INFS1200/7900 Information Systems

Assignment (30 Marks)

Part 1 Due: 18 April 2019 @ 5.00 PM

Part 2 Due: 24 May 2019 @ 5.00 PM

Assignment Part 1 consists of the design, which includes an Entity-Relationship diagram, mapping to the relational schema and further normalization of the relational schema if required. The necessary information for schema design is given in the Universe of Discourse (UoD) described below. This part can be done **individually or in a group of two students**.

In Part 2 of the assignment, a schema will be provided to be implemented in MySQL. The implementation will include creating tables, defining constraints, uploading or entering sample data, and writing SQL queries. The details will be released after the submission of the design part. This part is done **individually** by all students.

1. Part 1 Design

The following describes the Universe of Discourse for the assignment:

Housing websites have increased in popularity and are now one of the main ways people find places to live. In this assignment we will model a system similar to realestate.com.au or domain.com.au.

The system is required to store information on properties including their address (number, street, suburb, state etc), and a number of property features. Typically, there are either stand-alone properties e.g. houses, or properties that consist of multiple units e.g. apartments or townhouses. An apartment or townhouse will be identified by a number within the property (eg. Unit 1, 24 Main St). All properties will be described by a number of features such as the number of bedrooms, bathrooms, car parks and other features where relevant.

Real estate agencies are responsible for managing the sales and rentals of properties. Each agency must apply for an agency license and may have a number of branches. An agency will only have one branch in any given suburb and should have its contact details available.

Real-estate agents are responsible for managing individual properties. Each agent has a license number as well as personal information recorded. Agents list houses or individual town houses/apartments for sale or rental and the system must be able to track the sale and rental history of any listing. An agent works for one real estate agency at a time but are known to move between agencies from time to time.

Once a listing is made the agent may set up open houses to allow customers to view the property before applying for purchase or rental. The proposed system does not store any information on property sellers, buyers or tenants.

Based on the above UoD, you are required to undertake the following design tasks and then document them in the provided template. Refer to Part 2 Implementation – Views section to cross validate your design against expected outputs from the designed database and information system.

ER Diagram

Create an ER diagram using the notation taught in lectures to represent the conceptual schema described by the Universe of Discourse. Depending on your design choices, you may need to use the extended ER diagram.

State clearly any assumptions you make regarding your design approach on your ER Diagram. Please note that you cannot make assumptions to simplify or compromise the completeness of the Universe of Discourse. If there are any points that need clarification, in the specification of the Universe of Discourse as given above, you must first attempt to clarify them with your tutor and/or lecturer.

Mapping

Map the ER diagram created above to a relational schema. Document the final schema in the designated space in the provided template. Remember to include ALL your foreign keys!

Functional Dependencies & Normalisation

List all non-trivial functional dependencies in your relational schema. Then based on the FDs, identify any relations that are not in BCNF. Finally decompose any relations that are not in BCNF, to achieve BCNF.

2. Part 2 Implementation

The following describes the scope and requirements of the implementation in PHPMyAdmin. The schema to be implemented will be provided after the submission of the design part.

Tables and Constraints

Create a database in PHPMyAdmin called PROPERTY.

You need to implement the schema provided[1] by creating the tables and constraints. Your database should enforce basic constraints, such as:

- *Referential integrity*. Multiple referential integrity constraints can be extracted from the specification.
- *Domain*. Attribute values are restricted to the allowed data types.
- *Key* and *Entity* integrity constraints.
- *Semantic* constraints if any/as given in the Universe of Discourse.

Sample Data

Populate the database with enough meaningful sample data (at least 5 tuples per table) to allow us to test the functionality offered by your information system. This data can be made up or sourced from any websites you choose.

Views

Write the following queries in SQL in your PROPERTY database:

1. List all townhouses sold in the suburb *Annandale* in *2016*
2. Find the suburbs that have not had any sales since *2017*
3. Find the total number of apartments available for rent on *Parramatta Road*
4. Which street in the suburb *Camperdown* has the highest average rent
5. What is the total value of properties sold by *Poh Ling* from *Trusted Real Estate* since *2014*
6. Which suburb's properties have always been sold in less than 4 weeks
7. What is the average number of car spaces available for houses on *Pyrmont Bridge Road*
8. Which suburb(s) has had the highest increase in average rent between *2014* and *2016*

(Write at least two of the following challenging queries)

1. List the properties that have been sold by real estate agencies that are located in the same suburb as the property
2. Who is the most successful real estate agent in *2018*. (Hint: who has sold and rented properties of the greatest value)
3. List the agencies that have property listings in every suburb
4. Which real estate agency (or agencies) has most number of open houses available for inspection on 24 April 2018

Note: Be sure to have sample data that can generate 'some' result for these queries. You will not be assessed on the correctness of the data, but on the correctness of your query. You can build your queries progressively by building views on views.

[1] The schema will be provided after the submission for Part 1 has been completed

3. Assessment

Each part of the assignment is allocated 100 points which will be converted into 15 course marks.

Part 1 Design

ER Diagram	50 points
Mapping	30 points
Functional Dependencies	10 points
Normalization	10 points

Part 2 Implementation

Basic Implementation (Tables, Data Types, Primary Keys)	30 points
Foreign Keys	20 points
Queries 5 Marks each	50 points

This assignment targets the following assessment criteria of this course:

1. Analyse, extract and structure information system requirements from a variety of organizational contexts
2. Reason with the foundations of the relational data model to correctly undertake relational database design
3. Express queries using the SQL language to provide correct and secure retrieval of data from relational databases
4. Construct a small-scale information system in a relational database management system
5. Perform information systems analysis and design in a group setting

4. Submission

Submissions will be done via Blackboard. Marking will be done through an electronic marking tool called Gradescope, which will also be used for providing feedback.

Please use the supplied answer template for all answers. Your work must fit in the predefined sections or it cannot be marked

Submit your assignment electronically via the provided links on Blackboard under the Assessment folder.

For Part 1 Design, if you worked in a group, submission should be made by ONE team member only. Extra submissions will attract penalties.

For Part 2 Implementation, a second file containing your implementation export from PHPMyAdmin is also required. Use the export function in PHPMyAdmin to create a

.SQL file of your database. Submit this file via the “ SQL File Submission” link on Blackboard.

5. Collaboration and Plagiarism

Database design is best done in teams, not only to share the workload but also to share and exchange ideas and discuss design choices. Therefore, you are encouraged (not required) to work in a group of two. Students working in a group will submit only ONE copy of the solution, which can be done through the account of either member of the group. *Both students working in a group will be given a common mark.* Note that the implementation part of the assignment is done individually.

If you have any questions regarding acceptable level of collaboration with your peers, please see either the lecturer or your tutor for guidance. Remember that ignorance is not a defence!

The University has strict policies with regard to collusion and plagiarism. Penalties for engaging in unacceptable behaviour can range from cash fines or loss of grades in a course, through to expulsion from the University. You are required to read and understand the policies on academic integrity and plagiarism in the course profile and uq.edu.au/integrity.

6. Late Submissions

Late submissions will not be accepted at all, or if accepted, will incur a penalty of 20% reduction each date late. Students who believe they have sound reasons for late submission should refer to the course profile (section 5.3 on Late Submissions).