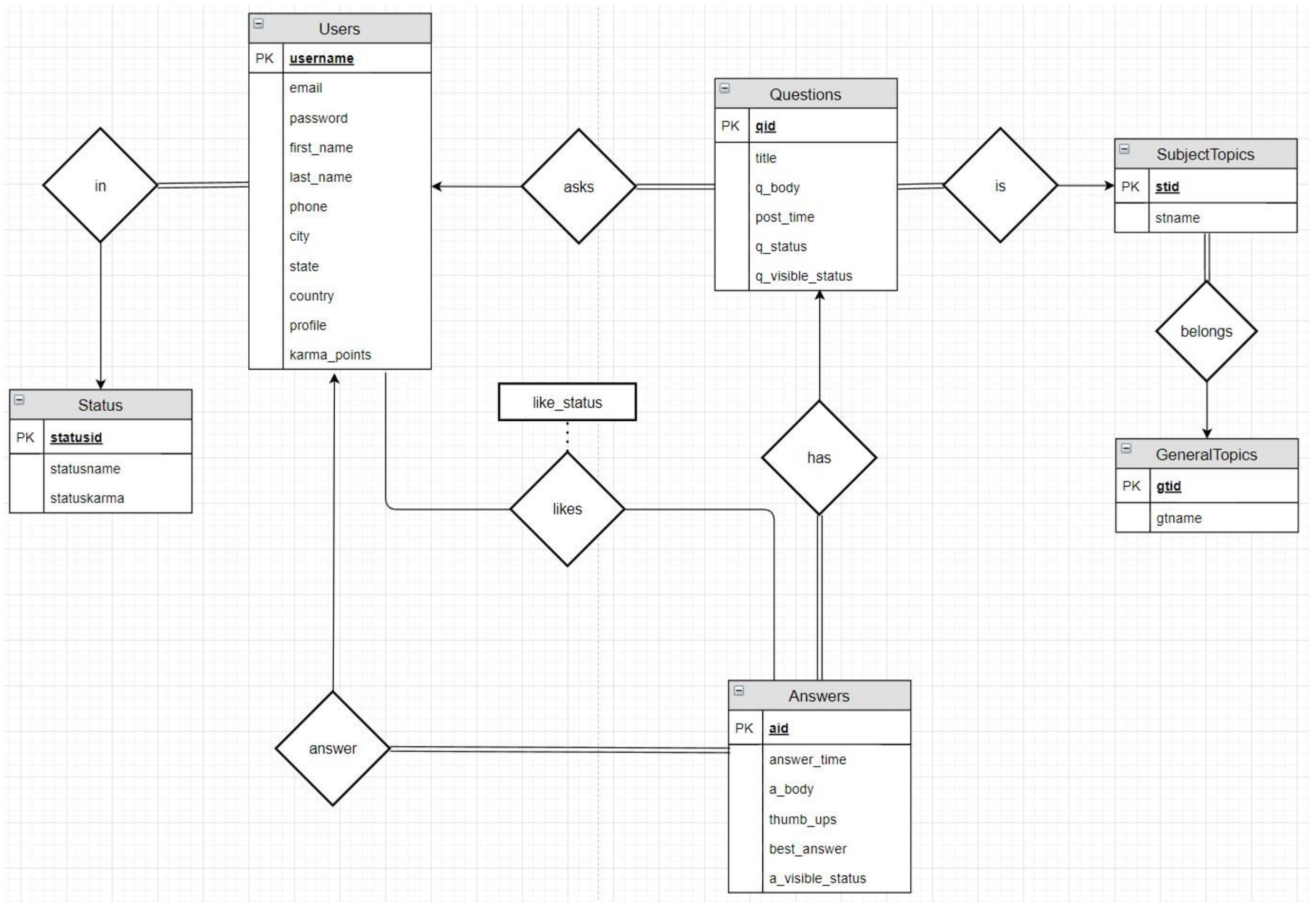
CS-GY 6083 Course Project

Minghao Shao (ms12416) & Yiming Li (yl7538)

1 ER Diagram



2 Relational Table Design

2.1 Users

```
CREATE TABLE Users (
   username VARCHAR(45) NOT NULL,
   email VARCHAR(50),
   password VARCHAR(128) NOT NULL,
   firstname VARCHAR(45),
   lastname VARCHAR(45),
   phone VARCHAR(10),
   city VARCHAR(60),
   state VARCHAR(2),
   country VARCHAR(60),
   profile VARCHAR(512),
   karma_points INT NOT NULL default 0,
   PRIMARY KEY (username)
);
```

The data started from the user table, which was used to store all the information may required about the user. For the primary key, a unique username was used, also used for login the session, instead of AUTO_INCREMENT field. The username was designed and set by the user instead of generated by the system automatically each time a new account created. There were some benefits using this design, as a primary key, each time a user registering a new account, the system would check the uniqueness of this username, in which case the duplicate issue could be avoided since the system would check this, no manual action required. For the password field, instead of storing the password in plain text, SHA256 hashing function was used. In the test data, it was a simple SHA256 hashing, however, in the part2, more complex policy would be applied such as salting. Other information fields such as firstname, lastname, email, phone, city, state and country were designed to be nullable since sometimes user would prefer not to provide their personal information, hence this is not forced. Another field designed was the karma point. Which was used to identify the status of the user, regarding their level.

The policy regarding the karma point was according to the quantity and the quality of the answers the current user posted, say 10 points added each time the user posted an answer, and each time any answer the user posted received a “like”, the user could also get 10 points bonuses. Another facet about earning the karma point was when the user’s answer was selected by the question owner, the user who posted the best answer would also receive the bonuses, in test data, 20 points was set. Accordingly, like cancel was also taken into consideration, each time a user canceled the like, the 10 karma points bonuses would be dismissed also, when the best answer updated and the user’s the answer was no longer the best answer, the 20 points bonuses would also be dismissed. The mechanism of question or answer deletion was also designed and would be discussed in Questions and Answers tables section respectively.

What is changed: In the previous design, we set the email, the firname, and the last name as not null, but in this part, they were designed that could be null, since not all the users decided to enter the real name with the mail address.

2.2 Status

```
CREATE TABLE Status (
  statusid INT AUTO_INCREMENT,
  statusname VARCHAR(20) NOT NULL,
  statuskarma INT NOT NULL,
  PRIMARY KEY (statusid)
);
```

A separate table called Status was also design, which was used to store the information of each level of the user, at current stage, three levels were designed, basic, advanced and expert. In this table, statusid was used as primary key which was AUTO_INCREMENT, also a status name, and the threshold called statuskarma, which stored the total karma point required for a user to reach this level. In this design, the basic level was from 0 to 50, advanced was from 50 to 100, and expert was larger then 100. Note that the numeric design was set only for the test data, to make some queries output the user with each level which could be optimized, for example, in part2, the threshold for each level would be increased, or the bonuses received by the user would be deducted.

2.3 UserStatus

```
CREATE TABLE UserStatus (
  username VARCHAR(45) NOT NULL,
  statusid INT NOT NULL default 1,
  PRIMARY KEY (username),
  FOREIGN KEY (username) REFERENCES Users(username),
  FOREIGN KEY (statusid) REFERENCES Status(statusid)
);
```

The status of the user was not stored in Users table to satisfy the 3NF. Since the status of the user was totally depended on the karma points the user received, and the number of karma points received by the user was depended on the username. Hence, they were designed to divide to 2 separate table. The primary key was the username, which was enough to identity a user and the relation between the user and their status was 1 to many, which means a user could only have one status. From the statusid of each user, a join could be performed to get the exact status of the user from the status table.

2.4 GeneralTopics

```
CREATE TABLE GeneralTopics (
  gtid INT NOT NULL AUTO_INCREMENT,
  gtname VARCHAR(45) NOT NULL,
  PRIMARY KEY (gtid)
);
```

The structure of the topic was designed as hierarchy. Two levels were designed and the GeneralTopics table was the first layer. In this table, only the general topic could be stored such as MATH, PHYSICS or COMPUTER SCIENCE. In this table, a AUTO_INCREMENT field as primary key was design called gtid (general topic id), and another one was the name of the general topic.

The topics was aimed to be selected by a list box in the code implementation, there were several pre-defined topics, or the user may add topics on their own.

2.5 SubjectTopics

```
CREATE TABLE SubjectTopics (
  stid INT NOT NULL AUTO_INCREMENT,
  stname VARCHAR(50) NOT NULL,
  gtid INT NOT NULL,
  PRIMARY KEY (stid),
  FOREIGN KEY (gtid) REFERENCES GeneralTopics(gtid)
);
```

The next level of the topics was called SubjectTopics, in which the subjects were stored under their general topics. Similar as general topic, a stid (subject topic id) was design as the primary key, with AUTO_INCREMENT, also with the stname which was the name of the subject topic. Each subject topic should be able to reference to their parent topic, hence a foreign key was set which refenced to the gtid of their parent topic. Such as a topic called ‘Database System’, which was certainly under the COMPUTER SCIENCE general topic, hence the gtid of this subject topic would be the gtid of the general topic CS.

Also same as the general topic, in code implementation they were designed to be selected with a list box after the general topic was selected when posting a new question. Some pre-defined topics would also be provided by the system. Or users were able to add new subject topics.

2.6 Questions

```
CREATE TABLE Questions (
  qid INT NOT NULL AUTO_INCREMENT,
  q_username VARCHAR(45) NOT NULL,
  stid INT NOT NULL,
  title VARCHAR(45) NOT NULL,
  q_body VARCHAR(512) NOT NULL,
  post_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  status VARCHAR(15) NOT NULL,
  q_visible_status INT NOT NULL default 1,
  PRIMARY KEY (qid),
  FULLTEXT (title),
  FULLTEXT (q_body),
  FOREIGN KEY (q_username) REFERENCES Users(username),
  FOREIGN KEY (stid) REFERENCES SubjectTopics(stid)
);
```

This table was designed to store the question information posted by the user. The primary key was called qid which was a AUTO_INCREMENT field with int type. For each question, the username who posted the question was also recorded into the field q_username. In consideration that users may either post questions, or give answers to questions, q_username was used instead of the username, to make the table clearer, but the q_username was still designed as a foreign key referenced to the username in Users table only the name of the field was different. In this design, general topic should not be used as the topic of the questions, but just to identity the subject topics which belonged to the general topic. Hence in the question table, a foreign key referenced to subject topic's stid field was used to identify the topic of the question. Also, the title and q_body field was used to store the title and the body of the question along with the post_time. In the test data, the post_time was inserted manually to show the test result and to simulate the real world condition, but the design for code implementation was always obtain the current time by the system by calling now().

The questions table also had a field called status, in this design, two statuses were defined, solved and unsolved, the user who posted the question could modify the status of the question. The status of the question would neither affect the karma points the user obtained nor affect how the question or the answers under this question would behave, but only a label which was used to make it convenient for the users who were also looking for the solution of similar questions. In in this, another situation was also considered for part2. Since in some cases, users would like to delete the questions they posted hence a field called a_visible_status was added to this table. Instead of deleting the question row directly, it would be safer to set the visible status of the question, if the question was not visible, then in the implementation in part2, it could also be regarded as deleted. Another benefit of using this field was that the user may be able to set the question as private/public in case they don't want others to see these questions (in case some privacy issue involved).

What is changed: Since in part2, for the keyword query technique, the fulltext search was supported to become the second layer of the keyword query, hence the fulltext key for title and q_body was added.

2.7 Answers

```
CREATE TABLE Answers (
  aid INT NOT NULL AUTO_INCREMENT,
  qid INT NOT NULL,
  answer_time DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
  a_username VARCHAR(45) NOT NULL,
  a_body VARCHAR(100) NOT NULL,
  thumb_ups INT NOT NULL,
  best_answer INT NOT NULL default 0,
  a_visible_status INT NOT NULL default 1,
  PRIMARY KEY (aid),
  FULLTEXT (a_body),
  FOREIGN KEY (qid) REFERENCES Questions(qid),
  FOREIGN KEY (a_username) REFERENCES Users (username)
);
```

This table was used to store all the answer posted by the users regarding the question. Same as most of strong entities in this design, aid was added as AUTO_INCREMENT primary key of this table. Also, there were two foreign keys, one was the qid references to question's id to obtain which question the current answer was belonged to. Another one was the username called a_username, with this field it was able to track the user who posted this answer, and each time the answer status changed, such as set to best answer or received a like, with this key it was also possible to provide the user with according karma points bonuses.

The table also designed with the answer's body and the answer_time which was the time the answer was posted, the likes that answer received. Another way to get the number of likes of the answer each time required such as loading the answer was to join the answer take with likes table and output the count value, in this design, a field recording the number of likes the answer received called thumb_ups was used instead with a trigger which would be discussed in the trigger section below because it was regarded as a more efficiency approach since the query of getting the count for each answer was no longer required with this field and the only operation to do to handle a new update of like was just update or insert likes table, then the trigger would update the answer table automatically.

There was also one field called best_answer with value 0 and 1 was used to store whether the answer was selected as best answer. The best answer information was not stored in the question since the query clause could get the best answer of each question directly, since it was regarded as an attribute of the answer but not the question. Also, same as question table, the possibility of deleting the answer was taken into consideration by adding a field called a_visible_status. Same as questions table, if the user deleted this answer, instead deleting it directly, it was better to set the answer invisible.

What is changed: Similar as the Questions table, the fulltext query was also supported, hence the fulltext key for a_body was also supported.

2.8 Likes

```
CREATE TABLE Likes (
  username VARCHAR(45) NOT NULL,
  aid INT NOT NULL,
  like_status INT NOT NULL default 1,
  PRIMARY KEY (username, aid),
  FOREIGN KEY (username) REFERENCES Users (username),
  FOREIGN KEY (aid) REFERENCES Answers (aid)
);
```

In order to track the user who gave a like to an answer and make sure they were able to take the like back especially they gave a like by mistake, this table was added to track this information. Since for each unique answer, a user could only give a single thumb_up, hence username and aid was used as primary key. Another field added was the like_status, the logic was when a new like inserted (the user never gave a thumb_up to this answer), the record was inserted directly, while each time the user update a thumb_up, such as withdrawing a thumb_up or give the thumb_up back, the system would only update the like_status to 1 or 0.

3 Trigger Design

3.1 Triggers for Likes

Likes table were designed with two triggers, an after insert trigger and an after update trigger.

```
create trigger Likes_after_insert after insert on Likes for each row
begin
     update Answers
    set thumb_ups = thumb_ups + 1
    where aid = new.aid;
end;
```

Each time the a new thumb_up added by a user, the thumb_ups field in answers table would be updated and added 1.

```
create trigger Likes_after_update after update on Likes for each row
begin
    if new.like_status <> old.like_status then
        if new.like_status = 0 and old.like_status = 1 then
            update Answers
            set thumb_ups = thumb_ups - 1
            where new.aid = Answers.aid;
        elseif new.like_status = 1 and old.like_status = 0 then
            update Answers
            set thumb_ups = thumb_ups + 1
            where new.aid = Answers.aid;
        end if;
    end if;
end;
```

Each time the like_status was changed, there were two cases, the first case was a user withdraw a like, then the thumb_ups number in answers would decrease by 1, respectively, if a user gave a like back to the answer that was once withdrawn like from that user, the thumb_ups value in answers table would be added back, by 1.

3.2 Triggers for Answers

Answers table was designed with two triggers, an after insert trigger and an after update trigger.

```
create trigger Answers_after_insert after insert on Answers for each row
begin
     update Users
    set karma_points = karma_points + 10
    where Users.username = new.a_username;
end;
```

According to the design, each time a user posted a new answer, this user could obtain 10 karma points, even the answer was deleted. The Answers_after_insert trigger would implement this logic, each time a new answer inserted, which means a user posted a new answer, this user's karma point would add 10.

```
create trigger Answers_after_update after update on Answers for each row
begin
     if new.best_answer <> old.best_answer then
          if new.best_answer = 1 and old.best_answer = 0 then
               update Users
               set karma_points = karma_points + 20
               where Users.username = new.a_username;
          elseif new.best_answer = 0 and old.best_answer = 1 then
               update Users
               set karma_points = karma_points - 20
               where Users.username = new.a_username;
          end if;
     end if;

     if new.thumb_ups - old.thumb_ups <> 0 then
          update Users
          set karma_points = karma_points + 10 * (new.thumb_ups - old.thumb_ups)
          where Users.username = new.a_username;
     end if;
end;
```

There were two cases when updating the answers table. 1. The best answer status changed. In this design, each time an answer was selected as best answer, the user who posted this answer would obtain 20 karma points. Respectively, if an answer was no longer the best answer, the karma points of the user who posted the old best answer would be taken back which was -20 karma points. 2. The answer's thumb_ups changed. If the answer was deprived of a like, the karma points of the user who posted this answer would increase 10, respectively, if a new like were given, or the like was given back to this answer, for each like, the karma point of the user who posted this answer would decrease 10.

3.3 Triggers for Users

Users table was designed with two triggers, an after insert trigger and an after update trigger.

```
create trigger User_after_insert after insert on Users for each row
begin
    insert into UserStatus(username) values(new.username);
end;
```

Each time a new user account was created, the link to the status table would be added. Since for each new user, the userstatus was always 1 which was the basic level, hence as described in the schema design section, the default value for UserStatus was 1.

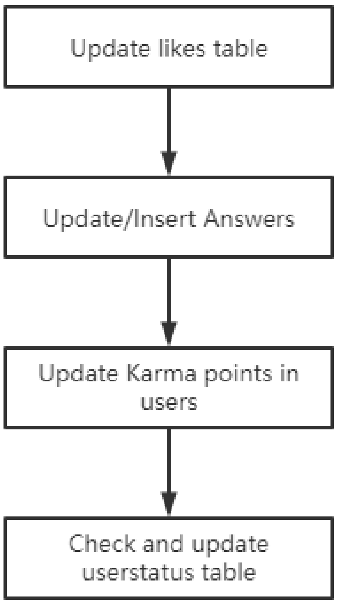
```
create trigger User_after_update after update on Users for each row
begin
    declare advance_thres INT;
    declare expert_thres INT;
    declare basic_status INT;
    declare acvanced_status INT;
    declare expert_status INT;
   if new.karma_points <> old.karma_points then
        select statusid into basic_status from status where statusname = 'basic';
        select statuskarma, statusid into advance_thres, acvanced_status from status where
statusname = 'advanced';
        select statuskarma, statusid into expert_thres, expert_status from status where
statusname = 'expert';
        if new.karma_points >= expert_thres and old.karma_points < expert_thres then
            update UserStatus
            set statusid = expert_status
            where UserStatus.username = new.username;
        elseif new.karma_points >= advance_thres and (old.karma_points < advance_thres or
old.karma_points >= expert_thres) then
            update UserStatus
            set statusid = acvanced_status
            where UserStatus.username = new.username;
        elseif new.karma_points < advance_thres and old.karma_points >= advance_thres
```

```
then
                update UserStatus
                set statusid = basic_status
                where UserStatus.username = new.username;
            end if;
        end if;
end;
```

Each time a user's karma point changed, the system would check the user's current status. The threshold of karma point of each status would be obtained from status table and the user's status in userstatus table would be updated according to the user's current karma point.

The default design for each level was: 0 – 50 was basic level, 50 – 100 was advanced level and karma points larger than 100 was expert level.

3.4 Workflow of the triggers

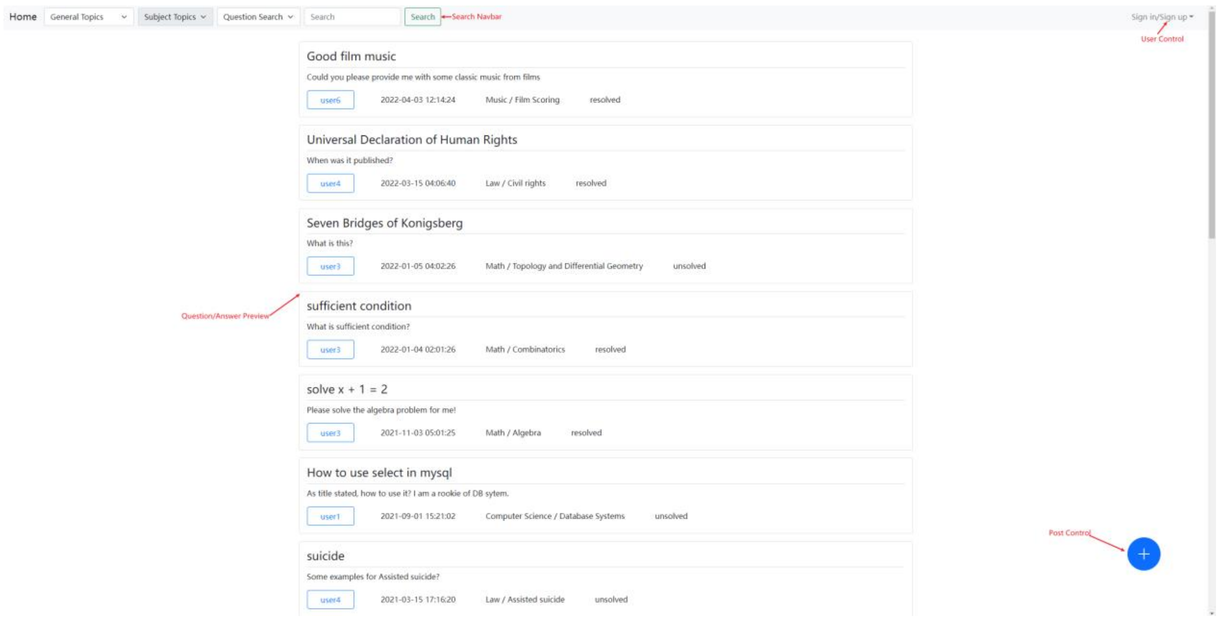

Update likes table
Update/Insert Answers
Update Karma points in users
Check and update userstatus table

The work flow of the triggers shown above, when updating any table listed in the workflow manually in the system, the trigger would be called from up to down. For example, updating or inserting the answers table would call the according triggers in users and then the userstatus.

4 Functionality

4.1 Main page and UI Design

As the entrance of the program, the main page would support the preview of all the questions and answers. By default, it would only show the question preview, but answers preview was also supported, which would be introduced in Query and Filter session. In this design, even the user was not login, they were still able to view the questions and answers but not be able to post any content. There were also a navbar which was enabled globally, which was used to support keyword query, and topic selection, and the user control. At the beginning, the main page would show all the questions discard of the topic from latest to earlies. User was able to click the title of either questions or answers card on that page to enter the detailed page. For each posted content, the user who posted the content was also designed to be a button, by clicking the button, the profile of the user could be inspected.



Good film music
Could you please provide me with some classic music from films
user6 2022-04-03 12:14:24 Music / Film Scoring resolved

Universal Declaration of Human Rights
When was it published?
user4 2022-03-15 04:06:40 Law / Civil rights resolved

Seven Bridges of Konigsberg
What is this?
user3 2022-01-05 04:02:26 Math / Topology and Differential Geometry unsolved

sufficient condition
What is sufficient condition?
user3 2022-01-04 02:01:26 Math / Combinatorics resolved

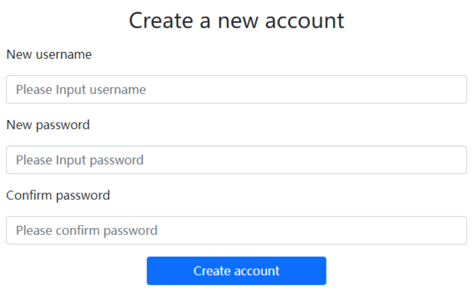
solve x + 1 = 2
Please solve the algebra problem for me!
user3 2021-11-03 05:01:25 Math / Algebra resolved

How to use select in mysql
As title stated, how to use it? I am a rookie of DB sytem.
user1 2021-09-01 15:21:02 Computer Science / Database Systems unsolved

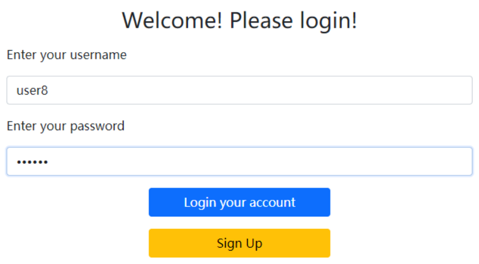
suicide
Some examples for Assisted suicide?
user4 2021-03-15 17:16:20 Law / Assisted suicide unsolved

4.2 User Session Control

In this design, PHP session was used to control the user. And all of user registration, login and logout were supported. The main page of this project was the main.html where the user control dropdown menu implemented. Starting the main page from initialised state, the user state was empty and there were two options on the menu: you can either select to create a new account, or you can login with existing accounts. In this case, we choose to create a new account:

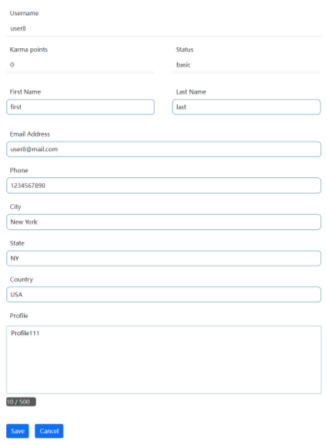


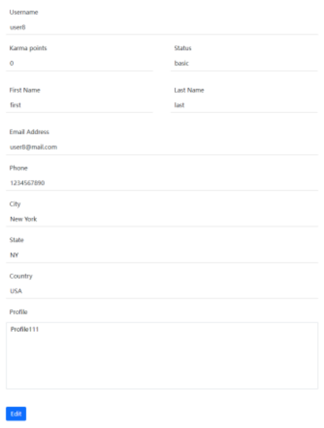
After the new account was create, the system would redirect to the main page, and we can choose to login with the account just created.



After login, the user menu would change to user control. There were three functions supported, user profile, user activity and logout.

First, the user profile, after clicking the profile button, the program would direct the user to the profile page with the current profile, all the information could be inspected here. In this design, user profile editing was also supported, the user was able to edit the first and last name, the email, the phone, and the city, state and country that user was currently in along with the profile information. Note the inspecting other user's profile was also supported but the edit function was disabled. After entering edit profile mode, all the editable components form was highlighted with blue border, the user could enter new information respectively to edit the profile. There were two buttons save and cancel, if the user clicked the save button, all the information the user updated would be updated to the database while when the cancel was clicked, all the information would be reset, and the update of the user's current edit would be aborted. Note that in the user profile page, user could also see their username, karma points as well as their level, from basic to expert. However, they were not be able to edit this information. For the profile field, a limit was also set to limit the profile length, which was 500 words. A word counter was made to monitoring the profile word length when in the editing mode.


Username
user8
Karma points
0
Status
basic
First Name
first
Last Name
last
Email Address
user8@mail.com
Phone
1234567890
City
New York
State
NY
Country
USA
Profile
Profile111
10 / 500


Username
user8
Karma points
0
Status
basic
First Name
first
Last Name
last
Email Address
user8@mail.com
Phone
1234567890
City
New York
State
NY
Country
USA
Profile
Profile111

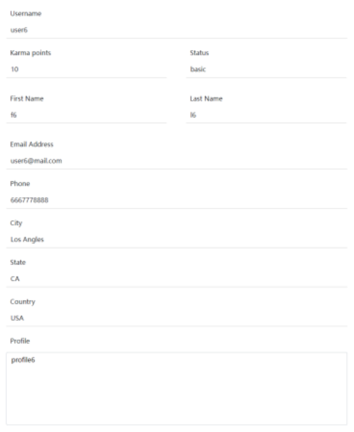
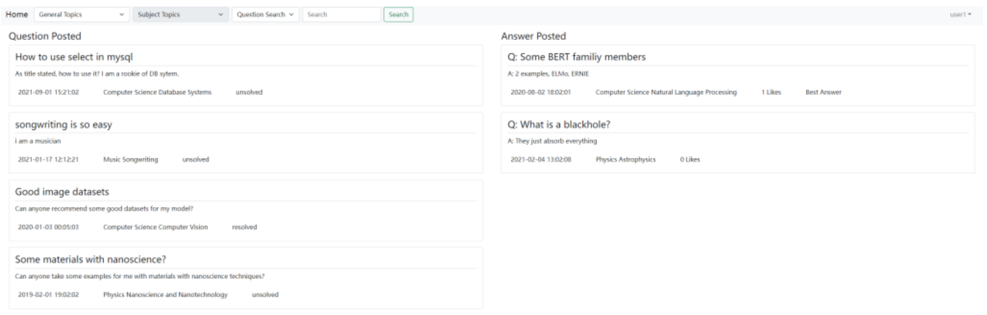

Username
user6
Karma points
10
Status
basic
First Name
f6
Last Name
l6
Email Address
user6@mail.com
Phone
6667778888
City
Los Angles
State
CA
Country
USA
Profile
profile6

Figure 1: In editing mode
Figure 2: Editable profile before editing
Figure 3: Profile view of other users (not editable)

Next, the user activity page. In this page, all the activity the user recently posted would be displayed. There were two parts of this page, the question user posted, and the answer user posted, ordered by time from latest to earliest. User was also able to click the title of each answer and question in this page to enter the question detail. To make the page clear to understand, since changing the answer and question status was not designed to support this page, all the status such as question solved/unsolved were displayed in texture format instead of a button or icon as it was in question detail view.



Question Posted

How to use select in mysql
As title stated, how to use it? I am a rookie of DB sytem.
2021-09-01 15:21:02 Computer Science Database Systems unsolved

songwriting is so easy
i am a musician
2021-01-17 12:12:21 Music Songwriting unsolved

Good image datasets
Can anyone recommend some good datasets for my model?
2020-01-03 00:05:03 Computer Science Computer Vision resolved

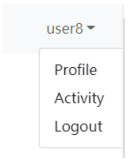
Some materials with nanoscience?
Can anyone take some examples for me with materials with nanoscience techniques?
2019-02-01 19:02:02 Physics Nanoscience and Nanotechnology unsolved

Answer Posted

Q: Some BERT familiy members
A: 2 examples, ELMo, ERNIE
2020-08-02 18:02:01 Computer Science Natural Language Processing 1 Likes Best Answer

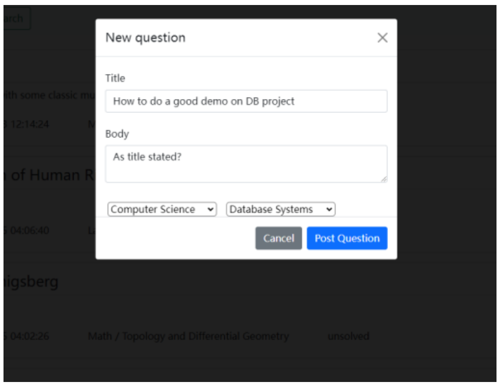
Q: What is a blackhole?
A: They just absorb everything
2021-02-04 13:02:08 Physics Astrophysics 0 Likes

User was also able to logout the account by simply click the logout menu. Then the session would be destroyed, and it would return to main page.



4.3 Posting Control

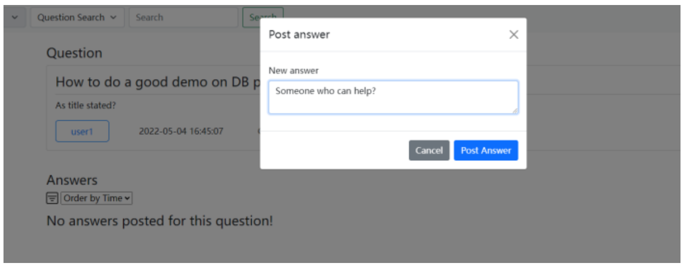
Users were able to post any content they would like to post, either questions or answers. To make the page clear, in this design, a floating button was put and was used to display posting modal. In the main page, this button was used to post new questions while in question details page, this button was used to post new answers for the current question the user was reviewing. In main page, after the button was pressed, a modal should show up:


New question

Title: How to do a good demo on DB project

Body: As title stated?

Computer Science / Database Systems


Question

How to do a good demo on DB p

As title stated?

user1 2022-05-04 16:45:07

Answers

Order by Time

No answers posted for this question!

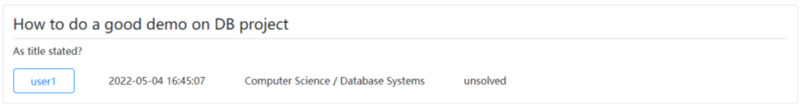
Post answer

New answer: Someone who can help?

Post new question and new answer

In this modal, user was able to input the title of the new question along with the body of this question. Or, in this design, inputting the body was not forced, if the user did not enter the question body, in the preview, the body would be replaced as some hints. For each post action of new questions, user must select an existing topic to post this question. Otherwise an alert about topic selection would show up.

By clicking the blank space, or the cancel button, the use was able to cancel this editing and return to the main page. If the user would like to post the question, then after editing all necessary fields, post button should be used. After the question was posted successfully, the user should receive another alert about the success posting. Then the modal would return to the main page and the user was able to see the new question posted. In this design, user was not able to delete the question as most of the forum did, however, they were able to edit the body of these questions they just posted.


How to do a good demo on DB project

As title stated?

user1 2022-05-04 16:45:07 Computer Science / Database Systems unsolved

Finally we successfully posted a new question

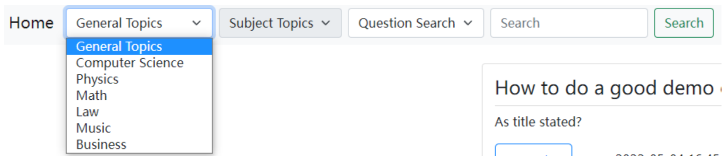
4.4 Query and Filter

In this program, the query and filter function were mainly implemented in the header navbar.

Following fields were supported:

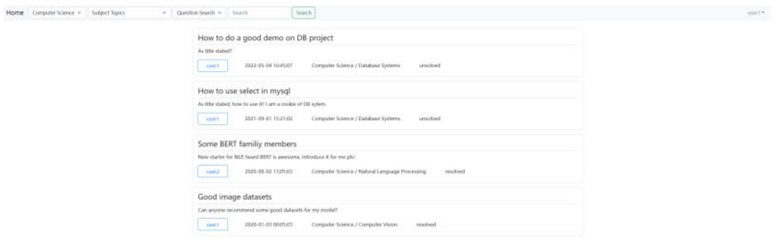
General Topics: By default, the filter would get all the results from all general topics. Each time when the header was ready, the program would obtain the existing general topics from the backend database. And appended them as options in General Topics filter. Here was an example of General Topic Filter.

In this case, we selected the general topics “Computer Science”, after the search button was applied, all question which was under the Computer Science general topics would be fetched ordered by posting / editing time.



How to do a good demo

As title stated?



How to do a good demo on DB project

As title stated?

user1 2022-05-04 16:45:07 Computer Science / Database Systems unsolved

How to use select in mysql

As title stated, how to use it? I am a rookie of DB sytem.

user1 2021-09-01 15:21:02 Computer Science / Database Systems unsolved

Some BERT familiy members

New starter for NLP, heard BERT is awesome, introduce it for me pls!

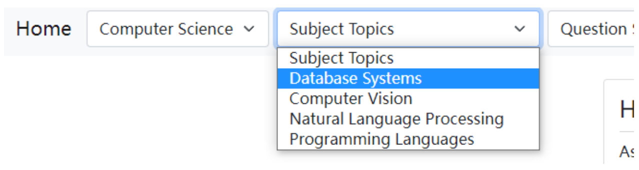
user2 2020-08-02 13:05:03 Computer Science / Natural Language Processing resolved

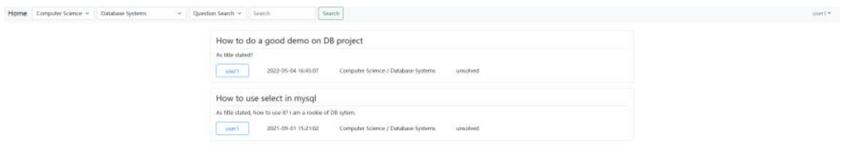
Good image datasets

Can anyone recommend some good datasets for my model?

user1 2020-01-01 00:05:03 Computer Science / Computer Vision resolved

Subject Topics: By default, without a general topic, the subject topic selection would be disabled. After the user selected a general topic, the program would fetch all the subject topics under the selected general topic and appended them as options in this menu. Similar as general topic filter, the user was able to select a subject topic to view all the questions or answers under this subject topic. Here was one example, we selected the subject topic “Database Systems”.





How to do a good demo on DB project

As title stated?

user1 2022-05-04 16:45:07 Computer Science / Database Systems unsolved

How to use select in mysql

As title stated, how to use it? I am a rookie of DB sytem.

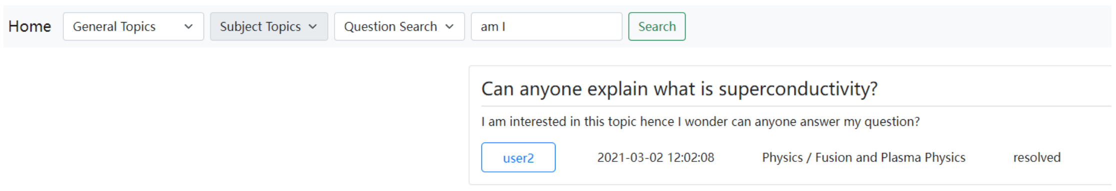
user1 2021-09-01 15:21:02 Computer Science / Database Systems unsolved

Keyword Query: The keyword query took the advantage of both like statement in MySQL to perform accurate query, and the fulltext search by InnoDB. Different weights were applied on both question title and question body, and the answer body. For the question, 0.6 weight for the title where the keyword found by the like statement and 0.4 weight for the body where the keyword found by the like statement. Also, the fuzzy matching was also considered by using fulltext search.

Two levels of query processing were applied. First the program would do a accurate matching with like statement, returning all the questions/answers whose weight was larger than 0, which means the keywork exactly appeared in either title or body or both of them. Then the program would perform a fulltext query on both title and body, then return all the results whose match score calculated by MySQL built-in fulltext search was larger than 0. Then these two levels of query result were combined and ordered from weights by like statement to the score by fulltext search. Take the question query for example, the formula was:

Weights = 0.6 * keyword_exists_title + 0.4 * keyword_exists_body
Score = fulltext_score_title + fulltext_score_body

Results were taken from whose weights > 0 or score > 0, then order by weights desc, score desc, post_time desc.



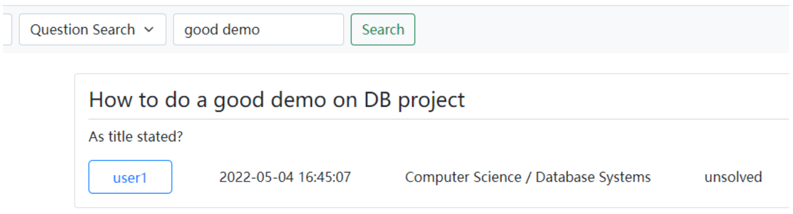
Can anyone explain what is superconductivity?

I am interested in this topic hence I wonder can anyone answer my question?

user2 2021-03-02 12:02:08 Physics / Fusion and Plasma Physics resolved

example of fulltext matching, input "am I", and question contained "I am", matched with fuzzy

With this method, which means if we cannot find a result which 'strictly' contain the keyword, the result may still find the questions/answers who contained these terms in the fulltext. But for ordering, the results which strictly contained the keyword would be preferred.



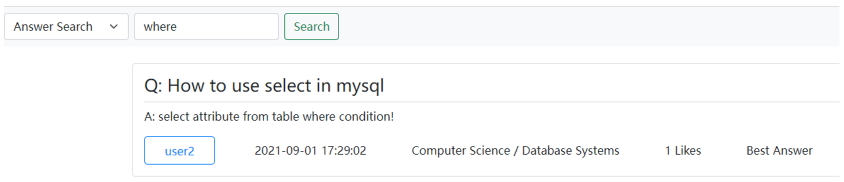
How to do a good demo on DB project

As title stated?

user1 2022-05-04 16:45:07 Computer Science / Database Systems unsolved

Query Type: In this design, both question and answer query were supported. Since the question query was by default, in this section, the answer query would be used. In this case, we would perform answer search.

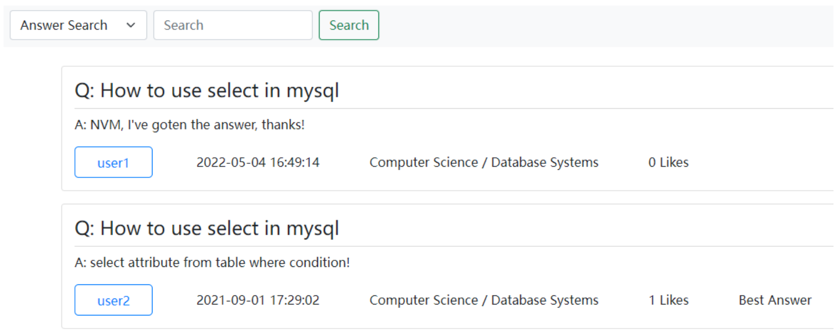
Similar with question search, the result was composed by the title of the question which contained the answer we found, the answer's body, the user who answered this question, the time, the thumb ups this answer received and whether the answer was a best answer.


Answer Search
where
Search

Q: How to use select in mysql

A: select attribute from table where condition!

user2 2021-09-01 17:29:02 Computer Science / Database Systems 1 Likes Best Answer


Answer Search
Search
Search

Q: How to use select in mysql

A: NVM, I've goten the answer, thanks!

user1 2022-05-04 16:49:14 Computer Science / Database Systems 0 Likes

Q: How to use select in mysql

A: select attribute from table where condition!

user2 2021-09-01 17:29:02 Computer Science / Database Systems 1 Likes Best Answer

Figure 1: We query with a keyword
Figure 2: We query without a keyword in Database System subjectopic

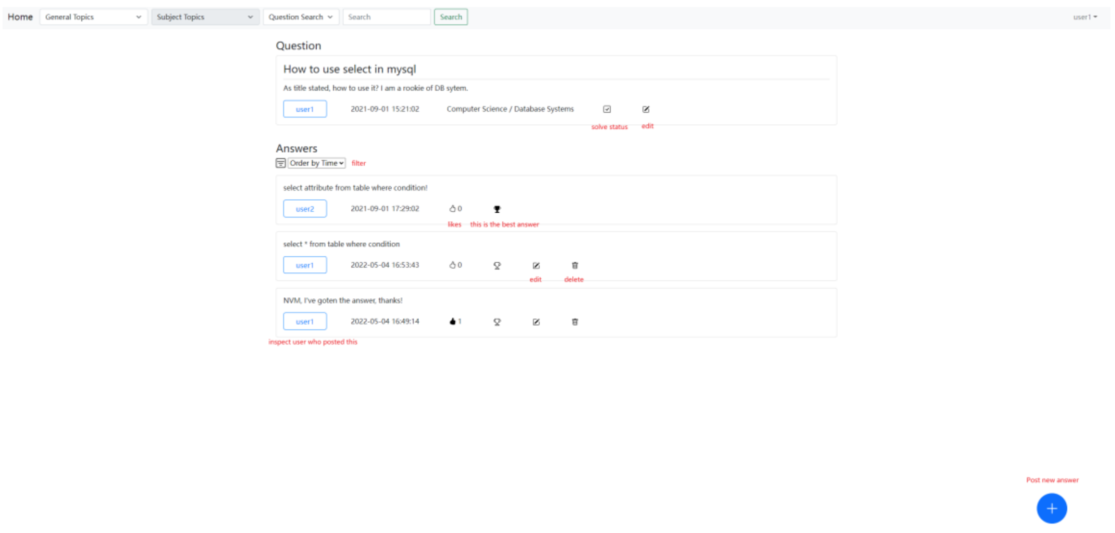
This was how we handle the keyword query, taken question query as example:

```
select qid, q_username, concat(gtname, ' / ', stname) as topics , title, q_body, post_time, status
from(
        select qid, q_username, title, q_body, post_time, match (q_body) against (? with query
expansion) as score, 0.0 as weight, stid, status, q_visible_status from questions
        union
        select qid, q_username, title, q_body, post_time, match (title) against (? with query
expansion) as score, 0.0 as weight, stid, status, q_visible_status from questions
        union
        select qid, q_username, title, q_body, post_time, 0.0 as score, 0.6 as weight, stid, status,
q_visible_status from questions where title like ?
        union
        select qid, q_username, title, q_body, post_time, 0.0 as score, 0.4 as weight, stid, status,
q_visible_status from questions where q_body like ?
        ) as q natural join subjecttopics natural join generaltopics where q_visible_status = 1
and stid = ? and gtid = ?
group by qid, q_username, topics, title, q_body, post_time, status having sum(weight) > 0 or
sum(score) > 0 order by sum(weight) desc, sum(score) desc, post_time desc
```

The clause above was just an example, the detailed clause would also depended on whether the user selected a gtid, stid, or the search type (question search or answer search) they would choose.

4.5 Question Details

By clicking the title of questions or answer preview card, the program would navigate to the question detail page. Two components were included in this page, the question card and the answer card. For the question card, similar as the question preview, but in this case, all the question status such as solved status was represented by button format for the owner of the question control the question information. In this page, the user who post the question would have the permit to edit the question, such as marking the question as solved/unsolved, editing the question body or select a best answer among all the answers. The deletion of the question was not allowed on purpose, since for a forum, deletion of a question would make all the answers under the question invisible, which was not prople normally expected. Hence though we implemented the question deletion, it was disabled.



Question

How to use select in mysql

As title stated, how to use it? I am a rookie of DB sytem.

user1 2021-09-01 15:21:02 Computer Science / Database Systems

Answers

Order by Time

select attribute from table where condition!

user2 2021-09-01 17:29:02 0

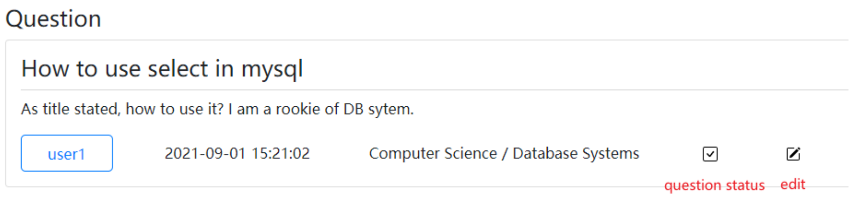
select * from table where condition

user1 2022-05-04 16:53:43 0

NVM, I've goten the answer, thanks!

user1 2022-05-04 16:49:14 1

general design of question details

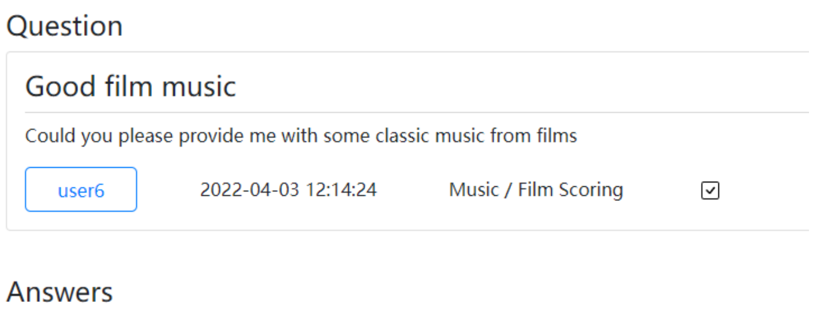


Question

How to use select in mysql

As title stated, how to use it? I am a rookie of DB sytem.

user1 2021-09-01 15:21:02 Computer Science / Database Systems



Question

Good film music

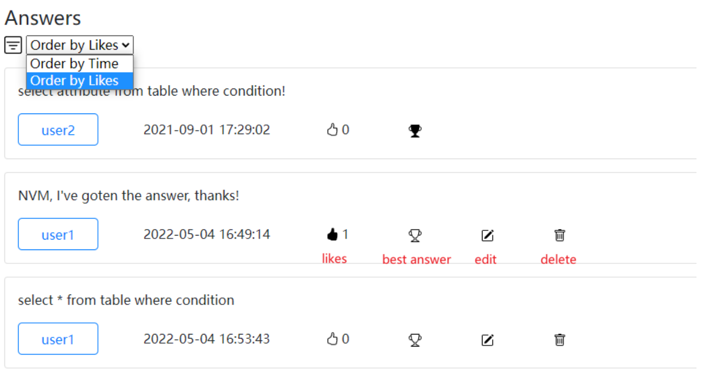
Could you please provide me with some classic music from films

user6 2022-04-03 12:14:24 Music / Film Scoring

Answers

editable question vs un-editable question (currently we were user1)

Above the answer card, a filter was designed, there were two options, users could either filtered the answers by the answer post time, or the number thumb ups this answer received. Since in this design, best answer was also implemented, hence for the answer with best answer tag, it was always in the first position of the answer card list.


Answers

Order by Likes
Order by Time
Order by Likes

select attribute from table where condition!

user2 2021-09-01 17:29:02 0

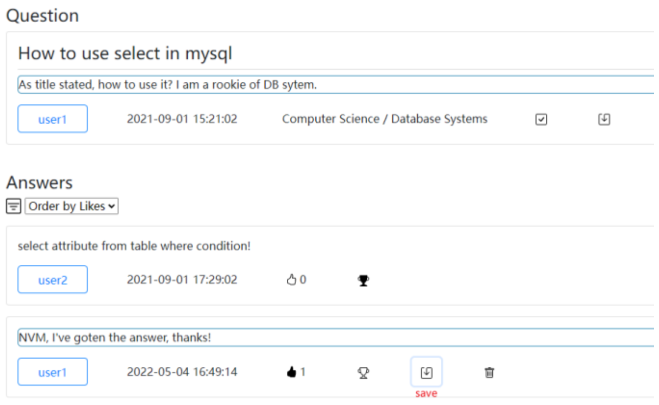
NVM, I've goten the answer, thanks!

user1 2022-05-04 16:49:14 1

select * from table where condition

user1 2022-05-04 16:53:43 0

For the answer card, the user was also able to control the answer card, such as for a login user, a thumb up could be given or the user could retreat the given thumb up. For the answer owner, the editing answer and deletion answer was enabled. The user could either edit the answer body by clicking the edit button or delete this answer by clicking the deletion button. When deletion, a warning would pop up to make confirmation. Once the answer was deleted, it could not be recovered and it would be removed from the user's activity.


Question

How to use select in mysql

As title stated, how to use it? I am a rookie of DB sytem.

user1 2021-09-01 15:21:02 Computer Science / Database Systems

Answers

Order by Likes

select attribute from table where condition!

user2 2021-09-01 17:29:02 0

NVM, I've goten the answer, thanks!

user1 2022-05-04 16:49:14 1

question edit and answer edit

4.6 Bookmark Support

All the websites supported bookmark function to save the current query/filter status. Here was all the parameter for each webpage:

main.html

parameter	valid values	description
gtid	existing gtid	general topic id
stid	existing stid	subject topic id
key	any keyword on search	keyword the user would like to search
stype	as or qs	search type, question and answer were supported

profile.html:

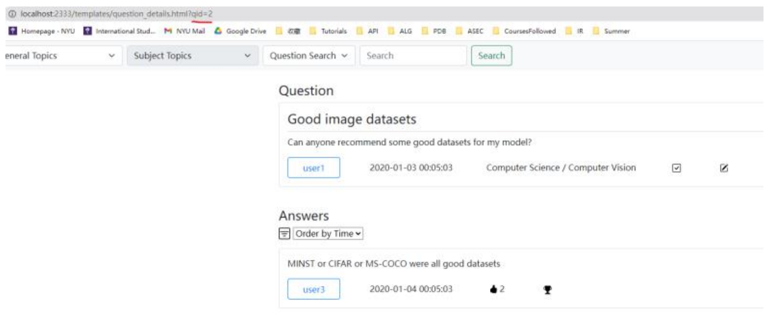
parameter	valid values	description
p_user	username	the username want to inspect

question_details.html

parameter	valid values	description
qid	existing qid	question id

Here was an example:

In question_details.html, we bookmark the qid=2, we enter the bookmark and the status would show:



localhost:2333/templates/question_details.html?qid=2

Question

Good image datasets

Can anyone recommend some good datasets for my model?

user1 2020-01-03 00:05:03 Computer Science / Computer Vision

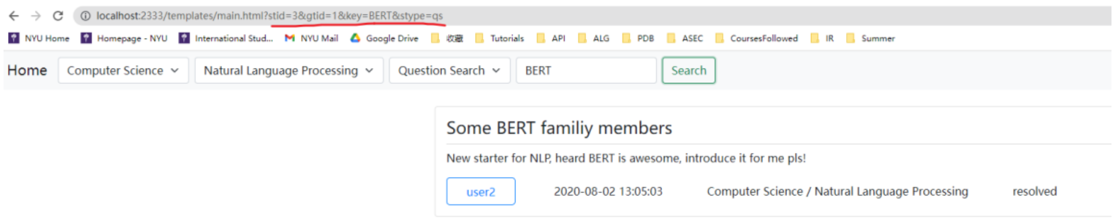
Answers

Order by Time

MINST or CIFAR or MS-COCO were all good datasets

user3 2020-01-04 00:05:03 2

In main.html, we search “BERT keyword under Computer Science / Natural Language Processing, we enter the bookmark and the page would show:

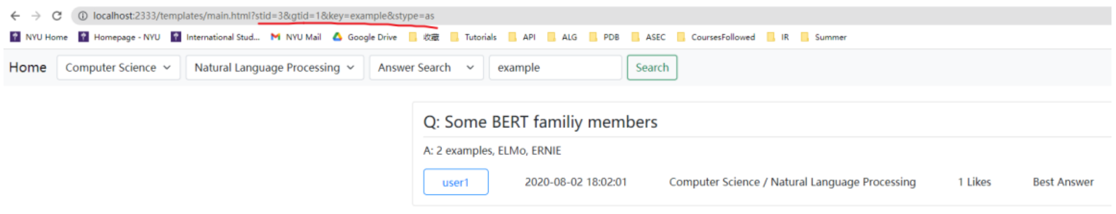


localhost:2333/templates/main.html?stId=3&gtId=1&key=BERT&stype=qs

Some BERT familiy members

New starter for NLP, heard BERT is awesome, introduce it for me pls!

user2 2020-08-02 13:05:03 Computer Science / Natural Language Processing resolved



localhost:2333/templates/main.html?stId=3&gtId=1&key=example&stype=as

Q: Some BERT familiy members

A: 2 examples, ELMo, ERNIE

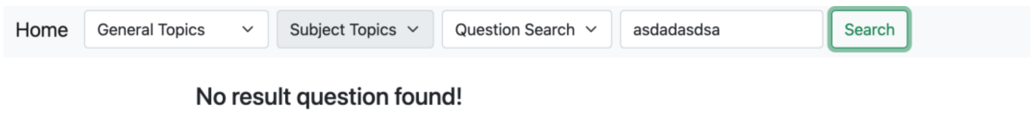
user1 2020-08-02 18:02:01 Computer Science / Natural Language Processing 1 Likes Best Answer

Example of bookmark support for both answer query and question query

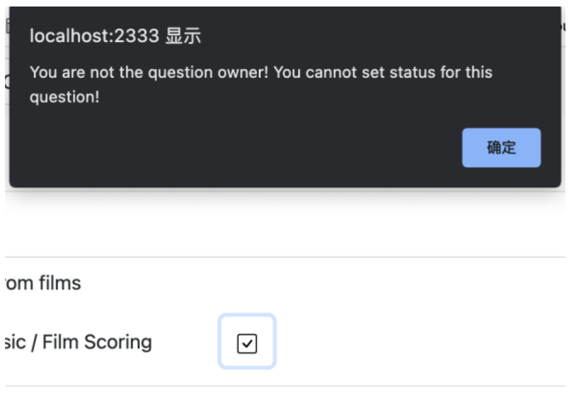
4.7 Exception Handling

Some of the cases were considered. Which was listed below:

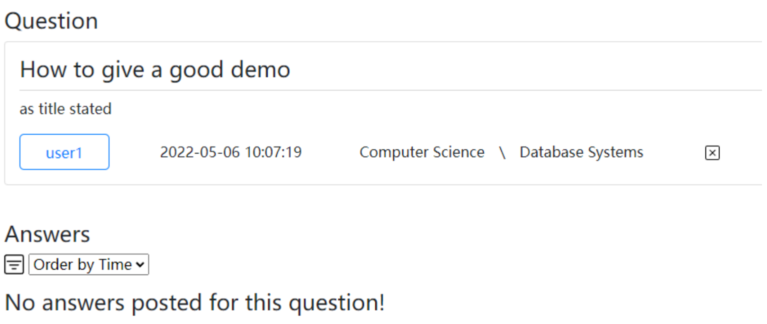
- If the keyword query does not find any result, the main page would show: No Result Found
- When creating a new account, if the username already exists, there would be an alert to inform the user about the existence of the username and asked the user to change one
- In login page, if the user input a username which does not exist, or the password does not match the existing username, an alert would show up
- If the user has not login, for each operation which requires login such as posting an answer or a question, like an answer, trying to control a question, an alert about login would show up
- If the user who is not the owner of a current question / answer, if they were trying to control this posting, the system would give an alert to inform the user to prevent the control.
- For posting a question, if the user did not select a general topic, or subject topic, an alert would show up.
- For question which did not have a single answer, in the answer card list it would show something like "No answer for this question"



No result question found!


localhost:2333 显示

You are not the question owner! You cannot set status for this question!


Question

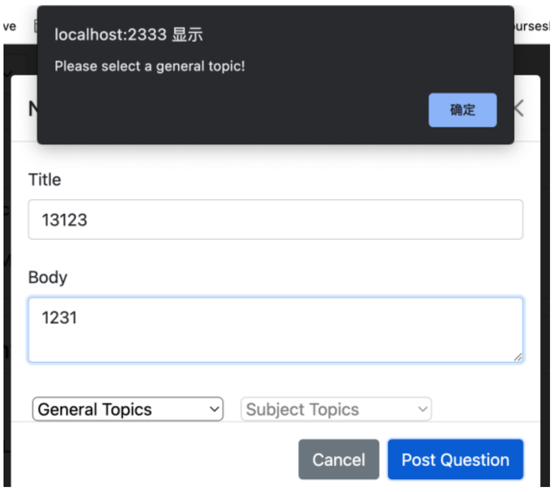
How to give a good demo

as title stated

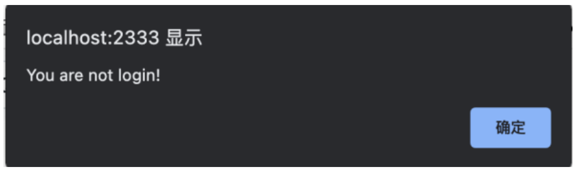
user1 2022-05-06 10:07:19 Computer Science \ Database Systems

Answers

No answers posted for this question!


localhost:2333 显示

Please select a general topic!


localhost:2333 显示

You are not login!

Some examples of exception handling, note that there were more details, these are just examples

4.8 Some other SQL clauses

Here were some other SQL clauses used in this project. To make it simple, only the core part of the clause would recorded below, removing the procedure part since we would introduce them in the next section and the clauses below were only a part of all the clauses which was typical which would make the project clearer:

Delete Answer:
update Answers set a_visible_status = 0 where aid = ?

Edit Answer (Similar logic as Question):

update Answers set a_body = ?, answer_time = ? where aid = ?

Get Answer in Activity (similar logic as Question):

select qid, concat(gtname, ' \ ', stname) as topic, title, a_body, answer_time, thumb_ups, best_answer from answers natural join questions natural join subjecttopics natural join generaltopics where a_username = ? and a_visible_status = 1 order by answer_time

Check the existence of Likes for current user:

select exists(select * from likes where username=? and aid=? and like_status = 1)

Get Subject Topics (Similar logic as General Topics):

```
select stid, stname from subjecttopics where gtid = ?
```

Create new account:

```
insert into Users (username, password) values (?, ?)
```

Login:

```
select username from Users where username = ? and password = ?
```

Update Best Answer:

```
select aid from Answers natural join Questions where qid = ? and best_answer = 1
update Answers set best_answer = 0 where aid = ?
update Answers set best_answer = 1 where aid = ?
```

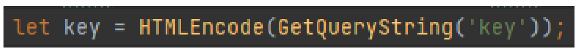
Post new Question (Similar Logic as Answer):

```
insert into Questions (q_username, stid, title, q_body) values (?, ?, ?, ?)
```

5 Security

Security of the program was vital. In this design, two facets were mainly focused. Cress Site Scripting (XSS) and SQL Injection. Basic implementation to avoid these two types of attack was applied.

For XSS attack, for each get request, the escape was used such as the symbol '<' would be translated to &lt, and '>' would be translate to &gt. Since for the content of question detail, profile and activities, GET method was used to support the bookmark of the browser, the parameter used for these sites should be protected. Here was an example of the application of XSS protection in the program

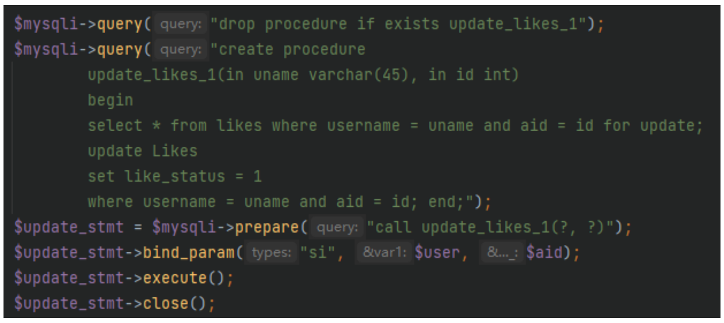


```
let key = HTMLEncode(GetQueryString('key'));
```

For SQL injection, for each API provided by the PHP script in this program, instead of fetching the result directly, prepare was used to avoid SQL injection. With prepare statement, the system would replace all the quote symbol with escaped version, hence the program would be able to avoid SQL injection.

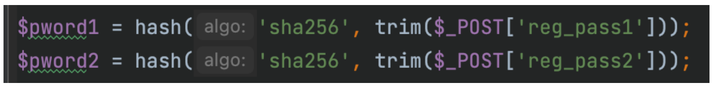
Also, to protect the program for concurrency, pessimistic lock was used for every update statement

in the backend in PHP. For example, in out edit profile, the sql clause would look like this:
update Likes set like_status = 1 where username = ? and aid = ?;
To execute this in PHP, procedure was used.
The question symbol was used to denote the value being inserted. Before this statement, a lock statement could be executed:
select * from likes where username = ? and aid = ? for update;
In which case, the Users table was locked when inserting the new data, hence to protect the concurrency with pessimistic lock in MySQL;



```
$mysqli->query( query: "drop procedure if exists update_likes_1");
$mysqli->query( query: "create procedure
        update_likes_1(in uname varchar(45), in id int)
        begin
        select * from likes where username = uname and aid = id for update;
        update Likes
        set like_status = 1
        where username = uname and aid = id; end;");
$update_stmt = $mysqli->prepare( query: "call update_likes_1(?, ?)");
$update_stmt->bind_param( types: "si",  &var1: $user,  &..._: $aid);
$update_stmt->execute();
$update_stmt->close();
```

For password storage, the passwords were not stored in the database in plain text, instead, they were stored with hashing SHA256 when the account were created.



```
$pword1 = hash( algo: 'sha256', trim($_POST['reg_pass1']));
$pword2 = hash( algo: 'sha256', trim($_POST['reg_pass2']));
```

6 Others

This project was implemented with PHP 8.1 with MySQL 8.0. Also, bootstrap-5.1.3, jquery-3.6.0 and bootstrap-icons-1.8.1 were also used as framework. It was noticed that for these libraries, there were enormous number of resource files such as icon .svg files or some .css files. Hence for the submission on the GradeScope, only our .js, .php, .css and .html files were uploaded due to the limitation of GradeScope zip file uploading. To access to the full project, we’ve made our remote repository public, which was:

https://github.com/NickNameInvalid/pdb_project