

Code Review Worksheet

CPSC 491/491L/492L

Review Identification Information

Date of Review:	11/11/23
Project Review is Part of:	Medcurity
Branch/PR Identifier:	91db575541563a30e432806c306d091ffd00b0b3
Code Reviewer(s):	Artis Nateephaisan
Code Author or Source of Code:	Brandon Huyck

These following categories should be addressed, though not all might be applicable, it will depend upon the project and code at hand.

#1 Project, Milestone, Issues, Features Included, and/or Pull Request description

This current iteration of the code addresses the initial database design, including how we want the tables to be set up, how to handle all the data to be stored, what the primary and foreign keys are going to be, what should be the auto increment, etc.

#2 Code Structural, Design Considerations, and Style Guide Suggestions

The styling and design look good. There are comments addressing potential design choices the team can decide, showing that the author is aware of these considerations. The only glaring structural issue is that the Inv_Item table has too many columns, so splitting the table into smaller tables may be a good idea.

#3 Code Architectural and Efficiency Considerations

Efficiency was addressed by our DAB member. He said that our database does not need to be 100 percent optimized, since the data we're handling is unlikely to cause significant concerns regarding performance. Since our current database only has tables with columns, there's not much we can test in terms of efficiency yet. The code architect looks fine with its initial setup.

#4 Execution, Runtime, and Bugs

There's no rows of data to be tested for yet.

#5 Documentation Quality and Completeness

No documentation so far since the database is our initial prototype. Complete in regards to accounting for all the data points to be collected in reference to the example excel file given by our sponsor.

#6 Testing, Tests, Coverage, and suggested testing improvements

No testing yet, but not much to be tested with the initial database layout.

#7 General Comments, Notes or Other Suggestions for Author

The only thing I would suggest, which is addressed in line 49 of the code as a comment, is to break up the Inv_Item table into smaller bits. While it is likely the biggest component of our database, it can likely be divided into smaller parts for organization purposes. The team will have to discuss how to do this.

#8 Appendix - Include up to five pages of the code under review as a diff.

See git diff or do a diff in VS code/IDE of choice.

Show what's being reviewed, code or otherwise (config files, scripts, docs, etc).

Brandon Huyck added prim and forgn keys

Code Blame 90 lines (77 loc) · 2.33 KB

```
1
2 SET FOREIGN_KEY_CHECKS = 0;
3 DROP TABLE IF EXISTS Client;
4 DROP TABLE IF EXISTS User;
5 DROP TABLE IF EXISTS Inv_Item;
6 DROP TABLE IF EXISTS Vender;
7 DROP TABLE IF EXISTS Server;
8 DROP TABLE IF EXISTS Location;
9 SET FOREIGN_KEY_CHECKS = 1;
10
11 CREATE TABLE Client (
12     name VARCHAR(255) PRIMARY KEY
13 );
14
15 CREATE TABLE User (
16     client VARCHAR(255) NOT NULL,
17     user_name VARCHAR(255) NOT NULL,
18     psw_hash_salted INT NOT NULL,
19     PRIMARY KEY (client, user_name),
20     FOREIGN KEY (client) REFERENCES Client(name)
21 );
22
23 CREATE TABLE Vender (
24     -- id INT UNSIGNED AUTO NOT NULL,
25     poc VARCHAR(255),
26     email VARCHAR(255) PRIMARY KEY,
27     baa BOOL,
28     date DATE
29 );
30
31 CREATE TABLE Location (
32     id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
33     cloud_prem ENUM('Cloud', 'On-Premise'),
34     details VARCHAR(255),
35     protection MEDIUMTEXT
36 );
37
38 CREATE TABLE Server (
39     id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
40     name VARCHAR(255),
41     ip_address BIGINT UNSIGNED,
42     ip_version ENUM('IPv4', 'IPv6'),
43     location_id INT UNSIGNED NOT NULL,
44     CHECK (ip_address > 4294967295 OR ip_version='IPv4'),
45     UNIQUE (ip_address, ip_version),
46     FOREIGN KEY (location_id) REFERENCES Location(id)
47 );
48
```

```

49  -- break into smaller tables?
50  CREATE TABLE Inv_Item (
51      item_id INT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
52      client VARCHAR(255) NOT NULL,
53
54      name VARCHAR(255),
55      type VARCHAR(255),
56      version VARCHAR(255),
57      os VARCHAR(255),
58      os_version VARCHAR(255),
59
60      vender VARCHAR(255),
61      auto_log_off_freq INT UNSIGNED,
62      server INT UNSIGNED,
63
64      ephi BOOL,
65      ephi_encrypted BOOL,
66      ephi_encr_method VARCHAR(255),
67      ephi_encr_tested BOOL,
68      interfaces_with MEDIUMTEXT, -- collection? references other Inv_Item entries? or varchar/MEDIUMTEXT text field?
69
70      user_auth_method VARCHAR(255),
71      app_auth_method VARCHAR(255),
72      psw_min_len INT UNSIGNED,
73      psw_change_freq INT UNSIGNED,
74
75      dept VARCHAR(255),
76      space VARCHAR(255),
77      date_last_ordered DATE,
78      purchase_price DECIMAL(13, 4), -- GAP guideline
79      warranty_expires DATE,
80      item_condition VARCHAR(255), -- larger? enum?
81      quantity INT UNSIGNED,
82      asset_value DECIMAL(13, 4),
83      model_num VARCHAR(255),
84      notes MEDIUMTEXT,
85      link VARCHAR(255),
86
87      FOREIGN KEY (client) REFERENCES Client(name),
88      FOREIGN KEY (vender) REFERENCES Vender(email),
89      FOREIGN KEY (server) REFERENCES Server(id)
90  );

```