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)

)