

## Project Deliverable 02

### Table Of Content

Introducing CodeXchange – An Error Exchange Platform .....	1
Data Glossary .....	2
Conceptual Model .....	3
Normalized Logical Model.....	4
SQL DDL Create Table .....	5
SQL DML INSERT Statements .....	7
SQL DDL For Views, Function, and Stored-Procedure .....	11
SQL DML SELECT Statements and Answers to Data Questions .....	13
GUI Prototype .....	15
Summary And Reflection .....	17

### Introducing CodeXchange – An Error Exchange Platform

Software developers often encounter obscure error message during their daily development works. As modern software becomes ever more complex, these error messages are getting more and more difficult to decipher and debug. Often time, developers would search these error messages on Google, hoping that someone else must had the same error before and was able to fix it. Since Google is a generic search engine and not tailored for error message search, the results are hit-and-miss. On a good day, one would work through a several suggestions/fixes and finally found one that really works. On a bad day, s/he worked through all possible solutions that could be found on Google but none of them worked. All in all, fixing obscure error is a tiresome process that may take hours, if not days. It also add additional uncertainty to the software development process. CodeXchange is born out of this specific need. CodeXchange is a platform for developers to share error messages and their fixes.

Users on CodeXchange are required to have username, name and email; city and age are optional. Additionally, users have to specify what kind of programming language, software frameworks, software platforms, and application they are proficient in. Users are categorized into 3 roles — site-curator, author, and wander. Site-curator is designated by CodeXchange platform owner. A site-curator can do everything author and wander can do. A wander can post any number of error messages. Error message consists of short description, long description, status (either solved or unsolved) and optional trace-stack. An error message may have contextual information associated with it. Contextual information may be programming language, software framework, software platform, and application. Once a wander posted an error message, s/he becomes the author of that error message. Author of an error message can edit the error message as long as it is not marked as resolved. Edit to error message shall be archived. Only author or site curator can edit contextual information. Once an error message is posted, any users can suggest a fix to it. A fix is simply a description of procedures. An error message

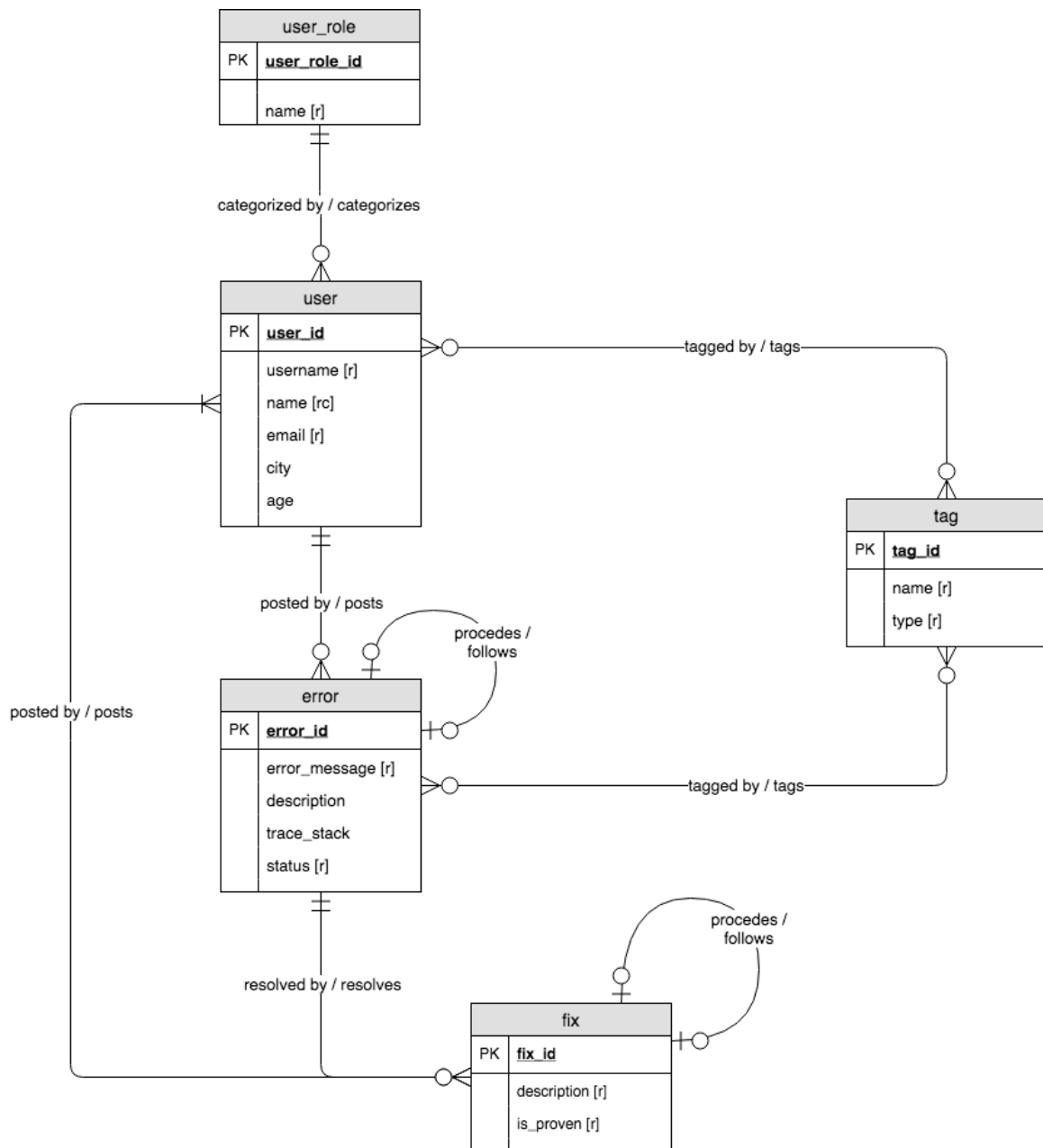
can be associated with more than 1 fix. Any users can edit fixes suggested by any other users as long as the associated error message is not marked as resolved. Similarly, edits to fixes shall be archived. Only author or site-curator can decide whether the fix works. If a fix is proven working, the error message will be marked as resolved.

Data gathered on CodeXchange platform will be used to answer the following questions:

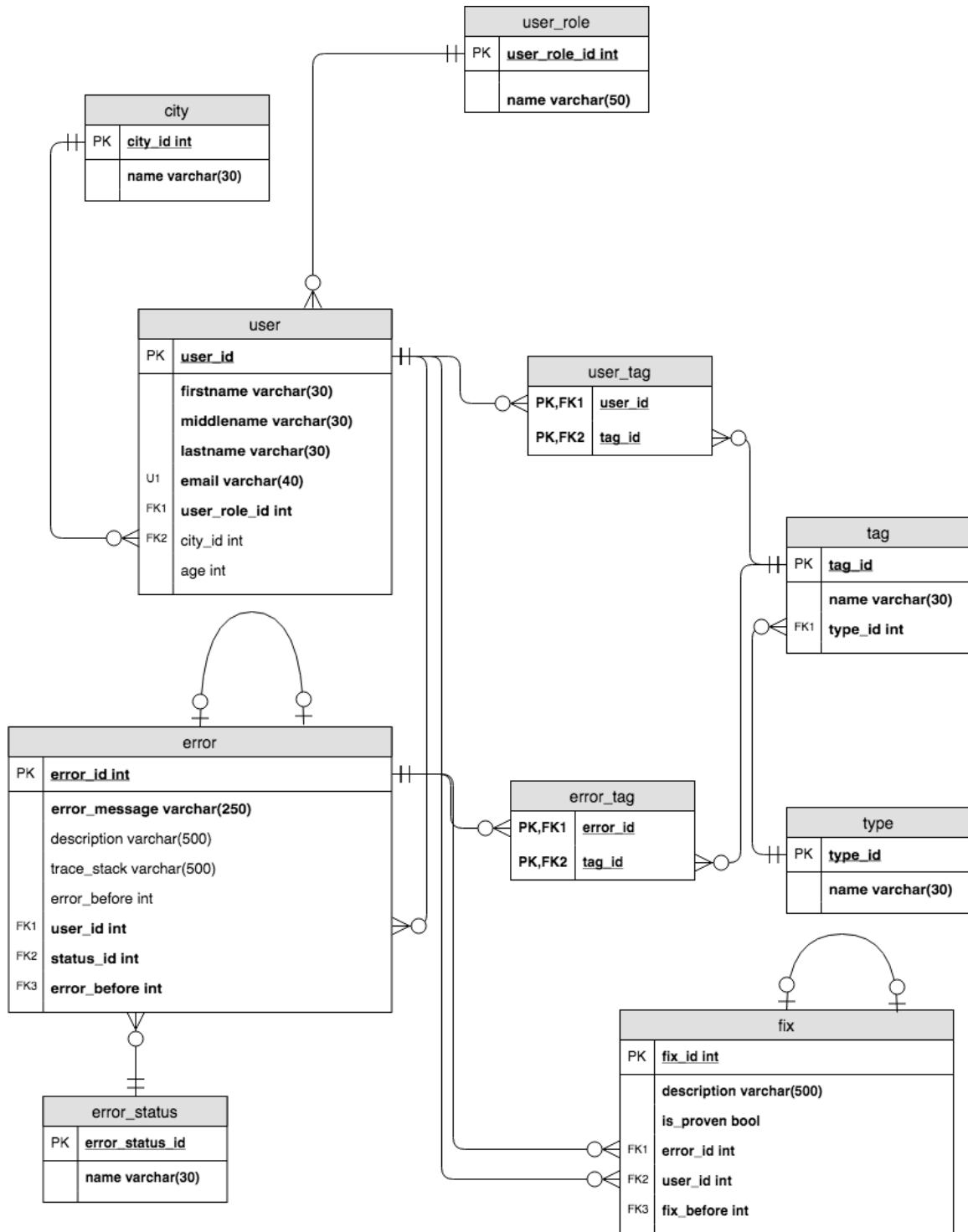
- Which application has the most errors?
- Which programming language cause most errors?
- Which framework has most errors?
- What are the hardest errors?
- Who fixed the most errors?

Data Glossary	
Entity	Attribute
User Role	User Role ID [i] Name [r]
User	User ID [i] Username [r] Name [rc] Email [r] City Age
Error	Error ID [i] Error Message [r] Description Trace Stack Status [r]
Fix	Fix ID [i] Description [r] Is Proven [r]
Tag	Tag ID [i] Name [r] Type [r]
Relationship	
<p>Each <b>user</b> can have only 1 <b>role</b>; each <b>role</b> can belong to 0 or more <b>Users</b>.  Each <b>user</b> can post 0 or more <b>errors</b>; each <b>error</b> must be associated with a <b>user</b>.  Each <b>user</b> can post 0 or more <b>fixes</b>; each <b>fix</b> is posted by at least 1 <b>user</b>.  Each <b>error</b> can be associated with 0 or more <b>tags</b>.  Each <b>user</b> can be associated with 0 or more <b>tags</b>.  Each <b>user</b> can be associated with 0 or more <b>fixes</b>.  Each <b>error</b> may be followed by 0 or 1 <b>error</b>.  Each <b>fix</b> may be followed by 0 or 1 <b>fix</b>.</p>	

## Conceptual Model



## Normalized Logical Model



## SQL DDL Create Table

```
-- Drop all tables if already exist
SET foreign_key_checks = 0;
DROP TABLE IF EXISTS user_role, city, error_status, user, error, fix, type,
                        tag, user_tag, error_tag;
SET foreign_key_checks = 1;

-- Table: user_role
CREATE TABLE user_role (
    user_role_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    PRIMARY KEY(user_role_id)
);

-- Table: city
CREATE TABLE city (
    city_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    PRIMARY KEY(city_id)
);

-- Table: error_status
CREATE TABLE error_status (
    error_status_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    PRIMARY KEY(error_status_id)
);

-- Table: user
CREATE TABLE user (
    user_id INT NOT NULL AUTO_INCREMENT,
    first_name VARCHAR(30) NOT NULL,
    middle_name VARCHAR(30),
    last_name VARCHAR(30) NOT NULL,
    email VARCHAR(40) NOT NULL,
    user_role_id INT NOT NULL,
    city_id INT,
    age INT,
    PRIMARY KEY(user_id),
    UNIQUE(email),
    FOREIGN KEY(user_role_id) REFERENCES user_role(user_role_id),
    FOREIGN KEY(city_id) REFERENCES city(city_id)
);

-- Table: error
CREATE TABLE error (
    error_id INT NOT NULL AUTO_INCREMENT,
```

```

    error_message VARCHAR(250) NOT NULL,
    description VARCHAR(500),
    trace_stack VARCHAR(500),
    user_id INT NOT NULL,
    status_id INT NOT NULL,
    error_before INT,
    PRIMARY KEY(error_id),
    FOREIGN KEY(user_id) REFERENCES user(user_id),
    FOREIGN KEY(status_id) REFERENCES error_status(error_status_id),
    FOREIGN KEY(error_before) REFERENCES error(error_id)
);

-- Table: fix
CREATE TABLE fix (
    fix_id INT NOT NULL AUTO_INCREMENT,
    description VARCHAR(500) NOT NULL,
    is_proven TINYINT DEFAULT 0,
    error_id INT NOT NULL,
    user_id INT NOT NULL,
    fix_before INT,
    PRIMARY KEY(fix_id),
    FOREIGN KEY(error_id) REFERENCES error(error_id),
    FOREIGN KEY(user_id) REFERENCES user(user_id),
    FOREIGN KEY(fix_before) REFERENCES fix(fix_id)
);

-- Table: type
CREATE TABLE type (
    type_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    PRIMARY KEY(type_id)
);

-- Table: tag
CREATE TABLE tag (
    tag_id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(30) NOT NULL,
    type_id INT NOT NULL,
    PRIMARY KEY(tag_id),
    FOREIGN KEY(type_id) REFERENCES type(type_id)
);

-- Table: user_tag
CREATE TABLE user_tag (
    user_id INT NOT NULL,
    tag_id INT NOT NULL,
    PRIMARY KEY(user_id, tag_id)
);

```

```

-- Table: error_tag
CREATE TABLE error_tag (
    error_id INT NOT NULL,
    tag_id INT NOT NULL,
    PRIMARY KEY(error_id, tag_id)
);

```

## SQL DML INSERT Statements

```

-- Insert user roles
INSERT INTO user_role(name)
VALUES('curator'),
      ('user');

-- Insert cities
INSERT INTO city(name)
VALUES('Los Angeles'),
      ('San Francisco'),
      ('San Diego'),
      ('New York'),
      ('Chicago'),
      ('Austin'),
      ('Las Vegas'),
      ('Utah'),
      ('Mexico City'),
      ('Toronto');

-- Insert error status
INSERT INTO error_status(name)
VALUES('open'),
      ('fixed');

-- Insert types
INSERT INTO type(name)
VALUES('programming language'),
      ('software frameworks'),
      ('application');

-- Insert tags
INSERT INTO tag(name, type_id)
VALUES('c#', 1),
      ('python', 1),
      ('javascript', 1),
      ('django', 2),
      ('react', 2),

```

```

('prism', 2),
('mssql', 3),
('elastic search', 3),
('access', 3);

-- Insert users
INSERT INTO user(firstname, lastname, email, user_role_id)
VALUES('Jason', 'Kang', 'jason@gmail.com', 1),
      ('Jack', 'Jackson', 'jack@yahoo.com', 2),
      ('Christine', 'Luk', 'luk@gmail.com', 2),
      ('Don', 'Benson', 'don@gmail.com', 2),
      ('James', 'Fox', 'james@outlook.com', 2),
      ('Bryan', 'Dave', 'bryan@mail.com', 2),
      ('Ruby', 'White', 'ruby@yahoo.com', 2),
      ('Sam', 'Kong', 'sam@gmail.com', 2),
      ('Ying', 'Tin', 'ying@microsoft.com', 2),
      ('Tiff', 'Li', 'tiff@esri.com', 2),
      ('Pitt', 'Mavis', 'pitt@lsi.com', 2);

-- Insert errors
INSERT INTO error(error_message, user_id, status_id)
VALUES('django.db.transaction.TransactionManagementError: An error occurred in
      the current transaction. You cannot execute queries until the end of
      the atomic block', 2, 2),
      ('Error: dictionary update sequence element #0 has length 1; 2 is
      required', 3, 2),
      ('Error: That port is already in use', 3, 2),
      ('{"error": "invalid_token", "error_description": ""} (Podbean api)', 6, 2),
      ('Uncaught Error: Invariant Violation: Element type is invalid: expected
      a string for built-in components or a class/function but got: object',
      11, 2),
      ('Unexpected token \<', 9, 2),
      ('XmlHttpRequest error: Origin null is not allowed by Access-Control-
      Allow-Origin', 6, 2),
      ('HTTP POST Returns Error: 417 "Expectation Failed."', 7, 2),
      ('MSSQL Error \'The underlying provider failed on Open\'', 2, 2),
      ('An attempt was made to load a program with an incorrect format error',
      5, 2),
      (' error: An exception of type
      \'Microsoft.Practices.ServiceLocation.ActivationException\' occurred in
      Microsoft.Practices.ServiceLocation.dll', 4, 2),
      ('System.Windows.Markup.XamlParseException', 5, 2),
      ('Elasticsearch no requests added Bulk API Error', 10, 2),
      ('error type mismatch', 10, 2),
      ('Setting DEBUG = False causes 500 Error', 8, 2),
      ('an error with models.DateField()', 9, 2),
      ('Looks like your browser is not configured to accept cookies. Please
      enable cookies, reload this page, and try again.', 5, 2),

```



```

('Error: Cannot set headers after they are sent to the client', 7, 2),
('pip install lxml error', 6, 2),
('rest_framework authtoken migration error', 7, 2);

INSERT INTO error_tag(error_id, tag_id)
VALUES(1, 2),
      (1, 4),
      (2, 2),
      (2, 4),
      (3, 2),
      (3, 4),
      (4, 2),
      (4, 4),
      (5, 3),
      (5, 5),
      (6, 3),
      (6, 5),
      (7, 3),
      (8, 1),
      (9, 7),
      (10, 7),
      (11, 1),
      (11, 6),
      (12, 1),
      (12, 6),
      (13, 8),
      (14, 9),
      (15, 2),
      (15, 4),
      (16, 2),
      (16, 4),
      (17, 2),
      (17, 4),
      (18, 3),
      (19, 2),
      (20, 2),
      (20, 4);

-- Insert fixes
INSERT INTO fix(description, is_proven, error_id, user_id)
VALUES('I was getting this error on running unit tests in my create_test_data
      function using django 1.9.7. It worked in earlier versions of django.',
      0, 1, 1),
      ('with transaction.atomic(): did the trick', 1, 1, 3),
      ('Just ran into this problem. I do not know if it is the same thing that
      hit your code, but for me the root cause was because I forgot to put
      name= on the last argument of the url function call.', 1, 2, 4),
      ('A more simple solution just type sudo fuser -k 8000/tcp. This should

```

```

    kill all the processes associated with port 8000', 1, 3, 7),
('client_id = App ID client_secert = App Secret', 1, 4, 9),
('var About = require(\"../components/Home\").default', 0, 5, 1),
('Have you just modularized any of your React components? If yes, you
  will get this error if you forgot to specify module.exports', 0, 5, 5),
('If you get this error, it might be because you are importing link using
  import { Link } from \'react-router\'', 0, 5, 8),
('import {MyComponent} from \'../components/xyz.js\';', 1, 5, 7),
('Make sure include the JSX pragma at the top of your files', 0, 6, 5),
('I solved it using type = "text/babel"', 1, 6, 2),
('header(\'Access-Control-Allow-Origin: *\');', 1, 7, 6),
('System.Net.ServicePointManager.Expect100Continue = false;', 1, 8, 2),
('In windows 7 you can open the DTC config by running dcomcnfg, Component
  Services -> Computers -> My Computer -> Distributed Transaction
  Coordinator -> Right click to Local DTC -> Security.', 0, 9, 1),
('context.Connection.Open()', 1, 9, 11),
('Change the value for Platform Target on your web project property page
  to Any CPU', 0, 10, 1),
('I have found the solution. I have recently upgraded my machine to
  Windows 2008 Server 64-bit. The SqlServer.Replication namespace was
  written for 32-bit platforms. All I needed to do to get it running
  again was to set the Target Platform in the Project Build Properties to
  X86.', 1, 10, 2),
('using Microsoft.Practices.Prism.PubSubEvents;', 1, 11, 2),
('You are probably calling a virtual function that was intended for you
  to override. That method will throw the NotImplementedException
  (usually developers throw this exception in virtual methods they want
  people to implement, instead of making them abstract)', 1, 12, 2),
('If you are using cURL, you must have a blank line at the end of your
  bulk items and you must use --data-binary (instead of plain -d).', 0,
  13, 3),
('Even though i had \\n on the last line I literally HAD to have a full
  carriage return after my last json line.', 1, 13, 7),
('You are passing a string into a date, change it to String.', 1, 14, 2),
('COMPRESS_ENABLED = os.environ.get(\'COMPRESS_ENABLED\', False)', 0, 15,
  1),
('Django 1.5 introduced the allowed hosts setting that is required for
  security reasons. A settings file created with Django 1.5 has this new
  section which you need to add', 0, 15, 11),
('I encountered the same issue just recently in Django 2.0. I was able to
  figure out the problem by setting DEBUG_PROPAGATE_EXCEPTIONS = True.',
  1, 15, 2),
('Your __unicode__ method must return a unicode string, not a date.', 1,
  16, 2),
('One thing you might want to look into is the setting
  SESSION_COOKIE_SECURE. If it is set to True make sure you are using
  https. Otherwise, set it to False and use http.', 1, 17, 2),
('When you add middleware to connect or express (which is built on

```

connect) using the app.use method, you are appending items to Server.prototype.stack in connect (At least with the current npm install connect, which looks quite different from the one github as of this post). When the server gets a request, it iterates over the stack, calling the (request, response, next) method.', 0, 18, 1),

('I had this same issue and realised it was because I was calling res.redirect without a return statement, so the next function was also being called immediately afterwards.', 0, 18, 2),

('The res object in Express is a subclass of Node.js http.ServerResponse (read the http.js source). You are allowed to call res.setHeader(name, value) as often as you want until you call res.writeHead(statusCode). After writeHead, the headers are baked in and you can only call res.write(data), and finally res.end(data).', 1, 18, 11),

('make sure the development packages of libxml2 and libxslt are installed', 1, 19, 2),

('Because of the custom user table name the migration is not happening. Go to the initial migration and specify your user table using db\_table in the migration where the code is trying to access your custom table. That should work.', 1, 20, 2);

## SQL DDL For Views, Function, and Stored-Procedure

```
-- Trigger/Store-Procedure to mark an error fixed if a fix is proven
DROP TRIGGER IF EXISTS mark_error_fixed;
DELIMITER $$
CREATE TRIGGER mark_error_fixed AFTER UPDATE ON fix
FOR EACH ROW BEGIN
    IF (NEW.is_proven = 1 AND OLD.is_proven = 0) THEN
        UPDATE error
        SET status_id = 2
        WHERE error_id = NEW.error_id;
    END IF;
END$$
DELIMITER ;

-- View showing applications with the most errors
DROP VIEW IF EXISTS wrost_application;
CREATE VIEW wrost_application AS
    SELECT ta.name, count(*) as errors
    FROM error e INNER JOIN error_tag et ON e.error_id = et.error_id
        INNER JOIN tag ta ON et.tag_id = ta.tag_id
        INNER JOIN type ty ON ta.type_id = ty.type_id
    WHERE ty.type_id = 3
    GROUP BY ta.name
    ORDER BY errors DESC;

-- View showing programming languages with the most errors
```

```

DROP VIEW IF EXISTS wrost_programming_language;
CREATE VIEW wrost_programming_language AS
    SELECT ta.name, count(*) as errors
    FROM error e INNER JOIN error_tag et ON e.error_id = et.error_id
        INNER JOIN tag ta ON et.tag_id = ta.tag_id
        INNER JOIN type ty ON ta.type_id = ty.type_id
    WHERE ty.type_id = 1
    GROUP BY ta.name
    ORDER BY errors DESC;

-- View showing software frameworks with the most errors
DROP VIEW IF EXISTS wrost_software_framework;
CREATE VIEW wrost_software_framework AS
    SELECT ta.name, count(*) as errors
    FROM error e INNER JOIN error_tag et ON e.error_id = et.error_id
        INNER JOIN tag ta ON et.tag_id = ta.tag_id
        INNER JOIN type ty ON ta.type_id = ty.type_id
    WHERE ty.type_id = 2
    GROUP BY ta.name
    ORDER BY errors DESC;

-- View showing the hardest error in descending order
DROP VIEW IF EXISTS hardest_errors;
CREATE VIEW hardest_errors AS
    SELECT e.error_id, e.error_message, count(*) as fixes
    FROM error e INNER JOIN fix f ON e.error_id = f.error_id
    GROUP BY e.error_id, e.error_message
    ORDER BY fixes DESC;

-- Function to calculate successful fix ratio for user
DROP FUNCTION IF EXISTS calc_success_fix_ratio;
DELIMITER $$
CREATE FUNCTION calc_success_fix_ratio(userId INT) RETURNS INT
BEGIN
    DECLARE all_fixes INT;
    DECLARE successful_fixes INT;

    SELECT count(*) INTO all_fixes
    FROM fix f
    WHERE f.user_id = userId;

    SELECT count(*) INTO successful_fixes
    FROM fix f
    WHERE f.user_id = userId AND f.is_proven = 1;

    IF (all_fixes = 0 OR successful_fixes = 0) THEN
        RETURN 0.0;
    END IF;

```

```

        RETURN successful_fixes / all_fixes * 100;
END$$
DELIMITER ;

-- View showing user who fixed the most errors
DROP VIEW IF EXISTS most_efficient_user;
CREATE VIEW most_efficient_user AS
    SELECT u.user_id, CONCAT(u.firstname, ' ', u.lastname) as name, count(*) as
successful_fixes, calc_success_fix_ratio(u.user_id) as 'success rate in %'
    FROM user u INNER JOIN fix f ON u.user_id = f.user_id
    WHERE f.is_proven = 1
    GROUP BY u.user_id, u.firstname, u.lastname
    ORDER BY successful_fixes DESC

```

## SQL DML SELECT Statements and Answers to Data Questions

The Five questions listed under introduction will be answered in the following.

1. Which application has the most error?

```
SELECT * FROM wrost_application;
```

name	errors
mssql	2
elastic search	1
access	1

Answer: Mssql has the most errors.

2. Which programming language cause most errors?

```
SELECT * FROM wrost_programming_language;
```

name	errors
python	9
javascript	4
c#	3

Answer: Python causes the most errors.

3. Which software framework has most errors?

```
SELECT * FROM wrost_software_framework;
```

name	errors
django	8
prism	2
react	2

Answer: Django framework has the most errors.

4. What are the hardest errors?

```
SELECT * FROM hardest_errors;
```

error_id	error_message	fixes
5	Uncaught Error: Invariant Violation: Element type is invalid: expected a string...	4
15	Setting DEBUG = False causes 500 Error	3
18	Error: Cannot set headers after they are sent to the client	3
1	django.db.transaction.TransactionManagementError: An error occurred in th...	2
6	Unexpected token '<'	2
9	MSSQL Error 'The underlying provider failed on Open'	2
10	An attempt was made to load a program with an incorrect format error	2
13	Elasticsearch no requests added Bulk API Error	2
2	Error: dictionary update sequence element #0 has length 1; 2 is required	1
3	Error: That port is already in use	1
4	{"error": "invalid_token", "error_description": ""} (Podbean api)	1
7	XmlHttpRequest error: Origin null is not allowed by Access-Control-Allow-Origin	1
8	HTTP POST Returns Error: 417 "Expectation Failed."	1
11	error: An exception of type 'Microsoft.Practices.ServiceLocation.ActivationE...	1
12	System.Windows.Markup.XamlParseException	1
14	error type mismatch	1
16	an error with models.DateField()	1
17	Looks like your browser is not configured to accept cookies. Please enable c...	1
19	pip install lxml error	1
20	rest_framework authtoken migration error	1

Answer: The hardest errors caused 4 fixes.

5. Who fixed the most errors?

```
SELECT * FROM most_efficient_user;
```

user_id	name	successful_fixes	success rate in %
2	Jack Jackson	11	92
7	Ruby White	3	100
11	Pitt Mavis	2	67
4	Don Benson	1	100
3	Christine Luk	1	50
6	Bryan Dave	1	100
9	Ying Tin	1	100

Answer: Jack Jackson fixed the most errors.

## GUI Prototype


Create a new user

A Web Page

← → ↻

http://cod.exchange/user

≡



first name

middle name

last name

email

role

user ▼

city


Los Angeles ▼

age

## Input an error

A Web Page

← → ↻



error message

description

trace stack



## List all fixes associated with an error

A Web Page

← → ↻

http://cod.exchange/errors

☰

error message

Elasticsearch no requests added Bulk API Error

description

trace stack

Christine Luk:

mark proven

If you are using cURL, you must have a blank line at the end of your bulk items and you must use -databinary (instead of plain -d).

Ruby White:

mark proven

Even though i had \n on the last line I literally HAD to have a full carriage return after my last json line.

## Summary And Reflection

This is a prototype implementation of my pet project idea. In hindsight, the actually implementation may need to use a NOSQL database such as MongoDB or Elastic Search to allow data type flexibility, and full text search capability. Originally, I only intended to allow users to share a simple error message. It turned out that it is really difficult to convey contextual information through a simple error message. How to allow users to convey a complete error message without the overhead of typing out a paragraph? I think it has to be a desktop tool/plugin instead of a website.

All and all, drawing out the business rules and conceptual diagram helps me to understand the project a lot better. I have a feeling that I'm ready to bring this pet project to the next step.