

ITCS496 PHYSICAL IMPLEMENTATION OF DATABASES
Semester I, 2023-2024

TERM PROJECT – Handout#1

GENERAL ISSUES

Throughout the remainder of the semester, you are required to develop an Oracle database application as a term project. The aim of the project is to practice actual database system development.

This and other project specification documents will serve as an essential guideline while you are developing your term project. So, please read it carefully and keep it throughout the semester. You can also use these documents as a *checklist* before submitting a report, i.e., at every major stage of the project, to see if you have satisfied the basic requirements of that particular stage. You should also check announcements on the Blackboard course page regularly to be informed on the project status.

The general issues about this term project are listed below:

- First, form project groups of **3-5 students**.
- Note that we are assuming that the design of the database system is over, and we are in the stage of the physical implementation of the system.
- Do a project demonstration During **Dec 25th, 2023 – Dec 31st, 2023**
- Submit a complete project (both a report and development code) by **Monday Jan 1st, 2024, at 23:59.**

Gym Management Database System

MyGYM is a gym located in Bahrain. You are to develop a gym management system (GMS) to help them improve the performance of the current situation and overcome the problems that arise nowadays. Currently, MyGYM uses manual system to record their members, fitness classes and other business information.

Gym Management System is designed to automate and streamline the management and administration tasks of a fitness center or gym. This system helps gym owners, staff, and members in various aspects of gym operations, including member management, class scheduling, attendance tracking, and billing.

Key Features:

1. **Member Management:**
 - Track member information, including name, address, contact details, emergency contact, and membership type.
 - Assign unique member IDs and track membership status (active, expired, suspended).
 - Allow for new member registrations and member profile updates.
2. **Class and Schedule Management:**
 - Create a database of fitness classes offered, including details such as class name, instructor, schedule, and maximum capacity.
 - Allow members to view and register for classes.
 - Implement a system to manage class schedules, cancellations, and substitutions.
3. **Attendance Tracking:**
 - Enable staff to record member attendance for classes and gym visits.
 - Generate attendance reports for individual members or classes.
 - Implement alerts for members with irregular attendance.
4. **Billing and Payments:**
 - Track membership fees, renewal dates, and payment history.
 - Generate invoices for members, including details of the charges.
 - Handle different membership plans (monthly, quarterly, annually).
5. **Employee Management:**
 - Maintain a database of gym staff, including trainers, front desk personnel, and management.
 - Assign roles and permissions to staff members.
 - Track employee attendance and performance.
6. **Reports and Analytics:**
 - Generate various reports such as membership reports, attendance reports, and financial reports.
 - Provide insights into class popularity, peak hours, and member engagement.
 - Support customizable reporting based on user requirements.

ER and database tables:

To get you started with of the project, assume that the designer of the database has come up with an appropriate ER diagram. A suggested set of tables is given below.

Members	Classes	ClassRegistrations	Staff	Attendance	Payments
member_id	class_id	registration_id	staff_id	attendance_id	payment_id
name	class_name	member_id	staff_name	member_id	member_id
address	instructor_name	class_id	role	class_id	amount
phone_number	schedule	registration_date	username	attendance_date	payment_date
email	max_capacity	Payment	password		
membership_type	cost				
membership_status					

Note that you may need to add more tables or add more fields to have a more logically related functioning database system.

System users:

1. Manager
2. Staff
3. Members

Give each user the right privileges to do their everyday work (not more and not less). Note that typical users and application administrators access the application only through its APIs. Therefore, they can change the data only by invoking package subprograms.

Sequences:

Most of the tables will require their own sequence to automatically generate unique primary keys. Make sure that you create such sequences and later use them in your code in INSERT statements. Add triggers if needed to enforce the generation.

Suggested Procedures (PL\SQL):

- **Monthly report**
This procedure has parameter for entering the year and month of the report. This procedure returns complete info about all the class instructors, the number of members enrolled in their classes and total profits for each instructor for that month.
- **Member services**
This procedure has parameter for entering the name of member. This procedure returns complete info about the member and all details of the classes they are enrolled in.
- **Add Member**
This procedure allows the employee to insert the details of a member in the system.
- **Class Status**
After entering the name of the class instructor, this procedure returns the details incoming classes (FUTURE), the number of members enrolled in the incoming classes and total expected profits from the incoming classes.
- **Class Registration**
This procedure allows the employee to insert a new member in a class into the system, if it's within capacity. The procedure needs to check that the class_id and member_id are valid. Otherwise, an error message must be given. You can implement and use the helper PL\SQL functions Check_member_id, and Check_class_id.

Tasks to be done

Each project team is required to develop GMS with varying levels of advanced features with varying levels of advanced features using Oracle APEX and PL\SQL.

Phase 0 (no credit): design the GMS, draw the ERD, write code to create the tables, some sample data to be stored in the table, the constraints, indexes, views, relating your design to components of the DBMS. Store all these in a script.

Purpose of phase 0: going through this project phase enables students learn use of DBMS and its components like SQL in building real life application.

Phase 1:

Part 1: PL\SQL:

Implement the five functionalities of the application in the suggested PL/SQL procedures and functions exploiting advanced features of the language.

Part 2: APEX application development:

Full Implementation of the database application for the three different users using Oracle APEX exploiting advanced features. Details of the minimum needed functionalities will be posted soon.

Purpose of phase 1: This project phase allows learning database application development with PL\SQL and Oracle APEX.

Note also that these are just examples of operations that they can do. You can add more operations if they make sense in such an environment.

Work to be submitted

1. SQL statements for (creating and populating the tables).
2. The PL/SQL scripts for the procedures, functions, and triggers.
3. All code written during development (SQL statements, Form triggers, Functions, Procedures, PL/SQL scripts, ... etc.). APEX Application.
4. A typed report that contains the following:
 - a) Title page that shows your name, ID and section number on the first page of the report.
 - b) Assumptions made by you while building the database application.
 - c) An ER diagram of the system.
 - d) ERD to relation schema mapping
 - e) Some screen captures of query/procedure results.
 - f) SQL commands to create and insert data into the tables along with the screen shots.
 - g) Screen captures of different Forms and Reports with explanation.

Grading Criteria

Your grade will be based upon the following components:

1. The completeness and quality of the database system.
2. Thorough analysis.
3. Clarity and organization of the final report.
4. Project demo/presentation (all students must be available during demo time).
5. Participation (i.e. each team member must know all of the components in the designed database system and can answer the questions)

Important Notes

1. You must submit a softcopy of your report on the due date and time on BB. You will be asked also to present a demo of your database system.
2. If you do not submit your report on the due date, there will be a penalty of four marks for submitting the work on the next day. NO report will be accepted after that and you will be given Zero in this assessment.
3. The project can be done in groups of 3-5 students.
4. Work copied from anybody else will be treated very seriously and F/0 will be given in the project.