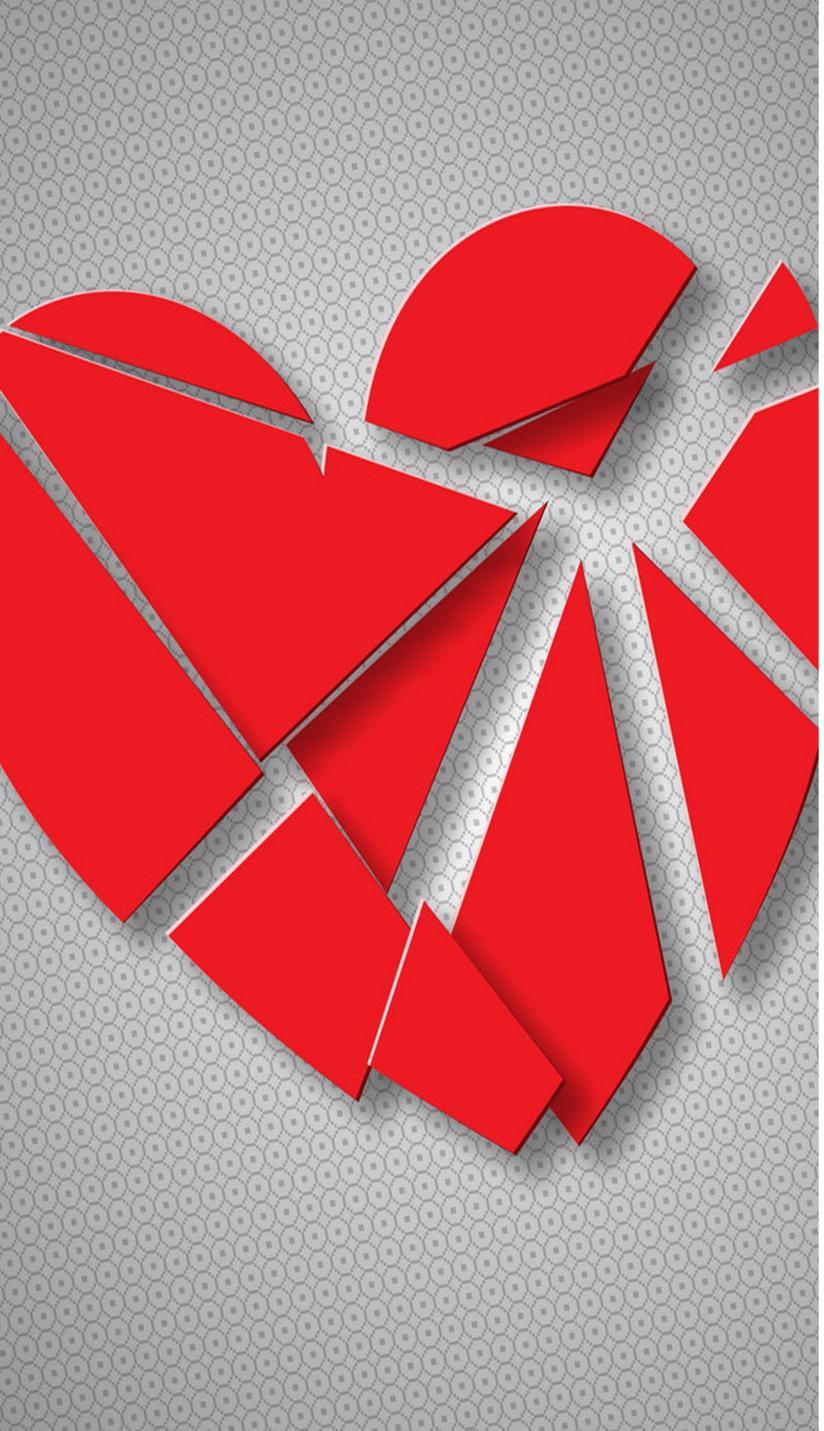


Lawrence Castillo

CS435 Term Project

Matchmaking Application

LIKE
LIKE
LOVE

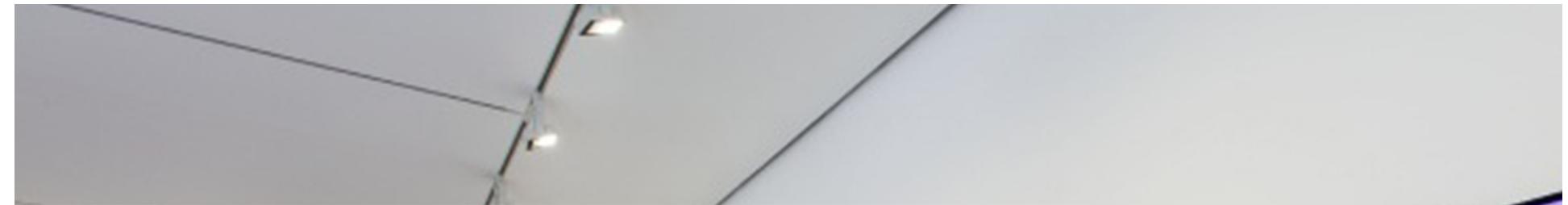


The Problem

Popular match-making site, *Like Like Love*, has been riding the high of its refreshing take on dating – or rather blind-dating -- to a boom in user activity. The service asks users to complete a questionnaire and connects people based on percentage of shared answers.

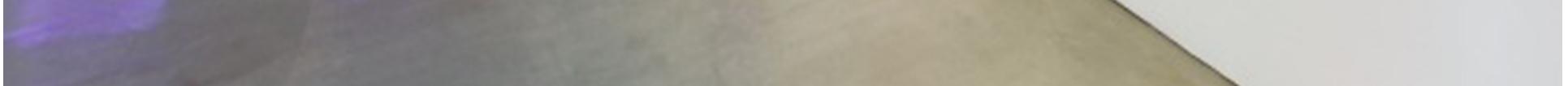
Recently, however, the site has seen a sharp decline in usage as existing patrons have been reporting a lack of depth in their matches. For the site to continue operating, user confidence in the site must be restored.

Business problem: revamp existing match algorithm by creating a system for pairing-up users that looks deeper into data relationships to create more likely matches, restore patron confidence, and renew interest.



Solution

Essentially, the aforementioned application, *Like Like Love*, is an amalgamation of various matching sites I have experienced in my life. My solution would be to re-imagine the generally straight-forward data analysis (sharing similar “likes”) of many matching sites with a more complex set of rotating binary questions (a or b, or “don’t care”), using that acquired data to generate a likely type (and likelihood of the type match), then matching compatible types based on user preference, location, and likelihood assessment. Established personality research – “The Four Tendencies” by Gretchen Rubin – will form the backbone of the matching system. Users will answer questions and have their type revealed along with compatible types and 1 user fitting that description.

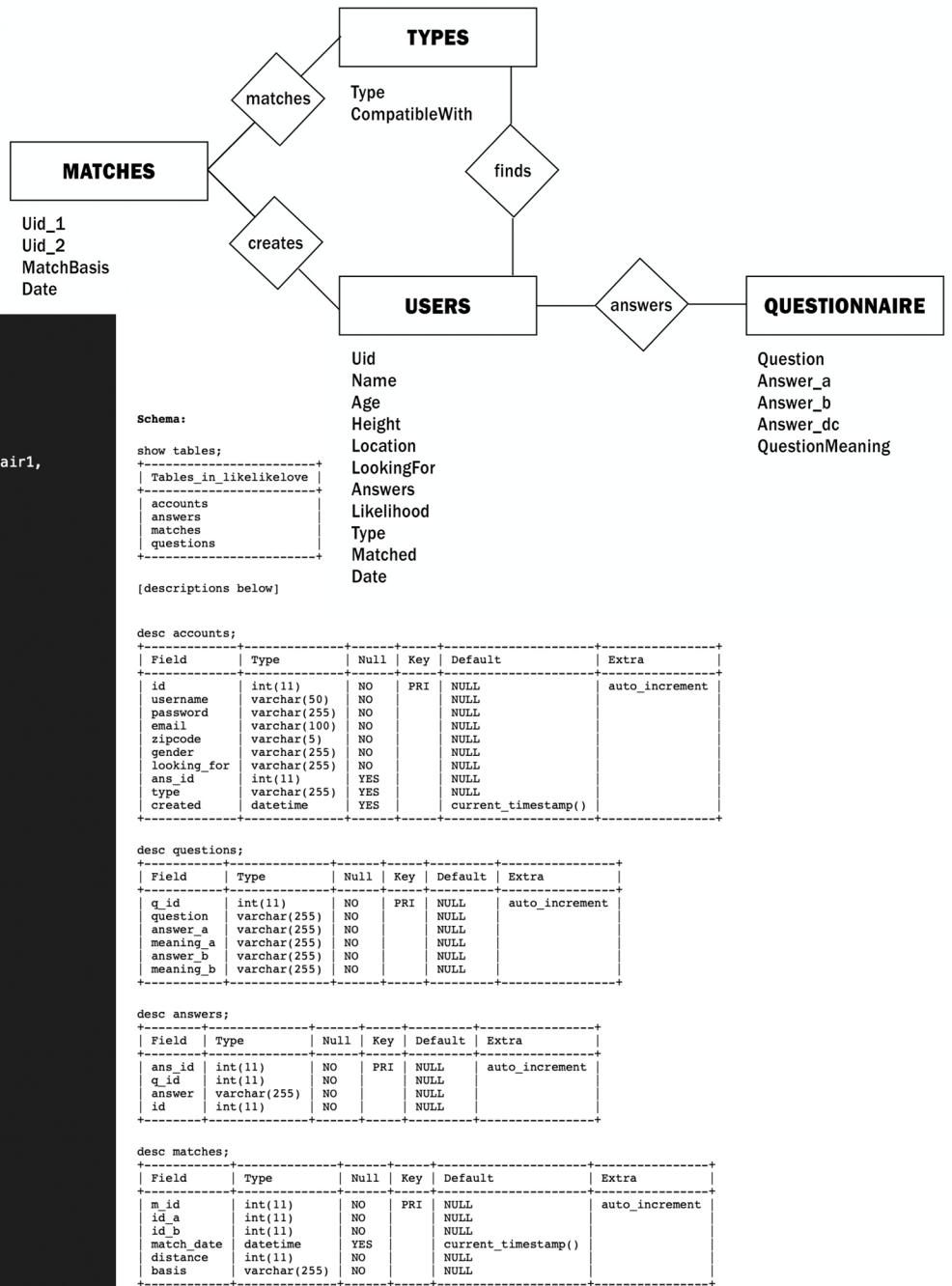


Discovery

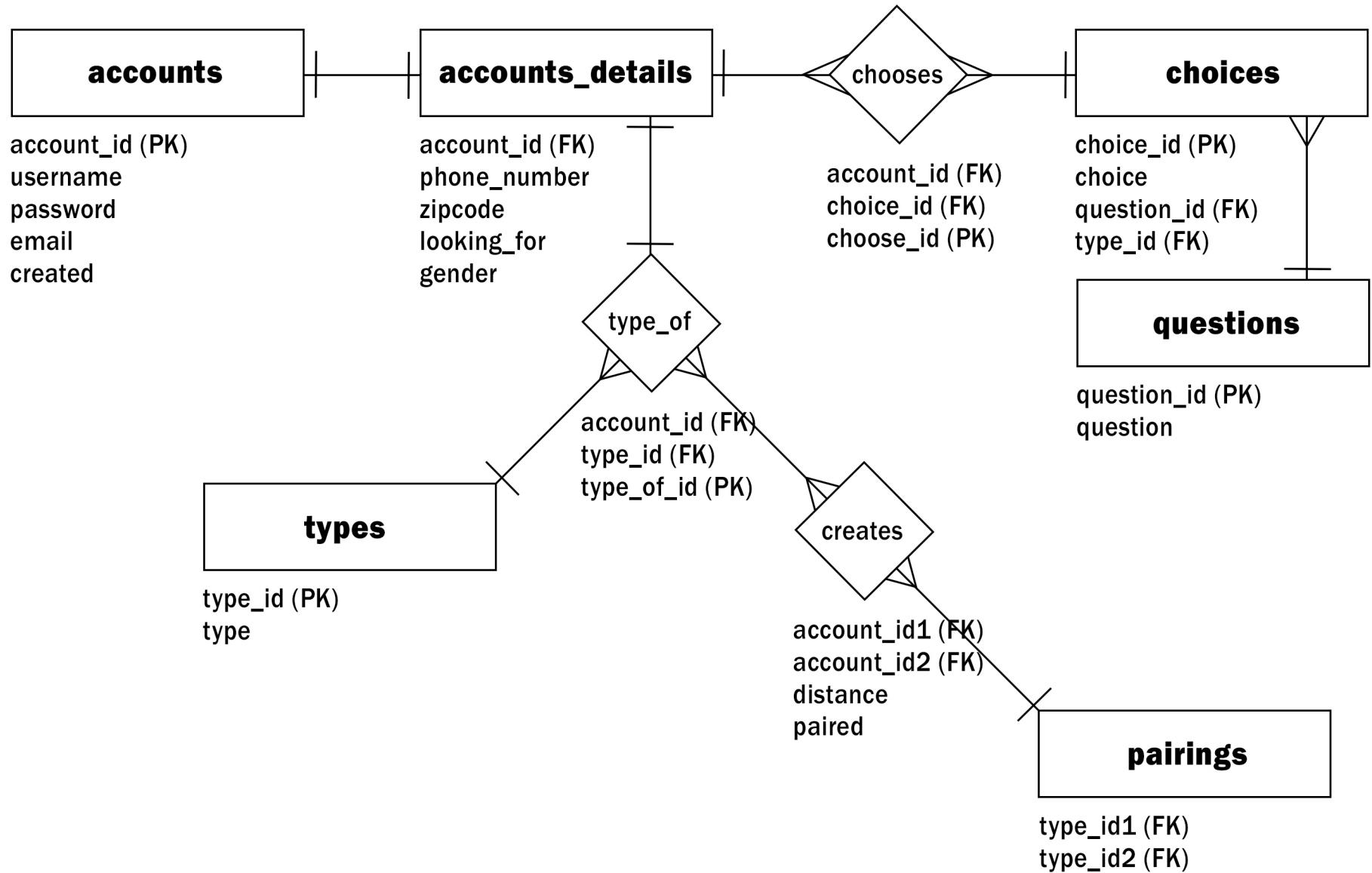
```
## What my answers say about me
SELECT c.question, c.choice, types.type
FROM chooses
JOIN
(SELECT questions.question question, choices.choice choice, choices.choice_id choice_id, choices.type
  FROM choices
  JOIN
    questions ON questions.question_id=choices.question_id) c
ON chooses.choice_id=c.choice_id
JOIN
  types ON types.type_id=c.type_id
WHERE account_id=3;

## Retrieve profile data
SELECT auth.email, det.phone_number, det.zipcode, det.looking_for, t.type, min(pairings.type_id2) type_pair1,
max(pairings.type_id2) type_pair2
FROM accounts auth
JOIN
  accounts_details det ON auth.account_id=det.account_id
LEFT JOIN
  (SELECT type_of.account_id account_id, type_of.type_id, types.type type
  FROM type_of
  JOIN
    types ON types.type_id=type_of.type_id) t
  ON t.account_id=auth.account_id
LEFT JOIN
  pairings ON pairings.type_id1=t.type_id
WHERE auth.account_id = 5;

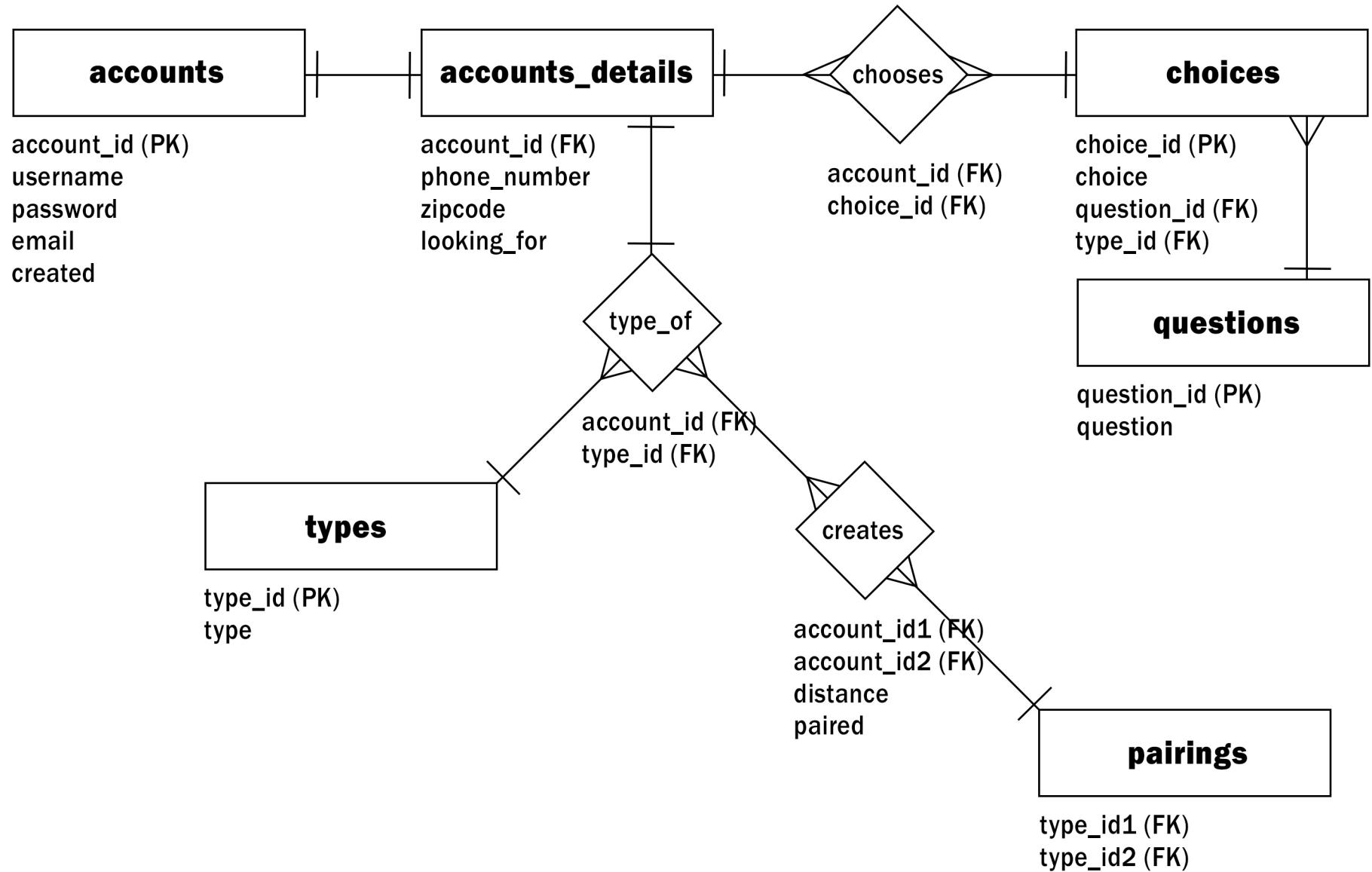
## Retrieve profile data with match types
SELECT auth.email, det.phone_number, det.zipcode, det.looking_for, t.type, p.type_pair1, p.type_pair2
FROM accounts auth
JOIN
  accounts_details det ON auth.account_id=det.account_id
JOIN
  (SELECT type_of.account_id account_id, type_of.type_id type_id, types.type type
  FROM type_of
  JOIN
    types ON types.type_id=type_of.type_id) t
  ON t.account_id=auth.account_id
JOIN (
  SELECT pair1.type_id type_id, pair1.type1 type_pair1, pair2.type2 type_pair2
  FROM pairings
  SELECT a.type_id1 type_id, a.type_pair1 type_pair1, types.type type1
  FROM types
  JOIN (
    SELECT type_id1, max(type_id2) type_pair1
    FROM pairings
    WHERE type_id1=4) a
  ON types.type_id=a.type_pair1) pair1
JOIN (
  SELECT a.type_id1 type_id, a.type_pair2 type_pair2, types.type type2
  FROM types
  JOIN (
    SELECT type_id1, min(type_id2) type_pair2
    FROM pairings
    WHERE type_id1=4) a
  ON types.type_id=a.type_pair2) pair2
ON pair1.type_id=pair2.type_id
  WHERE auth.account_id = 3;
```



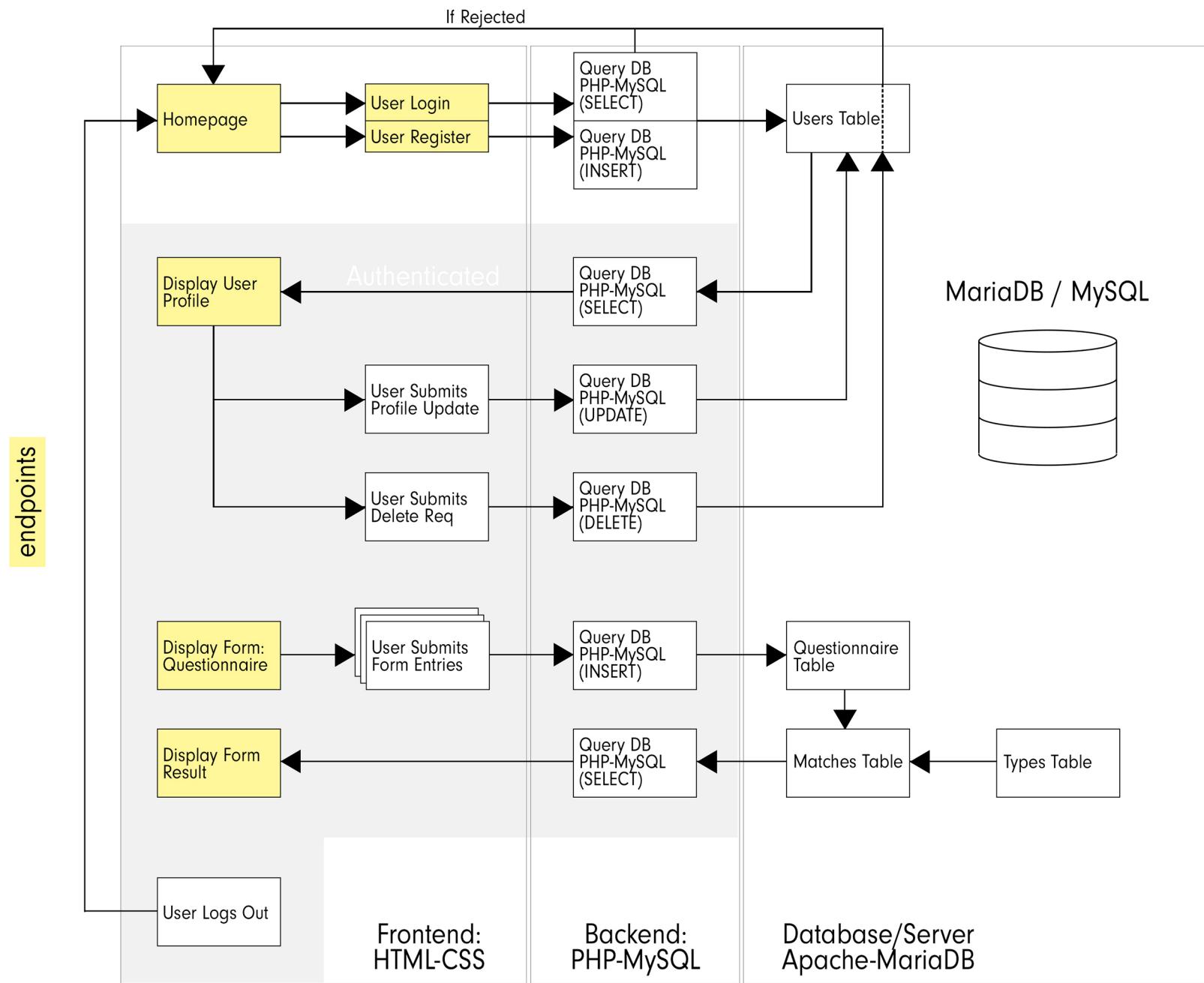
ER Diagram



ER Diagram



Flow



Implementation

phpMyAdmin

Recent Favorites

Server: localhost:3306 » Database: likelikelove » Table: accounts

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	account_id	int(11)	utf8_general_ci		No	None		AUTO_INCREMENT	Change Drop More
2	username	varchar(50)	utf8_general_ci		No	None			Change Drop More
3	password	varchar(255)	utf8_general_ci		No	None			Change Drop More
4	email	varchar(100)	utf8_general_ci		No	None			Change Drop More
5	created	datetime			No	current_timestamp()			Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index

Print Propose table structure Move columns Improve table structure

Add 1 column(s) after created Go

+ Indexes

Partitions

No partitioning defined!

Information

Table comments:

Console Space usage Row statistics



UI Skeleton

A UI skeleton for a personality quiz question. The top navigation bar has three dots. The title "PERSONALITYQUIZ" is centered above the question. The question text "I am more likely to break a commitment to:" is followed by three rounded rectangular buttons: "Myself", "Others", and "Skip". Below the buttons are two empty rounded rectangles, one black and one red.

PERSONALITYQUIZ

I am more likely to break a commitment to:

Myself

Others

Skip

Actual Product

LIKE
LIKE
LOVE

REGISTER

LOGIN

PROFILE

TAKE THE
QUIZ!

LOGOUT

Member Login

Username

Password

Login