**CS3550 Milestone 2 - Tyler Cazier**

**Resort Entertainment Live Event Access (RELEA)**

Contents

[OVERVIEW 2](#_Toc427170653)

[BUSINESS RULES 2](#_Toc427170654)

[BUSINESS RULES Satisfied by TRIGGERS 4](#_Toc427170655)

[-Name tr\_CollectUserData 4](#_Toc427170656)

[-Name tr\_PassPurchase 4](#_Toc427170657)

[-Name tr\_PassCancel 5](#_Toc427170658)

[-Name tr\_PurchaseCompletion 5](#_Toc427170659)

[-Name tr\_TicketCancel 5](#_Toc427170660)

[-Name tr\_TicketPurchase 5](#_Toc427170661)

[BUSINESS RULES Satisfied by STORED PROCEDURES 6](#_Toc427170662)

[-Name sp\_CalculateTax 6](#_Toc427170663)

[-Name sp\_CreateFolioTransact 6](#_Toc427170664)

[-Name sp\_ ApplyDiscount 6](#_Toc427170665)

[-Name sp\_UpdateDiscounts 7](#_Toc427170666)

[-Name sp\_UpdatePurchaseTotal 7](#_Toc427170667)

[-Name sp\_ViewBill 7](#_Toc427170668)

[-Name sp\_ViewEventGoers 8](#_Toc427170669)

[BUSINESS RULES Satisfied by USER DEFINED FUNCTIONS 8](#_Toc427170670)

[-Name fn\_CalculateTicketCancellation 8](#_Toc427170671)

[-Name fn\_CalculateTicketPrice 8](#_Toc427170672)

[-Name fn\_CalculatePassCancellation 9](#_Toc427170673)

[DISTRIBUTED DATABASE OR SERVER INFORMATION: 9](#_Toc427170674)

[ERD DIAGRAMS 10](#_Toc427170675)

## OVERVIEW

**RELEA –** RELEA is a management system that allows property guests and owners to book seating at live events that occur in the various TRAMS resort venues.

## BUSINESS RULES

**Portal Information**

* RELEA offers an online portal where users can create an account username/password.
* Users enter their TRAMS email address for their username. After entering their email, their user information is gathered from the TRAMS database, [see tr\_CollectUserData](#_-Name_tr_CollectUserData). If their email is not registered on TRAMS, they will be unable to connect their account. Their RELEA PersonID will be the same as their TRAMS PersonID.
* RELEA guests can browse TRAMS resorts and the resort’s scheduled live events from various performers.

**Seat Purchases**

* RELEA guests can purchase event seating on the website, see [tr\_TicketPurchase](#_-Name__tr_TicketPurchase).
* Events have a base price. This is the minimum cost for buying a ticket.
* Venue seats have a value rate. This rate is a percentage which will be used to calculate the ticket price for the seat.
* The ticket price is calculated by multiplying the seat rate percentage and the event base price. The minimum percentage will be 100%.
* In order to purchase a seat, RELEA guests must have an active reservation at the resort where the event is being held. The reservation must be during the time of the event. This will be enforced by the database.
* RELEA users will be billed for the ticket at the end of their reservation (It will be added to their reservation folio.) The database creates a new folio transaction in TRAMS for their ticket purchase
* If the user purchases a seat and cancels, see [tr\_TicketCancel](#_-Name__tr_TicketCancel), they will only be refunded 20% - 80% of their money depending on the time of cancellation. There is a one day grace period.
  + 12 days or more 80%
  + 11 – 7 days 50%
  + 6 – 2 days 30%
  + Day of 20%
* The amount refunded will be subtracted from the guest’s balance. A new negative folio transaction will be created.
* Users can pick up tickets or have them mailed. (If the user uses an all-inclusive pass for this purchase, they will receive a special ticket)

**Pass Purchases**

* RELEA guests can also purchase from the below passes on the website, see [tr\_PassPurchase](#_-Name__tr_PassPurchase).
  + Basic event pass. The pass is good for three event seats. The pass allows the user to purchase event seating for %60 of the original price. Three ticket discounts will be added to the user’s account.
  + All-access event pass (Includes performer memorabilia, preferred seating, and limitless, qualifying alcoholic beverages, and a discounted rate on events). The pass allows the user to purchase three event seats for %60 of the original price.

Three ticket discounts will be added to the user’s account.

* RELEA users can only have two passes at a time.
* RELEA users must have an active reservation before purchasing a pass.
* RELEA users will be billed for the pass at the end of their next active reservation (It will be added to their reservation folio.) Database creates a new folio transaction in TRAMS for every PASS and a number of available basic or all-inclusive seats are added to the user’s account.
* After a guest has purchased a pass, they must confirm their attendance. This confirmation will allow the guest to choose from available seating. This occurs as a regular seat purchase but, if the user has any ticket discounts on their account, the price will be reduced by %40.
* Just as with regular ticket purchases, guests must have an active reservation at the resort where the event is being held during the time of the event.
* If the user purchases a pass and cancels the reservation with the associated pass purchase, they will still be charged the price of the pass which will be included in the cancelled reservation folio.
* The user can cancel a pass purchase. There is a 20% penalty for cancellation, see [tr\_PassCancel.](#_-Name__tr_PassCancel)
* If a user cancels a ticket purchase that has been reduced with a pass purchase, the same penalties apply to the ticket cancellation. The ticket discount will, however, be re-added to the users account unless the cancellation is within 24 hours of the event.
  + 12 days or more 80%
  + 11 – 7 days 50%
  + 6 – 2 days 30%
  + Day of 20%
* All purchases will also have the appropriate tax applied using TRAMS database tax data see [tr\_PurchaseCompletion](#_-Name__tr_PurchaseCompletion). Many purchases can be made at once. All purchases, one to several, will be associated with a PURCHASE entity.

**Event Information**

* Events include a type (Comedian, Concert, Motivational, etc.)
* Events can also be adult only which will require ID at admittance.
* Performers are scheduled at the various event venues.
* Events require a date, a beginning time, ending time, performer, and event venue.

## BUSINESS RULES Satisfied by TRIGGERS

### -Name tr\_CollectUserData

**-Type** *On Insert INSTEAD OF*

**-Description** On INSERT into the REALEAUSER table, this trigger will collect the user’s data from the TRAMS database and will insert it into RELEA.

### -Name tr\_PassPurchase

**-Type**                *On Insert AFTER*

**-Description** On INSERT into the PASS table, checks to see if there is a valid reservation for the request. Also checks to see if the user has two active passes (or more than 3 discounts). If there is not a valid reservation or if there are already two active passes, returns an error and rolls back transaction. [sp\_ CreateFolioTransact](#_-Name__) is called to create a new transaction and [sp\_UpdatePurchaseTotal](#_-Name___1) to add the pass price to total purchase. Lastly, [sp\_UpdateDiscounts](#_-Name___2) is called to add the user’s discounts.

### -Name tr\_PassCancel

  -**Type**               *On Update AFTER*

**-Description** On UPDATE of PASS. Checks PassStatus. If it is ‘X’ for cancel, calls fn\_CalculatePassCancellation returns a negative %80 and [sp\_CreateFolioTransact](#_-Name__) to add the return money as a negative folio transaction.

### -Name tr\_PurchaseCompletion

**-Type**               *On Insert AFTER*

**-Description** Takes all items made in one purchase and calls [fn\_CalculateTax](#_-Name___5) to calculate the taxes. The tax is then added as new transaction via [sp\_ CreateFolioTransact](#_-Name__).

### -Name tr\_TicketCancel

**-Type**               *On Update AFTER*

**-Description** On UPDATE of TICKET. Checks TicketStatus. If it is ‘X’ for cancel, calls [fn\_CalculateTicketCancellation](#_-Name___4) to determine how much money the user will lose. [fn\_CalculateTicketCancellation](#_-Name___4) the calls [sp\_CreateFolioTransact](#_-Name__) to add the return money as a negative folio transaction and returns a discount to the user’s account if it is more than 24 hours before the event.

### -Name tr\_TicketPurchase

**-Type**               *On Insert INSTEAD OF*

**-Description** On INSERT into the TICKET table, checks to see if there is a valid reservation for the request. If not, returns error and rolls back transaction. Otherwise, calls fn\_CalculateTicketPrice to determine the ticket price which calls fn\_ApplyDiscount to help apply any available discounts and [sp\_CreateFolioTransact](#_-Name__) to add the purchase to the folio. [sp\_UpdatePurchaseTotal](#_-Name___1) is then called by fn\_CalculateTicketPrice to add the ticket price to total purchase. The amount returned by the above function is then added as the ticket price.

## BUSINESS RULES Satisfied by STORED PROCEDURES

### -Name sp\_CalculateTax

**-Description** Receives a dollar amount and a PassID. It finds the appropriate tax rate and calculates the tax of the dollar amount passed. It then calls [sp\_CreateFolioTransact](#_-Name__) to create a

**-Input Parameter(s)** @DollarAmount smallmoney, @TicketID bigint

**-Output Parameter(s)** None

### -Name sp\_CreateFolioTransact

**-Description** Takes a dollar amount and a TicketID or PassID and creates a new folio transaction in the TRAMS database. Depending on the type of purchase (ticket or pass), different data will be provided to the FOLIOTRANSACTION table

**-Input Parameter(s)** @TransactAmount smallmoney, @TicketID bigint, @TransactType char

**-Output Parameter(s)** None

### -Name sp\_ ApplyDiscount

**-Description** Receives a dollar amount and a TicketID and checks to see if the associated user has any all-inclusive then basic discounts. If so, the dollar amount is multiplied by 60% and returned. The TicketIsReduced bit is changed to true and the TicketIsPremium bit is changed to true. If there is not a premium discount, it checks for basic. If there is basic, the above still applies minus the TicketIsPremium bit. Discount applied or not the final amount is updated in the supplied ticket.

**-Input Parameter(s)** @DollarAmount, @TicketID

**-Output Parameter(s)** None

### -Name sp\_UpdateDiscounts

**-Description** Takes a PassID and adds three discounts of the right type to the associated user’s account

**-Input Parameter(s)** @ReleaUserID int, @PassID int

**-Output Parameter(s)** None

### -Name sp\_UpdatePurchaseTotal

**-Description** Takes a dollar amount and a PurchaseID and adds the dollar amount to the purchase total

**-Input Parameter(s)** @AmountToAdd smallmoney, @PurchaseID bigint

**-Output Parameter(s)** None

### -Name sp\_ViewBill

**-Description** Receives a PurchaseID and uses a cursor to scroll through all of the billable items of a given purchase including the applicable taxes.

**-Input Parameter(s)** @PurchaseID

**-Output Parameter(s)** None

### -Name sp\_ViewEventGoers

**-Description** Receives an EventID and uses a cursor to scroll through all users that have purchase tickets for a given event. It also displays how many tickets this particular user has purchased to that event.

**-Input Parameter(s)** @EventID

**-Output Parameter(s)** None

## BUSINESS RULES Satisfied by USER DEFINED FUNCTIONS

### -Name fn\_CalculateTicketCancellation

**-Description** Receives a TicketID and calculations the amount to be returned based on the date of the calculation. This value is returned.

12 days or more 80%

11 – 7 days 50%

6 – 2 days 30%

Day of 20%

**-Input Parameter(s)** @TicketID bigint

**-Returns\_\_\_\_\_\_\_\_\_\_\_\_?** smallmoney

### -Name fn\_CalculateTicketPrice

**-Description** Receives a TicketID and takes its associated SeatValueRate and EventBasePrice and multiplies these values. It then passes this value to [fn\_ApplyDiscount](#_-Name___3) which returns applies a discount, if available, and returns that value. The final returned value is passed to [sp\_UpdatePurchaseTotal](#_-Name___1) and, along with the TicketID, to [sp\_CreateFolioTransact](#_-Name__).

**-Input Parameter(s)** @TicketID bigint

**-Returns\_\_\_\_\_\_\_\_\_\_\_\_?** Smallmoney

### -Name fn\_CalculatePassCancellation

**-Description** Receives a PassID and calculations the amount to be returned. There is a 20% penalty for cancellation

**-Input Parameter(s)** @TicketID bigint

**-Returns\_\_\_\_\_\_\_\_\_\_\_\_?** smallmoney

## DISTRIBUTED DATABASE OR SERVER INFORMATION:

**-Type of DataSource** MSDASQL - DRIVER={SQL Server}

**-Connection String**

SERVER=titan.cs.weber.edu,10433;UID=tylercazier;PWD=\*\*\*\*\*\*\*\*;Initial Catalog=CAZIER\_TRAMS

**-UserID** tylercazier

**-Password \*\*\*\*\*\*\*\*\***

## ERD DIAGRAMS



