Database Description

xiao.zhan

February 2019

1 System Overall Data Requirements

1.1 Overview

- Input of user basic information, including the user's number, the name used for registration and login, and the user's password
- When a system error occurs, which file is required to be logged, and the cause of the error and the specific time of occurrence
- The input of the basic information of the file, including the id of the file. The name of the file during creation and saving and modification, the content of the file
- Store and record permissions for each user and related files, including file id, user's uid, specific permissions, and last modification time

1.2 Organizational structure of this document

- Section 1: this system's data need
- Section 2: System visual Design
- Section 3: System Logic Design

2 System Visual Design

Analyze the basic needs of the library management system, use the abstract mechanism of conceptual structure design, classify the information in the required analysis results, organize, get the system entity, entity attribute, entity key, entity connection and type of contact You can design a conceptual model of the system.

Analyze the basic needs of the library management system, and use the abstract mechanism of conceptual structure design. Through the above analysis, the basic entities of the system can be extracted: user and file. These two entities are linked through the permission table, and the user and the file are one-to-many relationships.

2.1 E-R graph

The following is the overall E-R diagram and the specific field description of each table:

Figure 1: ertable

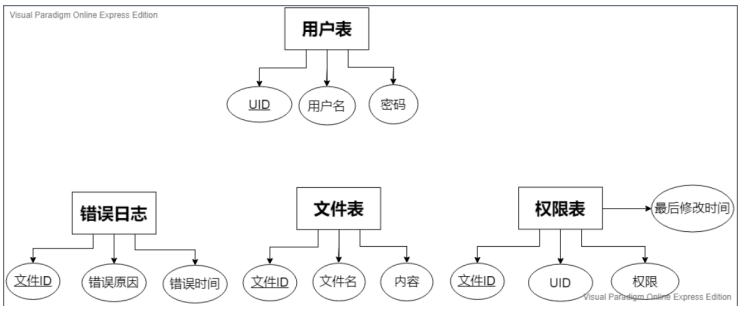

Figure 2: filetable

fieldname	datatype	field description	notes
fid	int	文件id	pk
filename	string	文件名称	
content	string	文件的内容	

Figure 3: usertable

fieldname	datatype	field description	notes
uid	int	用户编号	pk，唯一标识
username	string	用户名称	
password	string	用户密码	

3 System Logical Design

The logical structure of a relational model is a collection of relational patterns. The E-R diagram consists of three elements: the entity, the attributes of the entity, and the connections between the entities. So converting an E-R graph to a relational model is actually about transforming the attributes of entities, entities, and entities into relational patterns. The conversion principle is as follows.

- Entity type conversion: An entity type is converted into a relational pattern. The attribute of the entity is the attribute of the relationship, and the code of the entity is the code of the relationship.

Figure 4: errorlog

fieldname	datatype	datatype	notes
fid	int		pk
ereason	string	错误原因	
etime	char	错误时间	

Figure 5: permissions

fieldname	datatype	datatype	notes
fid	int	文件id	fk
Did	int	用户id	fk
permission	string	权限	
ltime	char	最后修改时间	

- 2. The conversion of the contact type is handled differently depending on the situation.
- A 1:1 relationship can be converted to an independent relationship mode, or it can be merged with the relationship mode corresponding to either end. If converted to an independent relational mode, the code of each entity connected to the contact and the attributes of the contact itself are converted into attributes of the relationship, and the code of each entity is a candidate code of the relationship. If corresponding to an end entity To merge a relational schema, you need to add the code of another relational schema and the properties of the association itself to the properties of the relational schema
- A 1:N connection can be converted to an independent relationship mode, or it can be merged with the N-side relational mode. If converted to an independent relational schema, the code of each entity connected to the association and the attributes of the association itself are converted into attributes of the relationship, and the code of the relationship is the code of the N-terminal entity.
- An M:N connection is converted to a relational mode. The code of each entity connected to the contact is a combination of each entity code
- Relationship patterns with the same code can be merged