Data Model

Week 2

STQD 6324

What is a Data Model?

- A data model is a conceptual representation of the data structures that are required by a database.
- The data structures include the data objects, the associations between data objects, and the rules which govern operations on the objects.
- As the name implies, the data model focuses on what data is required and how it should be organized rather than what operations will be performed on the data.
- To use a common analogy, the data model is equivalent to an architect's building plans.

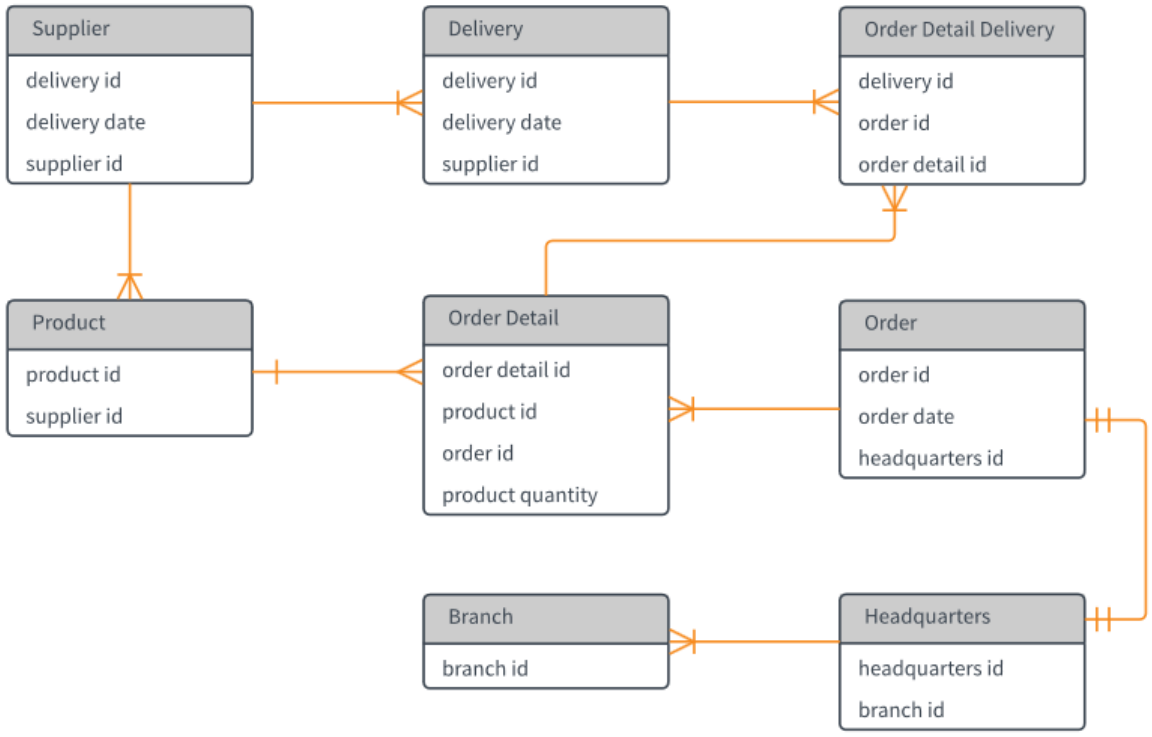
Building plan

What is a Data Model?

- A data model is independent of hardware or software constraints.
- Rather than try to represent the data as a database would see it, the data model focuses on representing the data as the user sees it in the "real world".
- It serves as a bridge between the concepts that make up real-world events and processes and the physical representation of those concepts in a database.

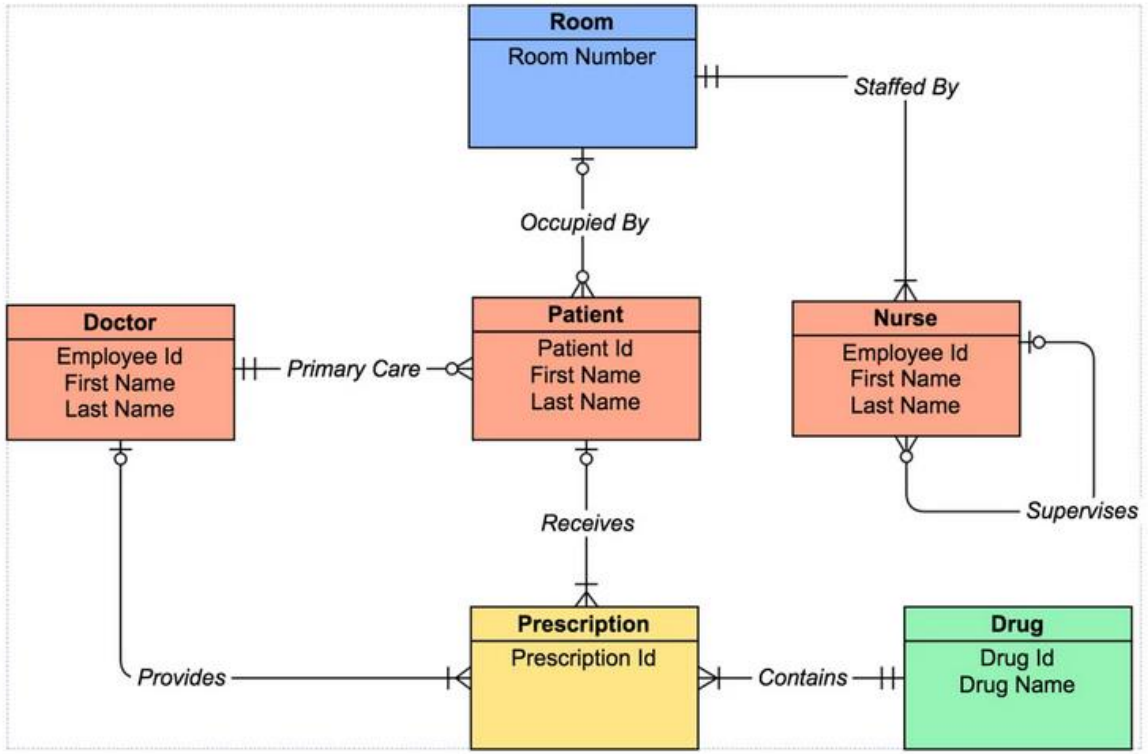
Components of Data Model

- The data model gets its inputs from the planning and analysis stage.
- Here the modeler, along with analysts, collects information about the requirements of the database by reviewing existing documentation and interviewing end-users.
- The data model has two outputs.
  - The first is an entity-relationship diagram which represents the data structures in a pictorial form. Because the diagram is easily learned, it is valuable tool to communicate the model to the end-user.
  - The second component is a data document. This a document that describes in detail the data objects, relationships, and rules required by the database.
  - The dictionary provides the detail required by the database developer to construct the physical database.



Entity-Relationship Diagram

ERD diagrams



This sheet explains al the codes that are used in the DATA sheet.

CODE	range	explanation
SITE_ID	text	Each site has his own unique abbreviation and number as is shown on the location sheet
PLOT_ID	1-2	Each site has two sample locations: one for oak habitat (1) and one for away from the oak habitat (2)
TREAT	1-4	There are four different treatments types: individually surface (1), clumped surface (2), individually burried (3) and clumped burried(4)
PHO_ID	text and #	Code for each acorn that was used during the experiments and storing the photo's. K stands for cage, p for metal plate, GK for clumped cage and GP for clumbed plate
ACORN_ID	1-64	Within a subplot each acorn has a unique ID
TAKEN	0-1	In some cases the acorn was missing in this case a 0 is scored when it's still pressence a 1 is given as a score for this category
MOLD	0-1	When there is any type of mold pressent on the acorn a 1 is scored if not a 0 is scored
INS_D	0-1	When there is vissible insect damage of an insect present a 1 is scored if not a 0 is scored
INTA	0-1	When there is no mold, blackrot or insect damage a 1 is scored in all other cases a 0 is scored
BL_RO	0-1	When a percentage of the acorn is affected by blackrot a 1 is scored if not a 0 is scored
Des	0-1	When the acorn is desiccated a 1 is scored if the acorn is still fresh a 0 is scored but an 1 score is given at the INTA collumn
p_RO	0-1	When the acorn produced a root a 1 is scored if there is no root present a 0 is scored, but at least one of the above categories should have an 1
p_RoS	0-1	When the acorn produced both a root an shoot a 1 is scored if not a 0 is scored
Em	0-1	When the acorn emerged above the surface a 1 is scored if not a 0 is scored
Mild	0-1	When Em is scored with a 1 and mildew is pressent a 1 is scored if not a 0 is scored
DRY	0-1	When the leaves are withered a 1 is scored if not a 0
Brow	0-1	When Em is scored with a 1 and the seedling is browsed a 1 is scored if not a 0 is scored
INTA	0-1	When Em is scored with a 1 and there is no sign of mildew, browsing or another kind of pest/disease a 1 is scored
Aff	0-1	When Em is scored with a 1 and the acorn is affected in another way a 1 is scored
S_LO	0-2	number of seed lobs germinated
STEM	0-2	number of stems
LEAV	0>	number of leaves
ST_L	0>	stem lenght
RO_th	0-1	thickness of the root

Data Document

DATA

emlployee_id	first_name	last_name	nin	dept_id
44	Simon	Martinez	HH 45 09 73 D	1
45	Thomas	Goldstein	SA 75 35 42 B	2
46	Eugene	Cornelsen	NE 22 63 82	2
47	Andrew	Petculescu	XY 29 87 61 A	1
48	Ruth	Stadick	MA 12 89 36 A	15
49	Barry	Scardelis	AT 20 73 18	2
50	Sidney	Hunter	HW 12 94 21 C	6
51	Jeffrey	Evans	LX 13 26 39 B	6
52	Doris	Berndt	YA 49 88 11 A	3
53	Diane	Eaton	BE 08 74 68 A	1

DATA DICTIONARY (METADATA)

Column	Data Type	Description
emlployee_id	int	Primary key of a table
first_name	nvarchar(50)	Employee first name
last_name	nvarchar(50)	Employee last name
nin	nvarchar(15)	National Identification Number
position	nvarchar(50)	Current postion title, e.g. Secretary
dept_id	int	Employee deparmtnet. Ref: Departmetns
gender	char(1)	M = Male, F = Female, Null = unknown
employment_start_date	date	Start date of employment in organization.
employment_end_date	date	Employment end date.

Data Dictionary

Why Data Modeling is important?

- The goal of the data model is to make sure that the all data objects required by the database are completely and accurately represented.
- The data model uses easily understood notations and natural language, it can be reviewed and verified as correct by the end-users.
- The information contained in the data model will be used to define the relational tables, primary and foreign keys, stored procedures, and triggers.
- Data models can facilitate interaction among the designer, the applications programmer, and the end user.

FirstName	LastName	CustID
Elaine	Stevens	101
Mary	Dittman	102
Skip	Stevenson	103
Drew	Lakeman	104
Eva	Plummer	105

Parent Table

Primary Key

CustID	ContactInformation	ContactType
101	555-2653	Work
101	555-0057	Cell
102	555-8816	Work
104	555-0949	Work
103	555-0650	Work
101	555-8855	Home
105	Plummer@akcomms.com	Email
101	Stevens@akcomms.com	Email
101	555-5787	Fax
103	Stevenson@akcomms.com	Email
105	555-5675	Work
102	Dittman@akcomms.com	Email

Foreign Key

Child Table

One to Many Relationship

Primary key and foreign key

Purchase table

Transaction ID	Customer ID	Product ID	Purchange date
1112	24221	8977	03-22-2010
1113	24222	8978	03-22-2010
1114	24223	8979	03-22-2010

Customer ID	Customer	Address
24221	Bob	123 East street
24222	Alice	223 Main street
24223	Martha	465 North street

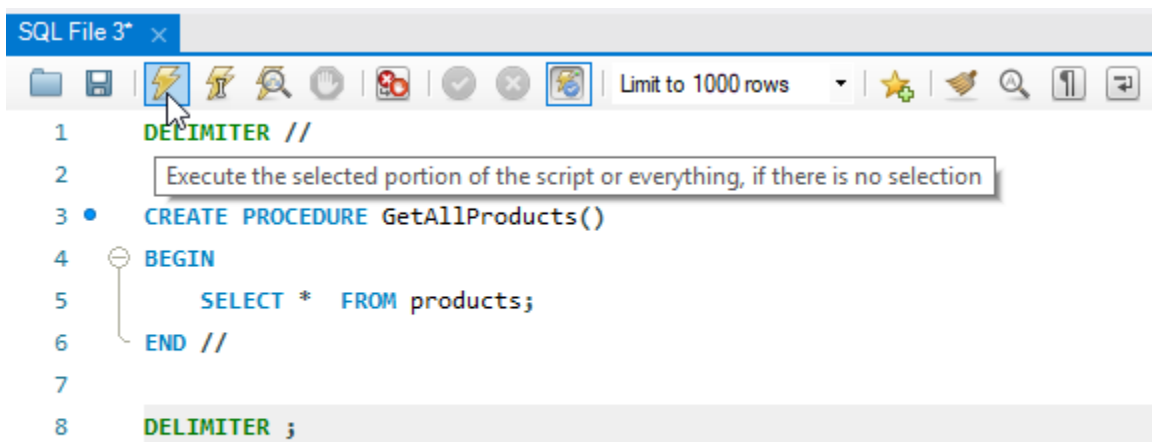
Customer table

Product ID	Name	Price
8977	Banana	.79
8978	TV	400
8979	Watch	50

Product table

Relational Table

Stored Procedure



Why Data Modeling is important?

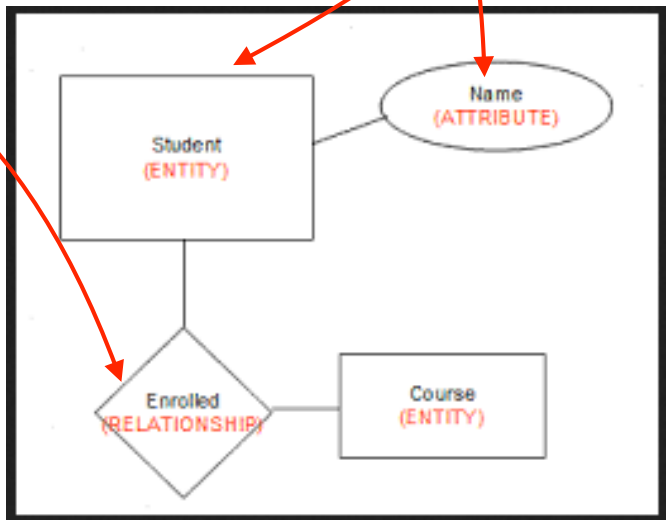
- Are a communication tool
- Give an overall view of the database
- Organize data for various users
- Are an abstraction for the creation of good database

Data Model Basic Building Blocks

(entities are typically depicted as rectangles)

- **Entity**: Unique and distinct object used to collect and store data
- **Attribute**: Characteristic of an entity *(depicted as ovals connected to their respective entities)*
- **Relationship**: Describes an association among entities
  - One-to-many (1:M)
  - Many-to-many (M:N or M:M)
  - One-to-one (1:1)

diamond shape



- **Constraint**: Set of rules to ensure data integrity
  - *uniqueness constraints (e.g., ensuring each customer has a unique ID)*
  - *referential integrity constraints (e.g., ensuring that foreign keys match primary keys)*
  - *domain constraints (e.g., ensuring that a date falls within a certain range)*

Data Model Basic Building Blocks : Entity

- An entity is a person, place, thing, or event about which data will be collected and stored.
- An entity represents a particular type of object in the real world, which means an entity is "distinguishable"—that is, each entity occurrence is unique and distinct.
- For example, a **CUSTOMER** entity would have many distinguishable customer occurrences, such as John Smith, Pedro Dinamita, and Tom Strickland.
- Entities may be physical objects, such as customers or products, but entities may also be abstractions, such as flight routes or musical concerts.

Data Model Basic Building Blocks : Attribute

- An attribute is a characteristic of an entity.
- For example, a CUSTOMER entity would be described by attributes such as customer last name, customer first name, customer phone number, customer address, and customer credit limit.
- Attributes are the equivalent of fields in file systems.

Data Model Basic Building Blocks : Relationship

- A relationship describes an association among entities.
- For example, a relationship exists between customers and agents that can be described as follows: an agent can serve many customers, and each customer may be served by one agent.
- Data models use three types of relationships: one-to-many, many-to-many, and one-to-one.
- Database designers usually use the shorthand notations 1:M or 1..*, M:N or *..*, and 1:1 or 1..1, respectively.

One-to-many (1:M or 1..*) relationship

- A painter creates many different paintings, but each is painted by only one painter.
- Therefore, database designers label the relationship "PAINTER paints PAINTING" as 1:M.
- Note that entity names are often capitalized as a convention, so they are easily identified.
- Similarly, a customer (the "one") may generate many invoices, but each invoice (the "many") is generated by only a single customer.
- The "CUSTOMER generates INVOICE" relationship would also be labeled 1:M.

Many-to-many (M:N or *..*) relationship

- An employee may learn many job skills, and each job skill may be learned by many employees.
- Database designers label the relationship "EMPLOYEE learns SKILL" as M:N.
- Similarly, a student can take many classes and each class can be taken by many students, thus yielding the M:N label for the relationship expressed by "STUDENT takes CLASS."

One-to-one (1:1 or 1..1) relationship.

- A retail company's management structure may require that each of its stores be managed by a single employee.
- In turn, each store manager, who is an employee, manages only a single store.
- Therefore, the relationship "EMPLOYEE manages STORE" is labeled 1:1.

Data Model Basic Building Blocks : Constraints

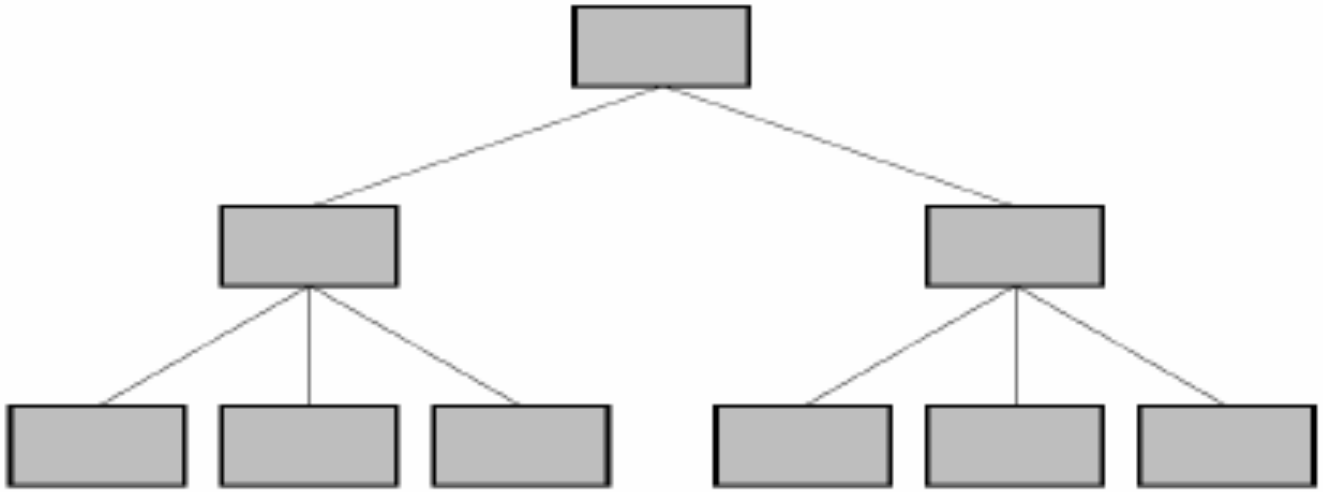
- A constraint is a restriction placed on the data.
- Constraints are important because they help to ensure data integrity.
- Constraints are normally expressed in the form of rules:
  - An employee's salary must have values that are between 6,000 and 350,000.
  - A student's GPA must be between 0.00 and 4.00.
  - Each class must have one and only one teacher.

Types of Data Model

- Hierarchical Model
- Network Model
- Relational Model
- Entity Relationship Model
- Object Oriented Model

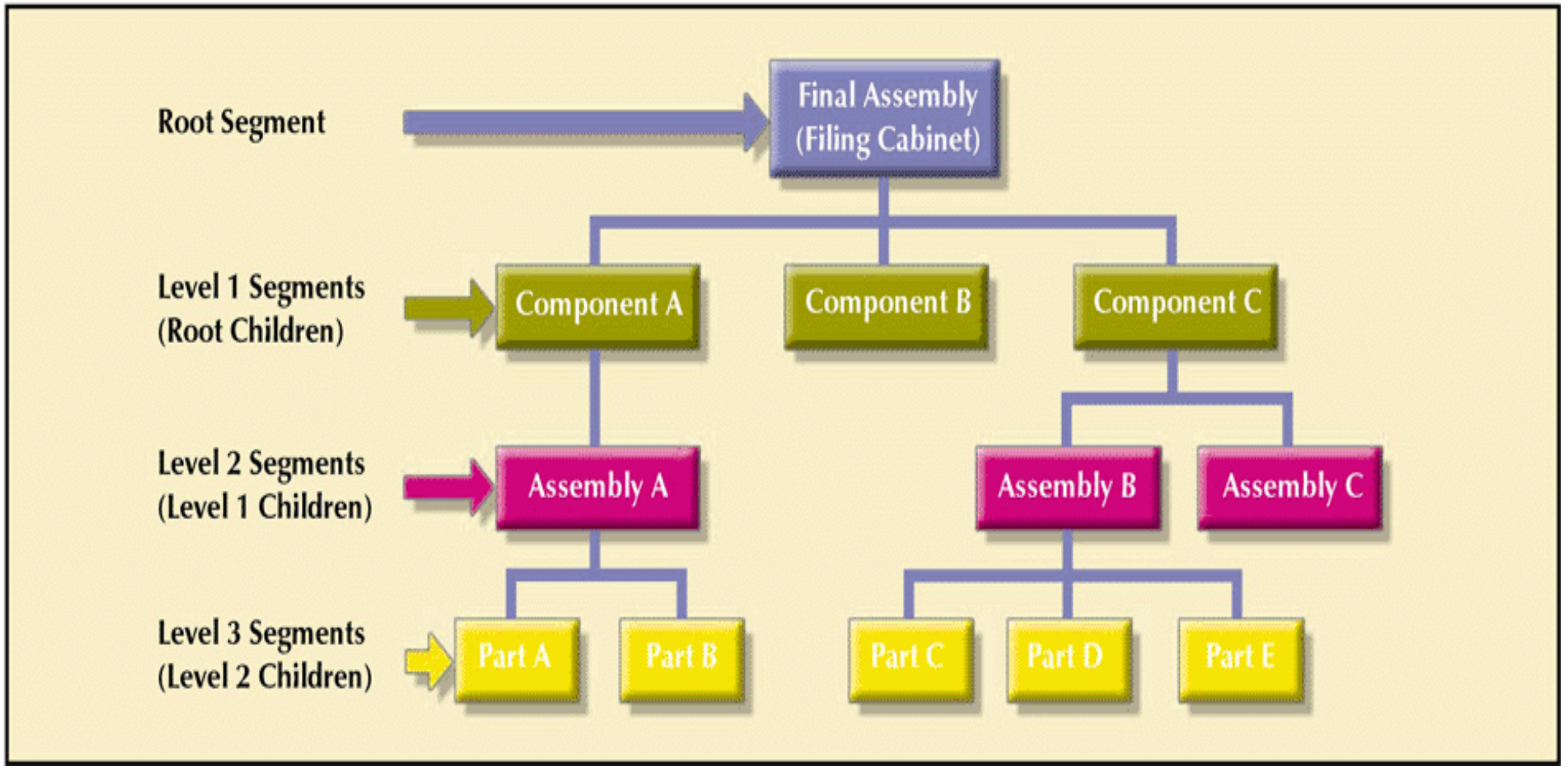
Hierarchical Model

- A hierarchical data model is a data model which the data is organized into a tree like structure.
- The structure allows repeating information using parent/child relationships:
  - Each parent can have many children but each child only has one parent.
  - All attributes of a specific record are listed under an entity type.



Hierarchical Model

FIGURE 2.1 A HIERARCHICAL STRUCTURE

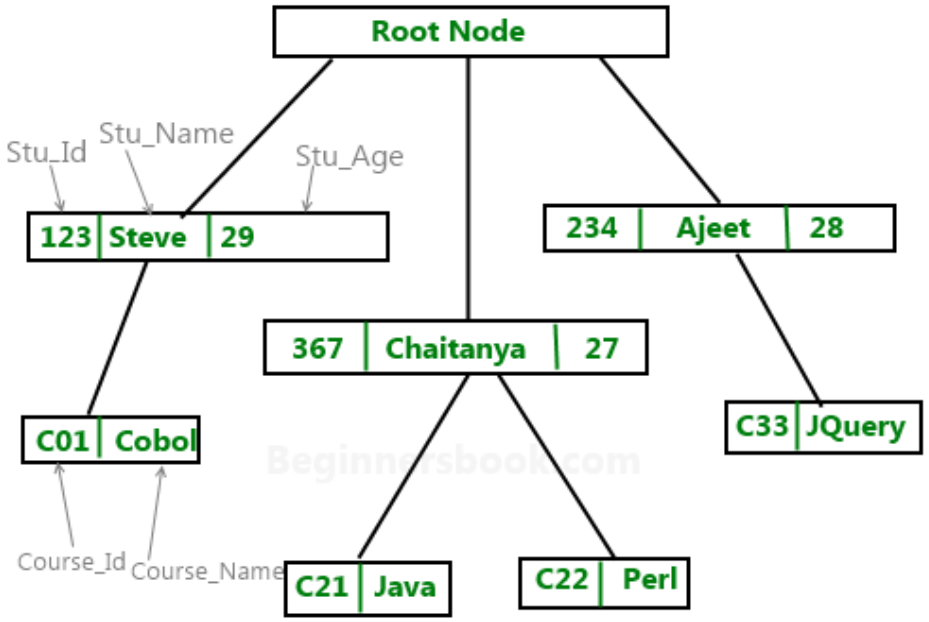


Best understood by examining manufacturing process

Hierarchical Structure—Characteristics

- Each parent can have many children
- Each child has only one parent
- Tree is defined by path that traces parent segments to child segments, beginning from the left
- Hierarchical path
  - Ordered sequencing of segments tracing hierarchical structure
- Preorder traversal or hierarchic sequence
  - “Left-list” path
  - If Part D is most frequently accessed and updated, change the database structure to place Part D closer to the left side of the tree
  - This will give a shorter traversal

Sample Hierarchical Model Diagram:



- In **hierarchical model**, data is organized into a tree like structure with each record is having one parent record and many children.
- The main drawback of this model is that, it can have only one to many relationships between nodes.

Example of hierarchical data represented as relational tables: The above hierarchical model can be represented as relational tables like this:

Stu_Id	Stu_Name	Stu_Age
123	Steve	29
367	Chaitanya	27
234	Ajeet	28

Student table

Course_Id	Course_Name	Stu_Id
C01	Cobol	123
C21	Java	367
C22	Perl	367
C33	JQuery	234

Course table

Hierarchical Model

- Advantages:
  1. The representation of records is done using an ordered tree, which is natural
  2. Method of implementation of one–to-many relationships.
  3. Proper ordering of the tree results in easier and faster retrieval of records.
  4. Allows the use of virtual records. This result in a stable database especially when modification of the data base is made.
  5. Database integrity – always a link between parent and child
  6. Efficiency – very efficient when it contains a large volume of data in 1:M relationships and whose relationships are fixed over time

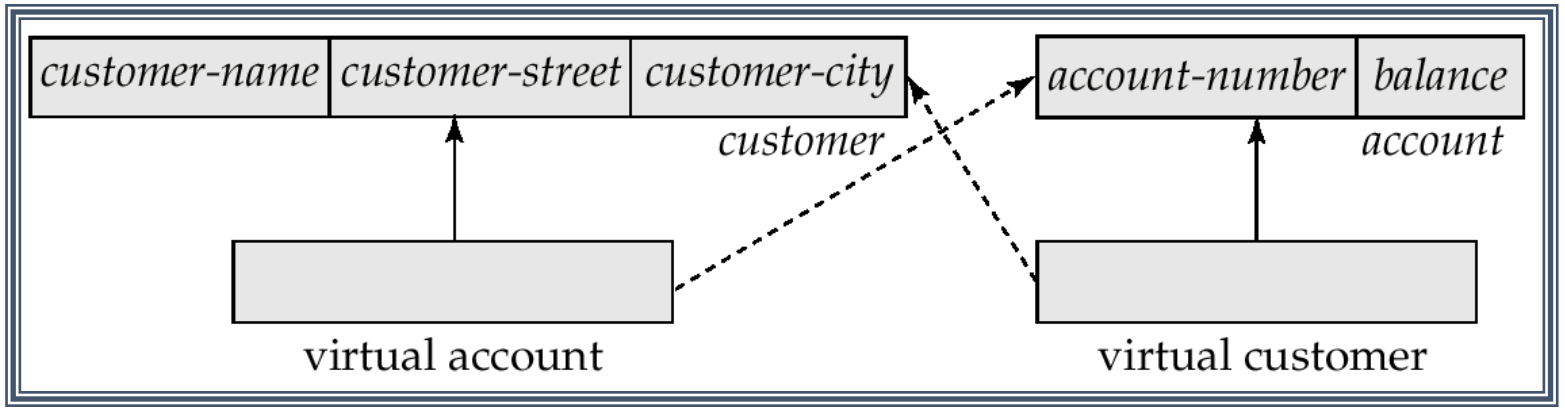
Virtual records

- Contains no data value, only a logical pointer to a particular physical record
- When a record is to be replicated in several database trees, a single copy of that record is kept in one of the trees and all other records are replaced with a virtual record
- Let *R* be a record that is replicated in T_1, T_2, … T_n. Create a new virtual record type *virtual-R* and replace *R* in each of the $n - 1$ trees with a record of type *virtual-R*

https://bit.ly/3iSyqUq

Virtual records (cont.)

- Eliminate data replication; create *virtual-customer* and *virtual-account*.
- Replace account with *virtual-account* in the first tree, and replace customer with *virtual-customer* in the second tree
- Add a dashed line from virtual-customer to customer, and from virtual-account to account, to specify the association between a virtual record and its corresponding physical record.



https://bit.ly/3iSyqUq

Hierarchical Model

- Disadvantages:

1. Complex implementation – detailed knowledge of the physical data storage characteristics is required by the designers and programmers
2. Difficult to manage – relocation of segments requires application changes
3. Lacks structural independence
4. Implementation limitations – difficult to support M:N relationships
5. Lack of standards – no standard **DDL** and no **DML**

Data Definition Language

create, modify, and delete database objects

Data Manipulation Language

manipulate data stored in the database

DDL and DML

Key	DDL	DML
Stands for	Data Definition Language	Data Manipulation Language
Usage	DDL statements are used to create database, schema, constraints, users, tables etc.	DML statement is used to insert, update or delete the records.
Commands	CREATE, DROP, RENAME and ALTER	INSERT, UPDATE and DELETE

https://bit.ly/35p809u

DDL vs. DML

DDL

```
[ ]  CREATE TABLE Students (
         ID INT PRIMARY KEY,
         Name VARCHAR(50),
         Age INT
     );
```

DML

```
[ ]  INSERT INTO Students (ID, Name, Age) VALUES (1, 'John Doe', 20);
```

Hierarchical model : Summary

- Deletion problem: If a parent is deleted the child is deleted automatically
- Minimize disk input and output time: Parent and child are stored close to each other on the storage device. It helps minimize the hard disk input and disk output time.
- Fast navigation: Due to short distance between parent to child, database access time and performance is improved. Navigation through the database is very fast.
- Predefined relationships between records : All relation ship are predefined, Root nodes, parent and child are predefined in the database schema.
- Difficult to re-organize: Difficult to re-organize because parent to child relationships can be disturbed.

Types of Data Model

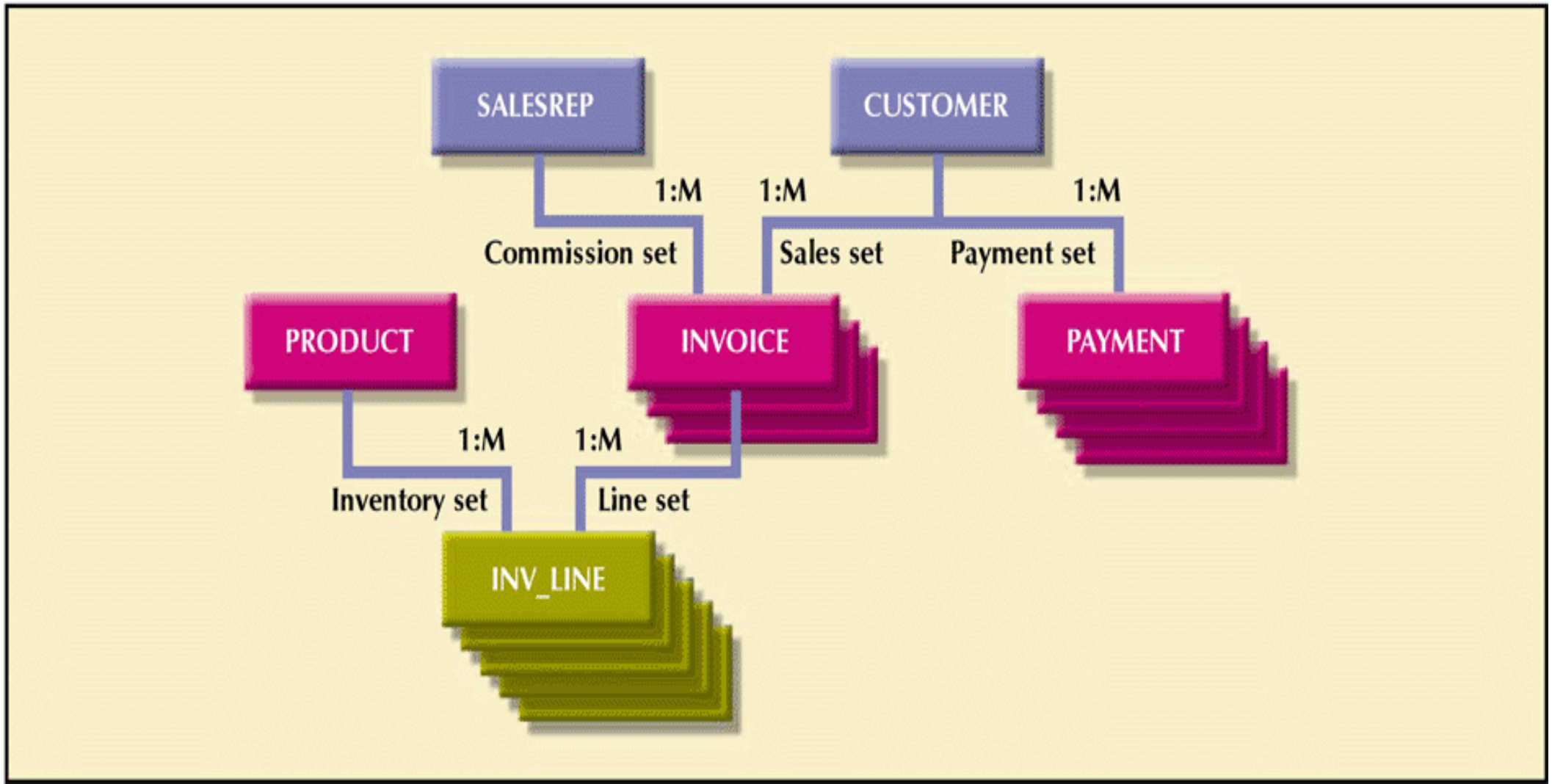
- Hierarchical Model
- Network Model
- Relational Model
- Entity Relationship Model
- Object Oriented Model

Network Model

- Created to
  - Represent complex data relationships more effectively
  - Improve database performance
  - Impose a database standard
- Resembles hierarchical model
- Collection of records in 1:M relationships
  - A relationship is called a Set
  - Composed of at least two record types
    - Owner
      - Equivalent to the hierarchical model's parent
    - Member
      - Equivalent to the hierarchical model's child
      - A record can appear as a member in more than one set i.e., a member may have multiple owners

Network Model

FIGURE 2.3 A NETWORK DATA MODEL



Network Model

- Advantages
  - Conceptual simplicity
  - Handles more relationship types
  - Data access flexibility – no need for a preorder traversal
  - Promotes database integrity – must first define the owner and then the member record
  - Data independence
  - Conformance to standards

Network Model

- Disadvantages
  - System complexity
  - Lack of structural independence

Network Model : Summary

- Better than hierarchical model
- Supports many to many relationships
  - Many parent can have many child
  - Many child can have many parents
- Entities are represented as network
- One child entity can have more than one parent entity
- Entities can have multiple parent entities and leads to a complex structure
- Not very flexible to re-organize the model
- High performance

Types of Data Model

- Hierarchical Model
- Network Model
- Relational Model
- Entity Relationship Model
- Object Oriented Model

Relational Model

- Developed by Codd (IBM) in 1970
- Considered ingenious but impractical in 1970
- Conceptually simple
- Computers lacked power to implement the relational model
- Today, microcomputers can run sophisticated relational database software

Relational Model

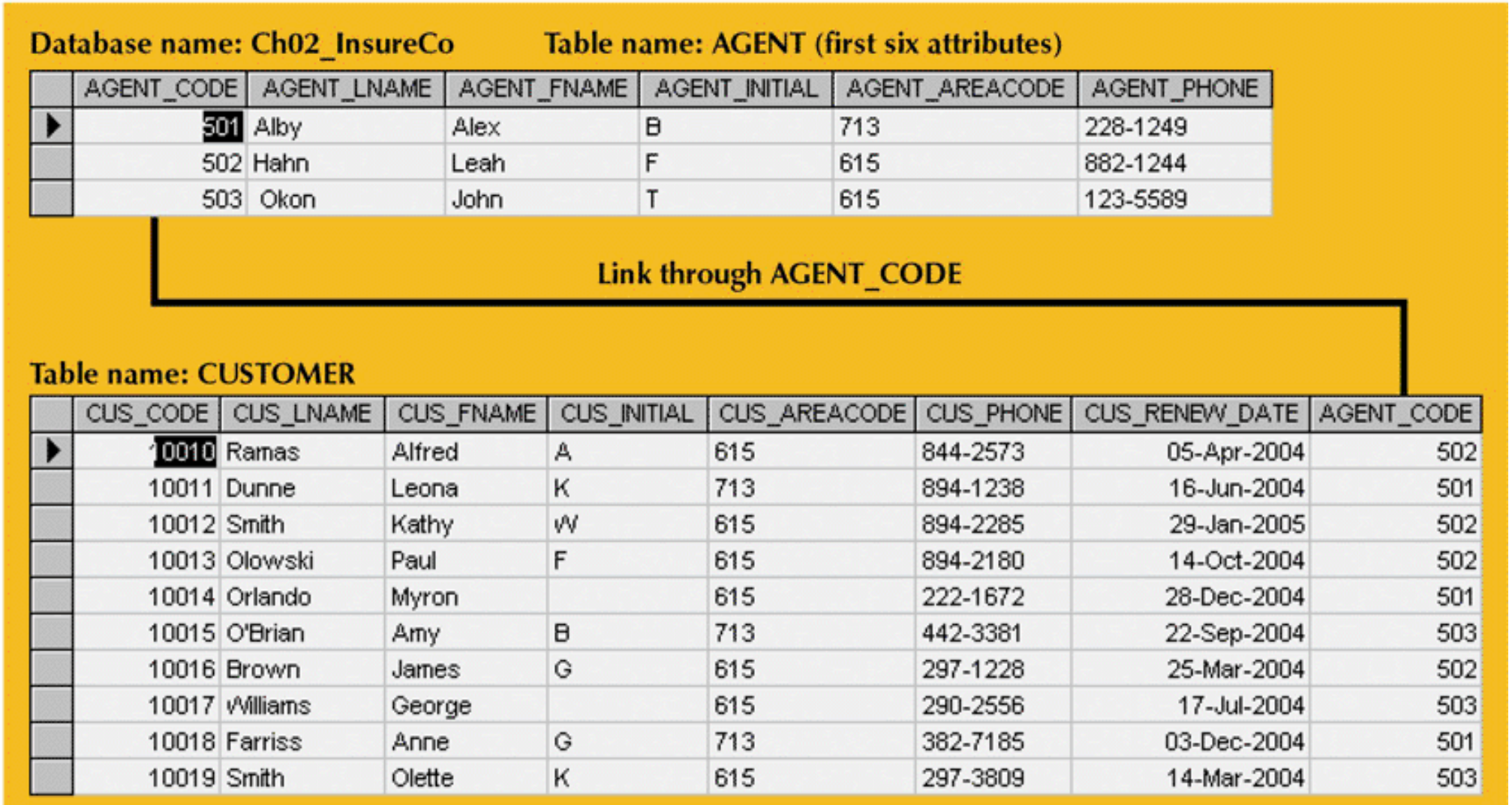
- Relational Database Management System (RDBMS)
- Performs same basic functions provided by hierarchical and network DBMS systems, plus other functions
  - RDBMS handles all the complex physical details
- Most important advantage of the RDBMS is its ability to let the user/designer operate in a human logical environment

Relational Model : Basic Structure

- Table (relations)
  - Matrix consisting of a series of row/column intersections
  - Related to each other by sharing a common entity characteristic
- Relational schema
  - Visual representation of relational database's entities, attributes within those entities, and relationships between those entities

Linking Relational Tables

FIGURE 2.4 LINKING RELATIONAL TABLES



Database name: Ch02_InsureCo **Table name: AGENT (first six attributes)**

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT_CODE

Table name: CUSTOMER

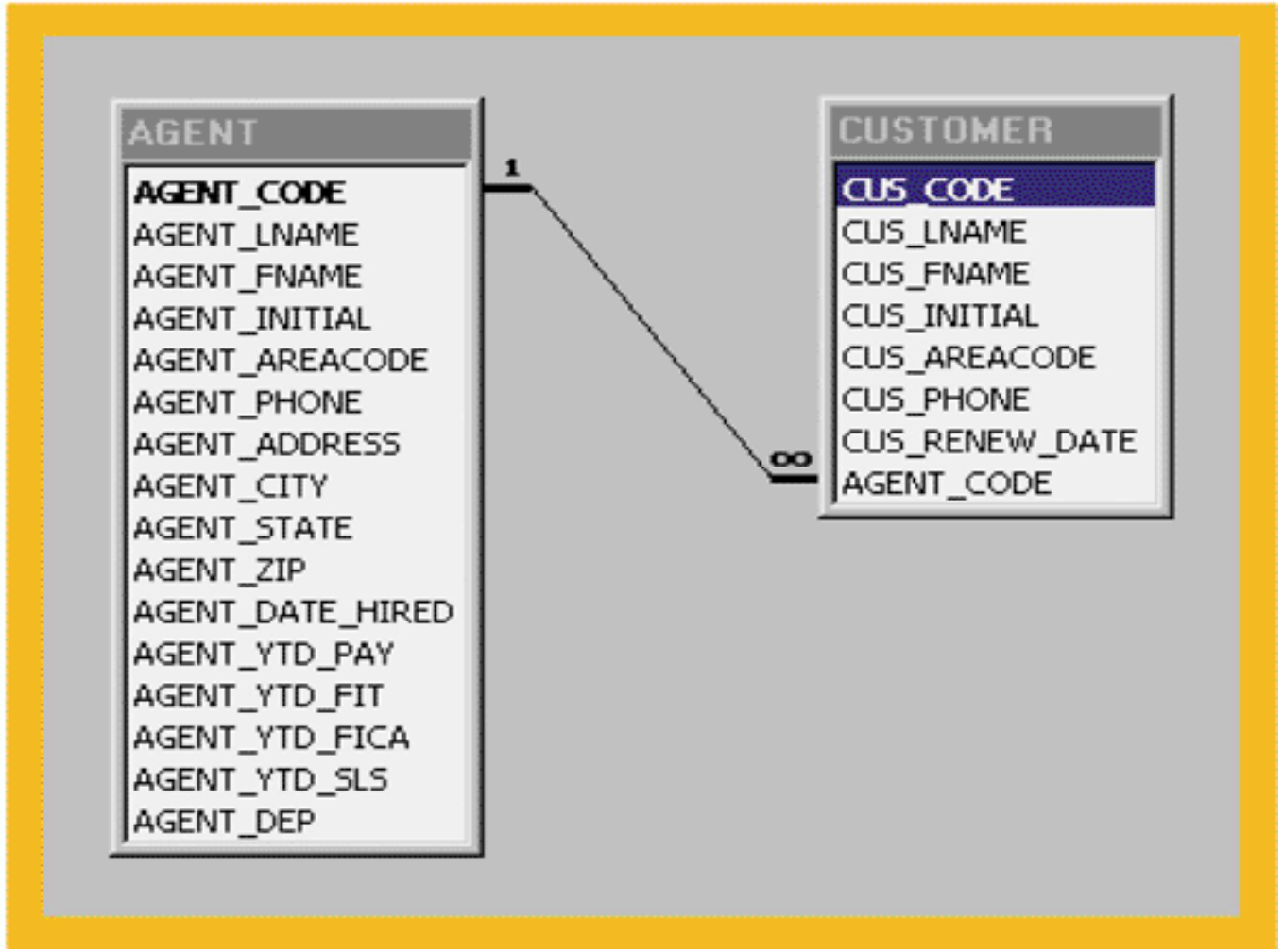
CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	05-Apr-2004	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2004	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2005	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2004	502
10014	Orlando	Myron		615	222-1672	28-Dec-2004	501
10015	O'Brian	Amy	B	713	442-3381	22-Sep-2004	503
10016	Brown	James	G	615	297-1228	25-Mar-2004	502
10017	Williams	George		615	290-2556	17-Jul-2004	503
10018	Farriss	Anne	G	713	382-7185	03-Dec-2004	501
10019	Smith	Olette	K	615	297-3809	14-Mar-2004	503

Relational Model

- Stores a collection of related entities
  - Resembles a file
- Relational table is purely logical structure
  - How data are physically stored in the database is of no concern to the user or the designer
  - This property became the source of a real database revolution

A Relational Schema

FIGURE 2.5 A RELATIONAL SCHEMA



Relational Model

- **Advantages**
  - Structural independence – changes in the relational data structure do not affect the DBMS's data access in any way
  - Improved conceptual simplicity by concentrating on the logical view
  - Easier database design, implementation, management, and use
  - Ad hoc query capability - SQL
  - Powerful database management system

Relational Model

- **Disadvantages**
  - Substantial hardware and system software overhead (Cost)
  - Can facilitate poor design and implementation
  - May promote “islands of information” problems
    - For example, maybe the hospital billing department used one database while the hospital personnel department used a different database.
    - Getting those databases to "talk" to each other can be a large, and expensive, undertaking, yet in a complex hospital system, all the databases need to be involved for good patient and employee care.

Types of Data Model

- Hierarchical Model
- Network Model
- Relational Model
- Entity Relationship Model
- Object Oriented Model

Entity Relationship Model

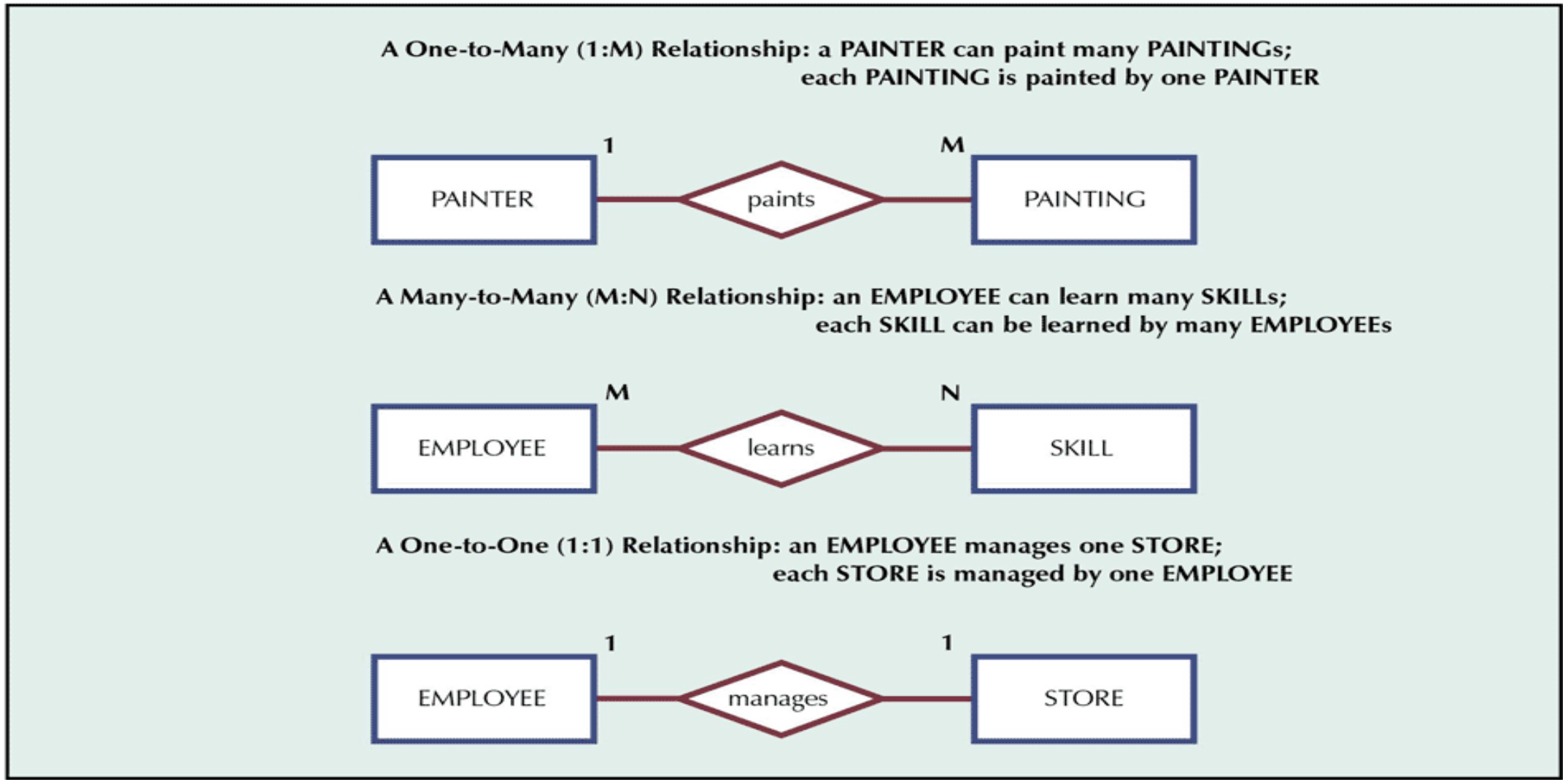
- Widely accepted and adapted graphical tool for data modeling
- Introduced by Peter Chen in 1976
- Graphical representation of entities and their relationships in a database structure

Entity Relationship Model—Basic Structure

- Entity relationship diagram (ERD)
  - Uses graphic representations to model database components
  - Entity is mapped to a relational table
- Entity instance (or occurrence) is row in table
- Entity set is collection of similar entities
- Connectivity labels types of relationships
  - Diamond connected to related entities through a relationship line

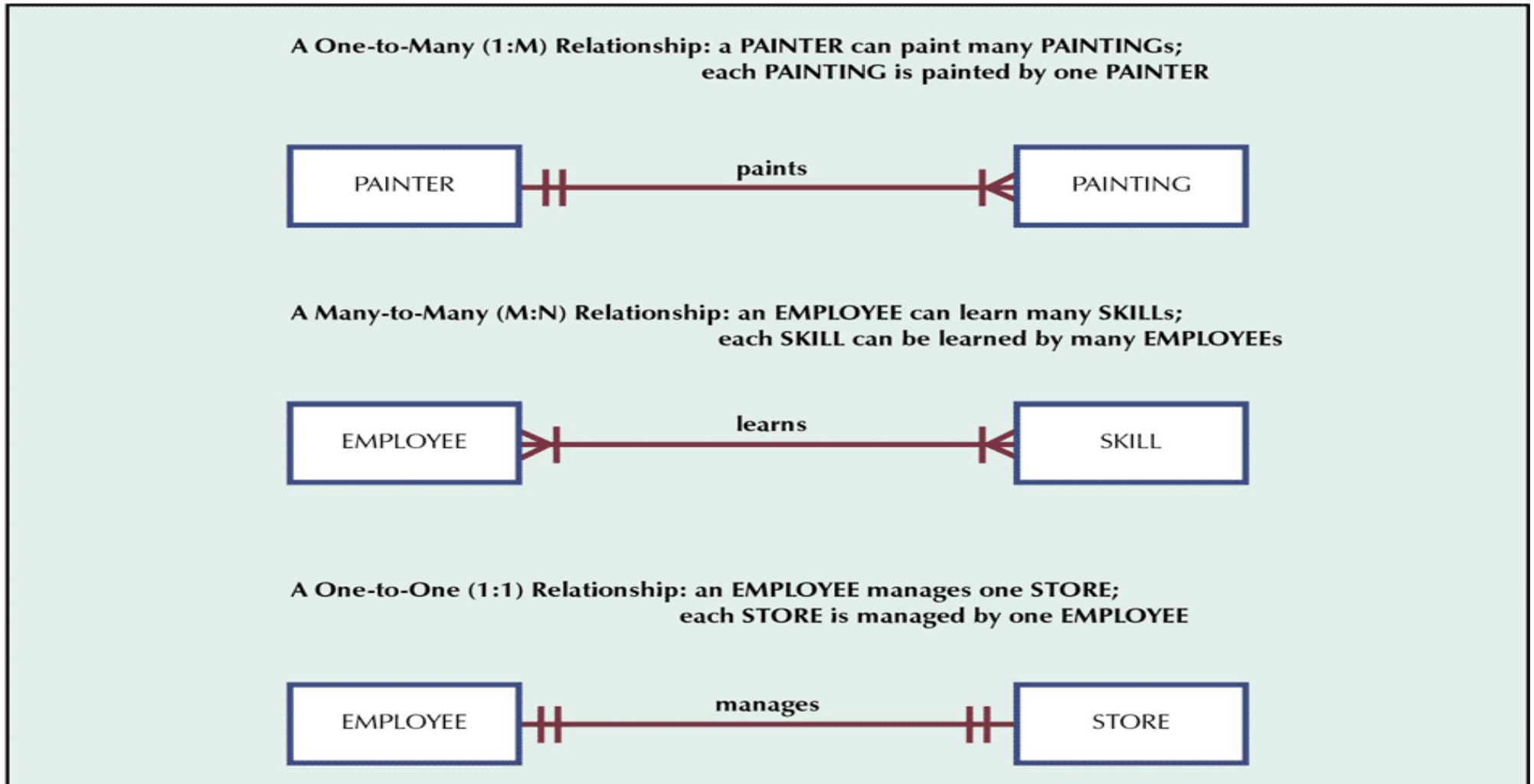
Relationships : The Basic Chen ERD

FIGURE 2.6 RELATIONSHIPS: THE BASIC CHEN ERD



Relationships : The Basic Crow's Foot ERD

FIGURE 2.7 RELATIONSHIPS: THE BASIC CROW'S FOOT ERD



Symbols in crow's foot notation

A dash represents “one”

A crow’s foot represents “many” or “infinite”

DESCRIPTION	SYMBOL
Ring and dash: Minimum zero, maximum one (optional)	
Dash and dash: Minimum one, maximum one (mandatory)	
Ring and crow's foot: Minimum zero, maximum many (optional)	
Dash and crow's foot: Minimum one, maximum many (mandatory)	

https://bit.ly/3wLNxXQ

Entity-Relationship model

- **Advantages**
  - Exceptional conceptual simplicity
  - Visual representation
  - Effective communication tool
  - Integrated with the relational data model

Types of Data Model

- Hierarchical Model
- Network Model
- Relational Model
- Entity Relationship Model
- Object Oriented Model

Object Oriented Model

- Semantic Data Model (SDM) developed by Hammer and McLeod in 1981
- Modeled both data and their relationships in a single structure known as an object
- Basis of Object Oriented Data Model (OODM)
- OODM becomes the basis for the Object Oriented Database Management System (OODBMS)

Object Oriented Model

- Object is described by its factual content
  - Like relational model's entity
- Includes information about relationships between facts within object and relationships with other objects
  - Unlike relational model's entity
- Subsequent OODM development allowed an object to also contain operations
- Object becomes basic building block for autonomous structures

Developments that Boosted OODM's Popularity

• **software projects became more complex and expensive**

- Growing costs put a premium on code reusability
- Complex data types and system requirements became difficult to manage with a traditional RDBMS
- Became possible to support increasingly sophisticated transaction & information requirements
- Ever-increasing computing power made it possible to support the large computing overhead required

"Write once, use many times"

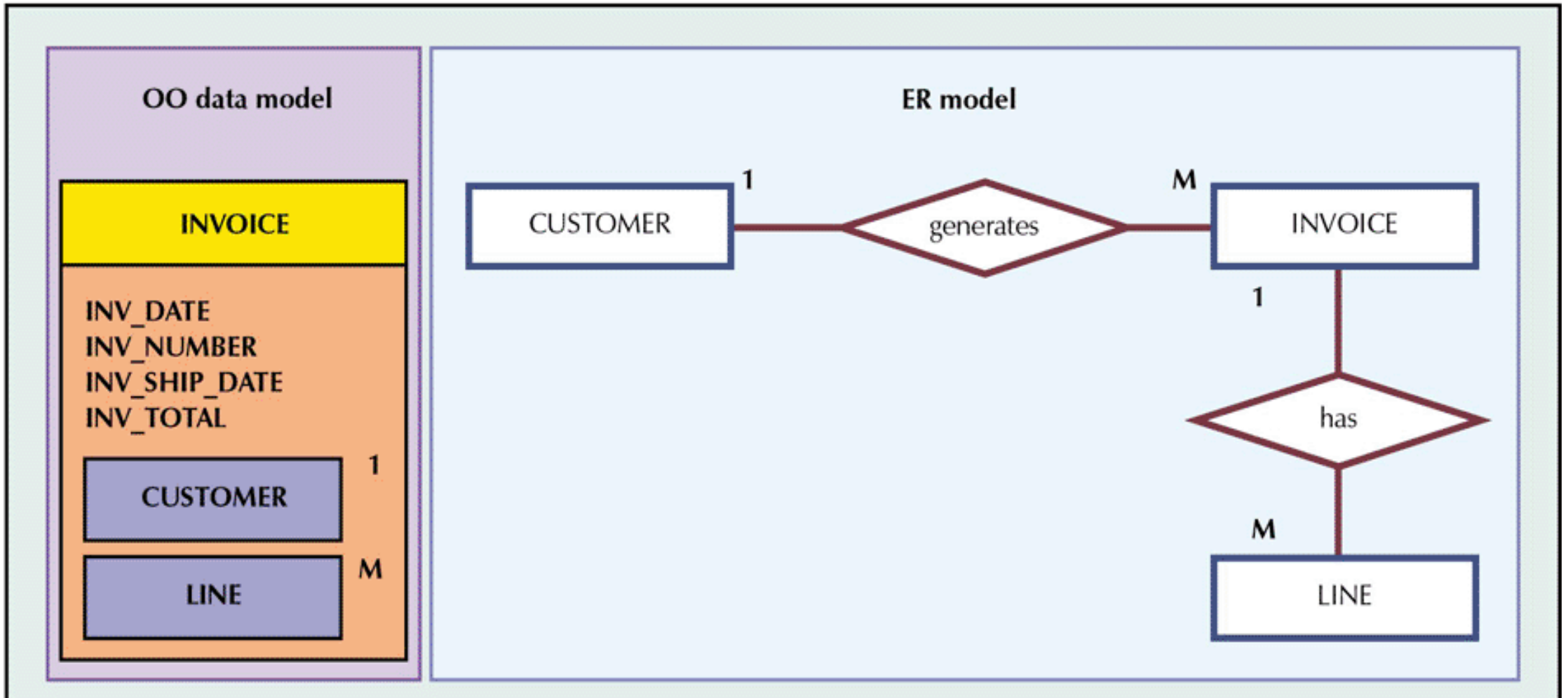
Object Oriented Data Model—Basic Structure

- Object: abstraction of a real-world entity
- Attributes describe the properties of an object
- Objects that share similar characteristics are grouped in classes
- Classes are organized in a class hierarchy
- Inheritance is the ability of an object within the class hierarchy to inherit the attributes and methods of classes above it

Example Python code: https://tinyurl.com/35znhrcy

A Comparison of the OO Model and the ER Model

FIGURE 2.8 A COMPARISON OF THE OO MODEL AND THE ER MODEL



Object Oriented Model

- **Advantages**
  - Adds semantic content **[allows real-world meaning to be embedded]**
  - Visual presentation includes semantic content
  - Database integrity
  - Both structural and data independence

Example:
- a `Car` object can have properties like ‘colour’, ‘speed’, and methods like ‘start()’ or ‘stop()’.
- easier to understand the context of data and its behaviour

Object Oriented Model

- **Disadvantages**
  - Slow pace of OODM standards development
  - Complex navigational data access
  - Steep learning curve
  - High system overhead slows transactions
  - Lack of market penetration

References

- Crow’s Foot Notation, http://www.vertabelo.com/blog/technical-articles/crow-s-foot-notation
- Chen Notation, http://www.vertabelo.com/blog/technical-articles/chen-erd-notation
- UML Notation, http://www.vertabelo.com/blog/technical-articles/uml-notation