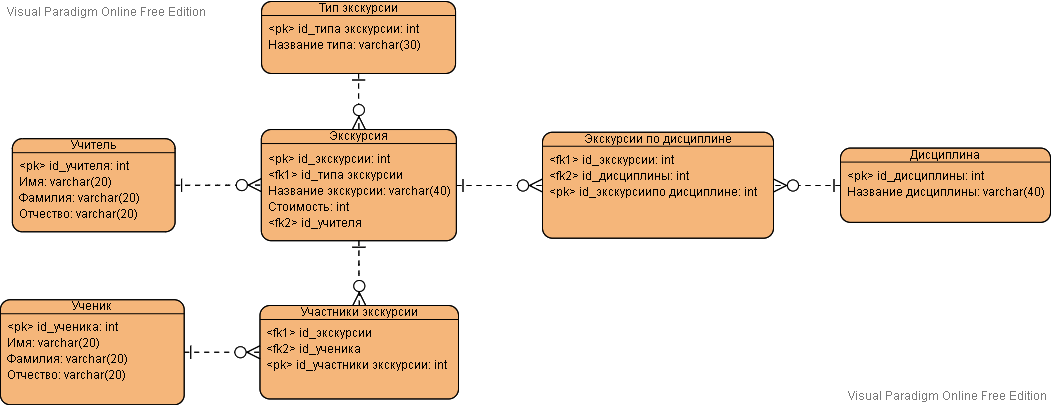
д. учитель, отвечавший за на наибольшее число экскурсий

е. самые дешевые экскурсии

ж. учащийся, который был на всех экскурсиях

**Ход Работы:**

1. Физическая модель базы данных



1. Текст процедур на SQL и наборы данных, возвращаемые процедурами:

— вставку с пополнением справочников

Ins\_trip:

CREATE DEFINER=`root`@`localhost` PROCEDURE `ins\_trip`(name\_type\_ varchar(30), name\_ varchar(20),

surname\_ varchar(20), patronymic\_ varchar(20), name\_trip\_ varchar(40),cost\_ int(11))

BEGIN

declare id\_type\_trip\_new int;

declare teacher\_id\_teacher\_new int;

declare id\_trip\_new int;

if exists (select \* from type\_trip where name\_type=name\_type\_)

then select id\_type\_trip into id\_type\_trip\_new from type\_trip where name\_type=name\_type\_;

else begin

set id\_type\_trip\_new=(select ifnull(max(id\_type\_trip)+1,0) from type\_trip);

INSERT INTO type\_trip(id\_type\_trip,name\_type) VALUES (id\_type\_trip\_new,name\_type\_);

end;

end if;

if exists (select \* from teacher where name\_=name\_ and surname\_=surname and patronymic\_=patronymic)

then select id\_teacher into teacher\_id\_teacher\_new from teacher where name\_=name\_ and surname\_=surname and patronymic\_=patronymic;

else begin

set teacher\_id\_teacher\_new=(select ifnull(max(id\_teacher)+1,0) from teacher);

INSERT INTO teacher(id\_teacher,name,surname,patronymic) VALUES (teacher\_id\_teacher\_new,name\_,surname\_,patronymic\_);

end;

end if;

set id\_trip\_new=(select ifnull(max(id\_trip)+1,0)from trip);

insert into trip (id\_trip, name\_trip, cost, id\_type\_trip, teacher\_id\_teacher)

VALUES (id\_trip\_new,name\_trip\_,cost\_,id\_type\_trip\_new,teacher\_id\_teacher\_new);

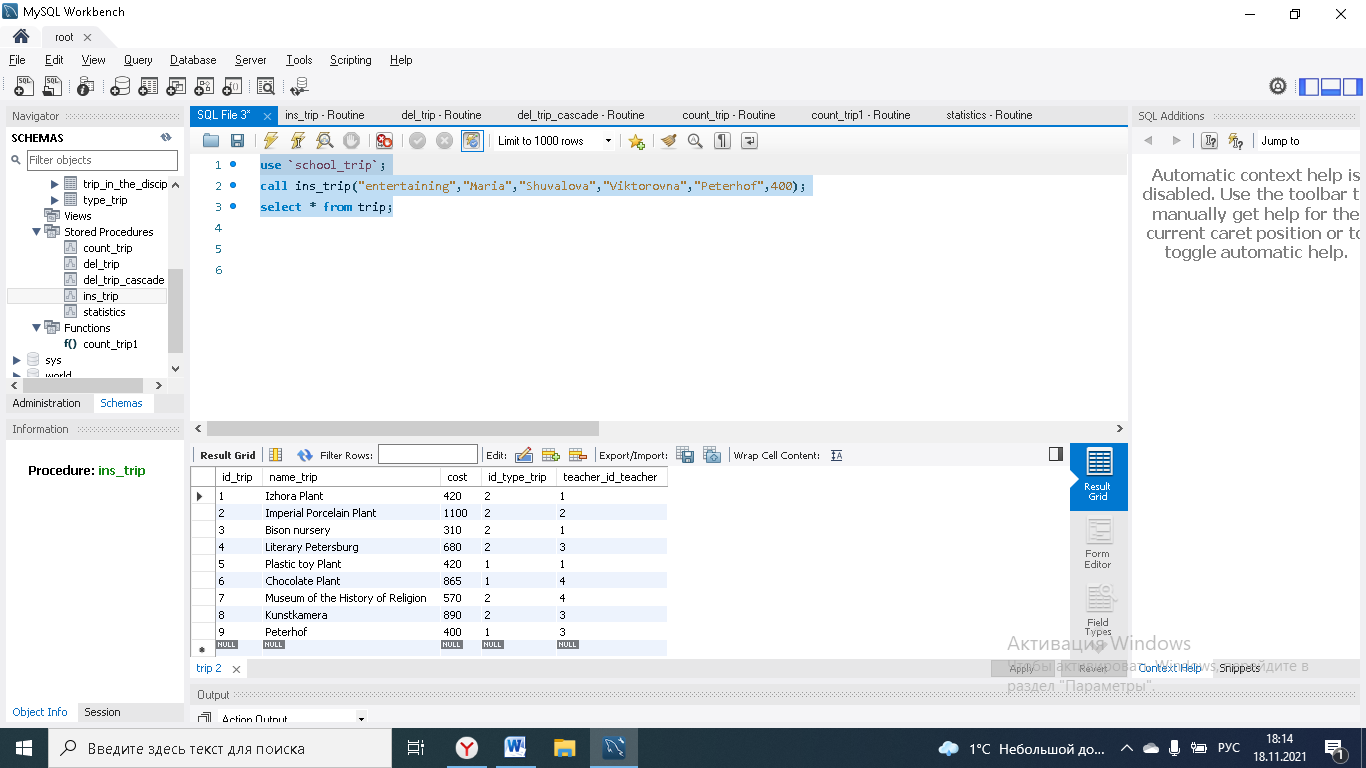
END

Вызов:

use `school\_trip`;

call ins\_trip("entertaining","Maria","Shuvalova","Viktorovna","Peterhof",400);

select \* from trip;



— удаление с очисткой справочников

Del\_trip:

CREATE DEFINER=`root`@`localhost` PROCEDURE `del\_trip`(id\_trip\_del int)

BEGIN

declare id\_type\_trip\_del int;

declare teacher\_id\_teacher\_del int;

select id\_trip into id\_trip\_del from trip where id\_trip=id\_trip\_del;

delete from trip where id\_trip=id\_trip\_del;

if not exists (select \* from type\_trip where id\_type\_trip=id\_type\_trip\_del)

then delete from type\_trip where id\_type\_trip=id\_type\_trip\_del;

end if;

if not exists (select \* from teacher where id\_teacher=teacher\_id\_teacher\_del)

then delete from teacher where id\_teacher=teacher\_id\_teacher\_del;

end if;

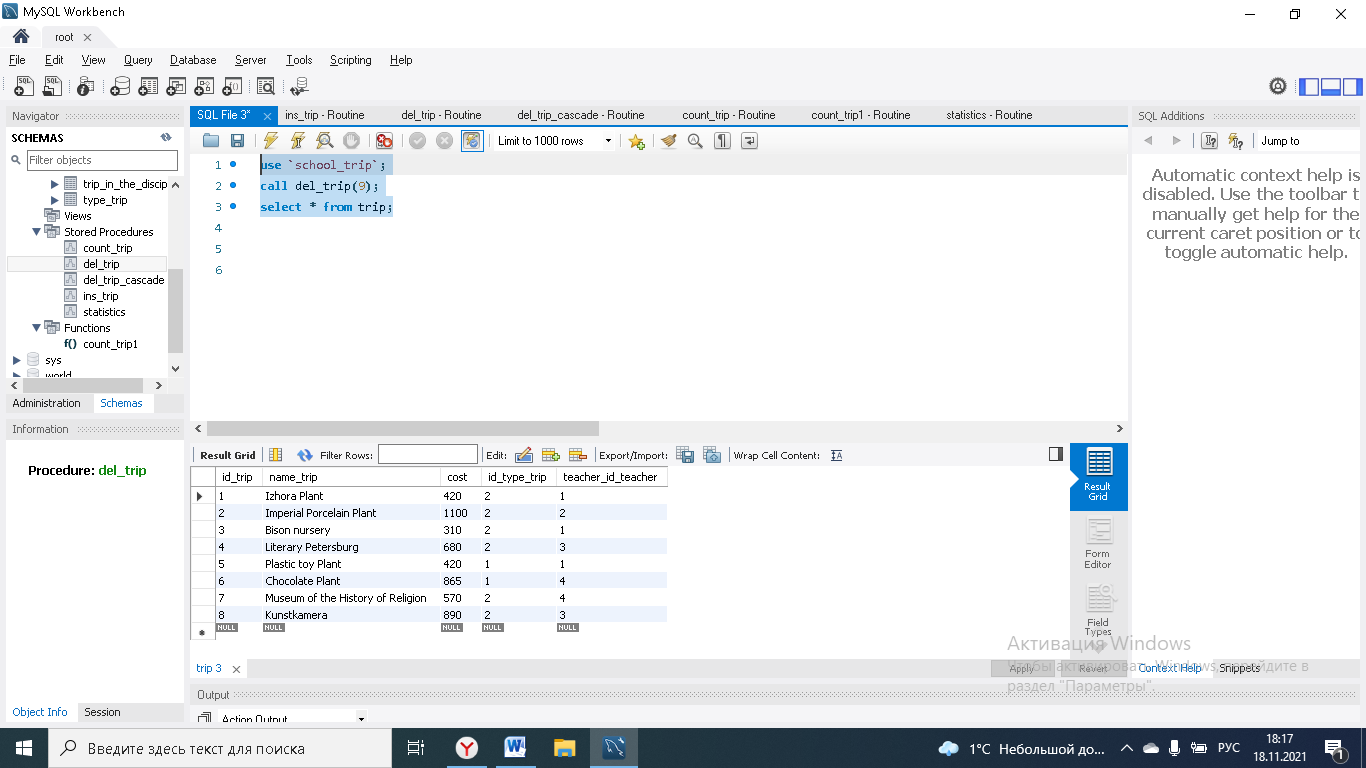
END

Вызов:

use `school\_trip`;

call del\_trip(9);

select \* from trip;



— каскадное удаление

Del\_trip\_cascade:

CREATE DEFINER=`root`@`localhost` PROCEDURE `del\_trip\_cascade`(id\_trip\_del int)

BEGIN

delete from participants\_trip where trip\_id\_trip=id\_trip\_del;

delete from trip\_in\_the\_discipline where trip\_id\_trip=id\_trip\_del;

delete from trip where id\_trip=id\_trip\_del;

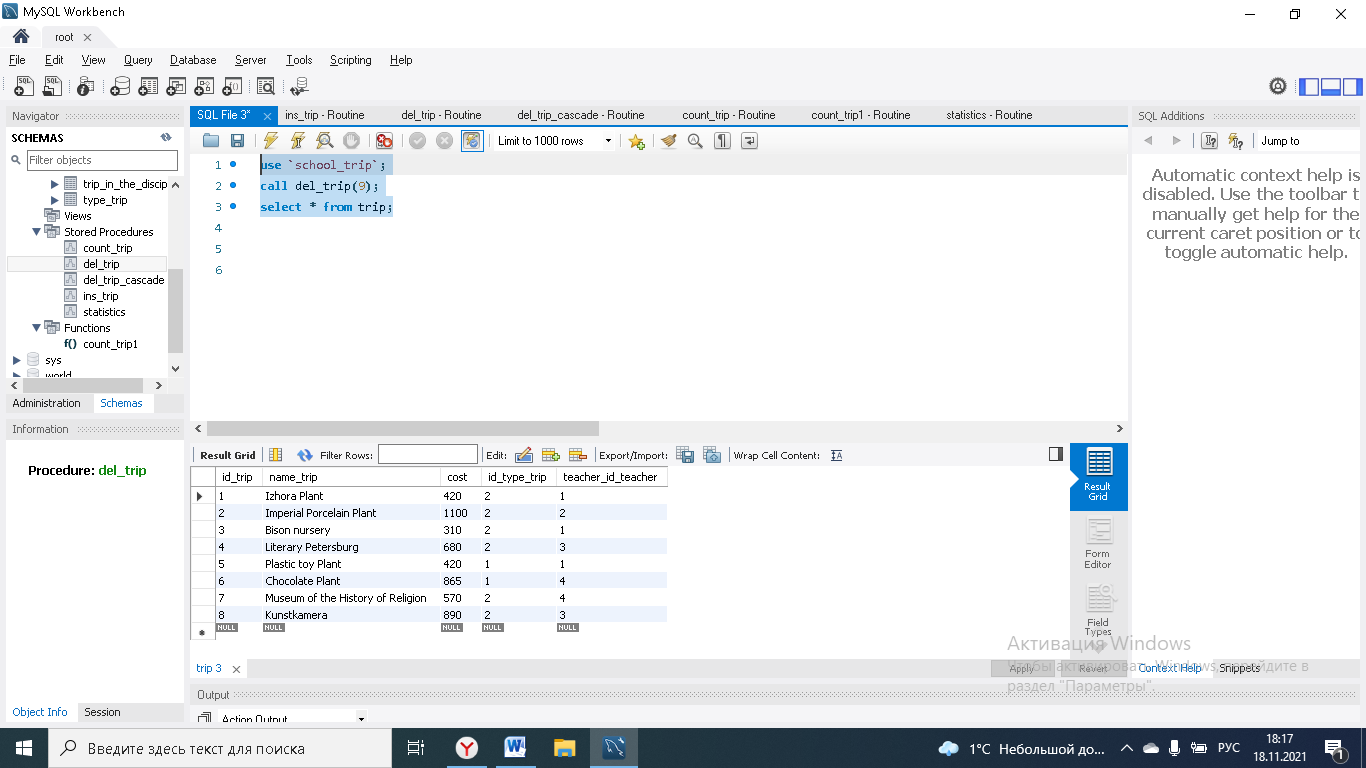
END

Вызов:

use `school\_trip`;

call del\_trip\_cascade(9);

select \* from trip;



— вычисление и возврат значения агрегатной функции

Count\_trip:

CREATE DEFINER=`root`@`localhost` PROCEDURE `count\_trip`(out cnt\_trip int)

BEGIN

select ifnull(count(id\_trip),0) into cnt\_trip from trip;

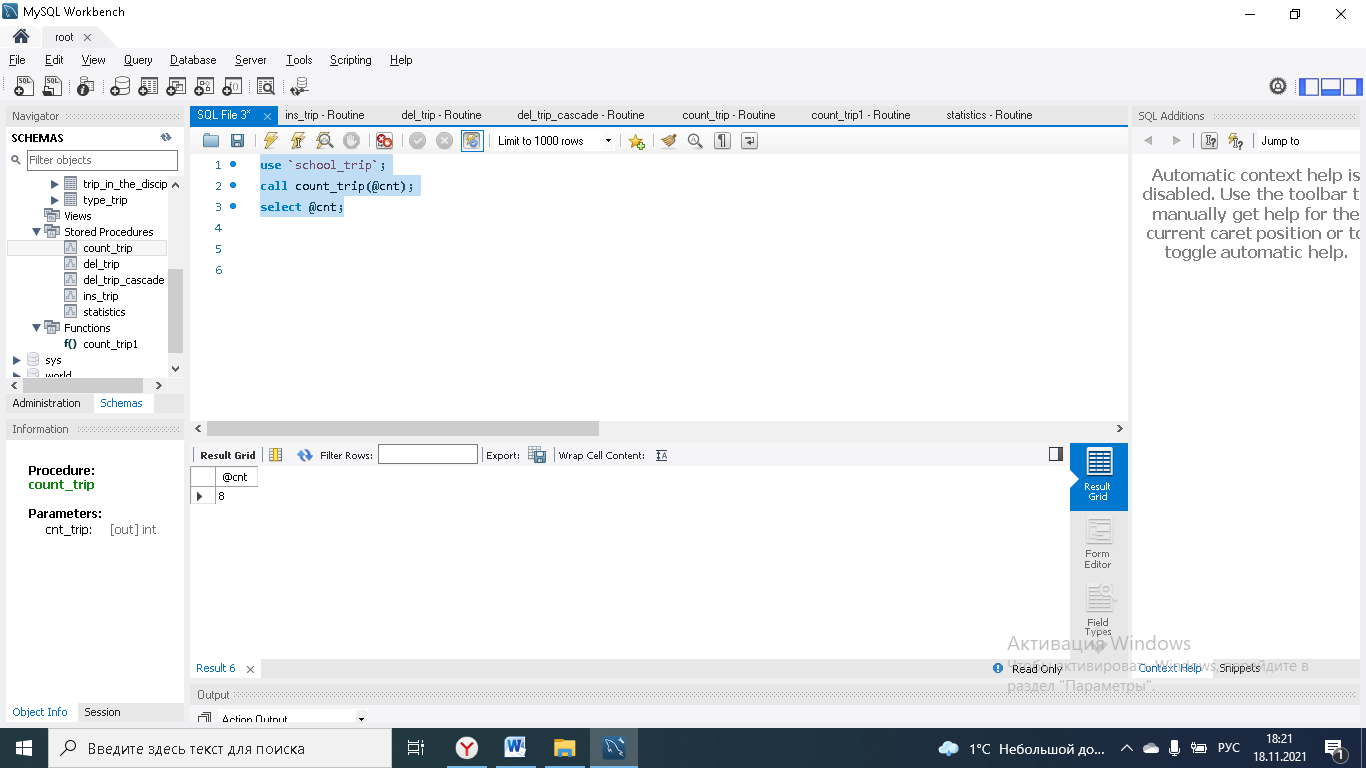
END

Вызов:

use `school\_trip`;

call count\_trip(@cnt);

select @cnt;



Count\_trip1:

CREATE DEFINER=`root`@`localhost` FUNCTION `count\_trip1`() RETURNS int

DETERMINISTIC

BEGIN

declare cnt\_trip int DEFAULT 0;

set cnt\_trip=(select ifnull(count(id\_trip),0) from trip);

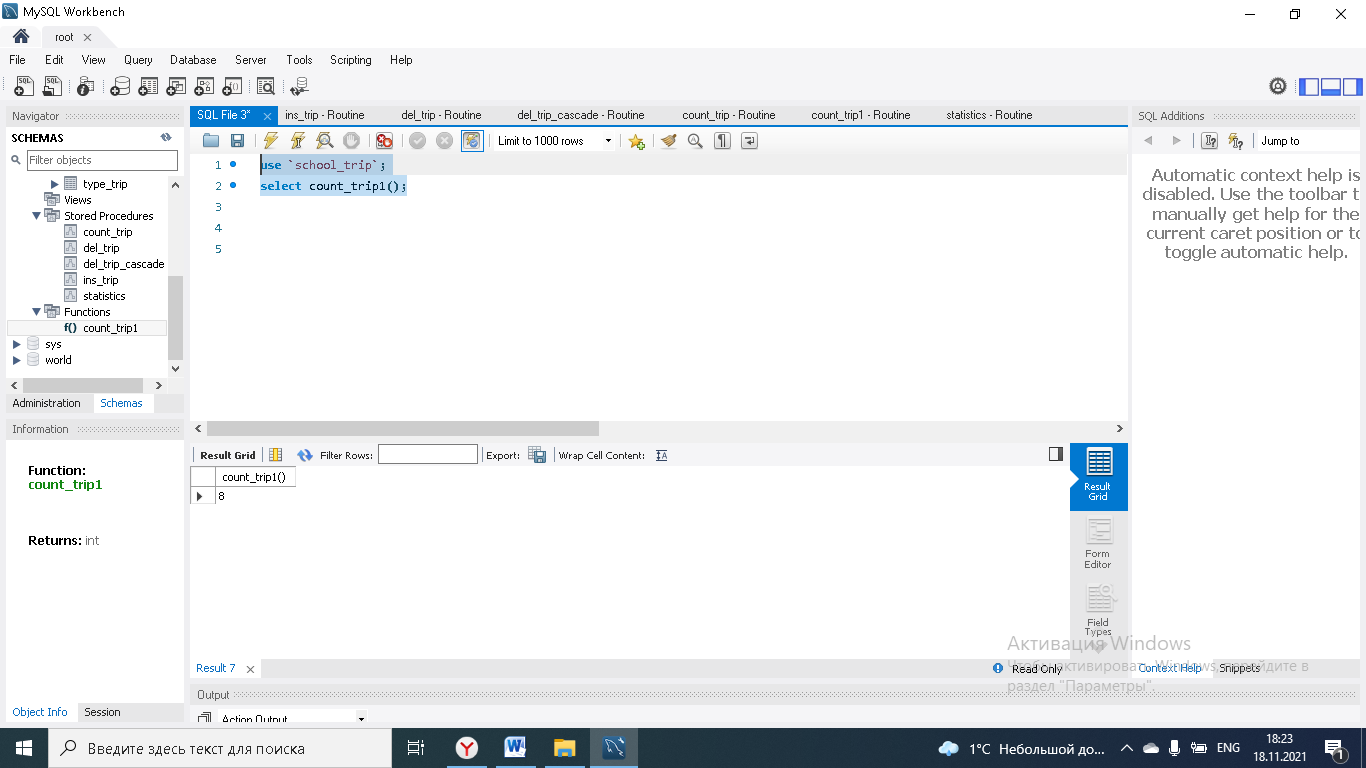
return cnt\_trip;

END

Вызов:

use `school\_trip`;

select count\_trip1();



— формирование статистики во временной таблице

Statatistics:

CREATE DEFINER=`root`@`localhost` PROCEDURE `statistics`()

BEGIN

create temporary table if not exists stat

(

id\_stat int auto\_increment primary key,

id\_st int,

count\_trip int,

avg\_trip int

);

insert into stat(id\_st,count\_trip)

select student.id\_student, count(trip\_id\_trip) as count\_trip from student

left join participants\_trip on student.id\_student=participants\_trip.student\_id\_student

group by id\_student;

update stat set avg\_trip=

(select avg(count\_trip) from (select student.id\_student, count(trip\_id\_trip) as count\_trip from student

left join participants\_trip on student.id\_student=participants\_trip.student\_id\_student

group by id\_student)q);

select \* from stat;

drop table sts;

END

Вызов:

use `school\_trip`;

SET SQL\_SAFE\_UPDATES=0;

call statistics();

SET SQL\_SAFE\_UPDATES=1;

