Winter - 2023

Course Project

☐ Due on:

- 6 Apr-23: Group presentation & Documentation submission

- 11- Apr-23: Group presentation

- 13- Apr-23: Group presentation

☐ Please write your name and student ID at the top of the word document

☐ Upload the document to Moodle

Consider designing a real estate J2EE Web based system. The system requires implementing the following set of technologies:
1) JSP
2) JavaServlet
3) JDBC
4) SQL as a programming database
5) XML
6) Tomcat Web Server
7) CSS
8) HTML
9) Pure Java Skills
10) And etc…

The system also requires building an Oracle database which keeps information about homes, appliances, agents, owners and locations in Michigan.
Typical information kept in the database includes

- **Homes:** FloorSpace, Floors, BedRooms, BathRooms, LandSize, YearConstructed. Homes can be further categorized into mansions, apartments, townhomes and condos. They will have all the properties of a home, but they will have distinguishing properties from homes and from one another. For example, mansions must have more than 6,000 sqft of floor space and more than 2 acres land size, and apartments cannot have more than one floor. The set of such homes are distinct - no two homes will be simultaneously an apartment and a mansion, for example.

- **Location:** A home can have an identification number and resides in a unique address. Notice that even though condos may have the same address, their unit number distinguishes them from each other. You need to account for this fact also. An address, on the other hand, while unique, its components may not. For example, a street may have several houses, a city will have several streets, and several zip codes. No two streets have the same name within a zip code. A city will have a changing population. Several cities will make up a county. All county names are unique.

- **Appliances:** Appliances have a model name or number, year, a maker, a name and a price. Homes will include numerous appliances made by different manufacturers. Appliances are identified either by their model number, or by the make and the name of the appliance.

- **Agents:** Agents are identified using a unique agent identity. Agents sell homes to different people called home owners. Agents sell the homes to an owner who will own the homes for some period of time, and can sell them to another person through an agent possibly at different price. Two agents cannot sell the same home at the same time. Agents receive commission (percentage) on the purchase price of a home. The rate of commission is determined by the real estate company the agent works for. She/he is allowed to work for many different companies such that the company will have an office in the city he/she sells homes.

- **Owners:** Home owners have a name, a unique social security number, number of dependents (family members), income, age and a profession.

Design the GUI system keeping in mind that users of this system may ask some of the following questions for a variety of reasons.

1) List all the homes owned by a given owner in a given city.
2) List all the homes that were sold more than once.
3) Find the most expensive home an owner ever bought.
4) Find all the homes that include all e appliances by the same maker.
5) Find owners who do not own the homes they used to own.
6) Find the total commissions earned by an agent. Assume that commission earned is on the purchased price of a home he/she sells.
7) Find people who own apartments as well as mansions.
8) List all the homes below a price in a given city.
9) List owners who own all the most expensive homes in a given city
10) Find homes that up for sale in a given city that meet certain buyer choices such as number of bedrooms, baths, etc

You also need to consider some of the following transactions:
1) Adding an agent into the database
2) Adding a new home to the database
3) Moving a home from available for sale list to the owned list
4) Making a person a home owner and consistently changing all related information.

The implementation should preserve all ICs
Implement all tables, constraints and queries using SQL.
Design a user friendly interface for populating the tables, and execute the following pre-fabricated queries.

What you should do: The deliverables

1) Submit final documentation on the due date.
2) Schedule a demo time before the due date.

Important Notes

1) For all phases, you need to submit only one report per group.
2) Write the identification for each member of the group on the cover page.
3) You are not allowed to submit handwritten reports.
4) You are allowed to work in groups of 4 or 5.
5) You should submit one final report for your group.
6) It will include implementation code, and sample executions of queries and transactions. Your project may not be a list of Oracle scripts which implement the queries and transactions. It should be similar to a commercial application with menus and interfaces.

Project Grading Scheme

1) Correct database design -- 10%.
2) Appropriate interface/forms design -- 20%.

3) Satisfactory implementation of queries -- 30%.
4) Satisfactory transaction implementation -- 20%.
5) Demonstration -- 10%.
6) Individual contributions -- 10%.
7) Project report -- pass/fail. Must pass this component.