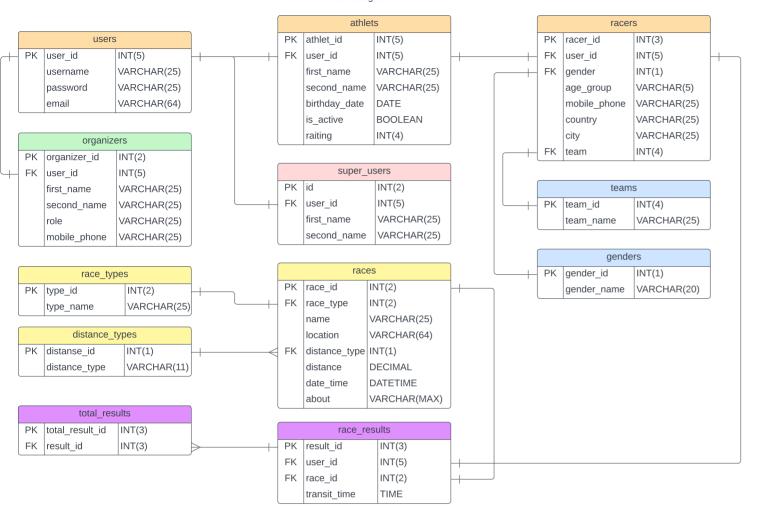


ER diagram



## **RaceTvpe**(Тип гонки)

race_types		S	Кубок объединяет несколько гонок, которые деляться на тип: шоссейные гонки и внедорожные гонки
PK	type_id	INT(2)	- ID типа гонки(заполняется автоматически, автоинкремент)
	type_name	VARCHAR(25)	- Наименования типа гонки(ENUM): road, gravel, crosscountry, downhill, time trial, track race

## DistanceType (Тип дистанции)

distance_types			При подаче заявки на участие в гонке, спортсмен может выбрать одну из типов дистанции.
Pł	PK distanse_id INT(1)		- ID дистанции(заполняется автоматически, автоинкремент)
	distance_type	VARCHAR(11)	- Тип дистанции (ENUM): "Марафон" / "1/2" / "1/4(Дети)"

## **Race** (Гонка)

	races		es	Гонка создается организатором. При создании запоняются следующие поля класса
İ	PK	race_id	INT(2)	- ID гонки(заполняется автоматически, автоинкремент)
	FK	type	INT(2)	- Тип гонки
		name	VARCHAR(25)	- Наименование гонки
		location	VARCHAR(64)	- Местоположение гонки
	FK	distance_type	INT(1)	- Тип дистанции(марафон/половина/четверть)
		distance	DECIMAL	- Протяженность гонки
		date_time	DATETIME	- Дата проведения гонки
		about	VARCHAR(MAX)	- Описание гонки
,		1		

## RaceResults (Результаты одной гонки)

race_results		ults	После каждой гонки формируется результат по гонщикам:.
PK	result_id	INT(3)	- ID результата(заполняется автоматически, автоинкремент)
FK	user_id	INT(5)	- ID пользователя
FK	race_id	INT(2)	- ID гонки
	transit_time	TIME	- Время прохождения дистанции

## TotalResult (Таблица результатов всех гонок)

	total_res	ults	Сводная таблица с орезультатами всех гонок(прошедших)
PK	total_result_id	INT(3)	- ID результата(заполняется автоматически, автоинкремент)
FK	result_id	INT(3)	- ID результата

#### User (Пользователь)

Все пользователи, которые прошли регистрацию для входа на сервис спортивных мероприятий

При регистрации пользователя на сервисе заполнится следующие поля класса:

				при регистрации пользователя на сервисе заполнится следующи
П	PK user_id INT(5)		INT(5)	- ID пользователя(заполняется автоматически, автоинкремент)
		username	VARCHAR(25)	- Имя пользователя для регистрации(уникальное значение)
		password	VARCHAR(25)	- Пароль для регистрации

#### Athlet (Спортсмен)

VARCHAR(64) - Электронная почта для регистрации

Пользователи, которые хотят принять участие в соревнованиях, должны дополнить информацию о себе.

	athlets				
PK	athlet_id	INT(5)			
FK	user_id	INT(5)			
	first_name	VARCHAR(25)			
	second_name	VARCHAR(25)			
	birthday_date	DATE			
	is_active	BOOLEAN			
	raiting	INT(4)			

INT(5)

INT(1)

INT(4)

VARCHAR(5)

VARCHAR(25)

VARCHAR(25)

VARCHAR(25)

emai

FK user\_id

FK gender

age\_group

country

city

FK team

mobile\_phone

Класс *Athlet* - наследник класса User. Запоняются следующие поля класса:

- ID спортсмена(заполняется автоматически, автоинкремент)

- ID пользователя(заполняется автоматически, подтягивается из таблицы users)

- иния спортсмена

- Фамилия спортсмена

- Дата рождения спортсмена (необходима для рассчета возрастной группы спортсмена)

- Поле активности спортсмена (автоматически присваивается значением true. При удалении профиля значение изменяется на false)

- Рейтинг спортсмена (автоматически присваивается значение 0)

## **Racer** (Гонщик)

Пользователи, которые хотят заявиться на участие в гонке, должны дополнить информацию о себе.

 racers
 Класс Racer - наследник класса Athlet. Запоняются следующие поля класса:

 РК | racer\_id | INT(3)
 - ID гонщика(заполняется автоматически, автоинкремент)

- Пол спортсмена (необходим для маркировки группы)

- Возрастная группа (рассчитывается автоматически от даты рождения спортсмена)

- ID пользователя(заполняется автоматически, подтягивается из таблицы users)

- Номер телефона спортсмена

- Страна, за которую выстпует гонщик. Необязательное поле.

Город, за который выстпует гонщик. Необязательное поле.

- Команда, за которую выстпует гонщик (выбор из зарегистрированных команд). Необязательное поле

## Organizer (Организатор)

Пользователи, которые хотят заявиться как организаторы гонок, должны дополнить информацию о себе.

	organizers		ers	Класс Organizer - наследник класса User. Запоняются следующие поля класса:
İ	PK	organizer_id	INT(2)	- ID организатора(заполняется автоматически, автоинкремент)
	FK	user_id	INT(5)	- ID пользователя(заполняется автоматически, подтягивается из таблицы users)
		first_name	VARCHAR(25)	- Имя организотра
		second_name	VARCHAR(25)	- Фамилия организатора
		role	VARCHAR(25)	- Должность/роль в организации соревнований
		mobile_phone	VARCHAR(25)	- Телефон организатора

# SuperUser (Суперюзер)

- Фамилия суперюзера

Пользователи, которые получают права на CRUD информации в базе данных и утверждеие заявок спортсменов на участие в гонках.

super\_usersКласс SuperUser - наследник класса User. Запоняются следующие поля класса:PKidINT(2)- ID суперюзера(заполняется автоматически, автоинкремент)FKuser\_idINT(5)- ID пользователя(заполняется автоматически, подтягивается из таблицы users)first\_nameVARCHAR(25)- Имя суперюзера

## <u>Теат</u> (Команда)

Спортсмены могут объединяться в комманды, которые они должны заявить при регистрации на гонку. Команда подтверждается суперюзером. Для добавления команды необходимо заполнить поля:

	5		
	PK	team_id	INT(4)
		team_name	VARCHAR(25)

second\_name

VARCHAR(25)

- ID команды(заполняется автоматически, автоинкремент)

- Название команды

## <u>Gender</u> (Пол гонщика)

genders			При подаче заявки на гонку, спортсмен обязан выбрать свой пол для определения в группу.
PK	gender_id	INT(1)	- ID гендера(заполняется автоматически, автоинкремент)
	gender_name	VARCHAR(20)	- гендер (ENUM): "M"/"Ж"/"Дети"