# HACETTEPE UNIVERSITY
# COMPUTER ENGINEERING DEPARTMENT
# BBM 473 –DATABASE MANAGEMENT SYSTEMS LABORATORY
# PHASE 2

**Group Members:**Oktay UĞURLU -21627725, Tuna Aybar TAŞ -21627648, Koray KARA -21803682
**Subject:** Database Based Web and Desktop Application
**Due Date:** 09.12.2019

## GYM MANAGEMENT SYSTEM

### Project Overview

The 'Gym Management System'is designed to keep track of the information of customers, trainers and other systems in the gym and manipulate them.

### Used Database and Database Development Environment

MySQL and MySQL Workbench 8.0 CE are used in the project . MySQL is an open source SQL DBMS which is supported by Oracle Corporation and MySQL Workbench is native development environment for MySQL.

### Procedures

The reasons behind creating stored procedures in this database can be explained as ; providing modularity , making some needed executions faster , reducing network traffic .
There are 19 tables in this system's database and each table has own insert,delete,update procedures. This procedures consist of main SQL delete , insert and update querries as explained below;

insert_[table_name] procedure : This procedure is used for adding a new record to the specified table. " insert_[Table_Name]" statement defined as naming convention. The procedure takes corresponded table's attributes as parameter and each insert procedure includes SQL "insert into" command.

delete_[table_name] procedure : This procedure is used for deleting an existing record from specified table.The main feature that distinguishes this procedure from others is taking only uniquely defining attributes such as primary key or foreign key if needed.“delete_[table_name]” statement defined as naming convention and each insert procedure includes SQL “delete from” command.

update[table_name] procedure: update procedure is used for manipulating or updating records that already exist in database. This procedure takes new attribute values as parameters and updates corresponding table content. “update[table_name]” naming convention is used to implement and it includes SQL’s “update” querry and inside sets the corresponding attributes to new values.

**Views**

Views are used in system to hide some columns from users by ensuring them access to the view and not to the table itself. Also views below help creating simplified , abstracted and user-friendly interfaces in web and desktop application which militate in favor of user experience in system.

**Amount_of_customer:** This view is created to check out if the quota of a lesson is full or not.
In order to create a virtual table , the view selects LessonID , EnrollTimeDay and also uses aggregate function count(*) from Purchase table grouped by LessonID and EnrollTimeDay respectively.

**Customer_System_User:** This view created to access information about customers. Due to the fact that a customer is also a system user , the view includes a querry with join (customer and system_user table). Personal information belong to the customer can be easily accessed from authorised user such as admin without sending querries each time. That makes this view useful because it can be used in any kind of processes and requests that require customer information.

**Trainer_System_User:** This view created to access information about trainers .A trainer is also a system user so that the view includes a querry with join (trainer and system_user table). All features about trainer such as username,password,name,surname,salary can be easily accessed from an authorised user such as admin without sending querries each time. That makes

this view useful because it can be used in processes (for instance:increasing salary) and requests that require trainer information.

**Program_Exercises_SportTools:** Inside this view Program , Consist_of , Exercise , Make_with , Sport_Tools are combined to create a virtual table to see the specific fitness program , exercises and sets . (The content of the program in briefly). It is created because probably it will be used frequently in requests coming from customer. Whenever a customer desired to see the program it can be implemented easily from this view.

**Customer_BMIs :** This system enables customers to get specified exercise programs according to their BMI's which is a globally recognized rate to learn body type. The programs are given customers acording to their BMI value. In case there would be lots of individual querries to get BMIs , this view is created to get the BMIs easily. It selects UsernameID and BMI value of customers via making some basic multiply and division arithmetic operations.

**Triggers**

**checkCreditCardExpireDate**: It is an essential event which care about customer's membership. It is used for checking out whether if the customer's membership is ended or not. The event controls the expire date every day. It sets the customer's EndDate attribute to current timestamp where CreditCardExpireDate attribue <= current date.

**insertTrainerTrigger**: Since a trainer is also a system_user whenever admin intend to add a new trainer , before it should be added to system_user table. This is what the "insertTrainerTrigger" literally does.

**insertCustomerTrigger**: Since a customer is also a system_user whenever a new customer is intended to add the system ,before it should be added to system_user table. This is the purpose of using "insertCustomerTrigger".

**deleteCustomerTrigger**: As well as adding a customer process , the opposite procedure needs to be executed while deleting it. After deleting a customer from delete this trigger works and also the data deleted in system_user table.

**deleteTrainerTrigger**: Since a trainer also a system_user as well as customer,similar process needs to be done. After deleting a trainer from trainer table this trigger works and also the data deleted in system_user table.

**Enhancements in Project**

There are some enhancements made since the project proposal phase. The main reason behind the improvements are fixing bugs , providing customers to furthermore smooth experience in system and to get rid of redundancy.

- Customer, trainer and system user id's turned out to "username" attribute. That enables customer and trainer to create username by own. Also data type changed from "int" to "varchar" to provide more variety.
- The table Lesson's 'quota' attribute transferred to EnrollTime table . The main reason behind that change is allowing indivial lessons to have various lesson quotas.
- The attribute 'creditcardnumber' turned into 'customerusername' since the Purchase table associated with customer's username.
- An urlImage attribute added to Sport_Tools table to insert visual data about sport tools .
- Another important update made on tables system_user and customer . A disjoint isA relationship created to prevent ambiguity about usernames of customers and trainers.
- "CreditCardNumber" is replaced into Customer table as an attribute rather than being an entity due to the redundancy issues . Also type of "CreditCardNumber" is changed from 'int' to 'varchar(16)'.
- In previous report the errata 'expertise' corrected as 'profession'
- The type of attributes 'StartTime' and 'EndTime' in table 'Enroll_Time' turn into 'time' instead of 'datetime' cause there won't be any need to save the date of the day. So another redundancy issue is solved by that way.
- 'ID' attribute added to 'Lesson' table so that a genre of a lesson (boxing , judo etc) can be given by multiple trainer.Another advantage of this improvement is to make indexing easier. Another structural bug fixed and provided more realistic environment.