РЕФЕРАТ

Отчет: 19 с., 10 рис., 2 библ., 4 прил.

База данных «МУЗЫКАЛЬНЫЕ ГРУППЫ» предназначена для менеджера музыкальных групп и обеспечивает доступ к информации о названии группы, годе и стране её основания, о её месте в хит-параде, о составе исполнителей (т.ч. их имени, возрасте и амплуа), о песнях группы (в т.ч. названии песни, композиторе и авторе текста), а так же о концертных турах группы (название, концертная программа)

Предусмотрена возможность как внесения изменений, так и получения справок с выводом информации на печать.

База данных функционирует под управлением веб-сервера Apache и сервера приложений Phusion Passenger, ядром базы является встраиваемая реляционная база данных SQLite3.

Ядро базы данных занимает на жестком диске 313 Кб, файл базы данных со 100 записями – 19,4 Кб.

СОДЕРЖАНИЕ

1. ПОСТАНОВКА ЗАДАЧИ……………………………………………….4

1.1 Задание………………………………………………………………..4

1.2 Анализ предметной области………………………………………...4

1.3 Проектирование базы данных с использование метода

ER- диаграмм…………………………………………….…………….....5

1. ОБЩИЕ СВЕДЕНИЯ ПРОГРАММИСТУ……………………………..7
   1. Минимальные системные требования для функционирования

базы данных……………………………………………………………....7

* 1. Состав базы данных «МУЗЫКАЛЬНЫЕ ГРУППЫ»..………….…7

1. ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ………………………………….....8

3.1 Назначение базы данных «КИНОТЕАТРЫ»………………...…….8

3.2 Начало и завершение работы…………………………….................8

3.3 Внесение изменений…………………………………….…………….9

3.4 Просмотр информации………………………………….…………...9

3.5 Рекомендации пользователю…………………………………….....10

Список использованных источников…………………………………............11

ПРИЛОЖЕНИЕ А………………………………………………………...........12

ПРИЛОЖЕНИЕ Б………………………………………………………………13

ПРИЛОЖЕНИЕ В………………………………………………………………18

1. **ПОСТАНОВКА ЗАДАЧИ**
   1. **Задание**

Спроектировать базу данных (БД), построить программу, обеспечивающую взаимодействие с ней в режиме диалога, для менеджера музыкальных групп.

В БД должны храниться сведения о группах: название, год образования, страна, состав исполнителей, положение в последнем хит-параде (может измениться); о репертуаре каждой группы: название песен, композитор, автор текста; данные о последних гастролях группы: название гастрольной программы, место, дата начала и окончания гастролей, цена билета (зависит от места гастролей и положения в хит-параде). Возможно появление новой группы и изменение состава исполнителей. Каждая песня может быть в репертуаре только одной группы.

Предусмотреть возможность получения сведений (в виде справки):

– год образования, страна группы данного названия;

– репертуар наиболее популярной группы;

– автор текста, композитор и дата создания песни с данным названием;

– место и продолжительность гастролей группы данного названия;

– цена билета на концерт указанной группы;

– состав исполнителей группы данного названия, их возраст и амплуа.

Администратор БД может вносить следующие изменения:

– ввод новой группы;

– изменение положения группы в хит-параде;

– удаление информации об исполнителе, покинувшем группу.

Необходимо предусмотреть возможность выдачи справки о лучших трёх группах в хит-параде и отчёта о гастролях групп (название группы, место и сроки гастролей, репертуар с указанием автора песен).

Предусмотреть вывод справок и отчётов на печать по требованию пользователя.

* 1. **Анализ предметной области**

Предметная область – музыкальные группы, их состав, песни и гастрольные программы.

При проектировании базы данных следует учесть то, что в каждой группе может быть несколько участников, группа может иметь несколько песен, несколько гастрольных программ, каждая из которых характеризуется названием и списком концертов, следовательно, надо иметь возможность хранить информацию об участниках, песнях, концертных программах и концертах.

Также следует учесть, что две группы не могут занимать одно место в хит-параде, возраст участников меняется со временем, а цена билета на концерт зависит от позиции группы в хит-параде и страны, в которой концерт проходит.

Для обеспечения автоматического вычисления возраста будем просить пользователя вводить его дату рождения при добавлении.

Т.к. в задании формула подсчёта стоимости билета не указана, возьмём следующую формулу:

При реализации базы данных следует учесть, что выбор из отсортированного списка позволит ускорить работ у пользователя, а также устранит возможность ошибки при вводе соответствующей информации.

* 1. **Проектирование базы данных с использование метода ER-диаграмм**

Проектируем базу данных методом ER-диаграмм.[1]

После проведения анализа предметной области выделим следующие сущности: ГРУППА (Название, Год создания, Страна, Позиция в хит-параде), УЧАСТНИК ГРУППЫ (Имя, Амплуа, Дата рождения), ПЕСНЯ (Название, Автор текста, Композитор), КОНЦЕРТНАЯ ПРОГРАММА (Название), КОНЦЕРТ (Дата, Город, Страна).

Рассмотрим сущности ГРУППА и УЧАСТНИК ГРУППЫ. У группы может быть несколько участников, но каждый участник входит в состав только одной группы, следовательно, имеем отношение «один-ко-многим» (рисунок 1).

ГРУППА

УЧАСТНИК

Имеет

n

1

Рисунок 1

По правилу 4 ER-диаграмма преобразуется в два отношения. Схема первого отношения содержит атрибуты сущности ГРУППА, схема второго – атрибуты сущности УЧАСТНИК и ключ сущности ГРУППА.

Ту же ситуацию имеем для пар сущностей ГРУППА – ПЕСНЯ, ГРУППА – КОНЦЕРТНАЯ ПРОГРАММА и КОНЦЕРТНАЯ ПРОГРАММА - КОНЦЕРТ (рисунки 2, 3 и 4)

ГРУППА

ПЕСНЯ

Имеет

n

1

Рисунок 2

ГРУППА

КОНЦЕРТНАЯ

ПРОГРАММА

Имеет

n

1

Рисунок 3

КОНЦЕРТНАЯ

ПРОГРАММА

КОНЦЕРТ

Имеет

n

1

Рисунок 4

В итоге получаем следующие отношения:

ГРУППЫ (Шифр\_группы\*, Название\_группы, Год\_создания, Страна, Позиция\_в\_хит-параде);

УЧАСТНИКИ (Шифо\_участника\*, Шифр\_группы, Имя, Дата\_рождения, Амплуа);

ПЕСНИ (Шифр\_песни\*, Шифр\_группы, Название, Композитор, Автор\_текста);

КОНЦЕРТНЫЕ ПРОГРАММЫ (Шифр\_программы\*, Шифр\_группы, Название);

КОНЦЕРТЫ (Шифр\_концерта\*, Шифр\_программы, Дата, Город, Страна).

1. **ОБЩИЕ СВЕДЕНИЯ ПРОГРАММИСТУ**
   1. **Минимальные системные требования для функционирования БД**

Для стабильной работы БД рекомендуется следующая минимальная конфигурация:

– процессор x86–совместимый с частотой 1000 МГц;

– ОЗУ 512 Мб;

– свободное место на жестком диске 1,5 Гб;

– POSIX-совместимая ОС.

* 1. **Состав базы данных «МУЗЫКАЛЬНЫЕ ГРУППЫ»**

Разработанная база данных содержит 5 таблиц, 38 запросов, 18 форм.

Схема данных приведена в Приложении А.

База данных состоит из следующих таблиц: GROUPS, MEMBERS, SONGS, TOURS, CONCERTS (рисунок 5).

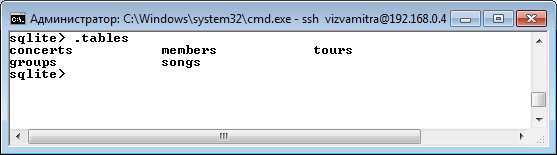


Рисунок 5

Таблица GROUPS содержит информацию о музыкальной группе и включает столбцы: id\*, title, formation\_year, country, top\_position\*.

Таблица MEMBERS содержит информацию об участниках урупп. Она включает столбцы: id\*, name, role, birth\_date, group\_id. Поле group\_id связано с ключевым полем таблицы GROUPS.

Таблица SONGS содержит информацию о песнях и включает столбцы: id\*, title, music\_by, lyrics\_by, group\_id. Поле group\_id связано с ключевым полем таблицы GROUPS.

Таблица TOURS содержит информацию о концертных программах и включает столбцы: id\*, title, group\_id. Поле group\_id связано с ключевым полем таблицы GROUPS.

Таблица CONCERTS содержит информацию о концертах и включает столбцы: id\*, date, city, country, tour\_id. Поле tour\_id связано с ключевым полем таблицы TOURS.

Так же, каждая таблица содержит служебные поля created\_at и updated\_ad.

Структура таблиц представлена в приложении A.

В базе данных предусмотрены запросы на выборку, добавление, обновление и удаление данных. Список запросов представлен в приложении Б.

В базе данных предусмотрены страницы для получения справочной информации, добавления и изменения данных, а так же поиска.

**ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЮ**

* 1. **Назначение базы данных «КИНОТЕАТРЫ»**

База данных «МУЗЫКАЛЬНЫЕ ГРУППЫ» предназначена для менеджеров музыкальных групп.

В БД хранятся сведения о группах: название, дата создания, страна, позиция в хит-параде; об участниках групп: имя, дата рождения, амплуа; о песнях: название, композитор, автор текста; о концертных программах: название, список концертов с указанием даты, города проведения, страны и цены билета. Цена билета определяется позицией группы в хит-параде и страной проведения концерта.

Пользователю доступны следующие сведения и действия:

- список групп;

- информация о группах;

- список участников группы с информацией о каждом из них;

- список песен группы с информацией о каждой из них;

- список гастрольных программ группы;

- список концертов в рамках гастрольной программы с информацией о каждом из них;

- поиск по ключевой фразе

Администратор БД может добавлять новые группы, дополнять списки участников, песен, концертных программ и концертов в рамках них, а так же менять данные записей и удалять их при необходимости.

* 1. **Начало и завершение работы**

После загрузки главной страницы базы данных автоматически открывается список музыкальных групп (рисунок 6).

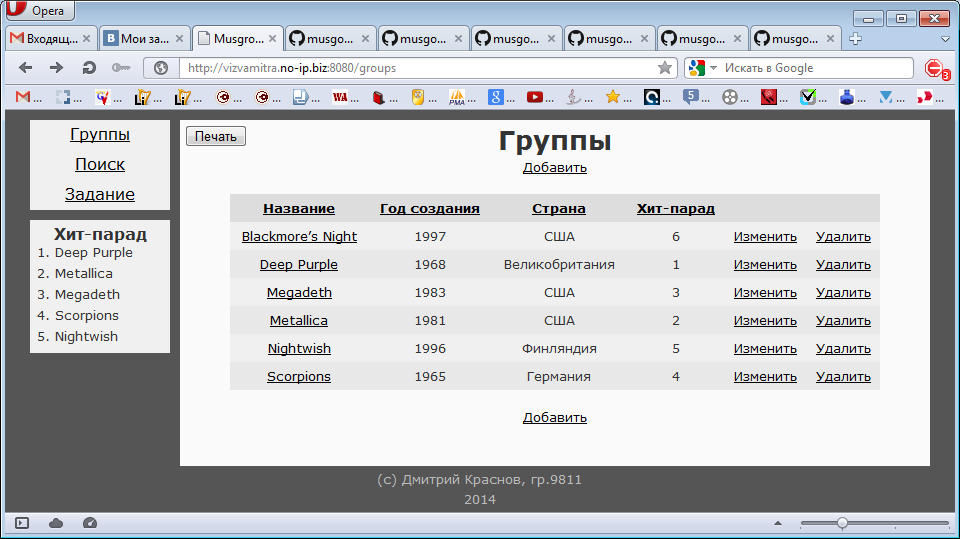


Рисунок 6

Нажатие ссылок главной страницы позволяет переходить на следующие страницы: «Информация о группе» (ссылки в столбце «название» таблицы «Группы»), «Изменение данных о группе» (ссылки «Изменить» в соответствующих рядах таблицы), «Добавление новой группы» (ссылки «Добавить» вверху и внизу таблицы), а так же «Поиск» (кнопка в меню слева).

Кроме того, запись может быть удалена администратором по нажатию кнопки «Удалить» в соответствующем ряду таблицы.

Описанные выше принципы изменения, добавления и удаления данных применимы ко всем информационным страницам базы, кроме поиска.

Для печати любой из страниц базы воспользуйтесь кнопкой «Печать».

Для выхода из базы данных воспользуйтесь кнопкой закрытия окна.

* 1. **Внесение изменений**

Для внесения изменений и добавления новых данных служат ссылки «Добавить» или «Изменить» на информационных страницах.

После их нажатия откроется страница с формой, которая позволяет добавить либо изменить данные о соответствующей сущности. На рисунке 7 представлена форма добавления нового участника.

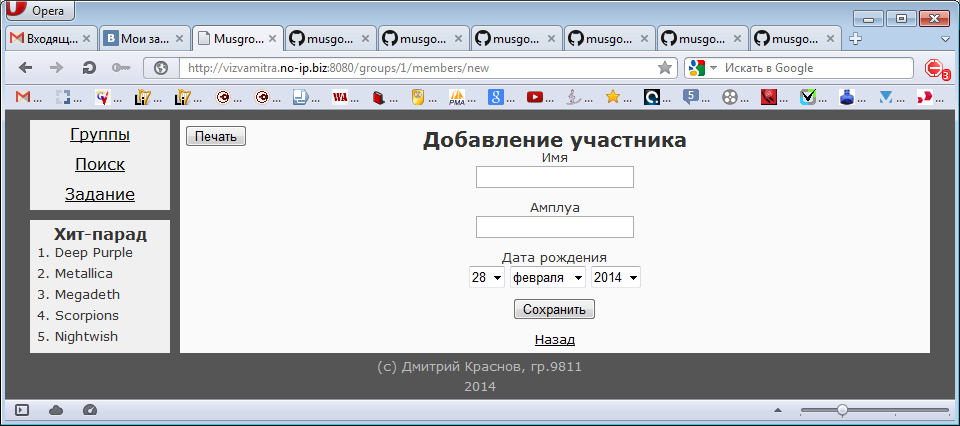


Рисунок 7

После заполнения необходимых полей нажмите кнопку с надписью «Сохранить».

При необходимости вернуться на предыдущую страницу нажмите кнопку «Назад».

* 1. **Просмотр информации**

При переходе с главной страницы на страницу информации о группе будет выведена информация о ней, а так же набор навигационных ссылок на страницы информации по её участникам, песням и гастрольным программам (рисунок 8)

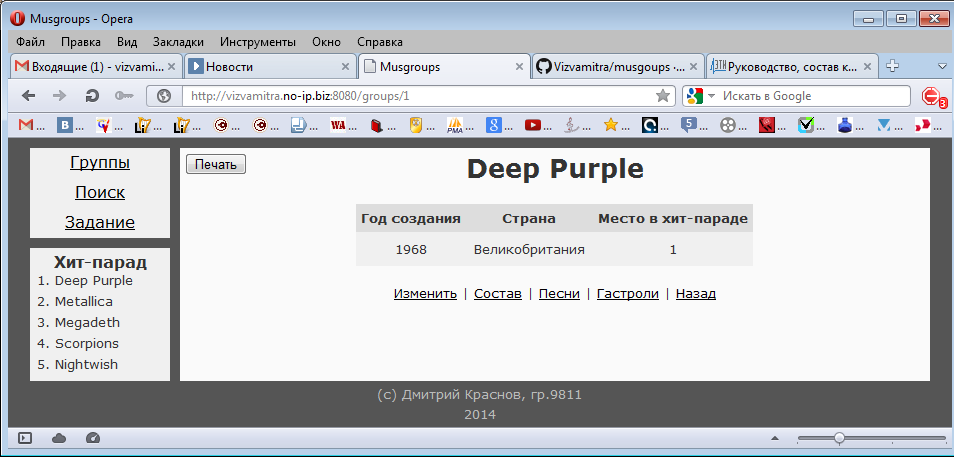


Рисунок 8

Для перехода на нужную информационную страницу необходимо нажать на соответствующую ссылку.

База предусматривает возможность сортировки данных. Для изменения порядка отображения записей на страницах с таблицами групп, участников групп, песен, концертных программ и концертов следует использовать ссылки в шапке таблицы.

Для печати отчёта на основе открытой страницы воспользуйтесь кнопкой «Печать».

Приложение В содержит примеры отчетов.

При необходимости вернуться на главную страницу нажмите кнопку с надписью «Группы» в меню слева.

* 1. **Рекомендации по использованию**

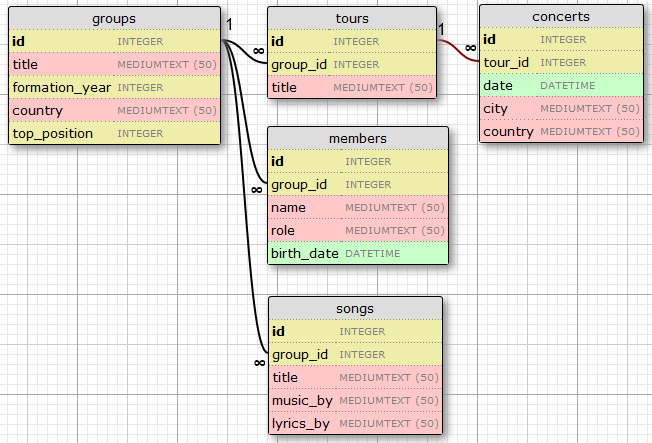
Работа с базой данных осуществляется с помощью ссылок на страницах. Главная страница появляется на экране автоматически после загрузки базы данных.

Наличие в системе принтера дает дополнительные возможности пользователю при работе с БД – печать отчетов.

**Список использованных источников**

1. Джексон, Г. Проектирование реляционных баз данных для использования с микроЭВМ [Текст]: пер. с англ. / Г. Джексон. - М: Мир, 1991. - 252 с., ил.
2. Васильев, В.В. Базы данных [Текст]: учеб. пособие / В.В. Васильев, А.О. Тимофеев. - СПб: СПбГЭТУ, 1998. - 80с.

**ПРИЛОЖЕНИЕ А**



**ПРИЛОЖЕНИЕ Б**

**Группы**

Получение записи по id:

SELECT \*

FROM groups

WHERE id= 15

LIMIT 1;

Получение всех записей групп (поле и порядок сортировки могут изменяться):

SELECT \*

FROM groups

ORDER BY title ASC;

Добавление:

INSERT INTO

groups (title,

formation\_year,

country,

top\_position)

VALUES (

‘Deep Purple’,

1968,

‘Великобритания’,

1);

Изменение данных:

UPDATE groups

SET title = Пантеон',

country = 'Россия',

formation\_year = 2007,

top\_position = 2,

WHERE id=1;

Удаление:

SELECT id FROM tours WHERE group\_id=1;

Для каждой полученной записи:

DELETE from concerts WHERE tour\_id = n;

Далее:

DELETE FROM tours WHERE group\_id=1;

DELETE FROM songs WHERE group\_id=1:

DELETE FROM members WHERE group\_id=1;

DELETE FROM groups WHERE id=1;

Поиск по таблице:

SELECT \* FROM groups

WHERE title LIKE '%запрос%'

OR country LIKE '%запрос%'

OR formation\_year LIKE запрос

**ПРОДОЛЖЕНИЕ** **ПРИЛОЖЕНИЯ Б**

**Участники**

Получение записи по id:

SELECT \*

FROM members

WHERE id= 15

LIMIT 1;

Получение всех записей (поле и порядок сортировки могут изменяться):

SELECT \*

FROM members

WHERE group\_id = 1

ORDER BY name ASC;

Добавление:

INSERT INTO members

(name,

role,

birth\_date,

group\_id)

VALUES (

'Ian Gillan',

'Вокал',

'1945-08-19',

1);

Изменение данных:

UPDATE members

SET name = 'Steven J. Morse',

role = 'Гитара',

birth\_date = '1954-07-24',

WHERE id=15;

Удаление:

DELETE FROM members WHERE id=1;

Поиск по таблице:

SELECT \* FROM members m

INNER JOIN

( SELECT id AS gid, title AS g\_title

FROM groups

) g ON g.gid = m.group\_id

WHERE name LIKE '%запрос%'

OR role LIKE '%запрос%'

**Песни**

Получение записи по id:

SELECT \*

FROM songs

WHERE id= 15

LIMIT 1;

**ПРОДОЛЖЕНИЕ** **ПРИЛОЖЕНИЯ Б**

Получение всех записей (поле и порядок сортировки могут изменяться):

SELECT \*

FROM songs

WHERE group\_id = 1

ORDER BY title ASC;

Добавление:

INSERT INTO songs (

title,

music\_by,

lyrics\_by,

group\_id)

VALUES (

'Highway Star',

'Jan Gillan',

'Jan Gillan ',

1);

Изменение данных:

UPDATE songs

SET title = 'Highway Star',

music\_by = 'Steven J. Morse ',

lyrics\_by = ' Steven J. Morse ',

WHERE id=15;

Удаление:

DELETE FROM songs WHERE id=15;

Поиск по таблице:

SELECT \* FROM songs s

INNER JOIN

( SELECT id AS gid, title AS g\_title

FROM groups

) g ON g.gid = s.group\_id

WHERE s.title LIKE '%запрос%'

OR s.music\_by LIKE '%запрос%'

OR s.lyrics\_by LIKE '%запрос%'

**Концертные туры**

Получение записи по id:

SELECT \*

FROM tours

WHERE id= 15

LIMIT 1;

**ПРОДОЛЖЕНИЕ** **ПРИЛОЖЕНИЯ Б**

Получение всех записей:

SELECT

t.id, t.title, t.group\_id, d.begin\_date, d.end\_date

FROM

tours t

INNER JOIN

( SELECT

MIN(date) AS begin\_date,

MAX(date) AS end\_date,

tour\_id

FROM concerts

GROUP BY tour\_id

) d ON t.id = d.tour\_id

WHERE t.group\_id= 1

SORT BY t.title ASC

Добавление:

INSERT INTO tours (

title,

group\_id)

VALUES (

'Now What?! World Tour',

1);

Изменение данных:

UPDATE songs

SET title = 'Now What?! Europe Tour'

WHERE id=15;

Удаление:

DELETE FROM concerts WHERE tour\_id = 15;

DELETE FROM tours WHERE id=15;

Поиск по таблице:

SELECT title, begin\_date, end\_date, id, gid, g\_title

FROM (

SELECT

t.id, t.title, t.group\_id, d.begin\_date,

d.end\_date, g.gid, g.g\_title

FROM tours t

INNER JOIN (

SELECT

MIN(date) AS begin\_date,

MAX(date) AS end\_date,

tour\_id

FROM concerts

GROUP BY tour\_id

) d ON t.id = d.tour\_id

INNER JOIN (

SELECT title AS g\_title, id AS gid

FROM groups

) g ON g.gid = t.group\_id

)

WHERE title LIKE '%запрос%'

**ПРОДОЛЖЕНИЕ** **ПРИЛОЖЕНИЯ Б**

**Концерты**

Получение записи по id:

SELECT \*

FROM tours

WHERE id= 15

LIMIT 1;

Получение всех записей по id концертной программы:

SELECT \*

FROM concerts

WHERE tour\_id = 1

ORDER BY date ASC

Добавление записи:

INSERT INTO concerts (

country,

city,

date,

tour\_id)

VALUES (

'Дубай',

'ОАЭ',

'21.01.2013',

1)

Изменение данных:

UPDATE concerts

SET country = 'Россия',

city = 'Санкт-Петербург',

date = '08.11.2013',

WHERE id=15;

Удаление:

SELECT COUNT(\*) FROM concerts WHERE tour\_id = 1;

Если концертов в туре больше 1, разрешено удаление:

DELETE FROM concerts WHERE id = 15;

Поиск по таблице:

SELECT id, date, city, country, t\_title, tid, g\_title, gid

FROM (

SELECT \*

FROM concerts c

INNER JOIN (

SELECT id AS tid, group\_id, title AS t\_title

FROM tours

) t ON t.tid = c.tour\_id

INNER JOIN (

SELECT title AS g\_title, id AS gid

FROM groups

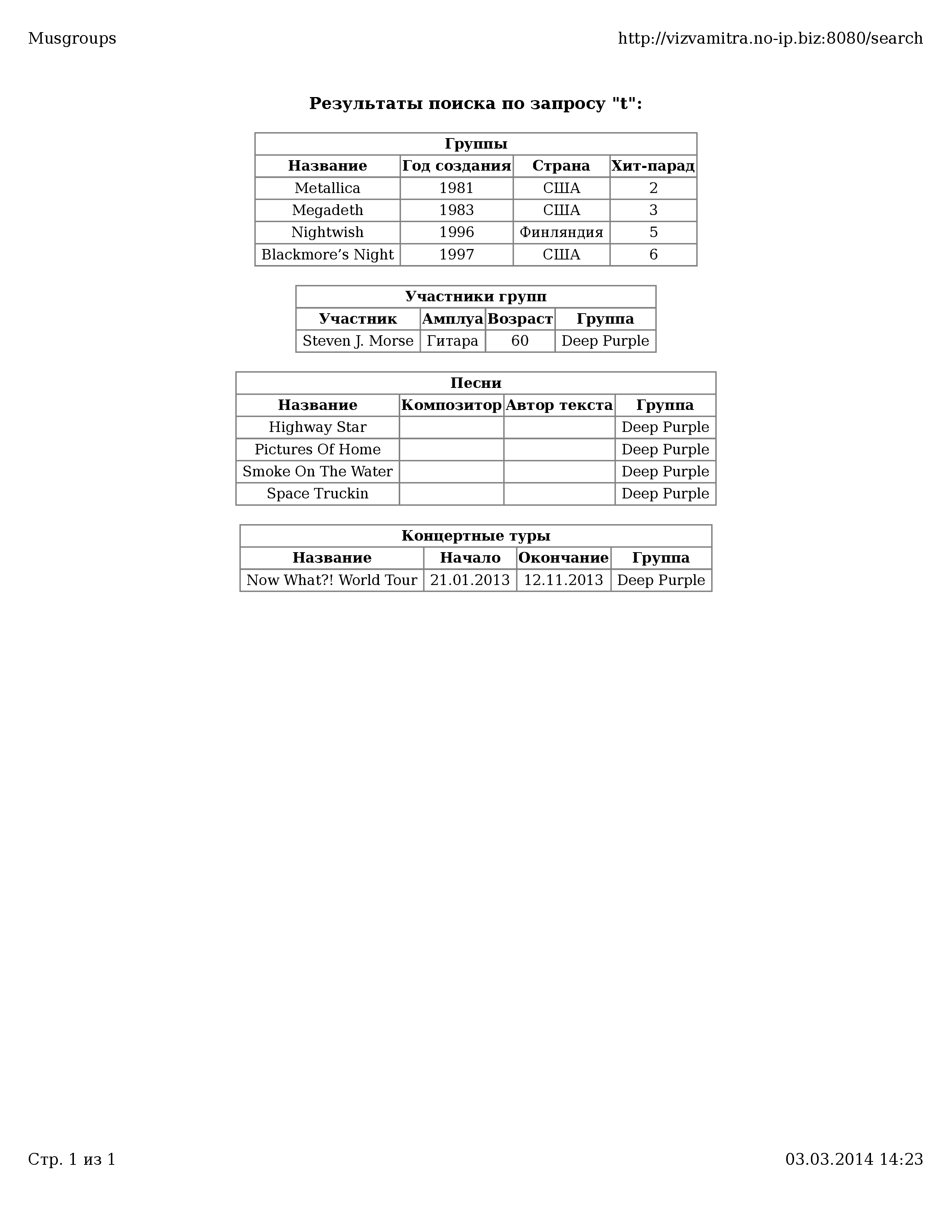
) g ON t.group\_id = g.gid )

WHERE city LIKE '%запрос%'

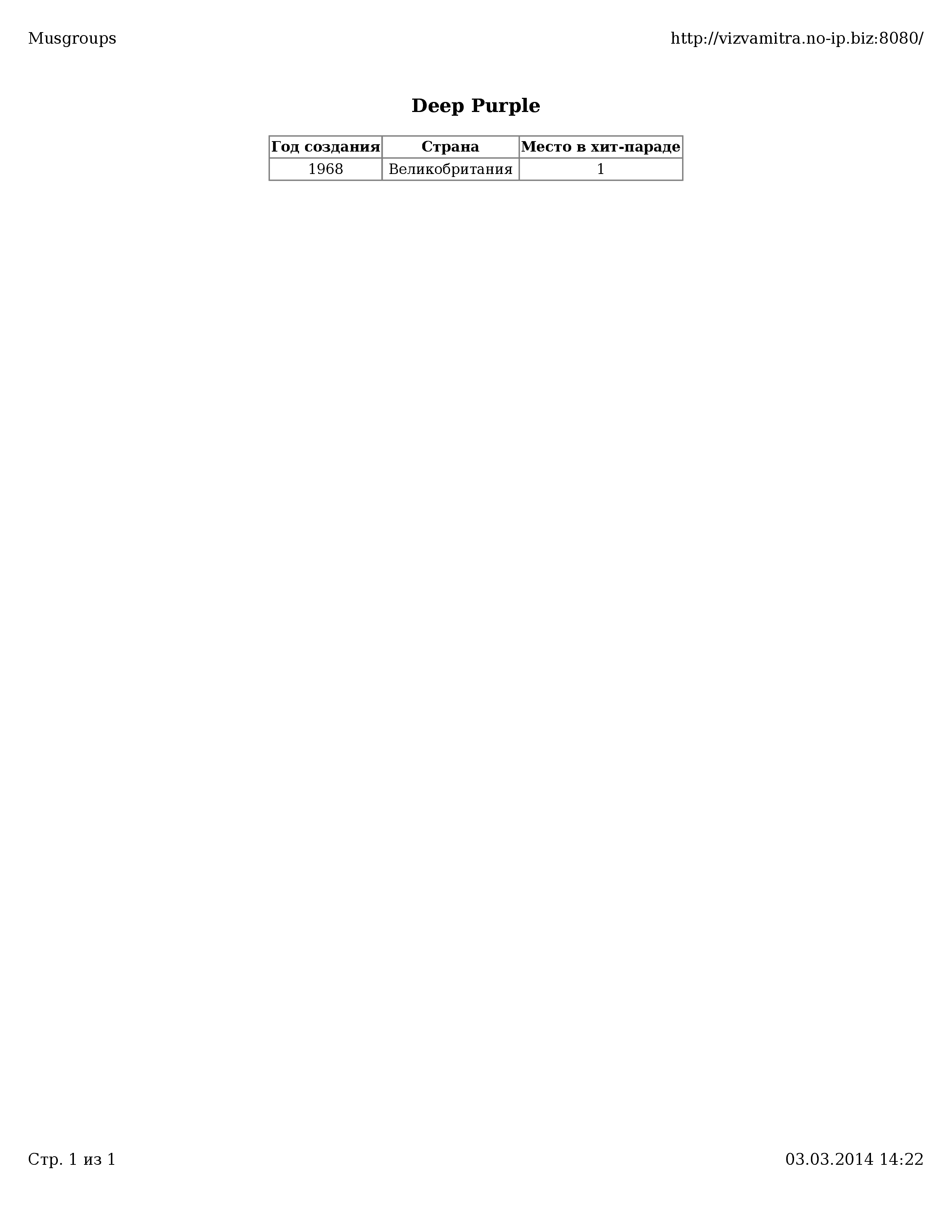
OR country LIKE '%запрос%'

OR date LIKE '%запрос%'

**ПРИЛОЖЕНИЕ В**



**ПРОДОЛЖЕНИЕ** **ПРИЛОЖЕНИЯ В**



**ПРОДОЛЖЕНИЕ** **ПРИЛОЖЕНИЯ В**

