New Plays Repertory Theater RDBMS

\*\*name

\*\*school

\*\*class

\*\*instructor

11/09/2015

*Introduction*

 In order to develop the Relational Database Management System for the New Plays Repertory Theater, some pre-requisites must be met. Ignoring the hardware requirements, software will require development of proper relation tables for each of the objects that can be manipulated in that system. This paper explores the requirements for the development of an RDBMS (Relational Database Management System) as required by the Theater.

**Ticket Sales**

The NPRT (New Plays Repertory Theater) tracks ticket sales for multiple reasons, and those reasons affect the data stored there. The multiple data types and methods for access mean that there must be solid links between the databases involved to create proper reports. These reports will be the output the users of the system see when they are accessing customer information, creating mailing labels, or checking to see if a seat is available. In order to support these activities, a certain amount of development is required in each area of the database.

**Transaction Records**

 Records that are kept specific to transactions often include data from all other databases in the system, and are usually the top-level on-screen report of any given sale. Transaction records include an index number, which is incremented whenever a new order is made, customer number, customer name, seat number, play or performance number, amount paid, and customer status if season or night-only, and possibly the date of purchase, and payment method. This record could also be the record that generates individual tickets, as display of the fields listed would be appropriate for such an application.

**Customer Information**

 Customer information must be maintained and all information must be accurate, with customer information being used in this case to identify the current customers for mailing and advertising purposes, a few additional items should be included in the customer information table. Customer information tables include the customer number, name, address, phone, customer type table output, and a “opt out” check box item to assure that customers who do not want to be mailed are not sent anything per the required laws regarding telecommunications and advertising.

**Seat Availability (500 total)**

Seating all by itself can have a number of characteristics, and to make system use easier, and develop a modular system with a relational structure, seating would have the following components: a seat ID number or seat number, seat type, section, and other pertinent data. A flag on the seat table should be included to flag if the seat is already reserved or taken. It is not necessary to include any information but the data directly related to the seats themselves in this table.

**Performance (7 plays per year, 20 performances)**

The number of performances for the Theater totals a minimal 140. So a total of new records a year expect to be entered into this system. The database used must be robust enough to allow these additions and still be functional. Data that would be included in the performance table would include of course, the performance ID, the performance date, performance name, actors in the performance, time of the performance, and possibly even the number of scheduled, current, or pending attendees based on the transaction record, the performance table should also include the number of available seats for a given performance, and be updated by each additional transaction record created.

**Customer Type (Season vs. Single Night)**

The customer types table holds the data regarding the season or non-season status of the customer. This table could be included in “customer information” as an enterable type, but it is better to keep all such selections programmatic instead of allowing the user to enter any data in the field. To create a drop-down menu easily, and allow easy reporting, the customer type table should have only the customer type identifications within it.

**Reporting**

A database is a long list of items, in order to locate the information that is needed quickly, a number of reports should be created for the RDBMS of the NPRT. These reports should be able to be printed, or displayed as necessary to the users of the system. Reports should also be automatic, for the most part.

**Available Seating**

 In order to determine the available seating for any given night, the available seating report should generate all the seats with the flag “reserved” when checked this seat is added to the list of unavailable seats in the report, and all others are listed as “available”. A relation between a given play and “available seats” will also need to be created to determine the total available seats for any given performance.

**All Plays and Actors**

 Querying all plays and all actors in those plays can be done by calling the performance table and outputting the results to a spreadsheet or database.

**Customers**

 To generate a report for all customers in the system, a query would need to be run on the customers’ record, and output of the customer table would be the result.

**Customer Mailing Labels**

To create customer mailing labels, a query would run to return all customers from the customer table database who do not have “opt out” flagged. These customers would then be queried for customer name and address only. These addresses then would be added to a spreadsheet for output to label generation software.

**Season Ticket Holders**

 In order to check on the number of season ticket holders (and possibly lock out a number of seats in preparation for them) a query could be run on the customer information table for those customers who have the flag, “season” ticked. This would allow easy definition of customers who are season ticket holders, and would make maintenance of season ticket information very simply the clicking of a box.

**Queries**

 In order to get the results that the user desires from the tables, there will be a need for queries that produce the proper results. See above for reporting guidelines (these are subject to change of course) that will allow the development of queries on the data in question. Queries can be created in Microsoft Access, as well as can the tables, to produce the proper forms for display. As seen in the picture here, this method produces robust and refined user interfaces while allowing the users to navigate and generate reports easily.

Figure

**Schema**

 One of the first things that must be considered before the actual development of the database truly begins, is how the user will access information. To this end, a flow diagram is often developed to illustrate the work-flow of the end-user as they navigate the system. This work-flow can be very detailed, or very simple, depending on the complexity and scope of the project in question. In this case, the scope of the project encompasses the end-user entering transaction data, printing mailing labels, and pulling various reports. The example below illustrates the typical schema associated with entering a credit card transaction and customer record.



Figure

References

No data references

Picture Reference

Fig 1. <http://www.bing.com/images/search?q=Sample+Access+Forms&view=detailv2&&id=69E30A13199A5CA494524B11036ECA702463C2F5&selectedIndex=2&ccid=LsMQlQz9&simid=608009177699976365&thid=OIP.M2ec310950cfd375d609a8221628b2970o0&ajaxhist=0>

Fig 2.

<http://www.bing.com/images/search?q=schema+enter+a+transaction&view=detailv2&&id=00DD845DA93D00EC4D87B7D6990A2F9228B58F4A&selectedIndex=0&ccid=kls89Miv&simid=608001652915048078&thid=OIP.M925b3cf4c8afa852e065b15d1a69f44fo0&ajaxhist=0>