

SECD2613

System Analysis and Design



PART IV

The Analysis Process #1:

Introduction to Data Flow Diagram (DFD)

www.utm.my

innovative • entrepreneurial • global



univteknologimalaysia

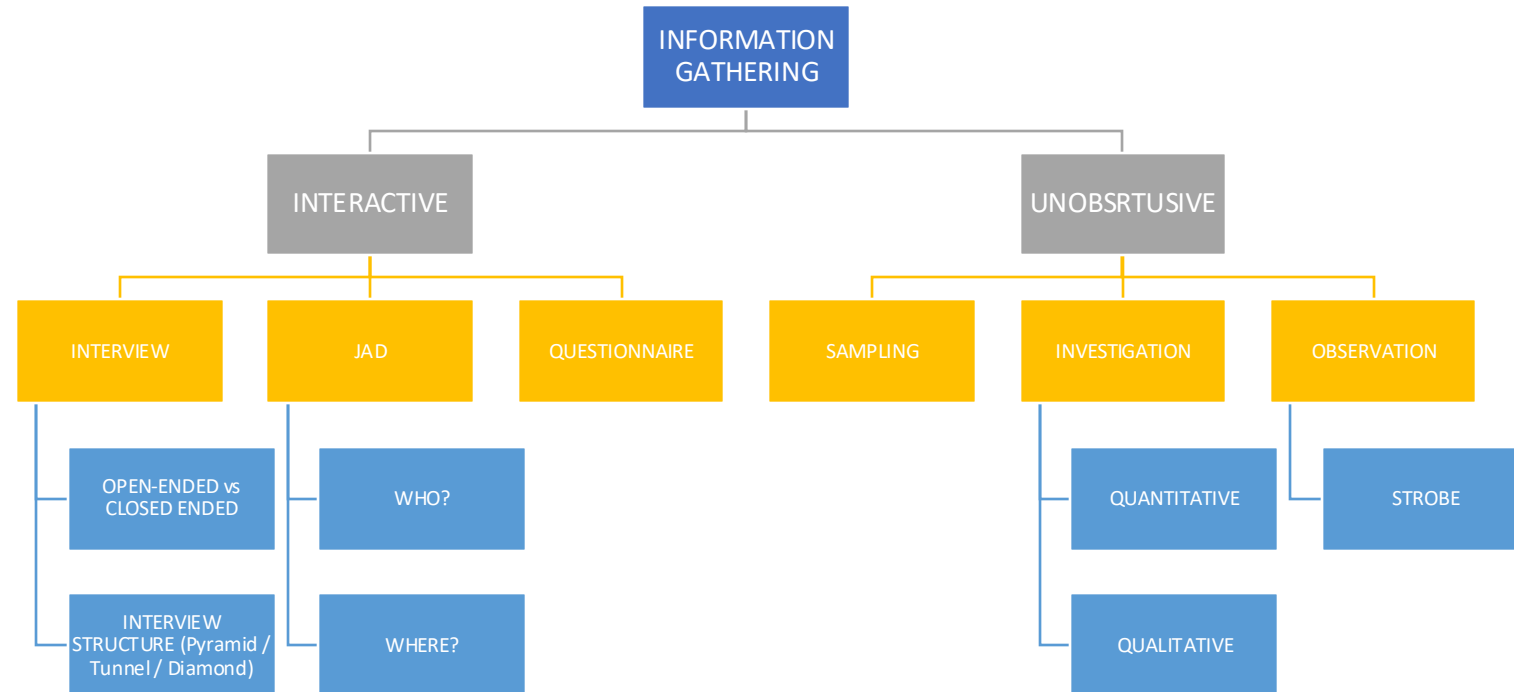


utm_my



utmofficial

Most Important Learning Yesterday (MILY)

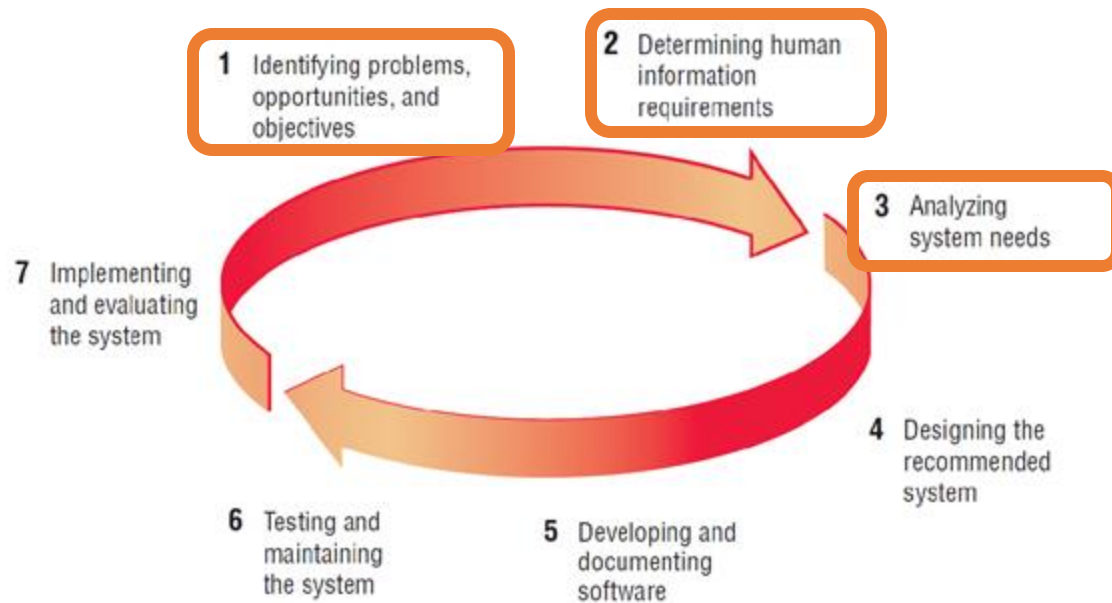


Output:

- Understands how users accomplish their work.
- Know the **BUSINESS FUNCTIONS** and have complete information on the people, goals, data and procedure involved.



SDLC – THE SEVEN PHASES



Activity:

- Create data flow diagrams
- Complete the data dictionary
- Analyze the structured decisions made
- Prepare and present the system proposal

■ OBJECTIVES

1. Comprehend the importance of using logical and physical data flow diagrams (DFDs) to graphically depict movement for humans and systems in an organization.
2. Create, use, and explode logical DFDs to capture and analyze the current system through parent and child levels.
3. Develop and explode logical DFDs that illustrate the proposed system.
4. Produce physical DFDs based on logical DFDs you have developed.
5. Understand and apply the concept of partitioning of physical DFDs.

■ MAJOR TOPICS

DATA FLOW DIAGRAM

- What is DFD?
- DFD Symbols

← PART I

CREATING DFD

- Steps to create DFD
- Context Diagram
- Diagram 0
- DFD Level

← PART II & III

LOGICAL & PHYSICAL DFD

- From logical to physical DFD
- CRUD matrix

← PART IV & V

PARTITIONING

- Partitioning DFD

← PART VI

PART I

- What is DFD?
- DFD Symbols

■ DATA FLOW DIAGRAM

- Graphically characterize **data processes and flows** in a business system
- Depict:
 - i. System inputs
 - ii. Processes
 - iii. Outputs



■ DATA FLOW DIAGRAM

An Example:

- Developing a Café Management System
- What happens at the café when you buy food?

What are the input, process and output?

■ DATA FLOW DIAGRAM

1. **External Entity:**

•**Student (Customer)**

- Comes to the café and places an order.

2. **Process:**

•**Take Order & Process Payment**

- The cashier inputs the order into the system and receives payment.

•**Prepare Food**

- The kitchen staff prepares the nasi lemak and drink.

•**Deliver Order**

- The prepared food is handed over to the student.

3. **Data Store:**

•**Menu Database**

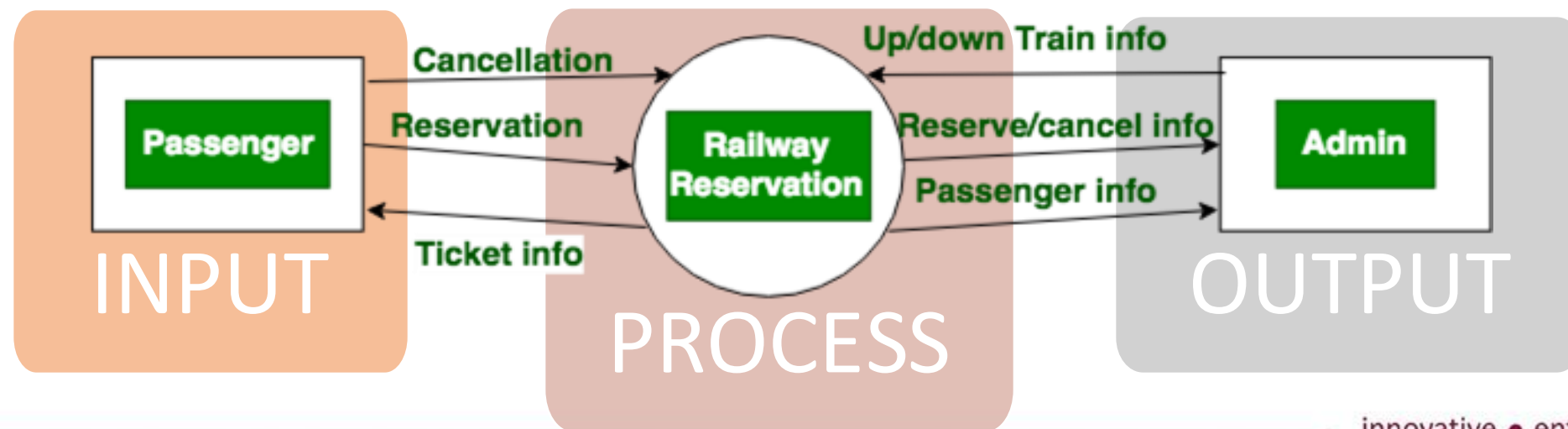
- The system checks what's available.

•**Transaction Record**


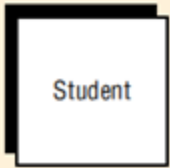
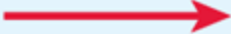
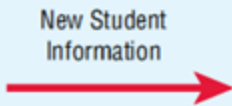

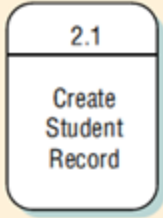

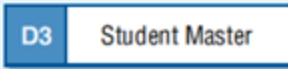
- The payment and order details are saved for reporting or stock management.

■ ADVANTAGE OF DATA FLOW APPROACH

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

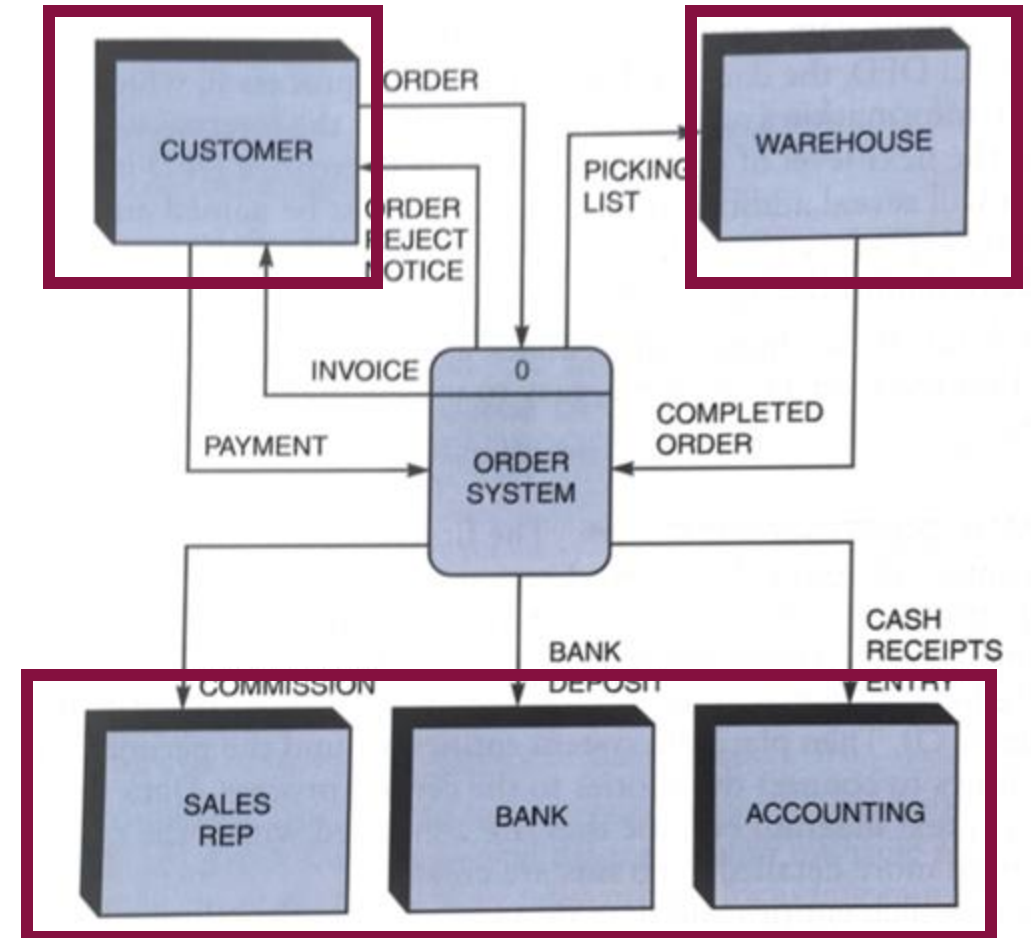


DFD BASIC SYMBOLS

| | Symbol | Meaning | Example |
|---|---|------------|---|
| A double square for an external entity |  | Entity |  |
| An arrow for movement of data from one point to another |  | Data Flow |  |
| A rectangle with rounded corners for the occurrence of a transforming process |  | Process |  |
| An open-ended rectangle for a data store |  | Data Store |  |

■ DFD – EXTERNAL ENTITIES

- Represent another department, a business, a person, or a machine
- A source or destination of data, outside the boundaries of the system
- Should be named **with a noun**



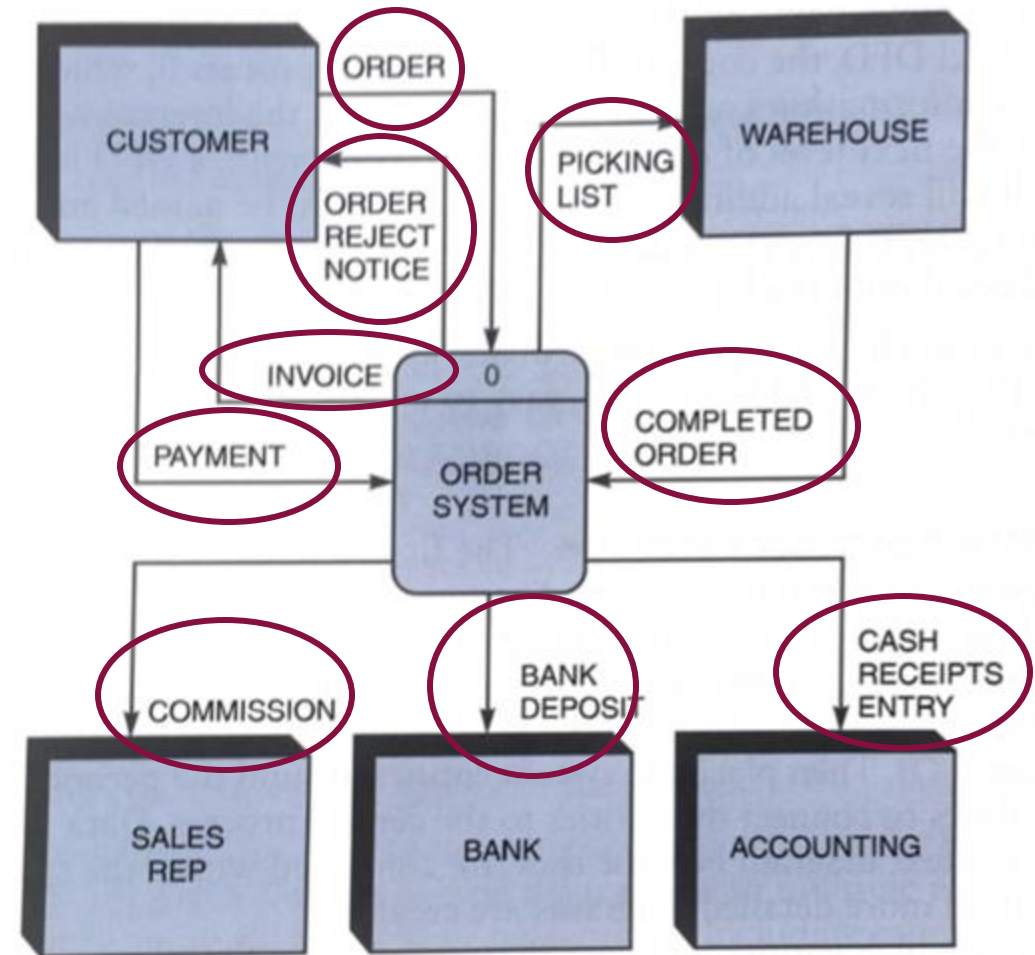
■ DFD – DATA FLOW

- Shows **movement of data** from one point to another
- Arrowhead indicates the flow direction
- Represents data about a person, place, or thing - **described with a noun**

student_ID, student_name, student_course

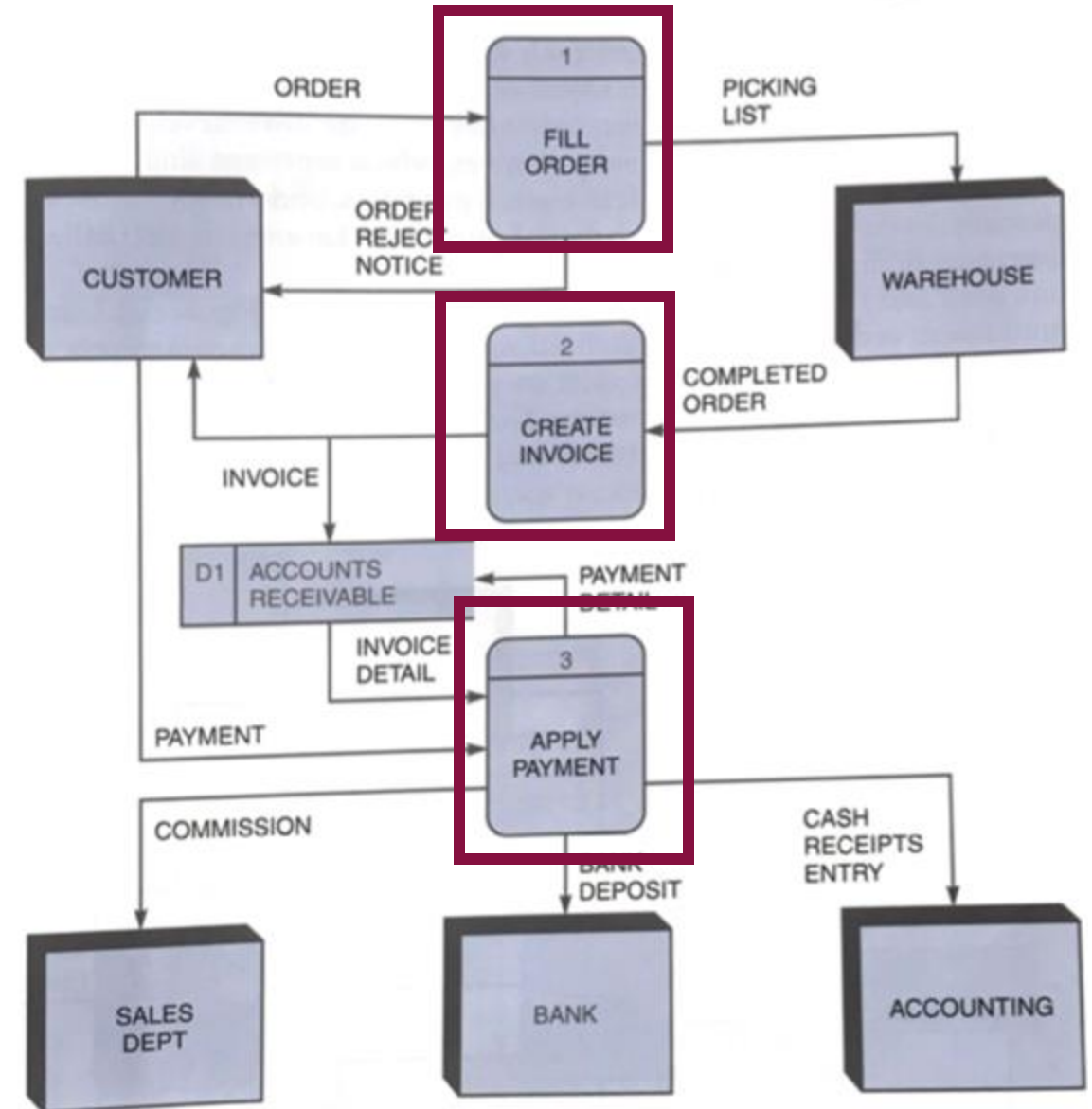


student_Information



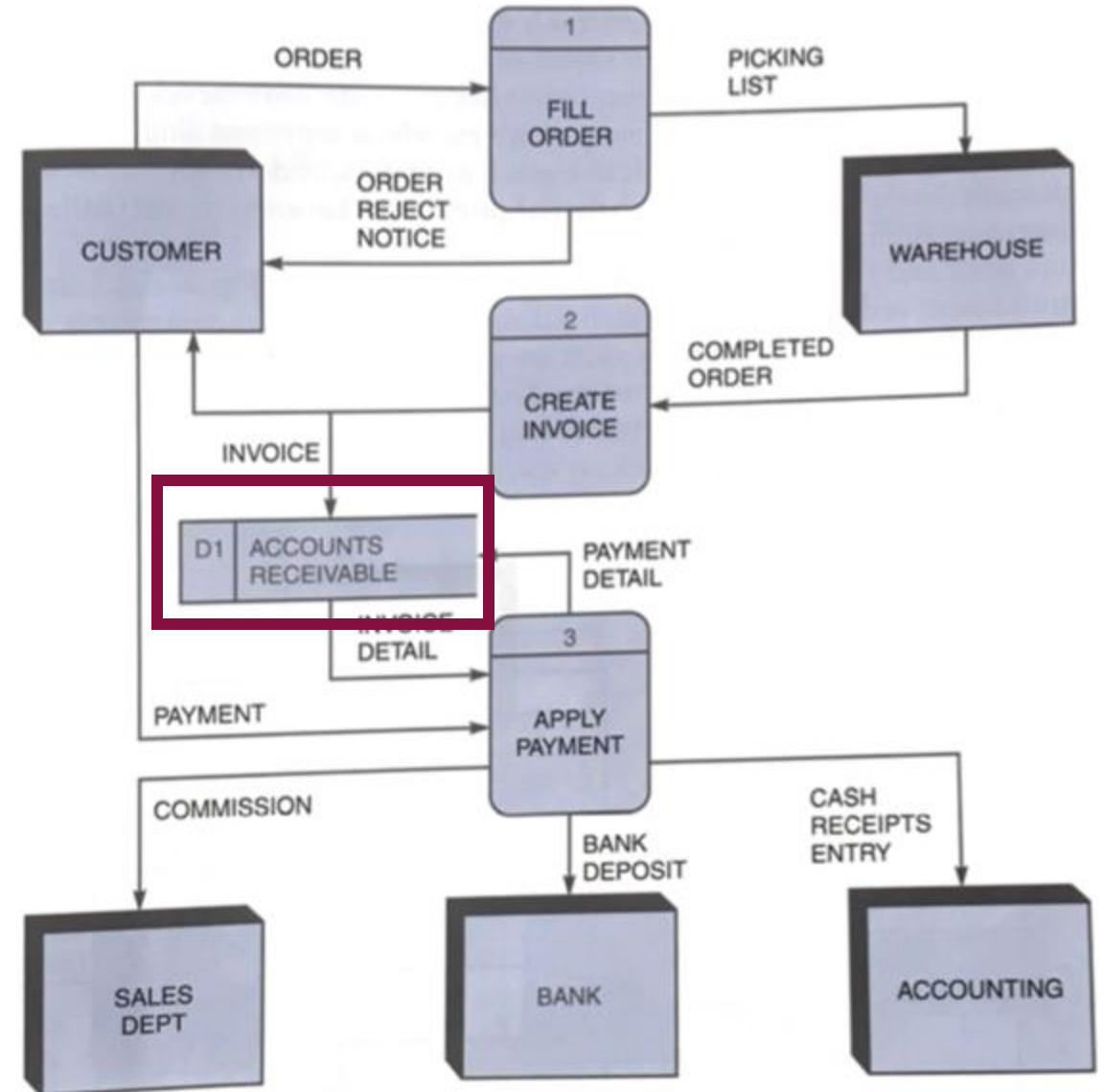
DFD – PROCESS

- Denotes a change in or transformation of data
- Represents work being performed in the system
- Naming convention:
 - Assign the **name of the whole system when naming a high-level process** (refer previous diagram)
 - To name a major subsystem attach the word subsystem to the name
 - Use the form verb-adjective-noun or verb-noun for detailed processes


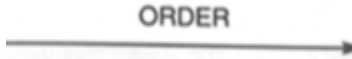
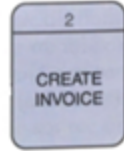
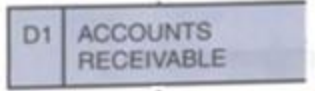


■ DFD – DATA STORE

- A depository for data that allows examination, addition, and retrieval of data
- **Named with a noun**, describing the data
- Data stores are usually given a unique reference number, such as D1, D2, D3
- Represents a:
 - Database
 - Computerized file
 - Filing cabinet



WHAT'S IMPORTANT?

| THE TERMS | ENTITY | DATA FLOW | PROCESS | DATA STORE |
|------------------------------------|--|---|---|---|
| THE SYMBOLS |  |  |  |  |
| THE MEANING (what it represent) | Source or destination | Movement of data | Work to be done (performed) in the system | Depository of the data |
| THE NAMING (how to describe) | NOUN | NOUN | VERB (except in Context Diagram) | NOUN |
| SOMETHING EXTRA (tips alert!) | This can be anything that interacts with the system, usually a PERSON (e.g.: user) or another system. | Arrow name can be represented by: 1. Individual list of all data. 2. A summarized name for related data | VERB VERB VERB! (most times students FORGET about this!) | Usually represent data storage of the system (a table of the database). |

PART II

- Steps to create DFD
- Context Diagram
- Diagram 0
- Child Diagram

■ STEPS TO DEVELOP DFD

Developing Data Flow Diagrams Using a Top-Down Approach

1. Make a list of business activities and use it to determine various
 - External entities
 - Data flows
 - Processes
 - Data stores
2. Create a context diagram that shows external entities and data flows to and from the system. Do not show any detailed processes or data stores.
3. Draw Diagram 0, the next level. Show processes, but keep them general. Show data stores at this level.
4. Create a child diagram for each of the processes in Diagram 0.
5. Check for errors and make sure the labels you assign to each process and data flow are meaningful.
6. Develop a physical data flow diagram from the logical data flow diagram. Distinguish between manual and automated processes, describe actual files and reports by name, and add controls to indicate when processes are complete or errors occur.
7. Partition the physical data flow diagram by separating or grouping parts of the diagram in order to facilitate programming and implementation.

■ BASIC RULES


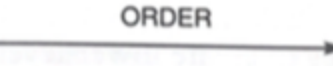

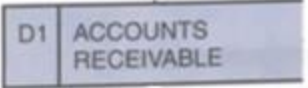
- The data flow diagram **must have at least one process**
- Must not be any freestanding objects
- A process **must have both an input and output data flow**
- A **data store** must be **connected to at least one process**
- External entities **should not be connected** to one another

STEP 1:

LIST OF POSSIBLE ENTITIES, DATA FLOW, PROCESS AND DATA STORE

7 STEPS OF CREATING DFD

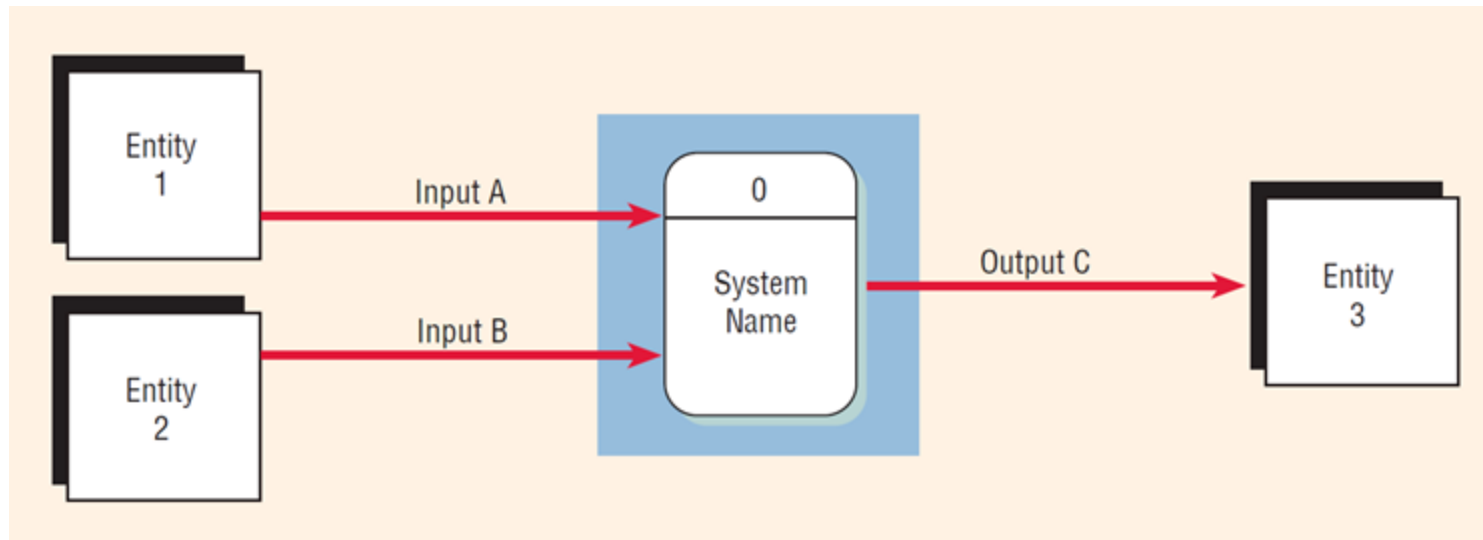
Step 1: List the ENTITY, DATA FLOW, PROCESS & DATA STORE

| THE TERMS | ENTITY | DATA FLOW | PROCESS | DATA STORE |
|------------------------------------|---|---|---|---|
| THE SYMBOLS |  |  |  |  |
| THE MEANING (what it represent) | Source or destination | Movement of data | Work to be done (performed) in the system | Depository of the data |
| THE NAMING (how to describe) | NOUN | NOUN | VERB - NOUN (except in Context Diagram) | NOUN |

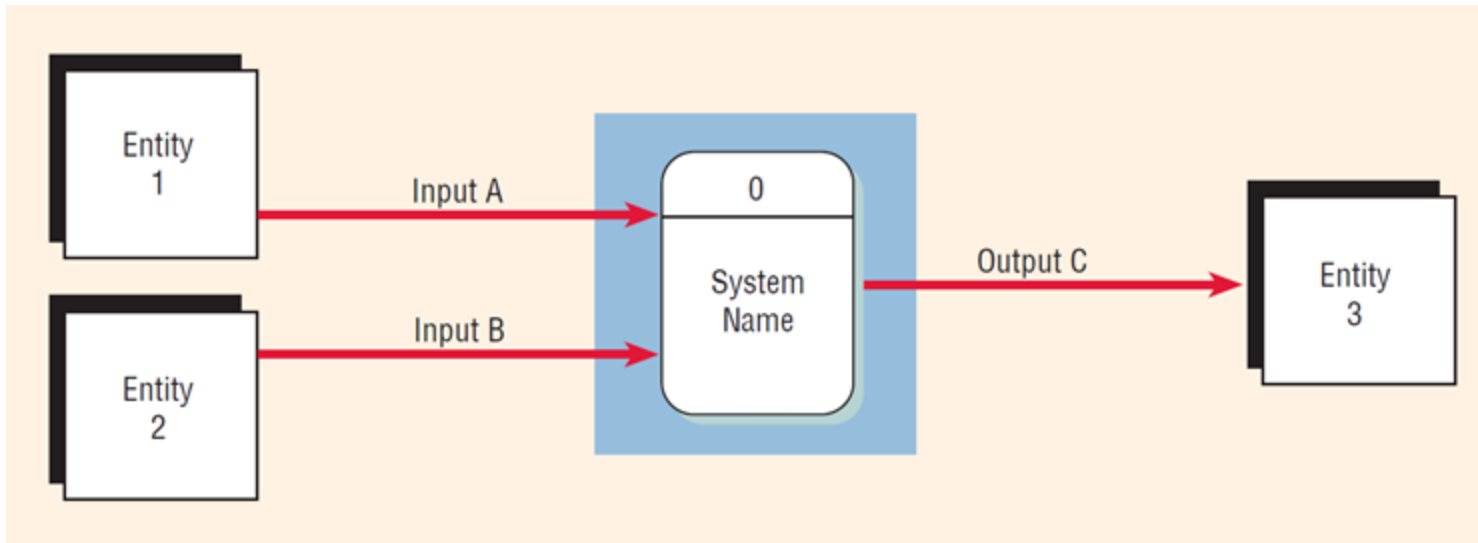
STEP 2: CONTEXT DIAGRAM

■ CREATING CONTEXT DIAGRAM

- The highest level in a data flow diagram
- Contains only one process, representing the **entire system**
- The process is given the number 0
- All external entities, as well as major data flows are shown



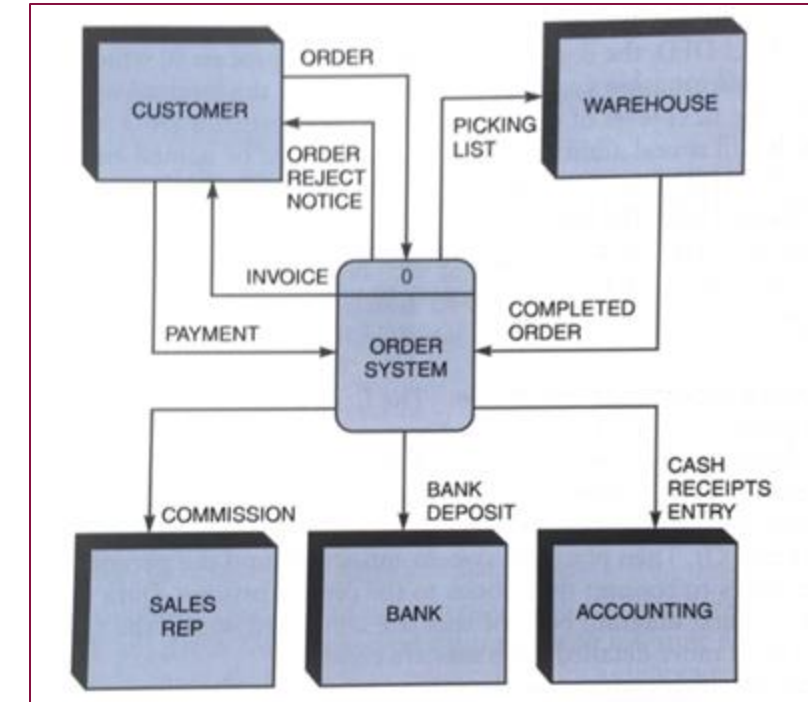
CREATING CONTEXT DIAGRAM



Basically the context diagram consists of:

- one process—depicting the entire system
- external entities
- data flows from the external entities to the process

The diagram does NOT contain any data stores.



CONTEXT DIAGRAM EXAMPLE

STEP 3: DIAGRAM 0 (ZERO)

■ DRAWING DIAGRAM 0

- The explosion of the context diagram
- May include up to nine processes
- To draw a Diagram 0:
 1. Start with the data flow from an entity on the input side
 2. Work backward from an output data flow.
 3. Examine the data flow to or from a data store
 4. Analyze a well-defined process
 5. Take note of any fuzzy areas

“What happens to the data entering the system?”

“Is it stored?”

“Is it input for several processes?”

“Where does it come from?”

“Is it calculated or stored on a file?”

“What processes put data into the store?”

“What processes use the data?”

■ DRAWING DIAGRAM 0

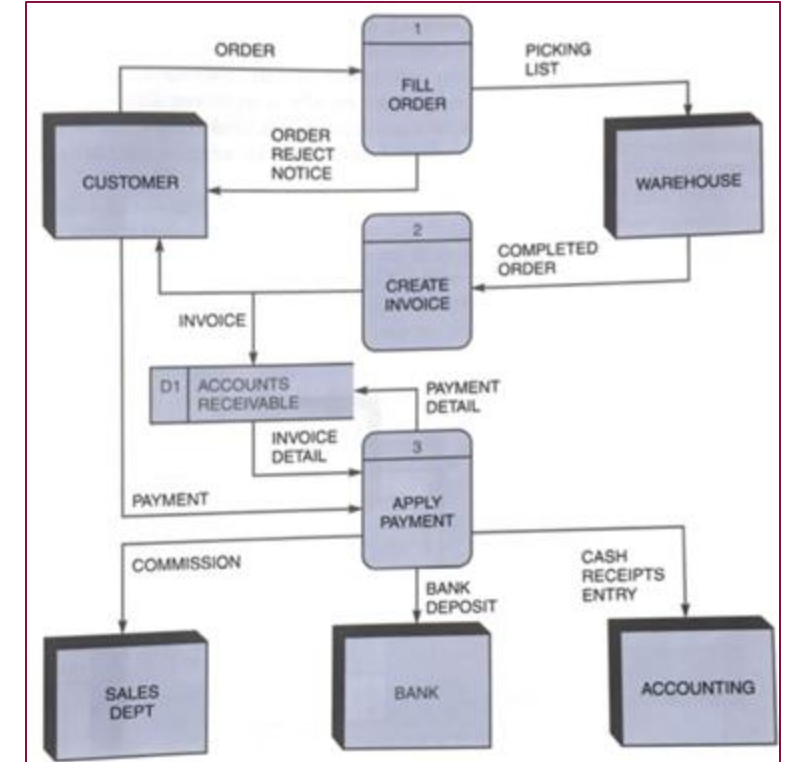
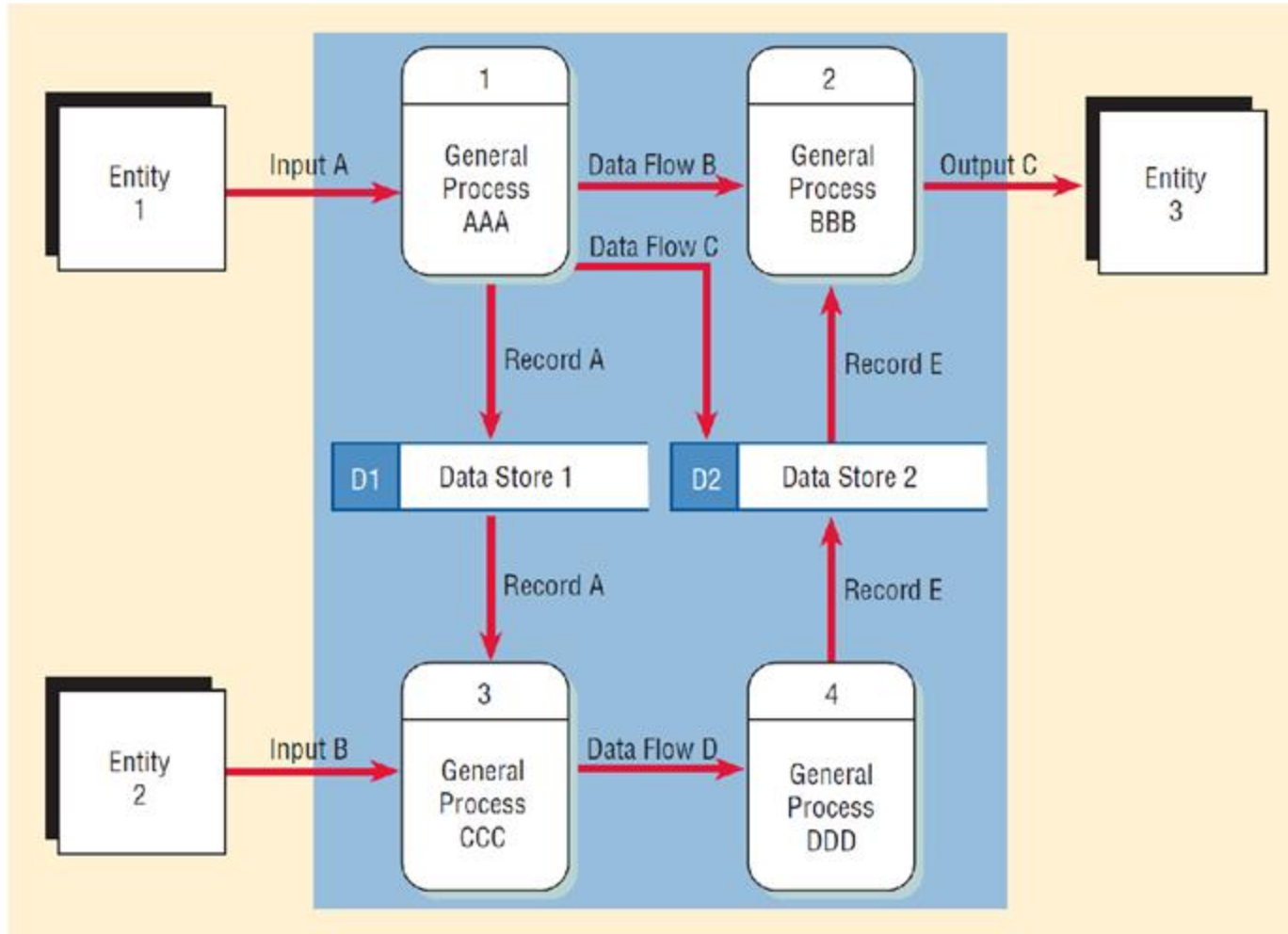


DIAGRAM 0 EXAMPLE

Each process is numbered
 Major data stores and all external entities are included

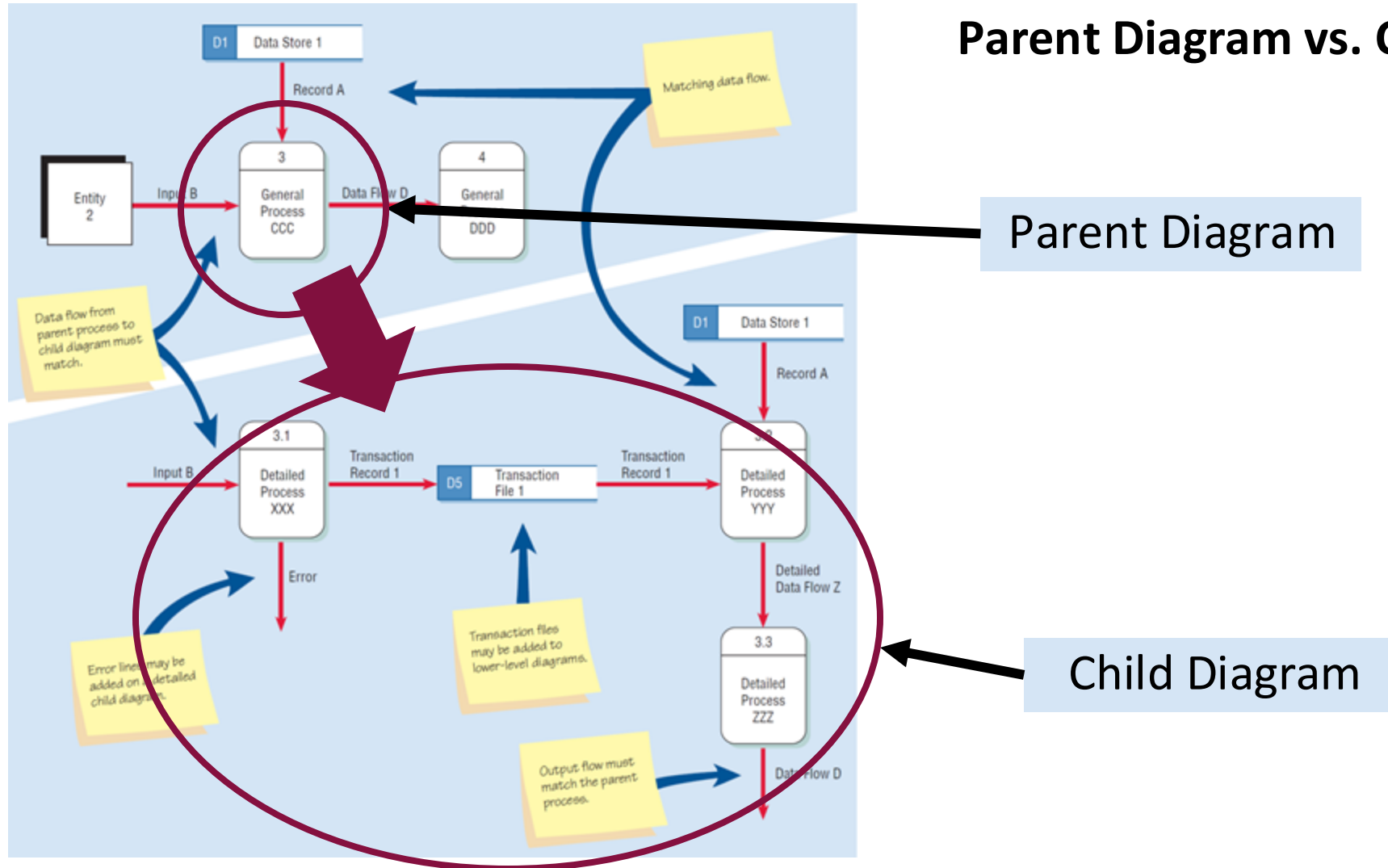
STEP 4: CHILD DIAGRAM

■ CREATING CHILD DIAGRAM

- Each process on diagram 0 may be exploded to create a child diagram
- A child diagram cannot produce output or receive input that the parent process does not also produce or receive
- The child process is given the same number as the parent process
 - Process 3 would explode to Diagram 3
- 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
- When a process is not exploded, it is called a primitive process

CREATING CHILD DIAGRAM

Parent Diagram vs. Child Diagram



CREATING CHILD DIAGRAM - example

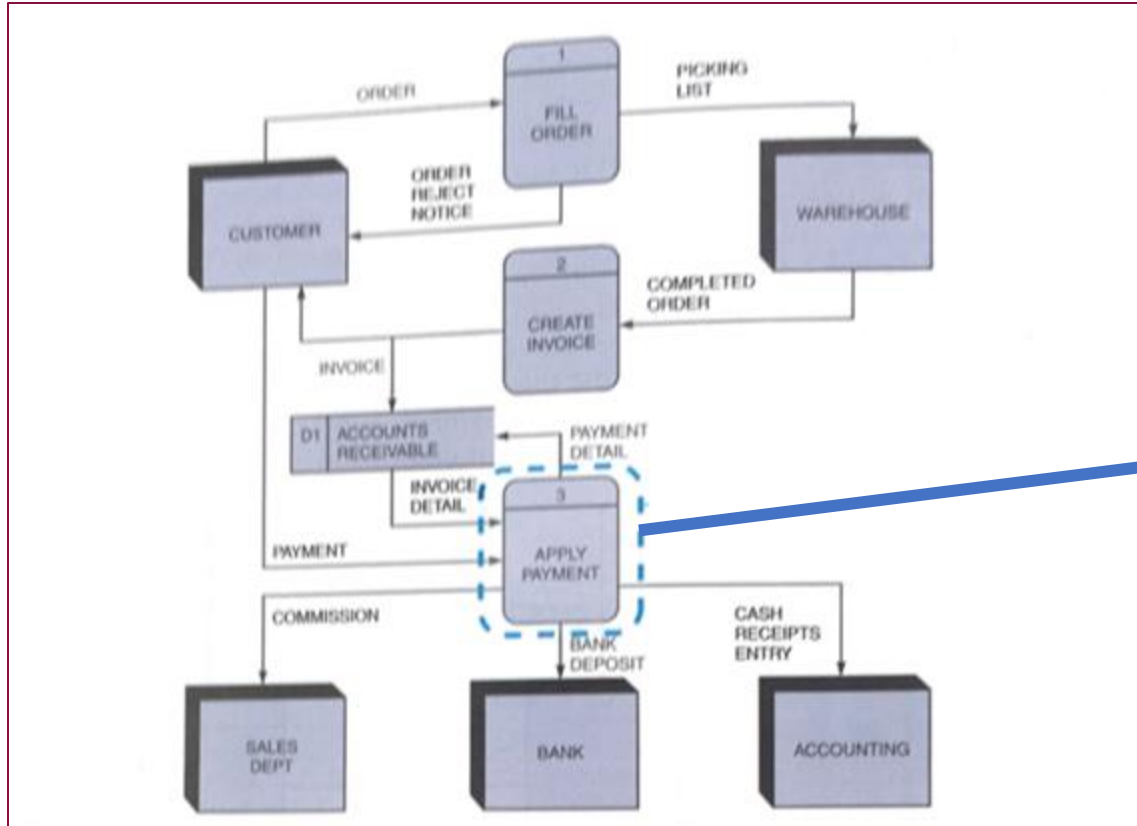
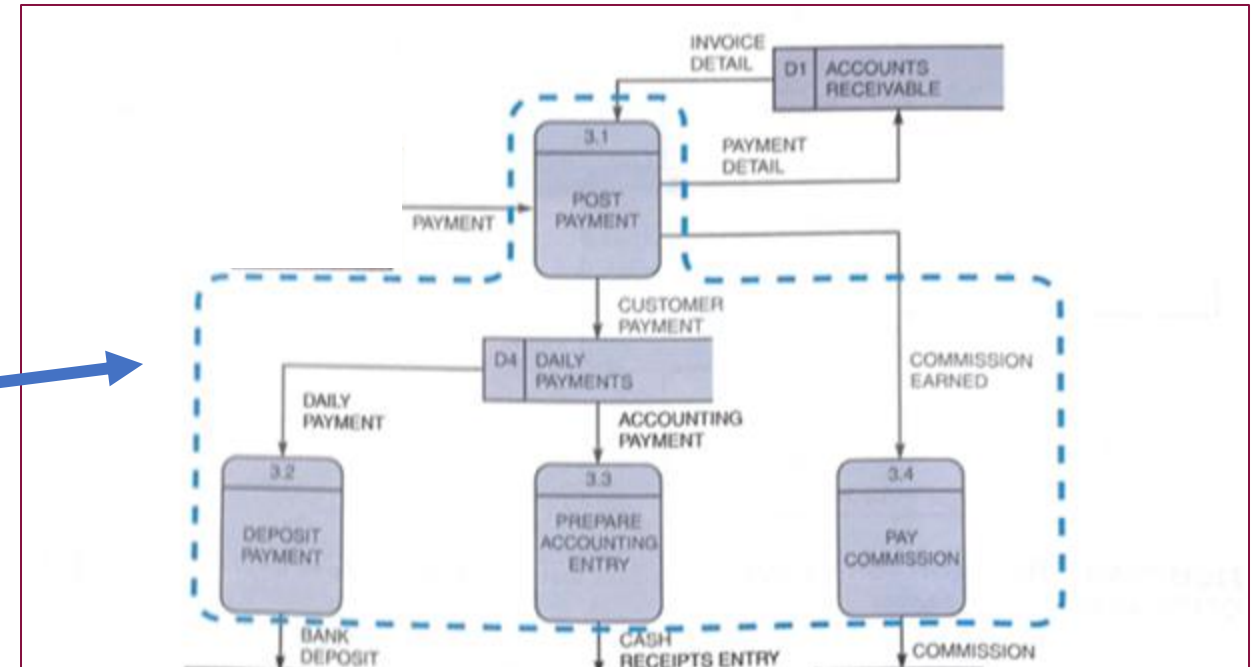


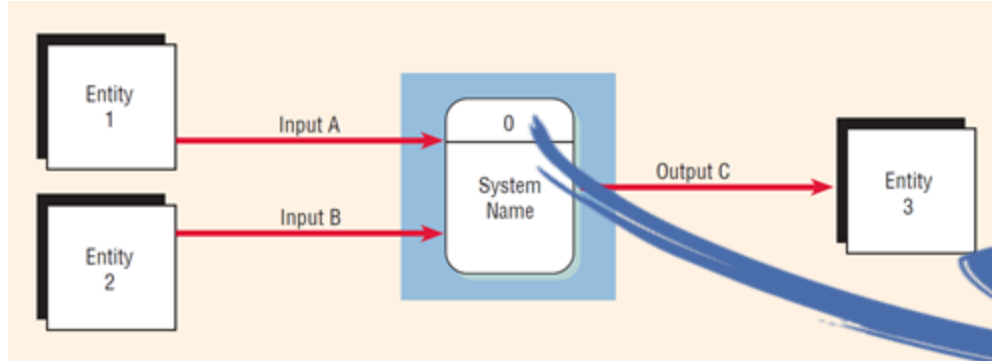
DIAGRAM 0 for ORDER SYSTEM
(Parent Diagram)



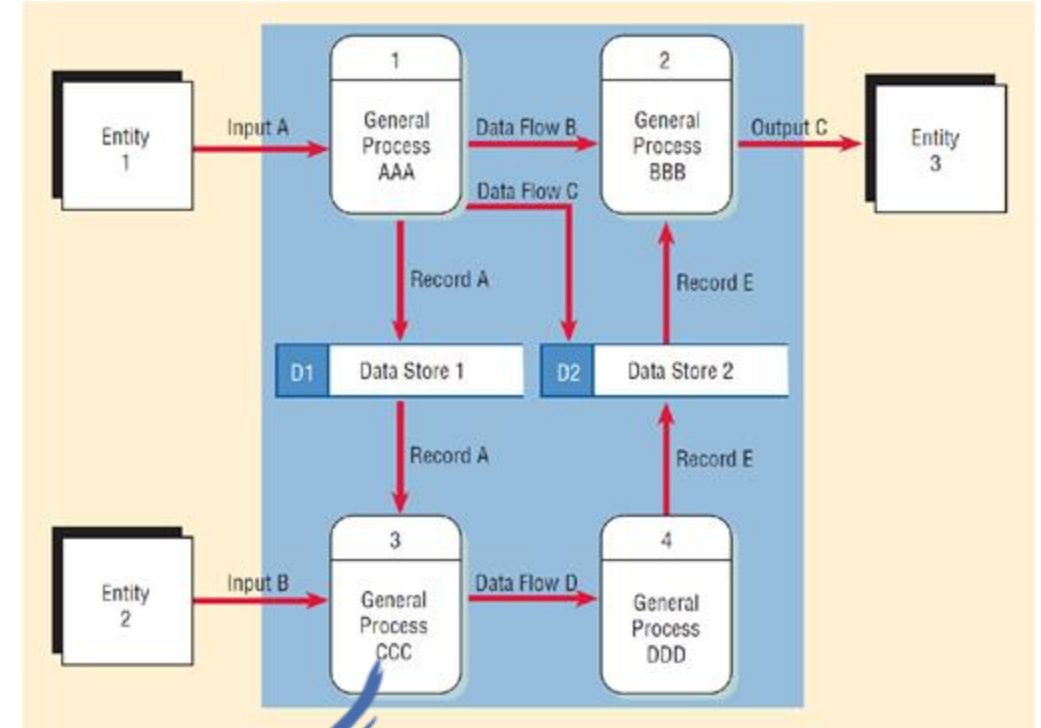
LEVEL 1 DIAGRAM for Process 3 APPLY PAYMENT
(Child Diagram)

WHAT'S IMPORTANT?

1. CONTEXT DIAGRAM



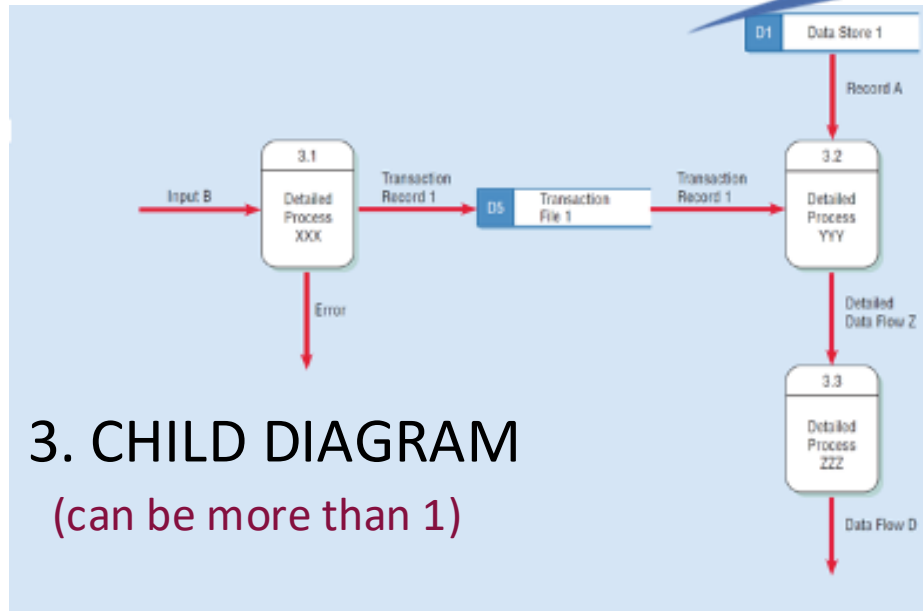
2. DIAGRAM 0



NOT TOO MANY PROCESSES!
 Some students do until 15-20 processes

3. CHILD DIAGRAM

(can be more than 1)


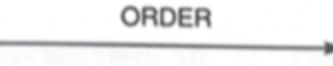

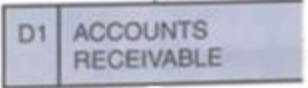


PART III

- DFD Errors

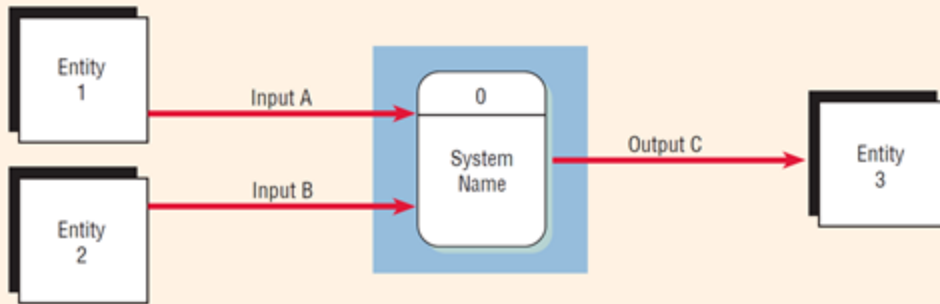
7 STEPS OF CREATING DFD

Step 1: List the ENTITY, DATA FLOW, PROCESS & DATA STORE

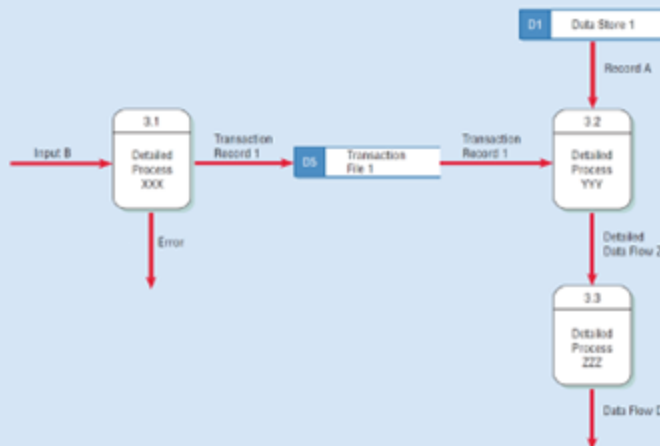
| THE TERMS | ENTITY | DATA FLOW | PROCESS | DATA STORE |
|------------------------------------|---|---|---|---|
| THE SYMBOLS |  |  |  |  |
| THE MEANING (what it represent) | Source or destination | Movement of data | Work to be done (performed) in the system | Depository of the data |
| THE NAMING (how to describe) | NOUN | NOUN | VERB - NOUN (except in Context Diagram) | NOUN |

7 STEPS OF CREATING DFD

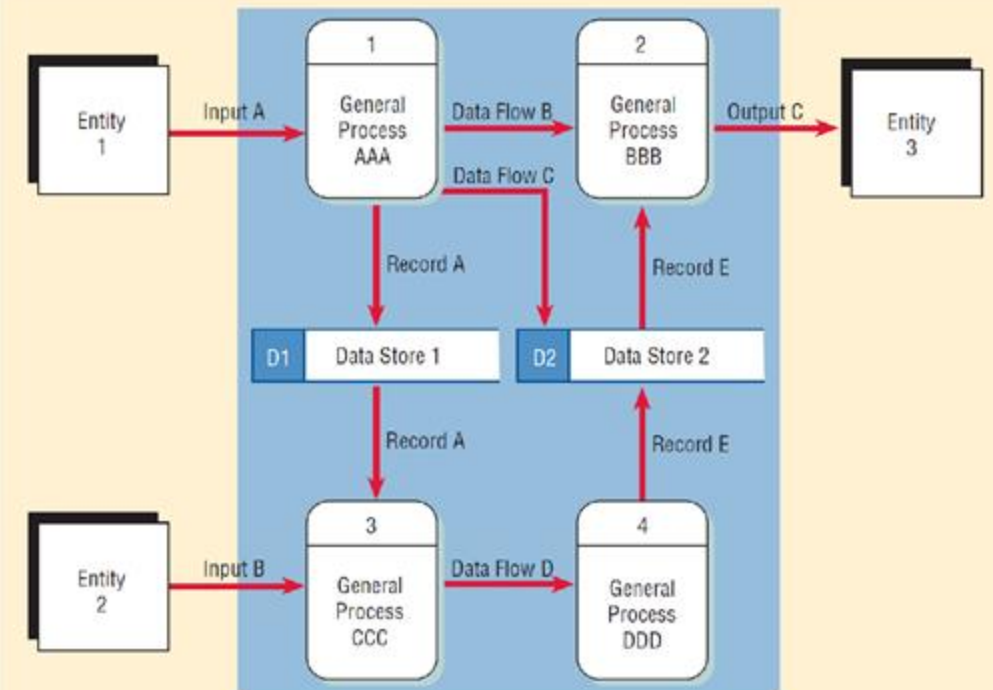
Step 2: CONTEXT DIAGRAM



Step 4: CHILD DIAGRAM



Step 3: DIAGRAM 0

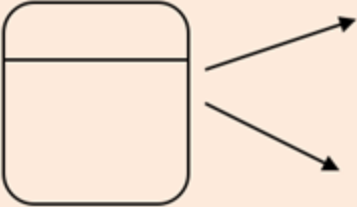
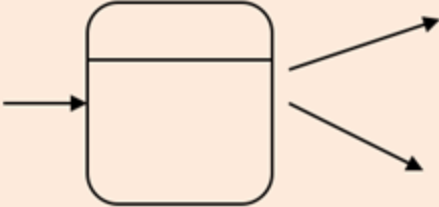
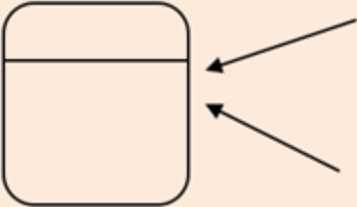
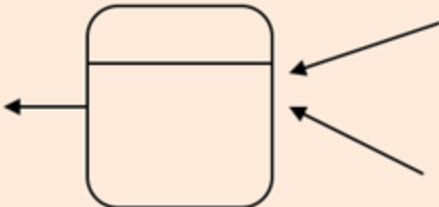
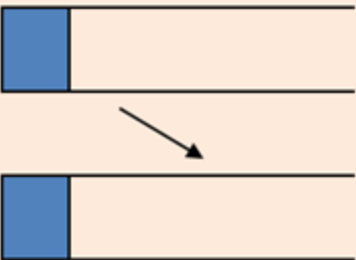
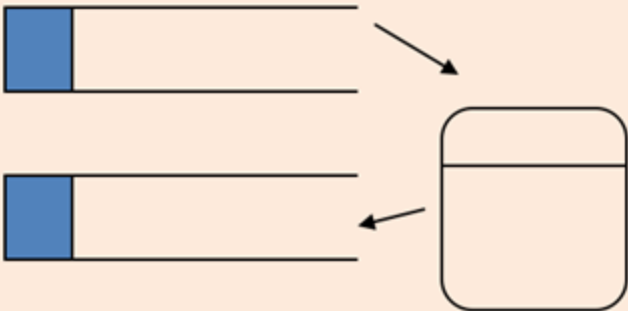


STEP 5: CHECK FOR ERRORS

■ DATA FLOW DIAGRAM ERRORS

- ✧ Forgetting to include a data flow or pointing an arrow in the wrong direction
- ✧ Connecting data stores and external entities directly to each other
- ✧ Incorrectly labeling processes or data flow
- ✧ Including more than nine processes on a data flow diagram
- ✧ Omitting data flow
- ✧ Creating unbalanced decomposition (or explosion) in child diagrams

DATA FLOW DIAGRAM ERRORS

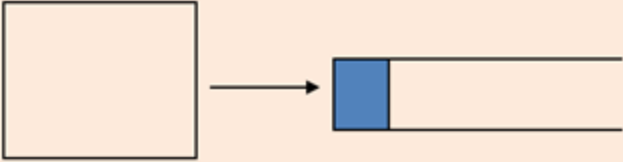
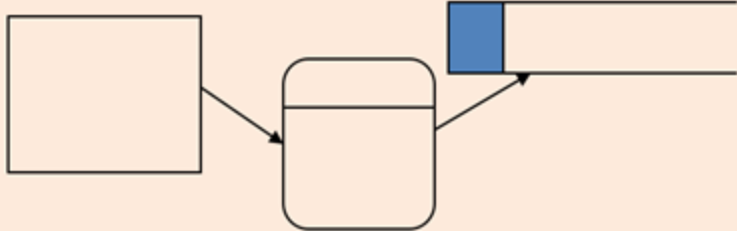
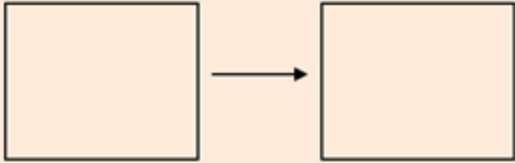
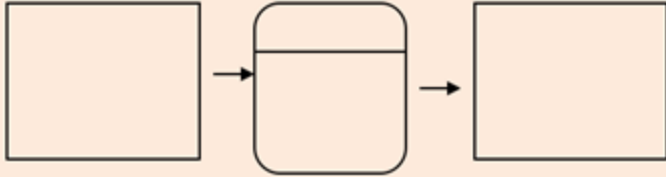

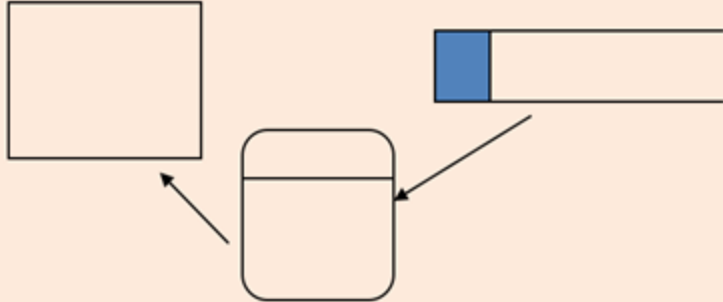
| Incorrect | Correct |
|--|---|
|  |  |
|  |  |
|  |  |

No process can have only output
(**MIRACLE**)

No process can have only input (**BLACK HOLE**)

Data cannot move directly from one data store to another data store – must be moved by a process that receives (input) and place it into another data store (output)

DATA FLOW DIAGRAM ERRORS

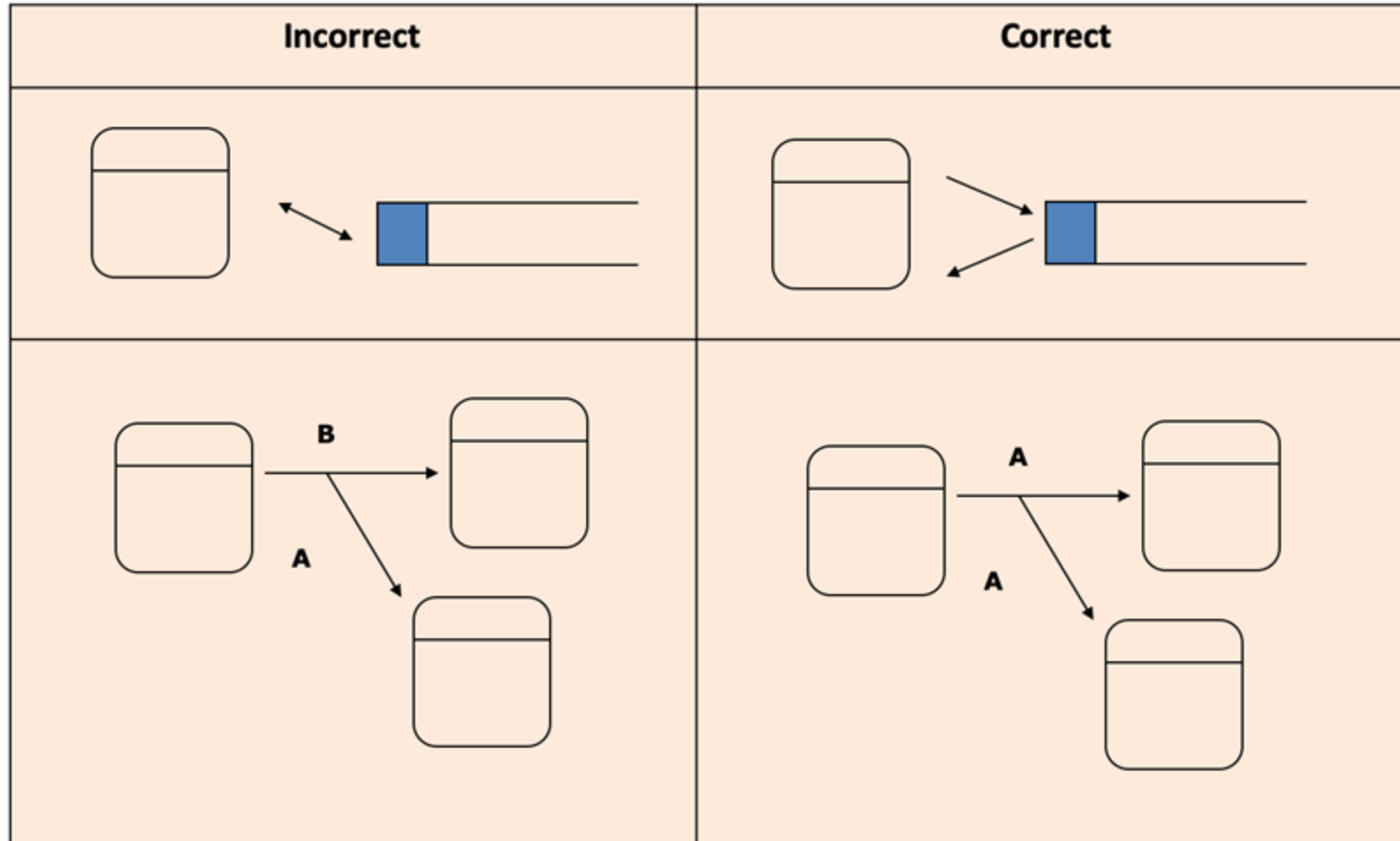
| Incorrect | Correct |
|---|---|
|  |  |
|  |  |
|  |  |

Data cannot move directly from an external entity to a data store – must be moved by a process that receives (input) from entity and place it into the data store (output)

Data cannot move directly from one entity to another entity – must be moved by a process that receives (input) from entity and place it into another entity (output)

Data cannot move directly from a data store to an external entity – must be moved by a process that receives (input) from data store and place it into the entity (output)

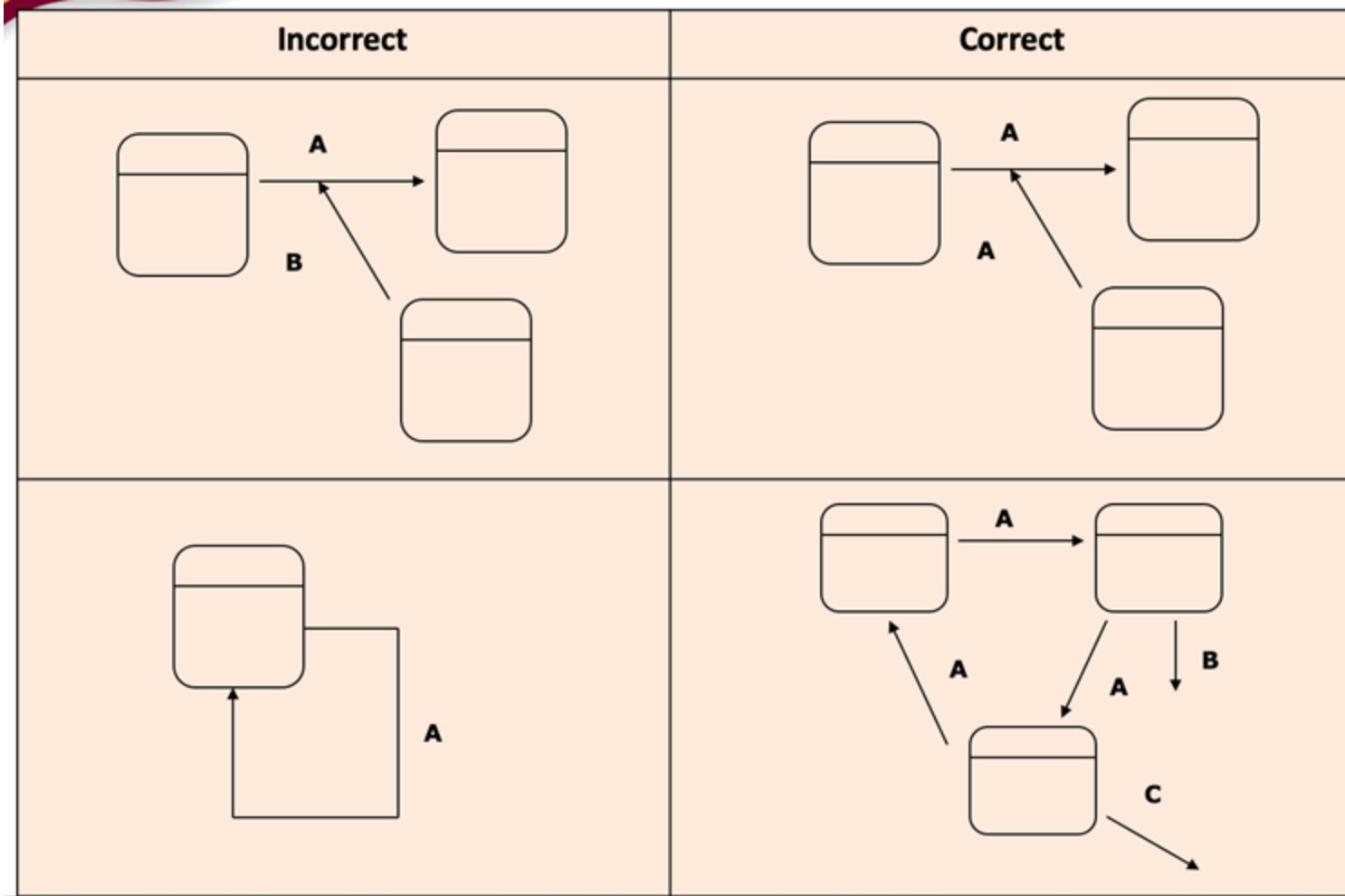
DATA FLOW DIAGRAM ERRORS



A data flow has only one direction between symbols. It may flow in both direction between process and a data store to show a READ before an UPDATE, thus two separate arrow is needed to show that the data flow occur at different time.

A **FORK** in a data flow means that exactly the same data goes from a common location to two or more different processes, data stores or sources. This indicate different copies of the same data goes to different places.

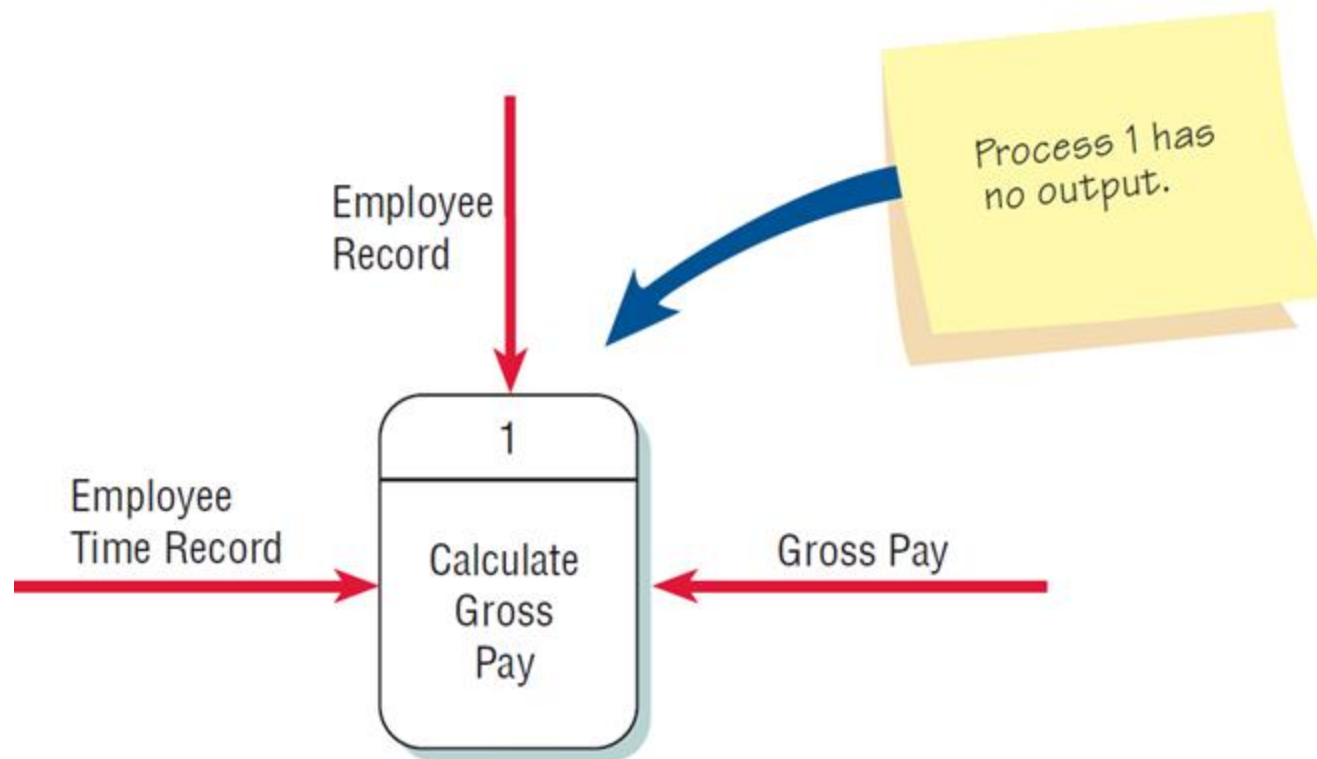
DATA FLOW DIAGRAM ERRORS



A **JOIN** in a data flow means that exactly the same data come from any of two or more different processes, data stores or sources.

A data flow **CANNOT** go directly back to the same process it leaves. There **MUST** be at least one other process that handle the data flow, produces some other data flow and returns original data flow to the beginning process.

CHECKING THE DIAGRAMS FOR ERRORS – Common Mistakes



ERROR?

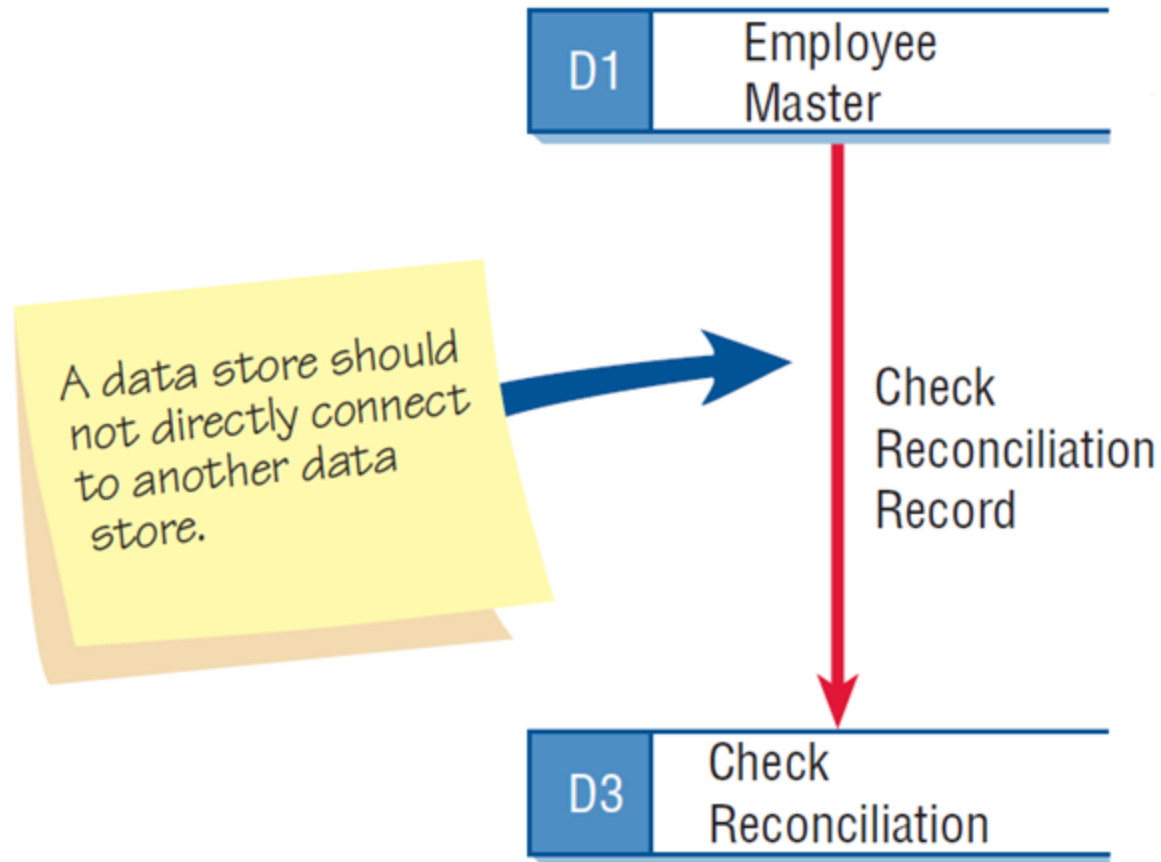
BLACK HOLE (process with input data flow ONLY – NO output flow)

HOW TO FIX IT?

Identify an output data flow.

Note: EVERY PROCESS MUST have input and output flow

CHECKING THE DIAGRAMS FOR ERRORS – Common Mistakes



ERROR?

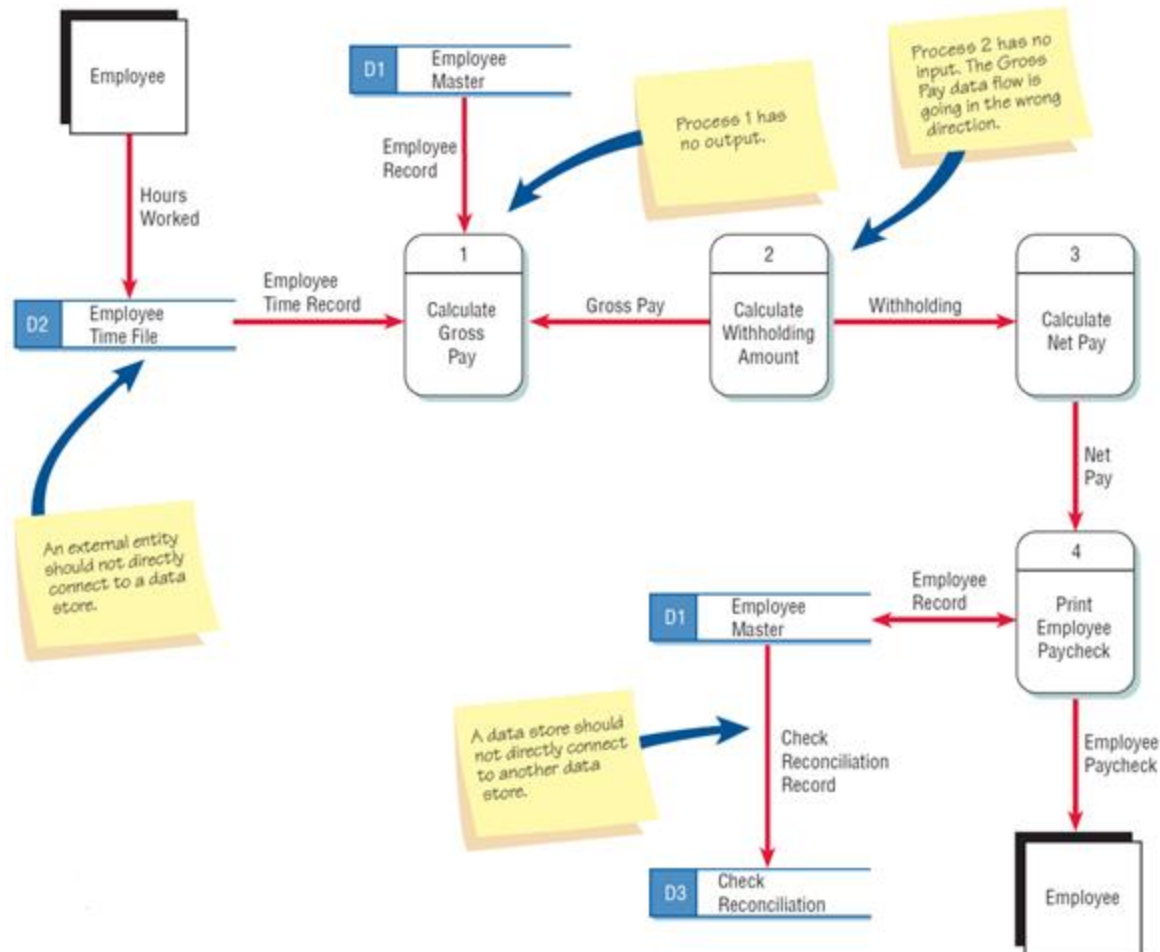
Connecting data stores to another data store directly to each other

HOW TO FIX IT?

Identify a PROCESS that receive the INPUT DATA FLOW from DATA STORE D1 and change it into an OUTPUT DATA FLOW to DATA STORE D2.

Note: Every DATA from DATA STORE must go through a PROCESS.

CHECKING THE DIAGRAMS FOR ERRORS – Common Mistakes



PART IV

From logical to physical DFD

■ FROM PART I to PART III

Expectation?

1. Should be able to understand a case study.
2. Able to identify ENTITY, DATA FLOW, PROCESS and DATA STORE
3. Understand the process/steps to draw a DATA FLOW DIAGRAM:
 - CONTEXT DIAGRAM
 - DIAGRAM 0
 - CHILD DIAGRAM
4. Able to identify DFD error (create an error-free DFD).

STEP 6: FROM LOGICAL DFD TO PHYSICAL DFD

■ TYPES OF DFDs

- **Current** - how data flows now
- **Proposed** - how we'd like it to flow
- **Logical** - the “essence” of a process
- **Physical** - the implementation of a process
- **Partitioned physical** - system architecture or high-level design

■ LOGICAL AND PHYSICAL DFDs

WHAT = LOGICAL DFD

- Focuses on the business and how the business operates
- Not concerned with how the system will be constructed
- Describes the business events that take place and the data required and produced by each event

WHAT + HOW = PHYSICAL DFD

- Shows how the system will be implemented
- Depicts the proposed system

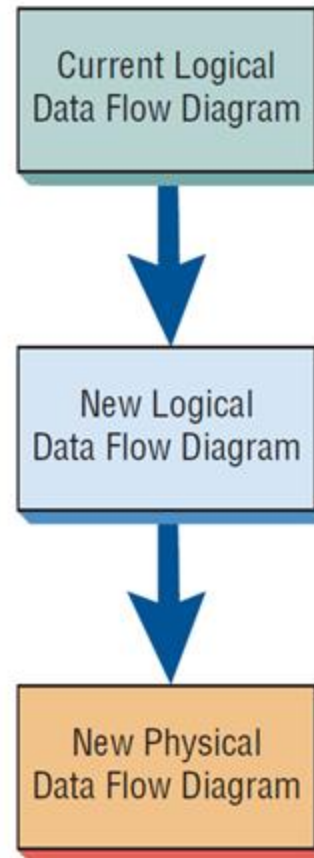
■ PHYSICAL DFDs

- Model the **implementation** of the system
- Start with a set of child diagrams or with Level 0 diagram
- Add implementation details
 - indicate manual vs. automated processes
 - describe form of data stores and data flows
 - extra processes for maintaining data

LOGICAL AND PHYSICAL DFDs – Common Features

| Design Feature | Logical | Physical |
|--------------------------------|--|---|
| What the model depicts | How the business operates. | How the system will be implemented (or how the current system operates). |
| What the processes represent | Business activities. | Programs, program modules, and manual procedures. |
| What the data stores represent | Collections of data regardless of how the data are stored. | Physical files and databases, manual files. |
| Type of data stores | Show data stores representing permanent data collections. | Master files, transition files. Any processes that operate at two different times must be connected by a data store. |
| System controls | Show business controls. | Show controls for validating input data, for obtaining a record (record found status), for ensuring successful completion of a process, and for system security (example: journal records). |

MODEL PROGRESSION FROM LOGICAL AND PHYSICAL DFDs

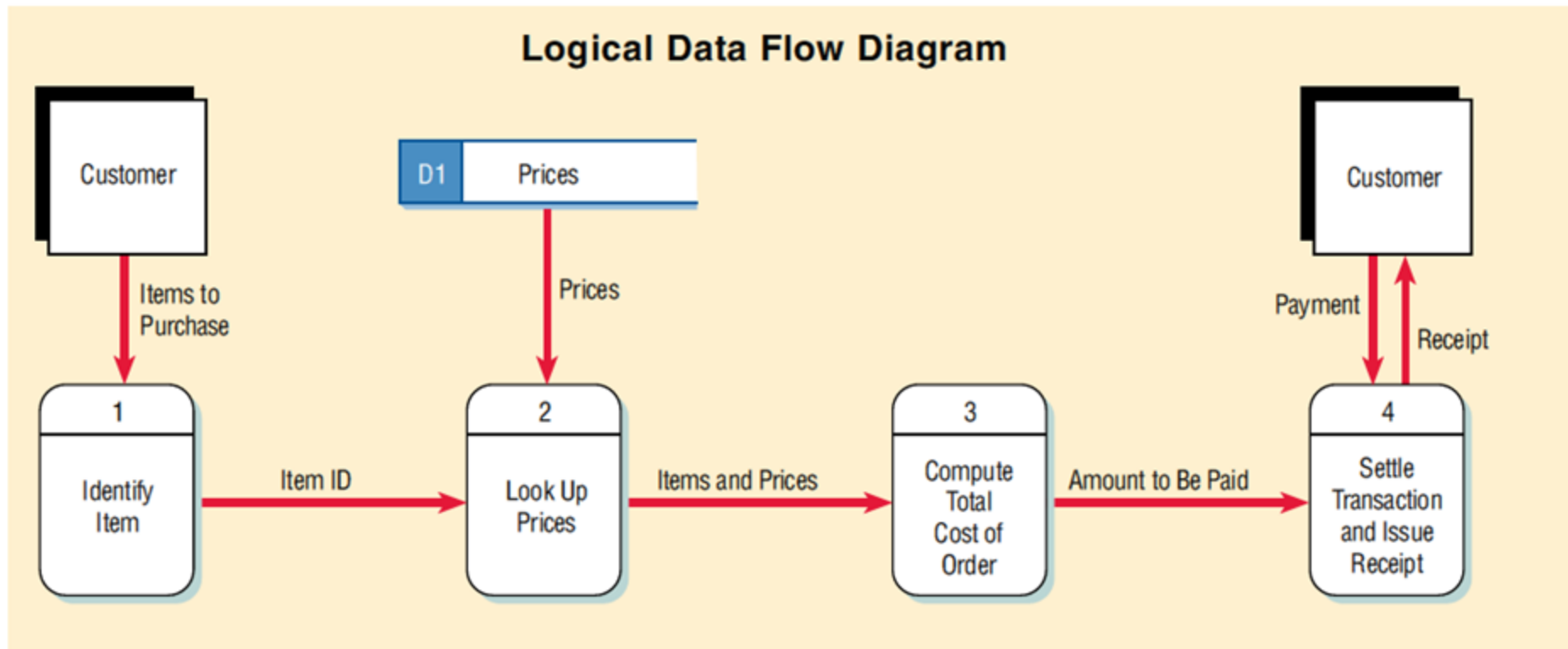


Derive the logical data flow diagram for the current system by examining the physical data flow diagram and isolating unique business activities.

Create the logical data flow diagram for the new system by adding the input, output, and processes required in the new system to the logical data flow diagram for the current system.

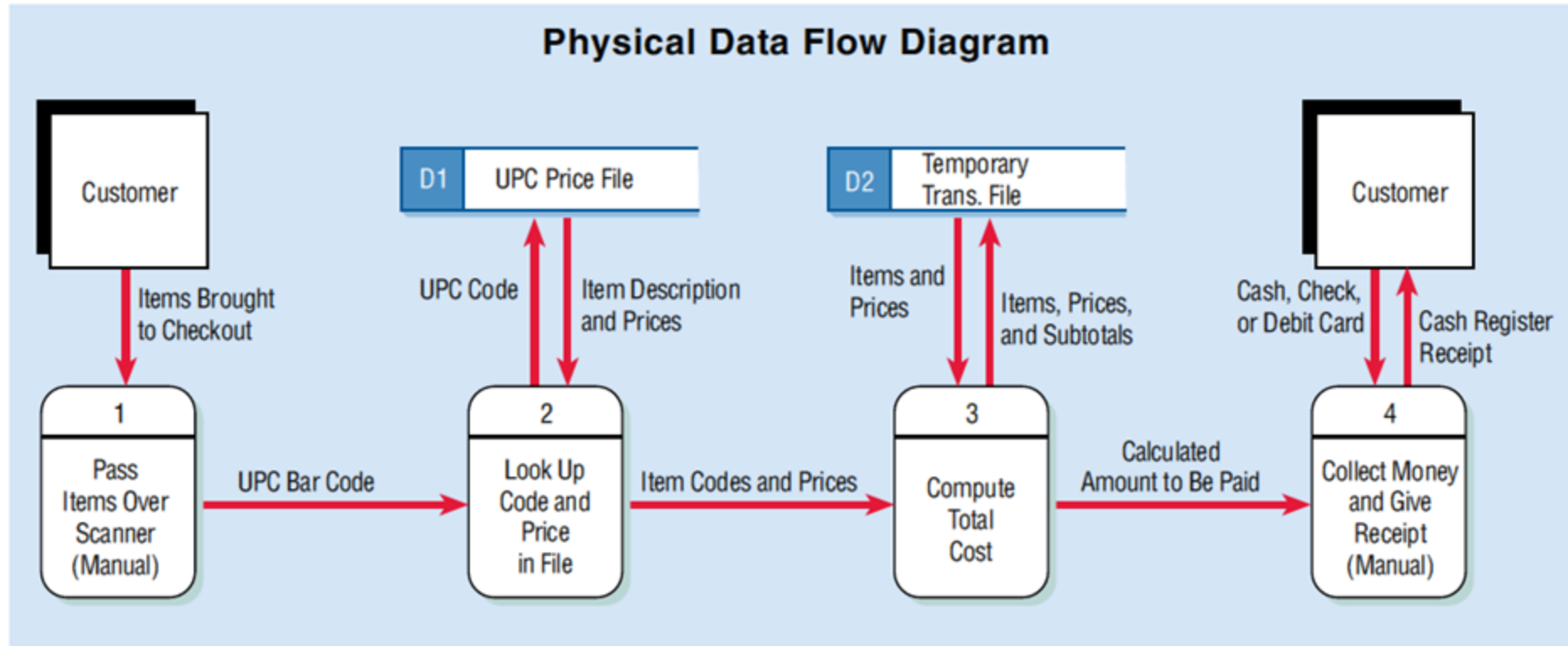
Derive the physical data flow diagram by examining processes on the new logical diagram. Determine where the user interfaces should exist, the nature of the processes, and necessary data stores.

LOGICAL DFD EXAMPLE



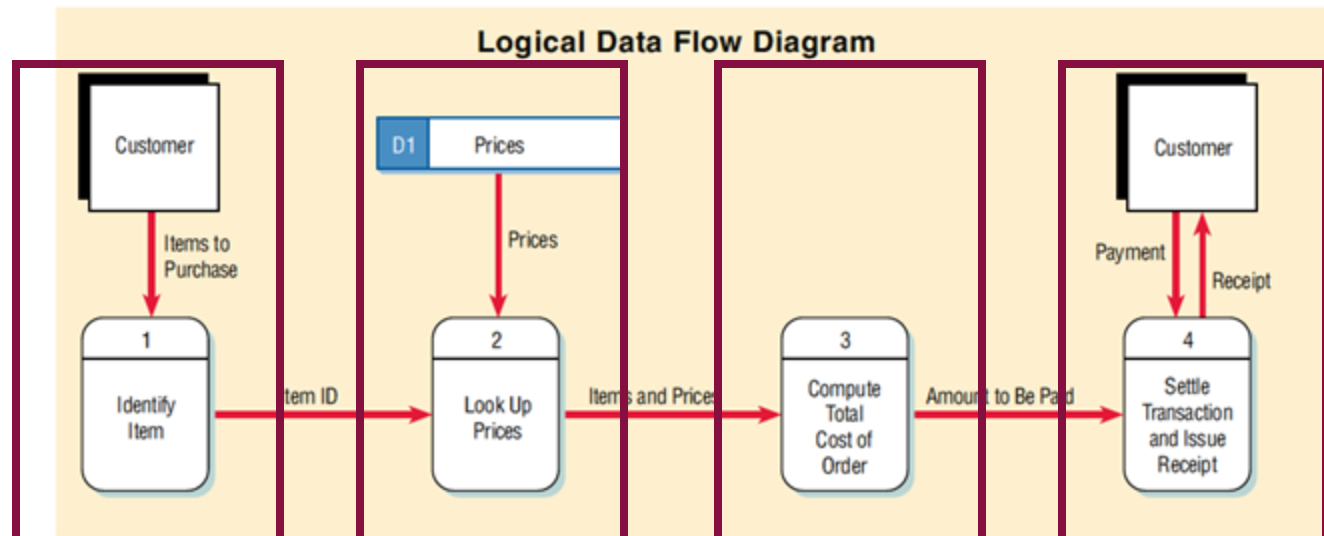
The logical DFD illustrates the processes involved without going into detail about the physical implementation of activities.

PHYSICAL DFD EXAMPLE

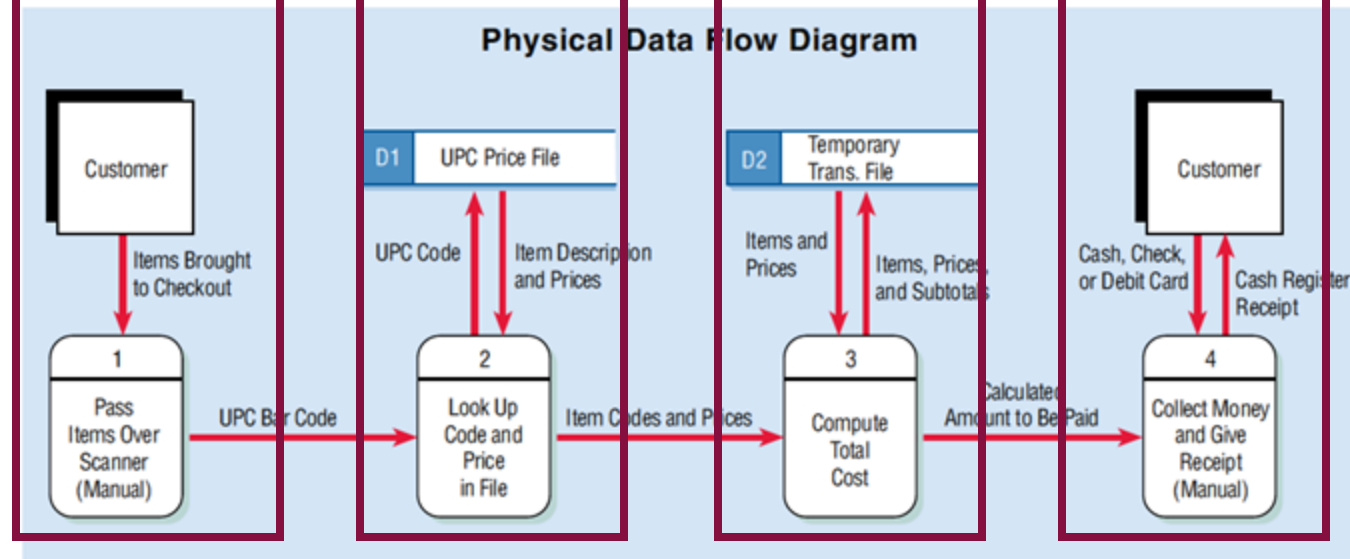


The physical DFD shows that processes involved with going into detail about the physical aspects of the activities

LOGICAL DFD



PHYSICAL DFD



DEVELOPING LOGICAL AND PHYSICAL DFDs

LOGICAL DFD

- Better communication with users
- More stable systems
- Better understanding of the business by analysts
- Flexibility and maintenance
- Elimination of redundancy and easier creation of the physical model

PHYSICAL DFD

- Clarifying which processes are performed by humans and which are automated
- Describing processes in more detail
- Sequencing processes that must be done in a particular order
- Identifying temporary data stores
- Specifying actual names of files and printouts
- Adding controls to ensure the processes are done properly

■ DEVELOPING LOGICAL AND PHYSICAL DFDs

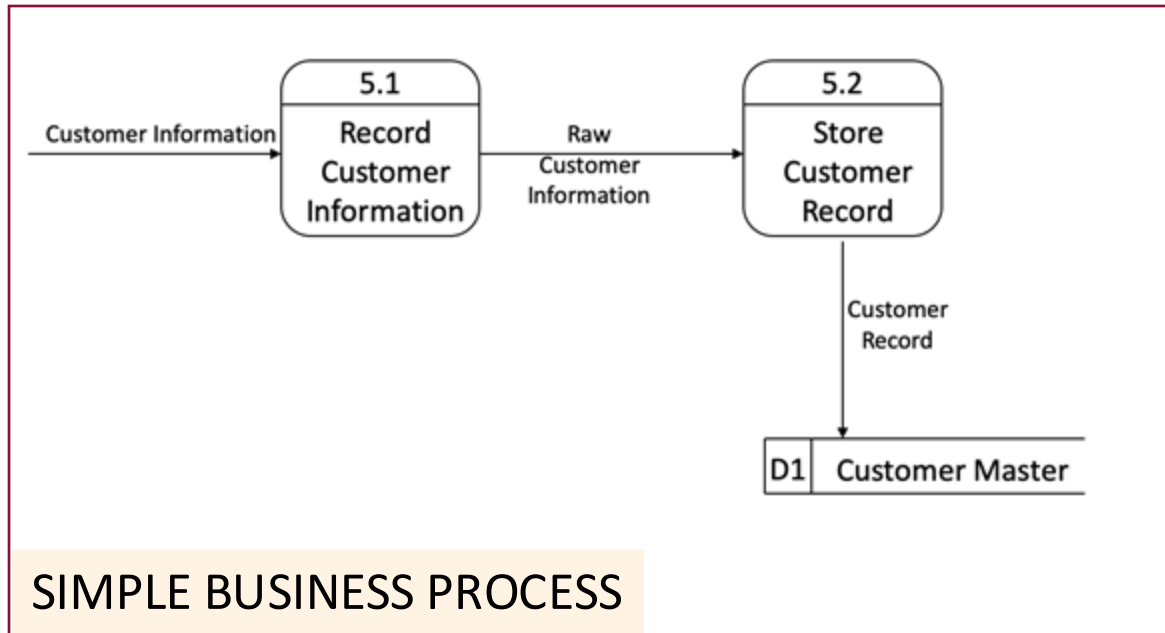
Physical DFD Contain Many Items Not Found in Logical DFD

Contents of Physical Data Flow Diagrams

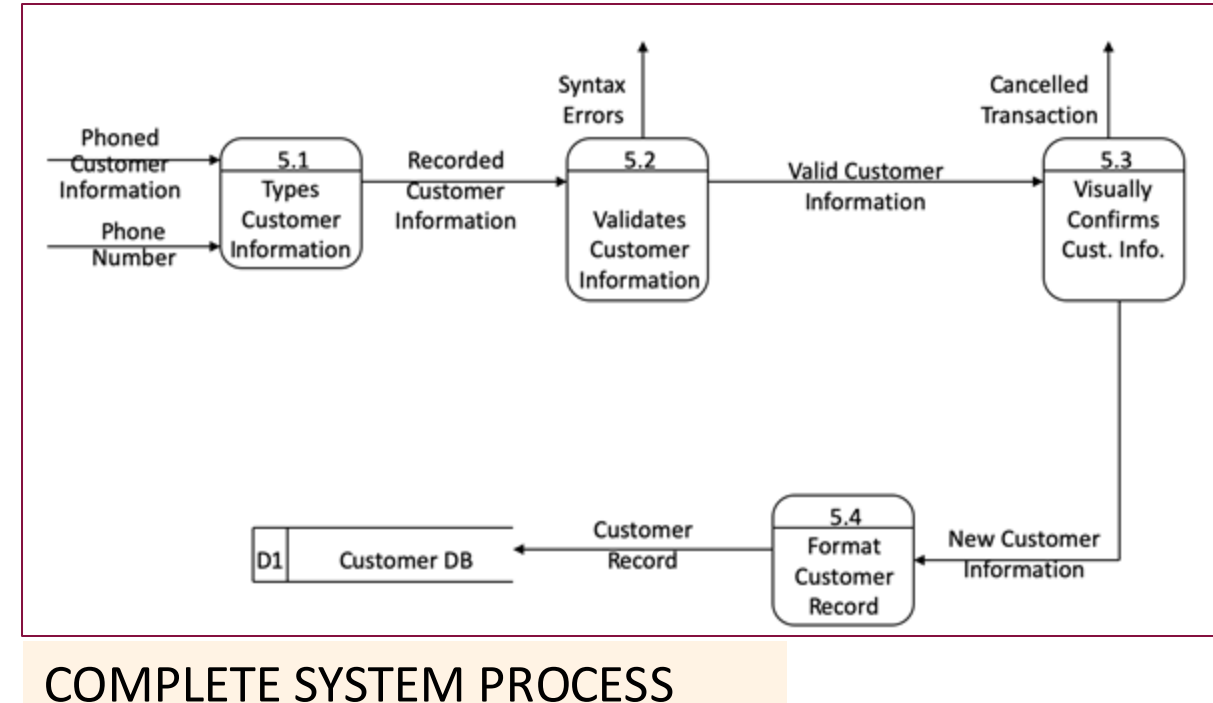
- Manual processes
- Processes for adding, deleting, changing, and updating records
- Data entry and verifying processes
- Validation processes for ensuring accurate data input
- Sequencing processes to rearrange the order of records
- Processes to produce every unique system output
- Intermediate data stores
- Actual file names used to store data
- Controls to signify completion of tasks or error conditions

DEVELOPING LOGICAL AND PHYSICAL DFDs: Example

Current Logical Child Diagram



Physical Child Diagram

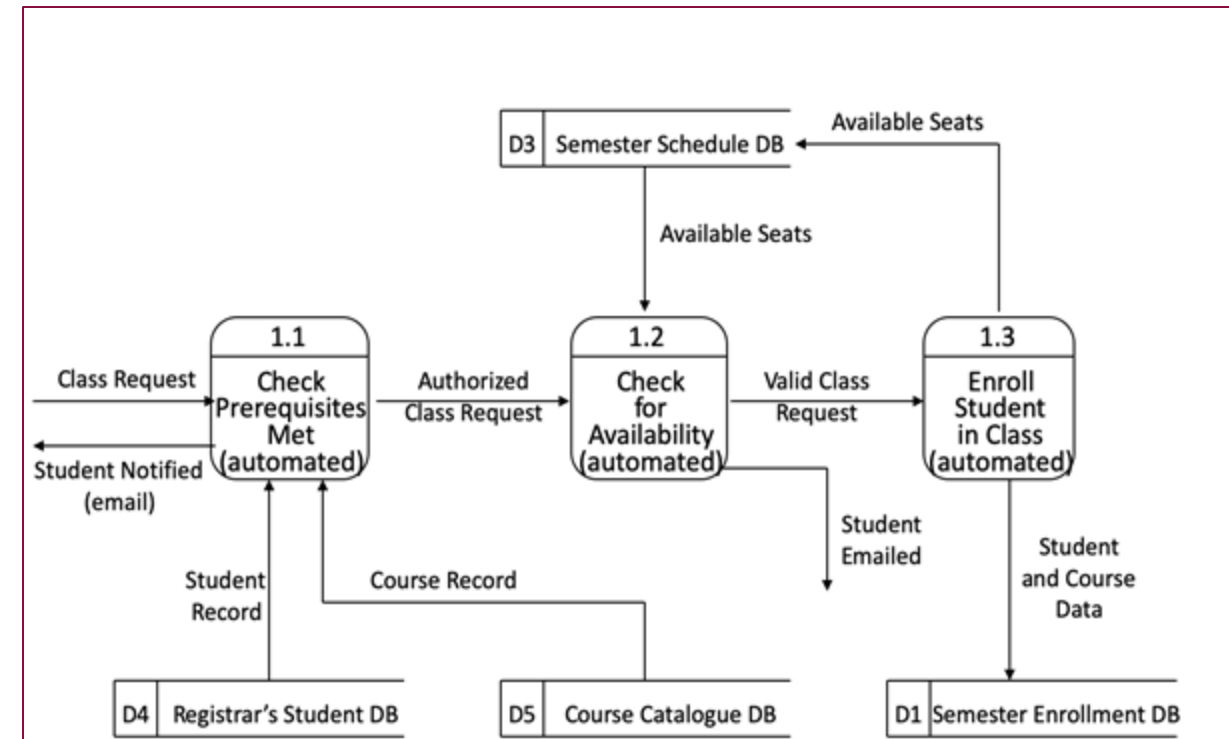
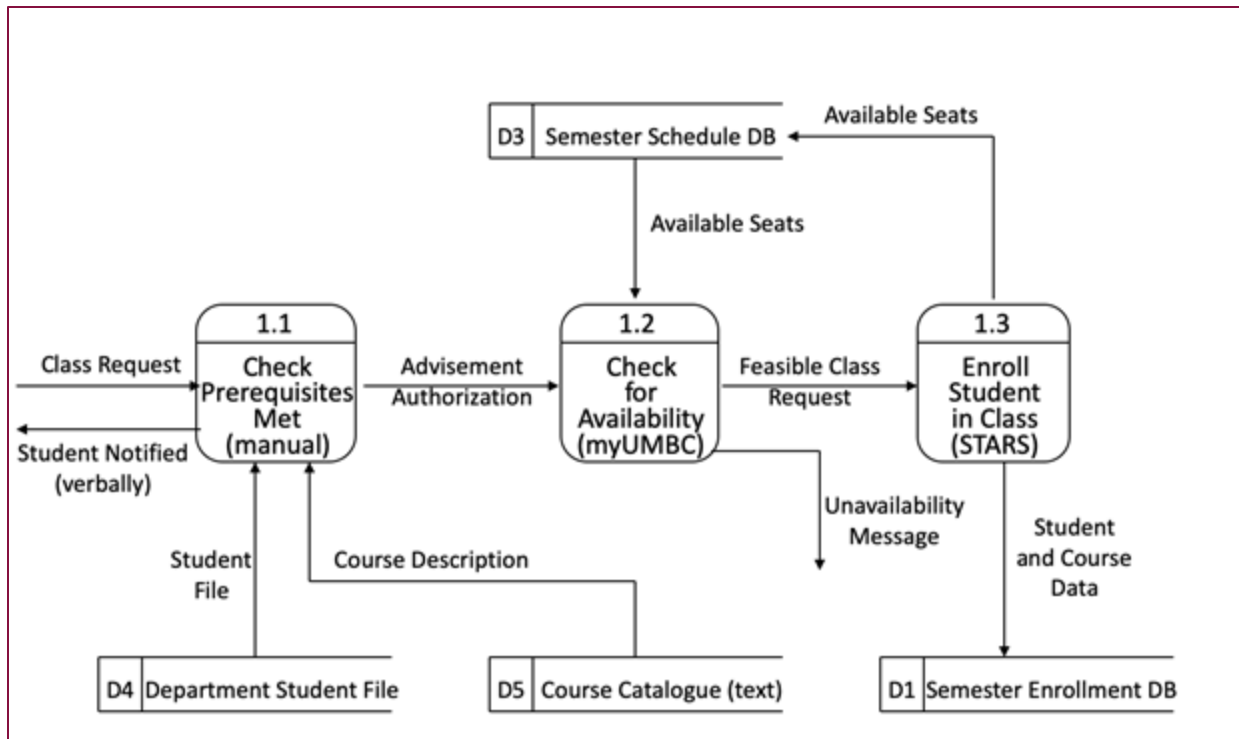


■ COMPARING CURRENT AND PROPOSED PHYSICAL DFDs: Example

Current Physical Child Diagram



Proposed Physical Child Diagram



PART V

CRUD matrix

FROM PART IV

Expectation?

NOTE:

PART I – Step 1 (Identify ENTITY, DATA FLOW, PROCESS & DATA STORE)

PART II – Step 2 (CONTEXT DIAGRAM), Step 3 (DIAGRAM 0) & Step 4 (CHILD DIAGRAM)

PART III – Step 5 (Identifying DFD Errors)

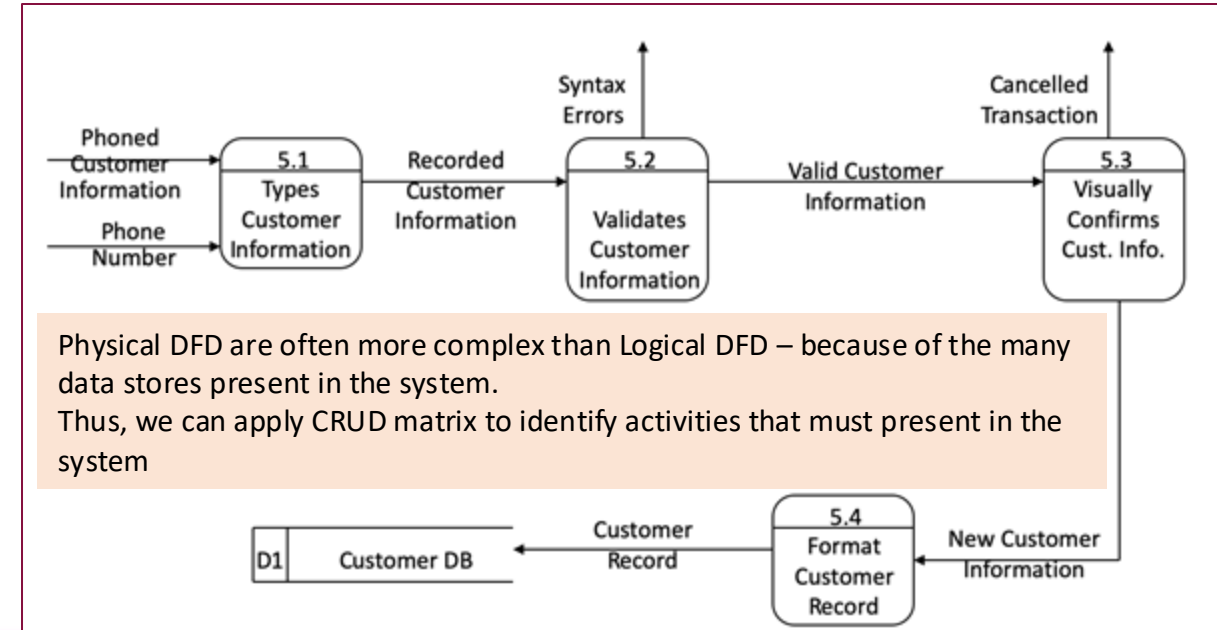
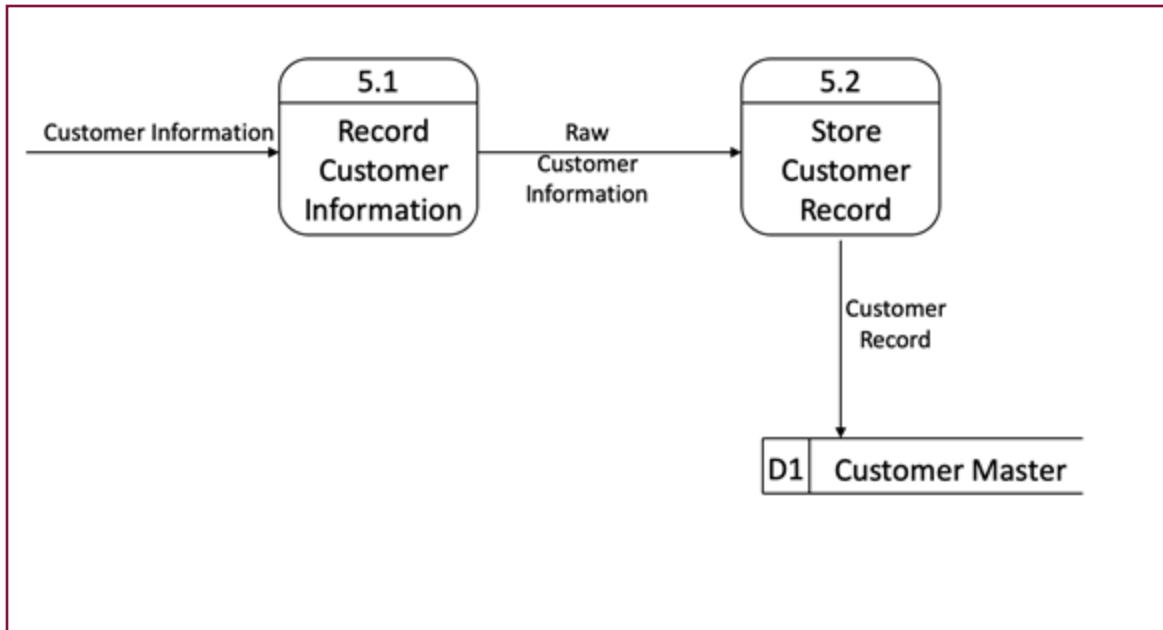
PART IV – Step 6 (From Logical to Physical DFD)

Understand the difference between LOGICAL and PHYSICAL DFD

Current Logical Child Diagram



Physical Child Diagram



■ CRUD MATRIX

- The acronym CRUD is often used for
 - **C**reate
 - **R**ead
 - **U**ppdate
 - **D**eflete
- These are the activities that must be present in a system for each master file (represented as in data store)
- A CRUD matrix is a tool to represent where each of these processes occurs in a system

■ CRUD MATRIX

| Activity | Customer | Item | Order | Order Detail |
|------------------------------|----------|------|-------|--------------|
| Customer Logon | R | | | |
| Item Inquiry | | R | | |
| Item Selection | | R | C | C |
| Order Checkout | U | U | U | R |
| Add Account | C | | | |
| Add Item | | C | | |
| Close Customer Account | D | | | |
| Remove Obsolete Item | | D | | |
| Change Customer Demographics | RU | | | |
| Change Customer Order | RU | RU | RU | CRUD |
| Order Inquiry | R | R | R | R |

■ EVENT MODELING AND DFD

- An input flow from an external entity is sometimes called a trigger because it starts the activities of a process
- Events cause the system to do something and act as a trigger to the system
- An approach to creating physical data flow diagrams is to create a data flow diagram fragment for each unique system event

■ EVENT RESPONSE TABLE

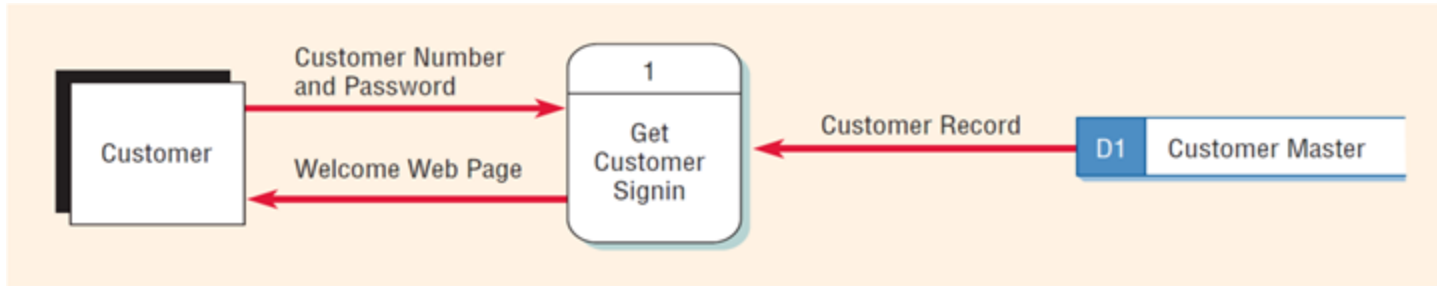
- An event table is used to create a data flow diagram by analyzing each event and the data used and produced by the event
- Every row in an event table represents a data flow diagram fragment and is used to create a single process on a data flow diagram

EVENT RESPONSE TABLE

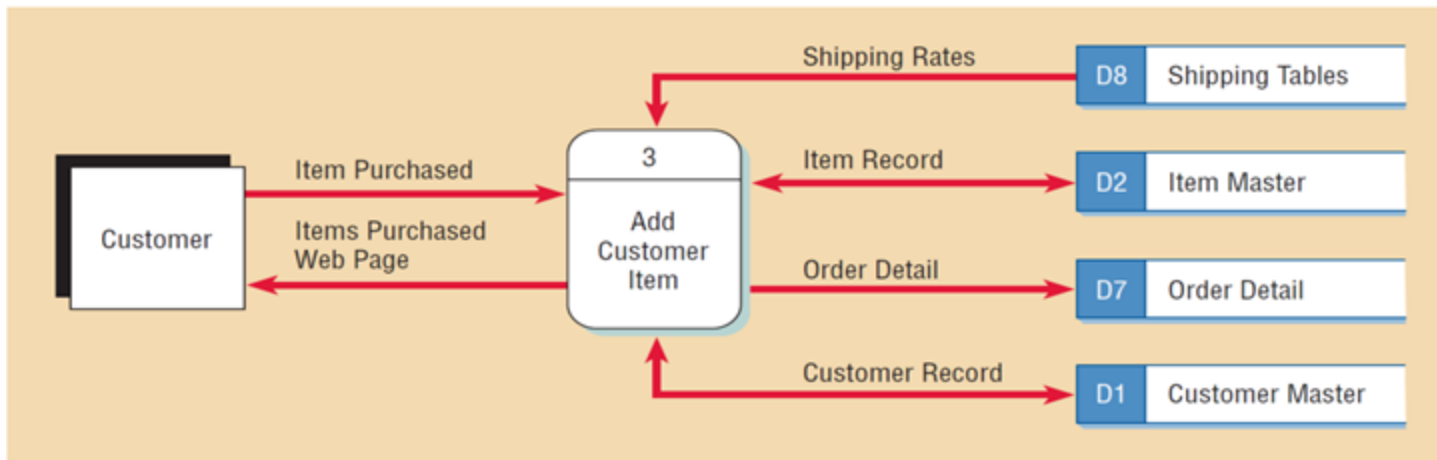
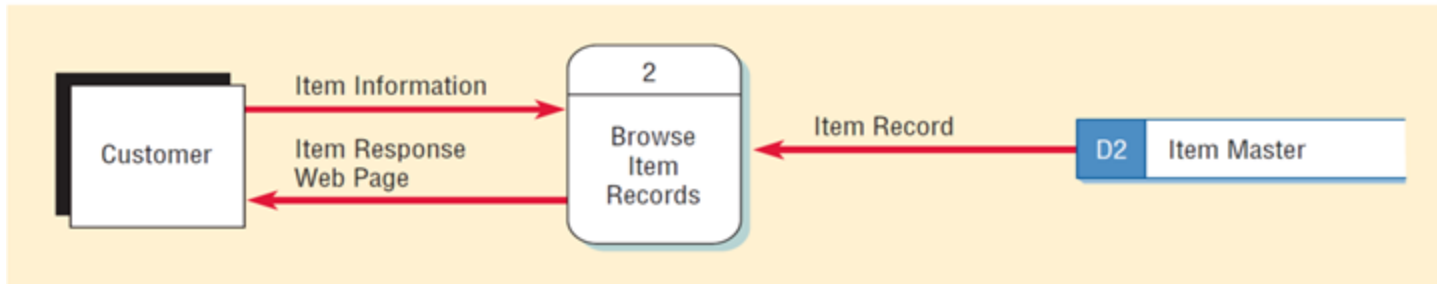
| Event | Source | Trigger | Activity | Response | Destination |
|---|----------|--|---|---------------------------------------|---------------------------------|
| Customer logs on | Customer | Customer number and password | Find customer record and verify password. Send Welcome Web page. | Welcome Web page | Customer |
| Customer browses items at Web storefront | Customer | Item information | Find item price and quantity available. Send Item Response Web page. | Item Response Web page | Customer |
| Customer places item into shopping basket at Web storefront | Customer | Item purchase (item number and quantity) | Store data on Order Detail Record. Calculate shipping cost using shipping tables. Update customer total. Update item quantity on hand. | Items Purchased Web page | Customer |
| Customer checks out | Customer | Clicks "Check Out" button on Web page | Display Customer Order Web page. | Verification Web page | |
| Obtain customer payment | Customer | Credit card information | Verify credit card amount with credit card company. Send. | Credit card data Customer feedback | Credit card company Customer |
| Send customer email | | Temporal, hourly | Send customer an email confirming shipment. | | Customer |

Example Internet Storefront Event Response Table

■ EVENT RESPONSE TABLE ? DFD

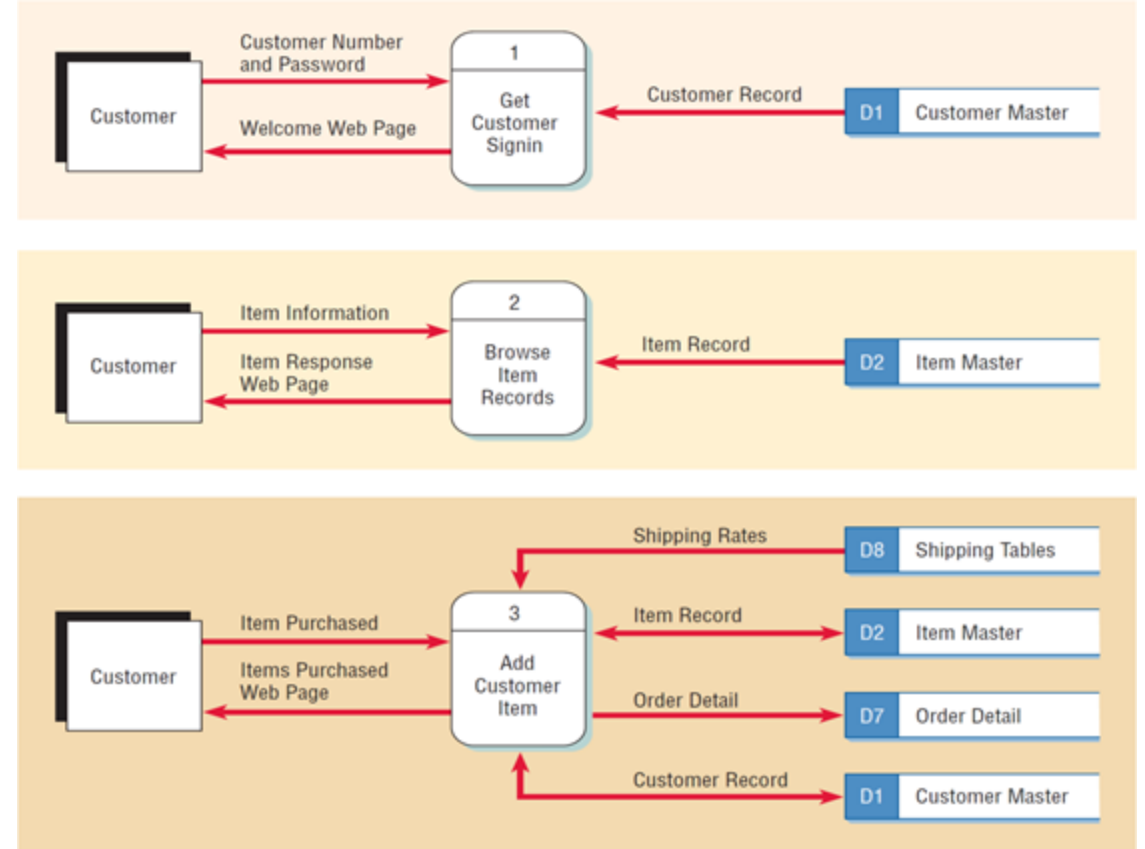


DFD of the Internet Storefront Event Response Table (from previous slide)



■ EVENT RESPONSE TABLE ? DFD

| Event | Source | Trigger | Activity | Response | Destination |
|---|----------|--|---|---------------------------------------|---------------------------------|
| Customer logs on | Customer | Customer number and password | Find customer record and verify password. Send Welcome Web page. | Welcome Web page | Customer |
| Customer browses items at Web storefront | Customer | Item information | Find item price and quantity available. Send Item Response Web page. | Item Response Web page | Customer |
| Customer places item into shopping basket at Web storefront | Customer | Item purchase (item number and quantity) | Store data on Order Detail Record. Calculate shipping cost using shipping tables. Update customer total. Update item quantity on hand. | Items Purchased Web page | Customer |
| Customer checks out | Customer | Clicks "Check Out" button on Web page | Display Customer Order Web page. | Verification Web page | Customer |
| Obtain customer payment | Customer | Credit card information | Verify credit card amount with credit card company. Send. | Credit card data Customer feedback | Credit card company Customer |
| Send customer email | | Temporal, hourly | Send customer an email confirming shipment. | | Customer |



WHAT'S IMPORTANT?

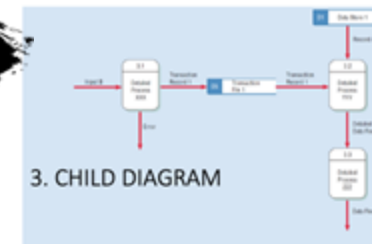
| Activity | Customer | Item | Order | Order Detail |
|------------------------------|----------|------|-------|--------------|
| Customer Login | R | | | |
| Item Inquiry | | R | | |
| Item Selection | | R | C | C |
| Order Checkout | U | U | U | R |
| Add Account | C | | | |
| Add Item | | C | | |
| Close Customer Account | D | | | |
| Remove Obsolete Item | | D | | |
| Change Customer Demographics | RU | | | |
| Change Customer Order | RU | RU | RU | CRUD |
| Order Inquiry | R | R | R | R |

CRUD
MATRIX

| Event | Source | Trigger | Activity | Response | Destination |
|---|----------|--|--|---------------------------------------|---------------------------------|
| Customer logs on | Customer | Customer number and password | Find customer record and verify password. Send Welcome Web page. | Welcome Web page | Customer |
| Customer browses items at Web storefront | Customer | Item information | Find item price and quantity available. Send item Response Web page. | Item Response Web page | Customer |
| Customer places item into shopping basket at Web storefront | Customer | Item purchase (item number and quantity) | Store data on Order Detail Record. Calculate shipping cost using shipping tables. Update customer total. Update item quantity on hand. | Items Purchased Web page | Customer |
| Customer checks out | Customer | Clicks "Check Out" button on Web page | Display Customer Order Web page. | Verification Web page | |
| Obtain customer payment | Customer | Credit card information | Verify credit card amount with credit card company. Send. | Credit card data Customer feedback | Credit card company Customer |
| Send customer email | | Temporal, hourly | Send customer an email confirming shipment. | | Customer |

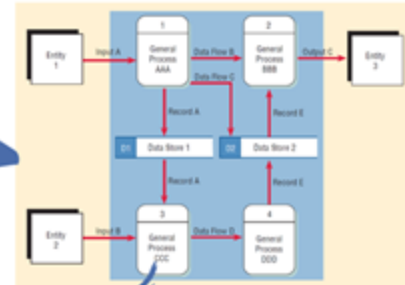
EVENT RESPONSE TABLE

1. CONTEXT DIAGRAM



DATA FLOW
DIAGRAM

2. DIAGRAM 0



PART VI

Partitioning DFD

FROM PART IV

Expectation? EVERYTHING in PART I – PART V

Case Study

Ria Pizza is a popular pizza restaurant in UTM. Most of its customers are the UTM students. In the current manual order processing system, employees have to go through three different activities in order to process a pizza order. The first activity, called Receiving the order, consists in getting customers' information (such as name, phone number, and address), and getting orders' information (such as the pizza size, the type of crust, the toppings, and the ingredients needed to make the pizza). The same process also checks the availability of the ingredients and sets the status of the pizza order as "valid". When a pizza order is valid, the employee in charge of Receiving the order provides the valid order information to the cooks who make the pizza. The second activity, Making the pizza, consists in getting the ingredients needed (from containers in the kitchen) and actually cooking the pizza. At the end of that process, the cooks have to update the manual file that keeps track of the ingredients. They then provide the drivers in charge of delivering the pizza with the (completed) status of the order. The final activity, Delivering the pizza, done by the drivers consists in getting address and payment information from the employee in charge of Receiving the order, and actually delivering the pizza. The drivers provide the customers with a receipt and get the payment.

CONTEXT DIAGRAM

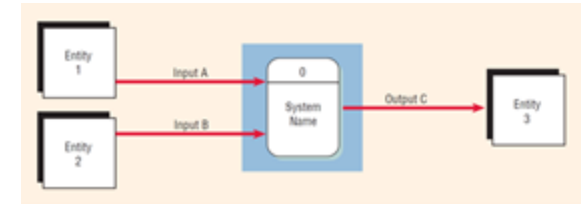
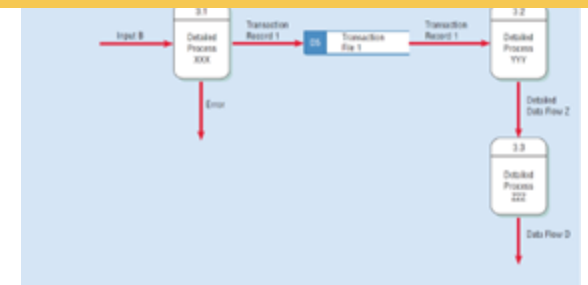


DIAGRAM 0



ERROR FREE DIAGRAM

CHILD DIAGRAM



Project

STEP 7: PARTITIONING THE PHYSICAL DFD

■ PARTITIONING DATA FLOW DIAGRAM

- Partitioning is the process of examining a data flow diagram and determining how it should be divided into collections of manual procedures and computer programs
- Decide what processes should be grouped together in the system components
- A dashed line is drawn around a process or group of processes that should be placed in a single computer program
- Reasons for partitioning
 - Different user groups
 - Timing
 - Similar tasks
 - Efficiency
 - Consistency of data
 - Security

FIGURE 7.20
Partitioning the data flow diagram
(showing part of Diagram 6)

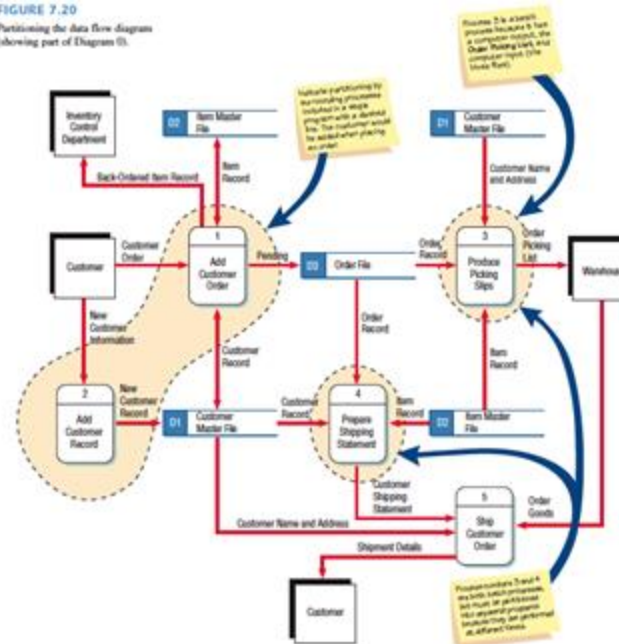
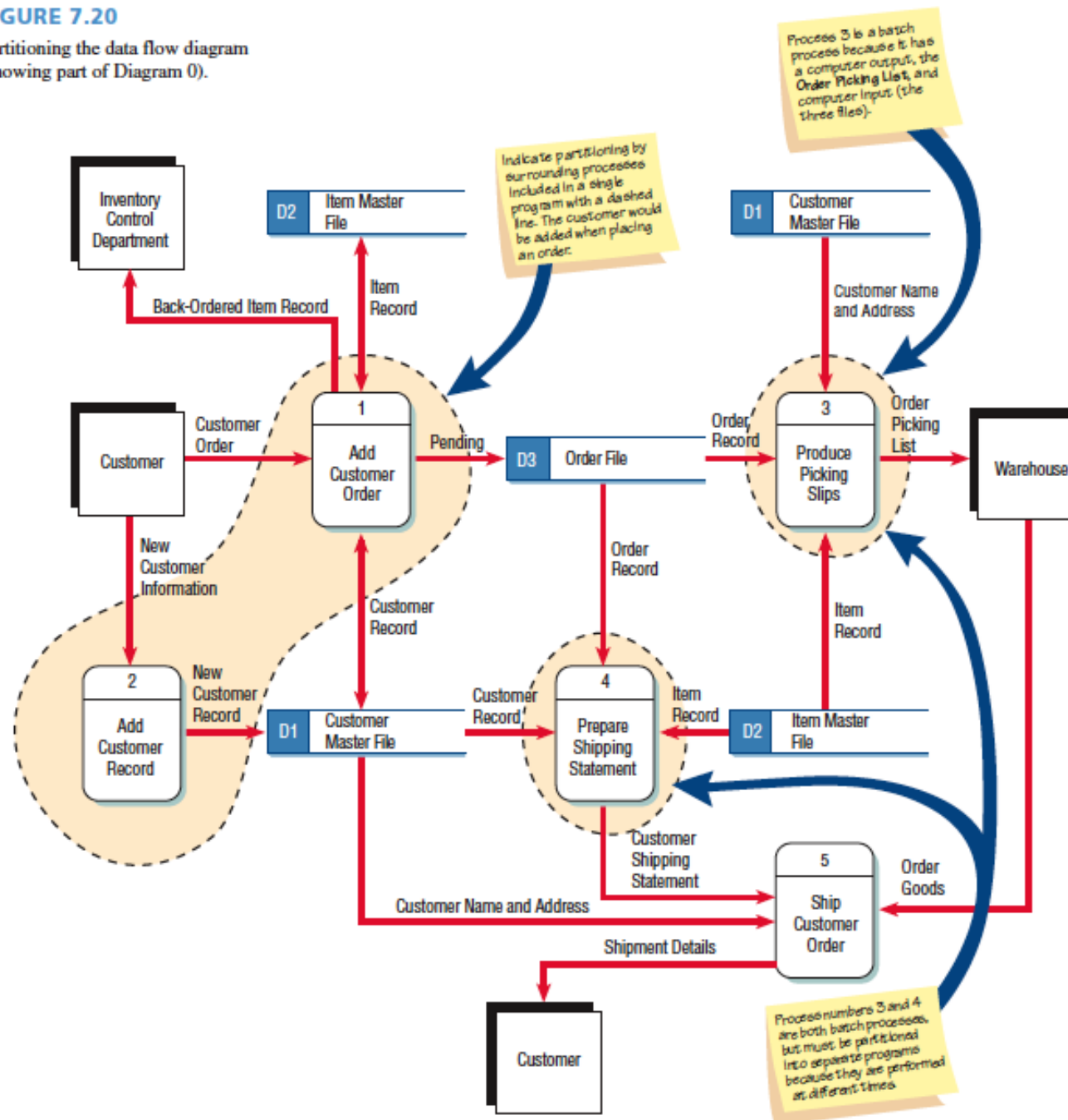
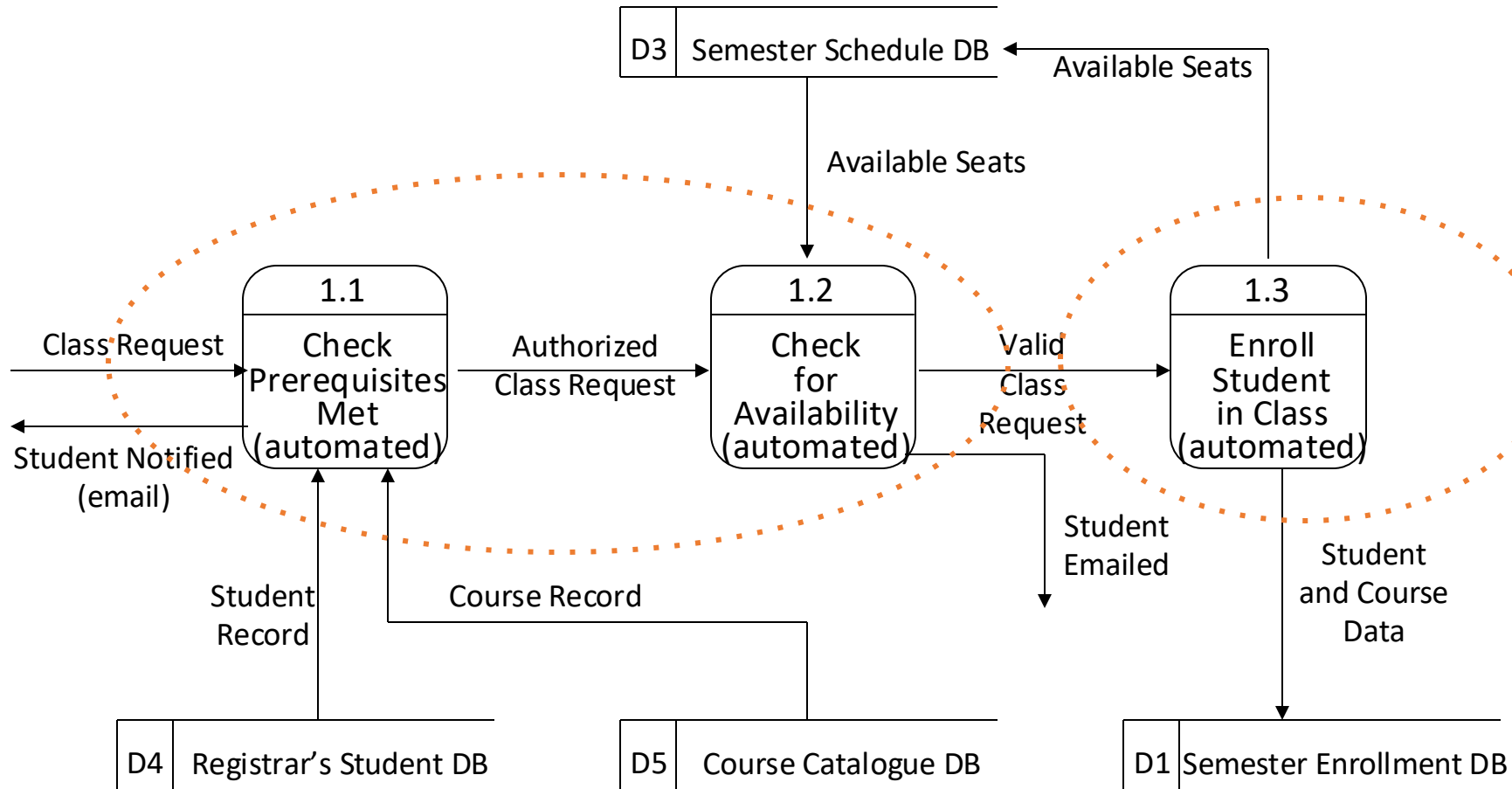


FIGURE 7.20

 Partitioning the data flow diagram
 (showing part of Diagram 0).


■ PARTITIONING DFD (EXAMPLE)

Indicates process that will do AUTOMATED CHECK for requirement and availability



Indicates process that will do AUTOMATED ENROLMENT (DATA INPUT) into a data store

Course Registration: Physical diagram (partitioned)

SUMMARY

DATA FLOW DIAGRAM

- Structured analysis and design tools that allow the analyst to comprehend the system and subsystems visually as a set of interrelated data flows

CREATING DFD

- Rounded rectangle (PROCESS)
- Double square (ENTITY)
- An arrow (DATA FLOW)
- Open-ended rectangle (DATA STORE)

LOGICAL & PHYSICAL DFD

1. Creating the logical DFD
 - Context-level data flow diagram
 - Level 0 logical data flow diagram
 - Child diagrams
2. Creating the physical DFD
 - Create from the logical data flow diagram
 - Partitioned to facilitate programming

PARTITIONING

- Whether processes are performed by different user groups
- Processes execute at the same time
- Processes perform similar tasks
- Batch processes can be combined for efficiency of data
- Processes may be partitioned into different programs for security reasons



UTM
UNIVERSITI TEKNOLOGI MALAYSIA



univteknologimalaysia



utm_my



utmofficial

Thank You

update: August 2019 (sharinhh)

www.utm.my

innovative • entrepreneurial • global