ANSWERS TO REVIEW QUESTIONS 5

1. **What are the basic modeling components of the E-R models? How would you (graphically) identify each of them?**

a. An entity An entity is represented by a rectangle containing the entity name. (Remember that, in E-R modeling, the word "entity" actually refers to the entity *set*.) A ***composite entity*** is identified through a diamond within a rectangle.
b. An attribute An attribute is identified by an oval and is connected to the entity through a:

- single line to indicate a single-valued, non-derived attribute
- dotted line to indicate a derived attribute
- double line to indicate multivalued attributes
- The attribute oval contains the attribute name. A primary key attribute is underlined.

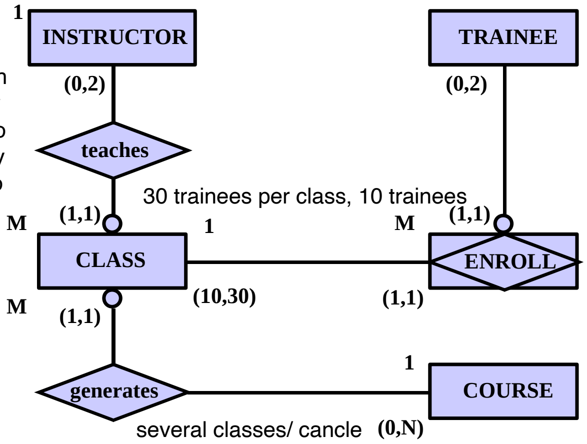
c. A relationship A relationship is indicated by a diamond-shaped symbol located between the entities whose relationship is being described. The diamond contains the relationship description, written in lowercase letters. The relationship description is usually an active verb such as *teaches, flies, writes,* etc. However, the relationship may warrant a more passive description to improve the E-R diagram's readability for humans: PILOT *is an* EMPLOYEE, INVOICE_LINE *is contained within* INVOICE, etc. Relationships are implemented through the use of foreign keys.

2. **The Hudson Engineering Group (HEG) has contacted you to create a conceptual model whose application will meet the expected database requirements for its training program. The HEG administrator gives you the following description of the training group's operating environment:**

The HEG has 12 instructors and can handle up to 30 trainees per class. HEG offers five "advanced technology" courses, each of which may generate several classes. If a class has fewer than 10 trainees in it, it will be canceled. It is, therefore, possible for a course not to generate any classes during a session. Each class is taught by one instructor. Each instructor may teach up to two classes or may be assigned to do research. Each trainee may take up to two classes per session.

Given this information, do the following:
a. Draw the E-R diagram for HEG.
b. Describe the relationship between instructor and course in terms of connectivity and cardinality.

Both questions, a and b, have been addressed in the following E-R diagram. Basically, three sets of relationships exist: A COURSE may generate one or more CLASSes, an INSTRUCTOR teaches up to two CLASSes, and a TRAINEE may enroll in up to two CLASSes. A trainee can take more than one class, and each class contains many (10 or more) trainees, so there is a M:N relationship between TRAINEE and CLASS. (We must, therefore, create a composite entity to serve as the bridge between TRAINEE and CLASS.) A class is taught by only one instructor, but an instructor can teach up to two classes. Therefore, there is a 1:M relationship between INSTRUCTOR and CLASS. Finally, a COURSE may generate more than one CLASS, while each CLASS is based on one COURSE, so there is a 1:M relationship between COURSE and CLASS. These relationships are all reflected in the following E-R diagram. Note the optional and mandatory relationships: to exist, a CLASS must have TRAINEEs enrolled in it, but TRAINEEs do not necessarily take CLASSes. (Some may take "on the job training.") An INSTRUCTOR may not be teaching any CLASSes, doing research instead, but each CLASS must have an INSTRUCTOR. If not enough people sign up for a CLASS, a COURSE may not generate any CLASSes, but each CLASS must represent a COURSE.



Each class is taught by one instructor. Each instructor may teach up to two classes or may be assigned to do research

Each trainee may take up to two classes per session

30 trainees per class, 10 trainees

several classes/ cancle

3. **What two actions are available to a designer when a multivalued attribute is encountered?**
The designer can split the multivalued attributes into its components and keep these components in the same entity.

Examples;

CAR color is decomposed into TOPCOLOR, TRIMCOLOR, and BODYCOLOR.
EMPLOYEE education is decomposed into HIGHSCHOOL, TWO-YEAR COLLEGE, FOUR-YEAR COLLEGE, and POST-GRADUATE.

The designer may also create a new entity composed of the multivalued attribute's components and link this new entity to the entity in which the multi-valued attributes occurred originally. This second option is especially desirable when the number of outcomes in the multivalued attribute is, for all practical purposes, unlimited. For example, employees classified as "technical" may have certifications in many different areas and at many different levels.

4. What is a derived attribute? Give an example.

Database designers who concentrate on elegant designs do not desire to store derived attributes in the database. Instead, they prefer that these derived attribute values are computed through appropriate algorithms when they are needed in a query. For example, a person's age may be calculated by using Julian dates to subtract the birth date from the current date and dividing the resulting number of days by 365. In other words, the attribute EMP_AGE is computed by

EMP_AGE = (DATE()-EMP_DOB)/365

Similarly, a sales clerk's total gross pay may be computed by adding a computed sales commission to base pay. For instance, if the sales clerk's commission is 1%, the gross pay may be computed by

EMP_GROSSPAY = INV_SALES*1.01 + EMP_BASEPAY

Or the invoice line item amount may be calculated by

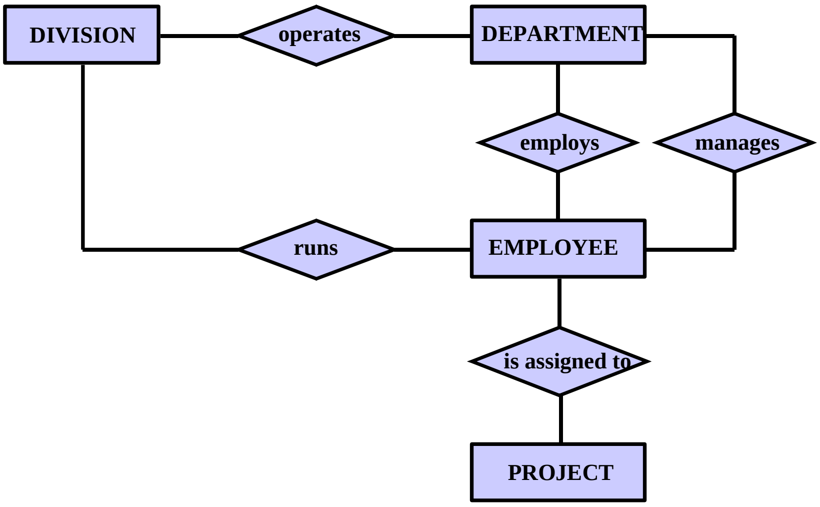
LINE_TOTAL = LINE_UNITS*PROD_PRICE

calculated attribute or a computed attribute, is an attribute that is derived or calculated from other attributes within the entity. It is not directly stored in the database but is determined based on the values of other attributes.

The problem with *not* storing derived attributes is that large databases tend to yield very slow queries when the derived attribute values are computed during the query. For example, if you need to get line item amounts for each product sold during a pass across ten million invoice line records, you will discover that the results are slow in coming! It is much less noticeable to the end-user when the required values are computed at the time of their generation and then to store the results in the table; the same query will execute much faster in this scenario.

ANSWERS TO PROBLEMS 5

The first three problems are based on the Chen E-R model below.

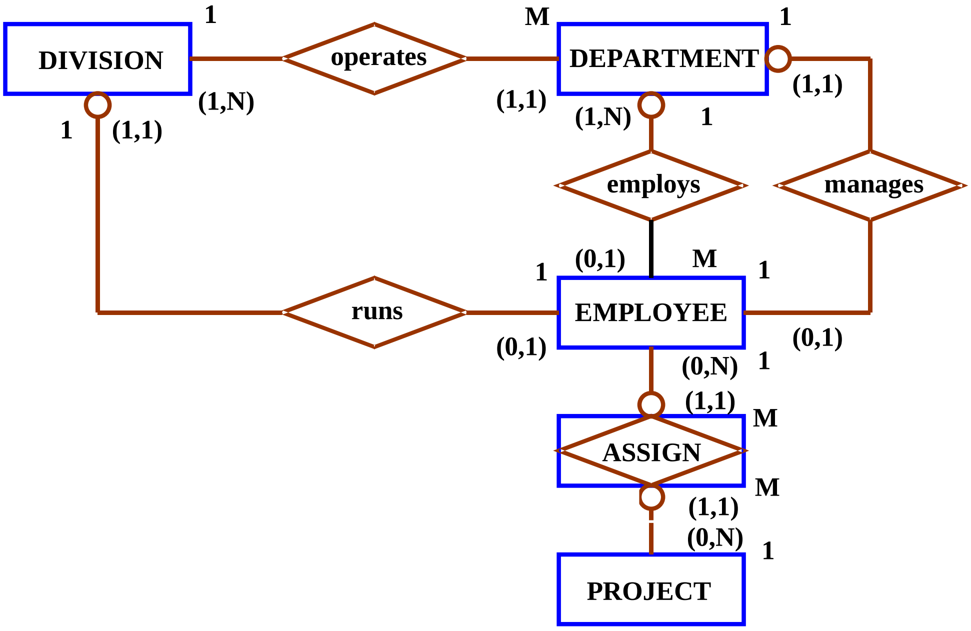


1. Use the following business rules to write all appropriate connectivities in the E-R diagram:
a. A department employs many employees, but each employee is employed by one department.
b. Some employees, known as "rovers," are not assigned to any department.
c. A division operates many departments, but each department is operated by one division
d. An employee may be assigned to many projects, and a project may have many employees assigned to it.
e. A project must have at least one employee assigned to it.
f. One of the employees manages each department, and each department is managed by only one employee.
g. One of the employees runs each division, and each division is run by one employee.

The answers to question 1 (all parts) are included in the E-R diagram that accompanies Problem 3.

2. Write all the cardinalities into the model.

have



- The ASSIGN entity is shown to be optional to the PROJECT, because EMPLOYEE is optional to PROJECT. This decision makes sense from a practical perspective, because it lets you create a new project record without having to create a new assignment record. If a new project is started, there will not yet be any assignments. There may be some new projects, without having any employee assignment.

- The ASSIGN entity is shown to be optional to the EMPLOYEE, because PROJECT is optional to EMPLOYEE. There may be some employees, without having any project assignment.

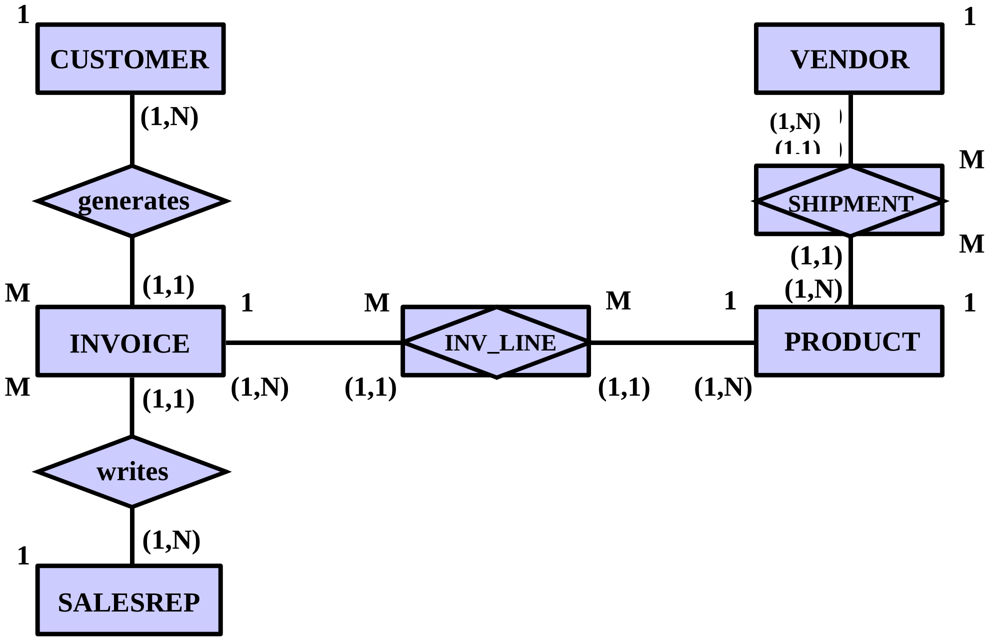
- The relationship expressed by "DEPARTMENT employs EMPLOYEE" is shown as mandatory on the EMPLOYEE side. This means that a DEPARTMENT must have at least one EMPLOYEE in order to have departmental status. However, DEPARTMENT is optional to EMPLOYEE, so an employee can be entered without entering a departmental FK value. If the existence of nulls is not acceptable, you can create a "No assignment" record in the DEPARTMENT table, to be referenced in the EMPLOYEE table if an employee is not assigned to a department.

- Note also the implications of the 1:1 "EMPLOYEE manages DEPARTMENT" relationship. The flip side of this relationship is that "each DEPARTMENT is managed by one EMPLOYEE". (This latter relationship is shown as mandatory in the ERD. That is, each department *must* be managed by an employee!) Therefore, one of the EMPLOYEE table's PK *values* must appear as the FK *value* in the DEPARTMENT table. (**Because this is a 1:1 relationship, the index property of the EMP_NUM FK in the DEPARTMENT table must be set to "unique."**)

- Although you ought to approach a 1:1 relationship with caution – most 1:1 relationships are the result of a misidentification of attributes as entities – the 1:1 relationships reflected in the "EMPLOYEE manages DEPARTMENT" and "EMPLOYEE runs DIVISION" are appropriate. These 1:1 relationships avoid the data redundancies you would encounter if you duplicated employee data – such a names, phones, and e-mail addresses – in the DIVISION and DEPARTMENT entities.

4. Create an ERD model, using the following requirements.

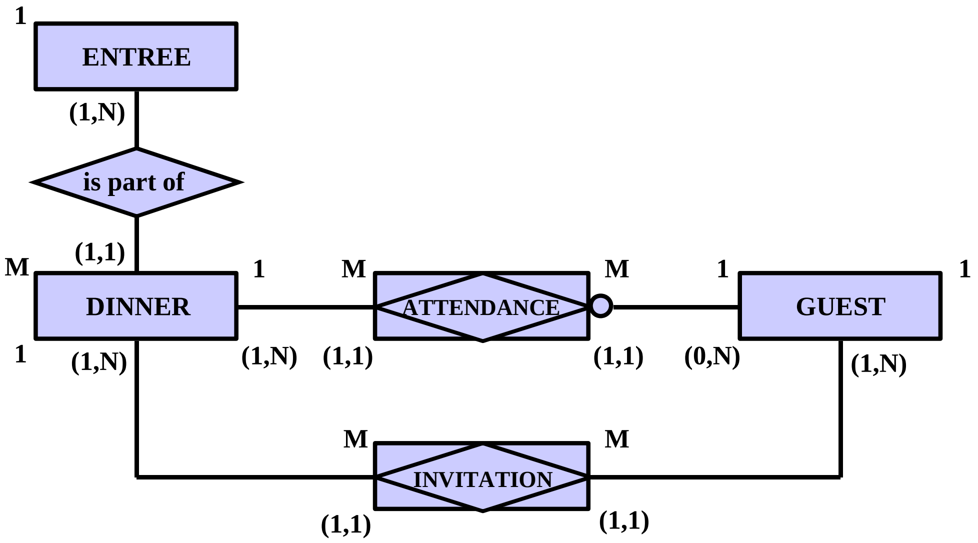
- An INVOICE is written by a SALESREP. Each sales representative can write many invoices, but each invoice is written by a single sales representative.
- The INVOICE is written for a single CUSTOMER. However, each customer can have many invoices.
- An INVOICE may include many detail lines (LINE) which describe the products bought by the customer.
- The product information is stored in a PRODUCT entity.
- The product's vendor information is found in a VENDOR entity.



3. Given the following summary of business rules for the ROBCOR catering service, draw the ERD. Make sure to include all appropriate entities, relationships, connectivities, and cardinalities.

> Note: Limit your ERD to entities and relationships based on the business rules shown here. In other words, do NOT add "realism" to your design by expanding or refining the business rules! However, make sure that you include the attributes that would permit the model to be successfully implemented.

Each dinner is based on a single entree, but each entree can be served at many dinners. A guest can attend many dinners, and each dinner can be attended by many guests. Each dinner invitation can be mailed to many guests, and each guest can receive many invitations.


1
ENTREE
(1,N)
is part of
(1,1)
M
DINNER
1
(1,N)
1
M
ATTENDANCE
M
1
GUEST
1
(1,N)
(1,1)
(1,1)
(0,N)
(1,N)
M
INVITATION
M
(1,1)
(1,1)