Project (120 Points) – COSC 5360 – Dr. Leonard Brown
Due: April 23, 2015 (at 11:59pm)

I. PROJECT DESCRIPTION

The management of the Three Seasons Apartment Complex would like you to build a database system to help manage their rental properties. The following information needs to be tracked:

The complex contains many buildings where each building has a unique number. Each building has a number of floors, an informal nickname, and a value, and a construction year. The complex recently built several new buildings and renovated the older ones. Thus, some buildings are new, and some have been renovated. If the building is renovated, the database should store the year it was renovated. If the building is new, the database should store the construction permit number.

Each building should be inspected every few years. Each inspection must correspond to exactly one building, and each building may have multiple inspections (each with a different date). Each inspection has a unique ID number, an inspection date, a result (passed or failed), inspector's full name, and a comment.

Each building contains multiple apartments, and every apartment must belong to a building. Each apartment has a three-digit number, a direction it is facing (N, S, E, W), a design scheme, a rental price, and a floor number. Each apartment number is unique only with respect to the apartments in the same building. The floor number, however, can be computed as the first digit of the three digit number. In addition, each apartment has a number of special features such as "balcony", "covered parking spot", "poolside", "fireplace", etc. Each apartment's design scheme has three components. The components are type of wood, wallpaper style, and color scheme. Each color scheme consists of a bathroom color, a carpet color, and a kitchen appliance color.

The rental price of an apartment is computed based on the floor plan. Each apartment must have exactly one specific floor plan, and several apartments have the same floor plan. Each floor plan has a letter, a number of bedrooms, a number of bathrooms, a base price, and an area measured in square feet. Together the floor plan letter and the number of bedrooms uniquely identify a floor plan (1A, 1B, 2A, etc.). The rental price of an apartment is computed as the base price plus $100 multiplied by the number of special features it contains. Thus, if an apartment has a floor plan with a base price of $625, and it has two special features (balcony and fireplace), its rental price would be $825.

Each apartment may be leased to one or more tenants. A tenant, however, may not lease multiple apartments (however, each tenant must lease at least one apartment, obviously). Each tenant has a name, a credit score, an income, and a collection of references. The tenant's name is composed of a first name, last name, and middle initial.

Tenants make requests to repair systems in and around an apartment. Each request has a unique request number, status (open or completed), request date, completion date, and a description of the problem. Some requests are modifications of other requests. A request may be a modification of at most one other request, and each request may only be modified at most once. When a request modifies another request, one request is the updated request, and the other is the original request.

All requests must be handled by one or more repair technicians. Each request may be handled by multiple technicians, and each technician may handle multiple requests. However, a single technician must serve as the primary point-of-contact for each request. When a technician handles a request, he or she must enter in the date the request was addressed, the amount of time spent that day, and a description of the work that was performed. Note that a technician may handle the same request multiple times (which is why each handling date needs to be recorded).

Each technician has a unique employee number, a name (consisting of first, middle, and last names), a phone number, an hourly rate, and a collection of specialties. The repair technicians are evaluated based on the quality of their work. Once a request is completed, the tenant making the request can evaluate the work on a scale from 1 (poor) to 5 (excellent). In addition, the tenant can add a text comment giving more information about the evaluation.

II. QUERIES

1. Construct SQL statements to create the tables and any views needed for all of the following queries, and implement them in Oracle.
2. Implement SQL statements in Oracle that will populate each table with at least 10 records to start the application.
3. List the value, nickname, and number of open apartments of each building that is new or renovated within the last three years. Sort the results by value then by number of open apartments (both in decreasing order).
4. List the permit number, number of floors, construction year, inspector name, and inspection date of inspections of new buildings that failed the inspection. Sort the list by inspection date listing the oldest inspection first.
5. List the nickname, value and total apartment area of each old building. The total apartment area for a building is computed of the sum of the areas of the floor plans of its apartments. Sort the list by total area in ascending order.
6. List the apartments that are not yet rented (currently have no tenants). For each such apartment, give its number, direction, building nickname, floor number, rental price, and color scheme. Sort the list by building nickname and rental price in decreasing order.
7. List the apartment number, number of tenants, type of wood, floor plan, and rental price of each apartment that has balcony as one of its special features. Sort the list by floor plan then number of tenants in decreasing order.
8. List the full name, credit score, apartment number, number of references, and number of requests made by each tenant. Sort the list by last name, then first name, then middle initial.
9. List the full name, credit score, apartment number, and number of references of the tenant that has made the most requests.
10. Of the apartments whose tenants have never made a repair request, list the apartment’s floor plan, number of bedrooms, number of bathrooms, rental price, and area. Sort the list by area in decreasing order.

11. List the request number, last name and apartment of the tenant that made it, and description of the request with the most modifications.
12. List the full name, hourly rate, and total number of requests of each technician. Sort the list by total number of requests in descending order.
13. List the name of the employee number, last name, and hourly rate of the technician with the most total modifications to requests he or she has worked as the primary technician.
14. List the total amount of money spent on all open requests. The amount of money spent on a request by a single technician is the hourly rate multiplied by the amount of time spent on the request.
15. List the apartment number and building number that have cost more money in repairs than they have made in the last year. Multiply the rental price by 12 to determine how much the apartment has made. Compute the cost based on the requests from that apartment. Specifically, calculate the hourly rate of the technicians multiplied by the hours they have spent working on the requests.
16. For each of the following threshold values of 4.0, 3.0, 2.0, and 1.0, display the number of technicians whose average evaluation score of his or her work is above the threshold. A technician should not be counted twice, meaning that the collection of technicians for the value 3.0 should not include the technicians whose evaluation score above 4.0.
17. List the specialties of the technician assigned to the oldest open request.
18. List the floor plan whose apartments have generated the most requests from its tenants.
19. List the primary technician's full name, phone number, and hourly rate, request number, completion date, and description of any request that was evaluated as 1 (poor).
20. Update the oldest open request to mark it as completed, and set today's date as the completion date.
21. For the open request with the most modifications, change the primary point-of-contact to the technician whose work has the highest average evaluation scores.
22. Change the color scheme for all open apartments (no tenants) in old buildings that have been renovated more than two years ago.
23. Remove the special feature that appears in the most apartments of newer buildings.
24. Remove the technician whose work has the lowest average evaluation score.
25. Remove all tables and views that were created in the above queries. No database objects should remain.

III. TASKS TO BE PERFORMED

Task 1: (*15 points*) Design an E-R diagram to represent the system described in Part I, and reduce your E-R diagram to a relational database. Provide a Data Dictionary that lists the names, types, definitions, and constraints of all attributes in each table.

Task 2: (*75 points*) Write example SQL statements for all of the queries defined in part II, and implement them in Oracle. Note that some of the queries may require multiple SQL statements.

Task 3: (*15 Points*) Write a Java program that initially prompts for a username and password and then uses that information to connect to an Oracle database. The program should allow the user to select and execute the queries in Task 2 on demand. This program should be able to connect to the Oracle database provided by IT. Thus, you should use the connection string provided in the sample program. Include a UML Class Diagram of the Java Program in the project, and follow the diagram with the source code.

Task 4: (5 points) Each member of the group must individually submit a document through Blackboard that describes the breakdown of tasks performed by all members of the group. For each person in the group, your document must list the tasks he or she completed. Example tasks include:

- Designed E-R Diagram
- Created E-R Diagram
- Designed Relational Tables
- Wrote Create/Drop Table Statements
- Wrote Create/Drop View Statements
- Populated Tables with Initial Records
- Solved Queries 1, 5, 7, 9, 13, 14, 15, …
- Designed Application Interface
- Wrote Following Classes/Methods: Main, DBInterface, …

Task 5: (10 points) The document in Task 4 must also include a score evaluating the contribution of each group member (except yourself). Each score **must** be between 1 and 10, inclusive (no scores of 0). Two people, however, **may not** receive the same score. Thus, if you are person A, and you have group members B, C, and D, you must submit a score for B (say 10), a score for C (say 8), and a score for D (say 3). Failure to complete this task will mean obtaining 0 points irrespective of the scores you receive from the other group members.

In addition, the document in Task 4 needs to clearly state which group member is submitting the final project electronically.

IV.GRADING

1. You must hand in a bound, paginated *project document* containing all of the tasks described in Section III. If groups were assigned, the document must include the names of all group members.

2. The *project document* must include a cover page that contains the following information: course name and number, semester and year, instructor's name, students' name(s), and project title.

3. The *project document* must include a Table of Contents. See Section V for more information.

4. You must also submit an electronic version of your project document through Blackboard. Your submission packet should contain the E-R diagram, the data dictionary, the queries and DDL statements in separate text files (see #5), and the source code in text files. Note that the source code and DDL/DML statements should be in text files and **not** MS Word files. Also please note that e-mailed versions of the project will **not** be accepted. If groups were assigned, your group evaluation document (see #6) should clearly state the group member that will electronically submit the document. Finally, the submission packet should contain a single PDF file that has a copy of all of above items.

5. In addition to the *project document*, you must electronically submit separate text files that contain **executable** SQL statements implementing all of the queries in Task 2. The file with the solution for query #*nn* should be called "*nn.sql*". For example, the file for query #4 should be "*04.sql*". These *.sql* files should contain ***only*** sql scripts. They should **not** contain any other text that prevents it from executing in the version of Oracle provided by IT. If I cannot execute the script successfully with the **start** command, *points will be deducted.*

6. (Group Project) Each member of the group must individually submit an electronic document for Tasks #4 and #5. This document will **not** be in the project submission document. Only one person needs to submit the electronic version of the complete project. The name of that person must be clearly stated in each group member evaluation document.

7. The project is due by the given time on the due date. Late projects will be accepted until **April 24, 2015 at 11:59pm** (with a 12-point penalty). Projects will not be accepted after that time. Both the bound *project document* and the electronic submission packet must be submitted on time to avoid the penalty.

V. SAMPLE TABLE OF CONTENTS

E-R Diagram	3
Data Dictionary	6
SQL Statements for Queries	18
Query 1	18
Query 2	18
Query 3	18
Query 4	19
Query 5	19
Query 6	20
Query 7	21
Query 8	22
Query 9	23
Query 10	24
Query 11	26
Query 12	27
Query 13	28
Query 14	29
Query 15	30
Query 16	32
Query 17	35
Query 18	36
Query 19	38
Query 20	39
Query 21	40
Query 22	41
Query 23	41
Query 24	42
Query 25	44
UML Class Diagrams.	46
Java Program	48