Design and implement a database on ***Gym Management System Project***

**Project is divided into four parts -**

* **PART 1 (3%)** - Write a 1-2 page Statement of Work (SOW) to describe the need to create, design and implement the database that you propose for your project.
* **PART 2 (5%)** - Create requirement definition document and use ER Assistant, Visio, or SQL Developer Data Modeler to create Entity Relationship Diagram (ERD) for your project.
* **PART 3 (6%)** - Use SQL DDL statements to create your tables/views/triggers and other required database objects for you project. Also include queries (select statements on database catalog/data dictionary such as user\_objects and user\_tables) to demonstrate all objects are created successfully.
* **PART 4 (6%)** - Use SQL DML statements to insert sample data into your tables. Use SQL SELECT statements to query your data. This submission should be a final consolidated document which includes Project Part 1, Part 2, Part 3 and Part 4.

**PART 1 - Statement of Work**

Prepare and submit a 1-2-page statement of work (SOW) to describe the need to create, design and implement the database that you are proposing for your project. Within your SOW, the following minimum requirements must be met:

***Minimum Requirements:***

1. ***Overview/Executive Summary (10 points)***

Provide a one-paragraph executive summary of your overall SOW. This section should not only effectively introduce your database concept but also highlight the key objectives, goals, and benefits. Recommend that you save this section for last to be completed once sections 2 through 7 below are done.

1. ***Objectives of your database project (20 points)***

Provide a comprehensive set of SMART objectives (see ProjectMinds reference in the Project Learning Demonstration) that outlines the scheduled implementation of each of your database project objectives from the submission of this SOW through your final project submission (all four project parts) at the end of week 11. This will require that you thoroughly examine and plan your database project using the Project Learning Demonstration under Hands-On Resources. A quality set of objectives will include not only the minimum requirements but also a critical analysis of second and third-order objectives that must be met in order to accomplish the primary database project objectives.

1. ***Project Scope (20 points)***

As a compliment to the objectives section, provide a one-paragraph statement along with a bullet-list of in-scope and out-of-scope work that outlines your database project scope which must answer two key questions:

What work is within the scope of my project and will be done?

What work is outside the scope of my project and will not be done?

1. ***Database Goals, Expectations, and Deliverables (10 points)***

This section should be no more than one-paragraph and it should address the end-state of your database project in terms of its usability and function as well as what items will and are required to be delivered when your final project is submitted.

1. ***Database Benefits (10 points)***

Using your imagination and ability to critically think, this one-paragraph section should describe the potential benefits that the implementation and usage of your database will have on the business/organization/individual/etc. that ultimately uses it. Be creative and challenge yourself to find unique benefits beyond simply storing or querying/reporting on the underlying data.

1. ***Project Hardware and Software Tools (10 points)***

Provide a comprehensive hardware and software inventory which will be a detailed description of all related hardware and software technologies to be used throughout your database project. This must include client-side and server-side technologies. The purpose of this section is two-fold: reproducibility and consistency. This will include, but is not limited to: diagramming and design tools, development tools, office productivity, connectivity tools and access method, the DBMS system, client hardware, and server hardware.

1. ***Diagramming Tool Identified***

To further clarify, your SOW must accurately and completely identify the entity relationship diagramming tool that you will use to design your entity relationship diagram (ERD). This information must include not only the software name, but also the version and platform (i.e. Windows, macOS, Linux).

1. ***Database and Access Method Identified***

In addition to the identification of your client (i.e. database design/development hardware and software) and server (database instance/platform), you must also completely identify the method or methods by which you will be accessing the server from the client. As with the previous objectives, you must not only identify the software and/or method itself but also the version (if applicable) and protocol.

1. ***SQL Usage and Style Guide (5 points)***

Provide a detailed overview of the relevant SQL usage standards and styles that you will employ in your database project. This will include, but is not limited to, SQL statement structure for readability, script format, comment usage, and object naming conventions. This section must also include a brief description of how data definition language (DDL) and data manipulation language (DML) statements will be used in your project.

1. ***Student-Defined Sections and Additional Content (5 points)***

The purpose of this objective is for the evaluation and inclusion of your own unique and individual contributions, i.e. not one of the pre-defined minimum objectives, that further expand the depth of your SOW. Be creative and adventurous in your report by exploring additional content areas from domains such as database technology and project management.

1. ***One-to-Two Page Report Length (5 points)***

At a minimum, your SOW should be at least one or two pages in length. A report that is less than one-page will not have satisfactorily met the minimum objectives listed above. Exceeding this minimum requirement is not only allowed but encouraged based on the depth of content and analysis provided.

1. ***Report Presentation (5 points)***

In addition to the above objectives, your SOW will also be evaluated on the quality of presentation such as professional aesthetic, organization, logical flow, tonal consistency, and grammatical/spelling accuracy. For ease of identification, the sections of your SOW that address the qualitative goals above should be marked in some way with the goal itself.

**Ensure that the following questions are addressed when completing the requirements above:**

1. What is the business need and business problem that your database will solve?
2. What is the purpose of your database project?
3. What is the scope of your database project?
4. What will be achieved by implementing this database?
5. What benefits will the new database offer?

**Project Part 1 Deliverable**

1. SOW in Word or PDF: LastName\_FirstName\_SOW.[doc|pdf]

Your Project Part 1 SOW is worth 15% of your Lab Project which is 20% of your course grade. The lab project is cumulative meaning that you will submit this SOW again during project part 4 with all errors corrected, content expanded, and formatting updated as required.

**PART 2 - Requirements and ERD (Entity Relationship Diagram)**

Prepare and submit your requirements definition document and entity relationship diagram (ERD) as two separate attachments to your project part 2 assignment folder. Your requirements definition document must be submitted as a DOC/DOCX file and your ERD must be submitted using ER Assistant as an ERD file (or other supported diagramming format; see Project Learning Demonstration). Within your requirements definition document and ERD, the following minimum requirements must be met:

***Requirements Definition Document Minimum Requirements***:

1. ***Business Rules (30 points)***

 Provide a minimum of five business rules that will provide the framework upon which your ERD will be designed and built. While there is a minimum number of rules, the expectation is that you will have rules documenting all relevant aspects of your database project. To achieve maximum credit for this section, your business rules should be comprehensive addressing your entire model. Refer to the Project Learning Demonstration eTutorials reference to identify what your business rules must look like.

1. ***Entity and Attribute Description (30 points)***

For each entity, describe the entity itself as well as each of the attributes of the entity. You must also identify the primary key and any foreign keys. See the Project Learning Demonstration for examples of the entity and attribute descriptions. Note: if you are creating domain-specific attributes that are not common knowledge, you must provide a correspondingly more in-depth description of the entity or attribute.

1. ***Relationship and Cardinality Description (30 points)***

For each relationship, describe the relationship itself and also describe the relationship cardinality/minimum cardinality using the description format in the Project Learning Demonstration. To meet this requirement, you need only to describe the relationship that connects two entities one time.

1. ***Assumptions and Special Considerations (10 points)***

This section should contain detailed descriptions of assumptions you are making about the project and any special considerations such as deviations from the lab project requirements. For example, if one of your entities is a join table and cannot meet the five-attribute minimum you would document this consideration here. As for assumptions, these are used for you to clarify the underlying reality that defines the thing you are modeling in your project. There are times when modeling something from the real-world requires compromises to be made in consideration of specific, one-off scenarios. If you encounter a scenario like this, then add an assumption that occurrences of that scenario will be treated in a particular way to fit your database model.

***Entity Relationship Diagram (ERD) Minimum Requirements:***

1. ***Five Entities Minimum, Six Entities Maximum (15 points)***

The purpose of the five-entity minimum, six-entity maximum is to establish a relatively small baseline for you to keep the scope of your database project constrained. While the minimum entity requirement is fixed, the six-entity maximum can be exceeded so long as it is properly justified and documented within your requirements definition documentation.

1. ***Five Attributes Minimum (15 points)***

Each of your entities must have a minimum of five natural attributes which include the primary key. Foreign keys will not count towards the five-attribute minimum requirement. Any entity which has less than five attributes must have this documented in your requirements definition documentation.

1. ***Proper Relationships with Crow’s Feet Notation (25 points)***

Each entity must be properly related to another using Crow’s Feet notation. Ensure that you properly denote the minimum cardinality and place the correct side of the relationship on the appropriate entity. There must be no many-to-many relationships; if these exist in your design they should be resolved before submitting project part 2.

1. ***Unique Primary Key (10 points)***

Each entity must have a unique primary key properly designated in your diagram that can be a natural, surrogate, or composite key.

1. ***Foreign Keys in Proper Locations (10 points)***

All foreign keys should be placed on the appropriate side of the relationship according to your requirements definition documentation. Caveat: If using ER Assistant, be aware that foreign keys are derived implicitly from the relationship itself meaning that adding the foreign keys as actual attributes will result in diagram errors. Consider the use of an attribute and key naming convention such as *attributename\_FK* to designate your foreign key attributes.

1. ***ERD Matches Requirements Definition (20 points)***

Your diagram must accurately and completely translate the rules/requirements you previously outlined in the requirements definition document.

1. ***Proper Labels and Comments (5 points)***

All relationships must be properly labeled with the verb or activity that they represent. From the Project Learning Demonstration, the EMPLOYEE to ORGANIZATION relationship would be labeled with ‘works’. Any other assumptions or special considerations should be placed on your ERD as comments.

**Project Part 2 Deliverables**

1. Requirements Definition document in Word or PDF: LastName\_FirstName\_businessrules.[doc|pdf]
2. ERD in diagram specific format (i.e. .ERD) or Word/GIF/JPEG/PNG/PDF: LastName\_FirstName\_ERD.[erd|vsd|pdf|doc|docx|gif|jpeg|png]

Your requirements definition document and ERD are each weighted at 50% of the project part 2 total grade which is worth 5% of your overall course grade. In total, your final lab project is worth 20% of your overall course grade. The lab project is cumulative meaning that you will submit the requirements definition document and ERD again during project part 4 with all errors corrected, content expanded, and formatting updated as required.

See Project Learning Demonstration posted under Course Content/Hands-on Resources for step-by-step instructions.

**PART 3 - DDL (Data Definition Language)**

Create a SQL script containing your data definition language (DDL) statements to create your tables, views, triggers, and other required database objects for your project. Your script must also include queries to demonstrate that all objects are created successfully (i.e. selecting from the database catalog/data dictionary using user\_objects and user\_tables). Additionally, you must submit a separate document that contains the complete textual output from running your DDL script successfully. Your score will include the evaluation of this output report as well as a live, error-free expected run of your script in the environment. Within your DDL SQL script, the following minimum requirements must be met:

***Data Definition Language (DDL) SQL Script Minimum Requirements:***

1. ***Drop Statements for All Objects as needed (5 points)***

At the beginning of your script, ensure that all objects that are required to be dropped are properly dropped.

1. ***Create/Alter Statements for All Tables and Constraints (30 points)***

In an order that is appropriate for repeated executions, ensure that all tables and constraints are properly created and/or altered.

1. ***Create Indexes for Natural, Foreign Key, and Frequently Queried Columns (10 points)***

Unlike primary keys which have unique indexes created automatically, you must create indexes for ever natural key that is not included in the primary/composite key as well as all foreign keys and frequently queried columns. Note: you may not yet have queries built for your database yet but you will during project part 4. Keep this in mind as you will need to create indexes to support these queries.

1. ***Create a Minimum of Two Views (10 points)***

You are required to create at least two views though it is recommended that you create the number of views that is most appropriate to support your business requirements.

1. ***Create a Minimum of Two Sequences (10 points)***

You are required to create at least two sequences though if you are using surrogate keys this number will at least be equal to the number of entities that use said keys.

1. ***Create a Minimum of Two Triggers (10 points)***

You are required to create at least two triggers though the number of triggers should exceed this minimum if more than two sequences are deployed and to accommodate the automatic population of the auditing columns (see next requirement).

1. ***Describe the Business Purpose of your Views and Triggers (5 points)***

Using comments in your SQL script, before each view and trigger provide a description of what business purpose or function they provide. Recommend that you also begin incorporating this data back into your SOW and Requirements Definition document as necessary in preparation for submitting your consolidated lab report during project part 4.

1. ***Database Catalog/Data Dictionary Queries (5 points)***

Demonstrate the successful creation of all aforementioned objects by querying the database catalog/data dictionary; see the Project Learning Demonstration for examples.

1. ***Output Report (5 points)***

All output from the execution of the statements in requirements 1 through 8 must be recorded and saved into an output report.

1. ***Executable, Error-Free Script (10 points)***

The script you submit must fully execute and be error-free.

***Project Part 3 Deliverables***

1. DDL Script in SQL or TXT: LastName\_FirstName\_DDL.[sql|txt]
2. Script Output Report: LastName\_FirstName\_output.[doc|docx]

Note: the database catalog/data dictionary queries you write are considered DML statements not DDL. Despite this fact, for clarity the entire deliverable is classified as your DDL script. Please ensure that you include these queries at the bottom of your DDL script as required above.

Your Project Part 3 Data Definition Language (DDL) script is worth 30% of your Lab Project which is 20% of your course grade. The lab project is cumulative meaning that you will submit this SOW again during project part 4 with all errors corrected, content expanded, and formatting updated as required.

***Project Part 3 Best Practices:***

1. If you use Word to create your script, ensure you turn off the curly quotes as this will cause an error in Oracle. To do this you usually go to the "Options" area, find "Proofing", then "AutoCorrect" and "AutoFormat". Under "Replace", uncheck "Straight quotes" with "smart quotes".

2.  You may want to put script comments in your DDL file to document your work. If you do, ensure to indicate to Oracle that your notes are "comments" and not DDL commands. Oracle will error on non-commented notes.

Example of comments

* Single-line comments start with two dashes. Example: --Query problem #6
* Multi-line comments are enclosed with /\* \*/ Example:

/\*Query 4

The business value of this query is to show all customers in the CUSTOMERS table \*/

This is a reference you can use for understanding this concept: <https://docs.oracle.com/cd/B14117_01/server.101/b10759/sql_elements006.htm>

3. Output documentation - the quickest way to capture script output is to use the "Save" option on the lower, "output" window section of SQL Developer. The "Save" icon will ask you to save the file. You can email yourself the file or save to Google drive if you use this. If you use screenshots, please put them in a Word file.

4. Your DROP TABLE statements - put them in reverse order of the CREATE TABLE statements. If you have them ordered correctly, you won't need to use CASCADE CONSTRAINTS options.

5. One further note is to use: set echo on; at the top of your DDL file so your commands display before the output does. This will help you locate errors should they pop up run your DDL script twice. The 2nd run is to check for DROPs in the correct order.

----------------------------------

HINT- there is a way to generate DDL statements like DROP TABLE using SQL Developer. see <http://www.oracle.com/technetwork/developer-tools/sql-developer/export-intro-1-161239.html> for an example.  
-------------------------------------------------

Below are some helpful links on using the SQL Developer Data Modeler.  
SQL Developer Data Modeler Documentation  
Release 4.1  
SQL Developer Data Modeler User's Guide  
<https://docs.oracle.com/cd/E57998_01/doc.41/e57984/toc.htm>

Data Modeler Concepts and Usage  
<https://docs.oracle.com/cd/E57998_01/doc.41/e57984/data-modeler-concepts-usage.htm#cs_default>

Data Modeler Tutorial: Modeling for a Small Database  
<https://docs.oracle.com/cd/E57998_01/doc.41/e57984/data-modeler-tutorial.htm#library>

**PART 4 - DML (Data Manipulation Language) and Queries**

Create and submit your final consolidated lab report which will include the new requirements outlined below that are from project part 4 as well as all of the previous project parts (1-3). From project part 4, you must create and submit a SQL script containing your data manipulation language (DML) statements to insert your sample data into your database and your twenty queries (12 basic, 8 advanced). Within the consolidated report, you must include the complete textual output from running your DDL, DML, and queries successfully. Your final score will include the evaluation of the collection of output as well as a live, error-free expected run of your script in the environment. Within your DML and query scripts, the following minimum requirements must be met:

***Data Manipulation Language (DML) SQL Script Minimum Requirements:***

1. ***All Tables Populated with Minimum of 10 Rows (40 points)***

Unless a valid and approved exception exists within your requirements definition document, all tables must have at least 10 rows of sample data.

1. ***All Surrogate Keys Populated Automatically (40 points)***

All of your project sequences and triggers must be used to automatically populate your surrogate keys.

1. ***Separate DML for Different Tables with Comments (10 points)***

For readability, each block or grouping of DML statements for each table must be separated with an appropriate comment header with a blank line after the last statement in the group. Note: the last group does not require a blank line afterwards.

1. ***Executable, Error-Free Script (10 points)***

The script you submit must fully execute and be error-free.

***20 SQL Queries (12 Basic, 8 Advanced) Minimum Requirements:***

1. ***Query 1: Select all columns and all rows from one table (5 points)***
2. ***Query 2: Select five columns and all rows from one table (5 points)***
3. ***Query 3: Select all columns from all rows from one view (5 points)***
4. ***Query 4: Using a join on 2 tables, select all columns and all rows from the tables without the use of a Cartesian product (5 points)***
5. ***Query 5: Select and order data retrieved from one table (5 points)***
6. ***Query 6: Using a join on 3 tables, select 5 columns from the 3 tables. Use syntax that would limit the output to 10 rows (5 points)***
7. ***Query 7: Select distinct rows using joins on 3 tables (5 points)***
8. ***Query 8: Use GROUP BY and HAVING in a select statement using one or more tables (5 points)***
9. ***Query 9: Use IN clause to select data from one or more tables (5 points)***
10. ***Query 10: Select length of one column from one table (use LENGTH function) (5 points)***
11. ***Query 11: Delete one record from one table. Use select statements to demonstrate the table contents before and after the DELETE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)***
12. ***Query 12: Update one record from one table. Use select statements to demonstrate the table contents before and after the UPDATE statement. Make sure you use ROLLBACK afterwards so that the data will not be physically removed (5 points)***
13. ***Perform 8 Additional Advanced Queries (40 points)***

These queries are open for you to design but must be complex in nature such as including multiple table joins, sub-queries, aggregate functions, etc. These queries will not only be evaluated on their design and execution but also on their depth of complexity. Challenge yourself and leverage the techniques learned throughout the course to design your advanced queries.

***Consolidated Final Lab Report***

1. ***Statement of Work (SOW) (20 points)***

Revised, updated, and expanded as needed to include incorporation of all feedback given as part of the project part 1 evaluation.

1. ***Requirements Definition Document (20 points)***

Revised, updated, and expanded as needed to include the incorporation of all feedback given as part of the project part 2 evaluation.

1. ***Entity Relationship Diagram (ERD) (20 points)***

Revised, updated, and expanded as needed to include the incorporation of all feedback given as part of the project part 2 evaluation.

1. ***DDL Source Code Embedded (10 points)***

Ensure that all source code embedded in your report is formatted professionally and legibly.

1. ***DML and Query Source Code Embedded (10 points)***

Project Part 4 DML script + 20 SQL queries assignment; see above. Ensure that all source code embedded in your report is formatted professionally and legibly.

1. ***DDL, DML, and Query Output (20 points)***

Show output from the execution of all DDL, DML, and SQL queries. Ensure that all output is formatted professionally and legibly.

**Project Part 4 Deliverables**

1. Consolidated Final Lab Project in Word or PDF: LastName\_FirstName\_final\_project.[doc|pdf]
2. Updated DDL Script in SQL or TXT: LastName\_FirstName\_DDL.[sql|txt]
3. DML Script in SQL or TXT: LastName\_FirstName\_DML.sql

To clarify, you are required to submit your final scripts (DDL and DML to include your queries) as embedded into your consolidated lab report along with their output as well as separately in SQL or TXT format to be executed and verified as error-free.

Your Project Part 4 deliverables are worth 25% of your Lab Project which is 30% of your course grade. The DML section is worth 25%, the 20 queries 50%, and the consolidated final lab project 25% of your project part 4 grade. In total, your final lab project is worth 20% of your overall course grade (3% original SOW submission, 5% original requirements document/ERD submission, 6% original DDL submission, and 6% from the project part 4 submission).

For those struggling with triggers, here are a few resources.

<http://razorsql.com/features/oracle_add_sequence_trigger.html> has a good setup for how to use triggers to populate keys from a sequence. <https://www.techonthenet.com/oracle/triggers/before_insert.php> has some more information on how to use a BEFORE INSERT trigger (which is typical for manipulating a table row before the actual data is written into the table.

<http://plsql-tutorial.com/plsql-triggers.htm> has some information on how various types of triggers interact hierarchically.

<https://docs.oracle.com/database/121/LNPLS/triggers.htm#LNPLS723> is the official Oracle 12 guide to PL/SQL triggers, but it might be a bit dry/technical for novice users.

The instructor and TA should be able to run your source file as script “as-is” (without adjustments) to generate the same output as shown in your second file. It should run error free. The script should have standard Oracle comments for each SQL reflecting its business purpose, as well as which requirement item you are satisfying (i.e., --1. Select all columns and all rows from one table).

**Recommendations for project:**

* Keep your project simple and limited to five entities. Keep in mind two entities with a M:N relationship between them will be converted into three entities.
* Come up with a M:N relationship early on so that your final ERD will have three tables related to each other (for example the Enroll relationship between STUDENT AND COURSE will be translated into three tables - STUDENT links to ENROLLMENT which links to COURSE). When writing advanced queries, you can easily develop ones with 3 table joins.
* Decide on your database project and plan not to revise your ERD midway, although you are free to do it. From past experiences students had difficulties when they did this because of time constraints.
* Format your deliverable as if you are submitting it professionally in a work environment (comments, structured output, etc.). Assume the reviewer is not an IT professional.