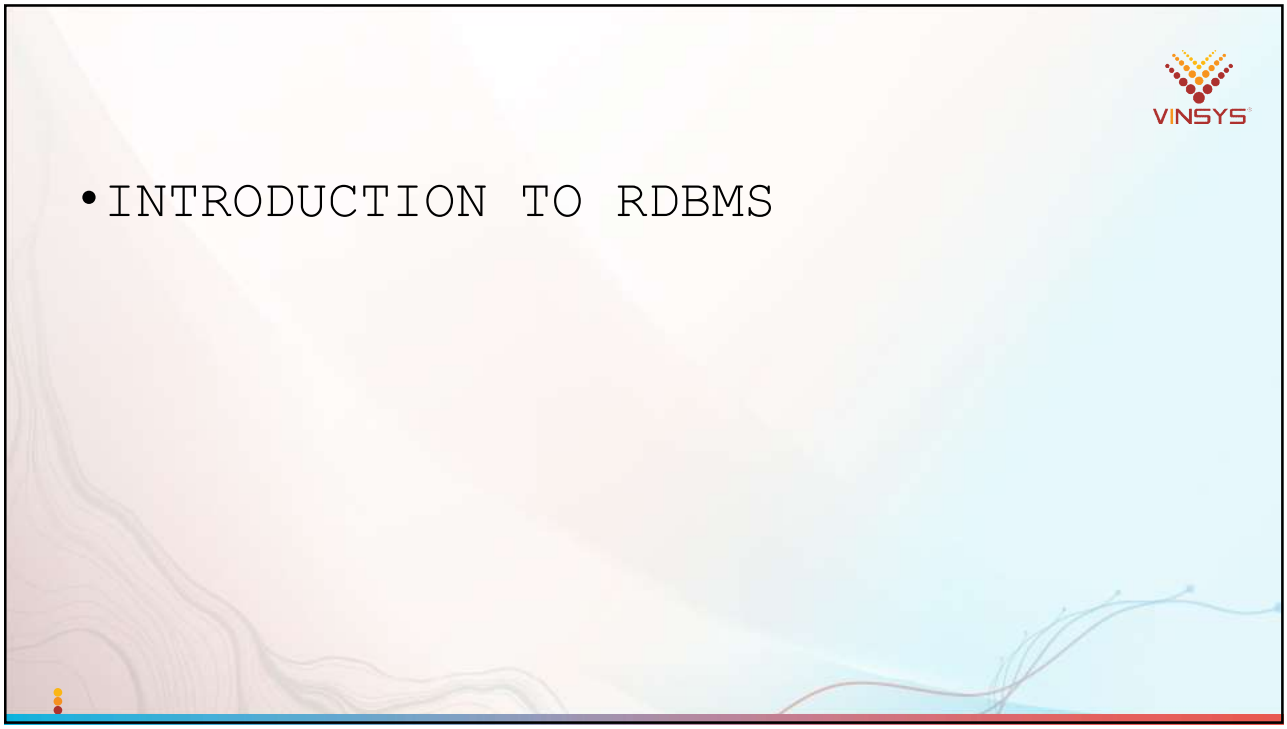

VINSYS
DELiVERiNGSKiLLS
DRiViNGSUCCESS
www.VINSYS.COM


VINSYS
• INTRODUCTION TO RDBMS

Introduction

- What is data
- What is information

Flat File System

- Flat files are data files that contain records with no structured relationships. Additional knowledge is required to interpret these files such as the file format properties.(Delimiters)

Advantages of using Flat Files:

- **Cost effective** - using a flat file costs practically nothing because data is stored as text files. No software is required other than the program that needs to access the data.
- **Platform Independent** - since text files are universally accepted by all server platforms, there is no problem moving your database from one server to another.
- **Very Simple To Understand** - how hard would it be to understand that records in a flat file are stored in one straight line and are separated by delimiters?

Disadvantages of using Flat Files:

- **Low Reliability & Integrity** - flat files are very prone to data corruption especially if the size of the database grows beyond what the server resources are prepared to handle. e.g. 50,000 records in flat file & DML firing constantly.
- **Low Security** - no security feature is built into a text file. It can be opened for viewing by anyone who happens to know where to look.
- **Limited Data Structuring** - as mentioned a while ago, records are stored as lines of text by delimiters
- **Difficult To Integrate With Other Programs** - relating data from one table with another is not possible
- Only Character Type of data can fit into it

Disadvantages of flat file system

- **Data Redundancy & Inconsistency :-**
- Same data may be stored across various files causing repetition of data
- This redundancy causes inconsistent state of data.

- **Concurrency Problem :-**
- Multiple users may access same data at same time causing inconsistency & locking problems

Relational Database Management System (RDBMS)

- Relational databases store data as logically related tables.
- Data is arranged in a tabular(rows-columns) manner
- An important feature of relational system is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.
- Avoids redundancy

RDBMS

- Supports Large Amount Of Data
- **High Security & Reliability** - security features such as password and user levels are usually built-in. Database records are also "locked" whenever they are accessed by a user hence preventing data corruption if and when a second user accesses the same record

RDBMS

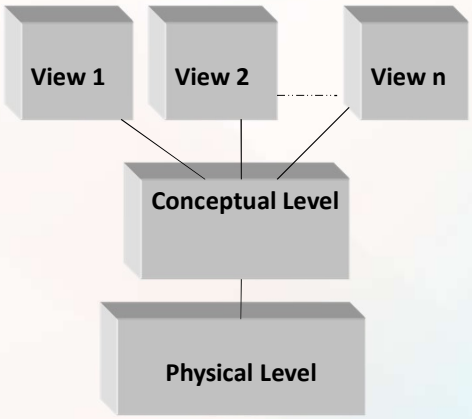
- **Data Abstraction**
- The system hides certain details of how data is stored and maintained.
- Actual complexity should be hidden from database users.

Levels of Abstraction

There are several levels of abstraction:

- Physical Level
- Conceptual Level
- View Level

Levels of Abstraction



Data Independence…

There are two kinds:

- **Physical data independence**
- **Logical data independence**

Different Database Products

- Access –Good for Stand Alone machines
  - Grocery market
- My SQL –Based applications(Free Downloadable Versions) But support only ANSII & work on only Windows .
- Microsoft SQL SERVER –Comes with .Net
- Sybase –For financial applications. Available for Unix, WinNT etc
- ORACLE –Scalability factor, One of the leading database products. Also has wide range of tools

Database Language

- DBMS normally provide specialized programming languages often called *Database Languages*
- Different DBMS provide different database languages although a language called SQL has recently taken on the role of standard.

Database Language…

- DRL/DQL – data query Language
- DDL-data Definition Language
- DML-data Manipulation Language
- DCL-data Control Language
- TCL-transaction Control Language

Database Design

- Stages of SDLC
- requirements specification,
- design
- implementation,
- Testing,
- Deployment,
- Maintenance.

Data Model

- What data is required
- How it should be organized.
- What data should be stored in the database

Is Similar to

- an architect's building plans.
- "blueprint" for building the physical database.

Data Model Types

- Logical Model
- Physical Model

Logical Model

- Logical modeling deals with gathering business requirements and converting those requirements into a model.
- The logical model revolves around the needs of the business, **not the database**, although the needs of the business are used to establish the needs of the database

Physical Model

- Deals with the conversion of the logical, or business model, into a relational database model.
- During physical modeling, objects such as tables and columns are created based on entities and attributes that were defined during logical modeling. Constraints are also defined, including primary keys, foreign keys, other unique keys, and check constraints.

Physical Model

- Physical modeling is **database software specific,** meaning that the objects defined during physical modeling can vary depending on the relational database software being used.

Hierarchical Model

- The hierarchical data model organizes data in a tree structure
- This structure implies that a record can have repeating information, generally in the child data segments.
- Data is in a series of records, which have a set of field values attached to it.
- To create links between these, the hierarchical model uses Parent Child Relationships.
- It restricts a child segment to having only one parent segment.
- *Relationship is One to Many

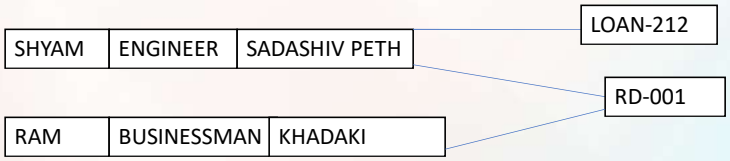
- E.g. Employees getting hardware from the company
  Employees reporting to managers

Network Model

- Some data were more naturally modeled with more than one parent per child.
- So, the network model permitted the modeling of many-to-many relationships in data.
- Here it is not necessary that 1 child must have single parent
- In this model,data is represented by collections of records, and relationships among data are rep-resented by links

Example

Link between Relation-Customer & Relation Account.



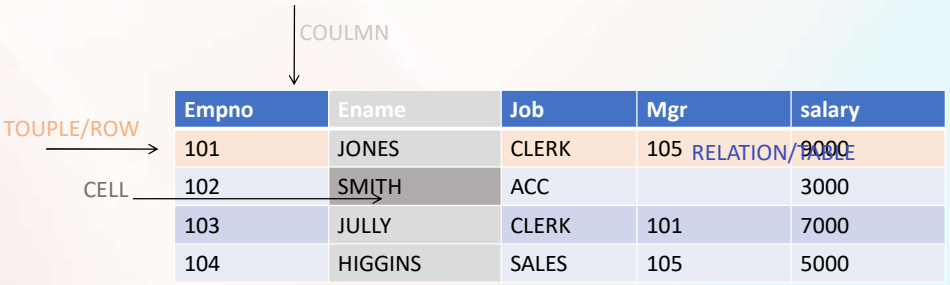
Relational Model

- In such a database the data and relations between them are organised in tables.
- A table is a collection of records and each record in a table contains the same fields.
- Properties of Relational Tables:
  - Values Are Atomic.
  - Each Row is Unique .
  - Column Values Are of the Same Kind .
  - The Sequence of Columns is Insignificant .
  - The Sequence of Rows is Insignificant .
  - Each Column Has a Unique Name

Relational Model

- The tables within same database may be related with each other by defining different cardinalities and constraints
- Entries of attributes are single valued
- Entries of attributes are of same kind
- No two rows are identical
- The order of attributes is unimportant
- The order of rows is unimportant
- Every column can be uniquely identified by its name and not by its position

Example



Empno	Ename	Job	Mgr	salary
101	JONES	CLERK	105	9000
102	SMITH	ACC		3000
103	JULLY	CLERK	101	7000
104	HIGGINS	SALES	105	5000

ORDBMS

- Is same RDBMS but with features of Object Orientation.
- objects, classes and inheritance are directly supported in database schemas and in the query language.
- You can create customized data types & methods
- It provides a middle ground between relational databases and *object-oriented databases*

Relational Database Terms

- **Entity:** an object about which you want to store data
- **Relationships:** links that show how different records are related
- **Key Fields:** establish relationships among records in different tables
- Five main types of key fields:
- - Unique
  - primary keys
  - candidate keys
  - foreign keys
  - composite keys

Primary Keys

- **Primary key**
  - Value must be unique for each record
  - Serves to identify the record
  - Present in every record
  - Can't be NULL
  - Should be numeric

Candidate Keys

- **Candidate key**
  - Any field that could be used as the primary key
  - Should be a unique, unchanging numeric field

Foreign Keys

- **Foreign key:** a field in a table that is a primary key in another table
- Foreign k
- Foreign k key

STUDENT

S_ID	S_LAST	S_FIRST	S_MI	S_ADDRESS	S_STATE	S_ZIP	F_ID
100	Miller	Sarah	M	144 Windridge Blvd.	WI	54703	1
101	Umato	Brian	D	454 St. John's Place	WI	54702	1
102	Black	Daniel		8921 Circle Drive	WI	54715	1
103	Mobley	Amanda	J	1716 Summit St.	WI	54703	2
104	Sanchez	Ruben	R	1780 Samantha Court	WI	54701	4
105	Connoly	Michael	S	1818 Silver Street	WI	54712	3

Relationship

Foreign keys

FACULTY

F_ID	F_LAST	F_FIRST	F_MI	F_PHONE	F_RANK
1	Cox	Kim	J	7155551234	ASSO
2	Blanchard	John	R	7155559087	FULL
3	Williams	Jerry	F	7155555412	ASST
4	Sheng	Laura	M	7155556409	INST
5	Brown	Phillip	E	7155556082	ASSO

Primary keys

Figure 1-9 Creating relationships using foreign keys

Composite Keys

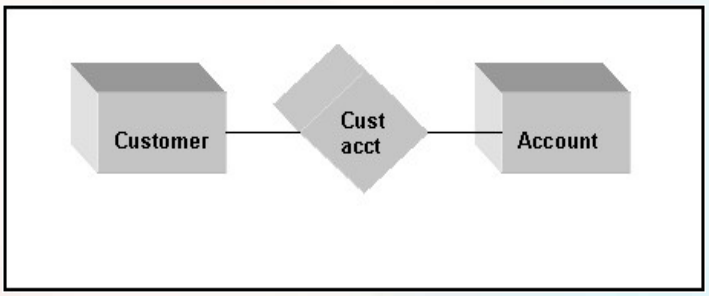
- **Composite key:** a unique key that you create by combining two or more fields
- Usually comprised of fields that are primary keys in other tables

Relationships & Relationship Sets

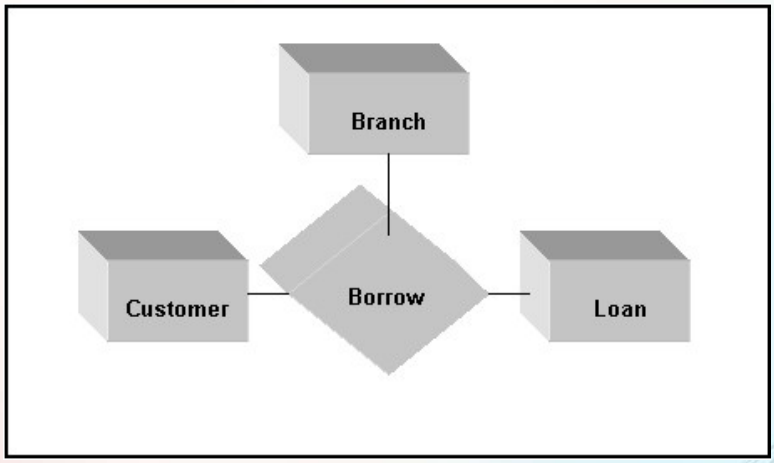
- A **relationship** is an association among two or more entities.
- A **relationship set** is a collection of relationships of the same type.

Binary Relationship

- Relationship that involves two entities is called binary.



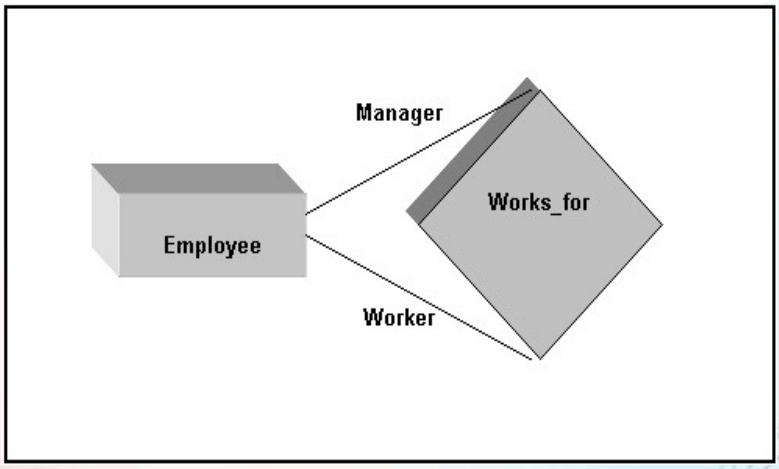
Relationships set may involve more than two entity sets



Roles in Entity

- The role of an entity is the function it plays in a relationship.
- For example, the relationship works-for could be ordered pairs of employee entities. The first employee takes the role of manager, and the second one takes the role of worker as shown in figure on the next slide.

Employee Entities Interact via the relationship "Works_for"

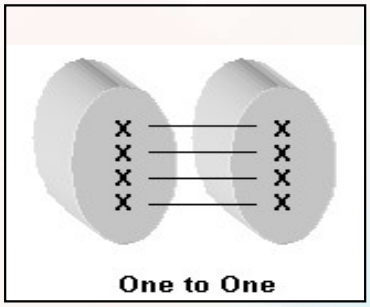


Mapping Cardinalities

- Mapping Cardinalities expresses the number of entities to which another entity can be associated via a relationship.
- For binary relationship between entity sets X and Y, the mapping cardinality must be one of:
  - One-to-One
  - One-to-Many
  - Many-to-One
  - Many-to-Many

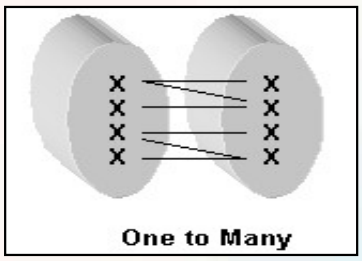
One-to-One

- An entity in X is associated with at most one entity in Y, and an entity in Y is a associated with at most one entity in X.



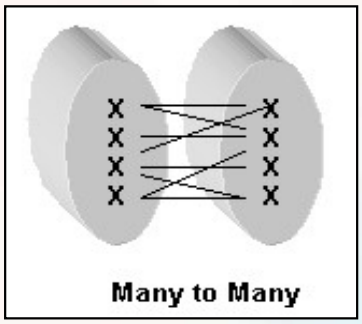
One-to-Many

- An entity in X is associated with any number in Y. An entity in Y is associated with at most one entity in X.

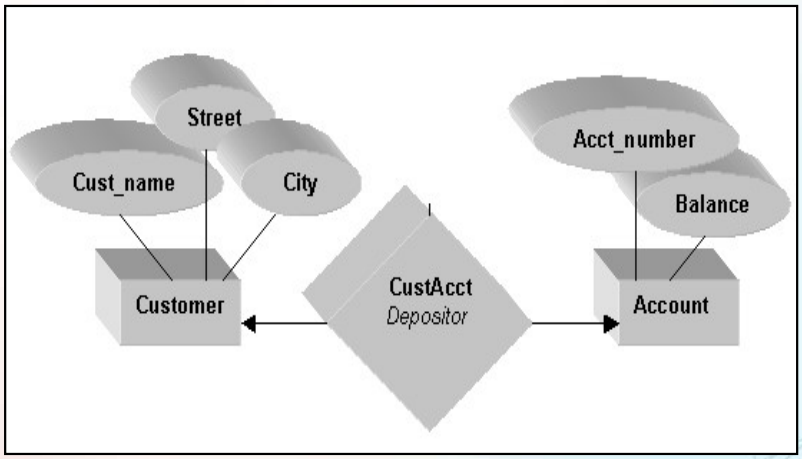


Many-to-Many

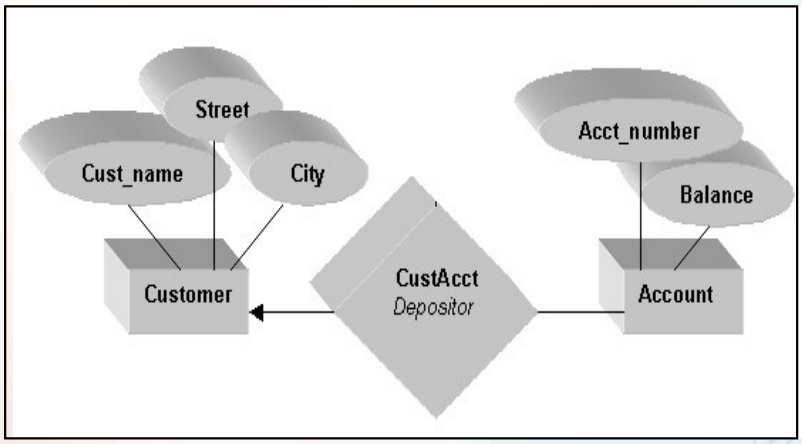
- Entities in X and Y are associated with any number from each other



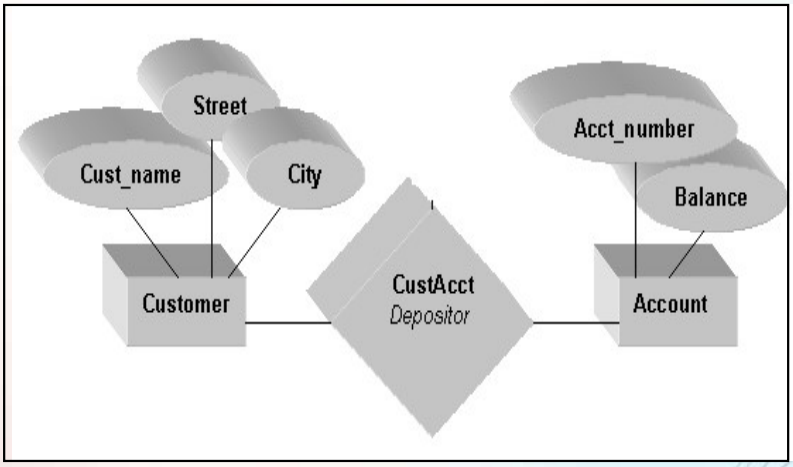
One to Many (1:N)



One to One (1:1)



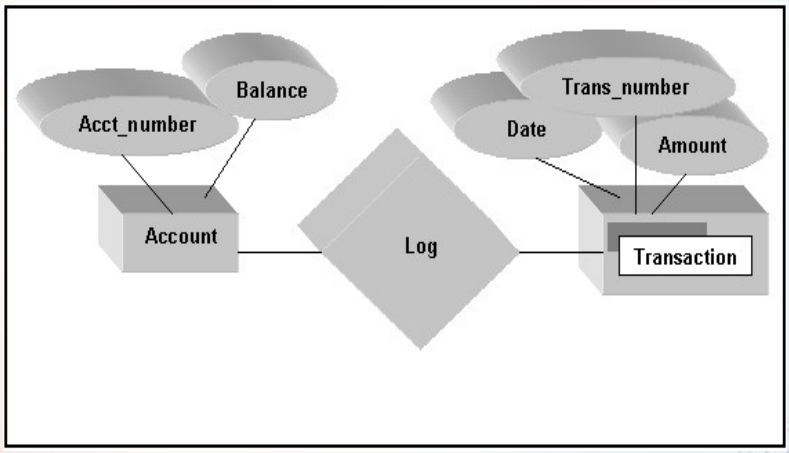
Many to Many (N:N)



Keys

- An entity that depends does not have sufficient attributes & depends on some other entity is called a weak entity. One that does have a primary key is called a strong entity set.
- The primary key of a weak entity is found by taking the primary key of the strong entity on which it is existence-dependent, plus the discriminator of the weak entity set.

E-R diagram with a weak entity set



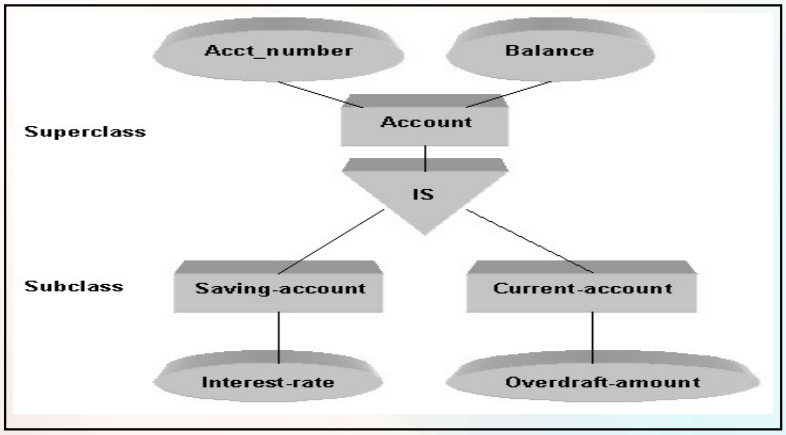
Specialization and Generalization

- An entity set may contain subgroupings of entities that are distinct in some way of other entities.
- An entity type E1 is a specialization of another entity type E2 if E1 has the same properties of E2 and perhaps even more.

Specialization and Generalization…

- In E-R diagrams, specialization or generalization is shown by a triangle,
  - Specialization = top-down design.
  - Generalization = bottom-up design.
- A lower-level entity set (subclass) inherits all the attributes and relationships participation of the higher-level entity set to which it is linked (superclass).

Generalization



Composite Attributes

- Composite attributes can be divided into smaller subparts, which represent more basic attributes with independent meanings of their own.
- For example, the Address attribute of the employee entity can be subdivided into Street, City, State and Zip.

Multivalue Attribute

- A multivalued attribute may have upper and lower and upper bounds on the number of values for an individual entity.
- For example, the colors attribute of a car may have between one and have five values, if we assume that a car can have at most five colors.

Derived Attribute

- For example, the Age and Birthdate attributes of a person.
- For a particular person entity, the value of age can be determined from the current date and the value of the person's Birthdate.
- The Age attribute is hence called a derived attribute and is said to be derivable from the Birthdate attribute, which is called a stored attribute.

Summary of ER Diagram Notation

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE
	COMPOSITE ATTRIBUTE
	DERVIED ATTRIBUTE

The Clearwater Traders Sales Order Database

Clearwater Traders

- Markets a line of clothing and sporting goods via mail-order catalogs
- Accepts customer orders via telephone, mail, and fax
- Wants to begin accepting orders using its Web site
- Has decided to offer 24-hour customer order service
- Existing microcomputer-based database system cannot handle current transaction volume
- Managers concerned that the current database does not have the failure-handling and recovery capabilities needed for an ordering system that cannot tolerate failures or downtime

Clearwater Traders Data Requirements

- Customer name, address, daytime and evening telephone numbers, user names, and passwords
- Order date, payment method (check or credit card), order source (catalog description or Web site), and associated item numbers, sizes, colors, and quantities ordered
- Item descriptions and photo images, as well as item categories (women's clothing, outdoor gear, and so on), prices, and quantities on hand. Many clothing items are available in multiple sizes and colors. Sometimes the same item has different prices depending on the item size
- Information about incoming product shipments

Clearwater Traders Table Rela

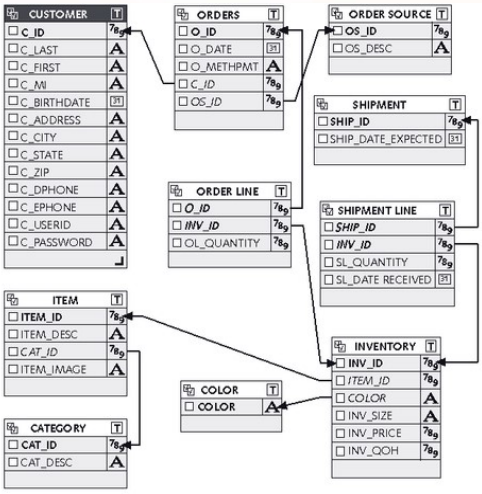


Figure 1-15 Visual representation of the Clearwater Traders database

The Northwoods University Student Registration Database

Northwoods University

- Decided to replace its aging mainframe-based student registration system with a more modern client/server database system
- School officials want students to be able to retrieve course availability information, register for courses, and print transcripts using personal computers located in the student computer labs

The Northwoods University Student Registration Database (cont)

- Faculty members must be able to retrieve student course lists, drop and add students, and record course grades
- Faculty members must also be able to view records for the students they advise
- Security is a prime concern, so student and course records must be protected by password access

Northwoods University Data Requirements

- Student name, address, telephone number, class (freshman, sophomore, junior, or senior), date of birth, PIN (personal identification number), and advisor ID
- Course call number (such as MIS 101), course name, credits, location, duration, maximum enrollment, instructor, and term offered
- Instructor name, office location, telephone number, rank, and PIN
- Student enrollment and grade information

Northwoods University Table Rela

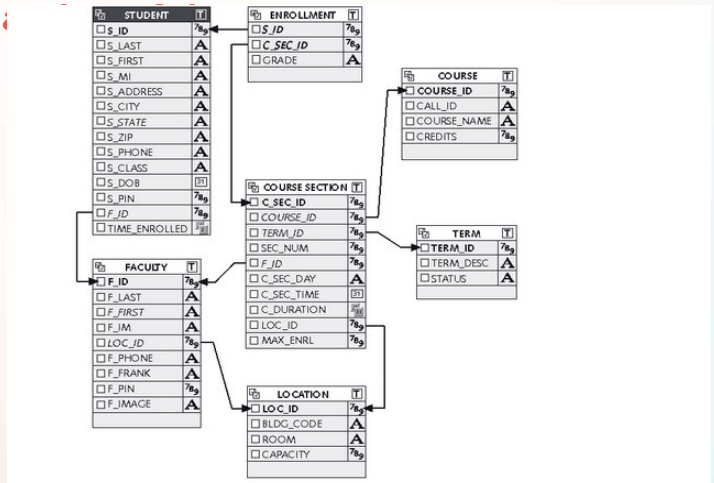
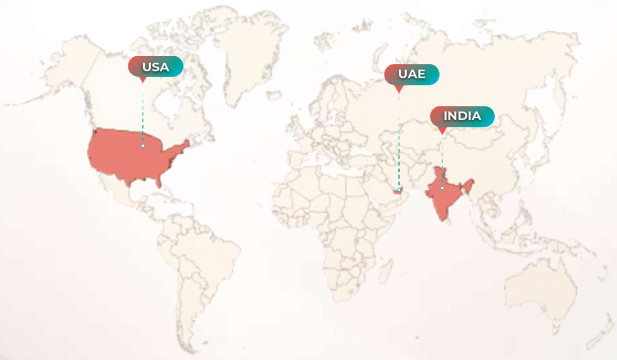


Figure 1-17 Visual representation of the Northwoods University database

Physical Presence



www.vinsys.com

enquiry@vinsys.com

Shivaji Niketan, Tejas Society, Behind Kothrud Bus Stand, Near Mantri Park, Kothrud, Pune - 411029.

304, City Tower 2, Near Crown Plaza, Sheikh Zayed Road. Dubai, UAE.
PO.Box - 213279

132 West 31st Street, First Floor, New York, 10001, USA