

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ - ПРОЦЕССОВ
УПРАВЛЕНИЯ**

КУРСОВАЯ РАБОТА

по дисциплине: “Базы данных и сетевые технологии”

на тему:

“База данных “Фанфики” (Ficbooks)”

Выполнил:

Студент 2 курса 17.Б03-пу группы

Писарева Софья Максимовна

Преподаватель:

Афонин С.М.

Санкт-Петербург

2018 г.

Содержание

Глава 1 Схема.....	2
Глава 2 Описание базы данных.....	4
Глава 3 Легкие запросы.....	9
Глава 4 Средние запросы.....	13
Глава 5 Сложные запросы.....	16
Ссылка на репозиторий: https://github.com/LazySofi/MyFirstDatabase	

Глава 1 Схема

Здесь представлена структура базы данных “Фанфики”.

Origin

- id
- name
- genre_id
- language_id
- writer_id
- release_date
- translation
- preface

Writer

- id
- name
- birth_date
- death_date
- language_id
- information

Ficbook

- id
- name
- genre_id
- ficwriter_id
- preface

Origin_ficbook

- origin_id
- ficbook_id

Language

- id
- name

Genre

- id
- name

Ficwriter

- id
- name

Глава 2 Описание базы данных

Фанфíк — жаргонизм, обозначающий любительское сочинение по мотивам популярных оригинальных литературных произведений, произведений киноискусства, комиксов, а также компьютерных игр и т. д. Авторами подобных сочинений — *фикра́йтерами* (англ. *Fic-writer* [fɪk'raɪtər]), — как правило, становятся поклонники оригинальных произведений. Обычно фанфики создаются на некоммерческой основе (для чтения другими поклонниками).

База данных “Фанфики” может помочь хранить информацию о написанных фанфиках и оригинальных произведениях, а также их авторах.

Описание взаимоотношений объектов

У каждой книги обязательно должен быть автор. Книги пишутся на каком-то языке и в каком-то жанре, они выходят в релиз в определенный день. Каждая книга имеет определенное количество переводов на различные языки. Также у книг можно посмотреть описание. А у авторов есть личная информация.

Таблица Первоисточников и таблица Фанфиков соединяется отношением m:m, так как в одном фанфике могут использоваться герои и события из разных первоисточников, и на один первоисточник может быть сделано несколько фанфиков.

Таблица Первоисточников и таблица Писателей соединяются отношением 1:m, так как у первоисточника один автор, а у Писателя может быть несколько произведений.

Origin

В этой таблице представлены наиболее интересные характеристики книг.

id PRIMARY KEY

Использован тип integer.

name

(Название книги). Название книги не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничение на написание отсутствует, кроме размера не более 50 символов).

genre_id

В данном поле записано id жанра книги и прикреплена ссылка (внешний ключ) к отдельной таблице с типами жанров.

language_id

В данном поле записано id языка книги и прикреплена ссылка (внешний ключ) к отдельной таблице с видами языков.

writer_id

В данном поле записано id автора книги и прикреплена ссылка (внешний ключ) к отдельной таблице с писателями.

release_date

Это поле показывает когда вышла книга (дату релиза). На дату наложено ограничение: дата не должна быть больше сегодняшней.

translation

Здесь записано на сколько языков была переведена книга. Наложено ограничение > 0 .

preface

Описание книги. Ограничение на размер в 1000 символов.

Writer

Эта таблица описывает писателей. В ней можно найти интересную информацию о них.

id PRIMARY KEY

Использован тип integer.

name

(Имя автора). Имя автора не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничения на написание: размер: не менее 1 и не более 50 символов; ФИО не должно совпадать с Иванов Иван Иванович).

birth_date

Дата рождения автора. Наложено ограничение: дата смерти не раньше даты рождения.

death_date

Дата смерти автора. Наложено ограничение: дата смерти не раньше даты рождения.

language_id

В данном поле записано id языка на котором говорил писатель и прикреплена ссылка (внешний ключ) к отдельной таблице с видами языков.

Information

Некоторая информация о писателе, не поддающаяся систематическому распределению. Ограничение на размер в 1000 символов, а также нельзя упоминать что писатель не любит яблоки.

Ficbook

В этой таблице представлены наиболее интересные характеристики фанфиков.

id PRIMARY KEY

Использован тип integer.

name

(Название фанфика). Название фанфика не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничение на написание отсутствует, кроме размера не более 50 символов, также есть проверка на нецензурные слова).

genre_id

В данном поле записано id жанра книги и прикреплена ссылка (внешний ключ) к отдельной таблице с типами жанров.

ficwriter_id

В данном поле записано id автора и прикреплена ссылка (внешний ключ) к отдельной таблице с фикрайтерами.

preface

Описание книги. Ограничение на размер в 1000 символов, также если описание есть, то оно не может быть пустым.

Origin_ficbook

Эта таблица отношения оригинальных произведений и фикбуков.

origin_id

В данном поле записано id оригинальной книги и прикреплена ссылка (внешний ключ) к отдельной таблице с книгами первоисточниками.

ficbook_id

В данном поле записано id фикбука и прикреплена ссылка (внешний ключ) к отдельной таблице с фикбуками.

Language

Эта таблица с различными языками.

id PRIMARY KEY

Использован тип integer.

name

(Название языка). Название языка не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничение на размер не более 50 символов, также есть проверка на то что этот язык должен быть настоящий).

Genre

Эта таблица с различными жанрами.

id PRIMARY KEY

Использован тип integer.

name

(Название жанра). Название жанра не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничение на написание отсутствует, кроме размера не более 50 символов).

Ficwriter

Эта таблица с профилями фикрайтеров.

id PRIMARY KEY

Использован тип integer.

name

(Никнейм фикрайтера). Имя а не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничение на размер не более 50 символов, также должно быть зарезервировано имя admin).

Глава 3 Легкие запросы

1 запрос

Выбрать писателя и информацию о нем, имя которого равно заданному значению.

```
SELECT w.name, w.information  
      FROM  
            writer w  
      WHERE  
            w.name = _NAME_;
```

Допустимые параметры _NAME_:

- 'Joanne Rowling'
- 'Sir Arthur Ignatius Conan Doyle'
- 'Mikhail Yuryevich Lermontov'

Необходимость:

Мы собираемся изучить биографию писателя, и для этого нам нужно узнать о нем больше.

Пояснение:

Результат запроса выведет два столбца. В первом будет имя писателя, во втором информация о нем (если она указана в базе).

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON writer (name text_pattern_ops);
```

2 запрос

Выбрать книгу, дату выпуска и то на сколько языков был переведен, вывести первые 5.

```
SELECT o.name, o.release_date, o.translation  
      FROM
```

```
        origin o
WHERE
        o.release_date > _DATE_
AND
        o.translation > _COUNT_
ORDER BY
        o.release_date DESC NULLS LAST
LIMIT 5;
```

Допустимые параметры _DATE_

- '1900_01_01'
- '1990_01_01'
- '1800_01_01'

Допустимые параметры _COUNT_

- 10
- 20
- 150

Необходимость:

Мы ищем книги после определенной даты и хотим посмотреть на скольких языках была издана эта книга.

Пояснения:

По запросу будет выведено три столбца. В первом будет название книги, во втором дата выпуска, в третьем кол-во переводов на различные языки.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON origin (release_date, translation);
```

3 запрос

Выбрать писателя, дату его рождения и дату его смерти. Вывести первые 5.

```

SELECT w.name, w.birth_date, w.death_date
      FROM
            writer w
      WHERE
            w.death_date - w.birth_date > _DAYS_
      ORDER BY
            char_length(w.name) DESC NULLS LAST
LIMIT 5;

```

Допустимые параметры _DAYS_ :

- 10950
- 14600
- 16425

Необходимость:

Мы хотим узнать кто из авторов базы данных прожил больше определенного кол-во лет (по смыслу проверки скорее больше определенного кол-во дней). и вывести первые 5 по длине имени. Нам это интересно так как чем больше автор прожил тем скорее всего больше книг он написал, а чем длиннее его имя тем интереснее он жил (если верить разным книгам по толкованию имен), а следовательно тем больше у него было впечатлений/вдохновения для книг.

Пояснение:

Результат запроса выведет три столбца. В первом будет имя писателя, во втором дата рождения, в третьем дата смерти.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON writer ((death_date - birth_date));
```

4 запрос

Выбрать название фанфика и информацию о нем. Вывести первые 5.

```
SELECT f.name, f.preface
```

```
FROM
    ficbook f
WHERE
    preface _SURPRISE_OR_NOT_SURPRISE_
ORDER BY
    f.name
LIMIT 5;
```

Допустимые параметры _SURPRISE_OR_NOT_SURPRISE_ :

- IS NOT NULL
- IS NULL

Необходимость:

Если мы хотим прочесть фанфик мы можем воспользоваться этим запросом. Если мы хотим знать, о чем будем читать, то воспользуемся вторым допустимым значением, если же хотим устроить себе сюрприз, то первым. Очень полезный в повседневной жизни запрос.

Пояснение:

Результат запроса выведет два столбца. В первом будет название фанфика, во втором его описание.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON ficbook (preface);
```

Глава 4 Средние запросы

1 запрос

Выбрать книгу, писателя этой книги и жанр, жанр задан.

```
SELECT o.name AS book, w.name AS writer, g.name AS genre
FROM origin o INNER JOIN genre g
      ON (o.genre_id = g.id) AND (g.name = _GENRE_)
INNER JOIN writer w
      ON (o.writer_id = w.id);
```

Допустимые параметры _GENRE_:

- 'fantasy'
- 'detective'
- 'horror'

Необходимость:

Этот запрос поможет подобрать книгу для чтения по параметру жанр.

Пояснение:

Результат запроса выведет три столбца. В первом будет название книги, во втором его автор, в третьем жанр.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON origin (genre_id);
```

2 запрос

Выбрать книгу, имя автора, язык на котором написана книга (можно задать), дату выпуска не позже заданной.

```
SELECT o.name AS book, w.name AS writer, l.name language, o.release_date
FROM origin o INNER JOIN language l
      ON (o.language_id = l.id) AND (l.name = _LANGUAGE_)
```

```
INNER JOIN writer w
  ON (o.writer_id = w.id)
WHERE o.release_date >= _DATE_;
```

Допустимые параметры _DATE_

- '1900-01-01'
- '1990-01-01'
- '1800-01-01'

Допустимые параметры _LANGUAGE_

- 'english'
- 'russian'
- 'french'

Необходимость:

Нам нужны книги на определенном языке вышедшие не позднее определенного числа.

Пояснения:

По запросу будет выведено четыре столбца. В первом будет название книги, во втором имя автора, в третьем язык на котором написана книга, в четвертом дата выпуска.

Оптимизация:

Были добавлены индексы:

```
CREATE INDEX ON origin (language_id, release_date);
```

3 запрос

Выбрать книгу, писателя, его возраст, жанр.

```
SELECT o.name AS book, w.name AS writer, age(o.release_date, w.birth_date)
AS age, g.name AS genre
  FROM origin o INNER JOIN writer w
    ON (o.writer_id = w.id)
  INNER JOIN genre g
```

```
ON (o.genre_id = g.id) AND (g.name = _GENRE_)
ORDER BY
age(o.release_date, w.birth_date) DESC NULLS LAST;
```

Допустимые параметры _GENRE_:

- 'fantasy'
- 'detective'
- 'horror'

Необходимость:

Мы проводим исследование в каком возрасте писатели больше всего выпускают книги.

Пояснение:

Результат запроса выведет 4 столбца. В первом будет название книги, во втором имя писателя, в третьем возраст в котором он выпустил книгу, в четвертом жанр книги.

Оптимизация:

Были добавлены индексы:

```
CREATE INDEX ON origin (writer_id, genre_id);
```


Глава 5 Сложные запросы

1 запрос

Определить у какого автора было больше всего книг, если максимальное значение книг было у нескольких авторов то вывести их всех.

Выводим имя автора и кол-во написанных им книг.

WITH quantity AS

```
(SELECT w.name, count(o.id) as number_of_books
FROM writer w
INNER JOIN origin o ON (w.id=o.writer_id)
GROUP BY (w.id)),
```

max AS

```
(SELECT max(q.number_of_books) AS maximum
FROM quantity q)
```

SELECT q.name, max.maximum

FROM quantity q

INNER JOIN max ON (max.maximum=number_of_books);

Необходимость:

Этот запрос поможет найти самого печатаемого автора.

Пояснение:

Результат запроса выведет два столбца. В первом будет имя автора, во втором количество написанных им книг.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON origin (writer_id);
```

2 запрос

Вывести список из оригинальных книг и список фанфиков написанных по ним в определенном жанре, перечислив через запятую, сортировать по названию оригинала.

WITH origins AS

```
(SELECT o.id, o.name  
FROM origin o  
JOIN genre g ON (g.id=o.genre_id) ),
```

ficbooks AS

```
(SELECT f.id, f.name  
FROM ficbook f  
JOIN genre g ON (g.id=f.genre_id) AND (g.name=_GENRE_)),
```

relations AS

```
(SELECT origin_id, ficbook_id  
FROM origin_ficbook)
```

```
SELECT ori.name AS books, string_agg(fi.name, ', ' ORDER BY fi.name) AS  
ficbooks
```

```
FROM relations o_f  
INNER JOIN origins ori ON (o_f.origin_id=ori.id)  
INNER JOIN ficbooks fi ON (o_f.ficbook_id=fi.id)  
GROUP BY ori.name;
```

Допустимые параметры _GENRE_:

- 'fantasy'
- 'detective'
- 'horror'

Необходимость:

Мы смотрим по каким классическим фильмам какие фанфики написаны.

Пояснение:

Результат запроса выведет два столбца. В первом будет название оригинала, во втором фанфики написанные по нему.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON ficbook (genre_id);
```

3 запрос

Вывести рейтинг авторов с подсчитанным средним количеством переводов их книг на различные языки, с учетом того что среднее количество подсчитывается только, если у автора есть хотя бы 3 книги у которых известно количество переводов на разные языки. И язык, на котором писал эту книгу сам автор, заданный.

WITH translations AS

```
(SELECT o."id", o.name, o.writer_id, o.translation
      FROM origin o
      JOIN language l ON (o.language_id=l.id) AND
(l.name=_LANGUAGE_)
      WHERE o.translation IS NOT NULL) ,
```

books AS

```
(SELECT tr.id
      FROM translations tr
      GROUP BY tr.id),
```

writers AS

```
(SELECT w.id
      FROM writer w
      LEFT JOIN origin o ON (w.id=o.writer_id)
      WHERE o.id IN (SELECT id FROM books)
      GROUP BY w.id HAVING COUNT(o.id)>2)
```

```
SELECT w.name AS writer, AVG(tr.translation)
      FROM translations tr
      JOIN writer w ON (tr.writer_id=w.id)
      WHERE tr.writer_id IN (SELECT id FROM writers)
      GROUP BY w.id ORDER BY AVG(tr.translation) DESC;
```

Допустимые параметры _LANGUAGE_

- 'english'
- 'russian'
- 'french'

Необходимость:

Мы хотим прочитать книгу на определенном языке (тот который задаем), но не знаем какого автора выбрать. Чтобы не прогадать мы составляем рейтинг авторов со средним количеством переводов их книг на различные языки.

Пояснения:

По запросу будет выведено два столбца. В первом будет имя автора, во втором среднее значение переводов его книг на различные языки.

Оптимизация:

Был добавлен индекс:

```
CREATE INDEX ON origin (language_id);
```