Final Project Database – FARMS Restaurants

This database is for the FARMS Restaurants. Each FARMS hotel has a restaurant in it that customers may make reservations to dine at. The restaurant wants to link up its FARMS hotel database to its FARMS restaurant database so that each restaurant that is in a hotel might be linked to that hotel through the databases.

The FARMS restaurant database will include their Billing, Menu, Reservation, Customer, Order, and restaurants. I will have the following business logic enforced upon the database.

Business Logic:

* The restaurant may add customer’s restaurant bill to the FARMS hotel bill
* A customer may only make a reservation at the hotel where they are staying
* A customer must currently be staying at the hotel to use the restaurant
* The restaurant must be able to insert customers into the Restaurants database from the Hotel database
* The restaurant must be able to insert customers
* The restaurant must be able to create reservations
* The restaurant must be able to add menu items
* The restaurant be able to create new orders
* The restaurant must be able to adjust the prices of menu items
* The restaurant must be able to update reservation statuses
* The restaurant must be able to calculate cancellation fees
* The restaurant must be calculate the earnings for a given night
* The restaurant must be able to get a menu for a given hotel
* The restaurant must be able to produce a bill for a customer
* The restaurant must be able to calculate total earnings for a night
* The restaurant must be able to get the current active orders
* The restaurant must be able to get a list of the currently reserved seats
* A reservation must always have a valid customerID associated with it
* When an order is placed a bill must be produced for the customer automatically
* If a reservation is cancelled then cancellation fees must be applied automatically if applicable
* An OrderItem cannot have a quantity that is zero or negative if that is attempted it must be replaced with a value of 1

User Defined Functions

* InsertCustomer(CustomerID, CustomerFirst, CustomerLast, CustomerEmail) – Inserts a customer row to the Customer table
* InsertReservation(ReservationTime, RestaurantID, CustomerID) – Creates a new reservation
* InsertMenuItem(ItemID, ItemName, ItemDescription, ItemPrice, ItemCategory) – Inserts a menu into the Menu table
* InsertOrder(OrderID, CustomerID, OrderMenuItems, OrderDate, RestaurantID) – Inserts an Order into the Order table
* UpdatePrices(MenuItemID) – Updates the prices of food on the Menu table
* UpdateReservationStatus(ReservationID, ReservationStatus) – Updates the status of a reservation

Stored Procedures

* GetMenu(HotelID) – Gets the taxes for an order
* ProduceBill(CustomerID) – Creates a bill for a given customer
* CalculateNightlyEarnings(RestaurantID) – Calculates how much money the restaurant has made for a given night
* CalculateCancellationFees(ReservationID) – Calculates the cancellation fee for a customer
* GetActiveOrders(RestaurantID) – Gets the schedule for a given employee
* GetReservedSeats(RestaurantID) – Finds which seats are reserved at a given restaurant

Triggers

* tr\_CustomerIDMustExist on Reservation – A reservation must have a valid customerID associated with it
* tr\_UpdateReservation on Reservation - If a reservation is cancelled then cancellation fees must be applied automatically if applicable
* tr\_GenerateBill on Order – When an order is created generate a bill for the customer
* tr\_InsertOrderItem on OrderItem – if the quantity of an order item is zero or less replace it with a 1

Milestone 1 questions

* Are there any flaws in the relational database schema you used?

None that I am aware of.

* Are there opportunities to combine relations without introducing redundancy? If so, indicate which, and if not, tell me there are none.

I do not see any relationships that could be combined without losing data.

* Are the tables in 3rd Normal Form?

Yes.

