**ISS4014 – Database Systems and Web Integration**

**Chapter 02 – Activities and Homework**

Name: Logan Strong

Date: January 25th, 2024

**Chapter 02 Review Questions and Activities (5 points)**

* Respond to the following Chapter 02 review questions.
  1. (1 pt.) Discuss the importance of data models.

Data models are great for helping the designer, the application programmer, and the end-user to communicate by facilitating their interaction together.

6. (2 pts.) Consider the scenario described by the statement, “A customer can make many payments, but each payment is made by only one customer” as the basis for an entity relationship diagram (ERD) representation. Using the MySQL Workbench Design tool, draw the two table ER diagram and copy and paste the image below.

One to many (One customer for every payment)

A screenshot of a computer

Description automatically generated

11. (1 pt.) What is a relationship, and what three types of relationships exist?

A relationship is something that describes an association among entities. The three are: One to one, one to many, and many to many.

12. (1 pt.) Give an example of each of the three types of relationships.

One to One: one project has one project manager and visa versa, One to many: one customer can have many non joint bank accounts, one non joint bank account cant have many customers, Many to Many: a student can have multiple courses, and a course can have multiple students.

many

**Chapter 02 Database Table Load and ERD (5 points)**

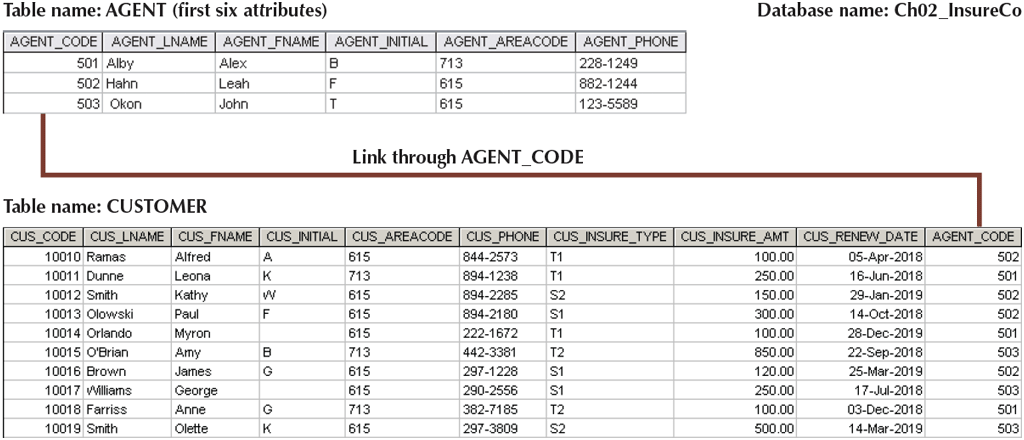
* Follow these steps:
  1. Using MySQL Workbench, open the Ch02\_Dealco\_MySQL.sql script.
  2. Create the ch02\_dealco database schema and run the script.
  3. Ensure no errors occur and correct any issues.
  4. Reverse Engineer the schema to create an ER diagram of the database.
  5. Copy and paste an image of the diagram below.

**A screenshot of a computer

Description automatically generated**

**Chapter 02 In-class/Homework Labs and Problems (20 points)**

Use the contents of Figure 2.1 (pg. 43) to work Problems 1 - 3.



1. (1 pt.) Write the business rule(s) that governs the relationship between AGENT and CUSTOMER. (Rember that there are two parts to each business rule)

Each customer must be assigned to an agent, an agent can have many customers. (one to many)

1. (1 pt.) Given the business rule(s) you wrote in Problem 1, Using Visio, create the basic Crow’s Foot ERD. Copy and paste the image of the diagram below.

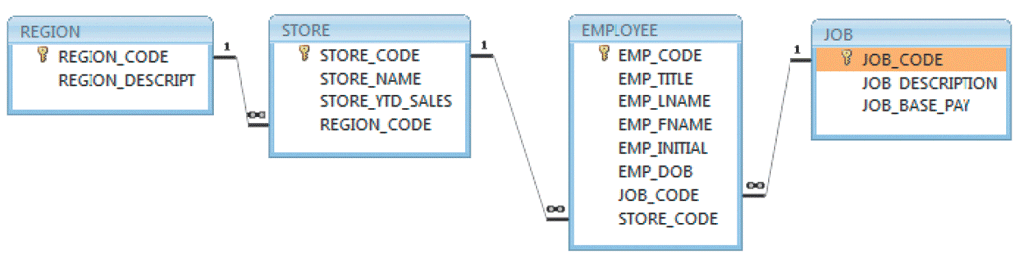


1. (1 pt.) Using the ERD you drew in Problem 2, using Visio, create the equivalent Object representation and UML class diagram. (Use Figure 2.4 on page 48 as your guide.) Copy and paste the image of the two diagrams below.





Use Figure P2.4 (pg. 64) to work Problems 4 - 5.



1. (2 pts.) Identify each relationship type and write all of the business rules.

One region can have many stores. Several stores are only in one region. (1 to many)

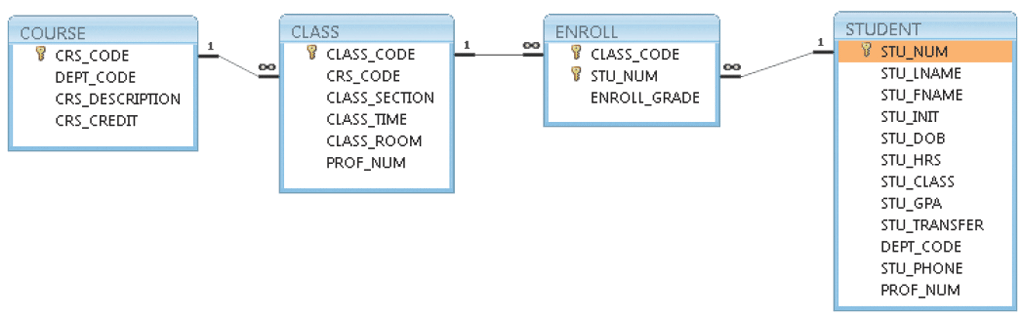
One store has many employees. The employees can only be employed at one store. (1 to many)

A type of job can have many employees, but an employee can only have one job. (Many to 1)

1. (2 pts.) Using Visio, create the basic Crow’s Foot ERD for DealCo.



Use Figure P2.6 (pg. 64) to work Problems 6 - 8.



1. (2 pts.) Identify each relationship type and write all of the business rules.

One course has several classes, several classes are under one course. (1 to Many)

One class can have several enrollments, you can only be enrolled in the class once. (1 to many)

Many students can be enrolled, a student can only be enrolled once (1 to Many)

1. (2 pts.) Using MySQL Workbench Designer or Visio, create the basic Crow’s Foot ERD for Tiny College. Copy and paste the image below.



1. (2 pts.) Using Visio, create the UML class diagram that reflects the entities and relationships you identified in the relational diagram. Copy and paste the image below.



1. Typically, a patient staying in a hospital receives medications that have been ordered by a particular doctor. Because the patient often receives several medications per day, there is a 1:M relationship between PATIENT and ORDER. Similarly, each order can include several medications, creating a 1:M relationship between ORDER and MEDICATION.
   1. (1 pt.) Identify the business rules for PATIENT, ORDER, and MEDICATION.

One patient can have many orders, an order must go to only one patient.

One order can have many medications in it, a medication can only be in one order

* 1. (1 pt.) Using MySQL Workbench Designer or Visio, create a Crow's Foot ERD that depicts a relational database model to capture these business rules. Copy and paste the image below.



1. United Broke Artists (UBA) is a broker for not-so-famous painters. UBA maintains a small network database to track painters, paintings, and galleries. A painting is painted by a particular artist, and that painting is exhibited in a particular gallery. A gallery can exhibit many paintings, but each painting can be exhibited in only one gallery. Similarly, a painting is painted by a single painter, but each painter can paint many paintings. Using PAINTER, PAINTING, and GALLERY, in terms of a relational database:
   1. (1 pt.) What tables would you create, and what would the table attributes be?

Id create a table for Painter, one for Painting, and one for Gallery. Attributes for painter would be the name of the painter. Painting would have the painting name and who painted it, Gallery would have a gallery name and number of paintings held.

* 1. (1 pt.) Using Visio, create Crow’s Foot ERD that depicts a relational database model of the tables you identified in part (a.) of this problem. For each entity, just fill out the table name. Copy and paste the image below



1. (2 pts.) Using the ERD from Problem 10, expand the ERD (still in Crow’s Foot notation) to include the attribute names of each table identified. Use appropriate naming conventions to name the attributes and identify an appropriate primary key for each entity. Copy and paste the image below.



1. (1 pt.) Using Visio, convert the ERD from Problem 11 into the corresponding UML class diagram.

