

TERM PROJECT

Create and Design an Online Auction System: A database management system to manage bidding, auctions, and sales of goods and services on an online platform.

Note: For the ER diagram, please follow Elmasri and Navathe textbook's notation (in chapter 3 of the textbook). You may use generic drawing tools in office applications, or diagram tool such as MS-Visio. There are several online appropriate ER diagramming tools i.e <https://app.diagrams.net/>. Just in case you opt to use free online diagram tool, you may need to know that a complete ER diagram has to show Entities, relationships, attributes with primary key identified, etc.

Step 1: Identify appropriate entities, relationships, attributes, domains, and keys as well as all Specializations and generalizations:

i.e Identifying entities, relationships, attributes, domains, and keys:

- Identify the main entities involved in the online auction system, such as users, items, auctions, sellers, etc. (**Minimum of 7 entity types**)
- Determine the relationships between these entities
- Define the attributes for each entity (Must be as complete as possible)
- Determine the domains for each attribute, such as the data type, range of values, and constraints
- Identify the keys for each entity

Step 2: Producing the ER diagram:

- Create an entity-relationship diagram (ERD) to visualize the relationships between the entities and attributes.

Step 3: Mapping the ER diagram and normalizing the database:

- Map the ERD to a relational database model, by converting the entities into tables and doing the proper mapping between them
- Normalize the database by dividing the data into smaller, more manageable tables and enforcing referential integrity
- List the integrity constraints for each table, such as primary key constraints, foreign key constraints, etc...

Step 4: Defining the database in MySQL and inserting tuples:

- Create the tables in a MySQL database and define the columns with their data types and constraints.
- Insert at least 20 tuples into each table, using sample data that represents real-world scenarios.
- Verify that the data has been inserted correctly by executing queries to retrieve and display the data.

Step 5: Implementing transactions:

Implement transactions to ensure the integrity and consistency of the data in the database.

- Use transactions to execute a series of database operations, such as i.e placing a bid, processing a sale, in a single, atomic operation.
 - Use the transactional properties of atomicity, consistency, isolation, and durability (ACID) to ensure that the data is consistent and accurate, even in the event of a system failure.
-

For each submission you **MUST** include only **ONE PDF** file with the solution inside, including any images (i.e ERD diagram, tables, etc...). As well as a detailed explanation paragraph on the Step

In addition to the PDFs, for **Steps 4-5** you will have to submit a short **1 minute* screencast** video of the creation of the database, and inserting the 20 tuples into the tables created, as well as go over the queries for verifying that the data has been inserted and the transactions that have been created i.e to place the bid etc..

YOU MUST also submit the **.sql** file separately as an attachment which includes steps 4-5 so I can also test the correctness of the code on my end.

The project will be graded based on completeness **YOU WILL NOT** get full credits for Steps — **EVEN IF YOU THINK** they are correct and complete — if the approach/implementation is **not reasonable** (or are **not complete**) **in my view**

Submission will only be allowed through **BLACKBOARD**, **NO** email submission will be accepted or graded. You will find a link on blackboard for each **due date** to submit your Steps. Any email submissions will be disregarded even if it was submitted before the due date.

This project will provide a hands-on experience in designing, normalizing, and implementing a database management system to support an online auction system. It will demonstrate the importance of data integrity, referential integrity, and transactions in ensuring the accuracy and consistency of the data in a database management system.