**Creat database DoubanDB**

**On(**

**Name=douban\_data,**

**Filename=’C:\DoubanDataBase.mdf’,**

**Size=30,**

**Maxsize=800,**

**Filegrowth=10)**

**Log on(**

**Name=douban\_log,**

**Filename=’C:\DoubanDataBaselog.ldf’,**

**Size=20,**

**Maxsize=800,**

**Filegrowth=10)**

**实体转换为关系：**

用户（UserId，UserName，password，手机号，邮箱，微博账号，微信账号，昵称，性别）

Creat table 用户

( UserId　int　identity(1,1) primary key,

UserName nvarchar(50) not null unique,

password nvarchar(30) not null,

手机号 char(11) null,

邮箱 nvarchar(20) null,

微博账号 nvarchar(20) null,

微信账号 nvarchar(20) null,

昵称 nvarchar(20) not null,

性别 nvarchar(10) not null check (性别=’男’ or 性别=’女’) default ‘男’ ,

)

豆列（豆列ID，豆列名称，UserId）

Create table 豆列

(

豆列ID int primary key identity(1,1)

豆列名称 nvarchar(50) not null,

用户ID int foreign key references 用户(用户ID)

)

评论（评论ID，UserId，电影ID，星级，评论内容，设备，评论时间，观看状态，有用个数）

Create table 评论

(

评论ID int identity(1,1) Primary key,

UserId int Foreign key references 用户(UserId),

电影ID int Foreign key references 电影（电影ID）,

星级 int check (星级>=0 and 星级<=5),

评论内容 ntext,

设备 nchar(10),

评论时间 datetime not null,

观看状态 nvarchar(10) check（观看状态=‘想看’ or 观看状态=‘看过’） default ‘想看’,

有用个数 int default 0,

）

电影（电影ID，电影介绍，官方网站，IMDb）

Create table 电影

(

电影ID int identity(1,1),

电影介绍 ntext,

官方网站 ntext,

IMDb nvarchar(20),

Primary key (电影ID),

)

讨论（讨论ID，UserId，电影ID，讨论标题，发起时间）

Create table 讨论

(

讨论ID int identity(1,1),

UserId int,

电影ID int,

讨论标题 ntext not null,

发起时间 datetime not null,

Primary key (讨论ID),

Foreign key (UserId) references 用户(UserId),

Foreign key (电影ID) references 电影(电影ID)

)

回应（回应ID，UserId，讨论ID，回应内容，回应时间）

Create table 回应

(

回应ID int identity(1,1)

UserId,

讨论ID,

回应内容 ntext not null,

回应时间 datetime not null,

Primary key (回应ID),

Foreign key (UserId) references 用户(UserId),

Foreign key (讨论ID) references 讨论(讨论ID)

)

问题（问题ID，UserId，电影ID，问题标题，发起时间）

Create table 问题

(

问题ID int identity(1,1),

UserId int,

电影ID int,

问题标题 ntext not null,

发起时间 datetime not null,

Primary key (问题ID)，

Foreign key (UserId) references 用户(UserId),

Foreign key (电影ID) references 电影(电影ID)

)

回答（回答ID，UserId，问题ID，回答内容，回答时间）

Create table 回答（

回答ID int identity（1,1）primary key，

UserId int foreign key references 用户(UserId) ,

问题ID int foreign key references 问题(问题ID) ,

回答内容 ntext not null

）

人员（人员ID，出生日期，性别，星座，国籍，人员名称）

Cteate table 人员(

人员ID int identity（1,1）primary key，

出生日期 date CHECK (date>'1949-1-1' and date<'2100-12-31'),

性别 nvarchar（20） check （性别=’男’or性别=’女’），

星座 nvarchar（20），check（星座=’白羊座 ’or星座=’金牛座 ’or星座=’双子座 ’or星座=’巨蟹座 ’or星座=’狮子座 ’or星座=’处女座 ’or星座=’天秤座 ’or星座=’天蝎座 ’or星座=’射手座 ’or星座=’魔蝎座 ’or星座=’水瓶座 ’or星座=’双鱼座 ’）

国籍 nvarchar（20），

人员名称 nvarchar（50）

)

~~会议奖项（会议名称）~~

~~奖项（奖项名称，奖项得主）~~

~~时长（电影时长，地区版本）~~

~~电影名称（电影名称）~~

~~国家（国家名称）~~

~~上映时间（上映时间，地区版本）~~

~~电影类型（类型名）~~

~~电影语言（语言名）~~

~~别名（名称）~~

**联系转换为关系：**

收藏（豆列ID，评论ID）

Create table 收藏（

豆列ID int foreign key references 豆列(豆列ID)，

评论ID int foreign key references 评论(评论ID),

Primary key (豆列ID,评论ID),

）

关注（关注者UserId，被关注者UserId）

Create table 关注（

关注者UserId int foreign key references 用户(UserId)，

被关注者UserId int foreign key references 用户(UserId)，

Primary key (关注着UserId,被关注着UserId)

）

同问（UserId，问题ID）

Create table 同问（

UserId int foreign key references 用户(UserId)，

问题ID int foreign key references 问题(问题ID)，

Primary key (UserId,问题ID)

）

制作（电影ID，人员ID，人员类型）

Create table 制作

(

电影ID int FOREIGN KEY REFERENCES 电影(电影ID),

人员ID int FOREIGN KEY REFERENCES 人员(人员ID),

人员类型 nvarchar（15），

PRIMARY KEY (电影ID，人员ID，人员类型)

)

使用（人员ID，名称）

Create table 使用

（

人员ID int FOREIGN KEY REFERENCES 人员（人员ID）

名称 nvarchar（15）

PRIMARY KEY （人员ID，名称）

）

包括（电影ID，语言名）

Create table 包括

（

电影ID int FOREIGN KEY REFERENCES 电影（电影ID）

语言名 nvarchar（15）

PRIMARY KEY（电影ID，语言名）

）

属于（电影ID，类型名）

Create table 属于

（

电影ID int FOREIGN KEY REFERENCES 电影（电影ID）

类型名 nvarchar（15）

PRIMARY KEY （电影ID，类型名）

）

上映（电影ID，上映时间，地区版本）

Create table 上映

（

电影ID int FOREIGN KEY REFERENCES 电影（电影ID）

上映时间 date

地区版本 nvarchar（15）

PRIMARY KEY （电影ID，上映时间，地区版本）

）

属于1（电影ID，国家名称）

Create table 属于1

(

电影ID int foreign key references 电影(电影ID),

国家名称 nvarchar(50),

Primary key(电影ID，国家名称)

)

包含（电影名称，电影ID）

Create table 包含

(

电影ID int foreign key references 电影(电影ID),

电影名称 nvarchar(50) primary key ,

)

拥有（电影ID，电影时长，地区版本）

Create table 拥有

(

电影ID int foreign key references 电影(电影ID),

电影时长 nvarchar(50) ,

地区版本 nvarchar(50),

Primary key(电影ID,电影时长,地区版本)

)

颁发（电影ID，会议名称，奖项得主，奖项名称）

Create table 颁发

(

电影ID int foreign key references 电影(电影ID),

会议名称 nvarchar(50) ,

奖项得主 nvarchar(50),

奖项名称 nvarchar(50),

Primary key(电影ID,会议名称,奖项得主,奖项名称)

)

标注（评论ID，评价标签）

Create table 标注

(

评论ID int foreign key references 评论(评论ID),

评价标签 nvarchar(50),

Primary key(评论ID,评论标签)

)

~~拥有（UserName，豆列ID）~~

~~发布（评论ID，UserName）~~

~~提出（问题ID，UserName）~~

~~关于（问题ID，电影ID）~~

~~上传（回答ID，UserName）~~

~~包含（回答ID，问题ID）~~

~~发表（讨论ID，UserName）~~

~~包含1（回应ID，讨论ID）~~

~~发布（回应ID，UserName）~~

~~关于1（讨论ID，电影ID）~~

~~评价（评论ID，电影ID）~~

~~获得（电影ID，会议名称）~~

Creat database DoubanDB

On(

Name=douban\_data,

Filename=’C:\DoubanDataBase.mdf’,

Size=30,

Maxsize=800,

Filegrowth=10)

Log on(

Name=douban\_log,

Filename=’C:\DoubanDataBaselog.ldf’,

Size=20,

Maxsize=800,

Filegrowth=10)

实体转换为关系：

user（userID，userName，password，call，email，microBlog，weChat，nickName，sex）

Creat table user

( userID　int　identity(1,1) primary key,

userName nvarchar(50) not null unique,

password nvarchar(30) not null,

call char(11) null,

email nvarchar(20) null,

microBlog nvarchar(20) null,

weChat nvarchar(20) null,

nickName nvarchar(20) not null,

sex nvarchar(10) not null check (sex=’男’ or sex=’女’) default ‘男’ ,

)

beanList（beanListID，beanListName，userID）

Create table beanList

(

beanListID int primary key identity(1,1)

beanListName nvarchar(50) not null,

userID int foreign key references user(userID)

)

comment（commentID，userID，movieID，starLevel，commentContent，device，commentTime，viewStatus，usefulNumber）

Create table comment

(

commentID int identity(1,1) Primary key,

userID int Foreign key references user(userID),

movieID int Foreign key references movie（movieID）,

starLevel int check (starLevel>=0 and starLevel<=5),

commentContent ntext,

device nchar(10),

commentTime datetime not null,

viewStatus nvarchar(10) check（viewStatus=‘想看’ or viewStatus=‘看过’） default ‘想看’,

usefulNumber int default 0,

）

movie（movieID，movieIntroduce，officialWebsite，IMDb）

Create table movie

(

movieID int identity(1,1),

movieIntroduce ntext,

officialWebsite ntext,

IMDb nvarchar(20),

Primary key (movieID),

)

discuss（discussID，userID，movieID，discussTitle，beginTime）

Create table discuss

(

discussID int identity(1,1),

userID int,

movieID int,

discussTitle ntext not null,

beginTime datetime not null,

Primary key (discussID),

Foreign key (userID) references user(userID),

Foreign key (movieID) references movie(movieID)

)

response（responseID，userID，discussID，responseContent，responseTime）

Create table response

(

responseID int identity(1,1)

userID,

discussID,

responseContent ntext not null,

responseTime datetime not null,

Primary key (responseID),

Foreign key (userID) references user(userID),

Foreign key (discussID) references discuss(discussID)

)

question（questionID，userID，movieID，questionTitle，beginTime）

Create table question

(

questionID int identity(1,1),

userID int,

movieID int,

questionTitle ntext not null,

beginTime datetime not null,

Primary key (questionID)，

Foreign key (userID) references user(userID),

Foreign key (movieID) references movie(movieID)

)

reply（replyID，userID，questionID，replyContent，replyTime）

Create table reply（

replyID int identity（1,1）primary key，

userID int foreign key references user(userID) ,

questionID int foreign key references question(questionID) ,

replyContent ntext not null

）

personnel（personnelID，birthday，sex，constellation，nationality，personnelName）

Cteate table personnel(

personnelID int identity（1,1）primary key，

birthday date CHECK (date>'1949-1-1' and date<'2100-12-31'),

sex nvarchar（20） check （sex=’男’orsex=’女’），

constellation nvarchar（20），check（constellation=’白羊座 ’orconstellation=’金牛座 ’orconstellation=’双子座 ’orconstellation=’巨蟹座 ’orconstellation=’狮子座 ’orconstellation=’处女座 ’orconstellation=’天秤座 ’orconstellation=’天蝎座 ’orconstellation=’射手座 ’orconstellation=’魔蝎座 ’orconstellation=’水瓶座 ’orconstellation=’双鱼座 ’）

nationality nvarchar（20），

personnelName nvarchar（50）

)

联系转换为关系：

collection（beanListID，commentID）

Create table collection（

beanListID int foreign key references beanList(beanListID)，

commentID int foreign key references comment(commentID),

Primary key (beanListID,commentID),

）

watch（watchUserID，watchedUserID）

Create table watch（

watchUserID int foreign key references user(userID)，

watchedUserID int foreign key references user(userID)，

Primary key (watchUserID,watchedUserID)

）

sameQusetion（userID，questionID）

Create table sameQusetion（

userID int foreign key references user(userID)，

questionID int foreign key references question(questionID)，

Primary key (userID,questionID)

）

make（movieID，personnelID，personnelType）

Create table make

(

movieID int FOREIGN KEY REFERENCES movie(movieID),

personnelID int FOREIGN KEY REFERENCES personnel(personnelID),

personnelType nvarchar（15），

PRIMARY KEY (movieID，personnelID，personnelType)

)

use（personnelID，Name）

Create table use

（

personnelID int FOREIGN KEY REFERENCES personnel（personnelID）

Name nvarchar（15）

PRIMARY KEY （personnelID，Name）

）

include（movieID，language）

Create table include

（

movieID int FOREIGN KEY REFERENCES movie（movieID）

language nvarchar（15）

PRIMARY KEY（movieID，language）

）

belong（movieID，typeName）

Create table belong

（

movieID int FOREIGN KEY REFERENCES movie（movieID）

typeName nvarchar（15）

PRIM

ARY KEY （movieID，typeName）

）

release（movieID，releaseTime，regionalVersion）

Create table release

（

movieID int FOREIGN KEY REFERENCES movie（movieID）

releaseTime date

regionalVersion nvarchar（15）

PRIMARY KEY （movieID，releaseTime，regionalVersion）

）

belongOne（movieID，countryName）

Create table belongOne

(

movieID int foreign key references movie(movieID),

countryName nvarchar(50),

Primary key(movieID，countryName)

)

contain（movieName，movieID）

Create table contain

(

movieID int foreign key references movie(movieID),

movieName nvarchar(50) primary key ,

)

own（movieID，movieDuration，regionalVersion）

Create table own

(

movieID int foreign key references movie(movieID),

movieDuration nvarchar(50) ,

regionalVersion nvarchar(50),

Primary key(movieID,movieDuration,regionalVersion)

)

confer（movieID，conferenceName，winner，awardsName）

Create table confer

(

movieID int foreign key references movie(movieID),

conferenceName nvarchar(50) ,

winner nvarchar(50),

awardsName nvarchar(50),

Primary key(movieID,conferenceName,winner,awardsName)

)

label（commentID，assessTag）

Create table label

(

commentID int foreign key references comment(commentID),

assessTag nvarchar(50),

Primary key(commentID,assessTag)

)