Team Member: Hang Biao Li And Andrew Rengel
Team contribution:
Coding: Hang Biao Li 50% and Andrew Rengel 50%
Testing: Hang Biao Li 50% and Andrew Rengel 50%
Debug: Hang Biao Li 50% and Andrew Rengel 50%
Planning: Hang Biao 65% and Andrew Rengel 35%
Research: Hang Biao 60% and Andrew Rengel 40%

Extra Credit - When a user submits a SQL, the application may perform a validation of the SQL: Hang Biao 30% and Andrew Rengel 70%

EC2 Login MYSQL WorkBench:
IPv4: 18.206.156.104
user name: connection
Password: Penguin03!

```
connect = DriverManager
.getConnection("jdbc:mysql://18.206.156.104/chad?"
"user=connection&password=Penguin03!");
```

Limitations:

- We change the column names for chad_encoded_data because while trying to convert the parameter into a '?', it also changes the column name into a question mark. Also since the regex in our program under convertSQL needs to find a way to not change any table rename names into a question mark by detecting if there is a column name on the right side of the string.
- When using Select ? from chad_encoded_data: it will print out a long list of just the parameter in a table, the reason might be how the system handles when users input the columns.

(How we renamed the columns):\

```
ALTER TABLE chad_encoded_data
```

```
CHANGE COLUMN Q28 to Question28;
ALTER TABLE chad_encoded_data
CHANGE COLUMN q29b to Question29b;
ALTER TABLE chad_encoded_data
CHANGE COLUMN S6 to QuestionS6;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q8 to Question8;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q9 to Question9;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q12 to Question12;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q16 to Question16;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q17 to Question17;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q18 to Question18;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q19 to Question19;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q24 to Question24;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q45 to Question45;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q48 to Question48;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q46 to Question46;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q47 to Question47;
ALTER TABLE chad_encoded_data
CHANGE COLUMN Q48 to Question48;
```

- Another limitation is the sql query must be in a single line and not have multiple lines. We use scan.nextLine() so if the following is inputed:

```
select count(*)
from chad_encoded_data
where Question16 = '5' AND
     Queestion47 in (select response_encoded_value
          from data_dictionary
          where question_label = 'Q47'
```

and ref_response_description between 44 and 47);

Then the program will only take in the "**select count(*)**" as the sql query.

Bugs:

- Ran into issues where scan.nextLine() would not work properly. To debug this we added extra scan.nextLine() statements and that fixed the issue.

- Copy and pasting over quotes sometimes copies curved single quotes instead of straight single quotes in the SQL statement and will result in error due to system (depends on text editor used)
- There may be some bugs that we have not yet encountered that may not convert parameters that are not a column name into question marks.

Test Case and (some) Results:

- **Question:**
  Count all responses who have an annual income of 100k or more (Q47 e - h) who find
  healthy foods as "5" easiest to obtain (Q16)
  SQL Query:
  select count(*) from chad_encoded_data where Question16 = '5' AND Question47 in (select response_encoded_value from data_dictionary where question_label = 'Q47' and ref_response_description between 44 and 47);
  Results:

```
hello User
Are you:
1. Specify a problem
2. contribute a SQL
3. display a list of runnable query
3
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is
`com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and
manual loading of the driver class is generally unnecessary.
Here is a list of already solved questons:
(1) Count all responses who have an annual income of 100k or more (Q47 e - h) who find
healthy foods as "5" easiest to obtain (Q16)
```

```
(2) For each zip code area (S6), count the number of people who aren't always able to
access healthcare services when needed because they could not get appointment(s)
and/or too much time to wait before an appointment Q28.
(3) Return a table of all the students(Q45) who make less than $25,000 (Q47) who could
not access a doctor due to not having insurance (Q28).
(4) For those who choose “no” in (Q29), how many do not have insurance or couldn’t
afford it (Q28 = a or b)
(5) For people who are employed full time (SQ45), count the number of people who earn
more than $75,000 and separate into their income tiers (SQ47 d-h)
Please select a problem that has already been solved:
1
Count all responses who have an annual income of 100k or more (Q47 e - h) who find
healthy foods as "5" easiest to obtain (Q16)
select count(?) from chad_encoded_data where Question16 = ? AND Question47 in (select
response_encoded_value from data_dictionary where question_label = ? and
ref_response_description between ? and ?);
The query has 5 parameters to fill.
Enter value type (String, Int, Double) for parameter 1 (Please do exactly as follow
such as Int cant be input as int):
String
Enter value for parameter 1:
*
Enter value type (String, Int, Double) for parameter 2 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 2:
5
Enter value type (String, Int, Double) for parameter 3 (Please do exactly as follow
such as Int cant be input as int):
String
Enter value for parameter 3:
Q47
Enter value type (String, Int, Double) for parameter 4 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 4:
44
Enter value type (String, Int, Double) for parameter 5 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 5:
47
This works!!
Query Result:
count('*'):3|
```

- **Question:**
  For each zip code area (S6), count the number of people who aren't always able to access healthcare services when needed because they could not get appointment(s) and/or too much time to wait before an appointment Q28.

SQL Query:

```
select c.response_description ZIP, t.a_count count from
(Select QuestionS6, count(*) a_count from chad_encoded_data
where (((cast(Question28 AS UNSIGNED) & 4) <> 0) or
((cast(Question28 AS UNSIGNED) & 16) <> 0) or
((cast(Question28 AS UNSIGNED) & 20) <> 0)) group by
Questions6) t, data_dictionary b, response_description c
where t.QuestionS6 = b.response_encoded_value and
b.question_label = 'S6' and b.ref_response_description =
c.response_description_pk;
```

Results:

```
hello User
Are you:
1. Specify a problem
2. contribute a SQL
3. display a list of runnable query
3
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is
`com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and
manual loading of the driver class is generally unnecessary.
Here is a list of already solved questons:
(1) Count all responses who have an annual income of 100k or more (Q47 e - h) who find
healthy foods as "5" easiest to obtain (Q16)
(2) For each zip code area (S6), count the number of people who aren't always able to
access healthcare services when needed because they could not get appointment(s)
and/or too much time to wait before an appointment Q28.
(3) Return a table of all the students(Q45) who make less than $25,000 (Q47) who could
not access a doctor due to not having insurance (Q28).
(4) For those who choose “no” in (Q29), how many do not have insurance or couldn’t
afford it (Q28 = a or b)
(5) For people who are employed full time (SQ45), count the number of people who earn
more than $75,000 and separate into their income tiers (SQ47 d-h)
Please select a problem that has already been solved:
2
For each zip code area (S6), count the number of people who aren't always able to
access healthcare services when needed because they could not get appointment(s)
and/or too much time to wait before an appointment Q28.
select c.response_description ZIP, t.a_count count from (Select QuestionS6, count(?)
a_count from chad_encoded_data where (((cast(Question28 AS UNSIGNED) & ?) <> ?) or
((cast(Question28 AS UNSIGNED) & ?) <> ?) or ((cast(Question28 AS UNSIGNED) & ?) <>
?)) group by Questions6) t, data_dictionary b, response_description c where
t.QuestionS6 = b.response_encoded_value and b.question_label = ? and
b.ref_response_description = c.response_description_pk;
The query has 8 parameters to fill.
Enter value type (String, Int, Double) for parameter 1 (Please do exactly as follow
such as Int cant be input as int):
String
Enter value for parameter 1:
*
```

```
Enter value type (String, Int, Double) for parameter 2 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 2:
4
Enter value type (String, Int, Double) for parameter 3 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 3:
0
Enter value type (String, Int, Double) for parameter 4 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 4:
16
Enter value type (String, Int, Double) for parameter 5 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 5:
0
Enter value type (String, Int, Double) for parameter 6 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 6:
20
Enter value type (String, Int, Double) for parameter 7 (Please do exactly as follow
such as Int cant be input as int):
Int
Enter value for parameter 7:
0
Enter value type (String, Int, Double) for parameter 8 (Please do exactly as follow
such as Int cant be input as int):
String
Enter value for parameter 8:
S6
This works!!
Query Result:
response_description:13440|a_count:37|
response_description:13502|a_count:14|
response_description:13501|a_count:13|
response_description:13492|a_count:6|
response_description:13413|a_count:1|
```

- **Question:**
  Return a table of all the students(Q45) who make less than $25,000 (Q47) who could not access a doctor due to not having insurance (Q28).
  SQL Query:
  Select * From chad_encoded_data Where mod(Question28,2) = 1 And Question45 not in (Select Question45 From

```
chad_encoded_data Where (mod(Question45, 32) <> 0) or
(Question45=64) or (Question45=128) or (Question45=192))
And Question47 = 0;
```

Results: (Not supposed to return anything since no data match)

- **Question:**
  For those who choose “no” in (Q29), how many do not have insurance or couldn’t afford it (Q28 = a or b)
  SQL Query:

```
Select count(*) a_count from chad_encoded_data F where
(((cast(Question28 AS UNSIGNED) & 1) <> 0) or
((cast(Question28 AS UNSIGNED) & 2) <> 0) or
((cast(Question28 AS UNSIGNED) & 3) <> 0)) and Question29b
= 1;
```

- **Question:**
  For people who are employed full time (SQ45), count the number of people who earn more than $75,000 and separate into their income tiers (SQ47 d-h)
  SQL Query:

```
select B.RD_DESCRIPTION, count(B.RD_DESCRIPTION) COUNT from
chad_encoded_data A, (select t1.response_encoded_value
ENCODED, response_description RD_DESCRIPTION from
data_dictionary t1, response_description t2 where
t1.question_label = 'Q47' and t2.response_description_pk =
t1.ref_response_description and t2.response_choice in ('d',
'e', 'f', 'g', 'h')) B where A.Question47 = B.ENCODED group
by B.RD_DESCRIPTION order by COUNT DESC;
```

- **Question:**
  For each zip code area (S6), count the number of people who aren't always able to access healthcare services due to limited availability of healthcare providers or a lack of insurance coverage, based on Q28.
- **Question:**
  For each zip code area (S6), count how many people have difficulty accessing proper aid due to an location issue of doctor being too far away or a lack of transportation coverage, as shown in (Q28) responses.

- **Question:**
  ```
  select count(*) from chad_encoded_data where Q16 = '5' and
  Q46 = (select response_encoded_value from data_dictionary
  t1, response_description t2 where question_label = 'Q46'
  and response_choice = 'f' and t1.ref_response_description =
  t2.response_description_pk);
  ```

Database Schemas:

- **Describe chad_encoded_data; :**

  ```
  'Question28','text','YES','',NULL,''
  'Question29b','text','YES','',NULL,''
  'QuestionS6','text','YES','',NULL,''
  'Question8','text','YES','',NULL,''
  'Question9','text','YES','',NULL,''
  'Question12','text','YES','',NULL,''
  'Question16','text','YES','',NULL,''
  'Question17','text','YES','',NULL,''
  'Question18','text','YES','',NULL,''
  'Question19','text','YES','',NULL,''
  'Question24','text','YES','',NULL,''
  'Question45','text','YES','',NULL,''
  'Question46','text','YES','',NULL,''
  'Question47','text','YES','',NULL,''
  'Question48','text','YES','',NULL,''
  ```

- **Describe data_dictionary; :**

  ```
  'data_dictionary_pk','int(11)','NO','PRI',NULL,'auto_increm
  ent'
  'question_label','varchar(45)','NO','',NULL,''
  'response_encoded_value','int(11)','NO','',NULL,''
  'ref_response_description','int(11)','YES','MUL',NULL,''
  ```

- **Describe response_description; :**

  ```
  'response_description_pk','int(11)','NO','PRI',NULL,'auto_i
  ncrement'
  'response_choice','varchar(35)','NO','',NULL,''
  ```

'response_description','varchar(1024)','NO','',NULL,''

- **Describe SQL_Contributions; :**
  'sql_id','int(11)','NO','PRI',NULL,'auto_increment'
  'problem_id','int(11)','YES','MUL',NULL,''
  'sql_statement','text','NO','',NULL,''
  'parameter_sql_statement','text','YES','',NULL,''

- **Describe Runnable_Queries; :**

  'query_id','int(11)','NO','PRI',NULL,'auto_increment'
  'problem_id','int(11)','YES','MUL',NULL,''
  'sql_id','int(11)','YES','MUL',NULL,''
  'parameter_name','varchar(255)','YES','',NULL,''
  'parameter_type','varchar(255)','YES','',NULL,''

- **Describe Problems; :**
  'problem_id','int(11)','NO','PRI',NULL,'auto_increment'
  'description','text','NO','',NULL,''