Chapter 1 The Database Environment and Development Process

Chapter Overview

The purpose of this chapter is to introduce students to the database approach to information systems development, the important concepts and principles of the database approach, and the database development process within the broader context of information systems development. This is an important chapter because it conveys a sense of the central importance of databases in today's information systems environment and in all modern enterprises. The idea of an organizational database is intuitively appealing to most students. However, many students will have little or no background or experience with the technical implementation of databases. Others will have had some experience with database management systems intended for personal or workgroup use (such as Microsoft Access). Consequently, they will have a limited perspective concerning an organizational approach to databases.

In this chapter we introduce the basic concepts and definitions of databases. We contrast data with information, and introduce the notion of metadata and its importance. We contrast the database approach with older file processing systems, and introduce the Pine Valley Furniture Company case to illustrate these concepts. We describe the range of database applications from databases with a personal or workgroup scope to enterprise databases and identify key decisions that must be made for each type of database. We describe both the potential benefits and typical costs of using the database approach. We also trace the historical evolution of database systems, in order to provide a context for understanding the database approach for data storage and retrieval.

The chapter also presents an expanded description of the systems development life cycle (including an introduction to rapid application development methods of prototyping and agile software development) and the role of database development within it. The chapter provides an updated description of the well-known three-schema architecture and uses it to summarize the various deliverables of database development. The chapter concludes with an example of database development situated in the context the Pine Valley Furniture Company case.

Chapter Objectives

Specific student learning objectives are included at the beginning of each chapter. *From an instructor's point of view, the objectives of this chapter are to:*

1. Create a sense of excitement concerning the data and database management field and the types of job opportunities that are available.
2. Acquaint students with the broad spectrum of database applications and how organizations are using database applications for competitive advantage.
3. Introduce the key terms and definitions that describe the database environment.
4. Describe data models and how they are used to capture the nature and relationships among data.
5. Describe the major components of the database environment and how these components interact with each other.

6. Provide a review of systems development methodologies, particularly the systems development life cycle, prototyping, and agile software development; build an understanding of how database development is aligned with these methodologies.
7. Develop an understanding of the different roles within in a database development team.
8. Make students aware of the three-schema architecture and its benefits for database development and design.
9. Introduce the Pine Valley Furniture Company case, which is used throughout the text to illustrate important concepts.
10. Introduce the Mountain View Community Hospital case, which is included at the end of each chapter as a source for student projects.

Key Terms

Agile software development	Database	Metadata
Conceptual schema	Database application	Physical schema
Constraint	Database management system (DBMS)	Prototyping
Data	Enterprise data modeling	Relational database
Data independence	Enterprise resource planning (ERP)	Repository
Data model	Entity	Systems development life cycle (SDLC)
Data modeling and design tools	Information	User view
Data warehouse	Logical schema	

Classroom Ideas

1. Start with a discussion of how students interact with systems built on databases on a daily basis (credit card transactions, shopping cards, telephone calls, cell phone contact lists, downloadable music, etc.). If you teach in a classroom with computers, ask students to find examples of Web sites that appear to be accessing databases.
2. Contrast the terms “data” and “information”. Using Figure 1-1 as a starting point, have the students provide some good examples of data and information from their own experiences. This may well lead to some differences of opinion, and the conclusion that one person’s data may be another person’s information.
3. Introduce the concept of metadata using Table 1-1. Ask the students to suggest other metadata that might be appropriate for this example.
4. Discuss file processing systems and their limitations, using Figure 1-2 and Table 1-2. Emphasize that many of these systems are still in use today.
5. Introduce data models using Figure 1-3. Discuss the differences between an enterprise data model and a project data model, using Figures 1-3 (a) and (b).
6. Discuss each of the advantages of the database approach (Table 1-3). Stress that these advantages can only be achieved through strong organizational planning and commitment. Also discuss the costs and risks of the database approach (Table 1-4).
7. Introduce the students to the major components of the database environment (Figure 1-5).

Stress the interfaces between these components and the fact that a proper selection of the components can "make or break" a database implementation.

8. Introduce the concept of a data warehouse as a type of enterprise database. This topic is described in detail in Chapter 9.
9. Review the evolution of database technologies and the significance of each era (Figure 1-10). Add your own perspective to the directions that this field is likely to take in the future.
10. Your students may have examples from their workplaces to contribute about client/server architectures. You may also provide them with an understanding of where the DBMS software and their data will be stored at your school as an illustration.
11. A quick in-class demo of Microsoft Access or similar product is useful to give the students an initial exposure to a DBMS and demonstrate a prototyping approach to database development. Consider using the PVFC prototyping request as an example.
12. If time permits, have the students answer several problems and exercises in class.
13. Use the project case to reinforce concepts discussed in class. Students can be assigned to work on this case in class if time permits, or it can be used as a homework assignment.
14. If time permits, use Teradata University Network resources to demonstrate the structure and contents of a relational database for some of the textbook datasets. Demonstrate, or lead students through, some simple SQL retrieval exercises against the textbook databases.

Answers to Review Questions

1-1. *Define each of the following key terms:*

a. *Data*. Stored representations of objects and events that have meaning and importance in the user's environment.
b. *Information*. Data that have been processed in such a way as to increase the knowledge of the person who uses it.
c. *Metadata*. Data that describes the properties or characteristics of end-user data and the context of that data.
d. *Enterprise resource planning* (ERP). A class of systems that integrate all functions of the enterprise, such as manufacturing, sales, finance, marketing, inventory, accounting, and human resources.
e. *Data warehouse*. An integrated decision support database whose content is derived from the various operational databases.
f. *Constraint.* A rule that cannot be violated by database users.
g. *Database*. An organized collection of logically related data.
h. *Entity* A person, place, object, event, or concept in the user environment about which the organization wishes to maintain data.
i. *Database management system.* A software system that is used to create, maintain, and provide controlled access to user databases.
j. *Client/server architecture.* A local area network-based environment in which database software on a server (called a database server or database engine) performs database commands sent to it from client workstations, and application programs on each client concentrate on user interface functions.
k. *Systems development life cycle (SDLC).* A traditional methodology used to develop,

maintain, and replace information systems.

l. *Prototyping.* An iterative process of systems development in which requirements are converted to a working system that is continually revised through close work between analysts and users.
m. *Enterprise data model.* The first step in database development, in which the scope and general contents of organizational databases are specified.
n. *Conceptual data model.* A detailed, technology-independent specification of the overall structure of organizational data.
o. *Logical data model.* The representation of data for a particular data management technology (such as the relational model). In the case of a relational data model, elements include tables, columns, rows, primary and foreign keys, as well as constraints.
p. *Physical data model.* A set of specifications that detail how data from a logical data model (or schema) are stored in a computer's secondary memory for a specific database management system. There is one physical data model (or schema) for each logical data model.

1-2. *Match the following terms and definitions:*

c data
b database application
l constraint
g repository
f metadata
m data warehouse
a information
j user view
k database management system
h data independence
e database
i enterprise resource planning (ERP)
r systems development life cycle (SDLC)
o prototyping
d enterprise data model
q conceptual schema
p internal schema
n external schema

1-3. *Contrast the following terms:*

a. *Data dependence; data independence*. With data dependence, data descriptions are included with the application programs that use the data, while with data independence the data descriptions are separated from the application programs.
b. *Structured data*; *unstructured data*. Structured data refers to facts related to objects and events of importance in the user's environment and represent the traditional data that is easily stored and retrieved in traditional databases and data warehouses. Unstructured data refers to multimedia data, such as images, sound and video

segments or to unstructured textual data. All these types of data are now stored as part of the user's business environment.

c. *Data; information*. Data consist of facts, text, and other multimedia objects, while information is data that have been processed in such a way that it can increase the knowledge of the person who uses it.
d. *Repository; database*. A repository provides centralized storage for all data definitions, data relationships, and other system components, while a database is an organized collection of logically related data.
e. *Entity; enterprise data model*. An entity is an object or concept that is important to the business, while an enterprise data model is a graphical model that shows the high-level entities for the organization and the relationship among those entities.
f. *Data warehouse; ERP system*. Both use enterprise level data. Data warehouses store historical data at a chosen level of granularity or detail, and are used for data analysis purposes, to discover relationships and correlations about customers, products, and so forth that may be used in strategic decision making. ERP systems enable organization's business processes and integrate operational data at the enterprise level, integrating all facets of the business, including marketing, production, sales, and so forth.
g. *Personal databases; multitier databases*. A personal database is intended for a single user to manage small amounts of data in an efficient manner, and it resides on a personal computing device (such as a laptop or a smart phone). Multitier databases share multiple (sometimes very large numbers of) users. They house the user interface on client devices and the business logic may be maintained on multiple server layers to accomplish the business transactions requested by client devices.
h. *Systems development life cycle; prototyping.* Both are systems development processes. The SDLC is a methodical, highly structured approach that includes many checks and balances. Consequently, the SDLC is often criticized for the length of time needed until a working system is produced, which occurs only at the end of the process. Increasingly, organizations use more rapid application development (RAD) processes, which follow an iterative process of rapidly repeating analysis, design, and implementation steps until you converge on the system the user wants. Prototyping is a widely used method within RAD. In prototyping, a database and its applications are iteratively refined through a close interaction of systems developers and users.
i. *Enterprise data model; conceptual data model.* In an enterprise data model, the range and contents of the organizational databases are set. Generally, the enterprise data model represents all of the entities and relationships. The conceptual data model extends the enterprise data model further by combining all of the various user views and then representing the organizational databases using ER diagrams.
j. *Prototyping; Agile software development.* Prototyping is a rapid application development (RAD) method where a database and its application(s) are iteratively refined through analysis, design, and implementation cycles with systems developers and end users. Agile software development is a method that shares an emphasis on iterative development with the prototyping method yet further emphasizes the people and rapidity of response in its process.

1-4. *Five disadvantages of file processing systems:*

a. Program-data dependence
b. Duplication of data
c. Limited data sharing
d. Lengthy development times
e. Excessive program maintenance

1-5. *Nine major components in a typical database system environment:*

a. CASE tools: automated tools used to design databases and database applications.
b. Repository: centralized storehouse of data definitions.
c. Database management system (DBMS): commercial software used to define, create, maintain, and provide controlled access to the database and the repository.
d. Database: organized collection of logically related data.
e. Application programs: computer programs that are used to create and maintain the database.
f. User interface: languages, menus, and other facilities by which users interact with the various system components.
g. Data administrators: persons who are responsible for the overall information resources of an organization.
h. System developers: persons such as systems analysts and programmers who design new application programs.
i. End users: persons who add, delete, and modify data in the database and who request information from it.

1-6. *Relationships between tables:*

Relationships between tables are expressed by <u>identical</u> data values stored in the <u>associated</u> columns of related tables in a relational database.

1-7. *Definition of data independence:*

Data independence refers to the separation of data descriptions from the application programs that use the data. It is an important goal because it allows an organization's data to change and evolve without changing the application programs that use the data. Additionally, data independence allows changes to application programs without requiring changes in data storage structure.

1-8. *10 Potential benefits:*

Potential benefits of the database approach are:

a. Program-data independence
b. Minimal data redundancy
c. Improved data consistency
d. Improved data sharing
e. Increased development productivity

f. Enforcement of standards
g. Improved data quality
h. Improved data accessibility and responsiveness
i. Reduced program maintenance, and
j. Improved decision support.

1-9. *Five costs or risks of the database approach are:*

a. New, specialized personnel
b. Installation, management cost, and complexity
c. Conversion costs
d. Need for explicit backup and recovery, and
e. Organizational conflict.

1-10. *Nine key components of a typical database environment*

a. CASE tools: automated tools used to design databases and database applications.
b. Repository: centralized storehouse of data definitions.
c. Database management system (DBMS): commercial software used to define, create, maintain, and provide controlled access to the database and the repository.
d. Database: organized collection of logically related data.
e. Application programs: computer programs that are used to create and maintain the database.
f. User interface: languages, menus, and other facilities by which users interact with the various system components.
g. Data administrators: persons who are responsible for the overall information resources of an organization.
h. System developers: persons such as systems analysts and programmers who design new application programs.
i. End users: persons who add, delete, and modify data in the database and who request information from it.

1-11. *Possibility of no database on a tier of a multi-tiered database?*

Yes, it is possible. The client tier — a PC or a mobile client — typically has presentation logic but no database installed on it.

1-12. *Five SDLC phases:*

a. Planning

Purpose**:** To develop a preliminary understanding of the business situation and how information systems might help solve a problem or make an opportunity possible ***Deliverable*****:** A written request to study the possible changes to an existing system; the development of a new system that addresses an information systems solution to the business problems or opportunities

b. Analysis

Purpose**:** To analyze the business situation thoroughly to determine requirements, to structure those requirements, and to select between competing system features
Deliverables**:** The functional specifications for a system that meets user requirements and is feasible to develop and implement

c. Design

Purpose**:** To elicit and structure all information requirements; to develop all technology and organizational specifications
Deliverables**:** Detailed functional specifications of all data, forms, reports, displays, and processing rules; program and database structures, technology purchases, physical site plans, and organizational redesigns

d. Implementation

Purpose**:** To write programs, build data files, test and install the new system, train users, and finalize documentation
Deliverables**:** Programs that work accurately and according to specifications, documentation, and training materials

e. Maintenance

Purpose**:** To monitor the operation and usefulness of a system; to repair and enhance the system
Deliverables**:** Periodic audits of the system to demonstrate whether the system is accurate and still meets needs

1-13. *Activities and five phases of SDLC?*

Database development activities occur in every phase of the SDLC. Actual database development is most intense in the design, implementation, and maintenance steps of the SDLC.

1-14. *Commonalities of SDLC, prototyping, and agile development methodologies:*

Procedures and processes that are common to SDLC, prototyping, and agile methodologies include:

- Understanding and analyzing the customer's business requirements for the system
- Translating the customer's requirements into specifications (logical & physical) for systems development
- Developing databases and software programs to meet specifications; and
- Implementing an operational system.

The methodologies are considered to be different not because of <u>what</u> is done, but because the <u>timing</u> of the methodologies differ. The SDLC methodology is methodical and thorough which makes it well-suited for systems that populate and revise databases. Prototyping, with its rapidly repeating analysis, design, and implementation phases, is well-suited for systems that retrieve data <u>and</u> for helping to refine a customer's requirements for a new system. Agile software development emphasizes quick responses and rests on high-involvement from knowledgeable customers. Agile software development is well-suited to projects with unpredictable and/or rapidly changing requirements and responsible developers (per text citation of Fowler, 2005).

1-15. *Differences between conceptual schema, user view, and internal schema:*

A conceptual schema defines the whole database without reference to how data are stored in a computer's secondary memory. A user view (or external schema) is also independent of database technology, but typically contains a subset of the associated conceptual schema, relevant to a particular user or group of users (e.g., an inventory manager or accounts receivable department). An internal schema consists of both a physical schema and a logical schema. A logical schema consists of a representation of the data for a type of data management technology. For example, if the relational model is the technology used, then the logical schema will consist of tables, columns, rows, primary keys, foreign keys and constraints. A physical schema contains the specifications for how data from a logical schema are stored in a computer's secondary memory.

1-16. *Three-schema architecture:*

a. external schema
b. conceptual schema
c. internal schema

1-17. *Phases and activities of SDLC within textbook scenario:*

Student answers may vary depending upon whether or not they read the section closely enough to realize that Chris is following a prototyping methodology approach to developing the database application for PVFC. The prototyping methodology is shown in Figure 1-8, while the traditional development approach is shown in Figure 1-7.

According to Figure 1-8, Chris' project activities would map to the following phases of the *prototyping* database development process:

Chris' activities	**Prototype phase (and comments)**
Project Planning	Identify Problem (Conceptual Data Modeling) - To some extent, a separate "planning" phase does not really exist under the prototyping approach as it happens continuously as the prototype evolves. On the other hand, the Identify Problem phase involves sketching a preliminary data model, which is work that Chris clearly completes.
Analyzing Database Requirements	Identify Problem (Conceptual Data Modeling) Develop initial prototype (Logical Database Design) - In this stage of Chris' work with Helen, he is still gathering iterations of the kinds of data that Helen needs to do her job. In some ways, Chris is refining the

	Conceptual Data Model and in other ways Chris is developing the more detailed Logical Database Design.
Designing the Database	Develop initial prototype (Logical Database Design, Physical database design and definition, database implementation) - Chris takes the knowledge he has gained from the initial sessions with Helen and begins to build a functioning example of the database in an agreed-upon relational database management system.
Using the Database	Implement and use prototype; Revise and enhance prototype (Database maintenance) - Chris provided enough of a working sample database that Helen could use it and make suggestions about how to revise it. Chris could iteratively make changes to improve the solution, and move some initial ad-hoc queries into more formal reports.
Administering the database	Convert to operational system (Database maintenance) - Chris and Helen agreed that the prototype was functioning efficiently enough to allow it to become the everyday, operational, “production” system for Helen to use. As requested by Helen, and when time allows, Chris is able to make changes to the operational database to better meet Helen’s needs and requests.

1-18. *Why does PVFC need a data warehouse?*

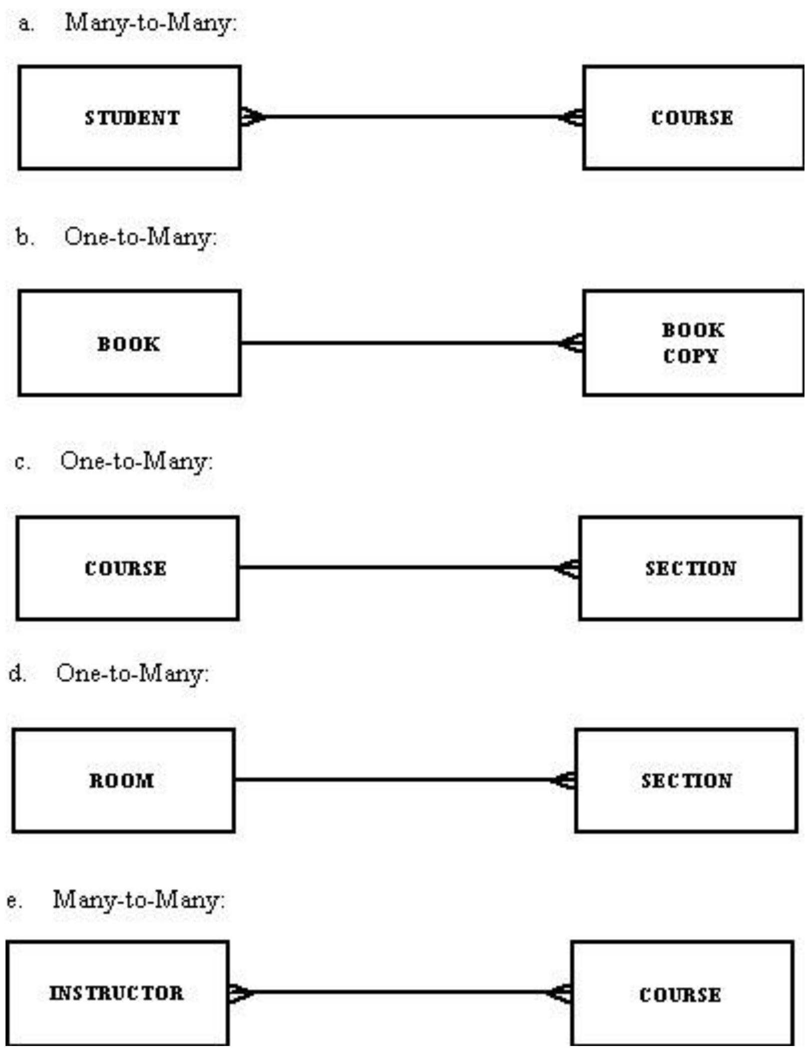
Pine Valley Furniture Company (PVFC) uses a database management system to support its operational functions. This database is not, however, structured in a way that supports timely analysis of trends or historical patterns. PVFC can benefit from a data warehouse that is appropriately structured for questions related to vendor pricing and/or customer order patterns over time. A data warehouse would enable PVFC to summarize data drawn from various operational databases (i.e., department, independent applications, and ERP) into meaningful structures for timely decision-making access.

1-19. *Three areas where very large databases are used:*

Very large databases are being used to improve customer relationship management (CRM) by creating CRM systems that react to individual customer’s purchase behavior. For example, such as database could be used to suggest other items that a customer may want to purchase based on that customer’s previous purchases. They are also being used to build a foundation for virtual products and services and to understand online users’ behavior at a detailed level. Online shopping sites are able to carry a large virtual inventory stored in a database for the customer to peruse.

Solutions to Problems and Exercises

1-20. *Examples of relationships:*



1-21. *Advanced data types have several special requirements:*

a. Storage requirements – multimedia objects (such as images, sound, and video clips) require substantial storage capacity, which needs to be justified.
b. Content management – this is the problem of storing, locating, and retrieving the multimedia objects. This process requires specialized software not generally available in a relational DBMS or extra effort to create a means to rapidly access multimedia objects (such as keyword indexes).
c. Maintenance – while conventional relational data are easily updated, multimedia objects may require maintaining multiple versions of the data. Usually the whole object needs to be restored because it is treated as a whole rather than a set of parts.

1-22. *Metadata for Class Roster:*

Please note that some columns have been omitted in order to save space. Columns "Created", "Updated", and "Responsible Party" were added to the metadata.

Name	Type	Description	Source	Created	Updated	Responsible Party
Course	Alphanumeric	Course ID and name	Academic Unit	5/10/2015	6/1/2015	Registrar
Section	Integer	Section number	Registrar	5/10/2015		Registrar
Semester	Alphanumeric	Semester and year	Registrar	5/10/2015		Registrar
Name	Alphanumeric	Student name	Student IS	8/07/2015		Student IS
ID	Integer	Student ID (SSN)	Student IS	8/07/2014		Student IS
Major	Alphanumeric	Student major	Student IS	8/07/2014	11/15/2014	Student IS
GPA	Decimal	Student grade point average	Academic Unit	8/07/2014	5/10/2015	Department Chair

1-23. *Why do organizations create multiple databases?*

There are several reasons. First, because of resource limitations, organizations fund development of their information systems one application at a time. Second, organizations may acquire some of their information systems from outside vendors. This also results in a proliferation of databases. Third, mergers and acquisitions generally result in multiple databases.

What organizational and personal factors lead an organization to have multiple, independently managed databases?

Perhaps the most common reason is that end users and user groups develop their own database applications, rather than wait for the central IS organization to develop a centralized database. Also the pressures associated with rapid business change result in organizations taking a short-term, suboptimal approach rather than a careful, long-term strategy.

1-24. *Data entities and Enterprise Data Model for student organization or group:*

This is a good in-class, interactive exercise for individuals or small groups. For individuals, have each student choose a student club, fraternity/sorority, or other organization to illustrate a "top-down" approach to develop an enterprise data model. For small groups, divide the class into groups and have each group work to develop an enterprise data model for a club, fraternity/sorority, or other organization. Reconvene as a large class to compare/contrast each of the small group enterprise data models. Identify the similarities and differences through class discussion.

1-25. *Data from driver's license bureau:*

a. Driver's name, address, and birthdate: structured data
b. The fact that the driver's name is a 30-character field: metadata; fact describing

property

c. A photo image of the driver: unstructured data
d. An image of the driver's fingerprint: unstructured data
e. The make and serial number of the scanning device that was used to scan the fingerprint: structured data
f. The resolution (in megapixels) of the camera that was used to photograph the driver: metadata; fact describing context
g. The fact that the driver's birth date must precede today's date by at least 16 years: metadata; fact describing context

1-26. *Great Lakes Insurance database suggestion:*

One suggested approach would be to create an enterprise database to contain all information about customers, policies, etc. The need for an enterprise database is clear, since policy information would need to be accessed not just by the sales team but also by the actuarial department and the claims department. For inside agents, access to the database would be through an intranet, utilizing a browser-based application as the front-end. Each outside agent would have a personal database on his or her notebook computer with only information for his or her territory. The personal database would then be synchronized periodically with the enterprise database through the use of an extranet.

1-27. *Record Store data model questions:*

a. one-to-many
b. one-to-many
c. If one is added, it is essential that the designer carefully define what the meaning of the new relationship is. It could, for example, mean that the store intends to carry all albums by a specific artist.

1-28. *Questions about Figure 1-11 database:*

Some common data elements that may be redundant are: Vendor ID, Vendor Name, Vendor Address, Customer ID, Customer Last Name, Customer First Name, Customer Middle Initial, Purchase Order Number, Purchase Order Date.

This duplication may lead to the problem that payments may not be properly matched to vendor orders, or that customer receipts are improperly matched to customer bills. These potential mismatches could cause issues in collection and payment of financial transactions for the organization, and may cause issues with relationships with customers and vendors.

At a first glance, these duplications appear to violate the principles of the database approach outlined in this chapter. However, the organization may have procedural or system checks-and-balances that periodically audit or synchronize the apparent data duplication throughout the organization in this three-tier scenario. These checks-and-balances are not apparent on this Figure, but if they exist, they might compensate for the apparent violation of database approach principles.

1-29. *Representation of SDLC:*

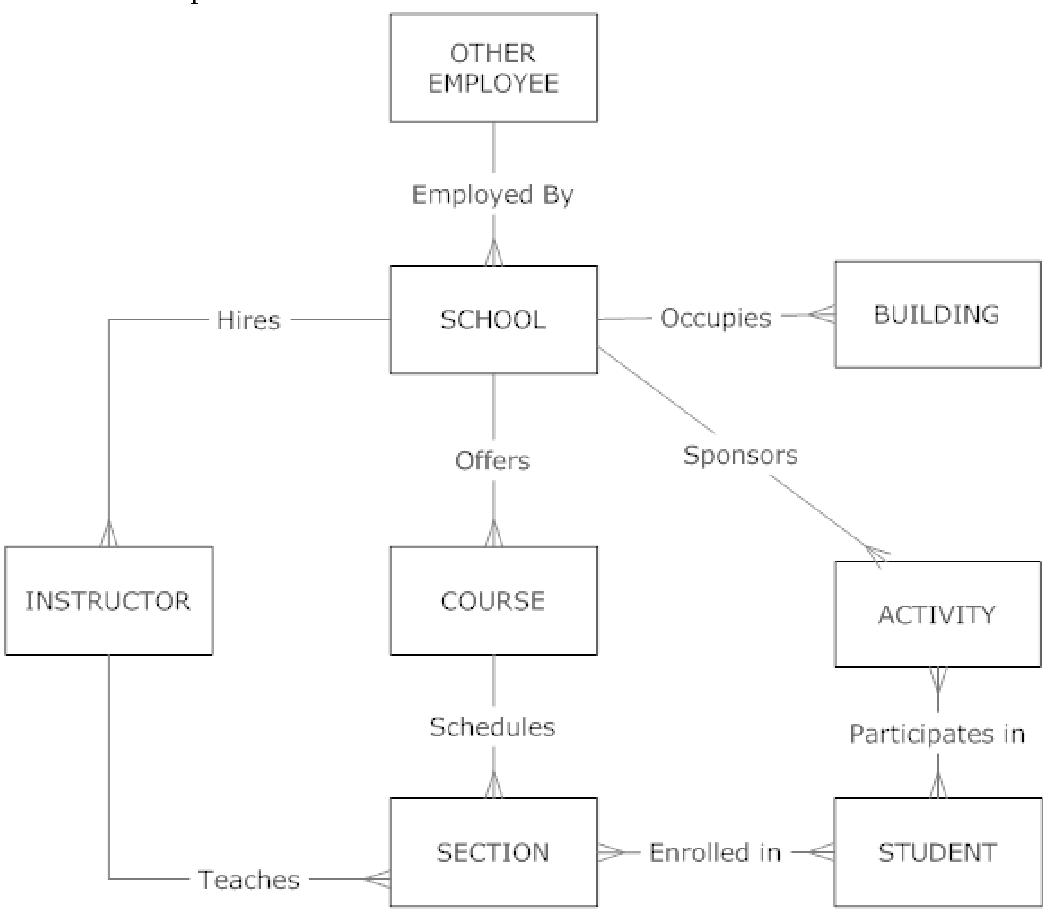
The representation of the systems development life cycle has changed from the original waterfall metaphor. While it is a more compact representation, there are still some problems. For example, it is not purely linear. Also, it is possible to conduct steps in parallel due to time overlaps. One additional problem of a pure SDLC model is the inability to go back from one step to another without completing the entire five-step process; most practical SDLC implementations allow movement between stages in both directions.

1-30. *Three additional entities for PVFC:*

EMPLOYEE, SUPPLIER, and SHIPMENT might be good examples since all of them represent major categories of data about the entities managed by the organization

1-31. *Consider Business Enterprise example:*

a. Enterprise Data Model



b. *Benefits from a multiple-tiered architecture for data?*

Considerations for using a multitiered architecture: Since much of the data may be updated from a large number of different functions, network traffic will be an issue of crucial importance. Processing close to the source data could reduce network traffic. Client technologies, however, can be mixed (desktops, laptops, smart phones, tablets, network computers, information kiosks, etc.) and yet, share common data. In addition, you can change technologies at any tier with limited impact on the system modules on other tiers. All this will allow for data consistency and maintaining academic standards — a critical success factor for the academic unit.

1-32. *Contrasting database development during conceptual data modeling and logical database design phases:*

It is often said that conceptual data modeling is done in a top-down fashion, driven from a general understanding of the business area, not from specific information processing activities. *Logical database design* approaches database development from two perspectives. First, the conceptual data model is transformed into a standard notation through normalization, based on relational database theory. Then, as each computer program in the information system is designed — including the program's input and output formats — a detailed review of the transactions, reports, displays, and inquiries supported by the database is performed. This bottom-up analysis verifies exactly what data are to be maintained in the database and the nature of those data as needed for each transaction, report, and so forth. During logical database design, you combine or integrate the original conceptual data model (more general information) along with the individual user views (more specific information) into a comprehensive design.

1-33. *Location of prototype database:*

Department databases are often developed in combination by end users, systems professionals working in business units, and central database professionals. A combination of people is necessary since a wide variety of issues must be balanced in the design of shared databases: processing speed, ease of use, differences in data definitions, and the like. Often, the prototype being developed will be located on a development area server, where those involved with the project will have easy access to the database and no likelihood of interfering with the organization's production databases. This answer assumes that the organization maintains separate development and production servers for the databases.

1-34. *Enterprise data models and conceptual data models:*

Enterprise data modeling results in a total picture or explanation of organizational data, not the design for a particular database. A particular database provides the data for one or more information systems, whereas an enterprise data model, which may encompass many databases, describes the scope of data maintained by the organization. Therefore, the level of abstraction associated with an enterprise data model is higher; conceptual data models are be more detailed.

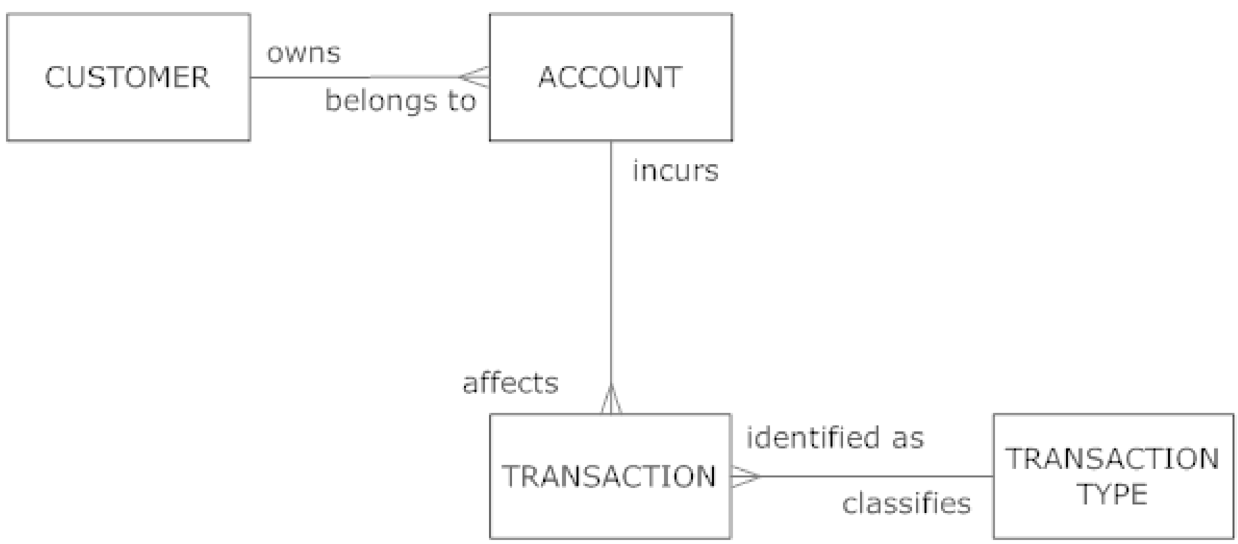
1-35. *Iteration between physical and logical database design:*

Database development activities occur in each of the SDLC phases, and it is not only possible but quite frequent that feedback received during a later stage causes a project to return to a prior phase. SDLC activities may find missing elements or errors when designing specific transactions, reports, displays, and inquiries. When a missing element is noticed, for example, it is necessary to revisit the logical database design.

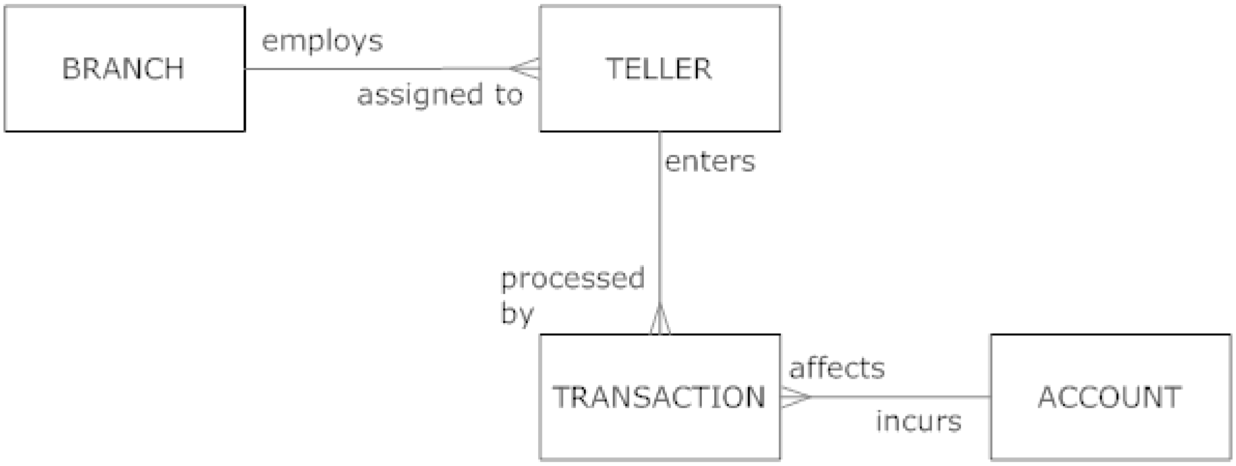
1-36. *User views of organizational data:*

A good approach in developing this problem for a bank might be to carefully select the views to be developed by collecting a transaction slip, monthly statement (representing each type of account), statement of earnings, etc. Examples of data included in each message are customer information, bank information, and transaction data (checks, deposits, service charges, maintenance fees, overdraft protection fees, and so forth). Statement and deposit slip views are given below, as is the combined conceptual data model. Combining the different views could lead to the addition of new attributes or possibly entities and relationships not being shown in the original views.

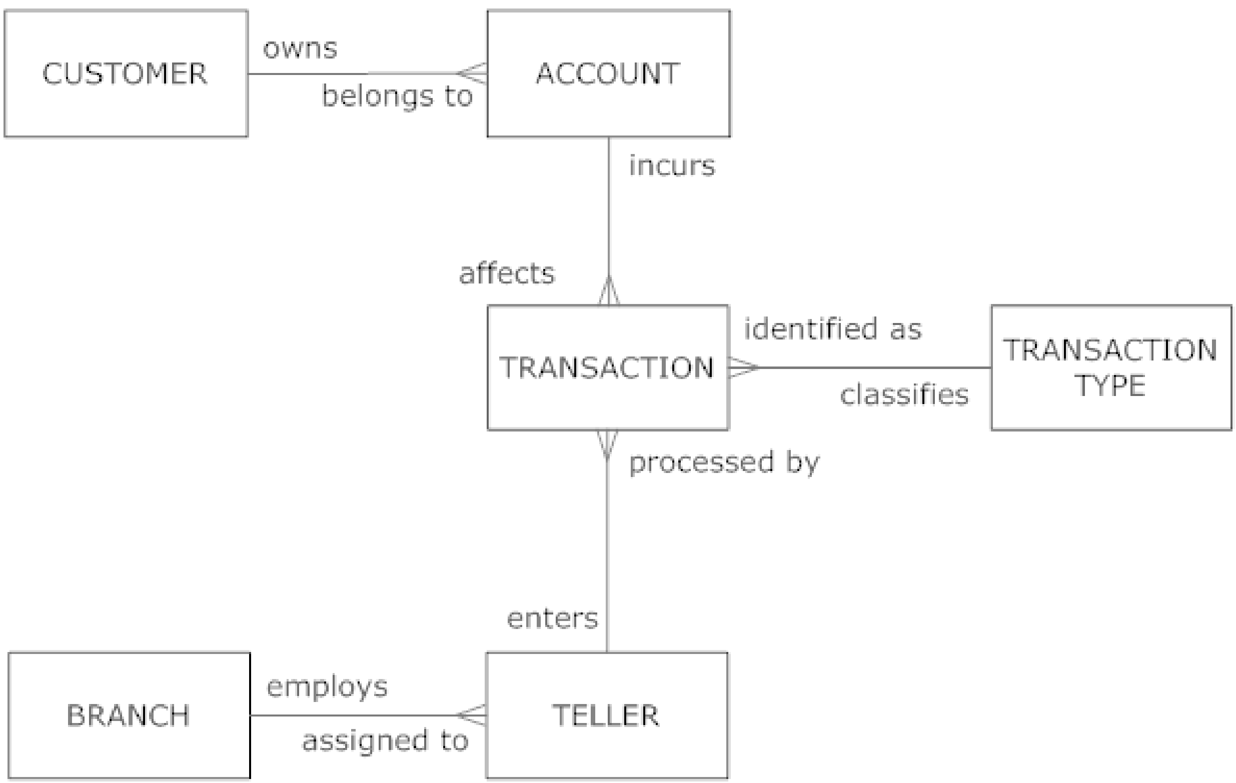
Statement view:



Deposit slip view:



Conceptual Data Model:



1-37. *Explain Figure 1-14:*

The single line connection indicates a one-to-one relationship between the entities. In business terms: each order is billed on exactly one invoice, each invoice is written for only one order. In contrast, many payments can be made on one order. Remember to model for the exceptions.

1-38. *Questions about Project Data Model in Figure 1-15:*

a. A Customer places many orders, yet an Order is placed by one Customer. Each Order is billed on a single Invoice, and an Invoice is related to a single Order. Each Payment is related to a single Invoice, but an Invoice may be paid with one or several Payments. Each Order contains multiple Order Lines but a specific Order Line refers to only one Order and one Product at a time. Each Product may be seen on multiple Order Lines. Each Product belongs to only one Product Line and a Product Line includes multiple Products. For each Customer, PVFC tracks the ID, Name, Type, Zip Code and number of years it has been a PVFC customer. For Orders, we track Order number along with each order's placement date, fulfillment date, and number of shipments. For Invoices, we track Invoice number, Order number, and Invoice Date. Each Payment tracks the Invoice Number, Payment Date, and Payment Amount. Each Order Line tracks the specific quantity of product for the specific order. For Products, we track its ID, description, finish, standard price, product cost, prior year's sale goal, and current year's sale goal. Each Product Line tracks its name, prior year's sales goal and current year's sale goal.

b. The aspect of the diagram that is difficult to describe (and may also be difficult to implement and maintain) is the sales goal attributes for the Product Lines and Products. Since it is expressed as Prior Year and Current Year values, this data will need to be maintained with additional programming in the database application, as there is currently no logical design to deal with this time-based data in the proposed design. In my opinion, the reason for this ambiguity is that the prototyping approach did not yet experience the maintenance issues of date-related data in the database, due to the shortened time working with the data and sample reports. As the end user and the analyst work through the database application over time, the need for an adjusted design will become painfully clear.

1-39. *SQL statements in Figure 1-17 and 1-18 questions:*

a. The field size for the ProductLineName field in the Product table may be 1-40 characters. This size is due to the SQL Create Table definition of ProductLineID field being set to 40 characters, with a VARCHAR datatype.

b. The ProductID field is specified as NOT NULL in the SQL statement. ProductID is a required attribute because it is designed as the Primary Key of the table, and in relational design Primary Key fields cannot be empty or missing.

c. The FOREIGN KEY clause in the SQL statement tells us that the ProductLineID field in the Product_T table refers back to the ProductLineID primary key that is stored in the ProductLine_T table. This FOREIGN KEY clause is the way that referential integrity in a relational database is enforced in the actual physical database.

1-40. *SQL Query in Figure 1-19:*

a. SalesToDate is calculated by multiplying the quantity by the price.

b. Eliminating "Home Office" from the Criteria line under ProductLineName would result in all product line names being displayed, not just the Home Office line.

1-41. *Figure 1-15 and new Query specification questions:*

a. The following entities will be needed: Payment, Invoice, Order, Order Line, Product, Customer
b. All of the above entities will be needed for the SQL query.

1-42. *Summary of important properties of data models:*

	All Entities?	All Attributes?	Technology Independent?	DBMS Independent?	Record Layouts?
Enterprise	Y	N	Y	Y	N
Conceptual	Y	Y	Y	Y	N
Logical	Y	Y	N	Y	Y
Physical	Y	Y	N	N	Y

Suggestions for Field Exercises

1-43.

You can accomplish this exercise either by arranging a field trip for your class (preferred), or by inviting the IS manager (or other key IS person) to visit your class. Following are some of the key steps to perform:

a. Select an organization. We suggest a mid-sized manufacturing company or a familiar organization in the service sector such as a hospital or bank.
b. Identify several mainstream applications such as human resource management, material requirements planning, and financial accounting.
c. Determine whether these applications were predominantly developed internally, or purchased from an outside vendor or vendors.
d. Now determine the mix between file processing and database processing for these applications. If a database approach is used, is the data shared among the applications?
e. To draw a figure depicting the files and databases, inquire whether the organization has system flowcharts (or similar documentation) that portray much of this information.

1-44.

This exercise is most easily performed as a continuation of Exercise 1. Arrange to interview a database administrator or key designer as part of the same field trip. Discuss whether the organization maintains user-oriented metadata, or only technical metadata. Where is this metadata maintained: within individual applications, in one or more CASE tool repositories, or elsewhere?

1-45.

If the student has selected an established company with a fairly extensive information systems department, they will find that the company is either already very involved in Web-enabling some or many parts of their business, or that they are actively planning to expand the scope of their involvement. You may want to encourage your students to use this question to explore where the company sees itself going with regard to the Internet. Many older companies are struggling very hard to adjust and remain competitive as the Internet dramatically affects business models.

1-46.

Again, as in the previous field exercise, this exercise will help students to integrate their new textbook knowledge with the realities that organizations must face. Some students will find organizations that are struggling to achieve client/server architecture and also deal with their legacy systems, for example.

1-47.

Students will probably find prototyping being used for smaller database applications. Prototyping is often the systems development methodology of choice where visual programming tools such as Visual Basic, Java, Visual C++, and fourth-generation languages are used because of the ease with which the user-system interface can be modified. Large systems that require a new database are more likely to use the SDLC, requiring a team of information systems professionals, including database designers and programmers. The systems development process for Web applications can be similar, with prototyping being employed. However, students may find that there is a lot more testing involved with Web applications, since one has less control over the client interface.

1-48.

Students are likely to find some of the following roles on the information systems development teams in larger organizations:

a. Systems analysts, who analyze the business situation and identify the need for information and information services to meet the problems or opportunities of the business
b. Database analysts, who concentrate on determining the requirements and design for the database component of the information system
c. Users, who provide assessment of their information needs and monitor appropriateness of the developed system
d. Programmers, who design and write computer programs that have embedded in them commands to maintain and access data in the database
e. Database and data administrators, who have responsibility for existing and future databases and ensure consistency and integrity across databases; as experts on database technology, they provide consulting and training to other project team members

f. Other technical experts, with facility in networking, operating systems, testing, and documentation
g. A senior systems or database analyst, who will be assigned to be project leader, who is responsible for creating detailed project plans as well as staffing and supervising the project team; this good project leader will possess leadership, management, customer relations and communications, technical problem solving, conflict management, team building, and risk and change management skills.

In smaller organizations, students may find that these roles are combined and performed by fewer people because the smaller organizations do not have the additional staff to assign to a diverse team. Thus, fewer people will cover these same functional responsibilities by each person having multiple roles to fulfill. Many employees in small organizations develop deep skill sets in programming, data administration, or technical database administration expertise.

1-49.

Note: See chapter section on "Managing the people involved in database development" for more background on this exercise.

If a CASE tool is being used, a repository stores descriptions of the data entities and the business rules, detailed descriptions of CSFs, and objectives. The information systems department analyzes all of them. Throughout the systems development process, CASE tools are used to develop data models and to maintain (in the repository) the metadata for the database and applications. A repository maintains all of the documentation too. Various people might use the CASE tools and associated repository during a database development project: systems analysts, database analysts, users, programmers, database and data administrators, and other technical specialists. As a significant new portion of a project is completed and entries are made in the repository, a review point occurs so that those working on the project and funding the human and capital resources of the project can assess progress and renew commitment based on incremental achievements.

1-50.

Student answers will vary depending upon the organization and the functional area chosen for the interview. As long as students are encouraged to contact individuals in larger or corporate environments, answers should reflect a variety of approaches to retrieving data, kinds of systems accessed for data retrieval, frequencies of data retrieval, levels of satisfaction with data access and retrieval, and challenges experienced. This would be a good exercise to assign for out-of-class investigation, and then use in-class time for students to meet in small groups to share their results with one another. A subsequent large-group discussion led by the instructor can readily illustrate the different experiences and raise student awareness of the diversity in organizational approaches to data storage, access, and retrieval.

1-51.

Student answers will vary.

Project Questions

1-52. *Initial Analysis Memo*

a. Approach to addressing the problem

There is obviously no one right solution to this question, and it does not make sense to provide a solution that would suggest that there is. The students should, however, discuss the following types of issues:

- FAME does not seem to have any computing-based solutions beyond personal productivity software, and they do not have any specialized IT personnel (and have no appetite to hire anybody). Any solution they will adopt needs to take this into account: heavy infrastructure or support requirements would, in all likelihood, make a specific approach impossible immediately.
- It appears that the solution could quite well be designed following an agile, iterative process. There are a number of elements of the system that can be implemented independently and as long as the data management architecture is (at least tentatively) designed in advance, the independent components can be integrated over time. Thus, suggesting a development model that is based on a set of time-boxed iterations is reasonable.
- The students should address the question of ownership of the required infrastructure resources. Would it make any sense for FAME to own its own hardware and system software infrastructure? It seems that an Infrastructure as a Service (IaaS) or Platform as a service (PaaS) solution would work well here.

b. Accomplishments, organizational goals, and functionality

- The new system would
    - Reduce errors in contracts, scheduling, and billing
    - Make information regarding artist availability centralized and more readily available for everybody, thus allowing faster decisions
    - Make it possible to speed up the business processes, at least theoretially allowing FAME to get the artists paid faster
    - Make it easier for the artist managers to review recommendations and other information regarding prospective artists.
- The system would perform the following functions (at later stages of development):
    - Artist scheduling
    - Performance billing
    - Accounts receivable
    - Accounts payable
    - Contract creation, management, and renewals
    - Prospect review and management

The system would help FAME serve its event organizer clients better and thus, provide its artists better and more frequent performance opportunities, thus also increasing FAME's commissions.

c. Key benefits

- Reduced errors in scheduling
- Reduced errors and faster speed in billing
- Reduced errors in accounts receivable management
- Faster payments to the artist
- Better management of artists' and artist managers' expenses
- Better negotiation position for FAME in conversations with event organizers
- Faster access to high-quality prospects
- Reduced risk of lapsed contracts

d. Timeline / roadmap

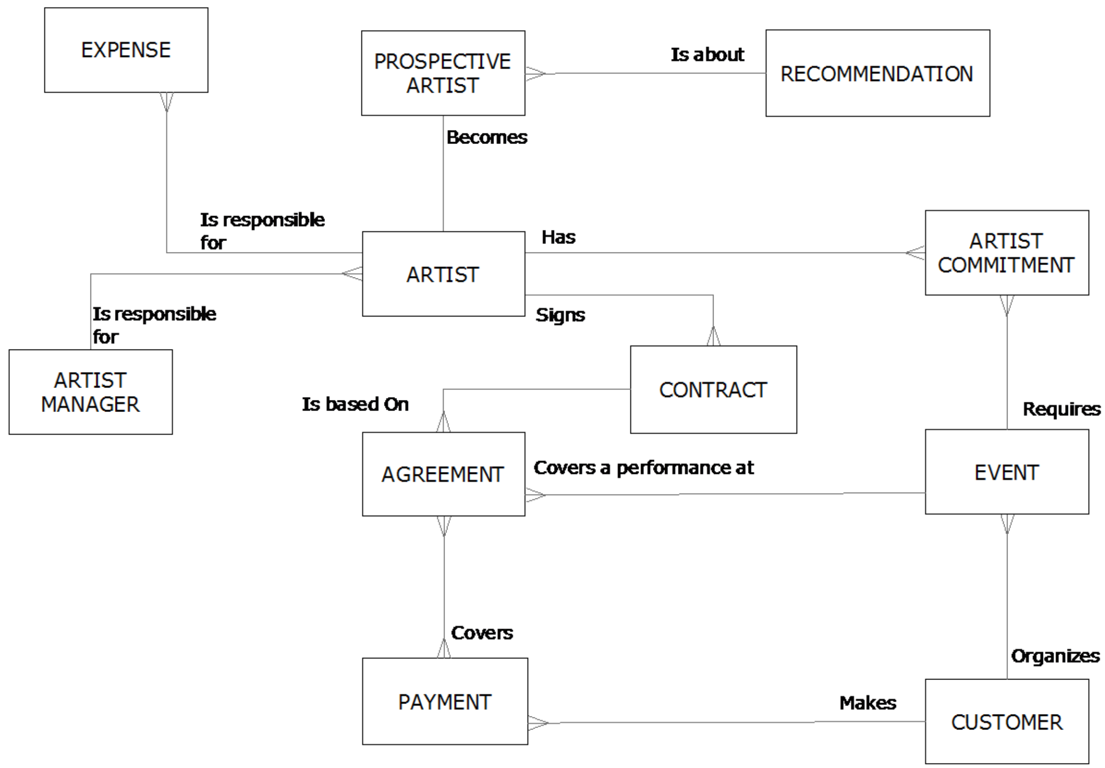
- The answer to this question depends on how the project is framed to the students. Our recommendation is that you frame the project in the context of your semester, using real time constraints.

e. Possible questions for Mr. Forondo

- How would he define the priorities between the various functionalities and potential benefits of the system?
- Determine whether or not Mr. Forondo has a reasonable idea of the cost of developing such a system from scratch and if he is willing to pay.
- Who are the key stakeholders your team should approach during the requirements specification phases?
- What are FAME's growth targets and what impact they will have on the system?
- Are there any particular risks Mr. Forondo is specifically worried about?

1-53. *Enterprise Data Model*

Please see the following page for the response to this question.



Question 1-53 Figure 1