System Design

5.1 Database Basics

Database

- A database is a collection of related data
- The collected data could be in any number of formats (electronic, printed, graphic, audio, statistical, combinations).
- There are physical (paper/print) and electronic databases.
- **Examples:**
    - phone book
    - address book
    - Census Bureau data

Database Management System

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database.

examples: MySQL, Microsoft Access, SQL Server, Oracle, dBASE.

Database System:

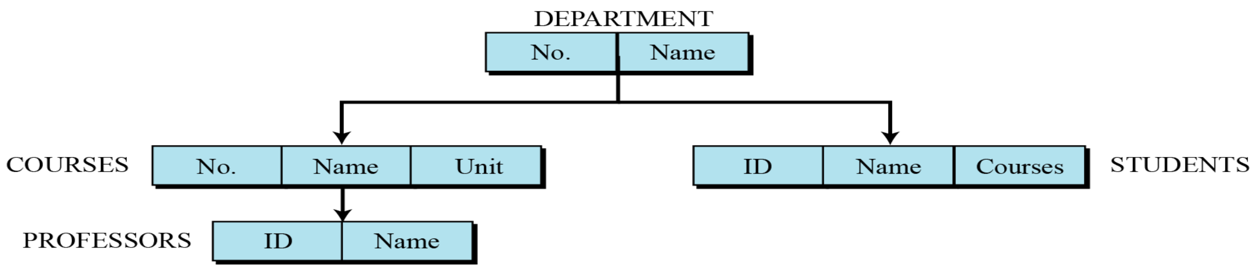
Database and DBMS software together

Database Models

- A database model defines the logical design of data.
- In the history of database design, three models have been in use: the hierarchical model, the network model and the relational model.

Hierarchical database model

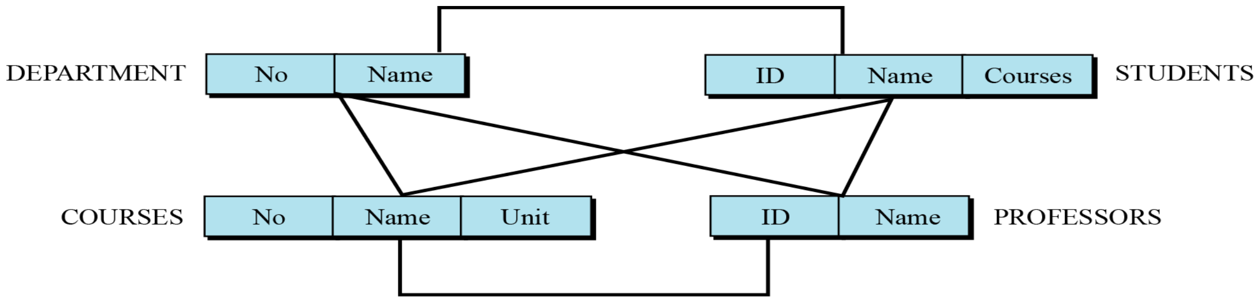
In the hierarchical model, data is organized as an inverted tree. Each entity has only one parent but can have several children. At the top of the hierarchy, there is one entity, which is called the root.



An example of the hierarchical model representing a university

Network database model

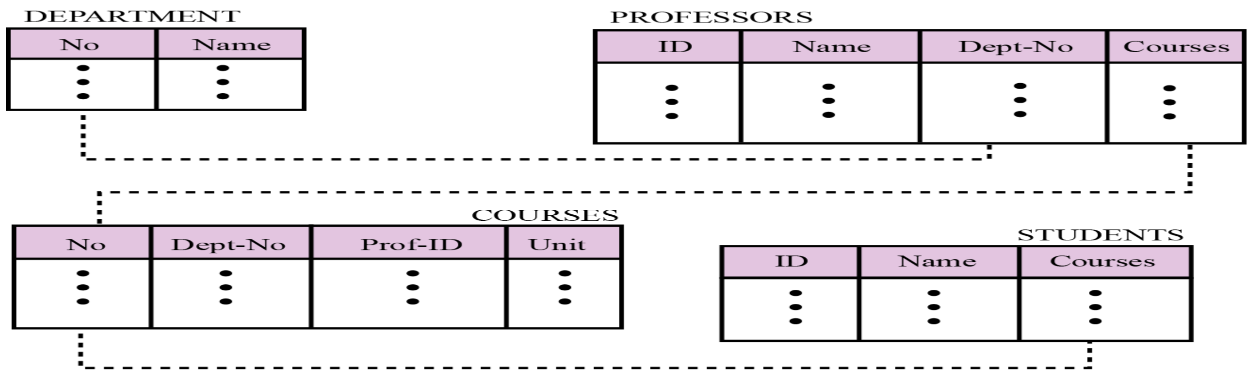
In the network model, the entities are organized in a graph, in which some entities can be accessed through several paths.



An example of the network model representing a university

Relational database model

In the relational model, data is organized in two-dimensional tables called relations. The tables or relations are related to each other.



An example of the relational model representing a university

Relational Database Model

- In the relational database management system (RDBMS), the data is represented as a set of *relations*.
- **Relational Database**: data represented as a set of related tables (or relations)
- **Relation**: a named, two-dimensional table of data. Each relation consists of a set of named columns and an arbitrary number of unnamed rows
- **Well-Structured Relation**: a relation that contains a minimum amount of redundancy and allows users to insert, modify, and delete the rows without errors or inconsistencies

Properties of a Relation

- Entries in cells are simple.
- Entries in columns are from the same set of values.
- Each row is unique.
- The sequence of columns can be interchanged without changing the meaning or use of the relation.
- The rows may be interchanged or stored in any sequence.

Keys

- Primary Key
  - An attribute whose value is unique across all occurrences of a relation.
- A primary key may involve a single attribute or be composed of multiple attributes.
- Foreign Key
  - An attribute that appears as a non primary key attribute in one relation and as a primary key attribute (or part of a primary key) in another relation
- Referential Integrity
  - An integrity constraint specifying that the value of an attribute in one relation depends on the value of the same attribute in another relation

5.2 Designing Database

5.2.1 Normalization

- Database Normalization is a technique of organizing the data in the database.
- Normalization is a systematic approach of decomposing tables to eliminate data redundancy and undesirable characteristic like insertion, Update and Deletion Anomalies.
- Normalization is used for mainly two purposes
    - Eliminating redundant(useless) data
    - Ensuring data dependencies make sense

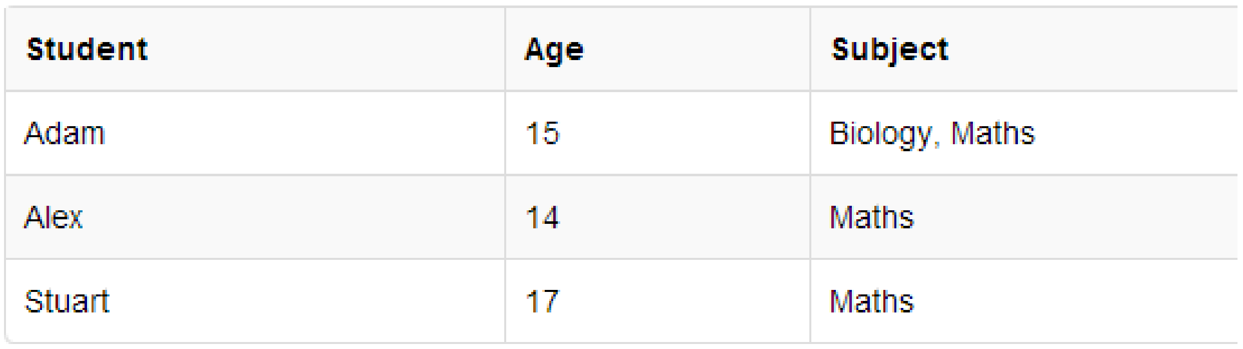
5.2.1.1 First Normal From (1NF)

A table (relation) is in 1NF if

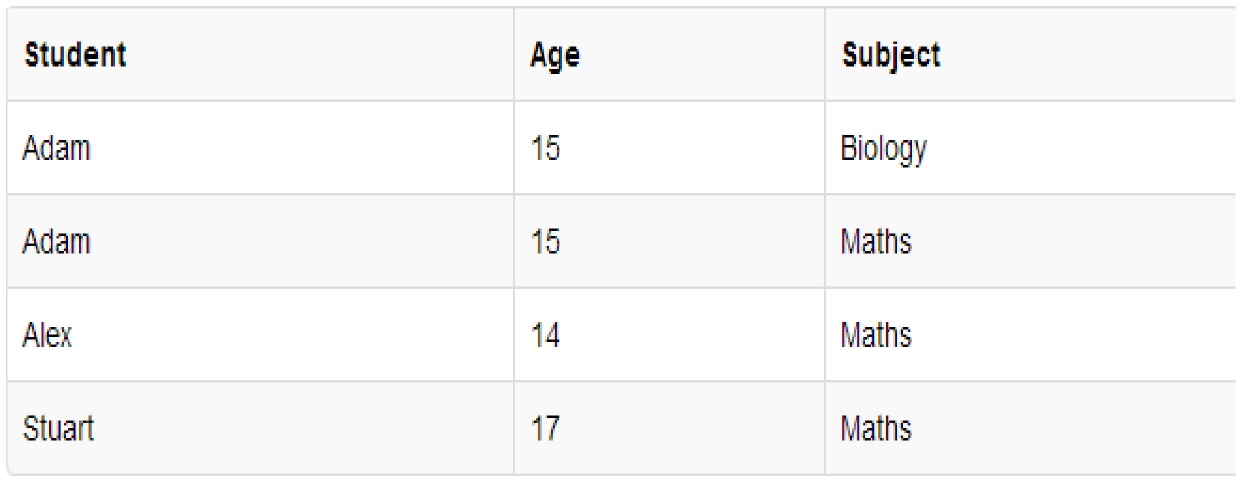
1. There are no duplicated rows in the table.

2. Each cell is single-valued.

3. Entries in a column (attribute, field) are of the same kind.



Student	Age	Subject
Adam	15	Biology, Maths
Alex	14	Maths
Stuart	17	Maths



Student	Age	Subject
Adam	15	Biology
Adam	15	Maths
Alex	14	Maths
Stuart	17	Maths

5.2.1.2 Second Normal Form (2NF)

- A table is in 2NF if it is in 1NF and if all non-key attributes are dependent on all of the key.
- A table is in 2NF if it is in 1NF and if it has no partial dependencies.

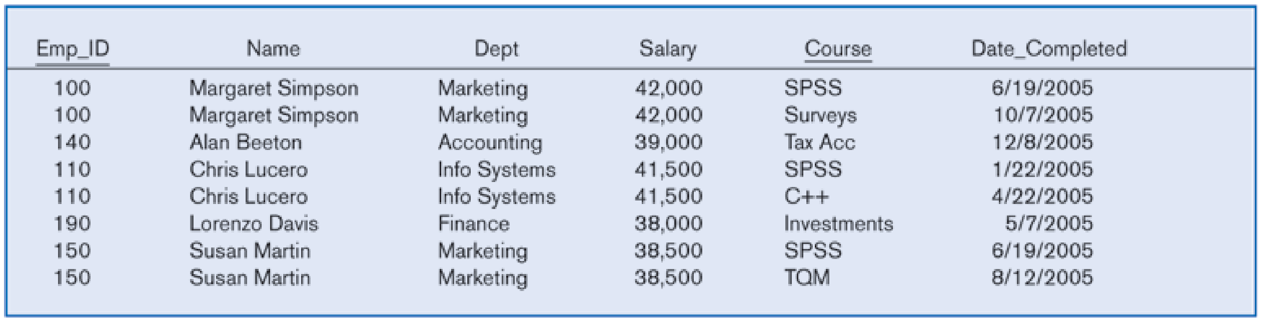
Functional Dependencies and Determinants

- Functional Dependency
  - For a given relation, attribute B is functionally dependent on attribute A is, for every valid value of A, that value of A uniquely determines the value of B
- Determinant: an attribute that determines the values of other attributes
  - All primary keys are determinants
- Symbolic notation:
  - $A \rightarrow B$
  - A is the determinant
  - B is functionally dependent on A
- Functional dependency is not a mathematical dependency.
- Instances (or sample data) in a relation do not prove the existence of a functional dependency.
- Knowledge of problem domain is most reliable method for identifying functional dependency.

1NF but not 2NF

Figure 10-6 Relation with redundancy

EMPLOYEE2



Emp_ID	Name	Dept	Salary	Course	Date_Completed
100	Margaret Simpson	Marketing	42,000	SPSS	6/19/2005
100	Margaret Simpson	Marketing	42,000	Surveys	10/7/2005
140	Alan Beeton	Accounting	39,000	Tax Acc	12/8/2005
110	Chris Lucero	Info Systems	41,500	SPSS	1/22/2005
110	Chris Lucero	Info Systems	41,500	C++	4/22/2005
190	Lorenzo Davis	Finance	38,000	Investments	5/7/2005
150	Susan Martin	Marketing	38,500	SPSS	6/19/2005
150	Susan Martin	Marketing	38,500	TQM	8/12/2005

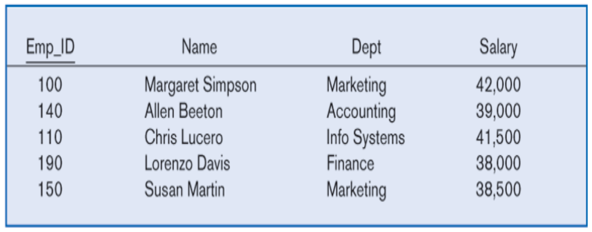
EMPLOYEE2(Emp_ID, Name, Dept, Salary, Course, Date_Completed)

Functional dependencies:

1. Emp_ID → Name, Dept, Salary ⬅ partial key dependency
2. Emp_ID, Course → Date_Completed

Figure 10-5 EMPLOYEE1 relation with sample data

EMPLOYEE1



Emp_ID	Name	Dept	Salary
100	Margaret Simpson	Marketing	42,000
140	Allen Beeton	Accounting	39,000
110	Chris Lucero	Info Systems	41,500
190	Lorenzo Davis	Finance	38,000
150	Susan Martin	Marketing	38,500

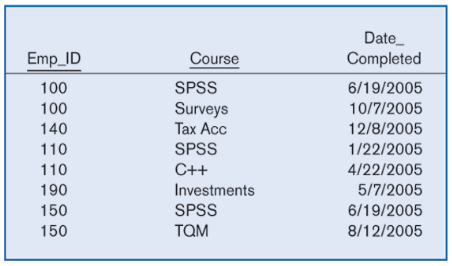
EMPLOYEE1(Emp_ID, Name, Dept, Salary)

Functional dependencies:

- Emp_ID → Customer_Name, SalesPerson

Figure 10-7 EMP COURSE relation

EMP COURSE



Emp_ID	Course	Date_ Completed
100	SPSS	6/19/2005
100	Surveys	10/7/2005
140	Tax Acc	12/8/2005
110	SPSS	1/22/2005
110	C++	4/22/2005
190	Investments	5/7/2005
150	SPSS	6/19/2005
150	TQM	8/12/2005

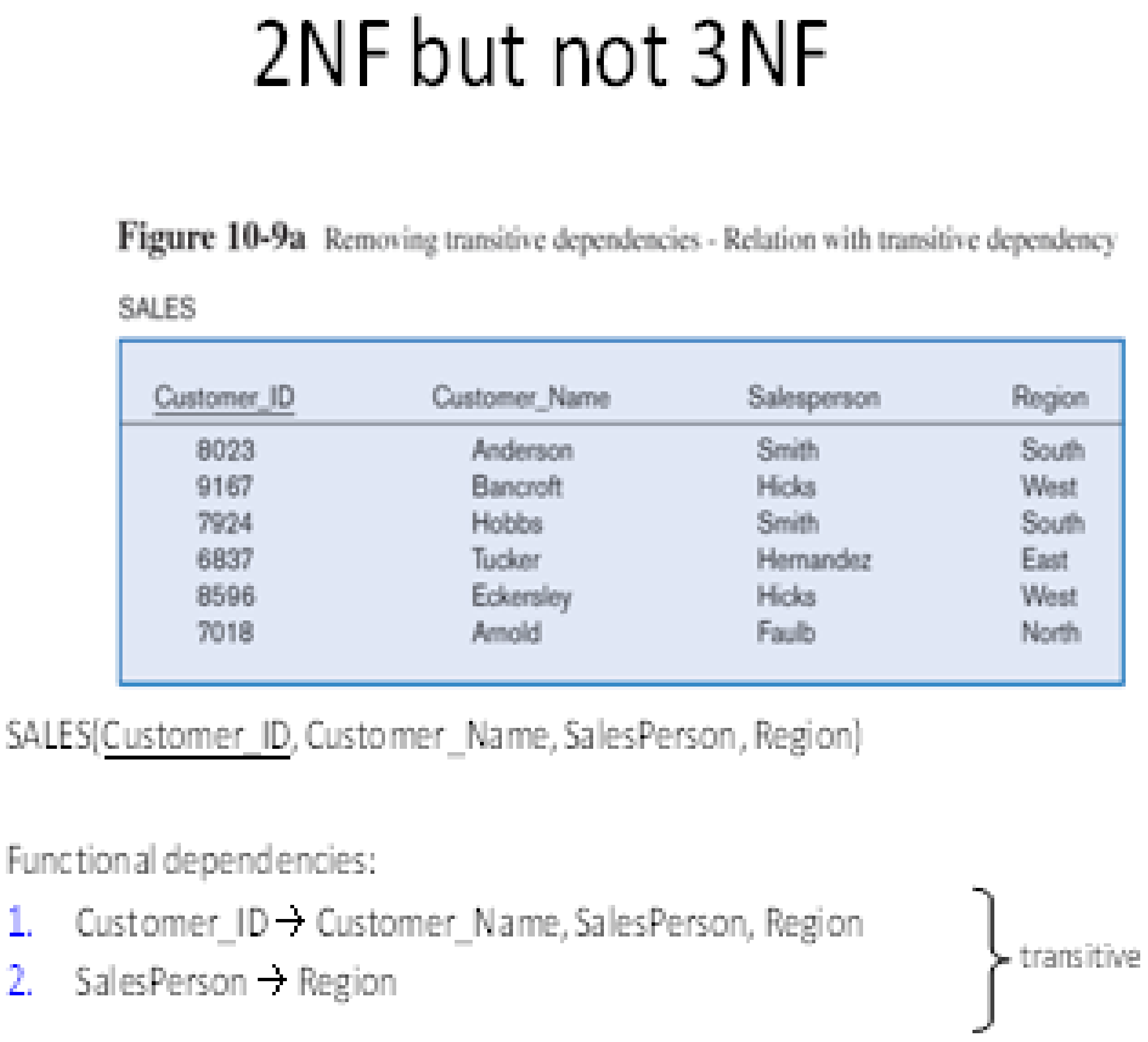
EMPCOURSE(Emp_ID, Course, Date_Completed)

Functional dependency:

- Emp_ID, Course → Date_Completed

5.2.1.3 Third Normal Form (3NF)

- A table is in 3NF if it is in 2NF and if it has no transitive dependencies.
- A relation is in third normal form (3NF) if it is in second normal form (2NF) and there are no functional (transitive) dependencies between two (or more) non primary key attributes.



2NF but not 3NF

Figure 10-9a Removing transitive dependencies - Relation with transitive dependency

SALES

Customer_ID	Customer_Name	Salesperson	Region
8023	Anderson	Smith	South
9167	Bancroft	Hicks	West
7924	Hobbs	Smith	South
6837	Tucker	Hernandez	East
8596	Eckersley	Hicks	West
7018	Arnold	Faulb	North

SALES[Customer_ID, Customer_Name, SalesPerson, Region]

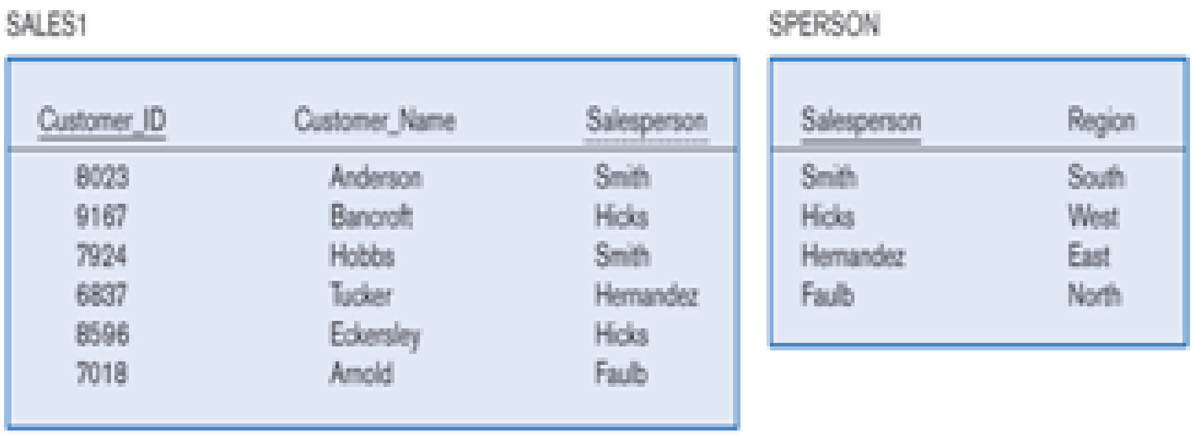
Functional dependencies:

1. Customer_ID → Customer_Name, SalesPerson, Region
2. SalesPerson → Region

transitive

Converted to 3NF

Figure 10-9b Removing transitive dependencies - Relations in 3NF



SALES1

Customer_ID	Customer_Name	Salesperson
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

SPERSON

Salesperson	Region
Smith	South
Hicks	West
Hernandez	East
Faulb	North

SALES1(<u>Customer_ID</u>, Customer_Name, SalesPerson)

Functional dependencies:

- Customer_ID → Customer_Name, SalesPerson

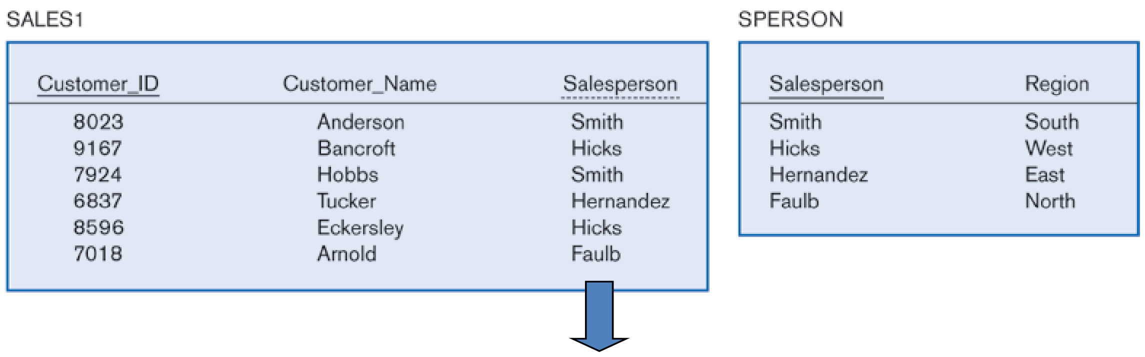
SPERSON(<u>SalesPerson</u>, Region)

Functional dependency:

- SalesPerson → Region

Foreign Key Example

Figure 10-9b Removing transitive dependencies - Relations in 3NF



SALES1

Customer_ID	Customer_Name	Salesperson
8023	Anderson	Smith
9167	Bancroft	Hicks
7924	Hobbs	Smith
6837	Tucker	Hernandez
8596	Eckersley	Hicks
7018	Arnold	Faulb

SPERSON

Salesperson	Region
Smith	South
Hicks	West
Hernandez	East
Faulb	North

The foreign key

The foreign key establishes a one-to-many relationship between SPERSON (one) and SALES1 (many)

There can be no SalesPerson value in SALES1 that does not exist in SPERSON (referential integrity)

*EmployeeID	LastName	FirstName	*ProjectNumber	ProjectTitle
EN1-26	O'Brien	Sean	30-452-T3	STAR manual
EN1-26	O'Brien	Sean	30-457-T3	ISO procedures
EN1-26	O'Brien	Sean	31-124-T3	Employee handbook
EN1-33	Guya	Amy	30-452-T3	STAR manual
EN1-33	Guya	Amy	30-482-TC	Web Site
EN1-33	Guya	Amy	31-241-TC	New catalog
EN1-35	Baranco	Steven	30-452-T3	STAR manual
EN1-35	Baranco	Steven	31-238-TC	STAR prototype
EN1-36	Roslyn	Elizabeth	35-152-TC	STAR pricing
EN1-38	Schaaf	Carol	36-272-TC	Order system
EN1-40	Wing	Alexandra	31-238-TC	STAR prototype
EN1-40	Wing	Alexandra	31-241-TC	New catalog

*ProjectNum	ProjectTitle
30-452-T3	STAR manual
30-457-T3	ISO procedures
30-482-TC	Web site
31-124-T3	Employee handbook
31-238-TC	STAR prototype
31-238-TC	New catalog
35-152-TC	STAR pricing
36-272-TC	Order system

*EmployeeID	*ProjectNum
EN1-26	30-452-T3
EN1-26	30-457-T3
EN1-26	31-124-T3
EN1-33	30-328-TC
EN1-33	30-452-T3
EN1-33	32-244-T3
EN1-35	30-452-T3
EN1-35	31-238-TC
EN1-36	35-152-TC
EN1-38	36-272-TC
EN1-40	31-238-TC
EN1-40	31-241-TC

*EmployeeID	Last Name	First Name
EN1-26	O'Brien	Sean
EN1-33	Guya	Amy
EN1-35	Baranco	Steven
EN1-36	Roslyn	Elizabeth
EN1-38	Schaaf	Carol
EN1-40	Wing	Alexandra

*ProjectNum	ProjectTitle	ProjectMgr	Phone
30-452-T3	STAR manual	Garrison	2756
30-457-T3	ISO procedures	Jacanda	2954
30-482-TC	Web site	Friedman	2846
31-124-T3	Employee handbook	Jones	3102
31-238-TC	STAR prototype	Garrison	2756
31-241-TC	New catalog	Jones	3102
35-152-TC	STAR pricing	Vance	3022
36-272-TC	Order system	Jacanda	2954

*ProjectNum	ProjectTitle	ProjectMgr
30-452-T3	STAR manual	Garrison
30-457-T3	ISO procedures	Jacanda
30-482-TC	Web site	Friedman
31-124-T3	Employee handbook	Jones
31-238-TC	STAR prototype	Garrison
31-241-TC	New catalog	Jones
35-152-TC	STAR pricing	Vance
36-272-TC	Order system	Jacanda

*ProjectMgr	Phone
Friedman	2846
Garrison	2756
Jacanda	2954
Jones	3102
Vance	3022

5.2.2 Transforming E-R Diagrams into Relations

- Steps
  - Represent entities
  - Represent relationships
  - Normalize the relations
  - Merge the relations

Representing Entities

- Each regular entity is transformed into a relation.
- The identifier of the entity type becomes the primary key of the corresponding relation.
- The primary key must satisfy the following two conditions.
  a. The value of the key must uniquely identify every row in the relation.
  b. The key should be non redundant.

Figure 10-10a Transforming an entity type to a relation - E-R Diagram

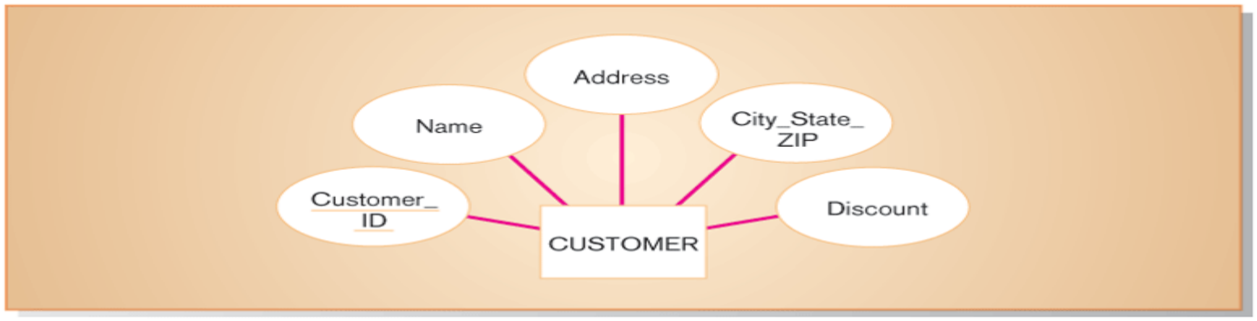
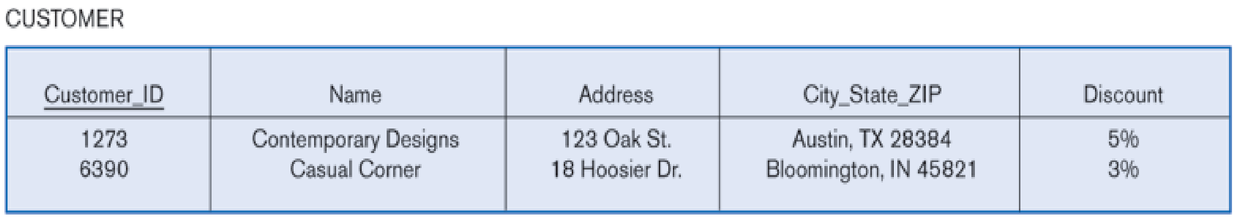


Figure 10-10b Transforming an entity type to a relation - Relation



CUSTOMER

Customer_ID	Name	Address	City_State_ZIP	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

Represent Relationships

- Binary 1:N Relationships
  - Add the primary key attribute (or attributes) of the entity on the one side of the relationship as a foreign key in the relation on the right side.
- Binary or Unary 1:1
  - Three possible options
    a. Add the primary key of A as a foreign key of B.
    b. Add the primary key of B as a foreign key of A.
    c. Both of the above.

Figure 10-11a Representing a 1:N relationship - E-R Diagram

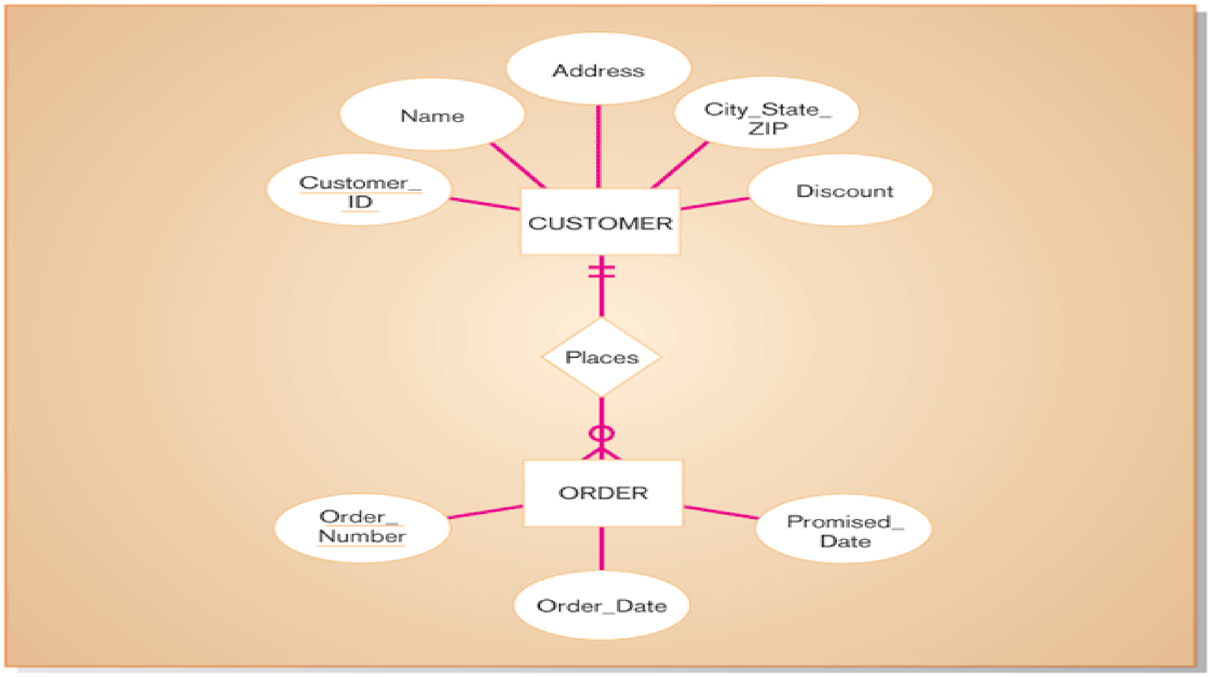
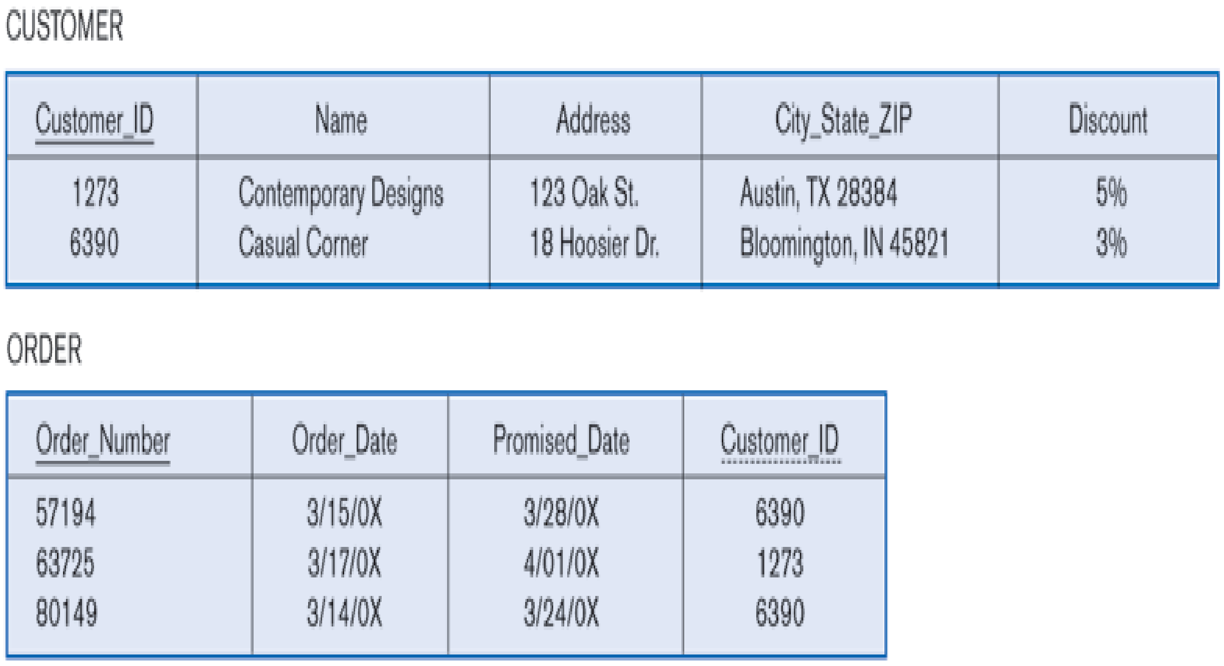


Figure 10-11b Representing a 1:N relationship - Relations



CUSTOMER

Customer_ID	Name	Address	City_State_ZIP	Discount
1273	Contemporary Designs	123 Oak St.	Austin, TX 28384	5%
6390	Casual Corner	18 Hoosier Dr.	Bloomington, IN 45821	3%

ORDER

Order_Number	Order_Date	Promised_Date	Customer_ID
57194	3/15/0X	3/28/0X	6390
63725	3/17/0X	4/01/0X	1273
80149	3/14/0X	3/24/0X	6390

- Binary and Higher M:N relationships
  - Create another relation and include primary keys of all relations as primary key of new relation.

Figure 10-12a Representing an M:N relationship - E-R Diagram

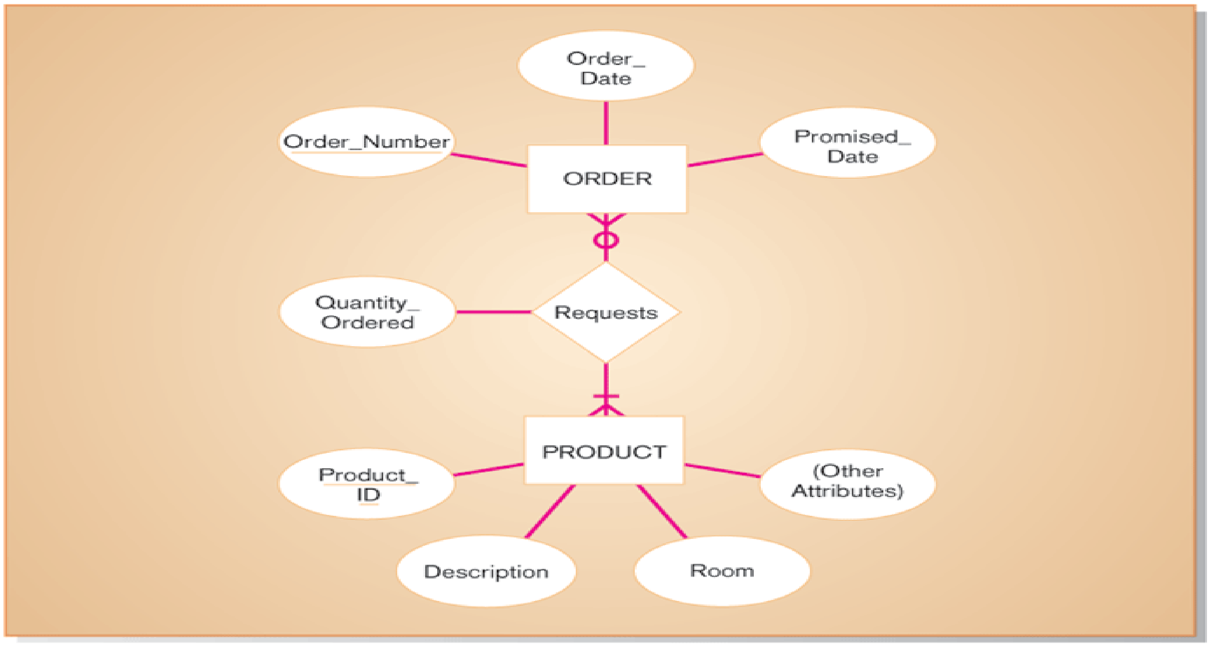


Figure 10-12b Representing an M:N relationship - Relations

ORDER

Order_Number	Order_Date	Promised_Date
61384	2/17/2005	3/01/2005
62009	2/13/2005	2/27/2005
62807	2/15/2005	3/01/2005

ORDER LINE

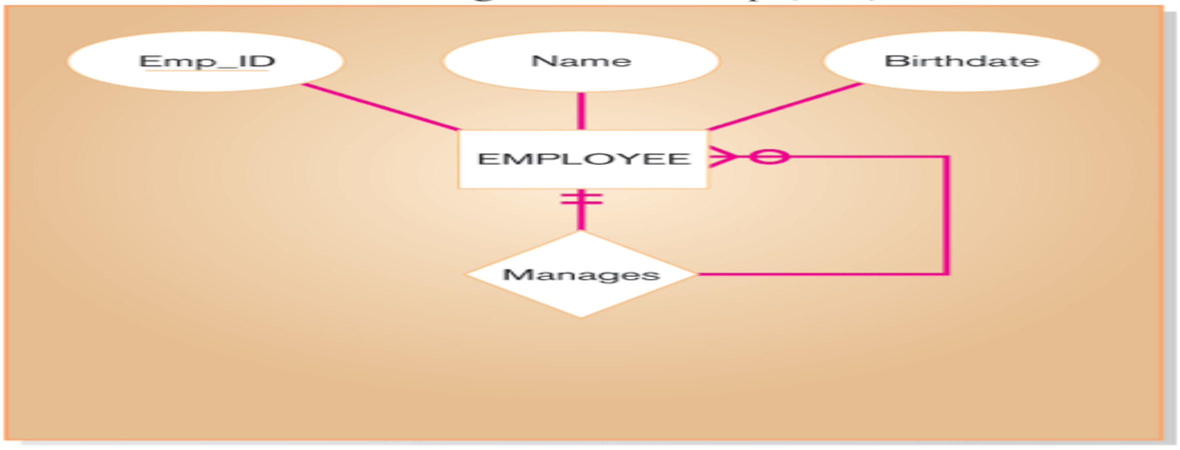
Order_Number	Product_ID	Quantity_ Ordered
61384	M128	2
61384	A261	1

PRODUCT

Product_ID	Description	Room	(Other Attributes)
M128	Bookcase	Study	–
A261	Wall unit	Family	–
R149	Cabinet	Study	–

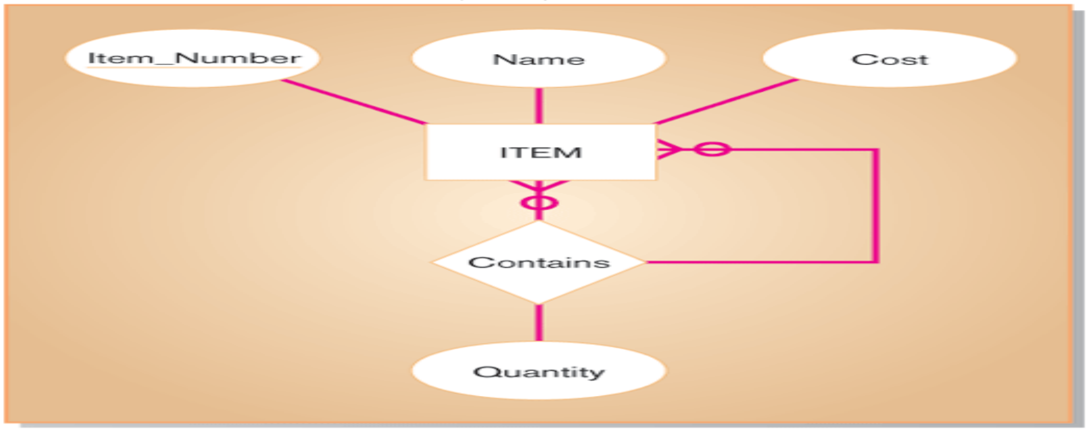
- Unary 1:N Relationships
  - Utilize a recursive foreign key
    - A foreign key in a relation that references the primary key values of that same relation.
- Unary M:N Relationships
  - Create a separate relation.
  - Primary key of new relation is a composite of two attributes that both take their values from the same primary key.

Figure 10-13a Two unary relations – EMPLOYEE with Manages relationship (1:N)



EMPLOYEE(Emp_ID, Name, Birthdate, Manager_ID)

Figure 10-13b Two unary relations – Bill-of-materials structure (M:N)



ITEM(Item_Number, Name, Cost)

ITEMCOMPONENT(Item_Number, Component_Number, Quatity)

Merging Relations (View Integration)

- **Purpose is to remove redundant relations**
- View Integration Problems
    - Synonyms
        - Two different names used for the same attribute
        - **When merging, get agreement from users on a single, standard name**
    - Homonyms
        - A single attribute name that is used for two or more different attributes
        - **Resolved by creating a new name**
    - Dependencies between nonkeys
        - Dependencies may be created as a result of view integration
        - **In order to resolve, the new relation must be normalized**

5.3 Designing Forms and Reports

5.3.1 Forms vs. Reports

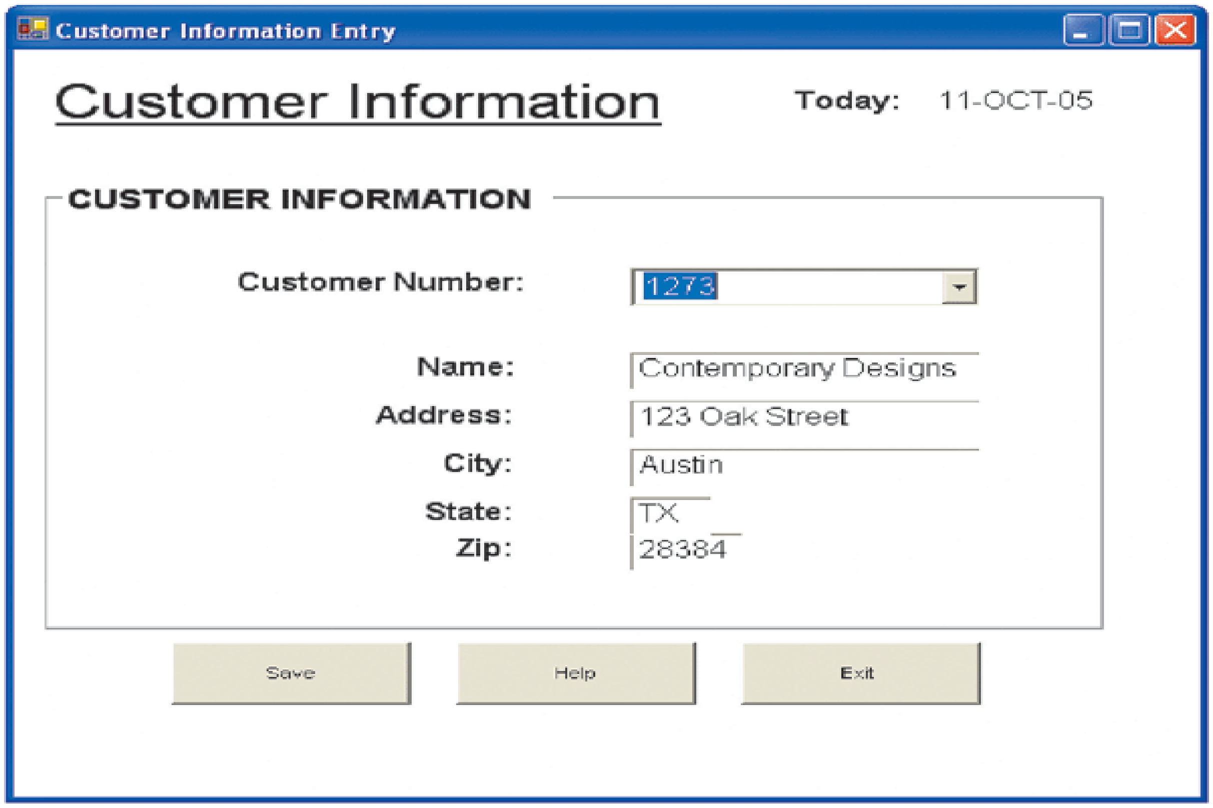
- Form
    - A business document that contains some predefined data and may include some areas where additional data are to be filled in.
    - An instance of a form is typically based on one database record.
- Report
    - A business document that contains only predefined data.
    - A passive document for reading or viewing data.
    - Typically contains data from many database records or transactions.

5.3.2 The Process of Designing Forms and Reports

- User-focused activity
- Follows a prototyping approach
- Requirements determination:
    - Who will use the form or report?
    - What is the purpose of the form or report?
    - When is the report needed or used?
    - Where does the form or report need to be delivered and used?

  - How many people need to use or view the form or report?
- Prototyping
  - Initial prototype is designed from requirements
  - Users review prototype design and either accept the design or request changes
  - If changes are requested, the construction-evaluation-refinement cycle is repeated until the design is accepted

Figure 11-3
A data input screen designed in Microsoft's Visual Basic .NET


Customer Information Entry

Customer Information

Today: 11-OCT-05

CUSTOMER INFORMATION

Customer Number:	1273
Name:	Contemporary Designs
Address:	123 Oak Street
City:	Austin
State:	TX
Zip:	28384

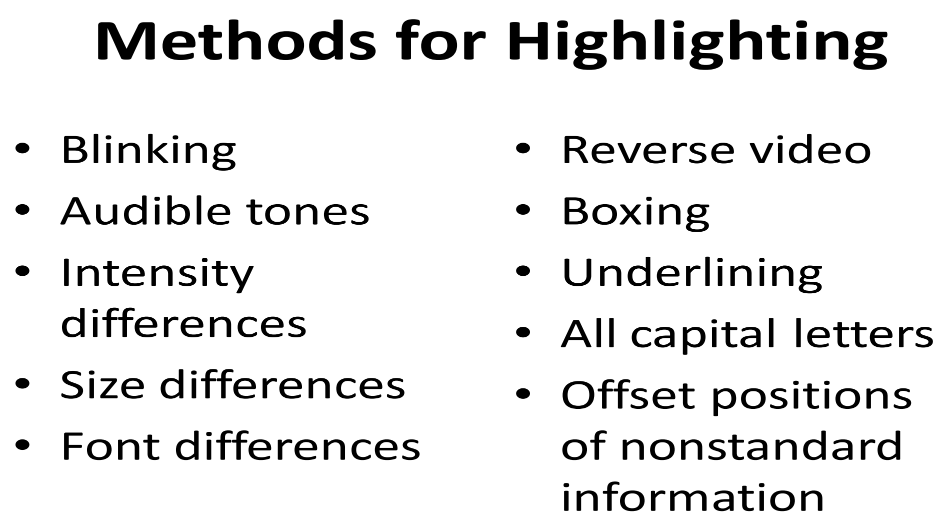
Guidelines for Form and Report Design

- Meaningful titles: clear, specific, version information, current date
- Meaningful information– include only necessary information, with no need to modify
- Balanced layout: adequate spacing, margins, and clear labels
- Easy navigation system: show how to move forward and backward, and where you are currently

Uses of Highlighting in Forms and Reports

- Notify users of errors in data entry or processing.

- Provide warnings regarding possible problems.
- Draw attention to keywords, commands, high-priority messages, unusual data values.



Methods for Highlighting

- Blinking
- Audible tones
- Intensity differences
- Size differences
- Font differences
- Reverse video
- Boxing
- Underlining
- All capital letters
- Offset positions of nonstandard information

Color vs. No Color

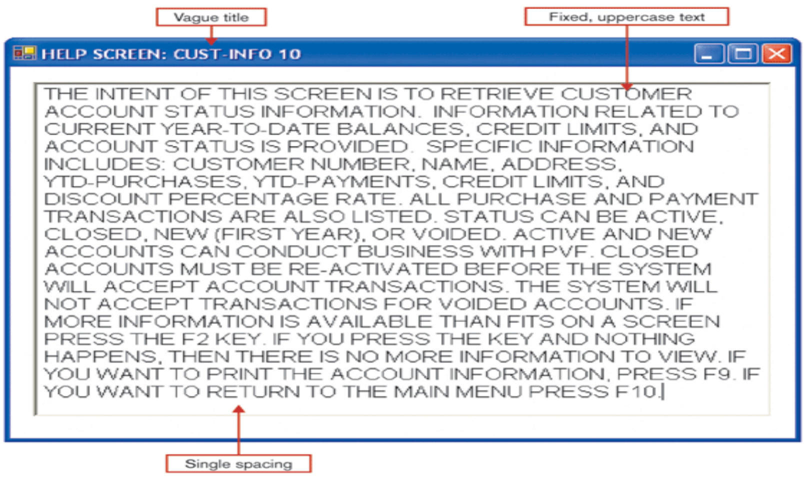
- **Benefits from Using Color**
  - Strikes the eye
  - Facilitates precise as to be difficult to analyze in complex displays
  - Emphasizes the logical organization of information
  - Draws attention to warnings
- **Problems from Using Color**
  - Color pairings may wash out or cause problems for some users
  - Resolution may degrade with different displays
  - Color fidelity may degrade on different displays
  - Printing or conversion to other media may not easily translate

Guidelines for Displaying Text

- Case: mixed upper and lower case, use conventional punctuation

- Spacing: double spacing if possible, otherwise blank lines between paragraphs
- Justification: left justify text, ragged right margins
- Abbreviations: only when widely understood and significantly shorter than full text

Figure 11-7a Contrasting the display of textual help information - Poorly designed help screen with many violations of the general guidelines for displaying text

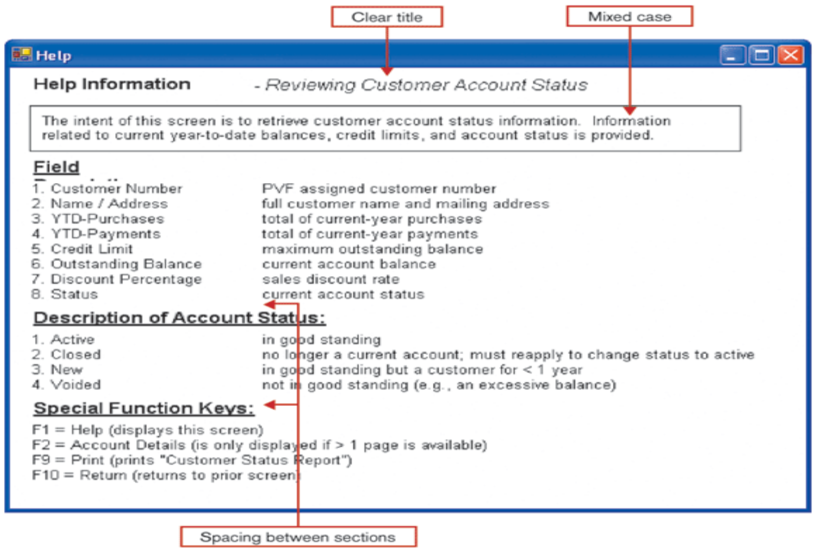


HELP SCREEN: CUST-INFO 10

THE INTENT OF THIS SCREEN IS TO RETRIEVE CUSTOMER ACCOUNT STATUS INFORMATION. INFORMATION RELATED TO CURRENT YEAR-TO-DATE BALANCES, CREDIT LIMITS, AND ACCOUNT STATUS IS PROVIDED. SPECIFIC INFORMATION INCLUDES: CUSTOMER NUMBER, NAME, ADDRESS, YTD-PURCHASES, YTD-PAYMENTS, CREDIT LIMITS, AND DISCOUNT PERCENTAGE RATE. ALL PURCHASE AND PAYMENT TRANSACTIONS ARE ALSO LISTED. STATUS CAN BE ACTIVE, CLOSED, NEW (FIRST YEAR), OR VOIDED. ACTIVE AND NEW ACCOUNTS CAN CONDUCT BUSINESS WITH PVF. CLOSED ACCOUNTS MUST BE RE-ACTIVATED BEFORE THE SYSTEM WILL ACCEPT ACCOUNT TRANSACTIONS. THE SYSTEM WILL NOT ACCEPT TRANSACTIONS FOR VOIDED ACCOUNTS. IF MORE INFORMATION IS AVAILABLE THAN FITS ON A SCREEN PRESS THE F2 KEY. IF YOU PRESS THE KEY AND NOTHING HAPPENS, THEN THERE IS NO MORE INFORMATION TO VIEW. IF YOU WANT TO PRINT THE ACCOUNT INFORMATION, PRESS F9. IF YOU WANT TO RETURN TO THE MAIN MENU PRESS F10.

A poor help screen design

Figure 11-7b Contrasting the display of textual help information - An improved design for a help screen



Help

Help Information - *Reviewing Customer Account Status*

The intent of this screen is to retrieve customer account status information. Information related to current year-to-date balances, credit limits, and account status is provided.

Field

1. Customer Number	PVF assigned customer number
2. Name / Address	full customer name and mailing address
3. YTD-Purchases	total of current-year purchases
4. YTD-Payments	total of current-year payments
5. Credit Limit	maximum outstanding balance
6. Outstanding Balance	current account balance
7. Discount Percentage	sales discount rate
8. Status	current account status

Description of Account Status:

1. Active	in good standing
2. Closed	no longer a current account; must reapply to change status to active
3. New	in good standing but a customer for < 1 year
4. Voided	not in good standing (e.g., an excessive balance)

Special Function Keys:

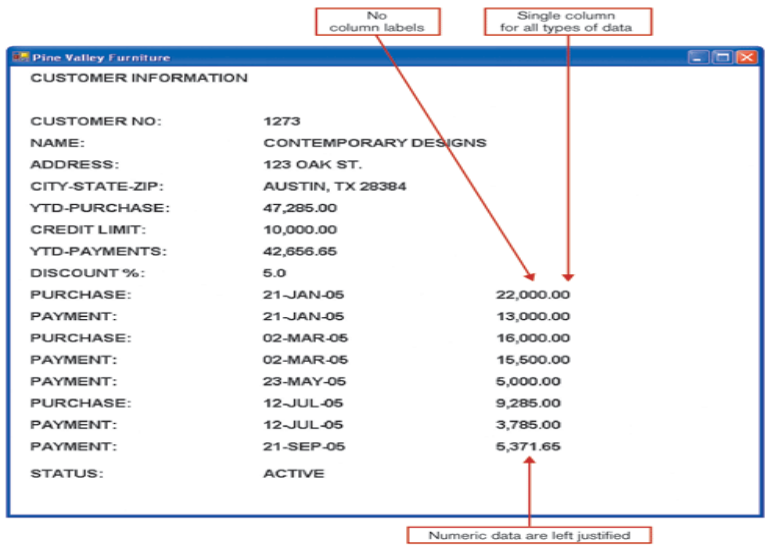
F1 = Help (displays this screen)
F2 = Account Details (is only displayed if > 1 page is available)
F9 = Print (prints "Customer Status Report")
F10 = Return (returns to prior screen)

A better help screen design

Guidelines for Tables and Lists

- Labels
  - All columns and rows should have meaningful labels.
  - Labels should be separated from other information by using highlighting.
  - Redisplay labels when the data extend beyond a single screen or page.
- **Formatting columns, rows and text**:
  - Sort in a meaningful order.
  - Columns should have at least two spaces between them.
  - Allow white space on printed reports for user to write notes.
  - Avoid overly fancy fonts.
- **Formatting numeric, textual and alphanumeric data:**
  - Right justify numeric data and align columns by decimal points or other delimiter.
  - Left justify textual data. Use short line length, usually 30 to 40 characters per line.

Figure 11-8a Contrasting the display of tables and lists (Pine Valley Furniture) - Poorly designed form


No column labels
Single column for all types of data
Pine Valley Furniture
Numeric data are left justified

CUSTOMER INFORMATION

CUSTOMER NO:	1273	
NAME:	CONTEMPORARY DESIGNS	
ADDRESS:	123 OAK ST.	
CITY-STATE-ZIP:	AUSTIN, TX 28384	
YTD-PURCHASE:	47,285.00	
CREDIT LIMIT:	10,000.00	
YTD-PAYMENTS:	42,656.65	
DISCOUNT %:	5.0	
PURCHASE:	21-JAN-05	22,000.00
PAYMENT:	21-JAN-05	13,000.00
PURCHASE:	02-MAR-05	16,000.00
PAYMENT:	02-MAR-05	15,500.00
PAYMENT:	23-MAY-05	5,000.00
PURCHASE:	12-JUL-05	9,285.00
PAYMENT:	12-JUL-05	3,785.00
PAYMENT:	21-SEP-05	5,371.65
STATUS:	ACTIVE	

A poor table design

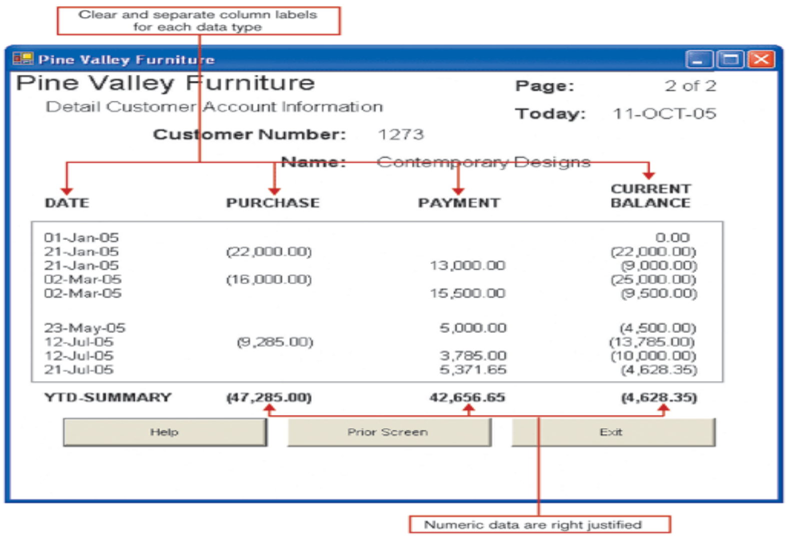

Clear and separate column labels for each data type
Numeric data are right justified
Pine Valley Furniture
Help
Prior Screen
Exit

Figure 11-8b Contrasting the display of tables and lists (Pine Valley Furniture) - Improved design for form

Pine Valley Furniture

Detail Customer Account Information

Page: 2 of 2

Today: 11-OCT-05

Customer Number: 1273

Name: Contemporary Designs

DATE	PURCHASE	PAYMENT	CURRENT BALANCE
01-Jan-05			0.00
21-Jan-05	(22,000.00)		(22,000.00)
21-Jan-05		13,000.00	(9,000.00)
02-Mar-05	(16,000.00)		(25,000.00)
02-Mar-05		15,500.00	(9,500.00)
23-May-05		5,000.00	(4,500.00)
12-Jul-05	(9,285.00)		(13,785.00)
12-Jul-05		3,785.00	(10,000.00)
21-Jul-05		5,371.65	(4,628.35)
YTD-SUMMARY	**(47,285.00)**	**42,656.65**	**(4,628.35)**

A better table design

5.4 File Organization

- A file is a collection of related sequence of records.
- A file organization is way of arranging the records in a file when the file is stored on the secondary storage(disk, tape, etc).
- The organization of records in a file is influenced by number of factors that must be taken into consideration while choosing a particular technique.
- These factors are :
    - Fast retrieval, updating and transfer of records
    - Efficient use of disk space
    - Efficient manipulation
    - Security from unauthorized access
    - Reduction in cost
    - Protection from failure

5.4.1 Heap File organization

- In this file organization , the records are stored in the file in the order in which they are inserted.
- All the new records are stored at the end of the file . This file organization is also called PILE FILE.
- Inserting a new record is very fast and efficient. But searching a record using any search condition involves a linear search through the file, which is comparatively more time consuming.
- It is generally used to store small files or in cases where data is difficult to organize.

5.4.2 Sequential File Organization

- In sequential file organization, records are stored in a sequential order according to the search key.
- A search key is an attribute or a set of attributes which are used to serialize the records. It is not necessary that search key must be primary key.
- It is simplest method of file organization, Sequential method is based on tape model.
- Devices who support sequential access are magnetic tapes, card readers etc.
- The records are stored in sequential order one after another.
- To reach at the consecutive record from any record pointers are used.
- The pointers are used for fast retrieval of records.

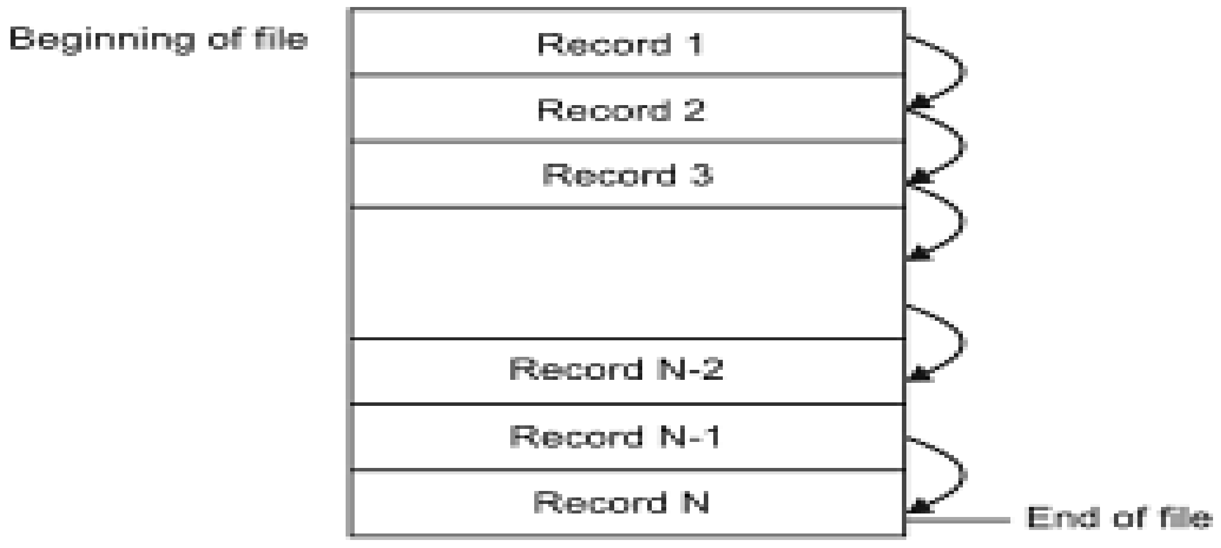


FIGURE 3.3. *Sequential file organization*.

5.4.3 Index sequential File

- Index sequential file organization is used to over come the disadvantage of sequential file organization.
- It also preserves the advantages of sequential access. This organization enables fast searching of records with the use of index.
- Some basic terms used:
  - **Block:** Block is a unit of storage in which records are saved.
  - **Index:** Index is a table with a search key by which block of a record can be find.
  - **Pointer:** Pointer is a variable which points from index entry to starting address of block.

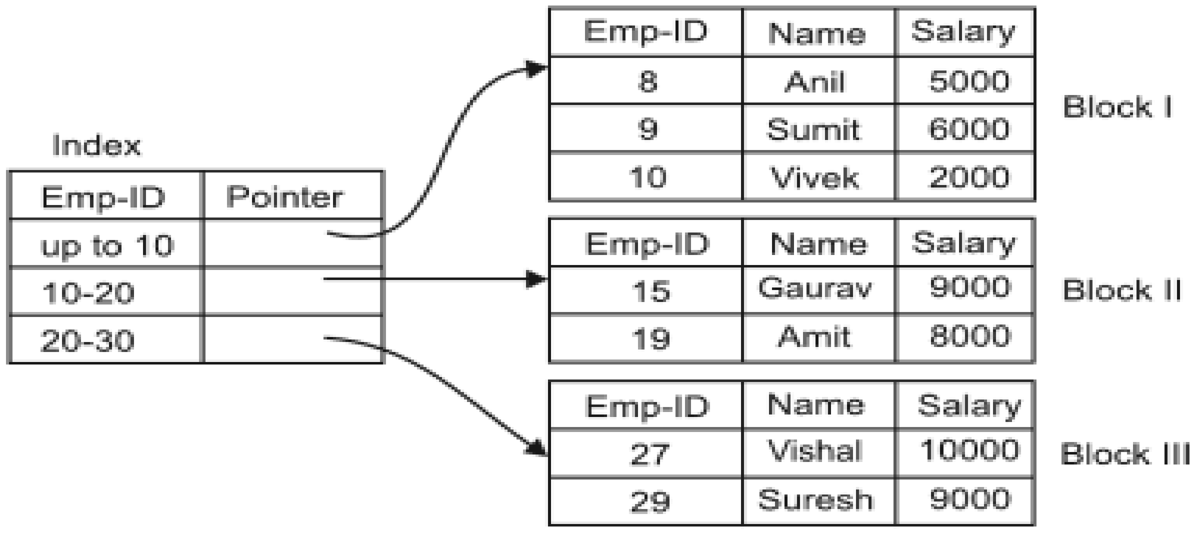


FIGURE 3.13. *Index sequential file organization.*

To manipulate any record, search key of index is entered to find the starting address of block and then required record is searched sequentially within the block

5.4.5 Hashing

Hashing is a technique by which key field is converted into address of physical location or record by using any function known as has hash function.

Hash function
Key field
Address of record

(*i*) **Mid square method :** In mid square hash method, first compute the square of key value and then take the central digits, which is the address of that record. Suppose you have 100 records in one file and the key value is 81. First compute square of 81 which is $(81)^2 = 6561$. After that take central digits of 6561. So, address of record having key value 81 is 56.

(*iii*) **Division method :** In division method, divide the key value with any number and take quotient as address of that record. Mostly prime number is taken as divisor. Consider a record having key value 550 which is divided by 5 gives 110 as quotient which is taken as address of that record.

(*iv*) **Division-remainder method :** In division-remainder method, divide the key value with any number (Prime number) and take remainder as address of that record. The divisor must be greater than total number of records and small then all key values. Suppose there are 100 records in any file. A record has key value 660 which is divided by 109 gives 6 as remainder which is taken as address of that record.

5.4.5 B Tree Index File

B-tree of order *m* is a tree which satisfies the following properties:

- Every node has at most *m* children.
- Every non-leaf node (except root) has at least $\lceil m/2 \rceil$ children.
- The root has at least two children if it is not a leaf node.

5.4.6 Direct File organization

- To meet the requirement to access records randomly direct file organization is used.
- In direct file organization records can be stored anywhere in the storage area but can be accessed directly without any sequential searching.

Assignment V

1. What is the difference between a 2 NF and 3NF relations?
2. What do you mean by file organization? Explain its types
3. Explain the six types of files used in information systems.
4. What is the normalization of a relation? Explain with example.
5. How can you normalize a relation? Explain with suitable example.
6. What do you mean by database normalization? Why is it important?
7. How can you transforming E-R diagram into relations? Explain with suitable example.
8. What are the process for designing forms and reports?
9. You have recently been hired by an appliance repair company. Your first task is to normalize the following relation.

Client No.	Last Name	Street Address	City	State	Technician No.	Technician Last Name	Service Date	Type of Service