7/15/2022

Northern Quest Resort & Casino Machine Activity

Database Re-Design

Keith Bruyer
MARYVILLE UNIVERSITY
SWDV 691 SOFTWARE DEVELOPMENT CAPSTONE

Overview

This project will address the needs of Northern Quest Resort & Casino to build and maintain a knowledge base of the various incidents encountered on select casino equipment. In addition, this application will also warehouse the solutions to these issues as well. A PostgreSQL database will be used to store all data related to this web application. I chose an object-relational database system, in large part, because of the landscape of the web application. There will be a need to create relationships between the tables and fields, primarily for search and reporting purposes. Also, since an SQL database is currently being used, the migration of the old data into the new system will be much easier than a document base system such as MongoDB. And finally, since the Django framework will be used to create this web app, PostgreSQL is a logical choice as it has the richest set of features supported by Django.

I. Data Specifications

The settings.py file of the project will contain all the project settings along with database connection details. Database connectivity requires all the connection details such as database name, user credentials, hostname drive name, etc. The settings.py file of our project will contain the following code will be used:

```
DATABASES = {
        “default”: env.dj_db_url(“DATABASE_URL”)
    }
```

To connect with PostgreSQL, the *psycopg2-binary* driver will be used to establish a connection between the application and the database.

II. Requirements

This project will have several key features which will be made available through a well-designed database. The database will have tables that will be used to track their equipment information and maintenance history together with a reporting package. In addition to this, the store will be able to track each store's inventory in real time. The Gaming Department's technicians will be updating the database through a user interface. Almost every feature of the application will interact with the database in one way or another. For instance, from the home screen, the daily activity reports will be saved to the database, and as incidents are reported, both the incident and the solution will be tracked and stored in the database. In this way, the data can be used to solve future incidents of the same nature. In addition to this, as incidents and solutions are tracked, this information can be used by management to track the activity of their techs and that of a specific piece of equipment.

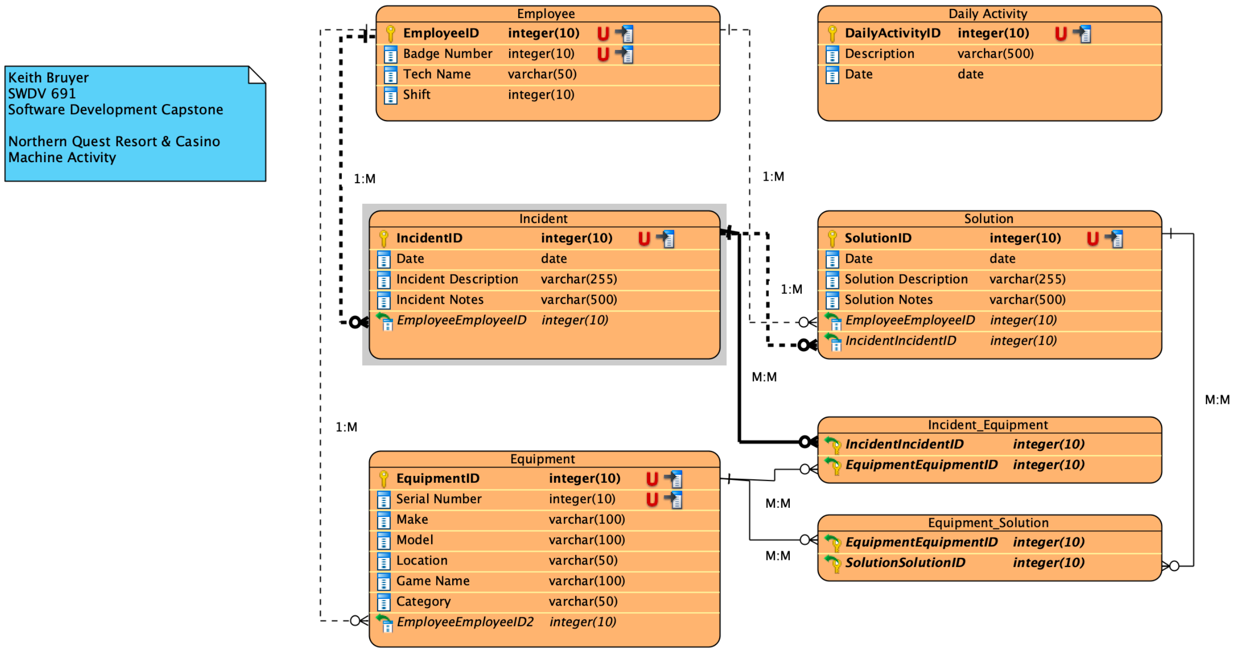
III. Design Considerations

Tables to be used:	**Attributes:**	**Keys:**	**Data Type**
Equipment	EquipmentID	Primary	int
	EmployeeID	Foreign	int
	Make		charvar(100)
	Model		charvar(100)
	Location		charvar(50)
	Game Name		charvar(100)
	Date		date
	Category		charvar(50)

	Serial Number		int
Employee	EmployeeID	Primary	int
	Badge Number		int
	Tech Name		charvar(50)
	Shift		int
Incident	IncidentID	Primary	int
	EquipmentID	Foreign	int
	EmployeeID	Foreign	int
	Incident Description		charvar(255)
	Incident Notes		charvar(500)
Solution	Solution ID	Primary	int
	EquipmentID	Foreign	int
	EmployeeID	Foreign	int
	Solution Description		charvar(255)
	Solution Notes		charvar(500)
Daily Activity	Daily ActivityID	Primary	int
	Description		varChar(500)

The design consideration outlines the various tables, attributes, keys, and data types needed to create a data warehouse for the information to be retained for future use. There will be a total of five tables that will individually store data on the equipment, the employees who interact with it, the incidents they encounter, and the solutions used to resolve those issues.

IV. Comprehensive Entity-Relationship Diagram (ERD)



Keith Bruyer
SWDV 691
Software Development Capstone

Northern Quest Resort & Casino
Machine Activity

Employee

EmployeeID	**integer(10)**
Badge Number	integer(10)
Tech Name	varchar(50)
Shift	integer(10)

Daily Activity

DailyActivityID	**integer(10)**
Description	varchar(500)
Date	date

Incident

IncidentID	**integer(10)**
Date	date
Incident Description	varchar(255)
Incident Notes	varchar(500)
EmployeeEmployeeID	*integer(10)*

Solution

SolutionID	**integer(10)**
Date	date
Solution Description	varchar(255)
Solution Notes	varchar(500)
EmployeeEmployeeID	*integer(10)*
IncidentIncidentID	*integer(10)*

Incident_Equipment

IncidentIncidentID	***integer(10)***
EquipmentEquipmentID	***integer(10)***

Equipment

EquipmentID	**integer(10)**
Serial Number	integer(10)
Make	varchar(100)
Model	varchar(100)
Location	varchar(50)
Game Name	varchar(100)
Category	varchar(50)
EmployeeEmployeeID2	*integer(10)*

Equipment_Solution

EquipmentEquipmentID	***integer(10)***
SolutionSolutionID	***integer(10)***

The above entity-relationship diagram (ERD) tells the story of how each table will relate to one another. Beginning with the employee table, it has a one-to-many relationship with both incidents and solutions, as a single employee will no doubt have more than one incident along with several different solutions. Equipment has a many-to-many relationship with the employee, incidents, and solutions tables as several pieces of equipment will interact with numerous employees and will have more than one incident and solution to these issues.

V. Relational Model

Employee			
EmployeeID	Badge Number	Tech Name	Shift
001	111111	Fred Flintstone	1
002	222222	Barney Rubble	2

Incident						
IncidentID	SolutionID	Serial Number	Badge Number	Date	Incident Description	Incident Notes
001	001	11111	111111	07/02/2022	xxxxxxx	xxx
002	002	22222	222222	07/03/2022	xxxxxxx	xxx

Solution						
IncidentID	SolutionID	Serial Number	Badge Number	Date	Incident Description	Incident Notes
001	001	11111	111111	07/02/2022	xxxxxxx	xxx
002	002	22222	222222	07/03/2022	xxxxxxx	xxx

Equipment							
EquipmentID	EmployeeID	Make	Model	Location	Game Name	Category	Serial Number
001	001	11111	111111	07/02/2022	xxxxx	xxx	xxxxx
002	002	22222	222222	07/03/2022	xxxxx	xxx	xxxxx

Daily Activity		
DailyActivityID	Description	Date
001	xxxxxxx	07/02/2022
002	xxxxxxx	07/03/2022

The relational model offers a rather simplified look at how each table depends upon the other. For instance, the employee table relates to each table within the database, whereas both the incident and solutions tables only depend upon each other and the equipment table.