2.

INSERT INTO owner(idOwner, Lastname\_Ow, Name, Address\_Ow, Number)

VALUES (1,'Петров','Иван','Волкстрит, 13', 89235461284);

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('2', 'Джордано', 'Петр', 'Винная. 6', '89235175834');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('3', 'Волков', 'Эдуард', 'Садовая, 153', '89245617482');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('4', 'Котов', 'Леонид', 'Шпалерная, 12', '89326578934');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('5', 'Поррелло', 'Франко', 'Кваренги, 55', '89244768924');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('6', 'Марчелло', 'Сара', 'Моисеенко, 90', '89154902314');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('7', 'Михайлова', 'Кристина', 'Невский проспект, 102', '89056186901');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('8', 'Аденина', 'Елизавета', 'Казанская, 29', '89013467890');

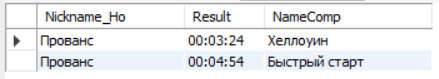
INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('9', 'Маятников', 'Дмитрий', 'Красноармейская, 10', '89132568901');

INSERT INTO `horse\_racing`.`owner` (`idOwner`, `Lastname\_Ow`, `Name`, `Address\_Ow`, `Number`) VALUES ('10', 'Корлеоне', 'Франческо', 'Лиговская, 64', '87904456237');

UPDATE `horse\_racing`.`owner` SET `Lastname\_Ow` = 'Петров', `Name` = 'Иван', `Address\_Ow` = 'Волкстрит, 13' WHERE (`idOwner` = '1');

1 запрос! SELECT horse.Nickname\_Ho , run.Result , competition.NameComp

FROM horse, run, competition

WHERE horse.idHorse = run.idHorse AND run.idCompetition=competition.idCompetition AND horse.Nickname\_Ho='Прованс';

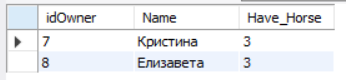
2 запрос! SELECT horse.idOwner, owner.Name, COUNT(horse.idHorse) AS Have\_Horse

FROM horse, owner

WHERE horse.idOwner=owner.idOwner

GROUP BY horse.idOwner

HAVING COUNT(horse.idHorse)>2;



5 запрос!

SELECT competition.NameComp, run.NumberRun, horse.Nickname\_Ho, run.Position

FROM competition,run, horse

WHERE competition.idCompetition=run.idCompetition AND run.idHorse=horse.idHorse

AND horse.Nickname\_Ho= 'Черная стрела' AND run.Position='1'



3 запрос! SELECT competition.NameComp, COUNT(jockey.idJockey) AS Sum\_Jockey

FROM competition,run, jockey

WHERE competition.idCompetition=run.idCompetition

AND run.idJockey=jockey.idJockey AND jockey.Rating>'850'

GROUP BY competition.NameComp

HAVING COUNT(jockey.idJockey)>=4



4 запрос!

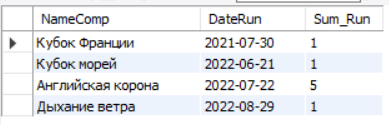
SELECT competition.NameComp, run.DateRun, COUNT(run.DateRun) AS Sum\_Run

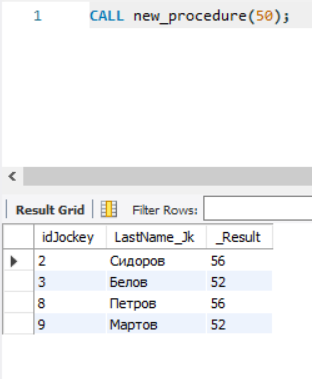
FROM competition,run

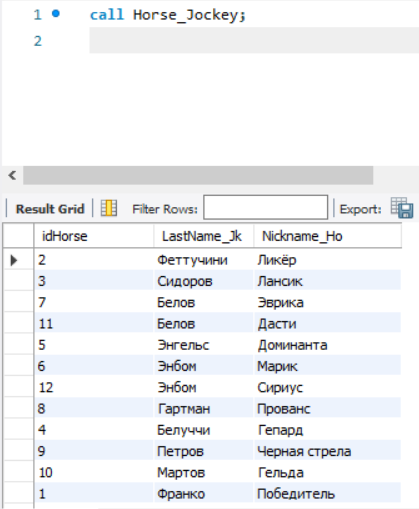
WHERE competition.idCompetition=run.idCompetition AND DATE(DateRun ) > "2021-06-15" AND DATE(DateRun) < "2022-10-31"

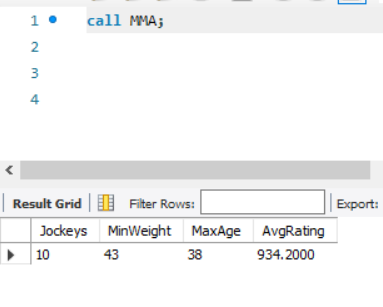
group by competition.NameComp, run.DateRun

ORDER BY run.DateRun









SELECT idRun, idCompetition,

CASE

WHEN DateRun>='2021-08-31' THEN 'Осень'

WHEN DateRun<='2021-05-31' THEN 'Лето'

ELSE 'Другое время года'

END Seasons,

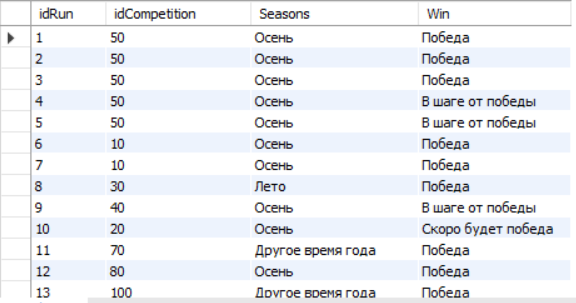
CASE

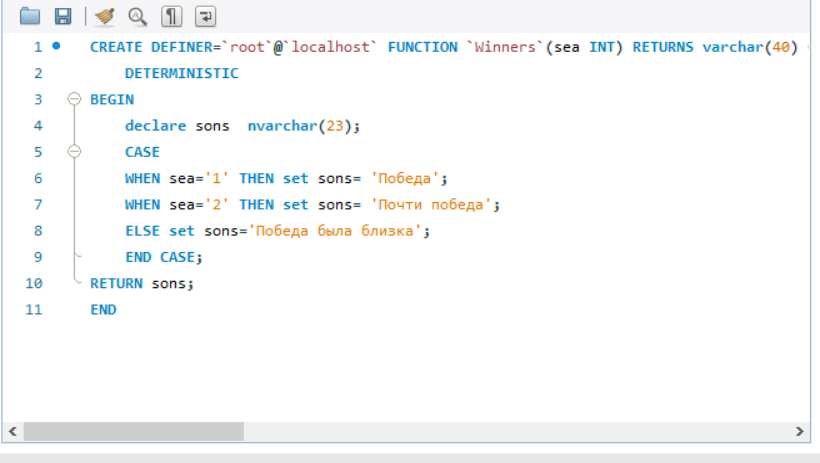
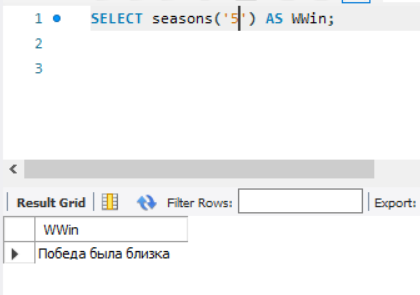
WHEN NumberRun=1 THEN 'Победа'

WHEN NumberRun=2 THEN 'В шаге от победы'

ELSE 'Скоро будет победа'

END Win



CREATE DEFINER=`root`@`localhost` PROCEDURE `race\_cur`()

BEGIN

declare idRacetrack\_ int;

declare Name\_Rt\_ varchar(40);

declare Address\_Rt\_ varchar(40);

declare Lenght\_ int;

declare done int default 0;

declare race\_cur cursor for

select \* from racetrack where racetrack.Lenght>2000;

declare continue handler for sqlstate '02000' set done = 1;

create table if not exists for\_race\_cur like racetrack;

truncate for\_race\_cur;

open race\_cur;

repeat fetch race\_cur into idRacetrack\_, Name\_Rt\_, Address\_Rt\_, Lenght\_;

if not done then

insert into for\_race\_cur set idRacetrack = idRacetrack\_, Name\_Rt=Name\_Rt\_, Address\_Rt=Address\_Rt\_,

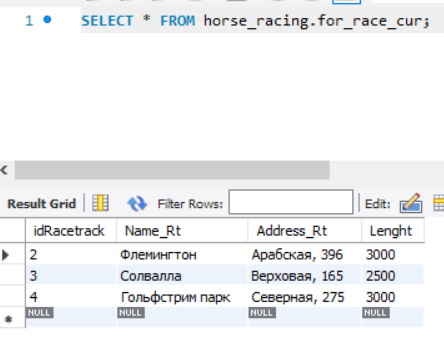
Lenght=Lenght\_;

end if;

until done end repeat;

close race\_cur;

END



1. Создать триггер:

a. запрещающий вставку в таблицу новой строки с заданным параметром;

CREATE DEFINER=`root`@`localhost` TRIGGER `racetrack\_BEFORE\_INSERT` BEFORE INSERT ON `racetrack` FOR EACH ROW BEGIN

IF new.Lenght<=0 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'DO NOT THIS';

END IF;

IF new.Lenght>=4000 THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'DO NOT THIS';

END IF;

END

b. заполняющий одно из полей таблицы на основе вводимых данных.

2. Создать триггер ведения аудита изменения записей в таблицах.

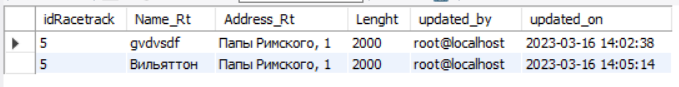
UPDATE `horse\_racing`.`racetrack` SET `Name\_Rt` = 'Вильяттон' WHERE (`idRacetrack` = '5');

CREATE DEFINER=`root`@`localhost` TRIGGER `racetrack\_AFTER\_UPDATE` AFTER UPDATE ON `racetrack` FOR EACH ROW BEGIN

Insert into racetrack\_a\_log(idRacetrack,Name\_Rt,Address\_Rt,Lenght,updated\_by, updated\_on)

VALUES (OLD.idRacetrack,NEW.Name\_Rt,OLD.Address\_Rt,OLD.Lenght,(SELECT USER()), NOW() );

END



3. Внести такие изменения в триггеры вставки и изменения записей таблиц, которые не позволят добавить или изменить записи с дублирующими названиями.

CREATE DEFINER=`root`@`localhost` TRIGGER `owner\_BEFORE\_INSERT` BEFORE INSERT ON `owner` FOR EACH ROW BEGIN

set @found := 0;

select 1 into @found from owner where owner.Number=new.Number;

if @found = 1

then SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'DUPLICATE';

end if;

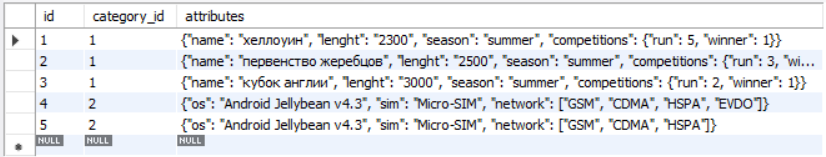
END

Insert into owner(idOwner,Lastname\_Ow,Name,Address\_Ow,Number)

VALUES ('12', '','dggthh','vfdg','89235461284');

JSON

4. Создать новую таблицу или изменить существующую, добавив поле типа JSON, заполнить таблицу данными. Минимум одно из значений записи должно представлять из себя вложенную структуру, одно – массив.

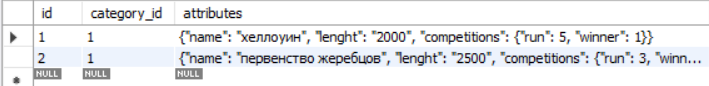


5. Выполнить запрос, возвращающий содержимое данной таблицы, соответствующее некоторому условию, проверяющему значение атрибута вложенной структуры.

SELECT\* FROM products

WHERE category\_id = 1

AND JSON\_EXTRACT(attributes , '$.competitions.run') > 2



6. Выполнить запрос, изменяющий значение по некоторому существующему ключу в заданной строке таблицы.

UPDATE products

SET attributes = JSON\_INSERT(

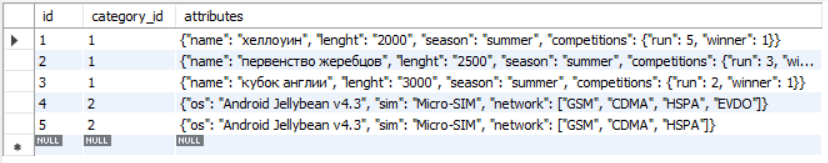
attributes ,

'$.season' ,

'summer'

)

WHERE category\_id = 1;



UPDATE products

SET attributes = JSON\_REPLACE(

attributes ,

'$.lenght' ,

'2300'

)

WHERE id = 1;

