

608 Database Systems Project 1

Name: Haifeng Jin UIN: 324003712

Name: Tao Wang UIN: 525004514

Topic

We created an application of *Person contact management*. It can help the user to manage one's contacts. This application can keep track of our contacts and the relationships. Multimedia information like sound, video and photos can be added to the contacts. The contact information can be searched and viewed anywhere and anytime through the website.

Database

Data collection:

First and Last name: we choose the 2016 top 100 popular first name and last name, then we create the full name with combinations of the first names and last names. The data type is varchar(45)

Phone: we create the phone number with permutations of the 1~9. The data type is varchar(45)

Email: The combinations of fullname and their company/university. The data type is varchar(45)

Location: the places where our contacts live in. Countries, states, cities were included. We have 2 countries, 10 states, 20 cities. All of them have the same data type, varchar(32)

Gender: Female & Male. Generated from their names. The data type is varchar(16)

Photo & Sound & Video: The data are downloaded from the internet. Only their address are stored in the database. varchar(45)

Database Schema:

Contact: basic information of each contact.

Name	Type	Description
contact_id	int(11)	Primary key.

name	varchar(45)	Full name.
email	varchar(45)	
phone_number	varchar(45)	
gender_id	int(11)	
location_id	int(11)	

Sound: store the path to all the sound files of each contact.

Name	Type	Description
path	varchar(45)	Path to the sound file.
sound_id	int(11)	Primary key.
contact_id	int(11)	Foreign key to contact.

Photo: store the path to all the photo files of each contact.

Name	Type	Description
path	varchar(45)	Path to the photo file.
photo_id	int(11)	Primary key.
contact_id	int(11)	Foreign key to contact.

Video: store the path to all the video files of each contact.

Name	Type	Description
path	varchar(45)	Path to the video file.
video_id	int(11)	Primary key.
contact_id	int(11)	Foreign key to contact.

Gender: mapping gender_id to gender_name. 1 to Male, 2 to Female.

Name	Type	Description
gender_id	int(11)	Primary key.
gender_name	varchar(16)	Male or Female.

Location: store the location of the contacts.

Name	Type	Description
location_id	int(11)	Primary key.
city	varchar(32)	City name.
state	varchar(32)	State or province name.
country	varchar(32)	Country name.

Relation: Mapping relation_id to its name: Family, Classmate, Friend, Colleague.

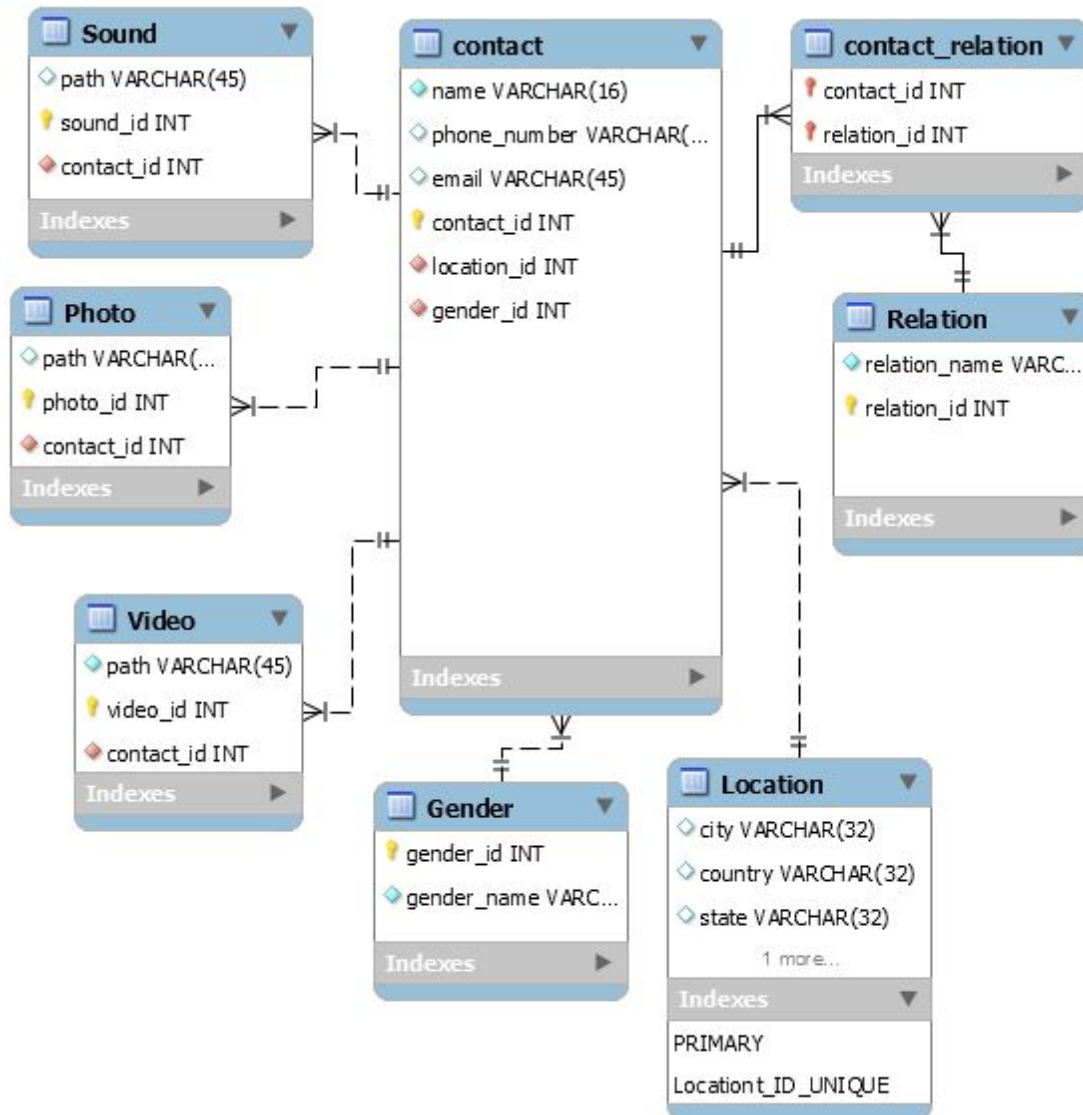
Name	Type	Description
relation_id	int(11)	Primary key.
relation_name	varchar(16)	Family, classmate and etc.

Contact_relation: connecting contact table with relation table.

Name	Type	Description
contact_id	int(11)	Primary key.
relation_id	int(11)	

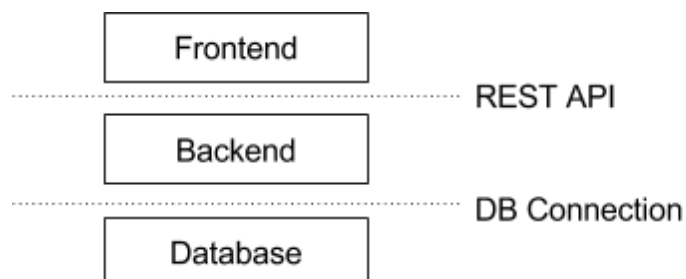
We have 7 tables in total. They are closely related to each other. Each contact could have multiple photos, sounds and videos. Multiple contacts may live in the same city. Multiple contacts may share the same gender. Multiple contacts may have the same relation. One contact may have multiple relations. Due to these complicated interconnections between tables. We have the following description of the relation between tables or entities.

- (1) sound->contact: n:1
- (2) video->contact: n:1
- (3) photo->contact: n:1
- (4) gender->contact: 1:n
- (5) location->contact: 1:n
- (6) relation->contact: n:n



Architecture

The application has three components, front-end, back-end, and database. The architecture is shown in the following figure.



Frontend:

Technology used in frontend: HTML, Javascript, Bootstrap, JQuery.

Front-end is the user-interface including some logic on the webpages. It send users' requests and exchange data with the backend server with REST API. Bootstrap is used to make the UI looks nicer. JQuery is used to build a better interactive UI to the users.

Whenever the user clicked some button on the web page. The frontend Javascript would update the webpage by pulling up a dialog or refreshing the contact list.

REST API

Address	Methods	Explanation
/rest	Get	Get contact's information using full name
/rest/all	Get	Get all information of all contacts
/rest/video/{id}	Get	Get the video' path using contat_id
/rest/sound/{id}	Get	Get the sound' path using contat_id
/rest/photo/{id}	Get	Get the photo' path using contat_id
/rest/relation/{id}	Get	Get the RelationName using relaiton_id
/rest/{id}	Get	Get contact's information using contact_id
/rest	Post	Create the new contact using the input information
/rest	Put	Update the information using the input information
/rest/{id}	Delete	Delete one contact

Backend:

Technology used in backend: Node.js, Rest API, express, body-parser.

The backend server is responsible for all the business logic. It queries or changes the data in the database according to the requests sent by the frontend. We use Node.js to build the backend service. Express, which is a library for Node.js, is used to build REST APIs.

Body-parser, another Node.js library, is used for parsing the body content of the POST requests from the frontend.

SQL:

MySQL is used as the database for this project. Backend will connect to database using a mysql connection library.

Difficulty and Learn

Difficulties & Solutions

Difficulty 1: The logic of our database's schema is complicated.

Solution : we first tried to design the schema using all of the knowledges we have learned, such as various kind of join operations, foreign keys etc. But how to define the relations between tables, what foreign keys should we have, to make my database more efficient. We solved these problems by looking up the database paradigms and follow its rules.

Difficulty 2: The concurrency of database's connection:

Solution: Node.js do not have much documentation on MySQL connection. We used a package called "mysql" to connect to our database. However, there are some issues about database concurrency connections. If the previous function did not release the connect, an error would occur at the next connection at the database. To solve this problem, we call another function called "createPool", which would create a connection pool for mysql service, which solves the concurrency problem. Every time some function tries to connect to the database, the pool would simply assign one of the many connections to it, so that there won't be any conflict at the next connection.

Difficulty 3: The violations of foreign-key constraint:

Solution: We encountered the problem that the teacher has talked about. So we find the solution in the teacher's PPT.

Learn

It is good to get the opportunity to have some experience in programming with real database system. In our database, there are more than 10000 tuples, and we wrote SQL programs to manipulate, query the database. In addition, we learn how to use REST API to let frontend interact with backend. Last but not least, we learned how to use Node.js to deal with the request from frontend, and use html & bootstrap to create amazing webpages.

Screenshots

Showing the contact list.



Contact List

e.g.: Haifeng Jin

Qcreate

Noah Smith Phone: 123456789			
Liam Smith Phone: 123456798			
Samuel Smith Phone: 123468975			
Caleb Johnson Phone: 123486795			
Liam Williams Phone: 123495876			

Showing contact details.

Contact BookHome

by Haifeng Jin & Tao Wang

Contact Detail

First Name
Noah

Last Name
Smith

Phone
123456789

Email
Noah Smith@tamu.com


Gender
Male

Relation
Family Classmate

Country
China

State
Jiangxi

City
Nanchang



Delete

Update

Contact List

Qcreate

Noah Smith Phone: 123456789			
Liam Smith Phone: 123456798			
Samuel Smith Phone: 123468975			
Caleb Johnson Phone: 123486795			
Liam Williams Phone: 123495876			