结课作业要求

数据库系统的分析和设计及实现

1. 系统需求分析

2. 设计系统ER图

3. 数据表

4. 系统视图

5. 系统存储过程

6. 触发器

**1. 系统需求分析**

**系统目标需求：**

半糖歌单交友匹配管理系统的数据库开发是基于SQL Server 2008开发的。

其系统的主要需求是完成对用户根据歌单相似度的匹配交友的相关管理。系统主要面向的对象是看中音乐口味选择好友的用户。系统具体的需求分析如下：

* 用户第一次登陆前，必须先进行注册。设置登录时的用户名和密码并填写保存个人相关信息。
* 用户从登陆界面进入系统，在登录页面输入用户名、密码和验证码，信息验证成功之后进入系统首页。如果输入信息不匹配，系统自动给出登陆错误的提示信息。
* 用户可以进行修改密码等修改个人信息的操作。
* 用户可以贡献（导入）自己在网易云等音乐软件创建的歌单。系统利用数据库中已有的歌单进行两两匹配，得到各个歌单之间的匹配度。
* 用户可以查看系统为用户个性化推荐的好友及歌曲。
* 用户可以添加系统推荐的好友，歌曲。
* 用户可以查询个人信息，好友信息。
* 用户可以对系统匹配结果进行评价。以便后期改进算法。
* 管理员可以通过本系统查看用户个人信息。
* 管理员可以通过本系统查看歌单信息。
* 管理员可以通过本系统冻结或解冻用户账号。
* 管理员可以查找歌单匹配度高的（即更大众化的歌单），发布在网站首页。
* 管理员可以查看用户的评价打分以及建议，用户体验等。

**业务流程分析：（可以不要）**

浏览者必须先进行注册，才能够以用户的身份匹配歌单进行交友。浏览者注册的信息提交到网站后台，由网站管理员进行管理。网站为用户提供推荐好友，推荐歌曲的功能，并在后台记录用户歌单间的匹配度，从而形成匹配信息。在系统后台，管理员除了管理用户信息，歌单信息外，还需将优质歌单发布在首页，同时也要及时查看用户的评价调查。主要业务流程如图所示。

**业务流程图**

用户信息

修改资料

匹配信息

满意度情况

注册

贡献歌单

歌单信息

系统进行匹配

推荐好友及歌曲列表

冻结/解冻

发布

发布优质歌单

评价系统

有效用户名

登录

添加好友或歌曲

图1-1系统业务流程图

**数据流程图**

歌单信息

评价

基本信息

贡献歌单

注册资料

D3

推荐好友及歌曲列表

D4

满意度信息

D5

优质歌单列表

D2

歌单信息表

D1

用户资料

P2

用户信息管理

用户

用户

P1

用户

注册

P3

歌单分析

浏览者

P4

匹配信息管理

管理者

P6

歌单管理

P5

评价管理

用户

图1-2系统数据流程图

**数据字典**

代码设计

系统中对用户id，歌曲id，歌单id,歌手id进行了代码设计，为实现系统自动编码，编码大多采用层次码加顺序码

用户id,层次码表示城市（城市名缩写）、注册年份后两位、注册日期、，顺序码则表示用户注册形成的先后次序。考虑到网站的规模和订单的数量，编号的顺序码采用三位。代码模型如图1-3所示。



图1-3用户编号的代码模型

2位+2位+4位+可变3位=可变11位

歌曲ID（Songid）顺序码 按照歌曲录入曲库的次序 可变8位

歌单ID（playlistid）层次码+顺序码 NNNN NNNNN

年份后两位和月份+创建歌单的次序 4位+可变5位=可变9位

歌手ID（artistid）顺序码 按照歌手录入曲库的次序 可变5位

数据字典

用户数据项描述

数据项编号：I02-01

数据项名称：用户编号

别名：userid

简述：用户注册时生成的代码

类型及宽度：可变字符型，11位

后9位取值范围：160101001~991231999

数据结构描述

歌单标识数据结构

数据结构编号：D03-02

数据结构名称：歌单标识

简述：通过几个特征属性标识一个歌单

组成：歌单名+用户评分+歌单类型标签+创建时间+所含歌曲

相关数据流：优质歌单列表、用户资料表、歌单匹配表

相关处理逻辑：P3、P4、P5、P6

数据流描述

用户-歌单信息流描述

数据流编号：用户信息流

数据流来源：用户资料列表

数据流去向：歌单列表

数据流量：30个/天

高峰数据流量：45个/天

数据存储描述

优质歌单存储描述

数据存储编号：D5

数据存储名称：优质歌单列表

简述：该存储列表里的歌单是经过用户投票及用户的各个匹配得出的匹配度比较高比较热门的一些歌曲

数据存储组成：歌单号+用户标签+创建日期+所含歌曲

关键字：歌单号、用户标签

处理逻辑描述

通过歌单匹配推荐好友处理逻辑描述

数据逻辑编号：P4

数据逻辑名称：歌单匹配推荐好友

简述：通过歌单的相似程度来推荐好友

处理：通过比较歌单里的歌名及歌曲风格来为该用户推荐潜在好友

输入：用户资料、用户歌单

输出：匹配成功的好友

处理频率：每人每天一次

**功能结构图**

半糖歌单管理系统前台

首页歌单

歌单分析

推荐好友

用户管理

好友管理

查询歌单

看详细信息

查看歌单

歌单匹配度

推荐好友信息

用户登录

用户注册

修改资料

添加好友

评价管理

后台登录

用户管理

歌单管理

退出后台

半糖歌单管理系统后台

用户信息

冻结解冻

查看评价信息

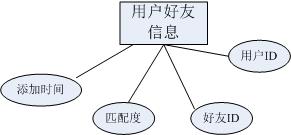
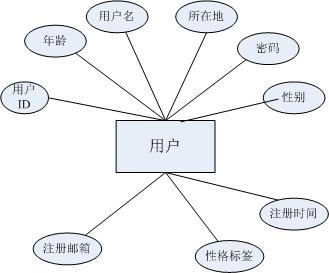
查看歌单

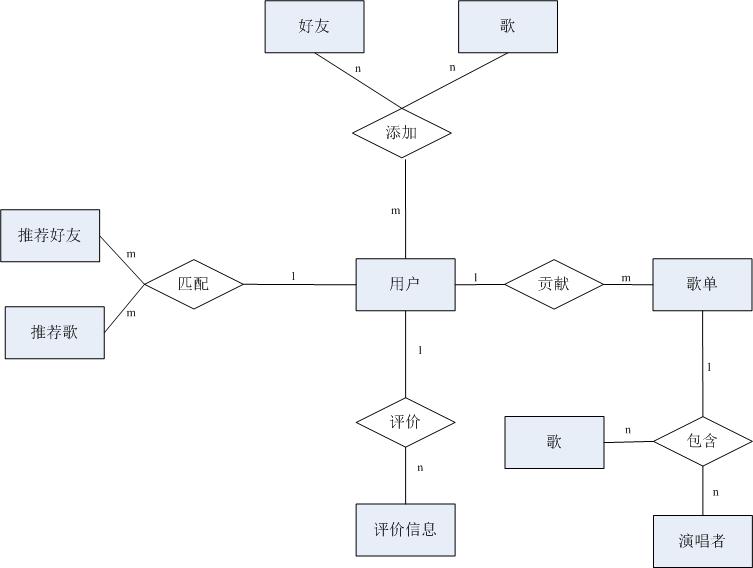
发布优质歌单

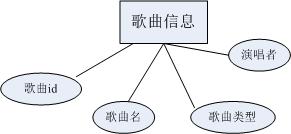
删除歌单

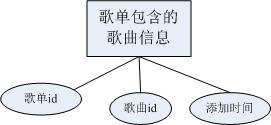
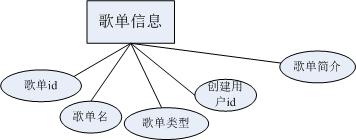
**2. 设计系统ER图**

参照酒店管理的例子做！

****

****

****

****

**3. 数据表设计**

**数据库**

根据上面的系统需求分析，我们在SQL Server 2008中建立一个名为“半糖歌单”的数据库存放本系统所需的数据表，创建数据库的SQL语句如下:

CREATE DATABASE [半糖歌单] ON PRIMARY

( NAME = N'半糖歌单', FILENAME = N'C:\Users\Administrator\Desktop\数据库\半糖歌单.mdf' , SIZE = 3072KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG ON

( NAME = N'半糖歌单\_log', FILENAME = N'C:\Users\Administrator\Desktop\数据库\半糖歌单\_log.ldf' , SIZE = 1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

**数据表**

为满足本系统功能需要，根据设计好的系统全局E R模型图，创建

“半糖歌单”的数据库中的数据表:用户表、好友列表、好友推荐表、。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。。表。创建表的同时实现数据的完整性。

1用户表

用来记录注册了本系统的用户的各种信息，用用户id作为该表的主键，该表的字段结构如下表所述：



创建表的语句：

CREATE TABLE [dbo].[用户](

[用户ID] [varchar](20) NOT NULL,

[用户名] [varchar](20) NOT NULL,

[密码] [varchar](20) NOT NULL,

[性别] [char](2) NOT NULL,

[年龄] [int] NOT NULL,

[所在地] [varchar](10) NOT NULL,

[注册时间] [datetime] NOT NULL,

[注册邮箱] [varchar](40) NOT NULL,

[性格标签] [varchar](10) NOT NULL,

CONSTRAINT [PK\_用户] PRIMARY KEY CLUSTERED

(

[用户ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

2好友列表

用来记录使用本系统的普通用户添加好友的信息，用用户id和好友id作为主键，该表的字段结构如下表所述：



创建表的语句：

CREATE TABLE [dbo].[好友列表](

[用户id] [varchar](20) NOT NULL,

[好友id] [varchar](20) NOT NULL,

[添加时间] [datetime] NULL,

[歌单匹配度] [float] NOT NULL,

CONSTRAINT [PK\_好友列表] PRIMARY KEY CLUSTERED

(

[用户id] ASC,

[好友id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

3推荐好友列表

用来记录使用本系统的普通用户贡献歌单后，系统对歌单进行分析，与其他用户歌单进行匹配的信息。用用户id和好友id作为主键，该表的字段结构如下表所述：



创建表的语句：

CREATE TABLE [dbo].[好友推荐](

[用户id] [varchar](20) NOT NULL,

[推荐好友id] [varchar](20) NOT NULL,

[匹配度] [float] NOT NULL,

CONSTRAINT [PK\_好友推荐] PRIMARY KEY CLUSTERED

(

[用户id] ASC,

[推荐好友id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

4歌单列表

用来记录使用本系统的普通用户贡献的歌单信息，用歌单id作为主键，该表的字段结构如下表所述：



创建表的语句：

CREATE TABLE [dbo].[歌单列表](

[歌单id] [varchar](9) NOT NULL,

[歌单名] [varchar](50) NOT NULL,

[歌单类型] [varchar](20) NULL,

[创建用户id] [varchar](20) NOT NULL,

[歌单简介] [varchar](max) NULL,

CONSTRAINT [PK\_歌单列表] PRIMARY KEY CLUSTERED

(

[歌单id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

5歌曲列表

用来记录使用本系统数据库中包含的歌曲，用歌曲id作为主键，该表的字段结构如下表所述：



创建表的语句：

CREATE TABLE [dbo].[歌曲列表](

[歌曲id] [varchar](8) NOT NULL,

[歌曲名] [varchar](50) NOT NULL,

[歌曲类型] [varchar](10) NULL,

[演唱者] [varchar](50) NOT NULL,

CONSTRAINT [PK\_歌曲列表] PRIMARY KEY CLUSTERED

(

[歌曲id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

6歌单歌曲信息



CREATE TABLE [dbo].[歌单歌曲信息](

[歌单id] [varchar](9) NOT NULL,

[歌曲id] [varchar](8) NOT NULL,

[添加时间] [datetime] NULL,

CONSTRAINT [PK\_歌单歌曲信息] PRIMARY KEY CLUSTERED

(

[歌单id] ASC,

[歌曲id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

**4. 系统视图**

1 wh161220002用户好友推荐视图

它由用户表的某些字段和好友推荐表连接而成

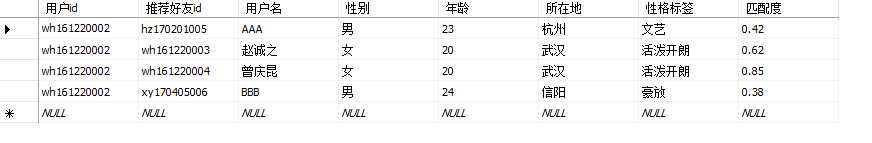
创建该视图的SQL语句如下

SELECT dbo.好友推荐.用户id, dbo.好友推荐.推荐好友id, dbo.用户.用户名, dbo.用户.性别, dbo.用户.年龄, dbo.用户.所在地, dbo.用户.性格标签, dbo.好友推荐.匹配度

FROM dbo.好友推荐 INNER JOIN

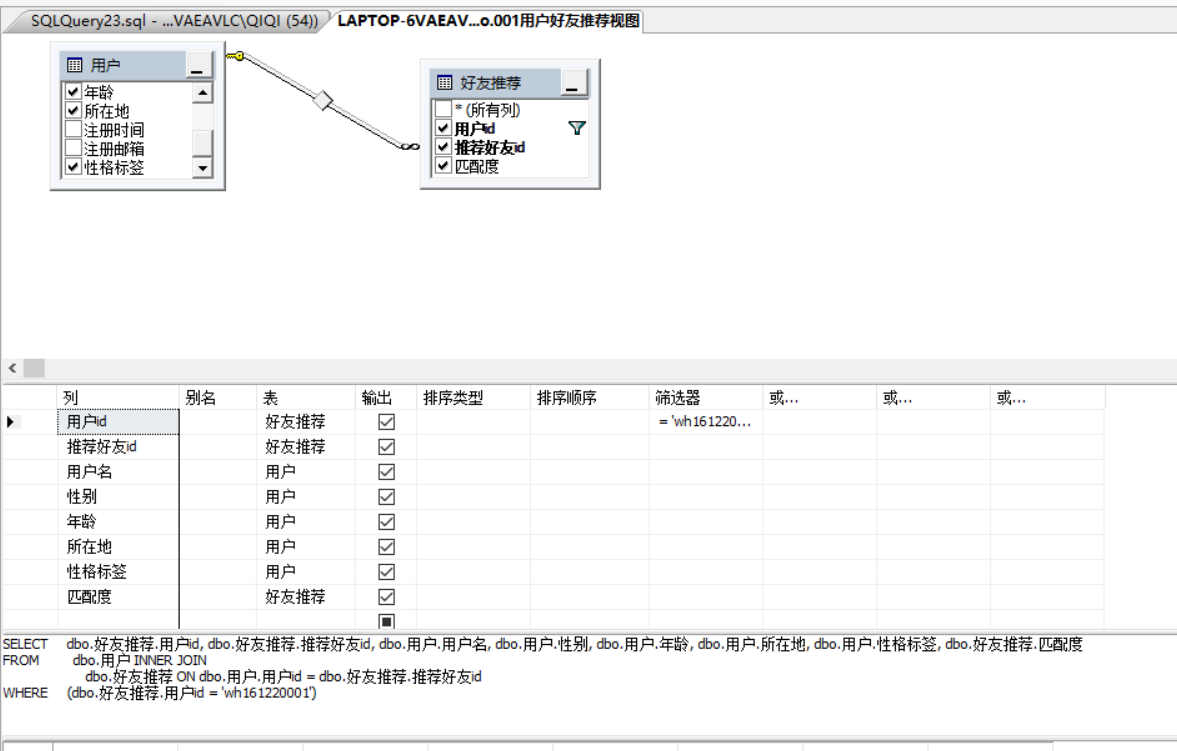
dbo.用户 ON dbo.好友推荐.推荐好友id = dbo.用户.用户id

WHERE (dbo.好友推荐.用户id = 'wh161220002')



2

001用户好友推荐视图：

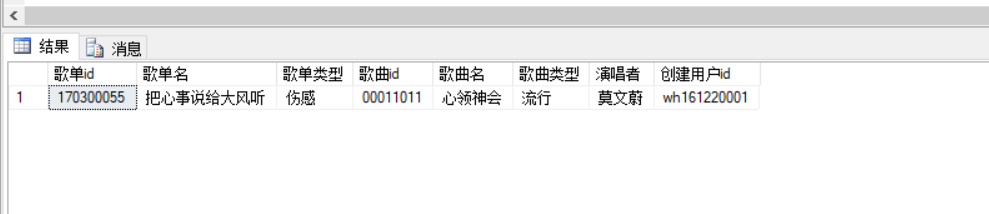




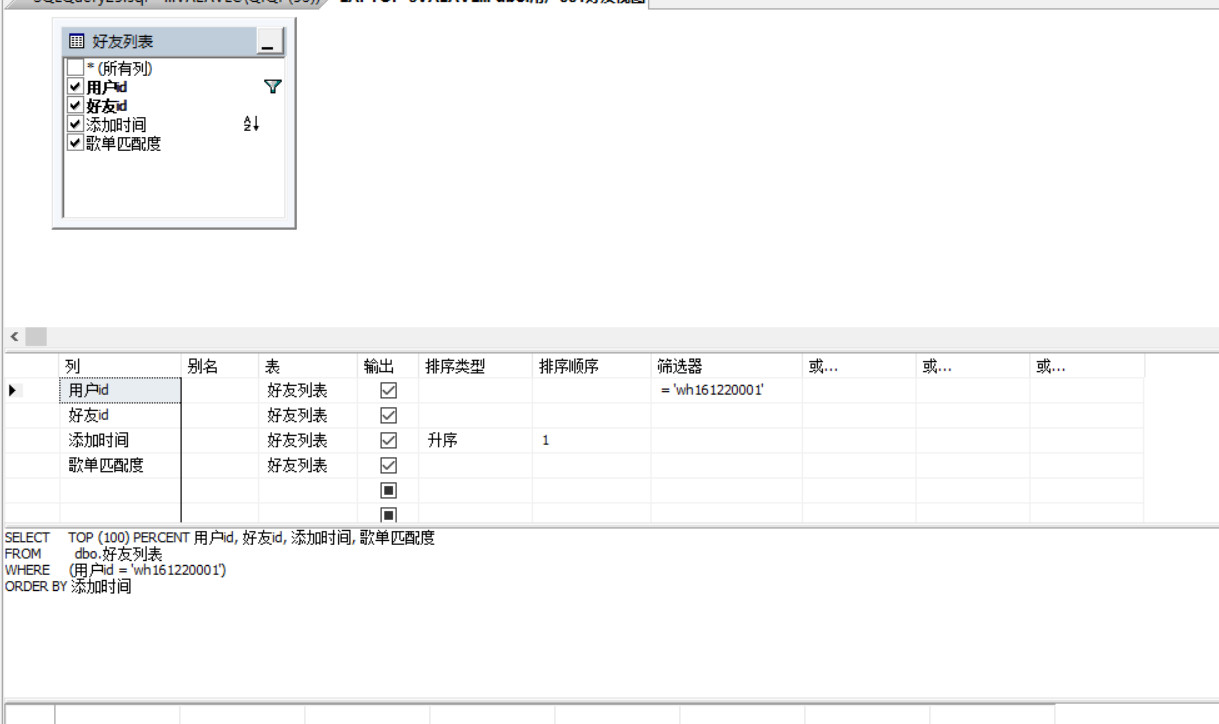
3

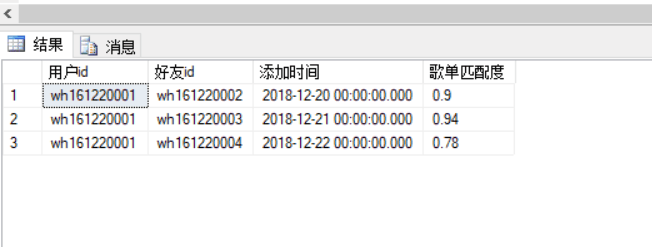
歌单055中的歌曲信息类型：





4用户001的好友视图：





**5. 系统存储过程**

1该存储过程用于普通用户查询系统推荐的好友信息，SQL语句如下

ALTER PROCEDURE [dbo].[获取推荐好友]

@number varchar(20)

AS

SELECT dbo.好友推荐.用户id, dbo.好友推荐.推荐好友id, dbo.用户.用户名, dbo.用户.性别, dbo.用户.年龄, dbo.用户.所在地, dbo.用户.性格标签, dbo.好友推荐.匹配度

FROM dbo.好友推荐 INNER JOIN

dbo.用户 ON dbo.好友推荐.推荐好友id = dbo.用户.用户id

WHERE (dbo.好友推荐.用户id =@number)

比如用户wh161220002查询系统给自己推荐的好友

****

****



**6. 触发器**

1为用户表创建一个简单的DML触发器，在修改个人信息的时候，会自动显示提示信息

USE [半糖歌单]

GO

Create TRIGGER [dbo].[reminder]

ON [dbo].[用户]

FOR

UPDATE

AS print '你在修改个人信息'

