Project 1 : Your Dream Home

Harshit Goel (2016CS10319), Kashish Jain (2016EE10434), Akanshu Gupta (2016EE10418)

Due date: February 24, 2019, 11:55pm IST

1 Project Description

The project involves a portal where users can buy, sell or rent houses across the 13 districts of Beijing, China. House, being a prime necessity arises the need for building a user-friendly portal where users from all fields can select houses by various specifications, which was primarily the motivation behind the project.

A user is first required to create an account on the site. Then he has the option to choose among more than 3,00,000 homes based on his personal filters like number of bedrooms, range of price, nearest university, flat area and various other user-oriented filters. Once a house is bought, user can see his transaction details, cash balance, houses he has. Furthermore, the user has the option of putting any of his houses he bought on sale or rent which any other user, once logged in can purchase.

The entities and attributes are shown in the table below. Also the ER diagram is also shown on the next page

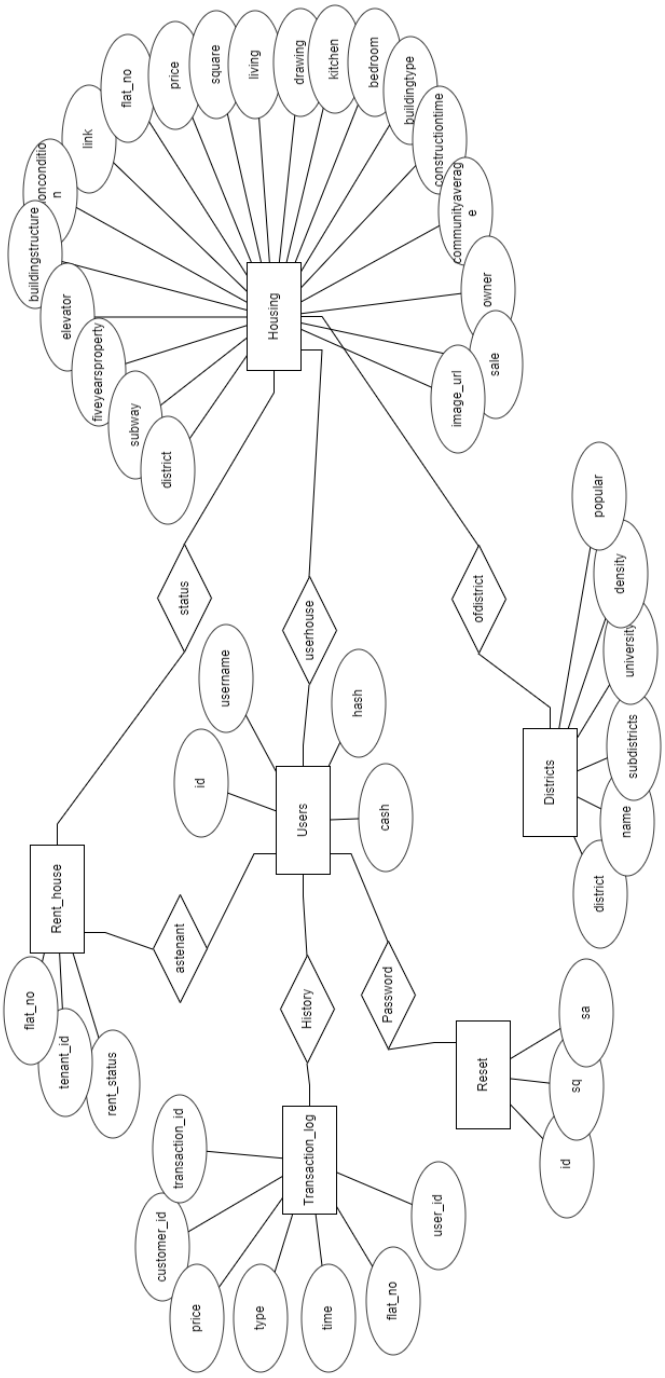
Entity	Attributes
districts	district, name, subdistricts, university, density, popular
housing	link, flatno, price, square, living, drawing, kitchen, bedroom, buildingtype, constructiontime, renovationcondition, buildingstructure, elevator, fiveyearsproperty, subway, district, communityaverage, sale, owner, image url
rent house	flat no , tenant id, rent status
reset	id, sq, sa
transaction log	transaction id, customer id, price, type, time, flat no, user id
users	id, username, hash, cash

2 Data Sources and Statistics

The main dataset for our project was taken from :- https://www.kaggle.com/ruiqurm/lianjia

The entity ” housing” was derived from it after cleanups. Firslty, the relevant columns were selected as filters. Many cells had some chinese characters and removing them was thus necessary. All the chinese characters were of a few types and hence were repeating many times. They were all replaced with null values. We added two columns defining status of house like rent, sale etc. The ” districts” entity was manually created by googling and finding relevant information about the 13 districts of Beijing

Table	No of Tuples	Time to Load	Raw Dataset Size	Raw Dataset Size(Clean)
districts	13	113 ms	-	834 bytes
housing	318850	1121 ms	56 MB	32.1 MB
rent house	4	98 ms	--	-
reset	5	53 ms	-	-
transaction log	42	178 ms	-	-
users	5	53 ms	-	-



Housing
link
flat_no
price
square
living
drawing
kitchen
bedroom
buildingtype
constructiontime
communityaverag e
owner
sale
image_url
onconditio n
buildingstructure
elevator
fiveyearsproperty
subway
district
status
userhouse
ofdistrict
Districts
popular
density
university
subdistricts
name
district
Users
username
id
hash
cash
Rent_house
flat_no
tenant_id
rent_status
astenant
History
Password
Reset
sa
sq
id
Transaction_log
transaction_id
customer_id
price
type
time
flat_no
user_id

3 Functionality and working

3.1 User’s view of the system

a) Homepage

The user sees 2 tabs on the top right hand corner, namely ” login” and ” register” . If the user already has an account, he can directly sign in from the home page itself

b) Register

The user needs to register here with a username, password and security question to create an account and login to the portal.

c) Dream Home

Once logged in, the user sees details of all his houses and their status, that is sale, rent or not on sale. He also sees all the specifications of his houses. Also now he has access to 5 tabs (Dream Home, Quote, Buy, Rent, History) on the top left and ’ Logout’ tab on the top right corner

d) Quote

Here, the user can see his user-id and remaining balance

e) Buy

This is the main area where the user has the privileges to choose from all the houses listed on sale and purchase them. On his right he sees various filters and he can enter his desired preferences to filter out the respective houses. Below each house, he can see all the available details of that house and all go to the website of that particular house to know more about it.

f) Rent

Similar to the buy tab, here the user can see all the houses available for rent and buy again specifying his desires, he can choose a house to take on rent.

g) History

Here the user see all the details pertaining to his previous transactions like flat id, trade type, price and time

h) Logout

The user is redirected to the homepage once he clicks on this tab

3.2 Special Functionality

a) Indexes
The flat id of all the 3,00,000+ houses are indexed using a B-Tree for fast lookups.
All the primary keys of other entities are also indexed similarly

b) Constraints
Asides from primary key constraints on each table, we have
inserted foreign key constraints wherever applicable.

c) Sequences
We have sequences which keep track of the users and of the next
transaction to be made

d) Views
We have created two views, one of the houses the user is as
tenant and the other of the houses he owns. On his Dream Home

tab, the users see the union of these two views.
e) Trigger
We have created a trigger, wherein whenever user buys/sells a house, the time is updated in the transaction log of the user.

f) Transaction
For the history, we have created a set of queries which either all execute simultaneously or dont execute at all

3.3 List of Queries

a) Register

- INSERT INTO users (username,hash) VALUES ($1, $2);
- SELECT id FROM users WHERE username = $1;
- INSERT INTO reset (id, sq, sa) VALUES ($1, $2, $3);

b) Login

- SELECT * FROM users WHERE username = $1 ;

c) Buy

- select flat_no, price, square, kitchen, living, bedroom, buildingtype, constructiontime, renovationcondition, buildingstructure, elevator, subway, name, communityaverage, link, university as near, image_url from housing,districts wherehousing.district = districts.district and sale = 1 order by flat no asc limit50;
- select price,owner from housing where flat no = $ 1 ;
- select cash from users where id = $ 1 ;
- INSERT INTO transaction log (transaction id,customer id,price,type,time,flat no,user id) VALUES ($1, $2, $3, 'BUY' , $4, $5, $6);
- UPDATE users SET cash= ($1) where id = $ 1 ;
- select cash from users where id = $ 1 ;
- UPDATE users SET cash= $1 where id = $2 ;
- INSERT INTO transaction log (transaction_id,customer_id,price,type,time,flat no,user id) VALUES ($1,$2,$3, 'sold' , $4, $5, $6);
- UPDATE housing set owner = $1, where flatno = $2 ;
- select flatno, price, square, kitchen, living, bedroom, buildingtype, constructiontime, renovationcondition, buildingstructure, elevator, subway, name, communityaverage, link, university as near, image_url from housing,districts where housing.district = districts.district and cast(kitchen as text) like $1 and cast(flatno as text) like $1 and cast(living as text) like $1 and price between and and cast(bedroom as text) like $1 and districts.district in and square between and and buildingstructure in s and cast(elevator as text) like $1 and sale = 1 and university like $1 limit 50

d) Rent

- select price,owner from housing where flat no = $ 1 ;
- select cash from users where id = $1;
- select cash from users where id = $1;
- INSERT INTO transaction log (transaction id,customer id,price,type,time,flat no,user id) VALUES($1,$2,$3, 'rent' , $4, $5, $6);
- UPDATE users SET cash= where id = $1 ;

- INSERT INTO transaction_log (transaction_id,customer_id,price,type,time,flat_no,user_id) VALUES($1,$2,$3, 'rent' , $4, $5, $6);
- UPDATE rent_house set tenant_id = $1, rent_status = 0 where flat_no = $2;

e) Reset Password

- UPDATE users SET hash= $1 WHERE username = $2;
- SELECT * from users JOIN reset ON users.id = reset.id where username = $1;

f) History

- SELECT * FROM transaction_log WHERE userid = $1;

g) Quote

- select cash from users where id = $1 ;

h) Index

- (select * from house owner where id = $1) union (select * from tenant where id = $2)limit 50;
- Update housing set sale = 0 where flat no = $1 ;
- Update housing set sale = 1 where flatno = $1;
- UPDATE rent house SET tenant id = NULL, rent status = 1 where flat no = $1 ;
- INSERT INTO rent house VALUES ($1, NULL, 1);
- DELETE FROM rent house where flatno = $1;

4. Query Run times

Query Number	Average Running Time (ms)
1	1.61
2	2.52
3	1.45
4	2.79
5	976.38
6	5.46
7	7.62
8	4.53
9	6.59
10	3.46
11	5.48
12	4.75
13	537.26
14	4.36
15	3.78
16	6.93
17	7.65
18	3.46
19	8.34
20	7.46
21	4.23
22	2.16
23	5.34
24	1.21
25	4.78
26	8.46
27	3.67
28	7.43
29	9.15
30	7.26
31	5.46