Systems Analysis and Design

Systems Analysis

Nguyễn Văn Diêu

HO CHI MINH CITY UNIVERSITY OF TRANSPORT

2024

Kiến thức - Kỹ năng - Sáng tạo - Hội nhập
Sứ mệnh - Tầm nhìn
Triết lý Giáo dục - Giá trị cốt lõi

Outline I

1 Determining System Requirements

1.1 Traditional Methods

Interviewing and Listening

Questionnaires

Directly Observing Users

Analyzing Documents

1.2 Modern Methods

Joint Application Design

2 Process Modeling

2.1 DFD

2.2 Rules

2.3 Developing DFD

2.4 Context Diagram

2.5 Next Level

2.6 More Detailed Levels

Outline II

2.7 Checking Diagrams for Errors

2.8 Example

3 Data Dictionary

3.1 Data Elements

3.2 Data Flows

3.3 Data Stores

3.4 Processes

3.5 Entities

3.6 Records

3.7 Data Dictionary Reports

Traditional Methods

- **Interviews with individuals:** Interview individuals informed about the operation and issues of the current system and needs for systems in future organizational activities.
- **Observations of workers:** Observe workers at selected times to see how data are handled and what information people need to do their jobs.
- **Business documents:** Study business documents to discover reported issues, policies, rules, and directions, as well as concrete examples of the use of data and information in the organization.

Steps in Planning the Interview

1. Read background material.
2. Establish interviewing objectives.
3. Decide whom to interview.
4. Prepare the interviewee.
5. Decide on question types and structure.

Question Types

There are three main question types when constructing interview:

- Open-ended questions.
- Closed questions.
- Follow-up questions.

Arranging Questions in a Logical Sequence

The ways of organizing your interviews:

1. Inductive.
   - Beginning with very detailed, often closed questions.
   - The interviewer then expands the topics by allowing open-ended questions and more generalized responses.
2. Deductive.
   - Beginning with generalized, open-ended questions.
   - Then narrowing the possible responses by using closed questions.
3. Combines both inductive and deductive.

Questionnaires

Is a set of written questions for obtaining information from individuals.

- Using when there is a large number of people from whom information and opinions are needed.
- Using for systems intended for use outside of the organization (e.g., by customers or vendors).
- Using for systems with business users spread across many geographic locations.
- Today, most questionnaires are being distributed in electronic form, either via e-mail or on the Web.

Directly Observing Users

- Watching users do their jobs.
- Obtaining how people behave in work situations.
- Can cause people to change their normal operating behavior.
- Very time consuming. Not only observe for a limited time, but also a limited number of people and a limited number of sites.

Analyzing Documents

In documents we can find information about:

- Problems with existing systems.
- Meet the new needs information.
- Organizational direction in information system requirements.
- Titles and names of key individuals who have an interest existing systems.
- Organization or individuals who can help determine priorities different.
- Special information-processing that occur irregularly.
- The reason why current systems are designed as they are.
- Data, rules for processing data.

Modern Methods

These techniques can support effective information collection and structuring while reducing the amount of time required for analysis.

- **Joint application design (JAD)**.
- **Prototyping**.

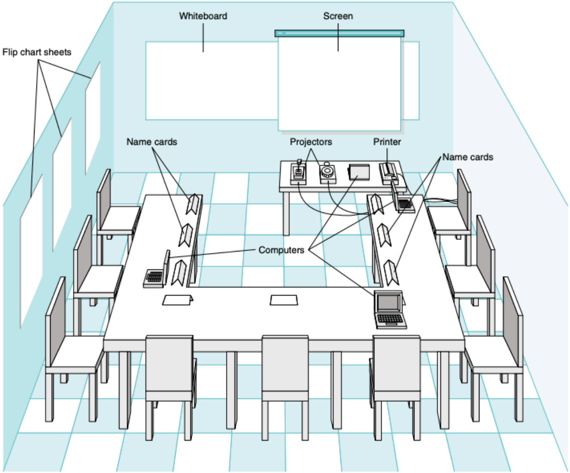
Joint Application Design

JAD started in the late 1970s at IBM as a means to bring together the key users, managers, and systems analysts involved in the analysis of a current system.

Typical JAD participants:

- JAD session leader.
- Users.
- Managers.
- Sponsor.
- Systems analysts.
- Scribe.
- IS staff.

Joint Application Design



Process Modeling

Process Model describes business processes - the activities that people do.

Process models are developed for the as-is system and/or the to-be system.

We use data flow diagram (**DFD**), one of the most commonly used process modeling techniques.

Data Flow Diagram (DFD)

Definitions and Symbols:

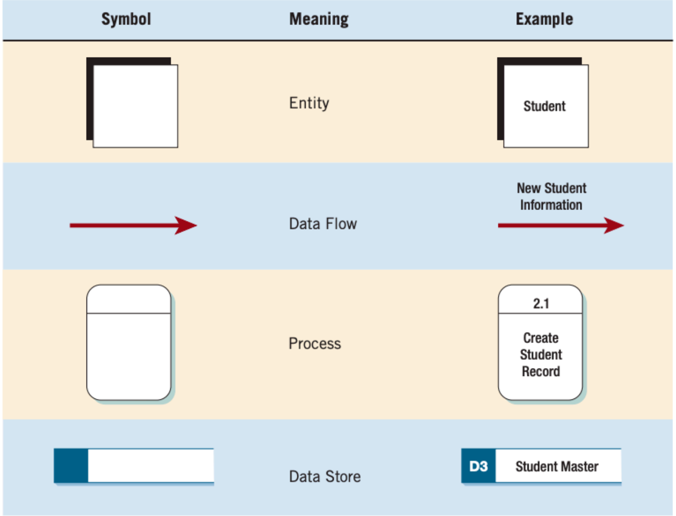
1. **External entity (Entity)**, Source or Destination of data. Named with a noun. The same entity may be used more than once on a given DFD to avoid crossing data flow lines.

2. **Data flow** shows movement of data from one point to another. Name with a noun.

3. **Process** is the work or actions performed on data so that they are transformed, stored, or distributed. A clear name makes it easier to understand what the process is accomplishing.

4. **Data store** is data at rest. Name with a noun.

Symbols used in DFD

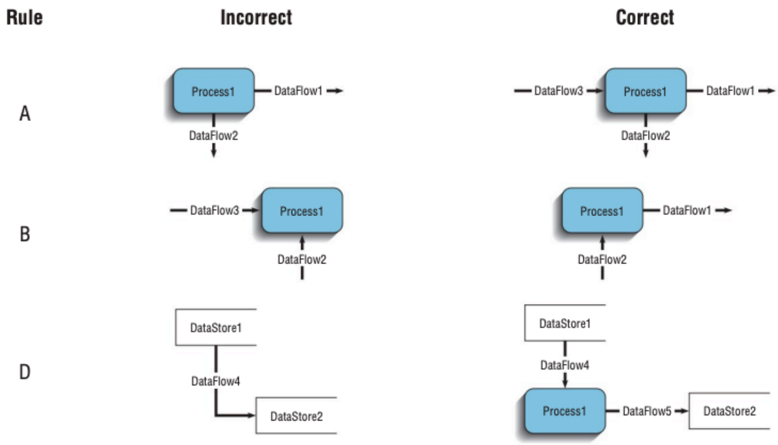


Symbol	Meaning	Example
	Entity	Student
	Data Flow	New Student Information
	Process	2.1 Create Student Record
	Data Store	D3 Student Master

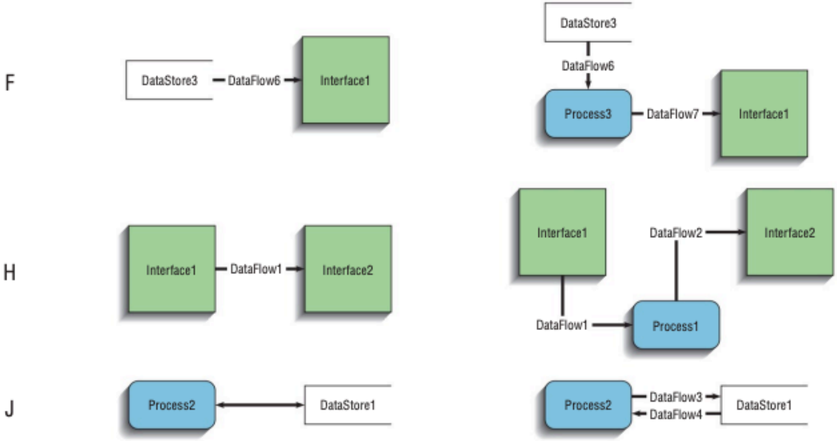
Basic rules to develop DFD

1. A DFD must have at least one process, and it must not have any freestanding objects or objects connected to themselves.
2. A process must receive at least one data flow coming into the process and create at least one data flow leaving from the process.
3. A data store should be connected to at least one process.
4. External entities should not be connected to each other. Although they communicate independently, that communication is not part of the system we design using DFDs.

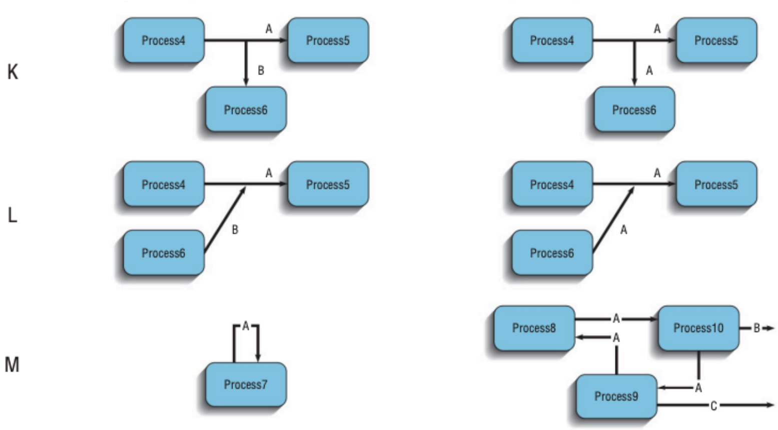
Basic rules to develop DFD



Basic rules to develop DFD



Basic rules to develop DFD



Developing DFD using a Top-Down Approach

1. Make a list of business activities and use it to determine:
   - 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.

Developing DFD using a Top-Down Approach

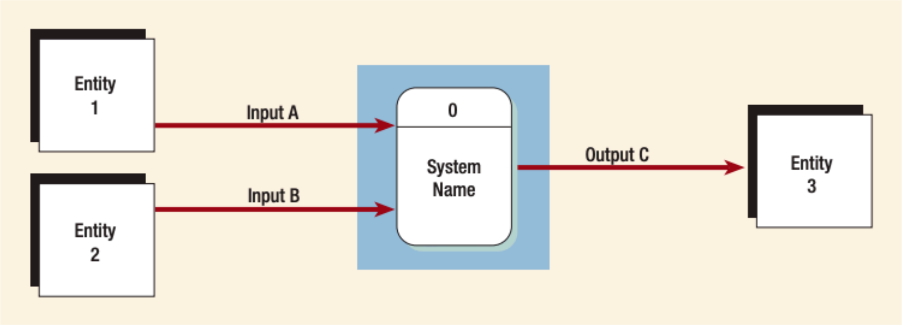
5. Check for errors and make sure the labels you assign to each process and data flow are meaningful.
6. Develop a physical data flow diagram from the logical data flow diagram. Distinguish between manual and automated processes, describe actual files and reports by name, and add controls to indicate 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.

Creating the Context Diagram

Context diagram

- Highest level in a DFD and 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.

Context Diagram

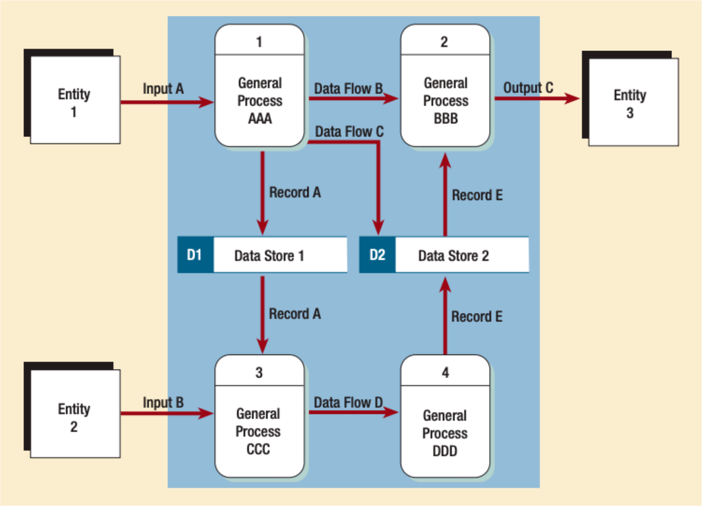


Next Level

Diagram 0

- Explosion of the context diagram and may include up to 9 processes.
- Including more processes at this level.
- Each process is numbered with an integer.
- Generally starting from the upper left-hand corner of the diagram and working toward the lower right-hand corner.

Diagram 0


Entity 1
Input A
1
General Process AAA
Data Flow B
2
General Process BBB
Output C
Entity 3
Data Flow C
Record A
Record E
D1
Data Store 1
D2
Data Store 2
Record A
Record E
Entity 2
Input B
3
General Process CCC
Data Flow D
4
General Process DDD

Development

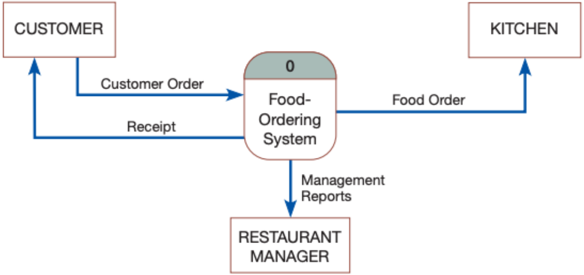
1. Start with the data flow from an entity on the input side. Ask questions such as: "What happens to the data entering the system?" "Is it stored?" "Is it input for several processes?"

2. Work backward from an output data flow.
   - Examine the output fields.
   - For each field on the output, ask **"Where does it come from?"** or **"Is it calculated or stored on a file?"**
   - For example, when the output is a **PAYCHECK**, the **EMPLOYEE NAME** and **ADDRESS** would be located on an **EMPLOYEE** file, the **HOURS WORKED** would be on a **TIME RECORD**, and the **GROSS PAY** and **DEDUCTIONS** would be calculated.
   - Each file and record would be connected to the process that produces the paycheck.

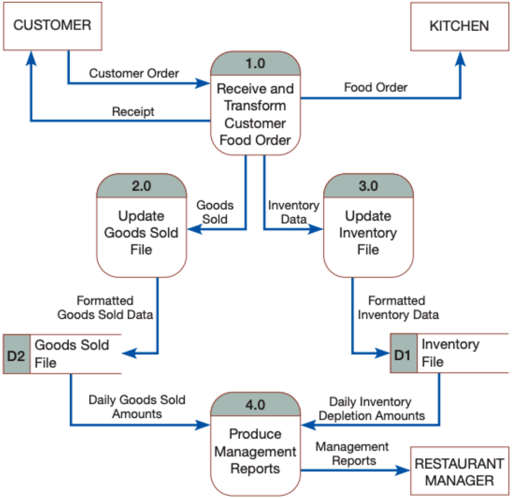
Development

3. Examine the data flow to or from a data store.
   - Ask **“What processes put data into the store?”** or **“What processes use the data?”**.
   - Note that a data store used in the system you are working on may be produced by a different system. Thus, from your point, there may not be any data flow into the data store.
4. Analyze a well-defined process.
   - Look at what input data the process needs and what output it produces.
   - Then connect the input and output to the appropriate data stores and entities.
5. Take note of any unsure areas of what should be included or what input or output is required.
   - Help create a list of questions for interviews with key users.

e.g. Context Diagram



e.g. Diagram 0


CUSTOMER
KITCHEN
Customer Order
Receipt
1.0
Receive and Transform Customer Food Order
Food Order
2.0
Update Goods Sold File
Goods Sold
Inventory Data
3.0
Update Inventory File
Formatted Goods Sold Data
Formatted Inventory Data
D2
Goods Sold File
D1
Inventory File
Daily Goods Sold Amounts
4.0
Produce Management Reports
Daily Inventory Depletion Amounts
Management Reports
RESTAURANT MANAGER

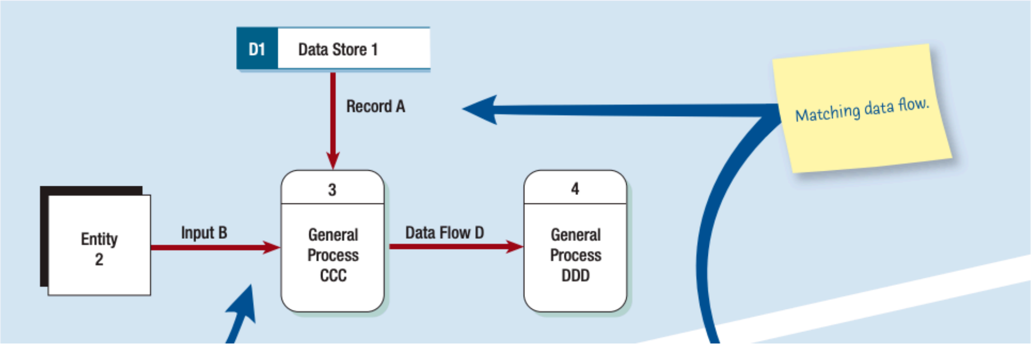
More Detailed Levels

- Each process on Diagram 0 may in turn be exploded to create a more detailed child diagram.
  - Diagram 0: Parent process.
  - Diagram results: Child diagram.
- The primary rule for creating child diagrams is vertical balancing:
  - Child diagram cannot produce output or receive input that the parent process does not also produce or receive.
- All data flow into or out of the parent process must be shown flowing into or out of the child diagram.
- Child diagram is given the same number as its parent process in Diagram 0.
  - e.g. process 3 would explode to Diagram 3.
  - On Diagram 3, the processes would be numbered 3.1, 3.2, ...

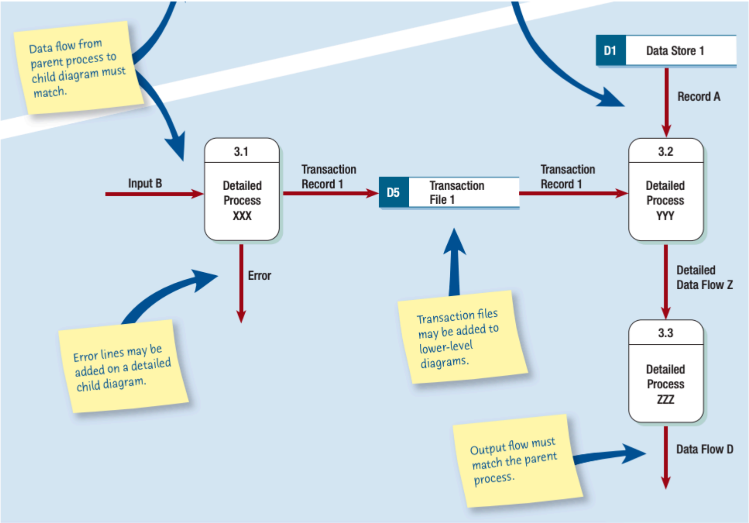
More Detailed Levels

- Entities are usually not shown on the child diagrams below Diagram 0.
- Data flow that matches the parent flow is called an interface data flow and is shown as an arrow from or into a blank area of the child diagram.
- If the parent process has data flow connecting to a data store, the child diagram may include the data store as well.
- In addition, this lower-level diagram may contain data stores not shown on the parent process.
- Processes may or may not be exploded, depending on their level of complexity.

e.g. Parent diagram



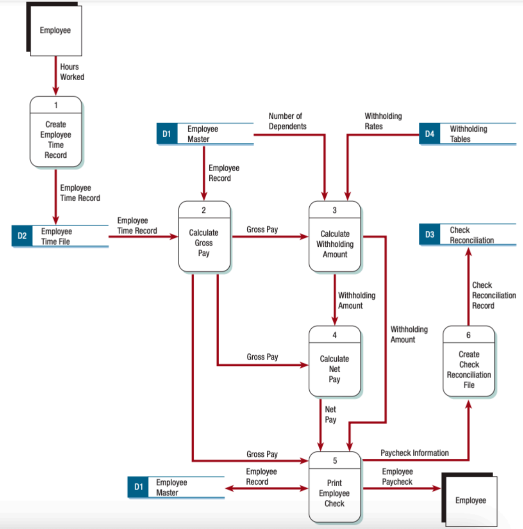
e.g. Child diagram


Data flow from parent process to child diagram must match.
D1
Data Store 1
Record A
3.1
Detailed Process XXX
Input B
Transaction Record 1
D5
Transaction File 1
Transaction Record 1
3.2
Detailed Process YYY
Error
Detailed Data Flow Z
3.3
Detailed Process ZZZ
Error lines may be added on a detailed child diagram.
Transaction files may be added to lower-level diagrams.
Output flow must match the parent process.
Data Flow D

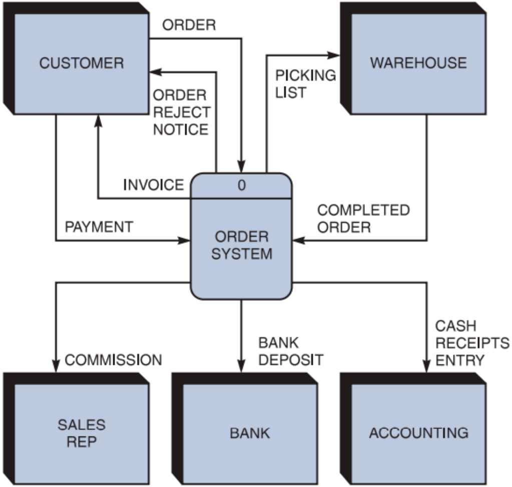
Checking Diagrams for Errors

1. Forgetting to include a data flow or pointing an arrow in the wrong direction.
2. Connecting data stores and external entities directly to each other.
3. Incorrectly labeling processes or data flow.
4. Including more than nine processes on a DFD.
5. Omitting data flow.
6. Creating unbalanced decomposition (or explosion) in child diagrams.

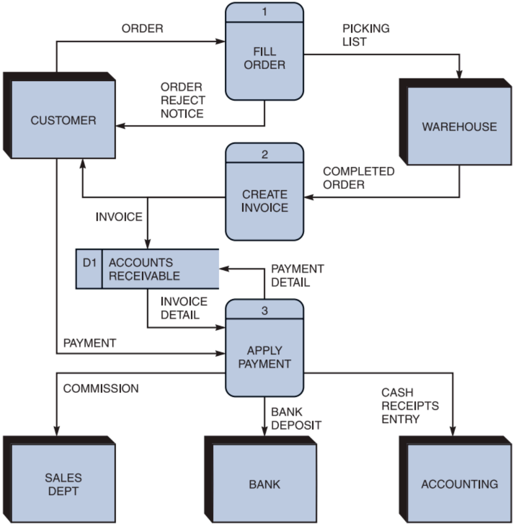
The correct data flow diagram for the payroll example



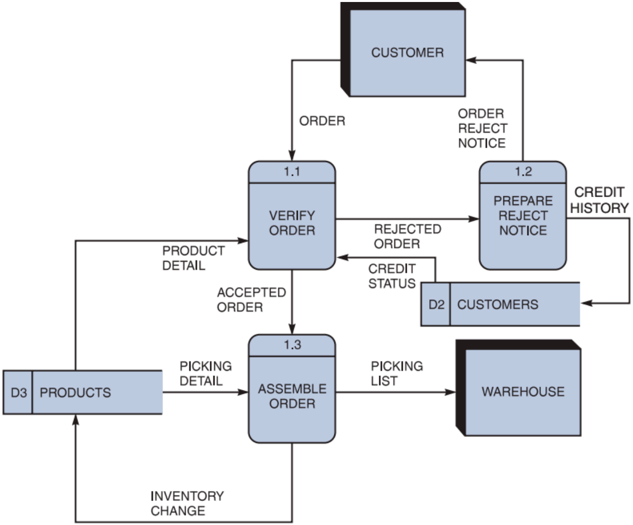
Order System :: Context Diagram



Order System :: Diagram 0



Order System :: Diagram 1: Fill Order Process



Data Dictionary

- Central storehouse of information about the system's data.
  - Data flows.
  - Data stores.
  - Entities.
  - Processes.
  - Data elemen (data item or field).
- Analyst uses the data dictionary to collect, document, and organize specific facts about the systemt.

Data Elements

1. **Name**. Data element name or label.
2. **Alias**.
3. **Type**. Type and length.
4. **Default**. Default value.
5. **Domain**.
6. **Source**.
7. **Security**. Identification for the individual or department that has access or update for each data element.
8. **Responsible**. Responsible user(s); Identification of the user(s) responsible for entering and changing values for the data element.
9. **Description**. Description and comments.

Data Flows

1. **Name**. Data flow name or label.
2. **Description**. Describes the data flow and its purpose.
3. **Alternate name(s)**. Aliases for the DFD data flow name(s).
4. **Origin**. The DFD beginning, or source, for the data flow; the origin can be a process, a data store, or an entity.
5. **Destination**. The DFD ending point(s) for the data flow; the destination can be a process, a data store, or an entity.
6. **Record**. Each data flow represents a group of related data elements called a record or data structure. In most data dictionaries, records are defined separately from the data flows and data stores. When records are defined, more than one data flow or data store can use the same record, if necessary.
7. **Volume**. Describes the expected number of occurrences for the data flow per unit of time.

Data Stores

1. **Name**. Data store name or label. The data store name as it appears on the DFDs.
2. **Description**. Describes the data store and its purpose.
3. **Alternate name(s)**. Aliases for the DFD data store name.
4. **Attributes**. Standard DFD names that enter or leave the data store.
5. **Volume**. Describes the estimated number of records in the data store and how frequently they are updated.

Processes

1. **Name**. Process name or label. The process name as it appears on the DFDs.
2. **Description**. A brief statement of the process's purpose.
3. **Process number**. A reference number that identifies the process and indicates relationships among various levels in the system.
4. **Process description**. This section includes the input and output data flows.

Entities

1. **Name**. Entity name. The entity name as it appears on the DFDs.
2. **Description**. Describe the entity and its purpose.
3. **Alternate name(s)**. Any aliases for the entity name.
4. **Input data flows**. The standard DFD names for the input data flows to the entity.
5. **Output data flows**. The standard DFD names for the data flows leaving the entity.

Records

1. **Name**. Record or data structure name. The record name as it appears in the related data flow and data store entries in the data dictionary.
2. **Definition**. A brief definition of the record.
3. **Alternate name(s)**. Any aliases for the record name.
4. **Attributes**. A list of all the data elements, or fields, included in the record. The data element names must match exactly those entered in the data dictionary.

Data Dictionary Reports

Many valuable reports can be obtained from a data dictionary, including the following:

1. An alphabetized list of all data elements by name.
2. A report describing each data element and indicating the user or department that is responsible for data entry, updating, or deletion.
3. A report of all data flows and data stores that use a particular data element.
4. Detailed reports showing all characteristics of data elements, records, data flows, processes, or any other selected item stored in the data dictionary.

We can use tables for data dictionary anh report.