**CMIS420 – Advanced Relational Database**

**Final Programming Project**

In this project, you will design a secure database with several tables using good relational database design techniques.

**Project Objectives:**

After completing this Project, you should be able to:

* Explain the purposes of advanced database objects including indexes, clusters, sequences, and views
* Develop and implement functions, triggers, and stored procedures/packages

**Project Requirements:**

1. Using the Oracle database engine, design and develop a relational database to store user information to support a large online shipping company. Information that needs to be stored includes user name, detailed mailing address, and shipping information including number of packages, weight, cost, destination, origin, and tracking information allowing users to determine where a package is, and when and who signed for the package once it arrives. Other business data should be present showing how much the shipment costs and invoicing capability to send bills to customers. Your design should be normalized and well-designed for scalability for additional customers and shipment locations.
2. Your design should allow for inserting, updating, selecting and deleting information
3. A trigger should be used to generate invoices once a customer reaches $200 in shipping costs.
4. An instead-of trigger should be employed to update and/or populate multi-table views.
5. You should include packages, indexes, views and other helpful database components that will assist you in your design.
6. Your scripts should work perfectly allowing the creation of all of the tables, triggers, views and other database components and provide test scripts that demonstrate the use of the system.
7. You do not need a GUI component but your scripts should demonstrate all of the functionality of the system.

**Deliverables:**

1. All sql scripts and run scripts and instructions for successfully building your database and testing your database.
2. An ERD diagram of your database design showing primary keys, foreign keys and other constraints.

3.      Well-written word document describing:

1. The database design
2. How to set up and test your database
3. The test data and results you used for your database,
4. Your approach, lessons learned, design strengths, limitations and suggestions for future improvement and alternative approaches, the most challenging aspects of developing this application.

**Projects submitted with evidence of plagiarism will be handled in accordance with UMUC Policy 150.25 - Academic Dishonesty and Plagiarism.**

**Format:**

**Documentation format and length:**

The word documentation describing and reflecting on your design and approach should be written using Microsoft Word and be no more than 5 pages in length and no less than 2 pages. The font size should be 12 point. The page margins should be 1 inch. The paragraphs should be with double line spacing. All figures, tables, equations and referenced should be properly labeled and formatted using APA style.

**Code format:**

         Header comment block,

         Appropriate comments within the code,

         Appropriate variable and function names, and

         Correct indentation.

**Grading:**

This activity is awarded 30% of the total grade in the course. In the gradebook, the total number of points will be set to 100. The project elements are assessed as follows:

|  |  |
| --- | --- |
| **Attributes**  | **Value**  |
| Project Design  | 20 points  |
| Project Functionality  | 40 points  |
| Test Data  | 20 points  |
| Approach documentation  | 15 points  |
| Approach documentation grammar and spelling  | 5 points  |
| **Total**  | **100 points**  |

The following grading rubric will be used to assign points to your project.

**Grading Rubric:**

| **Attribute**  | **Unacceptable**  | **Acceptable**  | **Distinguished**  | **Points**  |
| --- | --- | --- | --- | --- |
| Program Design  | (0-8) Points  - Database is not normalized  | (9-15) Points - Database is in 1st or 2nd normal form  | (15-20) Points - Database is in 3rd normal form  |    |
| Functionality  | (0-20) Points - Few requirements were fulfilled  | (21-30) Points -  Meets some requirements  | (31-40) Points-  Meets most requirements  |    |
| Test Data    | (0-10) Points- Little or no test data suite included | (11-17) Points - Some test data and results included but some vital functionality not tested  | (18-20) Points- Comprehensive test data suite included with results |    |
| Approach documentation  | (0-7) Points - Offers unsupported or irrelevant opinions - Does not consider alternatives - Rarely identifies strengths and weaknesses of the experimental design and of the data  | (8-12) Points - Supports opinions with argument, evidence, or proof - Considers alternatives but does not eliminate them -  Sometimes identifies strengths and weaknesses of the experimental design and of the data  | (13-15) Points - Conclusively argues/documents a point of view - Considers alternatives and shows how and why they should/should not be accepted - Identifies strengths and weaknesses of the experimental design and of the data  |    |
| Approach Documentation grammar/spelling  | (0-2) Points -Grammatical errors detract from the readability of the paper -Sentence structure (run-ons, fragments) needs improvement and impedes readability -Numerous spelling errors  | (3-4) Points -Few grammatical errors -Sentence structure needs improvement - Some spelling errors  | (5) Points - No grammatical errors -Sentence structure and diction are effective and clear - No obvious spelling errors  |    |