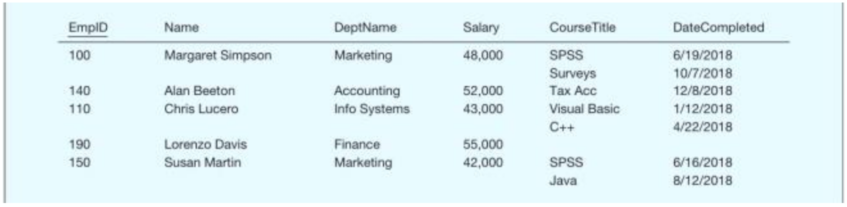
THE RELATIONAL DATA MODEL

Basic Definitions

- Data structure: data are organized in the form tables, with rows and columns
- Data manipulation: powerful operations (using something like sql) are used to manipulate data stored in the relations
- Data integrity: The model includes mechanisms to specify business rules that maintain the integrity of the data when they are manipulated.
- Relational data structure
    - Relation: is a named two dimensional table of data. Each relation consists of named columns and any number of rows.
        - A named column is known as an attribute
        - Each row is a record that contains data for a single entity
        - It is important to understand that a relation is the actual table framework and you can create instances of that table (populate it with data)
        - One way to express the structure of some relation would be like this:
          EMPLOYEE1(EmpID, Name, DeptName, Salary)
            - □ Name of relation and in parenthesis the attributes
- Relational Keys
    - Primary Key: is an attribute or a combo of attributes that uniquely identifies each row in a relation.
        - One way to denote this is like, EMPLOYEE1(EmpID, Name, DeptName, Salary)
        - Often called identifier
    - Composite key: is a primary key with more than one attribute
        - Example: the primary key for a relation DEPENDENT would likely consist of the combination EmpID and DependentName.
    - Foreign key: is an attribute in a relation that serves as the primary key of another relation
        - This is used to represent the relationship b/w two tables.
        - Example:
          EMPLOYEE1(EmpID, Name, DeptName, Salary) //foreign key is DeptName
          DEPARTMENT(DeptName, Location, Fax) //DeptName is a primary key of Department
            - □ In this example, DeptName in EMPLOYEE1 is a foreign key whose purpose serves as a way for to access some DEPARTMENT row/record whose ID's match to retrieve its data.
            - □ EMPLOYEE1 has a record in DEPARTMENT and it accesses it by using its foreign key which links up the primary key of some DEPARTMENT record

PROPERTIES OF RELATIONS:

- These are the properties of relational tables that distinguish them from nonrelational tables:
    a. Each relation (or table) in a database has a unique name
    b. Every value put into an attribute's column must be single valued (atomic); no multivalued attributes are allowed.
        i. For example,



EmpID	Name	DeptName	Salary	CourseTitle	DateCompleted
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/2018
				Surveys	10/7/2018
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/2018
110	Chris Lucero	Info Systems	43,000	Visual Basic	1/12/2018
				C++	4/22/2018
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/16/2018
				Java	8/12/2018

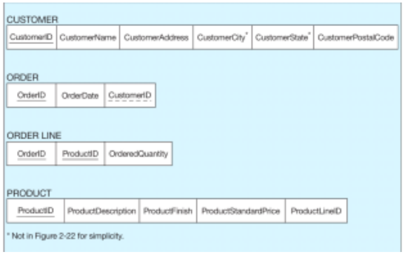
We have mulitvalued attribute called CourseTitle and DateCompleted which would make our table nonrelational

    c. Each row is unique, no two identical rows allowed
    d. Each attribute has a unique name
    e. The sequence of left to right in both columns and rows is insignificant

Sample Database

A relational database may consist of any number of relations (tables that are relational) and to describe them we would use either

- Short text statements like EMPLOYEE1(EmpID, Name, DeptName, Salary)
- Graphical representations like:



CUSTOMER

CustomerID	CustomerName	CustomerAddress	CustomerCity*	CustomerState*	CustomerPostalCode

ORDER

OrderID	OrderDate	CustomerID

ORDER LINE

OrderID	ProductID	OrderedQuantity

PRODUCT

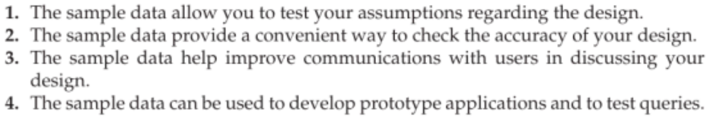
ProductID	ProductDescription	ProductFinish	ProductStandardPrice	ProductLineID

* Not in Figure 2-22 for simplicity.

Notes:

- Notice that order line has a composite primary key and it also has foreign keys which also happen to be the composite keys. In this case, we only show the composite keys by underlining them solid

- It is often wise to create an instance of your relational schema with sample data for four reasons:



1. The sample data allow you to test your assumptions regarding the design.
2. The sample data provide a convenient way to check the accuracy of your design.
3. The sample data help improve communications with users in discussing your design.
4. The sample data can be used to develop prototype applications and to test queries.

INTEGRITY CONSTRAINTS: The relational data model includes several types of constraints, or rules limiting acceptable values and actions, whose purpose is to facilitate maintaining the accuracy and integrity of data in the database.

The three major types of constraints are…

Domain constraints: is the set of values that may be assigned to an attribute

- It usually consists of domain name, meaning, data type, size, and allowable values or ranges.
- Example

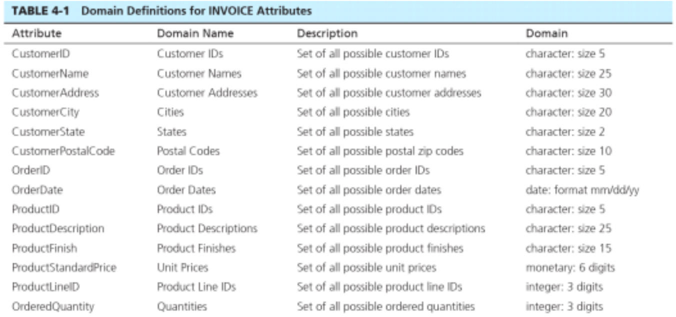


TABLE 4-1 Domain Definitions for INVOICE Attributes

Attribute	Domain Name	Description	Domain
CustomerID	Customer IDs	Set of all possible customer IDs	character: size 5
CustomerName	Customer Names	Set of all possible customer names	character: size 25
CustomerAddress	Customer Addresses	Set of all possible customer addresses	character: size 30
CustomerCity	Cities	Set of all possible cities	character: size 20
CustomerState	States	Set of all possible states	character: size 2
CustomerPostalCode	Postal Codes	Set of all possible postal zip codes	character: size 10
OrderID	Order IDs	Set of all possible order IDs	character: size 5
OrderDate	Order Dates	Set of all possible order dates	date: format mm/dd/yy
ProductID	Product IDs	Set of all possible product IDs	character: size 5
ProductDescription	Product Descriptions	Set of all possible product descriptions	character: size 25
ProductFinish	Product Finishes	Set of all possible product finishes	character: size 15
ProductStandardPrice	Unit Prices	Set of all possible unit prices	monetary: 6 digits
ProductLineID	Product Line IDs	Set of all possible product line IDs	integer: 3 digits
OrderedQuantity	Quantities	Set of all possible ordered quantities	integer: 3 digits

Entity integrity: is designed to ensure that every relation has a primary key and that the data values for that primary key are all valid.

- In particular, it guarantees that every primary key attribute is non-null.
  - There are times when other attributes that are not the primary key can be null (absence of value)

Referential integrity: is a rule that maintains consistency among the rows of two relations.

- The rule states that if there is a foreign key in one relation, either each foreign key value must match a primary key value in another relation or the foreign key value must be null.