

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ

Мелков Никита Александрович

Курсовая работа

Литературные поединки
(VERSUS BATTLE)

Направление 010400
Прикладная математика и информатика

Преподаватель

Малинин К.А.

Санкт-Петербург

2017

Содержание

Глава 1 Схема	3
Глава 2 Описание базы данных	5
Глава 3 Легкие запросы	11
Глава 4 Сложные запросы.....	16

Глава 1 Схема

Здесь представлена структура базы данных «Versus Battle».

Participants (участники)

- Partname (псевдоним)
- Name
- Surname
- Birthday

Cities

- Partname
- City

AlternativeNames

- Partname
- A.k.a. (Also known as)

Battle

- NumberOfBattle
- Part1 (Участник 1)
- Part2 (Участник 2)
- Winner
- SupportLevel1
- SupportLevel2
- SeasonNumber
- Date

Season

- SeasonNumber
- Place
- MC (Master of Ceremonies)
- CamQ (Number of Cameras)
- Value

Глава 2 Описание базы данных

База данных «Литературные поединки» описывает участников Versus Battle и их взаимодействия в проекте.

Описание взаимоотношений объектов Versus - это площадка для проведения литературных поединков между поэтами-рэперами (не вымышленные). Участники становятся друг напротив друга, ведущий называет их псевдонимы, кидает жребий и кто-то из участников начинает читать свой раунд против оппонента. Потом свой раунд читает второй участник. Во время раундов зрители шумят/аплодируют на хорошие строки, поддерживают участника. Определяется победитель или, в редких случаях, ничья. Всё происходящее снимается на камеры. Само действие проходит в каком-то заведении. Существует 9 сезонов, у каждого свой ведущий/кол-во камер/место. В один день может проходить несколько противостояний. Участников строго двое на одном сражении (батле), у каждого участника есть свой псевдоним, бессменный и единственный в рамках проекта Versus, но, возможно, есть и другие, вне рамок Versus-a. Также у участников есть известные имя-фамилия, по правилам отличные от псевдонима. Каждый участник из какого-то города (городов), также города могут быть по-разному записаны (Москва, Мск).

Participants

В этой таблице представлена краткая информация об участниках

Partname PRIMARY KEY

Псевдоним участника. Уникален, поэтому можно использовать его в качестве РК, тип text

Name, Surname

Имя и фамилия участника, тип text

Birthday

Дата рождения человека, тип date

Cities

Город(-а), который представляет участник. Имеет смысл разделять с Participants, потому что может быть несколько

Partname PRIMARY KEY

Псевдоним участника, внешний ключ к Participants Partname, text

City

Собственно город, текстовый тип

AlternativeNames

Здесь записаны все другие никнеймы участников, если таковые имеются

Partname **PRIMARY KEY**

Внешний ключ к Participants Partname, text

A.k.a.

Никнейм, text

Battle

NumberOfBattle **PRIMARY KEY**

Номер батла, integer

Part1

Никнейм первого участника, внешний ключ на Participants Partname, text

Part2

Никнейм второго участника, внешний ключ на Participants Partname, text

Winner

Победитель. Ограничен от 1 до 3, int

1 - победил Part1, 2 - победил Part2, 3 - ничья

SupportLevel1

Уровень поддержки зрителями первого выступающего, от 0 до 5, int

SupportLevel2

Аналогично для второго, int

SeasonNumber

Номер сезона, int, внешний ключ на Season.SeasonNumber

Date

Дата проведения батла

Season

Третья сущность - сезон,

SeasonNumber PRIMARY KEY

Собственно номер сезона, int

Place

Место проведения, text

MC

Ведущий, text

CamQ

Количество камер, снимающих все батлы в течение сезона, int

Value

Затраты на создание одного видео, int

Глава 3 Легкие и средние запросы

1. Выводит столбец с участниками

```
Select "Partname" From "Participants";
```

2.Выводит на экран участников, имена которых начинаются на М

```
SELECT "Participants"."Partname"  
FROM "Participants"  
WHERE "Participants"."Partname" LIKE 'M%';
```

3.Выводит номера сезонов, стоимость выпуска которых больше, чем 2000

```
SELECT "Season"."SeasonNumber"  
FROM "Season"  
WHERE (("Season"."Value") > 2000);
```

4. вывести дату рождения самого молодого участника

```
SELECT max("Birthday") as Youngest FROM public."Participants"
```

1. Происходит склейка трёх таблиц: таблица с участниками, с их городами и с их альтернативными именами, т.е. выводится вся сущность.

```
SELECT * FROM ( "Participants" LEFT JOIN "Cities" USING ("Partname") )  
LEFT JOIN "AlternativeNames" USING ("Partname")
```

2 средний - Вывести никнеймы проигравших в первых 5 батлах (по сортировке) в первых 3 сезонах, которые выступали вторыми, а так же место проведения.

```
Select "Part2", "Place" from "Battle"  
left join "Season" using ("SeasonNumber")  
where (("SeasonNumber" < 4) and ("Winner"=1))  
order by "NumberOfBattle"  
limit 5;
```

Глава 4 Сложные запросы

1 сложный: вывести на экран ведущих первых пяти батлов

```
select distinct "MC" from (Select * from "Battle" left join "Season" using  
("SeasonNumber") limit 5) as ss;
```

2 сложный - Вывести цену работы каждого оператора и средний уровень поддержки в поединках, сыгранных вничью. Отсортировать по цене.

```
Select ("Value" / "CamQ") As "Quantity", ((("SupportLevel1"+"SupportLevel2")/2) as  
"Support"  
from "Battle"  
inner join "Season"  
on ("Battle"."SeasonNumber"="Season"."SeasonNumber" and ("Winner"=3))  
order by "Quantity";
```

3 сложный - Вывести никнеймы и один из альтернативных никнеймов, дату рождения и номера всех поединков у участников, которым больше 23 лет, сортировать по дате рождения

```
Select "Battlepart"."Partname", "NumberOfBattle", "Birthday", "A.k.a." as "AlterName"  
from (Select * from "Participants"  
inner join "Battle"  
on ("Participants"."Partname"="Battle"."Part1" or  
"Participants"."Partname"="Battle"."Part2"))  
as "Battlepart"  
inner join (Select distinct on("Partname") * from "AlternativeNames") as  
"AlternativeNames"  
on (("Battlepart"."Partname"="AlternativeNames"."Partname")  
and ("Battlepart"."Birthday"<date(current_date - interval '23 year')))  
order by "Battlepart"."Birthday" desc
```

Ссылка на репозиторий: <https://github.com/ketami/postgre/>