

Проект БД Николаев Дмитрий, Пономарева Анна

Разрабатываемая программа будет служить для проведения соревнований по пулевой стрельбе. Во время соревнования участники делают выстрелы, которые сразу регистрируются компьютером, записываются в систему и отображаются на экранах индивидуальных результатов спортсменов и общих экранах результатов соревнований.

Пользователи и их потребности:

- Участник соревнований
 1. Отслеживает результат каждого выстрела
 2. Отслеживает свой общий результат по упражнению
 3. Отслеживает свое положение в турнирной таблице
 4. Отслеживает процесс присвоения ему звания
- Судьи
 1. Следить за выполнением выстрелов
 2. Следить за записью результатов
 3. Выявление победителей
 4. Дисквалифицированные спортсменов
- Тренеры:
 1. Заявляет своих спортсменов на соревнования
 2. Отслеживает участие своих спортсменов на соревновании
 3. Подает заявку на повышение звания своих спортсменов
- Зрители
 1. Наблюдают за результатами спортсменов
 2. Узнают где и когда проводятся соревнования
- Организатор
 1. Готовит общую сводку по результатам соревнований
 2. Сравнивает результаты спортсменов в разных упражнениях и их прогресс
 3. Выдвигает рекомендации, кто из спортсменов должен быть зачислен в сборную

Функциональные требования:

Конечно! Функциональные требования описывают, какие задачи должна выполнять система, какие функции предоставлять пользователям и как обеспечивать обработку данных в соответствии с бизнес-правилами. Ниже

приведены функциональные требования для вашей базы данных, связанной со спортивными соревнованиями.

1. Система должна позволять добавлять нового тренера/организатора/соревнование/выстрел/судью/ оружие/ с указанием атрибутов
2. Система должна позволять вносить изменения в атрибуты сущностей
3. Система НЕ должна позволять удалять данные о судье/тренере/организаторе/спортсмене/упражнениях, которые принимали участие хотя бы в одном соревновании, иначе произойдет потеря исторических данных
4. Возможность добавлять нового спортсмена с указанием всех необходимых атрибутов:
 - a. Имя, фамилия
 - b. Назначение тренера (trainer_id)
 - c. Выбор оружия (weaponNr)
 - d. Дата рождения
 - e. Спортивный клуб
 - f. Регион
 - g. Спортивный разряд или звание (grade)
 - i.
5. Возможность создавать новое соревнование с указанием:
 - a. Названия соревнования
 - b. Даты начала и окончания
 - c. Места проведения
 - d. Уровня соревнования
 - e. Организатора (Organizer_id)
6. Просмотр списка всех соревнований, спортсменов, выстрелов, оружия
7. Возможность удалять соревнование из системы.
8. При удалении соревнования должны автоматически удаляться связанные планы соревнований и записи о выстрелах.
9. Возможность добавлять новое упражнение с указанием:
 - a. Названия упражнения
 - b. Количества выстрелов (Nrshots)
 - c. Расстояния до мишени (distance)
 - d. Типа оружия (weapon_type)
10. Возможность создавать план соревнования с указанием:
 - a. Соревнования (Competition_id)
 - b. Упражнения (Exercise_id)
 - c. Даты и времени проведения
 - d. Стрельбища
 - e. Судьи (Judge_id)

11. Возможность удалять план соревнования из системы. При удалении плана соревнования должны автоматически удаляться связанные записи о выстрелах.
12. Возможность добавлять запись о выстреле спортсмена с указанием:
 - a. Номера выстрела (shot_nr)
 - b. Результата (score)
 - c. Координат выстрела (coord_x, coord_y)
 - d. Времени выстрела
 - e. Спортсмена (Sportsman_id)
 - f. Плана соревнования (Competition_plan_id)
13. Возможность удалять запись о выстреле из системы.
14. Возможность обновлять информацию о звании спортсмена с указанием:
 - a. Упражнения (Exercise_id)
 - b. Оценки (score)
15. Возможность добавлять новое оружие с указанием:
 - a. Типа оружия (weapon_type)
 - b. Названия оружия (weapon_name)
 - c. Номера оружия (weaponNr)
 - d. Ячейки хранения (storage_cell)
16. Возможность добавлять запись о дисквалификации спортсмена с указанием:
 - a. Спортсмена (sportman_id)
 - b. Причины дисквалификации (reason)
17. Возможность редактировать данные дисквалификации, включая причину.
18. Возможность удалять запись о дисквалификации из системы.
19. Автоматическое исключение результатов дисквалифицированных спортсменов из общего подсчета.
20. Система НЕ должна позволять записывать данные, нарушающие ограничения на данные, описанные ниже
21. Создание отчетов о результатах соревнований, судьях, оружии, спортсменах
22. Определение ролей (например, администратор, тренер, организатор, судья) с соответствующими правами доступа к данным.
23. Ограничение доступа к определенным операциям (например, только администраторы могут удалять записи).

UML диаграмма базы данных:

1. Trainer (id: PK, tr_first_name, tr_last_name)
2. Organizer (id: PK, org_first_name, org_last_name)

3. Sportsman(id: PK, _sp_first_name, sp_last_name, trainer_id: FK, grade, weapon_id: FK, birth_date, sports_club, region)
4. Judge (id: PK, first_name, last_name, qualification, certificateNr)
5. Competition(id: PK, titleComp, date_start, date_end , place, level, Organizer_id: FK)
6. Exercise (id: PK, titleEx, Nrshots, distance, weapon_type)
7. Shot (id: PK, shot_nr, score, coord_x, coord_y, time, Sportsman_id: FK, Competition_plan_id: FK)
8. Grade(id: PK, titleGrade, Exercise_id: FK, score)
9. Weapon (id: PK, weapon_type, weapon_name, weaponNr, storage_cell)
10. Competition_plan (id: PK, Competition_id: FK, Exercise_id: FK, date_time, shooting_range, Judge_id: FK)
11. Disqualification(id: PK, sportman_id: FK, reason)

Ограничения на данные:

1. У каждого спортсмена обязательно есть тренер, номер оружия, спортивный клуб и регион, причем каждый признак только один.
2. Каждый спортсмен может иметь или не иметь спортивный разряд или звание (grade)
3. У каждого упражнения на соревнованиях один судья
4. Для каждого судьи обязательно указывается уровень квалификации и номер документа, подтверждающего возможность судить (qualification, certificateNr)
5. У каждого выстрела есть результат причем результат - число от нуля до 10,9 с одним знаком после запятой. Координаты выстрела - отклонения от центра мишени по модулю не превосходят 12.
6. Дата начала соревнования меньше даты окончания соревнования
7. В каждой ячейке для хранения хранится только одно оружие
8. Каждое упражнение выполняется на одном тире для стрельбе
9. Для каждого участника результат упражнения на конкретном соревновании не может быть отрицательным и не может превышать (количество выстрелов в упражнении * 10.9 (максимальное значение выстрела))
10. Причины отстранения от соревнования могут быть только из множества {экипировка, нарушение правил соревнования, неспортивное поведение, допинг, неисправность оружия}
11. Если спортсмен был дисквалифицирован, его результаты не засчитываются
12. Номер выстрела является натуральным числом и принимает значения не более количества выстрелов на определенном упражнении

13. Тип оружия бывает двух видов - пневматический пистолет, пневматическая винтовка, малокалиберный пистолет, малокалиберная винтовка
14. Расстояние до мишени принимает следующие значения {10, 25, 50} (метров)
15. Звание спортсмена принимает значения {1-й разряд, 2-й разряд, 3-й разряд, кмс, мс, мсмк, змс}
16. Номер оружия - последовательность из 10 цифр
17. Одним оружием могут пользоваться несколько спортсменов, но для каждого спортсмена ровно 1 оружие

Функциональные зависимости по таблицам:

1. Trainer (id: PK, tr_first_name, tr_last_name)
 - id → tr_first_name, tr_last_name
2. Organizer (id: PK, org_first_name, org_last_name)
 - id → org_first_name, org_last_name
3. Sportsman(id: PK, sp_first_name, sp_last_name, trainer_id: FK, grade, weapon_id: FK, birth_date, sports_club, region)
 - id → все остальные атрибуты.
4. Judge (id: PK, first_name, last_name, qualification, certificateNr)
 - id → все остальные атрибуты.
5. Competition(id: PK, titleComp, date_start, date_end, place, level, Organizer_id: FK)
 - id → все остальные атрибуты.
6. Exercise(id: PK, titleEx, Nrshots, distance, weapon_type)
 - id → все остальные атрибуты.
7. Shot(id: PK, shot_nr, score, coord_x, coord_y, time, Sportsman_id: FK, Competition_plan_id: FK)
 - id → все остальные атрибуты.
8. Grade(id: PK, titleGrade, Exercise_id: FK, score)
 - id → все остальные атрибуты.
9. Weapon(id: PK, weapon_type, weapon_name, weaponNr, storage_cell)
 - id → все остальные атрибуты.
10. Competition_plan(id: PK, Competition_id: FK, Exercise_id: FK, date_time, shooting_range, Judge_id: FK)
 - id → все остальные атрибуты.
11. Disqualification(id: PK, sportman_id: FK, reason)
 - id → sportman_id, reason

Нормализация предварительной схемы:

Все представленные сущности находятся в **BCNF**, так как для каждой функциональной зависимости левая часть является суперключом.

SQL -запросы для создания таблиц

```
CREATE TYPE weapon_type_enum AS ENUM (  
    'пневматический пистолет',  
    'пневматическая винтовка',  
    'малокалиберный пистолет',  
    'малокалиберная винтовка'  
);
```

```
CREATE TYPE distance_enum AS ENUM ('10', '25', '50');
```

```
CREATE TYPE grade_enum AS ENUM (  
    '1-й разряд',  
    '2-й разряд',  
    '3-й разряд',  
    'кмс',  
    'мс',  
    'мсмк',  
    'змс'  
);
```

```
CREATE TYPE disqualification_reason_enum AS ENUM (  
    'экипировка',  
    'нарушение правил соревнования',
```

```
'неспортивное поведение',  
'допинг',  
'неисправность оружия'  
);
```

```
CREATE TABLE Trainer (  
    id SERIAL PRIMARY KEY,  
    tr_first_name VARCHAR(50) NOT NULL,  
    tr_last_name VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Organizer (  
    id SERIAL PRIMARY KEY,  
    org_first_name VARCHAR(50) NOT NULL,  
    org_last_name VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Weapon (  
    id SERIAL PRIMARY KEY,  
    weapon_type weapon_type_enum NOT NULL,  
    weapon_name VARCHAR(100) NOT NULL,  
    weaponNr CHAR(10) UNIQUE NOT NULL,  
    storage_cell VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE Sportsman (  
    id SERIAL PRIMARY KEY,
```

```
sp_first_name VARCHAR(50) NOT NULL,  
sp_last_name VARCHAR(50) NOT NULL,  
trainer_id INTEGER NOT NULL,  
grade grade_enum,  
weaponNr CHAR(10) NOT NULL,  
birth_date DATE NOT NULL,  
sports_club VARCHAR(100) NOT NULL,  
region VARCHAR(100) NOT NULL,  
CONSTRAINT fk_trainer  
    FOREIGN KEY(trainer_id)  
        REFERENCES Trainer(id)  
        ON DELETE CASCADE,  
CONSTRAINT fk_weapon  
    FOREIGN KEY(weaponNr)  
        REFERENCES Weapon(weaponNr)  
        ON DELETE RESTRICT,  
CONSTRAINT chk_birth_date  
    CHECK (birth_date <= CURRENT_DATE)  
);
```

```
CREATE TABLE Judge (  
    id SERIAL PRIMARY KEY,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    qualification VARCHAR(100) NOT NULL,  
    certificateNr VARCHAR(50) UNIQUE NOT NULL  
);
```



```
CREATE TABLE Competition (  
    id SERIAL PRIMARY KEY,  
    titleComp VARCHAR(100) NOT NULL,  
    date_start DATE NOT NULL,  
    date_end DATE NOT NULL,  
    place VARCHAR(100) NOT NULL,  
    level VARCHAR(50) NOT NULL,  
    Organizer_id INTEGER NOT NULL,  
    CONSTRAINT fk_organizer  
        FOREIGN KEY(Organizer_id)  
        REFERENCES Organizer(id)  
        ON DELETE CASCADE,  
    CONSTRAINT chk_dates  
        CHECK (date_start < date_end)  
);
```

```
CREATE TABLE Exercise (  
    id SERIAL PRIMARY KEY,  
    titleEx VARCHAR(100) NOT NULL,  
    Nrshots INTEGER NOT NULL CHECK (Nrshots > 0),  
    distance distance_enum NOT NULL,  
    weapon_type weapon_type_enum NOT NULL  
);
```

```
CREATE TABLE Competition_plan (  
    id SERIAL PRIMARY KEY,
```

```
Competition_id INTEGER NOT NULL,  
Exercise_id INTEGER NOT NULL,  
date_time TIMESTAMP NOT NULL,  
shooting_range VARCHAR(100) NOT NULL,  
Judge_id INTEGER NOT NULL,  
CONSTRAINT fk_competition  
    FOREIGN KEY(Competition_id)  
        REFERENCES Competition(id)  
        ON DELETE CASCADE,  
CONSTRAINT fk_exercise  
    FOREIGN KEY(Exercise_id)  
        REFERENCES Exercise(id)  
        ON DELETE CASCADE,  
CONSTRAINT fk_judge  
    FOREIGN KEY(Judge_id)  
        REFERENCES Judge(id)  
        ON DELETE RESTRICT  
);
```

```
CREATE TABLE Shot (  
    id SERIAL PRIMARY KEY,  
    shot_nr INTEGER NOT NULL,  
    score NUMERIC(3,1) NOT NULL CHECK (score >= 0 AND score <= 10.9),  
    coord_x NUMERIC(5,2) NOT NULL CHECK (ABS(coord_x) <= 12),  
    coord_y NUMERIC(5,2) NOT NULL CHECK (ABS(coord_y) <= 12),  
    time TIME NOT NULL,  
    Sportsman_id INTEGER NOT NULL,
```

```
Competition_plan_id INTEGER NOT NULL,  
CONSTRAINT fk_sportsman  
    FOREIGN KEY(Sportsman_id)  
        REFERENCES Sportsman(id)  
        ON DELETE CASCADE,  
CONSTRAINT fk_competition_plan  
    FOREIGN KEY(Competition_plan_id)  
        REFERENCES Competition_plan(id)  
        ON DELETE CASCADE,  
CONSTRAINT chk_shot_nr  
    CHECK (shot_nr > 0)  
);
```

```
CREATE TABLE Grade (  
    id SERIAL PRIMARY KEY,  
    titleGrade grade_enum NOT NULL,  
    Exercise_id INTEGER NOT NULL,  
    score NUMERIC(5,2) NOT NULL CHECK (score >= 0),  
    CONSTRAINT fk_exercise_grade  
        FOREIGN KEY(Exercise_id)  
            REFERENCES Exercise(id)  
            ON DELETE CASCADE  
);
```

```
CREATE TABLE Disqualification (  
    id SERIAL PRIMARY KEY,  
    sportman_id INTEGER NOT NULL,
```

```

reason disqualification_reason_enum NOT NULL,
CONSTRAINT fk_sportsman_disq
    FOREIGN KEY(sportman_id)
        REFERENCES Sportsman(id)
    ON DELETE CASCADE
);

-- Ограничение на уникальность сочетания Competition_id и Exercise_id в
Competition_plan

ALTER TABLE Competition_plan ADD CONSTRAINT unique_comp_exercise UNIQUE
(Competition_id, Exercise_id);

ALTER TABLE Weapon

ADD CONSTRAINT unique_storage_cell UNIQUE (storage_cell);

```

Запросы:

1. Найти топ 3 в соревновании по каждому упражнению для соревнования с id = 1

```

SELECT Exercise.titleEx, sp_first_name, sp_last_name
FROM Shot
    INNER JOIN Sportsman ON Sportsman.id = Shot.Sportsman_id
    INNER JOIN Competition_plan ON Shot.Competition_plan_id =
Competition_plan.id
    INNER JOIN Disqualification ON Disqualification.Sportsman_id !=
Sportsman.id
    INNER JOIN Exercise ON Competition_plan.Exercise_id = Exercise.id
GROUP BY Exercise.id

```

```
INNER JOIN Competition ON Competition.id = 1 AND Competition.id  
= Competition_id AND Competition_plan.Competition_id = Competition.id
```

```
GROUP BY Exercise.titleEx
```

```
ORDER BY Exercise.titleEx, SUM(score) DESC LIMIT 3
```

- 2. Вывести результаты конкретного спортсмена (Иванова Ивана) в конкретном упражнении (ВП-60) на данном соревновании (Первенство России 2024)**

```
WITH cte AS(  
SELECT NTILE (6) OVER (ORDER BY sh.shot_nr) AS gr, score  
FROM Shot sh  
LEFT JOIN Sportsman sp ON sh.Sportsman_id = sp.id  
LEFT JOIN Competition_plan cpl ON sh.Competition_plan_id = comp.id  
LEFT JOIN Exercise ex ON cpl.Exercise_id = ex.id  
LEFT JOIN Competition comp ON cpl.Competition_id = comp.titleComp  
LEFT JOIN Disqualification dis ON dis.sportsman_id != sp.id  
WHERE sp.sp_first_name = 'Иван'  
AND sp.sp_last_name = 'Иванов'  
AND ex.titleEx = 'ВП-60'  
AND comp.titleComp = 'Первенство России 2024')  
SELECT gr, (SELECT SUM(score) FROM cte), SUM(score)  
FROM cte  
GROUP BY gr;
```

- 3. Посчитаем эффективность судей (эффективным будем считать того, кто отсудил 10 раз и более)**

```
SELECT sp_first_name, sp_last_name, COUNT(*)
```

```
FROM Competition_plan
    INNER JOIN Judge ON Competition_plan.Judge_id = Judge.id
GROUP BY Judge.first_name, Judge.second_name
HAVING COUNT(*) > 9
ORDER BY COUNT(*) DESC
```

4. Найти среднюю точность каждой модели оружия

```
SELECT weapon_name, AVG(score)
FROM Shot
    INNER JOIN Sportsman ON Sportsman.id = Shot.Sportsman_id
    INNER JOIN Disqualification ON Disqualification.Sportsman_id !=
Sportsman.id
    INNER JOIN Weapon ON Weapon.Id = Sportsman.weapon_id
GROUP BY weapon_name
ORDER BY AVG(score) DESC
```

5. Проверка выполнения нормативов на мастера спорта спортсменов у тренера с id = 1 по упражнению 'ВП-60' на последнем соревновании

```
SELECT SUM(score) as total_res, sp.last_name AS last_name , sp.first_name AS
first_name
FROM Shot sh
    LEFT JOIN Sportsman sp ON sh.Sportsman_id = sp.id
    LEFT JOIN Competition_plan cpl ON sh.Competition_plan_id = comp.id
    LEFT JOIN Exercise ex ON cpl.Exercise_id = ex.id
    LEFT JOIN Competition comp ON cpl.Competition_id = comp.titleComp
    LEFT JOIN Grade g ON g.id = sp.grade
    LEFT JOIN Disqualification dis ON dis.sportsman_id != sp.id
```

```
WHERE sp.trainer_id = '1'

      AND date_time = (SELECT MAX(datetime) FROM Competition_plan)

GROUP BY sp.last_name, sp.first_name

HAVING SUM(score) >= (SELECT score FROM grade WHERE titleGrade =
'мс' )

ORDER BY last_name, first_name
```

Группировка запросов в транзакции

1. Замена региона влечет замену тренера

```
BEGIN
```

```
UPDATE Sportsman
```

```
SET region = 'MCK'
```

```
WHERE Sportsman_id = 1
```

```
UPDATE Sportsman
```

```
SET trainer_id = 555
```

```
WHERE id = 1
```

```
COMMIT
```

```
BEGIN;
```

2. Перенос соревнования на другую дату и обновление плана соревнования

```
BEGIN;
```

```
-- Перенос соревнования с id=1 на новые даты
```

```
UPDATE Competition
```

```
SET date_start = '2024-10-01', date_end = '2024-10-05'
```

```
WHERE id = 1;
```

-- Обновление дат и времени в плане соревнования

UPDATE Competition_plan

SET date_time = date_time + INTERVAL '30 days'

WHERE Competition_id = 1;

COMMIT;

3. Обновление данных о судьях и их назначениях

BEGIN;

-- Обновление квалификации судьи с id=8

UPDATE Judge

SET qualification = 'Судья первого класса'

WHERE id = 8;

-- Обновление номера сертификата судьи с id=8

UPDATE Judge

SET certificateNr = 'CERT-778899'

WHERE id = 8;

-- Обновление судьи на упражнении в соревновании с id=14

UPDATE Competition_plan

SET Judge_id = 8

WHERE Competition_id = 14 AND Exercise_id = 4;

COMMIT;