**Dance school DB documentation (course project)**

## Description of schema

The topic of the project is a database which is designed to contain information for a dance school business processes. The main functionality that should be implemented is buying class subscriptions and booking rooms for classes.

To implement that task, it was decided to create the following entities:

**Customers**

(Stores information about customers who enroll in classes)

* CustomerID (PK)
* FirstName
* LastName
* Email
* Phone
* DateOfBirth
* Address
* Gender - gender of the customer
* JoinDate - date when the customer joined

**Instructors**

(Stores information about instructors who teach classes at the studio and book rooms for classes)

* InstructorID (PK)
* FirstName
* LastName
* Email
* Phone
* Specialization - area of specialization or expertise of the instructor
* HireDate - date when the instructor was hired
* Rating - average rating of the instructor

**Classes**

(Stores information about different types of classes offered at the studio)

* ClassID (PK)
* ClassName
* Description
* Level - skill level required for the class (e.g., Beginner, Intermediate, Advanced)
* Duration - duration of the class in minutes
* Price - price per class
* Category - category of the class (e.g., Yoga, Ballet)

**Subscriptions**

(Stores information about different subscription plans available to customers)

* SubscriptionID (PK)
* SubscriptionName
* Price
* Duration - duration of the subscription plan in months
* NumberOfClasses - number of classes included in the subscription plan
* Benefits - list of benefits included in the subscription

**CustomerSubscriptions**

(Stores information about subscriptions purchased by customers)

* CustomerSubscriptionID (PK)
* CustomerID (FK)
* SubscriptionID (FK)
* StartDate
* EndDate
* RemainingClasses
* TotalClassesAttended - total number of classes attended under this subscription

**Rooms**

(Stores information about rooms available for classes and rentals)

* RoomID (PK)
* RoomName
* Capacity - number of people the room can accommodate
* Location - location of the room within the studio
* Equipment - list of equipment available in the room

**RoomRentals**

(Stores information about room rentals by instructors)

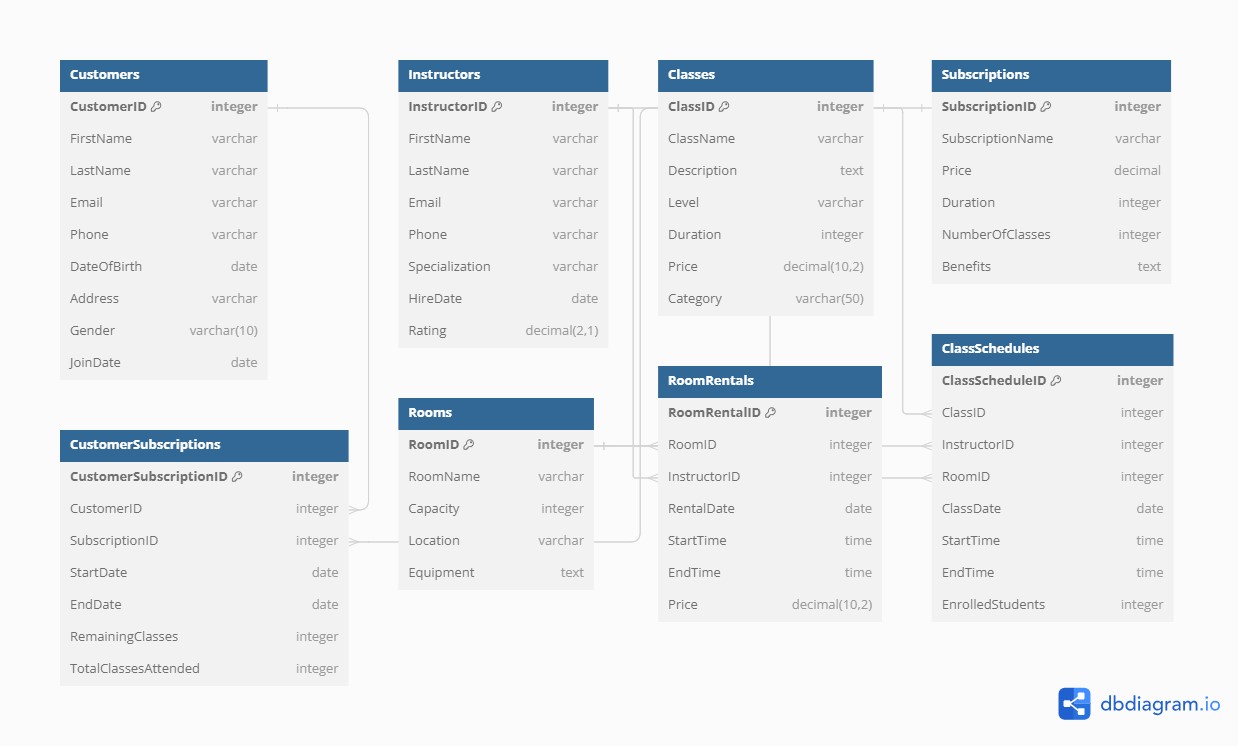
* RoomRentalID (PK)
* RoomID (FK)
* InstructorID (FK)
* RentalDate
* StartTime
* EndTime
* Price - price for the rental period

**ClassSchedules**

(Stores information about class schedules)

* ClassScheduleID (PK)
* ClassID (FK)
* InstructorID (FK)
* RoomID (FK)
* ClassDate
* StartTime
* EndTime
* EnrolledStudents - number of students enrolled in the class

Based on these entities an ER-diagram was created:



## Functions

**1. EnrollCustomerInClass Function**

**Purpose**:  
This function is designed to enroll a customer into a scheduled class if the customer has an active subscription with remaining classes. It decreases the count of remaining classes in the customer's subscription and increases the enrollment count of the class.

**Parameters**:

* p\_CustomerID INT: The ID of the customer attempting to enroll in the class.
* p\_ClassScheduleID INT: The ID of the class schedule the customer wishes to enroll in.

**Behavior**:

* The function first checks if the customer has an active subscription that has not yet expired and has remaining classes.
* If the customer has remaining classes, the function decrements the RemainingClasses field and increments the TotalClassesAttended field in the CustomerSubscriptions table.
* Additionally, it increments the EnrolledStudents field in the ClassSchedules table for the specified class schedule.
* If there are no remaining classes or the subscription has expired, an exception is raised notifying that there are no remaining classes in the subscription.

**Exceptions**:

* "No remaining classes in subscription": Raised if the customer does not have any classes left in their subscription or their subscription has expired.

**Example Usage**:

SELECT EnrollCustomerInClass(123, 456);

This would attempt to enroll customer with ID 123 in the class scheduled with ID 456.

**2. BookRoomForClass Function**

**Purpose**:  
This function books a room for a specific class on a given date and time, ensuring there are no overlapping room bookings.

**Parameters**:

* p\_ClassID INT: The ID of the class to be scheduled.
* p\_InstructorID INT: The ID of the instructor conducting the class.
* p\_RoomID INT: The ID of the room to be booked.
* p\_ClassDate DATE: The date on which the class is scheduled.
* p\_StartTime TIME: The starting time of the class.
* p\_EndTime TIME: The ending time of the class.

**Behavior**:

* The function first checks if the specified room is already booked during the given date and time slots.
* If the room is available, the function proceeds to insert a new record in the RoomRentals table and a corresponding record in the ClassSchedules table.
* If the room is not available due to overlapping bookings, an exception is raised.

**Exceptions**:

* "Room is already booked during the specified time": Raised if there is an overlapping booking that would conflict with the desired booking times.

**Example Usage**:

SELECT BookRoomForClass(101, 201, 301, '2024-07-01', '10:00', '12:00');

This would attempt to book room 301 for class 101 conducted by instructor 201 from 10:00 to 12:00 on July 1, 2024.

**General Notes:**

* These functions should be executed within a transactional context where possible to ensure data consistency in the event of a rollback.
* Proper access control should be maintained to ensure that only authorized users can execute these functions.

This documentation serves as a guide for database administrators and application developers interacting with your database, ensuring they understand the function purposes, parameters, and behaviors.

## Roles

**Admin Role:** The admin role should have the highest level of access, including the ability to manage roles and all database operations.

**Instructor Role:** Instructors should be able to view and manage their classes and schedules, and view but not modify customer information.

**Front Desk Role:** The front desk staff will need to manage customer subscriptions, enrollments, and basic room bookings.

## DWH Schema

**Sales\_Fact:** This fact table will contain information about class enrollments, subscription purchases, and other sales-related data.

* CustomerKey (from Customer dimension)
* InstructorKey (from Instructor dimension)
* ClassKey (from Class dimension)
* RoomKey (from Room dimension)
* SubscriptionKey (from Subscription dimension)
* ClassScheduleKey (from ClassSchedule dimension)
* DateKey (from Date dimension)
* TimeKey (from Time dimension)
* Price
* Quantity
* TotalAmount

**Room\_Rental\_Fact:** This fact table will specifically capture information about room rentals.

* RoomKey (from Room dimension)
* InstructorKey (from Instructor dimension)
* DateKey (from Date dimension)
* TimeKey (from Time dimension)
* Price
* Quantity
* TotalAmount

**Customer\_Dim:** Contains information about customers.

* CustomerKey (Primary Key)
* FirstName
* LastName
* Email
* Phone
* DateOfBirth
* Address
* Gender
* JoinDate

**Instructor\_Dim:** Contains information about instructors.

* InstructorKey (Primary Key)
* FirstName
* LastName
* Email
* Phone
* Specialization
* HireDate
* Rating

**Class\_Dim:** Contains information about classes.

* ClassKey (Primary Key)
* ClassName
* Description
* Level
* Duration
* Price
* Category

**Subscription\_Dim:** Contains information about subscriptions.

* SubscriptionKey (Primary Key)
* SubscriptionName
* Price
* Duration
* NumberOfClasses
* Benefits

**Room\_Dim:** Contains information about rooms.

* RoomKey (Primary Key)
* RoomName
* Capacity
* Location
* Equipment

**ClassSchedule\_Dim:** Contains information about class schedules.

* ClassScheduleKey (Primary Key)
* ClassID
* InstructorID
* RoomID
* ClassDate
* StartTime
* EndTime
* EnrolledStudents

**Date\_Dim:** Contains date-related information.

* DateKey (Primary Key)
* Date
* DayOfWeek
* Month
* Quarter
* Year

**Time\_Dim:** Contains time-related information.

* TimeKey (Primary Key)
* Time
* Hour
* Minute
* Second