

# DATA FLOW DIAGRAM (DFD)

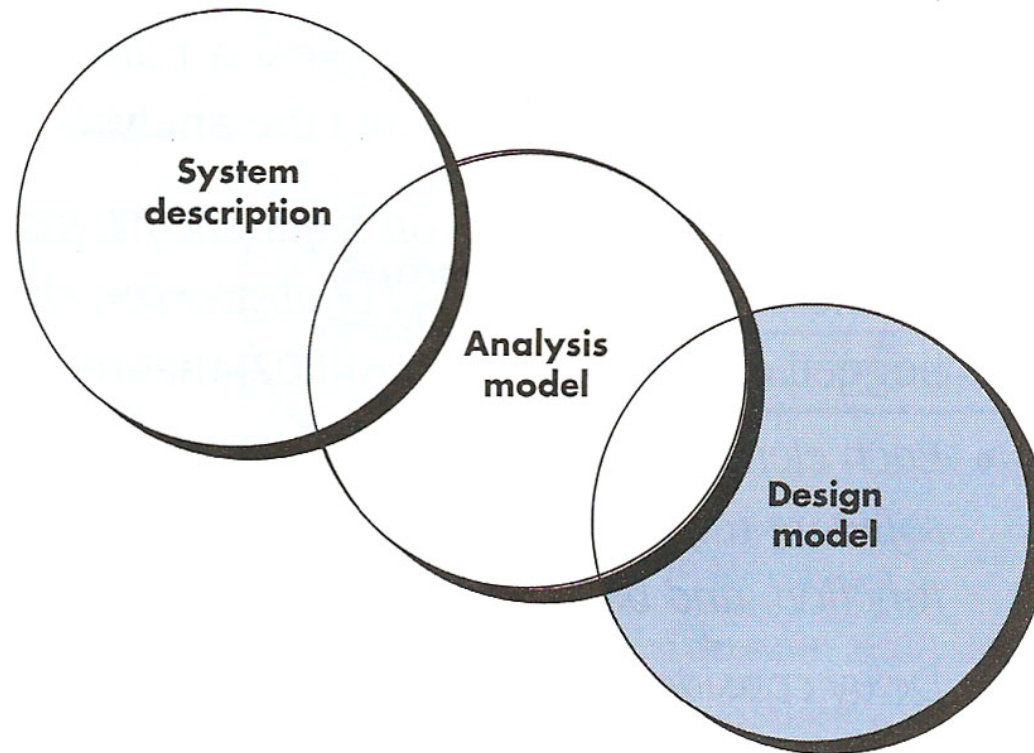
EGCO343 SOFTWARE DESIGN



KANAT POOLSAWASD  
DEPARTMENT OF COMPUTER ENGINEERING  
MAHIDOL UNIVERSITY

# REQUIREMENT ANALYSIS

- The analysis model as a bridge between the system description and the design model.



# STRUCTURED SYSTEMS ANALYSIS

- The use of graphics to specify software was an important technique of the 1970s.
- Three techniques using graphics became particularly popular:
  - DeMarco [1978]
  - Gane and Sarsen [1979]
  - Yourdon and Constantine [1979]
- Gane and Sarsen's approach is presented here because their notation, currently, probably is the most widely used in the industry.

# DATA FLOW DIAGRAM (DFD)

- A diagram that represents system requirements as processes, external agents, data flows, and data stores.
- Some will suggest that the DFD is “old school” and has no place in modern practice. That’s a view that excludes a potentially useful mode of representation at the analysis level. If it can help, use the DFD.
- Information flow continuity must be maintained as each DFD level is refined. This means that input and output at one level must be the same as input and output at a refined level.

# DIFFERENCES BETWEEN DFD AND FLOWCHARTS

- Processes on DFDs can operate in parallel (at-the-same-time)
  - Processes on flowcharts execute one at a time
- DFDs show the flow of data through a system
  - Flowcharts show the flow of control (sequence and transfer of control)
- Processes on a DFD can have dramatically different timing (daily, weekly, on demand)
  - Processes on flowcharts are part of a single program with consistent timing

# DFD SYMBOLS ( GANE AND SARSEN )



**Process**

Step by step in structures are followed that transform inputs into outputs



**Data Flow**

Data flowing from place to place, such as an input or output to a process



**External Agent**

The source or destination of data outside the system



**Data Store**

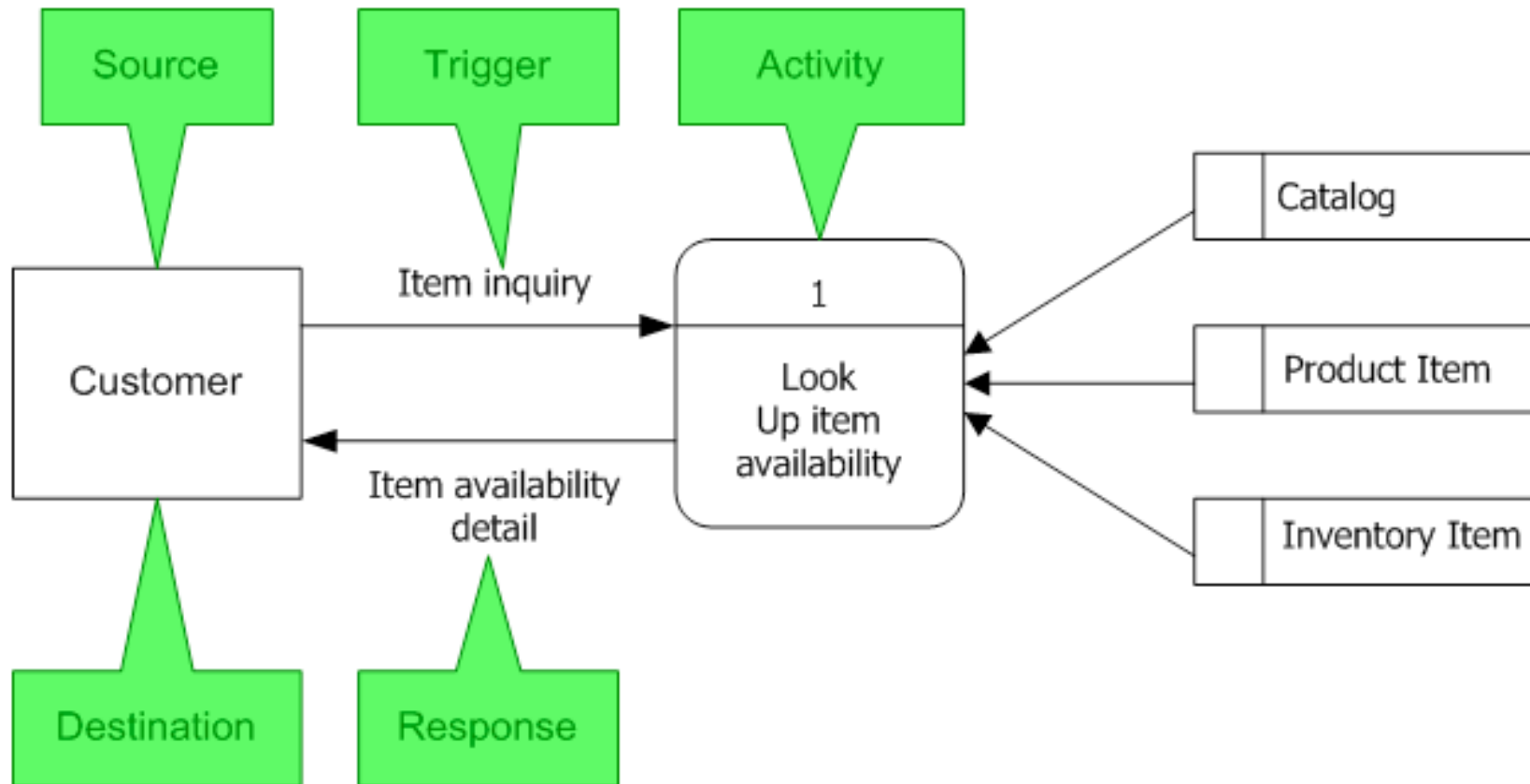
Data at rest, being stored for later use. Usually corresponds to a data entity on an entity-relationship diagram.

# EVENT TABLE (1)

- Information about each event in an event table

Event	Trigger	Source	Activity	Response	Destination
Customer want to check item availability	Item inquiry	Customer	Look up item availability	Item availability detail	Customer

# EVENT TABLE (2)



# DFD AND LEVELS OF ABSTRACTION (1)

- **Decomposition**

Any modeling technique that breaks the system into a hierarchical set of increasingly more detailed models

- **Context Diagram**

A DFD that summarized all processing activity within the system in a single process symbol

# DFD AND LEVELS OF ABSTRACTION (2)

- Decomposition – the act of breaking a system into sub-components. Each level of abstraction reveals more or less detail.
- Decomposition Diagram - A tool used to depict the decomposition of a system. Also called hierarchy chart.

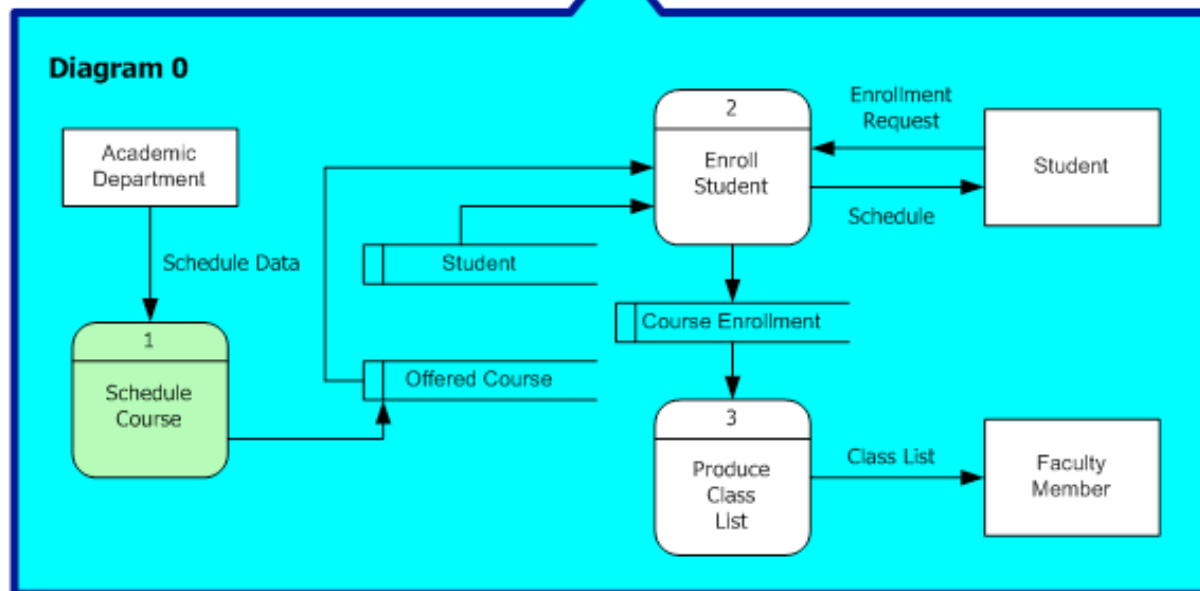
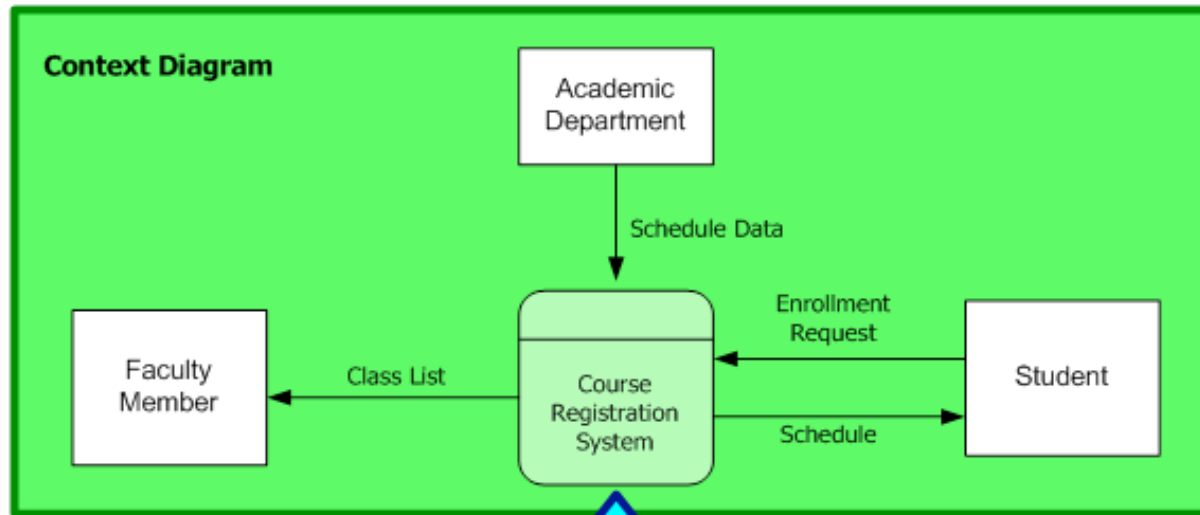


# EVENT-PARTITIONED SYSTEM MODEL (DIAGRAM 0)

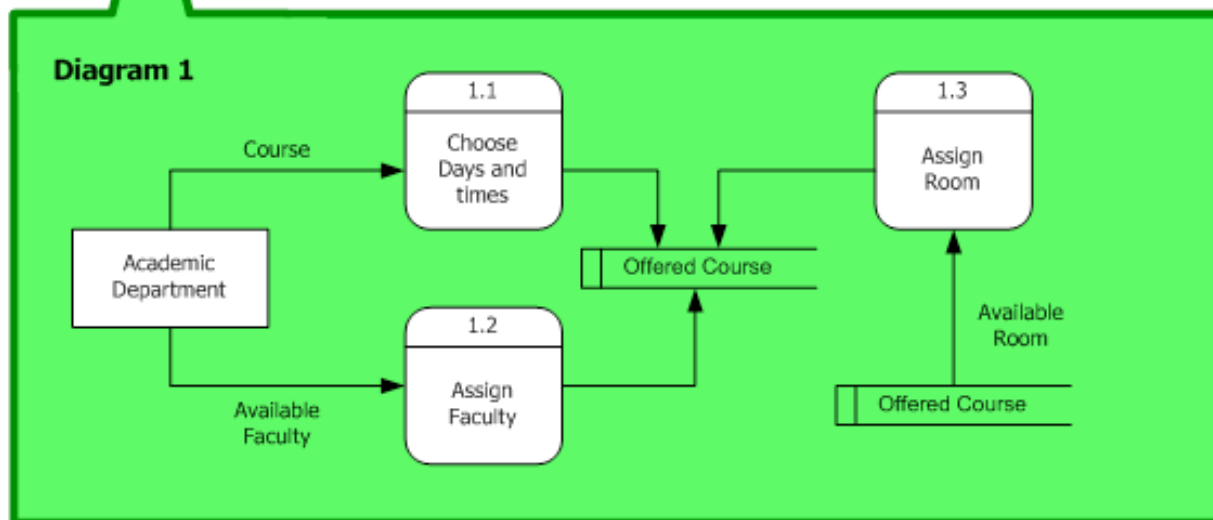
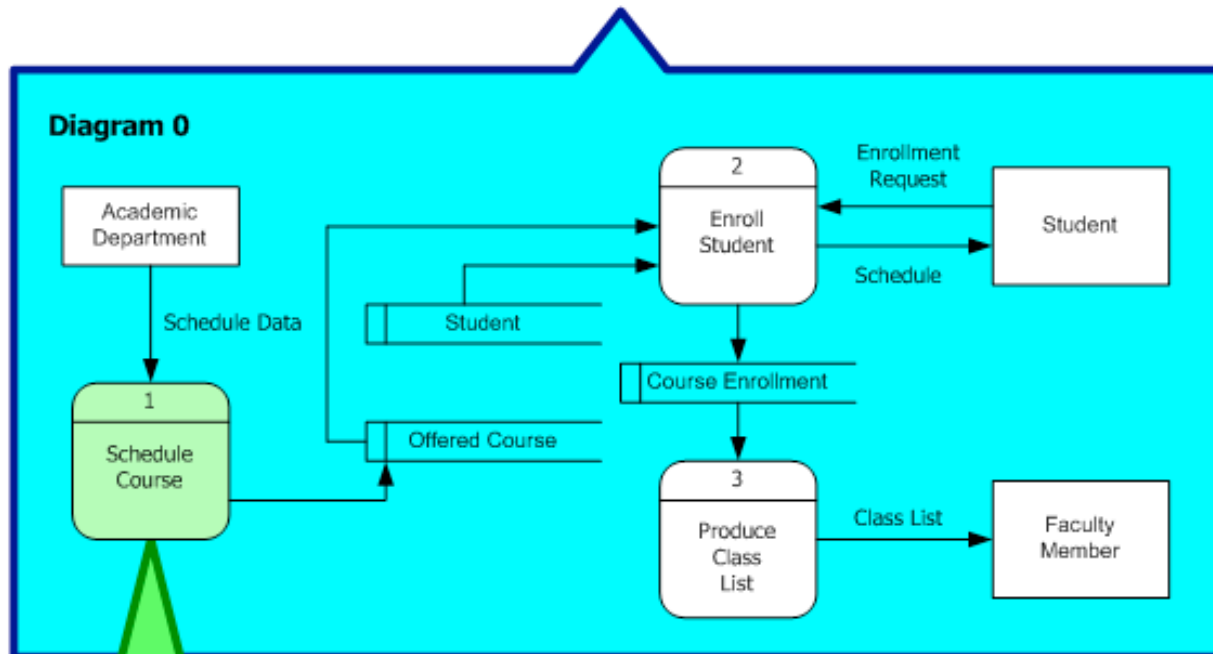
- **Event-Partitioned System Model**

A DFD that models system requirements using a single process for each event in a system or subsystem

# EXAMPLE 1(1): DFDS



# EXAMPLE 1(2): DFDS



# EVALUATING DFD QUALITY (1)

- **Information Overload**

Difficulty in understanding that occurs when a reader receives too much information at one time.

- **Rule of  $7 \pm 2$**

The rule of model design that items the number of model components or connections among components to no more than nine.

# EVALUATING DFD QUALITY (2)

- **Balancing**

Equivalence of data content between data flows entering and leaving a process and data flows entering and leaving a process decomposition DFD.

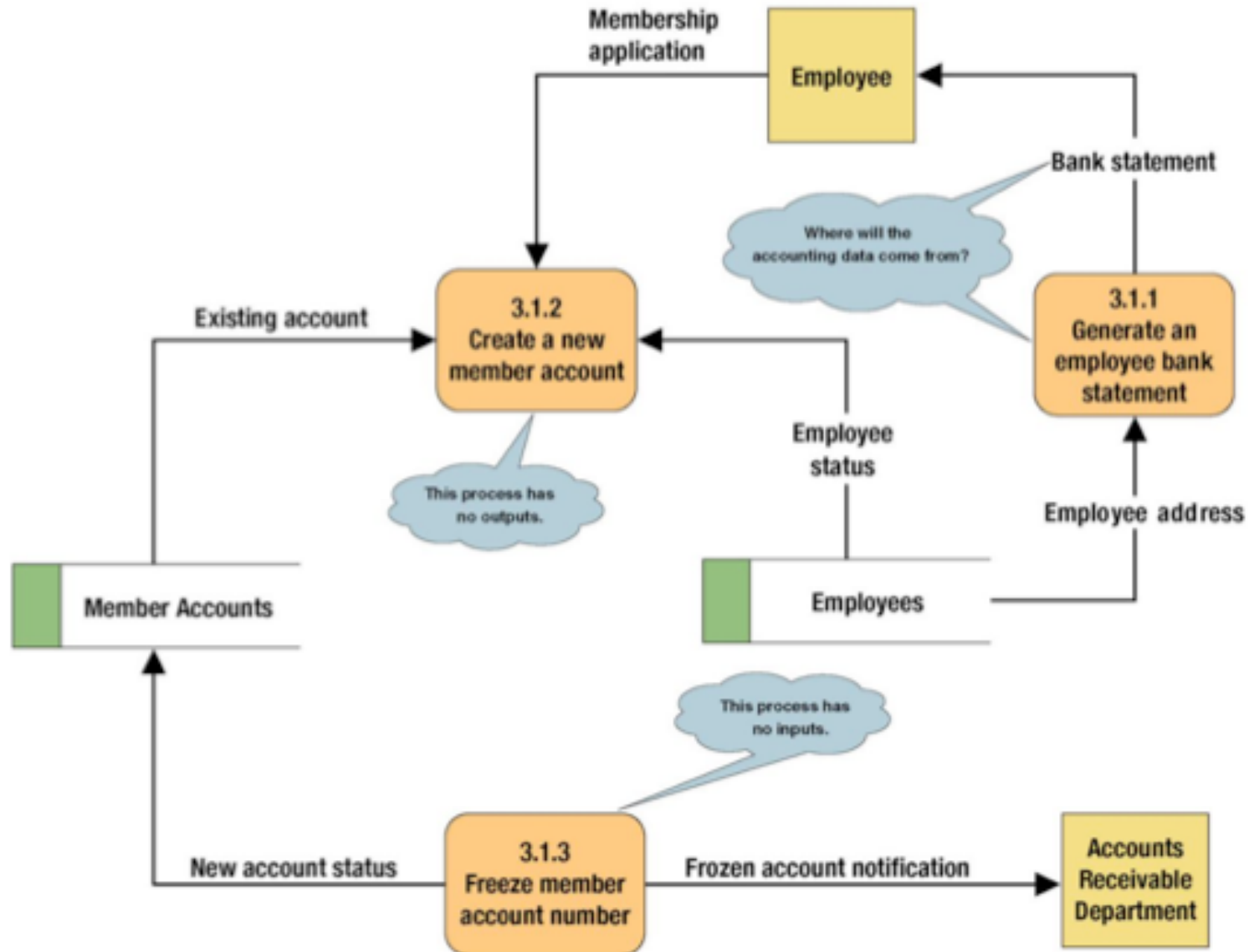
- **Black Hole**

A process or data store with a data input that is never used to produce a data output.

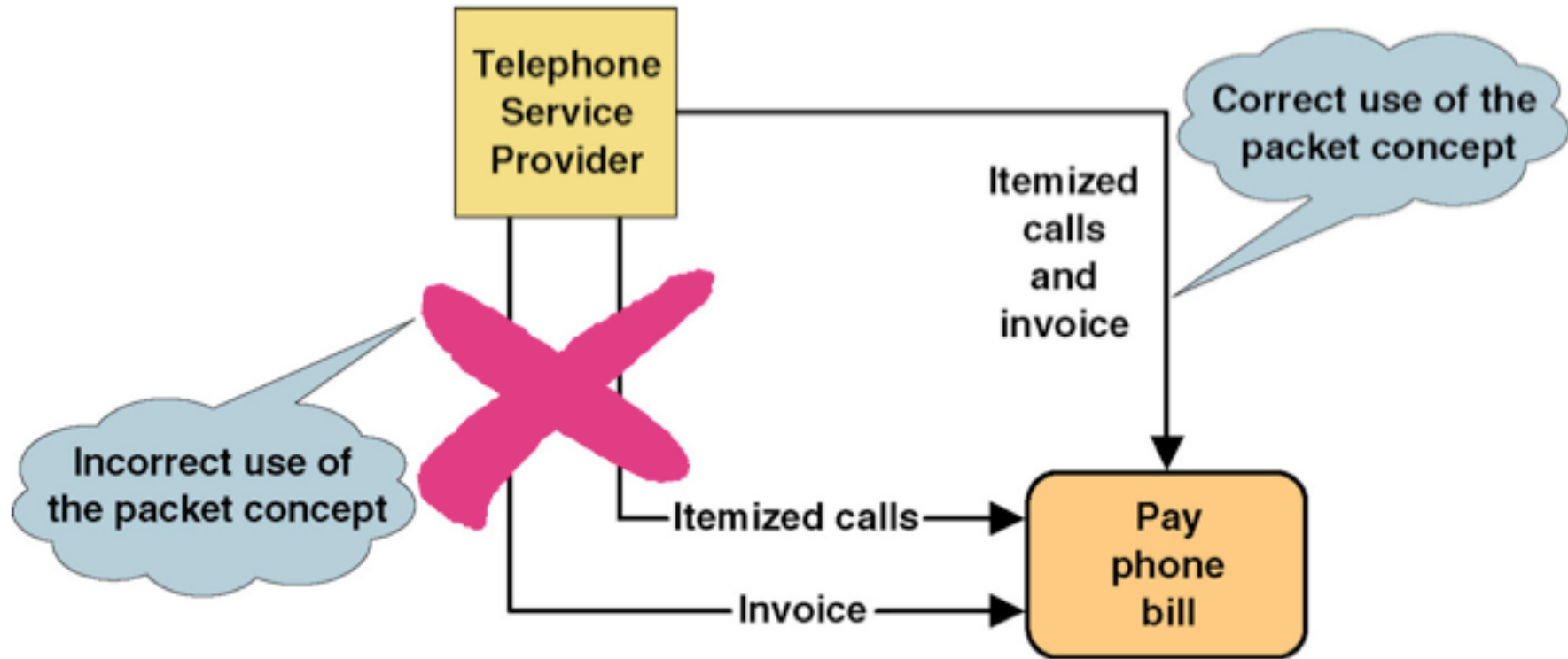
- **Miracle**

A process or data store with a data element that is created out of nothing.

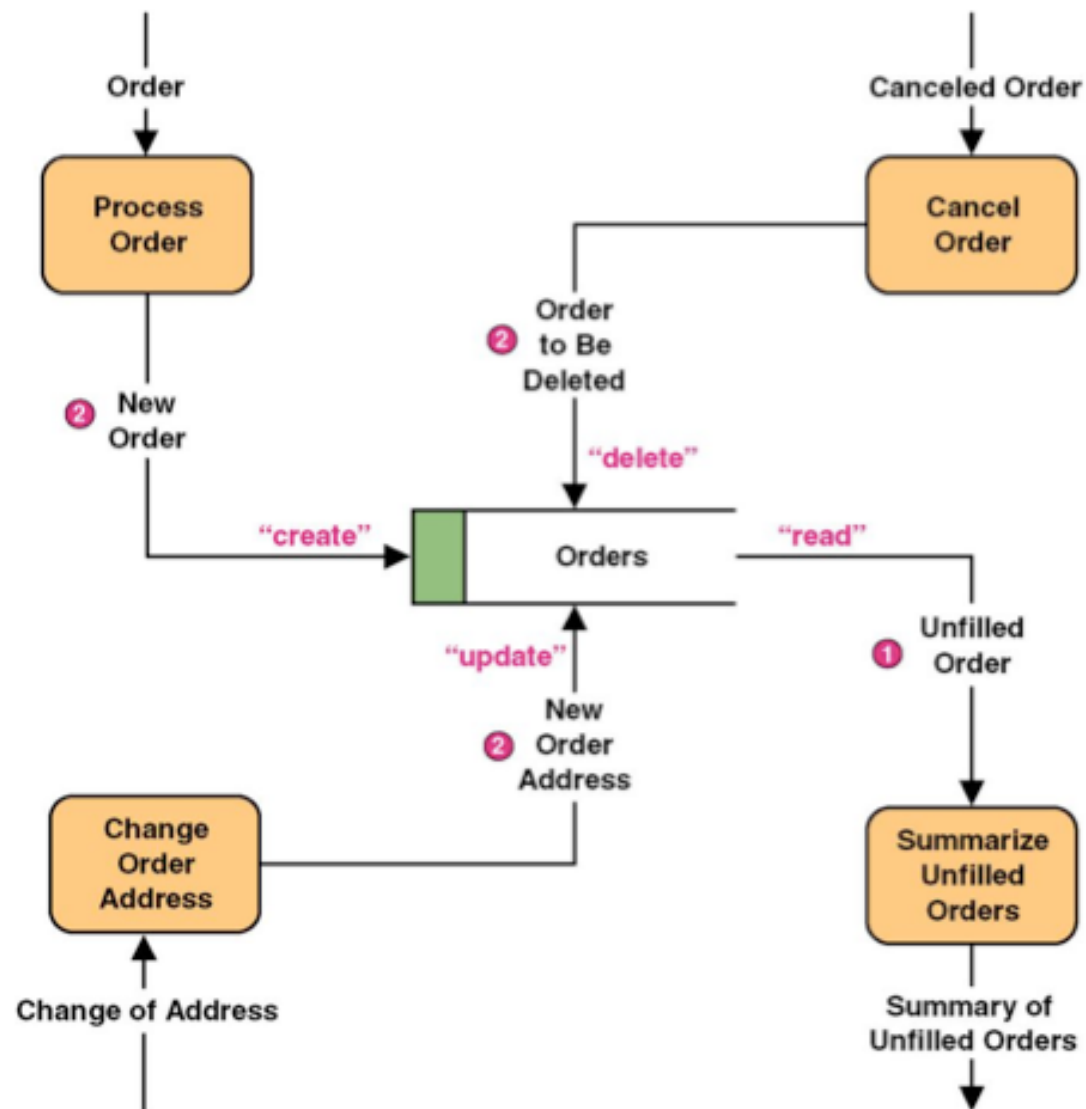
# COMMON PROCESS ERRORS ON DFDS



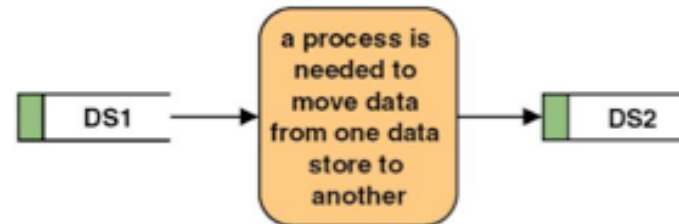
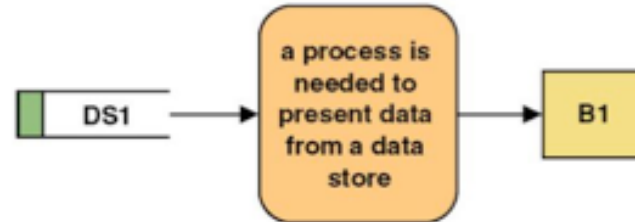
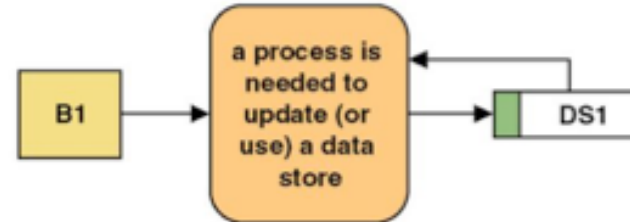
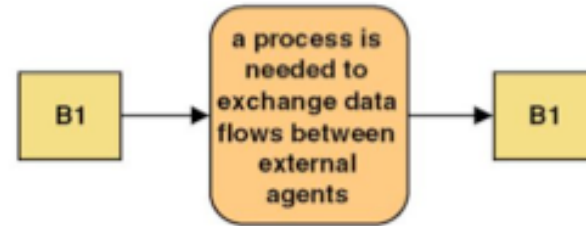
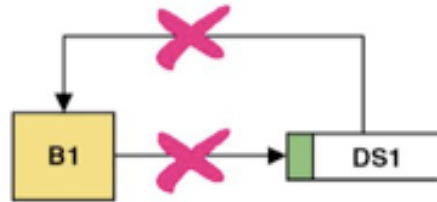
# DATA FLOW PACKET CONCEPT



# DATA FLOWS TO AND FROM DATA STORES



# ILLEGAL DATA FLOWS



# DOCUMENTATION OF DFD COMPONENTS

## Process 2.1 – Record Customer Information

---

Ask if customer has an account (or has made a previous order)

If customer has an account then

    Ask for identification information

    Query database with identifying information

    Copy query response data to order details

Else

    Create an empty customer record in the database

    Ask customer for customer attributes

    Update empty customer record with customer attributes

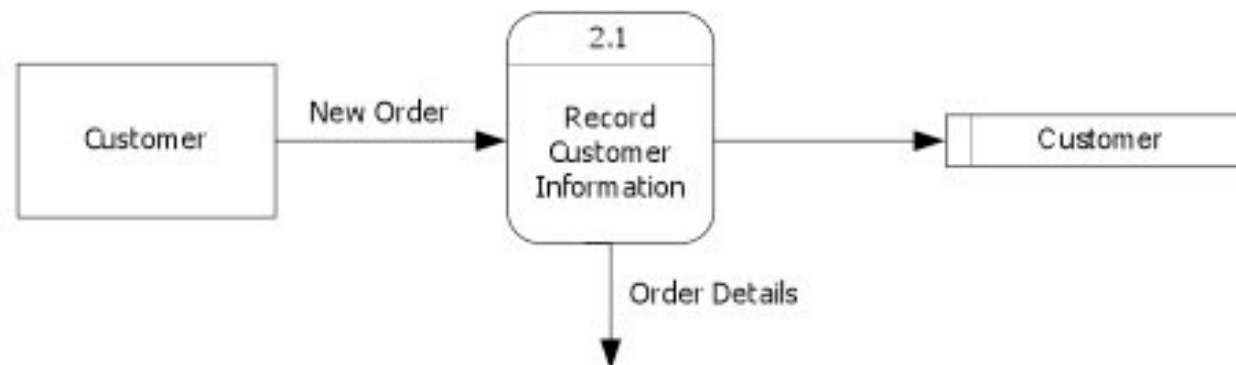
End if

Ask customer for order information for first item

While more order items Do

    Update order details with order information

End while



# DECISION TABLE & DECISION TREE

- **Decision Table**

A tabular representation of processing logic containing decision variables, decision variable values, and actions or formulas.

- **Decision Tree**

A graphical description of processes logic that uses lines organized like branches of a tree.

# DATA FLOW DEFINITION (1)

- **Simply Listing Elements**

Customer\_Name

Customer\_Address

Credit\_Card\_Information

- **Algebraic Notation**

New\_Order = Customer\_Name + Customer\_Address  
+ Credit\_Card\_Information + {Item\_Number +  
Quantity}

# DATA FLOW DEFINITION (2)

- **Data Structure**

A specific arrangement of data attributes that define a single instance of a data flow.

= Means "Consists of" or "is composed of"

+ Means "And" and designates sequence

[...] Means "Only one of the attributes within the brackets may be present" – Designates selection.

{...} Means that the attributes in the braces may occur many times for one instance of the data flow – Designates repetition.

(...) Means the attribute(s) in the parentheses are optional – no value – for some instances of the data flow

# EXAMPLE 2

## DATA STRUCTURE

ORDER =  
ORDER NUMBER +  
ORDER DATE +  
[ PERSONAL CUSTOMER NUMBER,  
CORPORATE ACCOUNT NUMBER] +  
SHIPPING ADDRESS=ADDRESS +  
(BILLING ADDRESS=ADDRESS) +  
1 {PRODUCT NUMBER +  
PRODUCT DESCRIPTION +  
QUANTITY ORDERED +  
PRODUCT PRICE +  
PRODUCT PRICE SOURCE +  
EXTENDED PRICE } N +  
SUM OF EXTENDED PRICES +  
PREPAID AMOUNT +  
(CREDIT CARD NUMBER+EXPIRATION DATE)  
(QUOTE NUMBER)

## ADDRESS =

(POST OFFICE BOX NUMBER) +  
STREET ADDRESS +  
CITY +  
[STATE, MUNICIPALITY] +  
(COUNTRY) +  
POSTAL CODE

# CRUD MATRIX

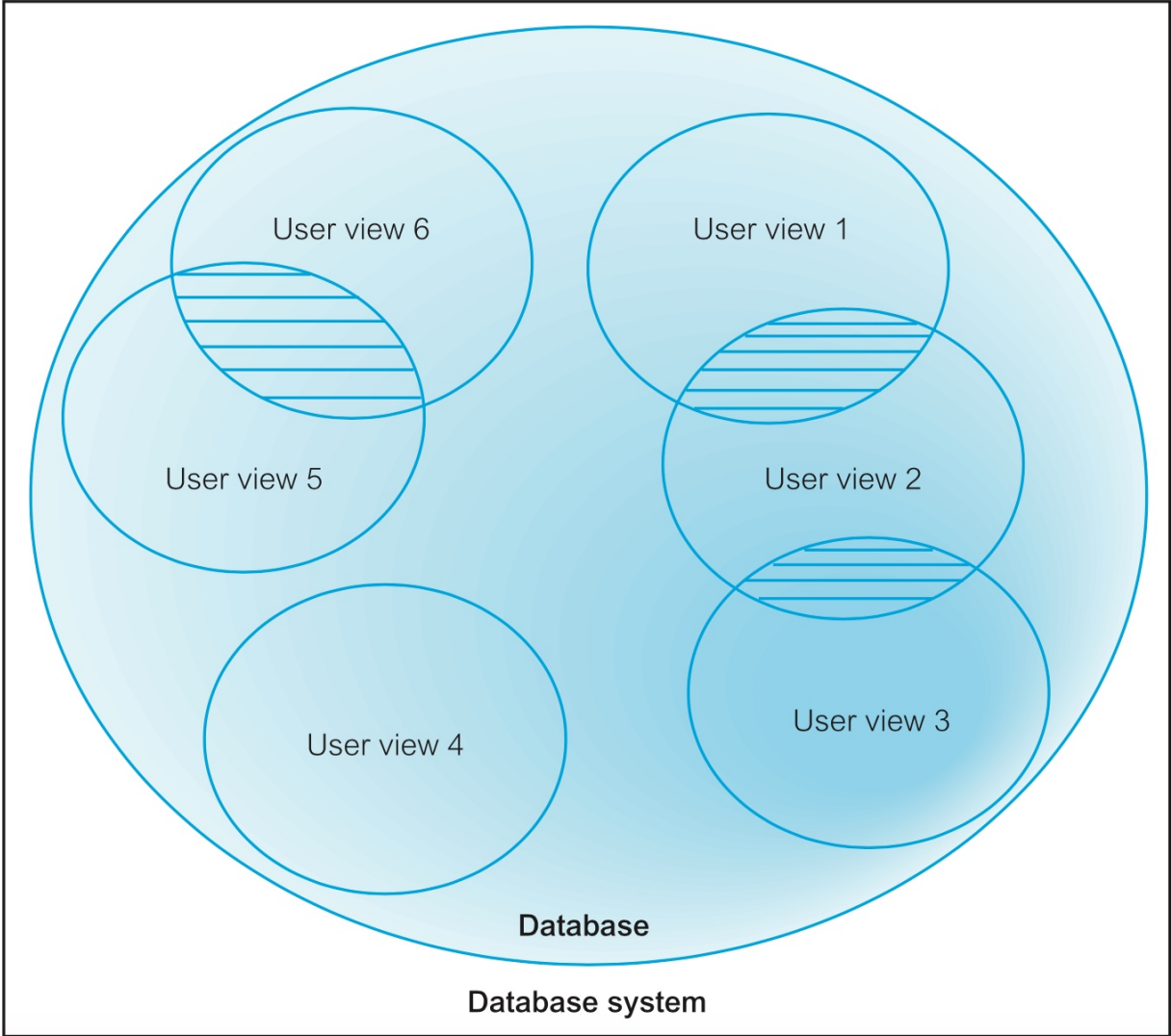
- Acronym of CREATE, READ, UPDATE, and DELETE

Data-to-Process-CRUD Matrix

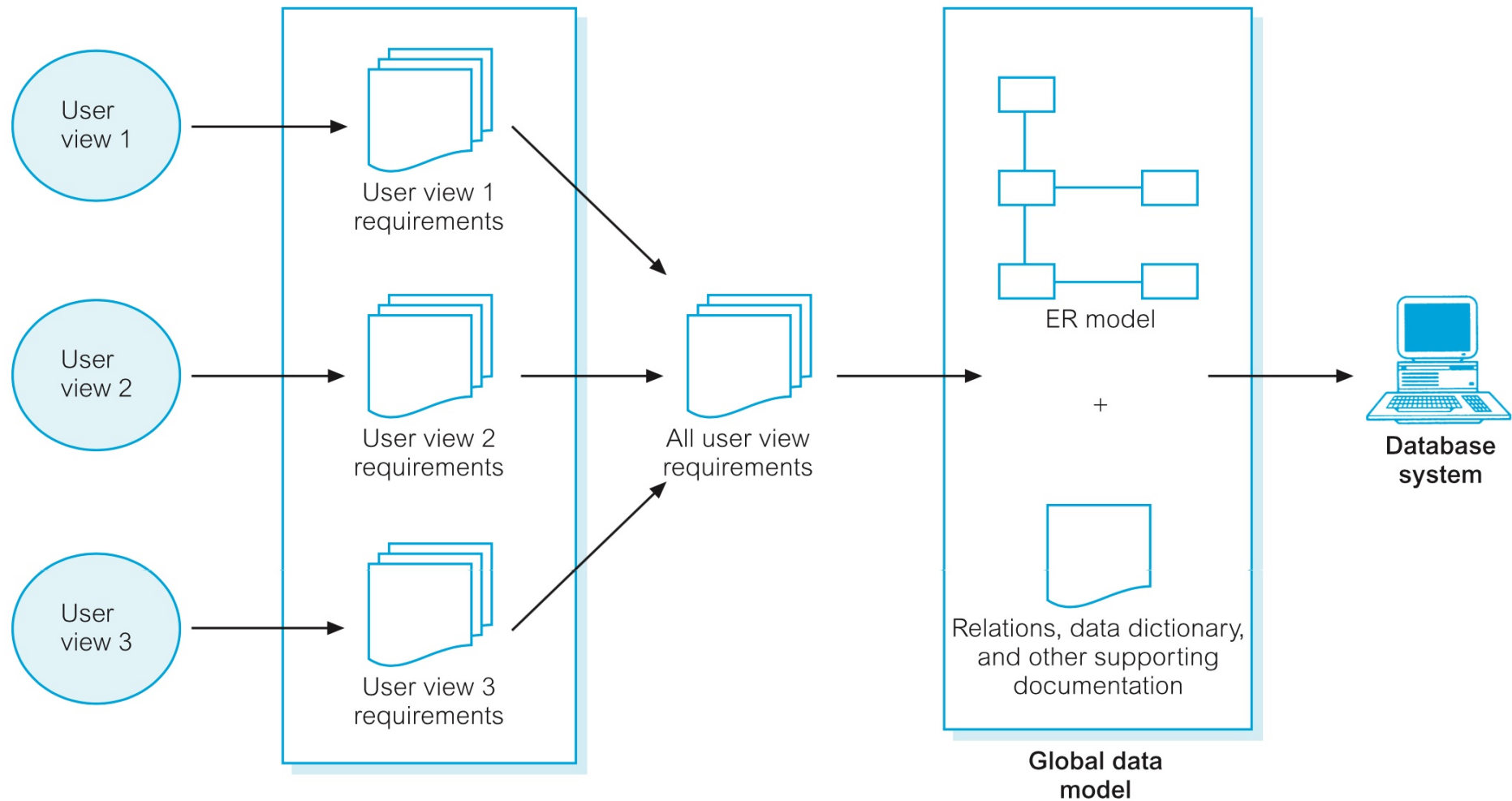
Entity . Attribute	Process Customer Application	Process Customer Credit Application	Process Customer Change of Address	Process Internal Customer Credit Change	Process New Customer Order	Process Customer Order Cancellation	Process Customer Change to Outstanding Order	Process Internal Change to Customer Order	Process New Product Addition	Process Product Withdrawal from Market	Process Product Price Change	Process Change to Product Specification	Process Product Inventory Adjustment
Customer	C	C			R	R	R	R					
.Customer Number	C	C			R	R	R	R					
.Customer Name	C	C	U		R		R	R					
.Customer Address	C	C	U		RU		RU	RU					
.Customer Credit Rating		C		U	R		R	R					
.Customer Balance Due					RU	U	R	R					
Order					C	D	RU	RU					
.Order Number					C		R	R					
.Order Date					C		U	U					
.Order Amount					C		U	U					
Ordered Product					C	D	CRUD	CRUD		RU			
.Quantity Ordered					C		CRUD	CRUD					
.Ordered Item Unit Price					C		CRUD	CRUD					
Product					R	R	R	R	C	D	RU	RU	RU
.Product Number					R	R	R	R	C			R	
.Product Name					R		R	R	C			RU	
.Product Description					R		R	R	C			RU	
.Product Unit of Measure					R		R	R	C		RU	RU	
.Product Current Unit Price					R		R	R			U		
.Product Quantity on Hand					RU	U	RU	RU					RU

C = create    R = read    U = update    D = delete

# THE REPRESENTATION OF A DATABASE SYSTEM



# DATABASE DESIGN APPROACH



# ASSIGNMENT 3

- จาก Assignment 2 เรื่อง “ระบบจองห้อง” จงวิเคราะห์ระบบดังกล่าวด้วย DFD โดยประกอบด้วย
  - Context Diagram
  - Diagram 0
  - Diagram 1
- ออกแบบฐานข้อมูลสำหรับระบบจองห้อง (E-R Diagram)