今日头条用户部分设计

**1. 需求分析**

今日头条用户部分主要包括以下功能模块：

（1）用户注册。

（2）用户发表作品、修改作品、删除作品，注意存储细节数据，比如作品的展现量、发表时间等。

（3）用户可以收藏别人的作品。

（4）用户可以评论别人的作品。

（5）用户和用户之间可以私信交流。

系统处理的主要数据对象是用户和作品。和用户相关的信息有：用户的基本信息；和作品相关的信息有：作品的基本信息、作品的展现量；和二者相关的信息有：用户收藏作品、用户评论作品、用户私信交流的信息。

涉及的数据：

用户表：用户编号、用户名、密码、性别、注册时间、联系方式等。

作品表：作品编号、用户编号、作品标题、内容、发表时间、修改时间、展现量等。

收藏表：收藏编号、用户编号、作品编号、收藏时间等。

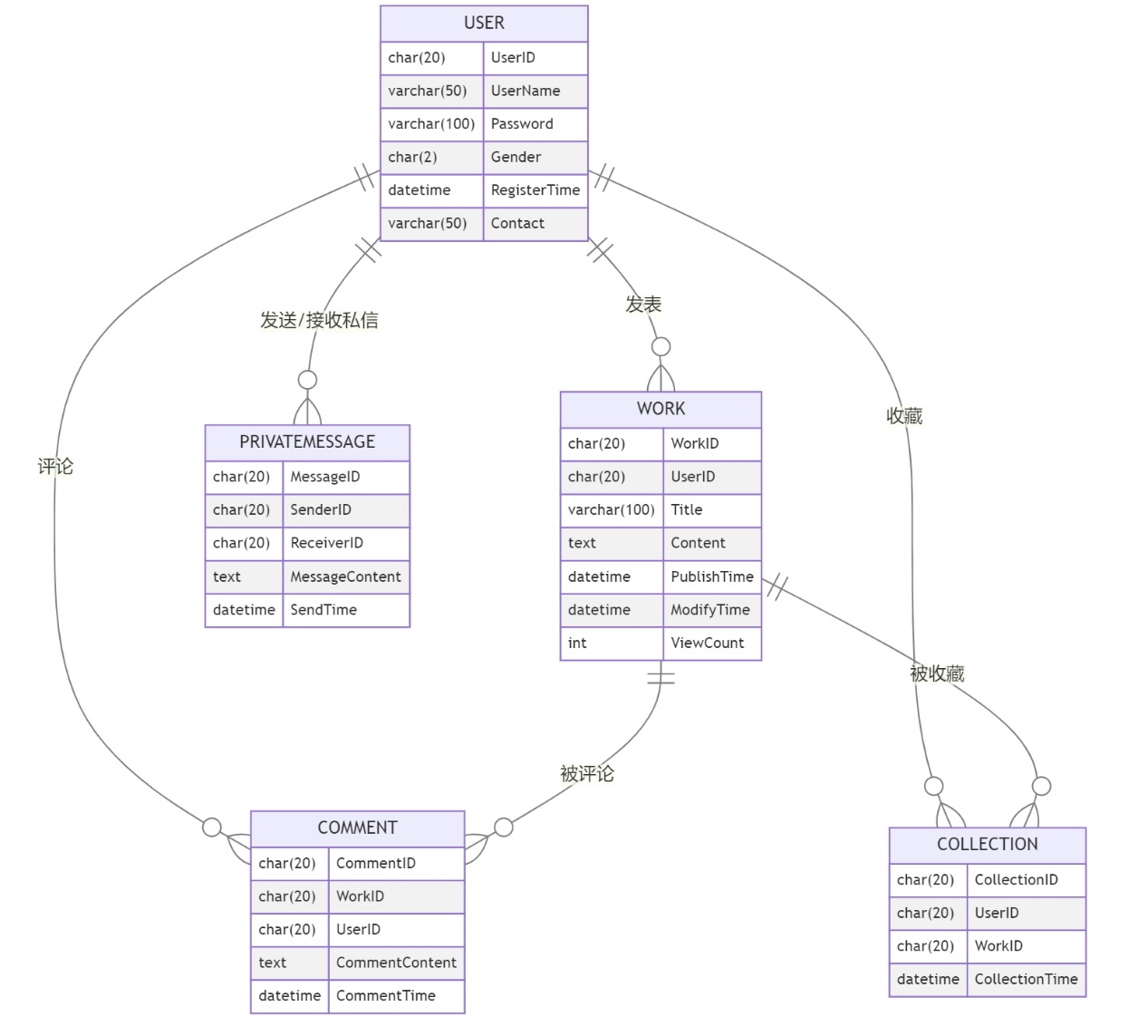
评论表：评论编号、作品编号、用户编号、评论内容、评论时间等。

私信表：私信编号、发送者编号、接收者编号、消息内容、发送时间等。

**2.概念设计**

在设计今日头条用户部分的数据库时，依据对系统做出的数据和功能的需求分析，确定要存储的有关对象的信息和各个对象的基本属性信息，还需要确定这些对象之间的相互关系。这一步设计称为数据库的概念设计，设计结果为数据库概念模型。

今日头条用户部分设计的概念模型如下图所示。



今日头条用户部分设计的的概念模型（E-R模型）

**3.逻辑结构设计**

逻辑结构设计是根据设计完成的概念模型，按照“实体和联系可以转换成关系”的转换规则，转换生成关系数据库管理系统支持的数据库表的数据结构。然后根据关系数据理论，对关系模式进行优化，使其达到BCNF。根据以上设计的今日头条用户部分的概念模型和实际应用中的需要，为系统设计出的各数据表的数据结构和完整性约束条件如下表所示。

（1）用户表(USER)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（宽度） | 是/否主键 | 空/非空 | 约束条件 |
| 用户编号(UserID) | char(20) | 是 | 非空 | 无 |
| 用户名(UserName) | varchar(50) | 否 | 非空 | 无 |
| 密码(Password) | varchar(100) | 否 | 非空 | 无 |
| 性别(Gender) | char(2) | 否 | 非空 | 值为：男或女 |
| 注册时间(RegisterTime) | datetime | 否 | 非空 | 无 |
| 联系方式(Contact) | varchar(50) | 否 | 非空 | 无 |

（2）作品表(WORK)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（宽度） | 是/否主键 | 空/非空 | 约束条件 |
| 作品编号(WorkID) | char(20) | 是 | 非空 | 无 |
| 用户编号(UserID) | char(20) | 否 | 非空 | 参照用户表 |
| 作品标题(Title) | varchar(100) | 否 | 非空 | 无 |
| 内容(Content) | text | 否 | 非空 | 无 |
| 发表时间(PublishTime) | datetime | 否 | 非空 | 无 |
| 修改时间(ModifyTime) | datetime | 否 | 空 | 无 |
| 展现量(ViewCount) | int | 否 | 非空 | 默认值为0 |

（3）收藏表(COLLECTION)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（宽度） | 是/否主键 | 空/非空 | 约束条件 |
| 收藏编号(CollectionID) | char(20) | 是 | 非空 | 无 |
| 用户编号(UserID) | char(20) | 否 | 非空 | 参照用户表 |
| 作品编号(WorkID) | char(20) | 否 | 非空 | 参照作品表 |
| 收藏时间(CollectionTime) | datetime | 否 | 非空 | 无 |

（4）评论表(COMMENT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列名 | 数据类型（宽度） | 是/否主键 | 空/非空 | 约束条件 |
| 评论编号(CommentID) | char(20) | 是 | 非空 | 无 |
| 作品编号(WorkID) | char(20) | 否 | 非空 | 参照作品表 |
| 用户编号(UserID) | char(20) | 否 | 非空 | 参照用户表 |
| 评论内容(CommentContent) | text | 否 | 非空 | 无 |
| 评论时间(CommentTime) | datetime | 否 | 非空 | 无 |

（5）私信表(PRIVATEMESSAGE)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 列表 | 数据类型（宽度） | 是/否主键 | 空/非空 | 约束条件 |
| 私信编号(MessageID) | char(20) | 是 | 非空 | 无 |
| 发送者编号(SenderID) | char(20) | 否 | 非空 | 参照用户表 |
| 接收者编号(ReceiverID) | char(20) | 否 | 非空 | 参照用户表 |
| 消息内存(MessageContent) | text | 否 | 非空 | 无 |
| 发送时间(SendTime) | datetime | 否 | 非空 | 无 |

**4.物理结构设计**

为了提高对数据表中数据的查找速度，在实现表的设计时，应该考虑建立适当的索引。今日头条用户部分数据库中建立的索引情况如下表所示。

|  |  |  |
| --- | --- | --- |
| 表名 | 列名 | 索引 |
| USER | UserName | 有重复索引 |
| WORK | Title | 有重复索引 |

由于经常进行批量数据的插入，建立存储过程，提高效率。

**5.数据库的创建**

根据以上对数据库的概念设计、逻辑设计和物理设计，得出了数据库中各表的数据结构。根据以上得出的各表设计结果，完成对各个数据表的数据结构创建。

CREATE DATABASE TouTiaoUser DEFAULT CHARACTER SET utf8;

USE TouTiaoUser;

CREATE TABLE USER (

UserID char(20) PRIMARY KEY NOT NULL,

UserName varchar(50) NOT NULL,

Password varchar(100) NOT NULL,

Gender char(2) NOT NULL CHECK (Gender IN ('男', '女')),

RegisterTime datetime NOT NULL,

Contact varchar(50) NOT NULL

);

CREATE TABLE WORK (

WorkID char(20) PRIMARY KEY NOT NULL,

UserID char(20) NOT NULL,

Title varchar(100) NOT NULL,

Content text NOT NULL,

PublishTime datetime NOT NULL,

ModifyTime datetime NULL,

ViewCount int DEFAULT 0 NOT NULL,

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

CREATE TABLE COLLECTION (

CollectionID char(20) PRIMARY KEY NOT NULL,

UserID char(20) NOT NULL,

WorkID char(20) NOT NULL,

CollectionTime datetime NOT NULL,

FOREIGN KEY (UserID) REFERENCES Users(UserID),

FOREIGN KEY (WorkID) REFERENCES Works(WorkID)

);

CREATE TABLE COMMENT (

CommentID char(20) PRIMARY KEY NOT NULL,

WorkID char(20) NOT NULL,

UserID char(20) NOT NULL,

CommentContent text NOT NULL,

CommentTime datetime NOT NULL,

FOREIGN KEY (WorkID) REFERENCES Works(WorkID),

FOREIGN KEY (UserID) REFERENCES Users(UserID)

);

CREATE TABLE PRIVATEMESSAGE (

MessageID char(20) PRIMARY KEY NOT NULL,

SenderID char(20) NOT NULL,

ReceiverID char(20) NOT NULL,

MessageContent text NOT NULL,

SendTime datetime NOT NULL,

FOREIGN KEY (SenderID) REFERENCES Users(UserID),

FOREIGN KEY (ReceiverID) REFERENCES Users(UserID)

);

插入数据实例：

INSERT INTO USER (UserID, UserName, Password, Gender, RegisterTime, Contact)

VALUES ('1', '张三', 'password123', '男', '2022-01-01 10:00:00', '1234567890');

INSERT INTO WORK (WorkID, UserID, Title, Content, PublishTime)

VALUES ('1', '1', '我的第一篇文章', '这是我的第一篇文章内容。', '2022-01-01 12:00:00');

INSERT INTO COLLECTION (CollectionID, UserID, WorkID, CollectionTime)

VALUES ('1', '1', '1', '2022-01-01 13:00:00');

INSERT INTO COMMENT (CommentID, WorkID, UserID, CommentContent, CommentTime)

VALUES ('1', '1', '1', '这是一个评论。', '2022-01-01 14:00:00');

INSERT INTO PRIVATEMESSAGE (MessageID, SenderID, ReceiverID, MessageContent, SendTime)

VALUES ('1', '1', '2', '你好，这是我的私信内容。', '2022-01-01 15:00:00');