

Data Flow Diagrams

DFD

- Data flow diagram symbols
- Data flow diagram levels
- Creating data flow diagrams
- Physical and logical data flow diagrams
- Partitioning
- Event driven modeling
- Use case and data flow diagrams



Data Flow Diagrams

- ❑ DFDs are one of the main methods available for analyzing data-oriented systems.
- ❑ DFDs emphasize the logic underlying the system.
- ❑ The systems analysts can put together a graphical representation of data movement through the organization.



Advantages of the Data Flow Diagram Approach

Four advantages over narrative explanations of data movement:

- Freedom from committing to the technical implementation too early.
- Understanding of the interrelationships of systems and subsystems.
- Communicating current system knowledge to users.
- Analysis of the proposed system.







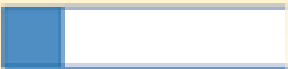
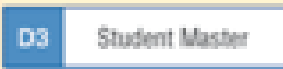


Basic Symbols

Four basic symbols are:

- A double square for an **external entity**--a source or destination of data.
- An arrow for (**data flow**) movement of data from one point to another.
- A rectangle with rounded corners for the occurrence of transforming **process**.
- An open-ended rectangle for a **data store**.

Basic Symbols

Symbol	Meaning	Example
	Entity	
	Data Flow	
	Process	
	Data Store	



External Entities

- ❑ Represent people or organizations outside of the system being studied
- ❑ Shows the initial source and final recipient of data and information
- ❑ Should be named with a noun, describing that entity

Customer



External Entities (2)

- External entities may be:
 - A person, such as CUSTOMER or STUDENT.
 - A company or organization, such as BANK or SUPPLIER.
 - Another department within the company, such as ORDER FULFILLMENT.
 - Another system or subsystem, such as the INVENTORY CONTROL SYSTEM.



Processes

□ Represent either:

- A whole system
- A subsystem
- Work being done, an activity



□ Names should be in the form verb-adjective-noun

- The exception is a process that represents an entire system or subsystem.



Data Stores

- Name with a noun, describing the data
- Data stores are usually given a unique reference number, such as D1, D2, D3.
- Include any data stored, such as:
 - A computer file or database.
 - A transaction file .
 - A set of tables .
 - A manual file of records.

D1

Customer
Master



Data Flow

- ❑ Data flow shows the data about a person, place, or thing that moves through the system.
- ❑ Names should be a noun that describes the data moving through the system.
- ❑ Arrowhead indicates the flow direction.
- ❑ Use double headed-arrows only when a process is reading data and updating the data on the same table or file.



New Customer



Customer Record



Developing Data Flow Diagrams

Use the following guidelines:

- Create the context level diagram, including all external entities and the major data flow to or from them.
- Create Diagram 0 by analyzing the major activities within the context process.
 - Include the external entities and major data stores.
- Create a child diagram for each complex process on Diagram 0.

Context - Diagram

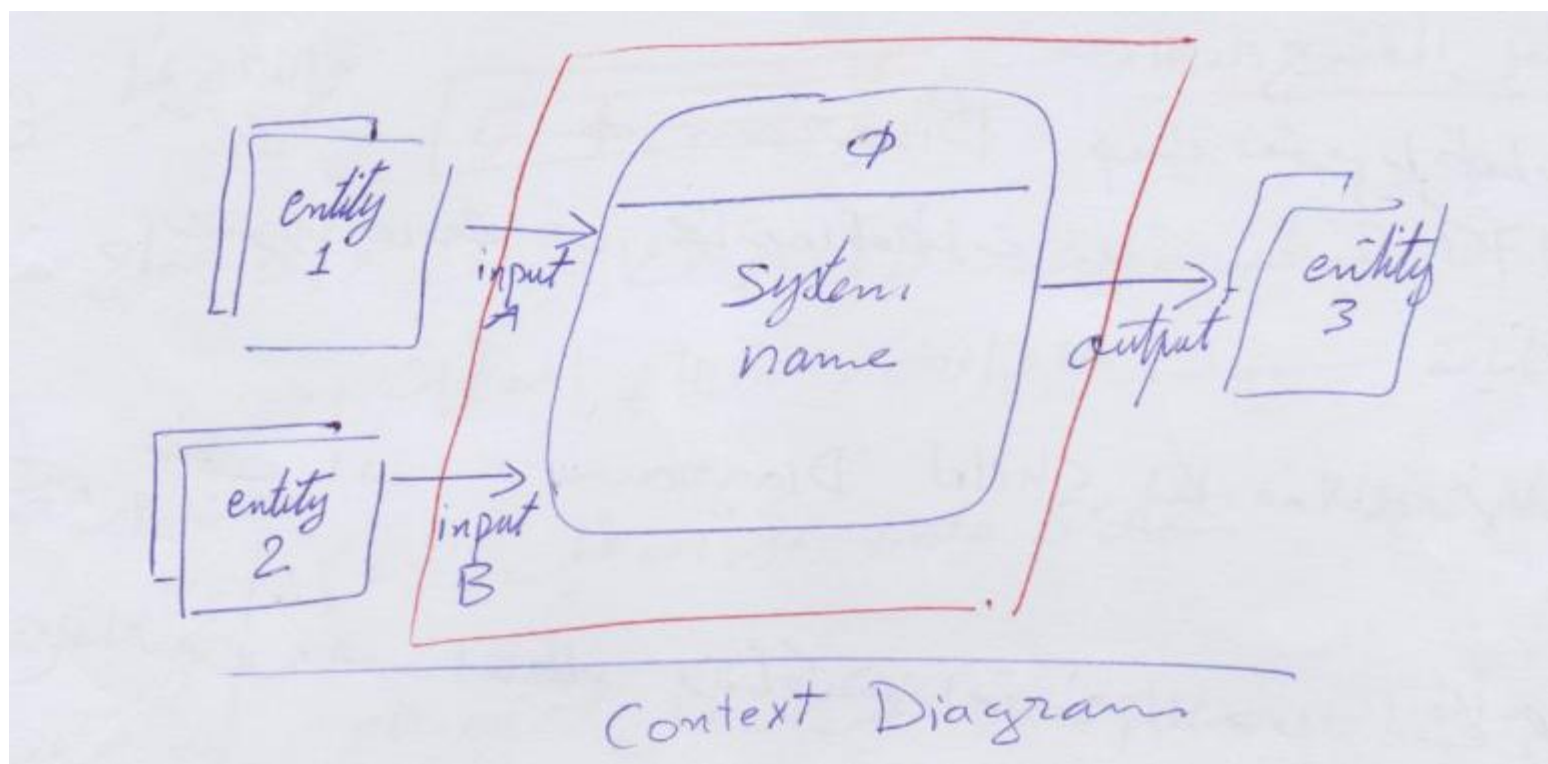
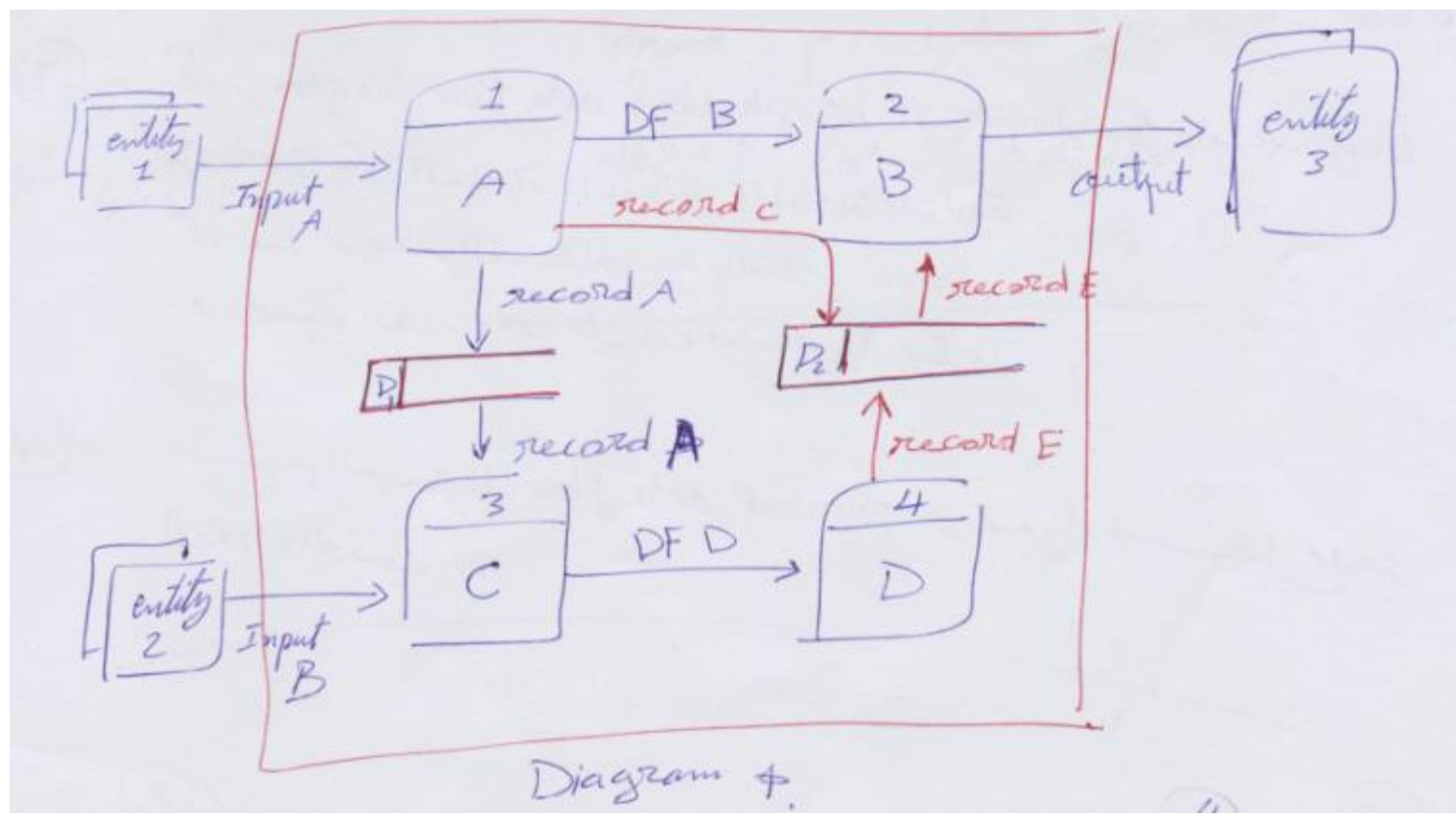
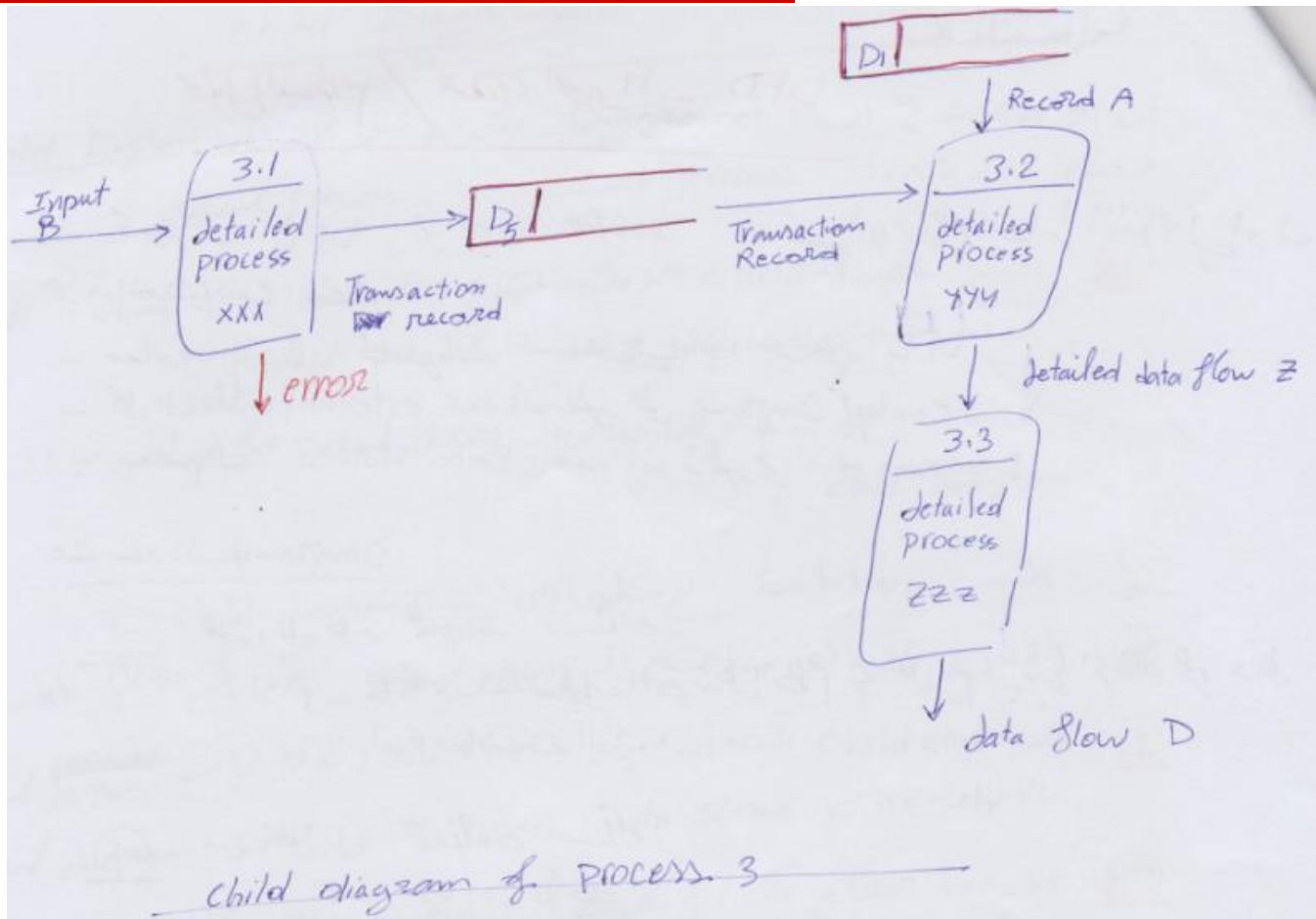


Diagram 0



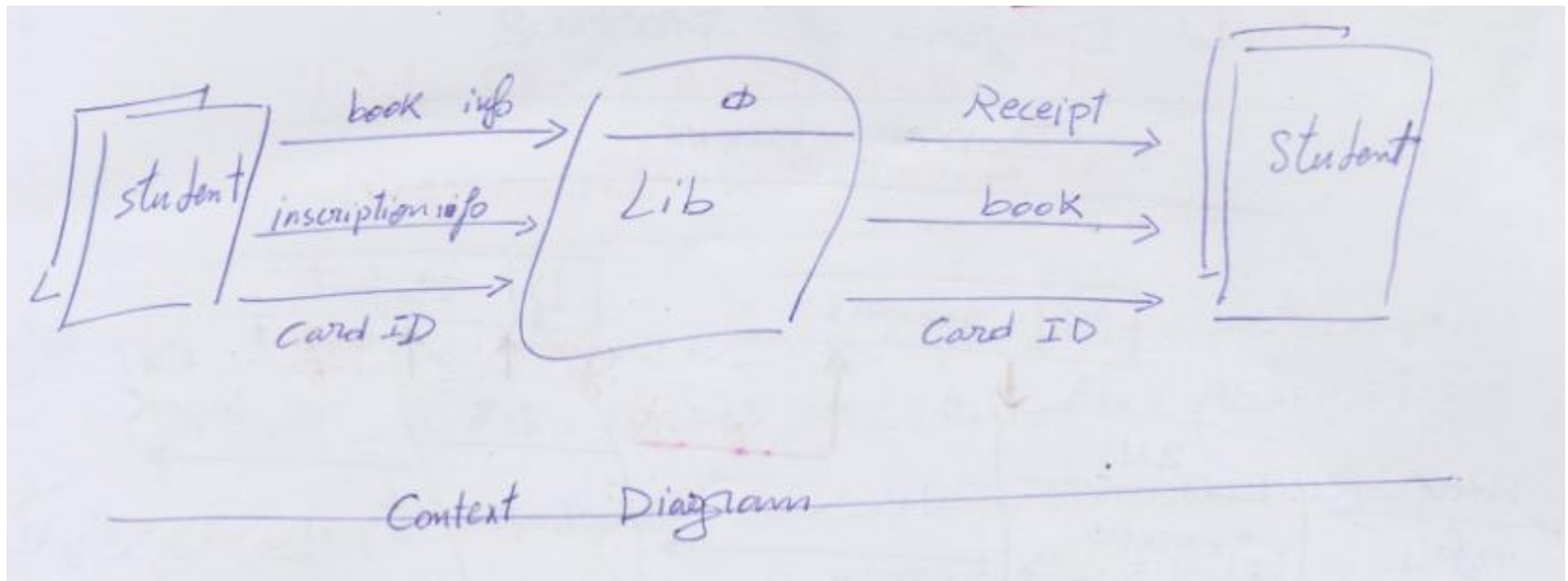


Child process



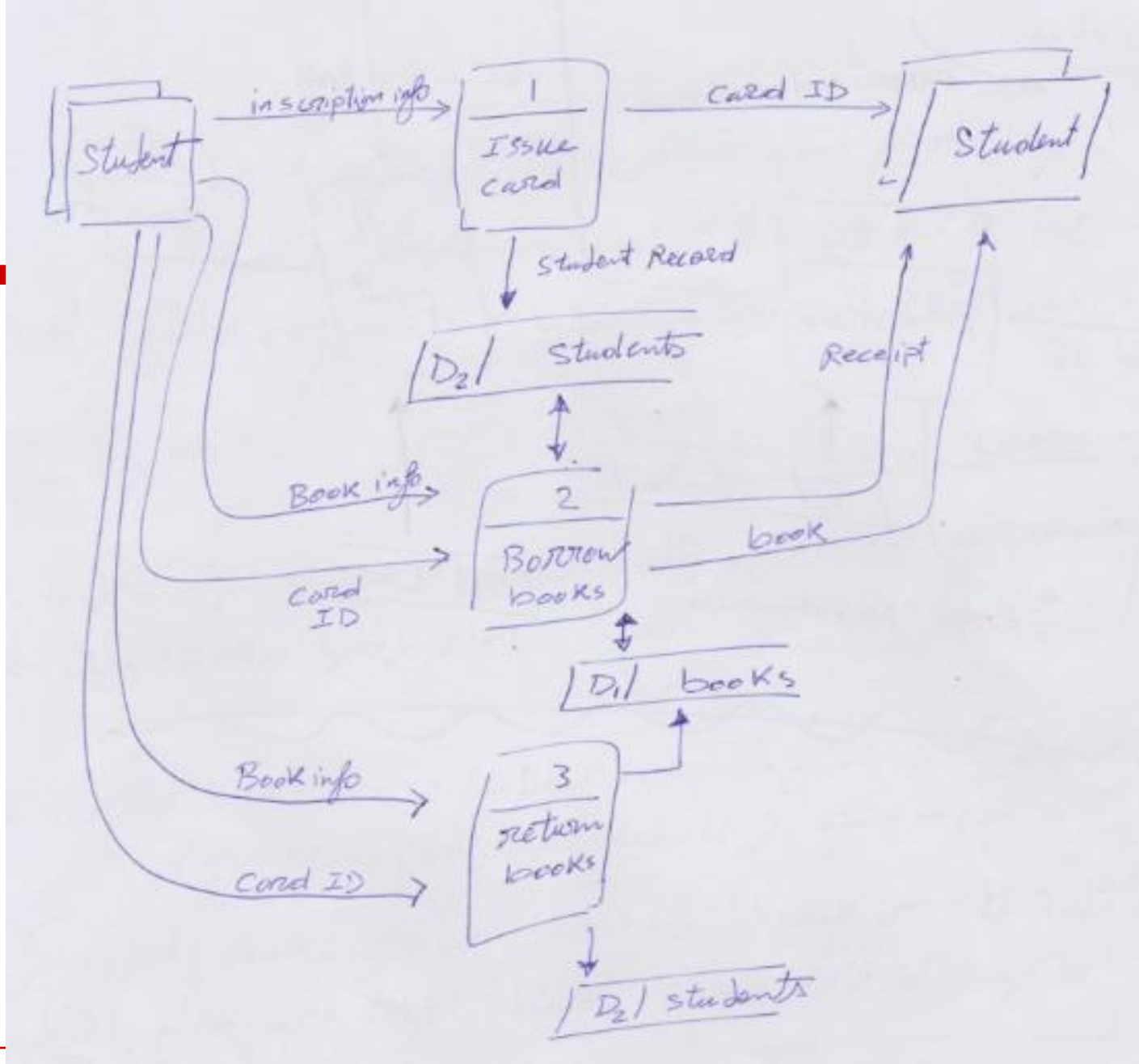


Library Context - Diagram





Library Diagram 0

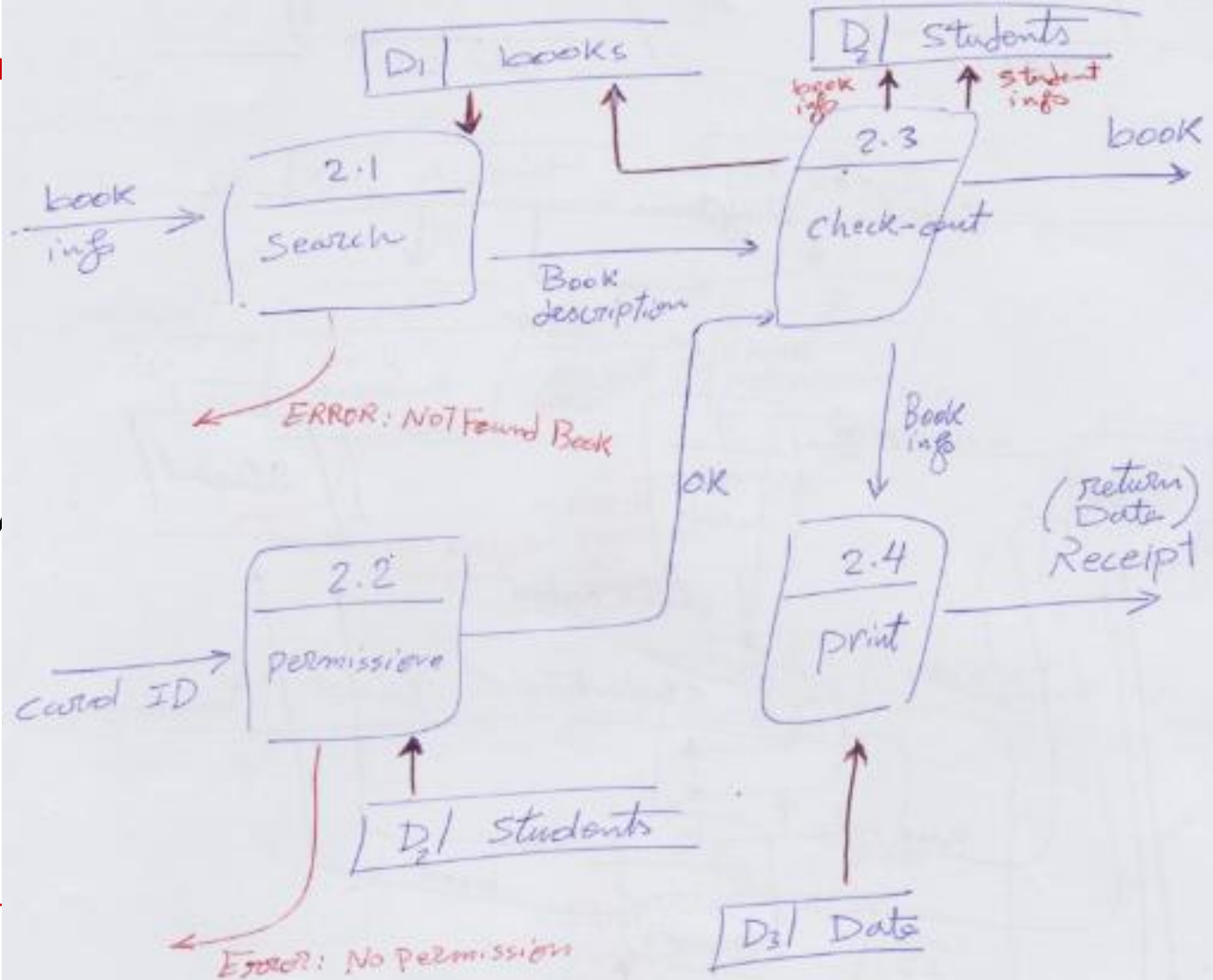


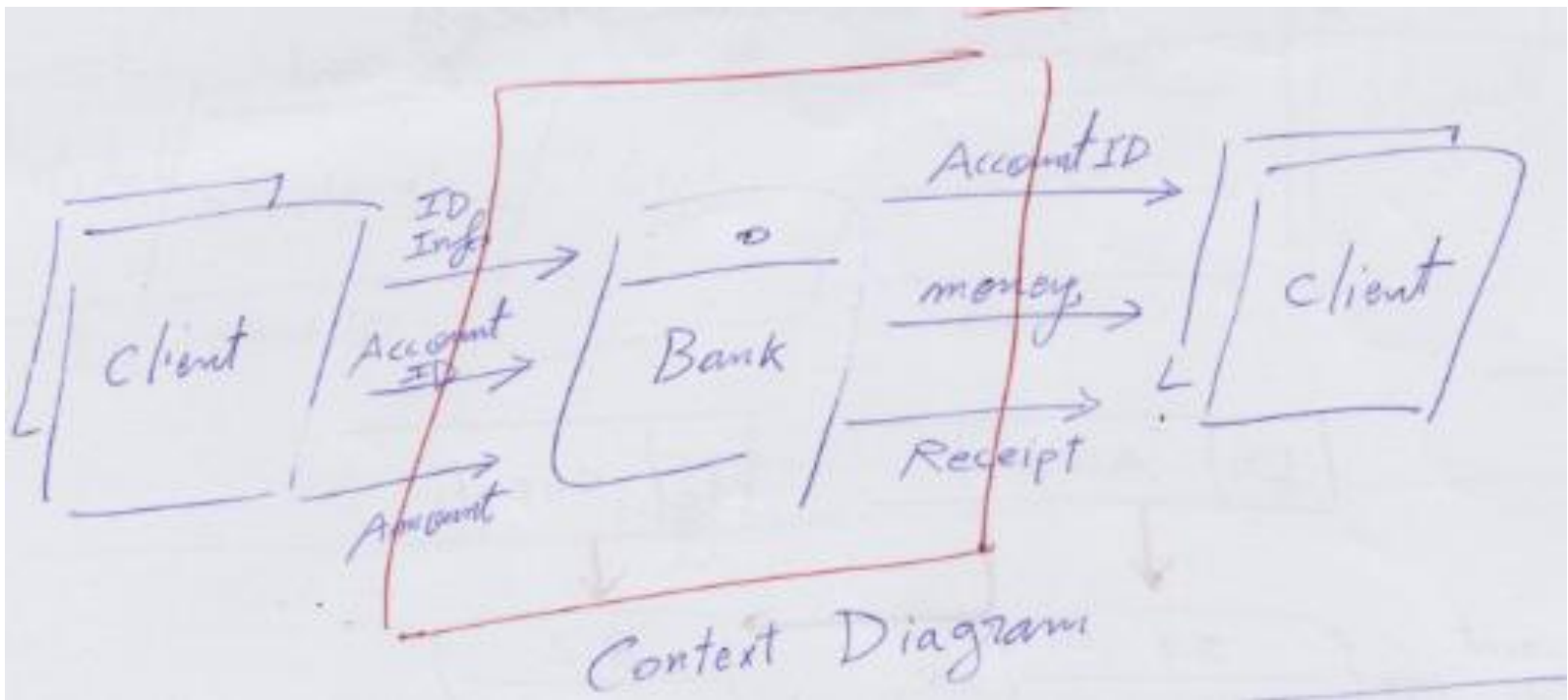


Library
child
process of
borrow book

Child Diagram of Process 2

Borrow books





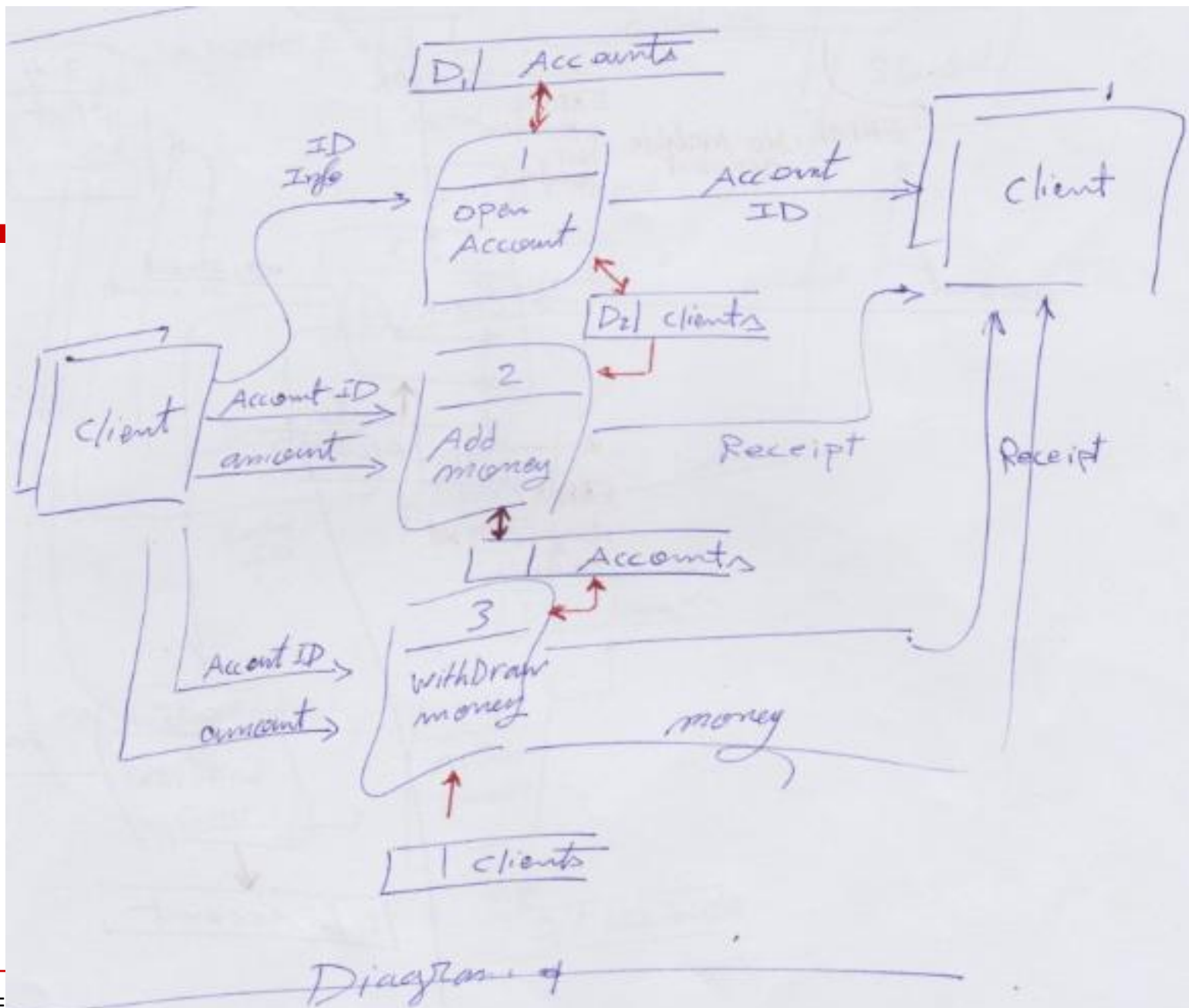
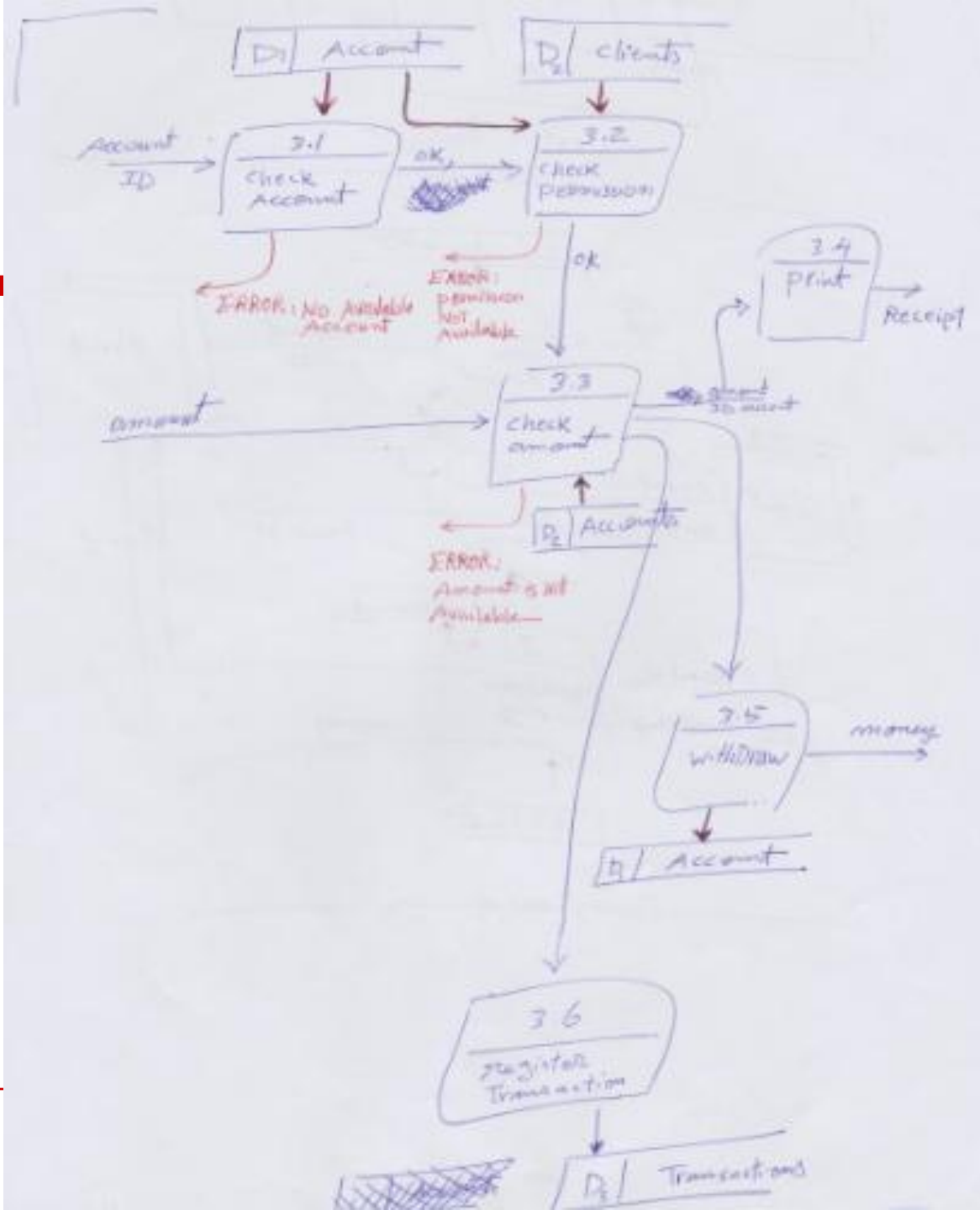


Diagram of





Creating Data Flow Diagrams

Detailed data flow diagrams may be developed by:

- Making a list of business activities.
- Analyzing what happens to an input data flow from an external entity.
- Analyzing what is necessary to create an output data flow to an external entity.



Creating Data Flow Diagrams

Detailed data flow diagrams may be developed by (continue):

- Examining the data flow to or from a data store.
- Analyzing a well-defined process for data requirements and the nature of the information produced.
- Noting and investigating unclear areas.



Data Flow Diagram Levels

- ❑ Data flow diagrams are built in layers.
- ❑ The top level is the Context level.
- ❑ Each process may explode to a lower level.
- ❑ The lower level diagram number is the same as the parent process number.
- ❑ Processes that do not create a child diagram are called primitive.



Context-Level Data Flow Diagram

- ❑ It contains only one process, representing the entire system.
- ❑ The process is given the number zero.
- ❑ All external entities are shown on the context diagram as well as major data flow to and from them.
- ❑ The diagram does not contain any data stores.



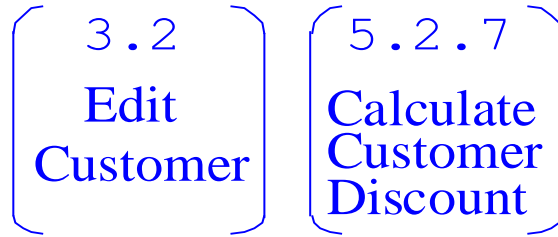
Diagram 0

- ❑ Diagram 0 is the explosion of the context level diagram.
- ❑ It should include up to 7 or 9 processes.
 - Any more will result in a cluttered diagram.
- ❑ Processes are numbered with an integer.
- ❑ The major data stores and all external entities are included on Diagram 0.



Child Diagrams

- ❑ Each process on diagram zero may be exploded to create a child diagram.
- ❑ Each process on a lower-level diagram may be exploded to create another child diagram.
- ❑ These diagrams found below Diagram 0 are given the same number as the parent process.
 - Process 3 would explode to Diagram 3.



Child Diagrams (2)

- Each process is numbered with the parent diagram number, a period, and a unique child diagram number.
- Examples are:
 - 3.2 on Diagram 3, the child of process 3.
 - 5.2.7 on Diagram 5.2, child of process 5.2.
 - On Diagram 3, the processes would be numbered 3.1, 3.2, 3.3 and so on.



Child Diagrams (3)

- ❑ External entities are usually not shown on the child diagrams below Diagram 0.
- ❑ If the parent process has data flow connecting to a data store, the child diagram may include the data store as well.



Child Diagrams (4)

- A lower-level diagram may contain data stores not shown on the parent process, such as:
 - A file containing a table of information (such as a tax table).
 - A file linking two processes on the child diagram.
- Minor data flow, such as an error line, may be included on a child diagram.



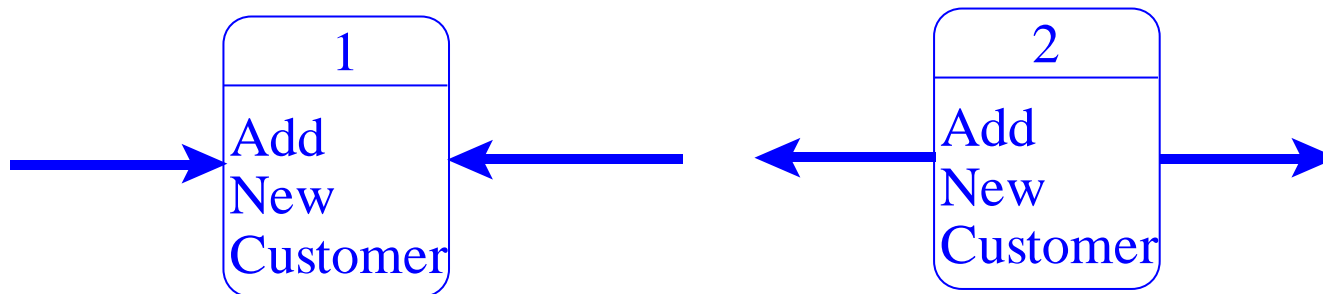
Child Diagrams (5)

- ❑ An interface data flow is data that are input or output from a child diagram that matches the parent diagram data flow.
- ❑ Processes that do not create a child diagram are called primitive processes.
- ❑ Logic is written for these processes.



Data Flow Diagram Errors

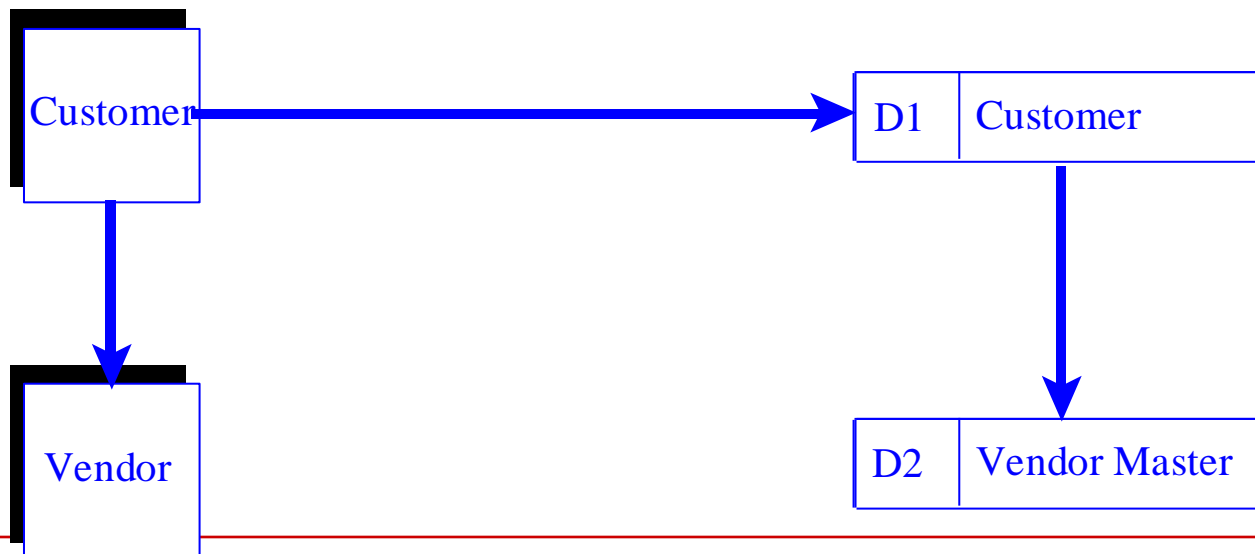
- ❑ The following conditions are errors that occur when drawing a data flow diagram:
- ❑ A process with only input data flow or only output data flow from it.





Data Flow Diagram Errors (2)

- ❑ Data stores or external entities are connected directly to each other, in any combination.





Data Flow Diagram Errors (3)

- ❑ Incorrectly labeling data flow or objects
 - Examples are:
 - ❑ Labels omitted from data flow or objects.
 - ❑ Data flow labeled with a verb.
 - ❑ Processes labeled with a noun.
- ❑ Too many processes on a data flow diagram.
 - Nine is the suggested maximum.



Data Flow Diagram Errors (4)

- ❑ Omitting data flow from the diagram
- ❑ Unbalanced decomposition between a parent process and a child diagram
 - The data flow in and out of a parent process must be present on the child diagram.



Logical Data Flow Diagrams

- ❑ Logical data flow diagrams show how the business operates.
- ❑ They have processes that would exist regardless of the type of system implemented.



Data Flow Diagram Progression

The progression of creating data flow diagrams is:

- Create a logical DFD of the current system.
- Next add all the data and processes not in the current system that must be present in the new system.
- Finally derive the physical data flow diagram for the new system.



Data Flow Diagram Progression

Figure 7.8 The progression of models from logical to physical.

