Data-flow diagrams and exchange-system relationships

What a data-flow diagram shows

A data-flow diagram (DFD) represents where data comes from, how a system transforms it, where it is stored and where it goes. It focuses on logical data relationships rather than screen layout, program-control order or network hardware.

> **Ask four questions:** Who supplies or receives data? What process transforms it? Where is it stored? What data crosses each connection?

A DFD can contain physical objects and manual storage as well as digital data. For example, a library process may receive a book identifier, update a digital record and release a physical book from a shelf.

DFD or flowchart?

Determine which representation best suits each purpose.

1. Show the decisions and repetition in an account-login algorithm.
2. Show how login credentials move from a user to a verification process and user store.

Check your answer

1. A flowchart or pseudocode is appropriate because the purpose is to show algorithm control and decisions.
2. A DFD is appropriate because the purpose is to show data sources, transformation, storage and movement.

The four Gane–Sarson elements

QCAA Digital Solutions uses Gane–Sarson notation. Copy the shapes accurately enough that their different functions are unambiguous.

Examples of an external entity, process, data store and labelled directional data flow. Student External entity 1.0 Check booking Process D1 Bookings Data store booking request Labelled data flow; arrow shows direction

Element	Job	Naming convention
External entity	A source or destination outside the area being studied.	Noun, such as *Student* or *Payment service*.
Process	Transforms incoming data into outgoing data.	Verb or verb–object name, such as *Check booking*.
Data store	A repository holding data for later use.	Numbered noun, such as *D1 Bookings*.
Data flow	Shows data moving in the arrow's direction.	Noun describing the actual data, such as *booking request*.

Classify each system component

A canteen ordering system includes a student, a process that calculates the order total, a collection of menu records and the order details sent into that process.

Recognise each DFD element and give it a suitable label.

Check your answer

- **External entity:** Student
- **Process:** 1.0 Calculate order total
- **Data store:** D1 Menu
- **Data flow:** order details

Name and number elements precisely

Good labels allow a reader to understand the system without guessing.

- Use nouns for entities, flows and stores: *Customer*, *payment details*, *Orders*.
- Use verbs or verb–object names for processes: *Validate payment*, *Record order*.
- Number processes: *1.0 Validate request*, *2.0 Record booking*.
- Number and type stores: *D1 Accounts*, *M1 Paper forms*, *T1 Session cache*.

Prefix	Store type	Example
D	Data on disk, retained when power is off	D1 Accounts database
M	Manual data: physical files, paper records or manually processed digital data	M1 Consent forms
T	Temporary or transient data, not retained long-term	T1 Session cache

> **Avoid vague labels:** *data*, *information*, *do booking* and *system* hide the relationship. Name the actual data and transformation.

Repair the labels

A draft DFD uses *Member* for an entity, *Bookings* for a process, *Save* for a data store and *data* for a flow containing a requested date and room identifier.

Analyse the labels and rewrite the three unsuitable ones.

View a sample solution

- *Member* is already a suitable noun for the entity.

- Rename the process *1.0 Create booking* to state its transformation.
- Rename the store *D1 Bookings* because a store uses a noun and identifier.
- Rename the flow *date and room ID* or *booking request* to identify its contents.

Connections must pass through a process

A process is the active element that transforms, validates, records or retrieves data. Therefore an external entity cannot exchange data directly with another external entity or a data store.

Connection	Valid?	Reason
Entity → process	Yes	The process receives data to transform.
Process → store	Yes	The process writes data to storage.
Store → process	Yes	The process retrieves stored data.
Entity → store	No	Data must be processed first.
Entity → entity	No	That exchange is outside the represented system unless a process handles it.

Use a separate arrow for each direction. A request and reply are different flows and should be labelled separately.

Find three convention errors

A draft diagram shows a *Customer* entity sending *details* directly to *D1 Orders*. The store sends *stuff* directly to a *Kitchen* entity. A process between them is named *Order*.

Analyse the draft and describe three corrections.

Check your answer

1. Route the customer's *order details* through a process before D1 Orders.

2. Route stored order data through a process before outputting a *kitchen order* to Kitchen.
3. Rename and number the process with a verb–object label, such as *1.0 Record order*.

Worked example: equipment loans

A student submits a student ID and equipment ID. The system checks the equipment record, stores an approved loan and returns either a collection notice or an unavailable notice.

A Student entity sends student ID and equipment ID to Check availability. That process reads equipment status from D1 Equipment. Available loan details go to Record loan, which writes a loan record to D2 Loans and sends a collection notice to the Student. Check availability can also send an unavailable notice to the Student. Student 1.0 Check availability D1 Equipment 2.0 Record loan D2 Loans student ID, equipment ID equipment ID equipment status approved loan details loan record collection notice unavailable notice

Trace the worked example

Do not merely list connected shapes. Explain what moves, in which direction, and what each process does with it.

1. The Student sends a student ID and equipment ID to *1.0 Check availability*.
2. The process sends the equipment ID to D1 Equipment and receives the equipment status.
3. If unavailable, the process returns an unavailable notice to the Student.
4. If available, approved loan details flow to *2.0 Record loan*.
5. The second process writes a loan record to D2 Loans and sends a collection notice to the Student.

Explain a relationship

Explain the relationship between the Student entity and D2 Loans. Include any process between them.

View a sample solution

The Student supplies a student ID and equipment ID to Check availability. If D1 Equipment returns an available status, approved loan details pass to Record loan. Record loan then writes a loan record to D2 Loans and returns a collection notice to the Student. The Student does not access D2 Loans directly.

Explain relationships, not just arrows

An explanation should connect evidence from the complete path.

> **[Source]** sends [named data] to [process]. The process [transformation] using [data read from a store, if relevant], then sends or stores [named output] in [destination].

If the diagram mixes manual, physical and digital flows, state that distinction. A paper consent form can be an M store; a physical item can flow from a manual store through a delivery process; a digital identifier can trigger that action.

Build the explanation

A volunteer sends a ticket code to *1.0 Validate ticket*. The process reads valid-code data from *D1 Tickets* and returns an entry decision to the volunteer.

Explain the relationship between the Volunteer and D1 Tickets in two or three sentences.

View a sample solution

The Volunteer sends a ticket code to Validate ticket. That process uses the code to retrieve valid-code data from D1 Tickets, compares the records and returns an entry decision to the Volunteer. The process mediates the relationship, so the Volunteer does not read the store directly.

Data flow within and between systems

A digital solution may contain a front end, back end and database while also exchanging data with external services.

- **Front end:** collects input and presents output through the user interface.
- **Back end:** validates requests, applies rules and coordinates data exchange.
- **Database:** retains structured data.
- **External service:** another system outside the area of study, such as a weather or payment service.

Whether something is an external entity depends on the chosen system boundary. If the diagram studies a booking app, a payment service may be external. If it studies the payment provider itself, that provider contains processes and stores.

> **Do not draw technology for its own sake:** a phone, Wi-Fi router or API is not automatically a DFD element. Represent a source, transformation, store or flow only when it helps explain the studied data process.

From scenario to DFD plan

A beach-safety app receives a location from a Beachgoer. Its back end sends the location to an external Weather service, receives current conditions, combines them with alert rules from a database and returns a safety message.

Before drawing, extract the elements:

- **Entities:** Beachgoer; Weather service
- **Process:** 1.0 Generate safety message
- **Store:** D1 Alert rules
- **Flows:** location, conditions request, current conditions, alert rules, safety message

Notice that *conditions request* and *current conditions* are separate arrows because they travel in opposite directions.

Check for understanding

Symbolise this scenario as a DFD on paper. Then check that every entity and store connects only through the process.

Check your answer

A correct diagram shows Beachgoer → *location* → 1.0 Generate safety message; the process → *conditions request* → Weather service; Weather service → *current conditions* → the process; D1 Alert rules → *alert rules* → the process; and the process → *safety message* → Beachgoer.

Develop a DFD systematically

1. Set the system boundary: decide what is inside the area of study.
2. Underline external sources and destinations.
3. Circle transformations described by verbs.
4. Identify retained, manual and temporary stores.
5. List the actual data entering and leaving each process.
6. Draw and number processes and stores.
7. Add one labelled arrow for each direction.
8. Trace every path and test the conventions.

Final convention check

- Does every process have at least one meaningful input and output?
- Does every arrow have a noun label and a clear direction?
- Do all entity and store flows pass through a process?
- Are processes named with verbs and numbered?
- Are stores numbered and correctly identified as D, M or T?
- Does the diagram show the scenario without inventing unsupported data?

Independent practice: community garden lockers

A community garden lends tools through secure lockers. A Member enters a membership code and tool ID at a terminal. The system must:

- check the membership code against retained member records
- check whether the requested tool is listed as available
- when both checks pass, record the loan and send an unlock instruction to the external Locker controller
- return a collection message or rejection reason to the Member
- store loans on disk so they remain after the terminal is turned off.

Use these required components:

- external entities: Member and Locker controller
- processes: Validate request and Record loan
- data stores: Members, Tools and Loans.

On paper, **develop** a complete Gane–Sarson DFD. Number all processes and stores, identify each store type, and label every flow with the data it carries. Then **explain** the relationship between the Member and Loans store.

View a sample solution

Elements and flows

- Member → *membership code, tool ID* → 1.0 Validate request
- 1.0 Validate request → *membership code* → D1 Members
- D1 Members → *member status* → 1.0 Validate request
- 1.0 Validate request → *tool ID* → D2 Tools
- D2 Tools → *tool availability* → 1.0 Validate request
- 1.0 Validate request → *approved loan details* → 2.0 Record loan
- 1.0 Validate request → *rejection reason* → Member
- 2.0 Record loan → *loan record* → D3 Loans
- 2.0 Record loan → *unlock instruction* → Locker controller
- 2.0 Record loan → *collection message* → Member

D1, D2 and D3 are disk stores because the scenario describes retained records.

Relationship explanation

The Member sends a membership code and tool ID to Validate request. That process checks member status from D1 Members and tool availability from D2 Tools. When both checks pass, approved loan details flow to Record loan, which writes a loan record to D3 Loans and returns a collection message. The Member therefore affects and receives confirmation of stored loan data through two processes rather than accessing the store directly.

Revision #5
Created 2026-10-06 00:40:19 UTC by Mr Napper
Updated 2026-10-06 01:31:58 UTC by Mr Napper