Oracle Events System Demo

Copyrights

Open Source Project Repositories

Sourceforge:
Github:
Bitbucket:

Project Contributors

Anthony Harper

Oracle Version Support

Minimum version Oracle 19c.

Supported versions: Oracle 19c and 21c.

Conditional Compilation: Any features coded using updated syntax that is incompatible with version 19 must implement conditional compilation to support the earlier version.

Events System Business Entities and Processes

The events system is intended for use by venue managers, event organizers, ticket resellers and customers. This may involve multiple applications and web services that are out of scope for the database project.

Venues

Venues are locations that will host events. A venue has a maximum capacity based on actual space, seating available. Events cannot exceed venue capacity.

Venue managers will schedule events at their venue, define tickets and assign tickets to resellers in blocks of any quantity.

Resellers

Resellers are companies that work with the venue to sell tickets to customers. Resellers are paid a commission for all tickets they sell. Tickets for an event are assigned to resellers in limited quantities.

Customers

Customers will purchase tickets to attend events. Customers must provide a unique email address to record their ticket purchases.

Events

Events are scheduled by the day they occur. The initial system design does not allow multiple events on the same day, but may be enhanced to allow multiple events at a venue in different time slots. Events can also be scheduled as a repeating weekly series that occurs on a certain day every week for the duration of the series.

Event planned size may be at full venue capacity or less. A large venue can still plan a smaller exclusive event. Available tickets for an event are limited to the defined event size.

Event Tickets

Tickets are set up in different categories with prices and quantity available assigned by category. For example, an event may have 50 Backstage Passes, 150 VIP tickets, 200 Reserved Seating tickets and 1000 General Admission tickets. Ticket prices in a category can be changed at any time, customers should be charged the price that is currently in effect for the ticket category.

Ticket categories may change based on the event, sample categories might be anything: General Admission, VIP Seating, Backstage Passes, Platinum Sponsor, Early Purchase Discount, etc.

Ticket Assignments

After ticket categories are assigned, tickets can be assigned to multiple resellers in blocks of any size. A block of tickets would all have the same ticket category. Blocks of tickets assigned to resellers can be reassigned to other resellers, or more tickets in a category can be assigned to a reseller. A reseller can be assigned blocks of tickets in multiple price categories.

For example, the event manager could assign 50 VIP tickets and 100 General Admission tickets to one reseller and 50 VIP tickets and 200 General Admission tickets to another reseller. The event manager could then assign an additional 100 General Admission tickets to the first reseller.

Any tickets not assigned to resellers can be sold directly by the venue. If all tickets have been assigned to resellers, no tickets are available directly from the venue.

Ticket Sales

A customer may purchase tickets from resellers or directly from the venue based on availability. If sold by a reseller, the sale must record the reseller.

When tickets are sold, the availability for that price category and reseller/venue should be updated immediately. All sales should validate current availability at the time of purchase. This will prevent overselling a show.

All ticket sales are final. Future enhancements could support ticket refunds for cancelled events.

Tickets can be purchased by ticket price category. Tickets may be purchased in multiple quantities.

A ticket sale is defined as a quantity of tickets in a specific price category purchased by a customer from a reseller or directly from the venue at a moment in time.

Customers may make multiple purchases of tickets in multiple price categories..

Because ticket prices can be changed by the event organizers at any time up to the event, ticket sales must record the purchase price in effect at the time of sales.

Resellers are paid commissions on tickets they sell. Because ticket prices can be changed, each sales transaction made through a reseller should calculate the commission in effect at the time of sale.

Customer Tickets

Initially the system design defines a ticket sale as a quantity of tickets in a specific price category purchased by a customer from a reseller or directly from the venue at a moment in time.

Enhancing the system to print the actual tickets will require each ticket in a purchase to have a unique serialization. Ticket serializations must be defined at the time of purchase.

Tickets should be able to be checked on admission to an event to see if the specific ticket has already been used to enter the event.

Ticket Serialization

Ticket serialization is intended to prevent counterfeiting and aid with customer loss replacement. The ticket serialization identifier should include an abstraction of the customer identity as well. This will aid in identifying counterfeit tickets. If tickets were just serialized numerically it would be easy to produce fakes with apparently valid serial numbers for an event.

If a customer loses their tickets and requests replacements, the serial numbers for the lost tickets should be identified as invalid and new serial numbers issued with the replacements.

Reseller Commissions

Reseller commissions are initially set up at the reseller level. Future enhancements could support resellers negotiating a commission rate with each venue.

Initial system design will record commissions associated with ticket sales. Enhancements should include recording when commissions are paid to a reseller. These payments should be reconcilable with the ticket sales commissions calculated.

Events System Scope

The events system is intended to demonstrate an example Oracle data structure with a PL/SQL API to interact with all client applications. Client application design and coding are out of scope for the project.

Data Structures

The design includes all tables necessary to implement the business entities and processes.

Database Application Programming Interface

All APIs will use stored procedures organized into packages. The initial design only uses a single schema for data and API code, planned enhancements will separate code and data schemas.

The API will use stored procedures for all data creation and update processes. These stored procedures will contain internal validation logic to enforce business rules.

Because the basic API will contain all business logic, any extended API (reporting or web services) will call the basic API internally.

The basic reporting API will return all non scalar data to calling applications as ref cursors.

Extended APIs can expose these cursors as pipelined table functions for client applications that cannot consume ref cursors.

Web services APIs will expose all API functionality to web services using JSON or XML documents for all interactions with the web service. Creation and coding of the web services is out of scope for the project.

Basic API Functions

Create Venues. Update venue information.

Create Events.

Create Resellers. Set reseller commissions. Update reseller information.

Create Customers.

Create Ticket Categories with prices and quantities.

Create Ticket Assignments to resellers. Update ticket assignments.

Purchase tickets from a reseller.

Purchase tickets directly from the venue.

Reporting API functions

Provide reports of venue scheduled events.

Reports to support defining and updating ticket categories.

Reports to support assignment of tickets to resellers.

Reports to display ticket availability for resellers or venues.

Reports of ticket purchases by customers.

Performance reports for the venue comparing reseller sales performance.

Commission reports for each venue displaying monthly reseller commissions by event.

Other reports to be defined as needed.

Web Services Interfaces

Web services interfaces will expose the basic API and the reporting API to web service application consumers using a document based approach including JSON and XML.

The initial API design includes support for JSON documents for all system transactions and reports. Next phases will include support for XML documents.