Phase 2 Abstract Code w/SQL

CS 6400 - Summer 2020

Team 012
Jiatang Dong (jdong86@gatech.edu)
Boyan Lu (blu71@gatech.edu)
Ben Li (bli417@gatech.edu)
Pei Tang (ptang39@gatech.edu)

Table of Contents

Login

Abstract Code

- User enters *email* ($Email), *password* ($Password) input fields.
- If data validation is successful for both *email* and *password* input fields, then:
    - When ***Enter*** button is clicked:

```
SELECT password FROM `User` WHERE email= '$Email';
```

        - If result is not empty:
            - If Hash(User.password) = Hash('$Password'):
                - Store login information as session variable '$UserSession'.
                - Successfully login, jump to **Dashboard**
            - Else:
                - Go back to **Login** form
                - Error message: Wrong password
        - Else:
            - Go back to **Login** form
            - Error message: Not a registered User
- Else *email* and *password* input fields are invalid, display **Login** form, with error message.

Dashboard

Abstract Code

- Extract the $OrderAttribute from request, otherwise set it to 'surrender_date'
- Fetch all the dogs that's not associated with AdoptionApplication and store it in '$Dogs'

```
SELECT D.dogID, name, sex, altered, birth_date, description,
is_animal_control_surrender, surrender_date, surrender_reason,
microchipID, created_by
FROM `Dog` D LEFT JOIN `AdoptionApplication` A
ON D.dogID = A.dogID
WHERE A.application_number IS NULL
ORDER BY $OrderAttribute
```

- For each $Dog in the '$Dogs':
    - Render the $Dog.name, $Dog.breed, $Dog.sex, $Dog.altered, $Dog.birth_date (convert it into age in month) and adoptability status (derive from $Dog.altered and $Dog.microchipID)
    - Render the ***Detail*** button
    - Jump to **Dog Detail View** interface when click the ***dog detail*** button
- Render the current capacity (SHELTER_CAPACITY(15) - $Dogs.count)
- If the $Dogs.count < SHELTER_CAPACITY (15):

        - Print the ***Add dog*** button
            - Jump to **Dog** form When click the button
- Render ***Add Application*** button
    - Jump to **Adoption Application** form when click the Button
- If User.is_admin:
    - Render ***Review Applications*** Button
        - Jump to **Pending Application View** when click the button
- Render ***Report*** Button
    - Jump to **Reports** when click the button

Add Dog

Abstract Code

- User enters each attribute of $Dog
- If data validation is successful for $Dog
    - When **Save** button is clicked
        - Get the total amount of dog '$DogCount' in shelter

```
SELECT count(*)
FROM `Dog` D LEFT JOIN `AdoptionApplication` A
ON D.dogID = A.dogID
WHERE ISNULL(A.application_number)
```

        - If the '$DogCount' >= SHELTER_CAPACITY (15):
            - Reject
        - Insert the '$Dog' into `Dog`

```
INSERT INTO `Dog`
(name, sex, altered, birth_date, description,
is_animal_control_surrender, surrender_date,
surrender_reason, microchipID, created_by)
VALUES
( '$Dog.name', '$Dog.sex', '$Dog.altered', '$Dog.birth_date',
'$Dog.description', '$Dog.is_animal_control_surrender',
'$Dog.surrender_date', '$surrender_reason', '$Dog.microchipID',
'$UserSession.email')
```

        - Return to **Dashboard**

Update Dog

Abstract Code

- User enters each attribute of $Dog
- If data validation is successful for $Dog

- When **Save** button is clicked
  - Update the information from `Dog` where Dog.dogID = '$Dog.dogID'

```
UPDATE  `Dog` SET
name = '$Dog.name',
sex='$Dog.sex',
altered='$Dog.altered',
birth_date='$Dog.birth_date',
description='$Dog.description',
is_animal_control_surrender='$Dog.is_animal_control_surrender',
surrender_date='$Dog.surrender_date',
surrender_reason='$Dog.surrender_reason',
microchipID='$Dog.microchipID'
WHERE dogID = $Dog.dogID
```

  - Delete the dog's breeds from `Belong` where Belong.dogID = '$DogID'

```
DELETE FROM Belong WHERE dogID = $Dog.dogID
```

  - For each item '$breedID' in the $Dog.breed list
    - Insert the item $breedID' into to `Belong`

```
INSERT INTO  `Belong` VALUES ($DogID, $breedID)
```

- Return to **Dog Detail View** for the updated dog

View Dog Detail

Abstract Code

- User click the ***Detail*** Button from each dog item in **Dashboard**
- '$DogID' is passed to here
- Fetch the information from `Dog` where Dog.dogID = '$DogID'

```
SELECT dogID, name, sex, altered, birth_date, description,
is_animal_control_surrender, surrender_date, surrender_reason,
microchipID, created_by
FROM `Dog` WHERE dogID = $DogID
```

- If '$DogID' doesn't exist:
  - Return error: dogID doesn't exist
- Fetch the dog's breeds from `Belong` where Belong.dogID = '$DogID'

```
SELECT BR.name FROM `Belong` BE
LEFT JOIN Breed BR ON BE.breedID = BR.breedID
WHERE dogID = $DogID
```

- Fetch the dog's expense from `Expense` where Expense.dogID = '$DogID'

```
SELECT dogID, expense_date, vendor, amount, description
```

```
FROM `Expense` WHERE dogID = '$DogID'
```

- Render ***Edit*** Button
    - Jump to **Dog** form when the button is clicked
- Render ***Adopt*** Button
    - Jump to **Adoption** form when the button is clicked
- Render the dog information
- Render ***Add Expense*** button
    - Jump to **Expense** form when the button is clicked
- Render expenses in list

Add Dog Expense

Abstract Code

- User click ***Add Expense*** from **Dog Detail View**
- $DogID are passed to here
- User enter the attributes of $Expense
    - When click the ***save*** button
    - If the expense is not valid:
        - Reject
    - Add the expense to database

```
INSERT INTO `Expense`
(dogID, vendor, expense_date, amount, description)
VALUES
('$DogID', '$Expense.vendor', '$Expense.expense_date, '$Expense.amount',
'$Expense.description')
```

Add Adoption Application

- User clicks on ***Add Adoption Application*** button from **Dashboard**
- User enters adopter's *email* input fields
- When ***enter*** button is clicked:

```
SELECT email from `Adopter` WHERE email = '$Email';
```

    - If *email* ('$Email') exists:
        - User enters *apply_date* ('$ApplyDate') input fields
        - User enters *coApp_first_name* and *coApp_last_name* input fields

```
INSERT INTO `AdoptionApplication`
(apply_date, coApp_first_name, coApp_last_name, approval_state,
adopter_email)
```

```
VALUES
('$ApplyDate', '$CoAppFirstName', '$CoAppLastName', '$Approval_State',
'$Email');
```

- User clicks on ***submit*** button, display application_number ('$ApplicationNumber')

- Else:
  - User enters adopter's *first_name* and *last_name* input fields
  - User enters adopter's *street*, *city*, *state*, *zip_code* input fields
  - User enters adopter's *cell_phone* input fields
  - User enters *apply_date* input fields
  - User enters *coApp_first_name* and *coApp_last_name* input fields

```
INSERT INTO `Adopter` (email, last_name, first_name, cell_phone, street,
city, state, zip_code)
VALUES ('$Email', '$LastName', '$FirstName', '$CellPhone', '$Street', '$City',
'$State', '$ZipCode');
INSERT INTO `AdoptionApplication` (apply_date, coApp_first_name,
coApp_last_name, approval_state, adopter_email)
VALUES ('$ApplyDate', '$CoAppFirstName', '$CoAppLastName',
'$Approval_State', '$Email');
```

- User clicks on ***submit*** button, display application_number ('$ApplicationNumber')

Update Adoption Application

Admin clicked on ***Adoption Application Review*** button from **Dashboard**

- Find Adoption Application where AdoptionApplication.approval_state = "pending approval", Sorted by AdoptionApplication.application_number and store in $applications

```
SELECT application_number, coApp_first_name, coApp_last_name, adopter_email,
first_name, last_name, cell_phone, street, city, state, zip_code
FROM `AdoptionApplication`
JOIN `Adopter` ON Adopter.email = AdoptionApplication.adopter_email
WHERE approval_state = 'pending approval'
ORDER BY application_number ASC;
```

- For each $application in the $applications:
  Render $application attributes and ***approve/reject*** button
  - If Admin click ***approve*** button:
    - Set '$application.approval_state' to be "approved"

```
UPDATE `AdoptionApplication`
SET approval_state = 'approved'
WHERE application_number = '$application.application_number';
```

- Elif Admin click ***reject*** button:
  - Set '$application.approval_state' to be “rejected”

```
UPDATE `AdoptionApplication`
SET approval_state = 'rejected'
WHERE application_number = '$application.application_number';
```

Search Adopter

Abstract Code

- If Admin clicked on ***add adoption*** button from **Dog Detail View:**
  - Render **associate** form with '$DogID' passed to here
- Else: Render **associate** form with '$DogID' as NULL
- Admin enters *last_name*('$LastName') into the input field
- If text validation is successful for adopter and co-applicant on last name, then:
  - When ***Search*** button is clicked:
    - If any beginning or ending letter(s) Adopter.last_name or AdoptionApplication.coApp_last_name matches with '$LastName' and AdoptionApplication.approval_state == “approved”:
      - For each adopter:
        - Display adopter(s) with contact information, co-applicant last name only if matched, and approval state

```
SELECT email, first_name, last_name, coApp_last_name, street, city, state, zip_code,
cell_phone, approval_state
FROM  `Adopter` A
LEFT JOIN `AdoptionApplication` AP ON A.email = AP.adopter_email
WHERE AP.approval_state = 'approved'
          AND (LOWER(a.last_name) LIKE LOWER('%$LastName%')
          OR LOWER(ap.coApp_last_name) LIKE LOWER('%$LastName%'))
GROUP BY email, first_name, last_name,  coApp_last_name,  street, city, state, zip_code,
cell_phone, approval_state
ORDER By last_name, first_name ASC;
```

  - If any case insensitive letters in (‘$CoAppLastName’) contains case insensitive letter(s) input of ('%$LastName%')
    - Show co-applicant last name column

- Else Hide co-applicant last name column
- Else display “No match”

- Else *last_name* input field is invalid, display error message
  - Admin can filter and sort by adopter/co-applicant name, address, and phone
- If '$DogID' is NOT NULL:
  - **Email** link is active:
    - When admin click on **Email** link:
    - Select and display approved application with max of application date and exclude post associated application(s)

```
SELECT adopter_email,application_number, apply_date, coApp_first_name, coApp_last_name,
approval_state, dogID, adopted_date, fee
FROM `AdoptionApplication` AP
WHERE AP.adopter_email = '$Email'
AND AP.approval_state = 'approved'
AND AP.apply_date = (SELECT MAX(AP1.apply_date) FROM `AdoptionApplication` AP1
WHERE AP.adopter_email=AP1.adopter_email)
AND AP.adopted_date IS NULL
ORDER BY adopter_email, application_number ASC;
```

- Render **associate** form with adoption application details

- Else **Email** link is inactive

Associate Dog

Abstract Code

- Admin clicked on **Email** link:
  - '$DogID' is passed here
  - '$ApplicationNumber' is passed here
  - Render Adoption Fee with $0.00 on the **associate** form
- Select and sum all '$expenses' from the `Expense` table of that '$DogID'
  - If Dog.is_animal_control_surrender is True:
    - Total expenses*0.15
  - Else Total expenses*1.15

```
SELECT IF(D.is_animal_control_surrender IS TRUE, SUM(E.amount)*0.15 , SUM(E.amount)
*1.15) AS Fee
FROM `Dog` D LEFT JOIN  `Expense` E ON D.dogID = E.dogID
WHERE E.dogID = '$DogID';
```

- Update total application fee to AdoptionApplication table and render the **associate** form

```
UPDATE `AdoptionApplication` SET dogID = '$DogID', fee = '$Fee'
```

```
WHERE application_number = '$ApplicationNumber';
```

- Admin clicked on **Email** link:
    - Render **date input field** on the adoption form
- Admin **enter** adoption date
- If text validation is successful for date, then:
    - If ***Submit*** button is clicked:
        - Update into AdoptionApplication.*dogID* and AdoptionApplication.*adopted_date*

```
UPDATE `AdoptionApplication` SET dogID = '$DogID', fee = '$AdoptedDate'
WHERE application_number = '$ApplicationNumber';
```

- Else *adopted_date* input field is invalid
    - Display error message

Animal Control Report

Abstract Code

- Admin clicked on **Animal Control Report** link on **Dog Dashboard**
- Find Dog brought in by animal controls in current and last 6 months for each month ('$DogSetA').
    - Count the number of dogs in ('$DogSetA') as ('$DogCountA') by month.

```
SELECT DATE_FORMAT[1](surrender_date, '%Y-%m') AS month,  COUNT(dogID) AS
dog_count
FROM `Dog`
WHERE is_animal_control_surrender = True AND surrender_date >=
DATE_FORMAT(DATE_SUB[2](NOW(), INTERVAL 6 MONTH) ,'%Y-%m-01 00:00:00')
GROUP BY DATE_FORMAT(surrender_date, '%Y-%m')
```

    - Display ('$DogCountA') by month in the **Animal Control Dog Count** section.
- Among dogs from ('$DogSetA'), find Dog adopted during each month who had spent in the rescue 60 days or more - ('$DogSetB').
    - Count the number of dogs in ('$DogSetB') as ('$DogCountB') by month.

```
SELECT DATE_FORMAT(D.surrender_date , '%Y-%m') AS month,  COUNT(D.dogID) AS
dog_count
FROM `Dog` D
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
```

[1] MySQL DATE_FORMAT(): https://www.w3schools.com/sql/func_mysql_date_format.asp
[2] MySQL DATE_SUB(): https://www.w3schools.com/sql/func_mysql_date_sub.asp

```
WHERE D.is_animal_control_surrender = True
AND D.surrender_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 6 MONTH)
,'%Y-%m-01 00:00:00')
AND AA.adopted_date IS NOT NULL AND TIMESTAMPDIFF[3](DAY, D.surrender_date,
AA.adopted_date) >= 60
GROUP BY DATE_FORMAT(D.surrender_date , '%Y-%m')
```

- Display the number by month in the **Adopted Dog Count** section.

- Calculate total expense for Dog adopted for each month among ('$DogSetA'). Group result by month.

```
SELECT DATE_FORMAT(surrender_date, '%Y-%m') AS month,  SUM(E.amount) AS
dog_expense
FROM `Dog` D
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
JOIN `Expense` E ON  D.dogID = E.dogID
WHERE D.is_animal_control_surrender = True
AND D.surrender_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 6 MONTH)
,'%Y-%m-01 00:00:00')
AND AA.adopted_date IS NOT NULL
GROUP BY DATE_FORMAT(surrender_date, '%Y-%m')
```

- Display ('$DogExpense') by month in the **Dog Expense** section.

- When a particular month ('$Month_A') in **Animal Control Dog Count** section is clicked:
  - Find Dog details for dogs in ('$DogSetA') of ('$Month_A').

```
SELECT D.dogID, D.sex, D.altered, D.microchipID, D.surrender_date,
        IFNULL((SELECT GROUP_CONCAT[4](BR.name)
                FROM Belong BE
                JOIN Breed BR ON BE.breedID = BR.breedID
                WHERE  D.dogID = BE.dogID),'Unknown') AS breed,
        IF( TIMESTAMPDIFF(DAY, D.surrender_date, AA.adopted_date) >= 60,
        TIMESTAMPDIFF(DAY, D.surrender_date, AA.adopted_date), NULL) AS
        days_in_rescue
FROM `Dog` D
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
WHERE D.is_animal_control_surrender = True AND DATE_FORMAT(D.surrender_date ,
'%Y-%m') = '$Month_A'
ORDER BY D.dogID ASC
```

[3] MySQL TIMESTAMPDIFF():
https://www.w3resource.com/mysql/date-and-time-functions/mysql-timestampdiff-function.php
[4] GROUP_CONCAT() Function:
https://www.w3resource.com/mysql/aggregate-functions-and-grouping/aggregate-functions-and-grouping-group_concat.php

- Display Dog details for each dog (dog ID, breed, sex, alteration status, microchip ID, and surrender date). Sort data by dog ID ascending.

- When a particular month ('$Month_B') in the **Adopted Dog Count** section is clicked:
  - Find Dog details for dogs in ('$DogSetB') of ('$Month_B').
  - For dogs in ('$DogSetB'), calculate the number of days they were in rescue ('$DaysInRescue').

```
SELECT D.dogID, D.sex, D.altered, D.microchipID, D.surrender_date,
        IFNULL((SELECT GROUP_CONCAT[5](BR.name)
                FROM Belong BE
                JOIN Breed BR ON BE.breedID = BR.breedID
                WHERE  D.dogID = BE.dogID),'Unknown') AS breed,
        TIMESTAMPDIFF(DAY, D.surrender_date, AA.adopted_date) AS days_in_rescue
FROM `Dog` D
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
WHERE D.is_animal_control_surrender = True AND AA.adopted_date IS NOT NULL AND
TIMESTAMPDIFF(DAY, D.surrender_date, AA.adopted_date) >= 60 AND
DATE_FORMAT(surrender_date, '%Y-%m') = '$Month_B'
ORDER BY TIMESTAMPDIFF(DAY, D.surrender_date, AA.adopted_date) DESC,
D.dogID DESC
```

  - Display Dog details for each dog (dog ID, breed, sex, alteration status, microchip ID, and surrender date). Display ('$DaysInRescue'). Sort data by ('$DaysInRescue') descending, then by dog ID descending.

Monthly Adoption Report

Abstract Code

- Admin click on **Monthly Adoption Report** link on **Dog Dashboard**
- Find Dog surrendered in the last 12 months for each month ('$DogSet_Sur').
  - Count the number of dogs from ('$DogSet_Sur') as ('$DogCount_Sur'), grouping results by month, then by breed.

```
SELECT DATE_FORMAT(D.surrender_date , '%Y-%m') AS sur_month,
        IFNULL((SELECT GROUP_CONCAT( BR.name)
                FROM Belong BE
                JOIN Breed BR ON BE.breedID = BR.breedID
                WHERE  D.dogID = BE.dogID
                ORDER BY BR.name ASC), 'Unknown') AS breed,
        COUNT(D.dogID) AS sur_count
FROM `Dog` D
```

[5] GROUP_CONCAT() Function:
https://www.w3resource.com/mysql/aggregate-functions-and-grouping/aggregate-functions-and-grouping-group_concat.php

```
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
WHERE D.surrender_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11
MONTH) ,'%Y-%m-01 00:00:00')
GROUP BY DATE_FORMAT(D.surrender_date, '%Y-%m'), breed
```

■ Find Dog adopted in the last 12 months for each month ('$DogSet_Ado').

■ Count the number of dogs from ('$DogSet_Ado') as ('$DogCount_Ado'), grouping results by month, then by breed.

```
SELECT DATE_FORMAT(AA.adopted_date , '%Y-%m') AS ado_month,
        IFNULL((SELECT GROUP_CONCAT( BR.name)
                FROM Belong BE
                JOIN Breed BR ON BE.breedID = BR.breedID
                WHERE  D.dogID = BE.dogID
                ORDER BY BR.name ASC), 'Unknown') AS breed,
        COUNT(D.dogID) AS ado_count
FROM `Dog` D
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
WHERE AA.adopted_date IS NOT NULL AND AA.adopted_date >=
DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11 MONTH) ,'%Y-%m-01 00:00:00')
GROUP BY DATE_FORMAT(AA.adopted_date, '%Y-%m'), breed
```

■ Calculate total expense for Dog grouping results by month, then by breed. Save results in ('$DogMoney_Expense')

```
SELECT DATE_FORMAT(E.expense_date, '%Y-%m') AS exp_month,
        IFNULL((SELECT GROUP_CONCAT( BR.name)
                FROM Belong BE
                JOIN Breed BR ON BE.breedID = BR.breedID
                WHERE  D.dogID = BE.dogID
                ORDER BY BR.name ASC), 'Unknown') AS breed,
        SUM(E.amount) AS sur_expense
FROM `Dog` D
JOIN `Expense` E ON  D.dogID = E.dogID
WHERE E.expense_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11 MONTH)
,'%Y-%m-01 00:00:00')
GROUP BY DATE_FORMAT(D.surrender_date, '%Y-%m'), breed
```

■ Calculate total adoption fees for Dog in ('$DogMoney_Ado') grouping results by month, then by breed. Save results in ('$DogMoney_Ado')

```
SELECT DATE_FORMAT(AA.adopted_date, '%Y-%m') AS ado_month,
        IFNULL((SELECT GROUP_CONCAT( BR.name)
                FROM Belong BE
                JOIN Breed BR ON BE.breedID = BR.breedID
                WHERE  D.dogID = BE.dogID
                ORDER BY BR.name ASC), 'Unknown') AS breed,
        SUM(E.amount)*1.15 AS ado_fee
FROM `Dog` D
```

```
JOIN `Expense` E ON  D.dogID = E.dogID
JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
WHERE AA.adopted_date IS NOT NULL AND AA.adopted_date >=
DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11 MONTH) ,'%Y-%m-01 00:00:00')
GROUP BY DATE_FORMAT(AA.adopted_date, '%Y-%m'), breed
```

■ Join data from ('$DogCount_Sur'), ('$DogCount_Ado') and ('$DogMoney_Sum') on month and breed, save results as ('$Mon_Ado_Rep'). Sort ('$Mon_Ado_Rep') by month in ascending order, and by breed alphabetically

```
SELECT FJ.month, FJ.breed, IFNULL[6](SC.sur_count,0) AS sur_count, IFNULL(AC.ado_count,0)
AS ado_count, IFNULL(TE.expense,0) AS total_expense, IFNULL(AF.ado_fee,0) AS
total_adoption_fee,
IFNULL(AF.ado_fee,0) - IFNULL(TE.expense,0) AS net_profit
FROM (
--  Since MySQL doesn't support FULL OUTER JOIN, we need union all 'month' and 'breed'
combination for all other results to LEFT JOIN with
        SELECT DATE_FORMAT(D.surrender_date, '%Y-%m') AS month,
                IFNULL((SELECT GROUP_CONCAT( BR.name)
                        FROM Belong BE
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC), 'Unknown') AS breed
        FROM `Dog` D
        WHERE D.surrender_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11
        MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(D.surrender_date, '%Y-%m'), breed

        UNION

        SELECT DATE_FORMAT(AA.adopted_date, '%Y-%m') AS month,
                IFNULL((SELECT GROUP_CONCAT( BR.name)
                        FROM Belong BE
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC), 'Unknown') AS breed
        FROM `Dog` D
        JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
        WHERE AA.adopted_date IS NOT NULL AND AA.adopted_date >=
        DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11 MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(AA.adopted_date, '%Y-%m'), breed

        UNION

        SELECT DATE_FORMAT(E.expense_date, '%Y-%m') AS month,
                IFNULL((SELECT GROUP_CONCAT(BR.name)
                        FROM Belong BE
```

[6]MySQL IFNULL() Function: https://www.w3schools.com/sql/func_mysql_ifnull.asp

```
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC),'Unknown')  AS breed
        FROM `Dog` D
        JOIN `Expense` E ON  D.dogID = E.dogID
        WHERE E.expense_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11
        MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(D.surrender_date, '%Y-%m'), breed
) AS FJ

LEFT JOIN (
        SELECT DATE_FORMAT(D.surrender_date, '%Y-%m') AS sur_month,
                IFNULL((SELECT GROUP_CONCAT(BR.name)
                        FROM Belong BE
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC),'Unknown')  AS breed,
                COUNT(D.dogID) AS sur_count
        FROM `Dog` D
        WHERE D.surrender_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11
        MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(D.surrender_date, '%Y-%m'), breed
) AS SC
ON SC.sur_month = FJ.month AND SC.breed = FJ.breed

LEFT JOIN (
        SELECT DATE_FORMAT(AA.adopted_date, '%Y-%m') AS ado_month,
                IFNULL((SELECT GROUP_CONCAT(BR.name)
                        FROM Belong BE
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC),'Unknown')  AS breed,
                COUNT(D.dogID) AS ado_count
        FROM `Dog` D
        JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
        WHERE AA.adopted_date IS NOT NULL AND AA.adopted_date >=
        DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11 MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(AA.adopted_date, '%Y-%m'), breed
) AS AC
ON AC.ado_month = FJ.month AND AC.breed = FJ.breed

LEFT JOIN (
        SELECT DATE_FORMAT(E.expense_date, '%Y-%m') AS exp_month,
                IFNULL((SELECT GROUP_CONCAT(BR.name)
                        FROM Belong BE
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC),'Unknown')  AS breed,
                SUM(E.amount) AS expense
```

```
        FROM `Dog` D
        JOIN `Expense` E ON  D.dogID = E.dogID
        WHERE E.expense_date >= DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11
        MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(D.surrender_date, '%Y-%m'), breed
) AS TE
ON TE.exp_month = FJ.month AND TE.breed = FJ.breed

LEFT JOIN (
        SELECT DATE_FORMAT(AA.adopted_date, '%Y-%m') AS ado_month,
                IFNULL((SELECT GROUP_CONCAT(BR.name)
                        FROM Belong BE
                        JOIN Breed BR ON BE.breedID = BR.breedID
                        WHERE  D.dogID = BE.dogID
                        ORDER BY BR.name ASC
                        ), 'Unknown') AS breed,
                SUM(E.amount)*1.15 AS ado_fee
        FROM `Dog` D
        JOIN `Expense` E ON  D.dogID = E.dogID
        JOIN `AdoptionApplication` AA ON D.dogID = AA.dogID
        WHERE AA.adopted_date IS NOT NULL AND AA.adopted_date >=
        DATE_FORMAT(DATE_SUB(NOW(), INTERVAL 11 MONTH) ,'%Y-%m-01 00:00:00')
        GROUP BY DATE_FORMAT(AA.adopted_date, '%Y-%m'), breed
) AS AF
ON AF.ado_month = FJ.month AND AF.breed = FJ.breed
ORDER BY FJ.month ASC, FJ.breed ASC
```

- Display ('$Mon_Ado_Rep').

Volunteer Lookup

Abstract Code

- Admin click on **Volunteer Lookup** link on **Dashboard**
- Admin enters *search_text* ('$SearchText').
    - When ***Enter*** button is clicked:
        - Find any User (Volunteer) with FirstName or LastName contains ('$SearchText'), case insensitive, sorting by LastName ascending and FirstName ascending. Save results as ('$MatchedUser')

```
SELECT first_name, last_name, email, cell_phone
FROM `User`
WHERE is_admin = false AND ( LOWER[7](first_name) like
LOWER('%$SearchText%') OR LOWER(last_name) like
LOWER('%$SearchText%'))
ORDER BY last_name ASC, first_name ASC
```

[7] LOWER() Function: https://www.w3schools.com/sql/func_sqlserver_lower.asp

■ Display User's first name, last name, email address, and phone number of ('$MatchedUser').

Expense Analysis

Abstract Code

■ Admin click on **Expense Analysis** link on **Dashboard**

■ Sum up expenses with grouping by vendor name. Sort results by total expenses descending.

```
SELECT vendor, SUM(amount) AS total_expense
FROM `Expense`
GROUP BY vendor
ORDER BY SUM(amount) DESC
```

■ Display the results.