CIS 451 FINAL PROJECT

Ticket Apprentice

Cameron Palk

December 9, 2015

URL

`http://ix.cs.uoregon.edu/~cpalk/TicketApprentice/`

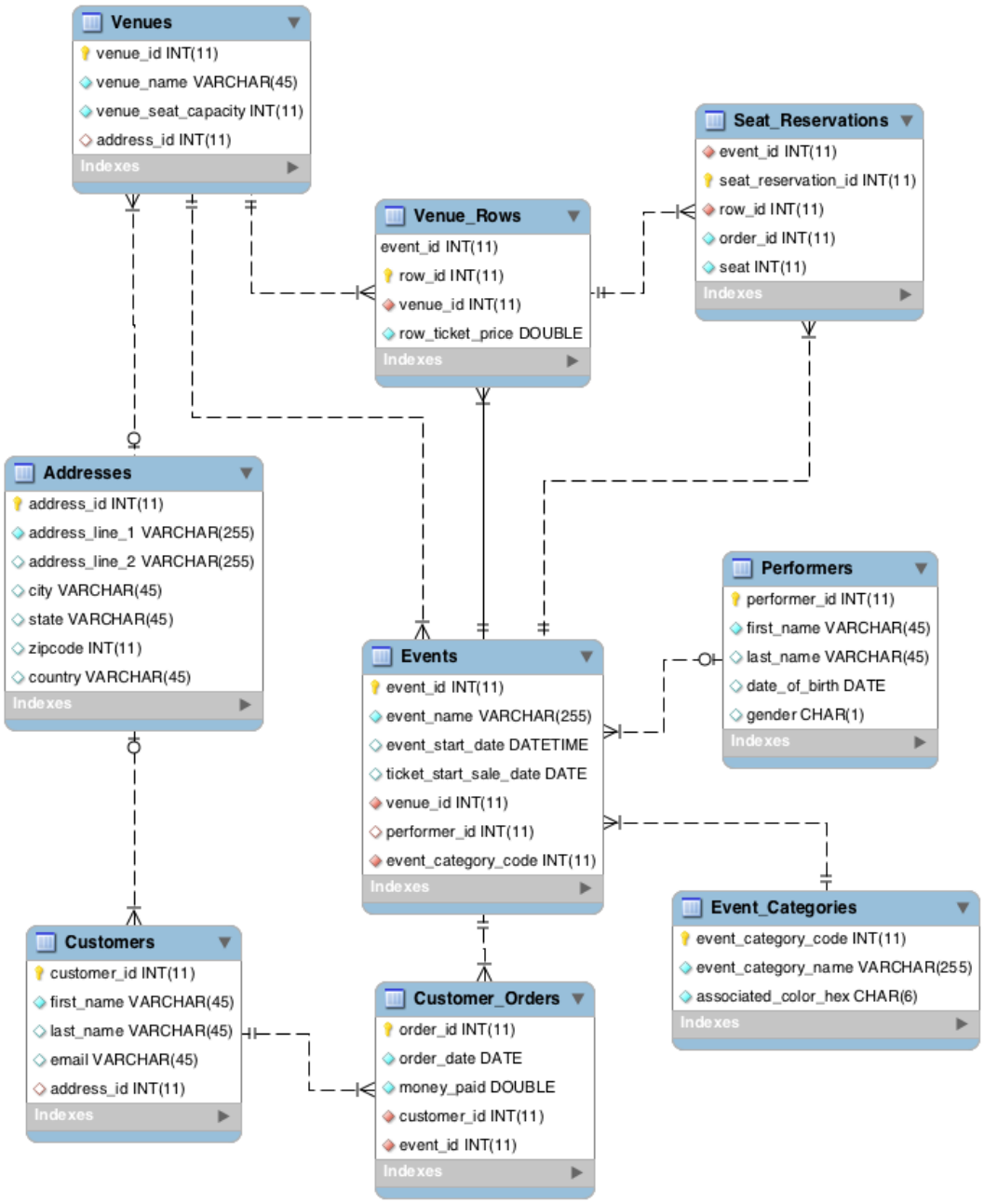
SUMMARY

My final project will incorporate a database to manage ticket sales for large events. It stores information about the Event such as where it is being held and then keeps track of purchases for each event. This entails keeping information such as who will be performing, where the event will be held (the venue), customer records to associate ticket sales, and the actual tickets.

The main focus of my project is to divide what each user sees based on if they are a customer or a manager. Manager will have the ability to create content such as events and venues which customers can then search for a purchase tickets for.

My ‘applications’ are embedded in pages to try and deliver a more concise view of information to the user.

LOGICAL DESIGN - EER DIAGRAM



PHYSICAL DESIGN

- **Venues**: Venue locations name, such as a stadium or theaters name. Also contains the seat capacity for the venue. Contains an address identifier that connects with an address in the Addresses relation.
- **Venue_Rows**: Used to hold pricing information and the number of rows allowed for an event.
- **Seat_Reservations**: Holds information about each order and what seats have been reserved, this prevents the same seats from being purchased again. These can be through of as the customers tickets.
- **Addresses**: Used to store addresses for both Venues and Customers
- **Events**: Information such as the name of the event, when it starts (date and time), when tickets go on sale in case they are not immediately available to purchase. Also connects what venue the event will be held at, the performer/group that will perform, and the type of event it is.
- **Event_Categories**: used to differentiate the types of events
- **Performers**: This is either a single performer or a group of performers. It requires a first/group name but all other information is optional as it wouldn't always make sense.
- **Customer_Orders**: These are the orders that customers make, it keeps track of how much money they spent, when they purchased their tickets, and connects to their Seat_Reservations entries (which is really their tickets).
- **Customers**: The customers information, only general information right now, such as their name, email, and address.

APPLICATIONS

MANAGER APPLICATIONS

- **Add Performers/Group**: This allows you to add a new performer as a manager.
  - Performer/Group name is required, no other fields are required, adds a new performer into Performers.
- **Add Venue**: This allows you to add a new venue
  - All fields required, adds a venue into Venues.
- **Add Event**: This connects the performers with the venue and creates a new Event that can immediately be seen by customers. Information such as ticket pricing and the number of rows of the venue you want to sell can be inputted here.
  - All fields required, used to create a event, inserts a new event into Events.
  - You will need a performer and venue created before creating your event.
  - Number of rows to sell tickets for will start rows at 0 and create entries for Venue_Rows for each row with the price also specified on this page.
  - Front row ticket price will be the most expensive tickets and Last row tickets will be the least expensive. These have a minimum of 0 but will create a linearly changing ticket prices through every row.
- **Customer Search**: You can search for customers by any part of their name, this can used to locate customers or get a list of customers.
  - Any part of the customers name, it will search for customers with the name %'search term'% using the LIKE sql operator.

CUSTOMER APPLICATIONS

- **Search by Venue**: Allows you to search for any available events by their venue.

    - Lists all the venues available which you can then narrow your event search by the requested venue
- **Search by Performer**: Allows you to search for any available events by their Performer/ Group
    - Lists all the performers available which you can then narrow your event search by the requested performer
- **Events Coming Up**: This sorts the upcoming events by their starting date and displays the 10 events that are coming up the soonest.
- **Look for Tickets**: This will search for tickets of the event selected and allow you to choose the number of tickets you want to purchase as well as the row that you want to sit at.
    - Once you find an event that you want to purchase tickets for you have to put in your customer information to connect your purchase with.
    - Then you select the row that you want to pay for and the number of tickets you want to buy on that row.
    - Seat_Reservations will have inserts for each ticket that you decided to purchase.
    - You will then be taken to a page describing your ticket purchase selection.

USERS GUIDE

Your flow will be fairly linear.

Create a new event and then purchase tickets for said event (with suggestions)

— This setup happens on the Manager side of the website —

1. Create an entry for the performer at your event (Ex. Name: Oregon Ducks Football Team)
2. Create a venue for the event to be held at (Ex. Name: The Alamodome, any information for actual address)
3. Create an event (Ex. Name: Valero Alamo Bowl: UO v TCU)
    I. You will now have a working event that can be found on the customers page

— This usage happens on the Customer side of the website —

1. Search by venue OR Search by performers OR just click [FIND EVENTS], it might also show up in the upcoming events depending on the start date that you input before
2. Click on the event you created -> click [Look for Tickets]
3. Input your information, select the number of tickets you want to purchase, and the row that you want to sit at.
4. View your purchased tickets

IMPLEMENTATION CODE

Implementation code in .txt files can be found at this link for all pages.

`http://ix.cs.uoregon.edu/~cpalk/TicketApprentice/implementation_code`

Also the ‘guest’ account has been setup with requested username/password.

CONCLUSION

This project was much more work then I had previously imagined. Breaking my mini-world into two sections (a manager side & customer side) turned out to be more challenging than I had anticipated. I have created a system where events can be created and linked to any venue and any performer. I tried to ask for information that I could see being useful later on. The issue was mostly time, had I been able to continue working there would have been much more analytics like I had originally planned. My MVP was essentially to have a working flow for creating events and then allowing customers to immediately see these them and purchase tickets. With more time I would have liked to add detailed views for managers to see what tickets were selling the best in terms of performer, venue, when the event was held, etc. I could easily see this project continue to grow and build useful information with the data collected.