Exercise 2

Using Figure 1 as your guide, work Problems 1-2. The DealCo relational diagram shows the initial entities and attributes for the DealCo stores, located in two regions of the country.

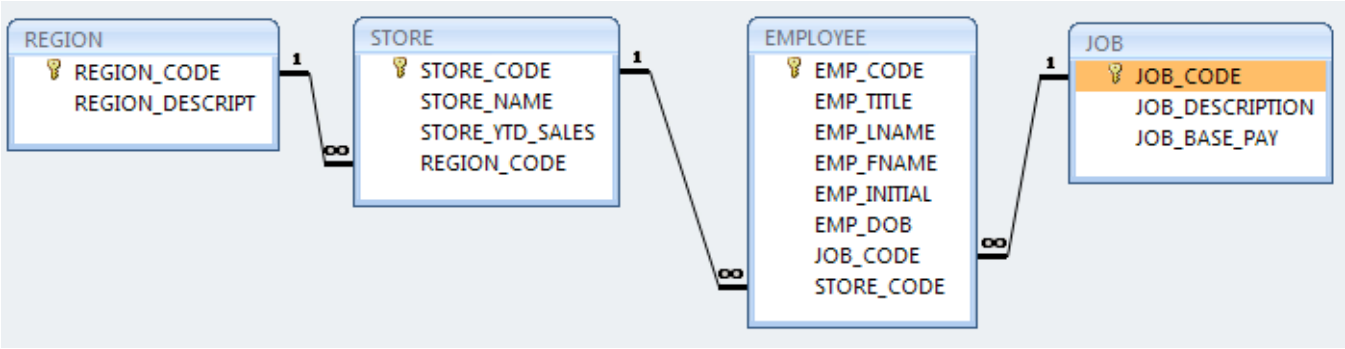


Figure 1 The DealCo relational diagram

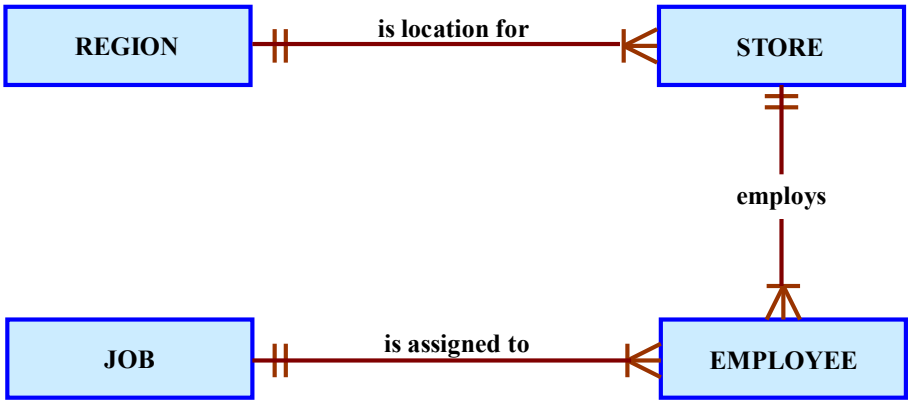
1. **Identify each relationship type and write all of the business rules.**

   One region can be the location for many stores. Each store is located in only one region. Therefore, the relationship between REGION and STORE is 1:M.

   Each store employs one or more employees. Each employee is employed by one store. (In this case, we are assuming that the business rule specifies that an employee cannot work in more than one store at a time.) Therefore, the relationship between STORE and EMPLOYEE is 1:M.

   A job – such as accountant or sales representative -- can be assigned to many employees. (For example, one would reasonably assume that a store can have more than one sales representative. Therefore, the job title "Sales Representative" can be assigned to more than one employee at a time.) Each employee can have only one job assignment. (In this case, we are assuming that the business rule specifies that an employee cannot have more than one job assignment at a time.) Therefore, the relationship between JOB and EMPLOYEE is 1:M.

2. **Create the basic Crow's Foot ERD for DealCo. (using different shapes in MS Word or you may draw one manually and scan to the word doc)**

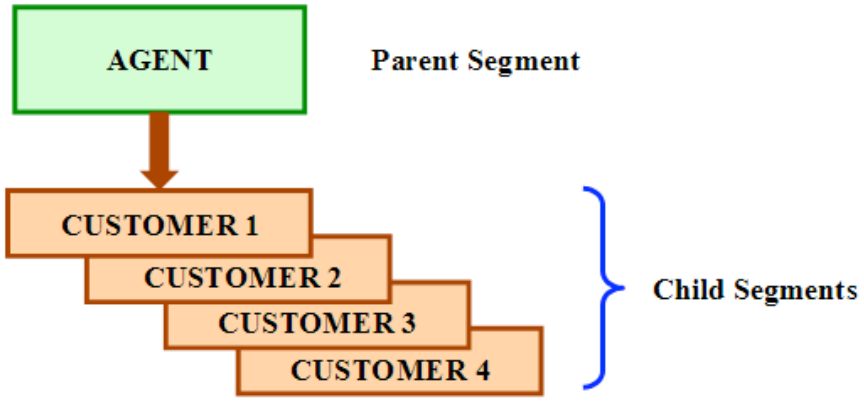


3. **Given the business rules below, draw a relational data model, a hierarchical data model, and a network datamodel respectively. (using different shapes in MS Word or you may draw one manually and scan to the word doc)**

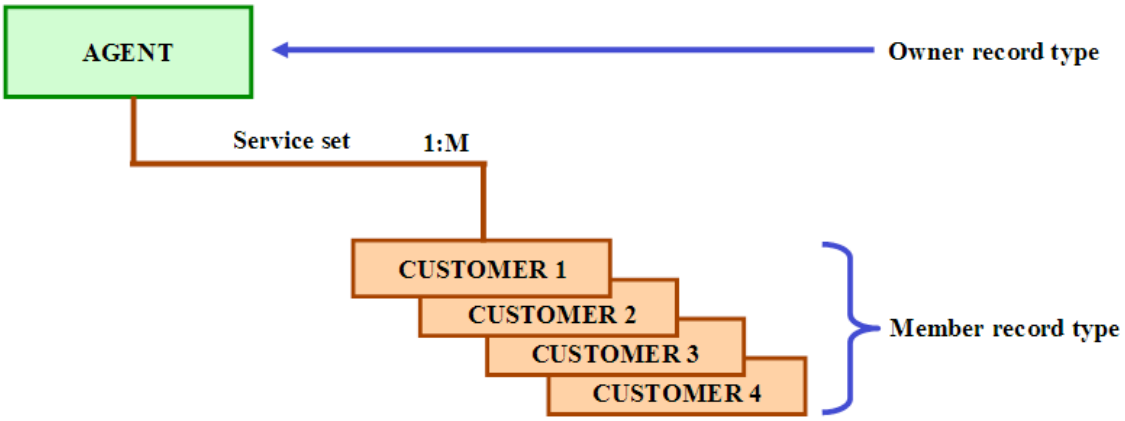
-- One agent can have many customers.
-- Each customer has only one agent.

Relational data model (E-R Diagram)

Hierarchical data model



Network data model



4. What is logical independence? What is physical independence?

Logical independence exists when you can change the internal model without affecting the conceptual model.

When you discuss logical and other types of independence, it's worthwhile to discuss and review some basic modeling concepts and terminology:

- In general terms, a *model* is an abstraction of a more complex real-world object or event. A model's main function is to help you understand the complexities of the real-world environment. Within the database environment, a data model represents data structures and their characteristics, relations, constraints, and transformations. As its name implies, a purely *conceptual* model stands at the highest level of abstraction and focuses on the basic ideas (concepts) that are explored in the model, without specifying the details that will enable the designer to *implement* the model. For example, a conceptual model would include entities and their relationships and it may even include at least some of the attributes that define the

entities, but it would not include attribute details such as the nature of the attributes (text, numeric, etc.) or the physical storage requirements of those atttributes.

- The terms *data model* and *database model* are often used interchangeably. In the text, the term *database model* is be used to refer to the implementation of a *data model* in a specific database system.
- **Data models** (relatively simple representations, usually graphical, of more complex real-world data structures), bolstered by powerful database design tools, have made it possible to substantially diminish the potential for errors in database design.
- The **internal model** is the representation of the database as "seen" by the DBMS. In other words, the internal model requires the designer to match the conceptual model's characteristics and constraints to those of the selected implementation model.
- An **internal schema** depicts a specific representation of an internal model, using the database constructs supported by the chosen database.
- The **external model** is the end users' view of the data environment.

You have **physical independence** when you can change the *physical model* without affecting the *internal model*. Therefore, a change in storage devices or methods and even a change in operating system will not affect the internal model.

The terms physical model and internal model may require a bit of additional discussion:

- The **physical model** operates at the lowest level of abstraction, describing the way data are saved on storage media such as disks or tapes. The physical model requires the definition of both the physical storage devices and the (physical) access methods required to reach the data within those storage devices, making it both software- and hardware-dependent. The storage structures used are dependent on the software (DBMS, operating system) and on the type of storage devices that the computer can handle. The precision required in the physical model's definition demands that database designers who work at this level have a detailed knowledge of the hardware and software used to implement the database design.
- The **internal model** is the representation of the database as "seen" by the DBMS. In other words, the internal model requires the designer to match the conceptual model's characteristics and constraints to those of the selected implementation model. An **internal schema** depicts a specific representation of an internal model, using the database constructs supported by the chosen database.