What a value in DHIS2 actually is

Five coordinates, not one number. Data element, category option combination, org unit, period and dataset — and the difference between a data element and an indicator, which is where most confusion about routine figures starts.

Pierre Robentz Cassion

Lesson 1 of 8

Routine Data and DHIS2 — How the database is shaped

What this lesson covers

- A value has five coordinates
- Data element: the thing that is counted
- Category combinations: the breakdown built into the element
- Data element versus indicator
- Datasets: the form, and what "expected" means
- Ask for the metadata, not just the data
- What comes next

A value has five coordinates

Coordinate	In the extract	What it answers
Data element	`antigen` plus the measure	What was counted
Category option combination	not present	Which breakdown of it
Org unit	`facility_id`	Where
Period	`period`	When
Dataset	not present	Which form it was collected on

Data element: the thing that is counted — In Python

```
import pandas as pd

vax = pd.read_csv("vaccination-coverage-2024.v1.csv", parse_dates=["period"])
print(sorted(vax["antigen"].unique()))
print(vax.groupby("antigen")["doses_administered"].sum())
```

Data element: the thing that is counted — In R

```r
library(dplyr)
library(readr)

vax <- read_csv("vaccination-coverage-2024.v1.csv")

vax |> summarise(doses = sum(doses_administered), .by = antigen)
```

Data element: the thing that is counted

- Ask which it is — A denominator stored as an annual data element can be revised retrospectively; one stored as an...

Category combinations: the breakdown built into the element — Example

```
Data element:  Penta3 doses administered
Category combo: Age (<1, 1+) x Sex (F, M)
Stored values:  4 per org unit per period
```

Category combinations: the breakdown built into the element

- **The data element has no disaggregation**, so there is one value and nothing was lost.
- **The export aggregated across category option combinations**, in which case the disaggregation exists in the system...

Category combinations: the breakdown built into the element — In Python

```
expected = (vax["facility_id"].nunique() * vax["period"].nunique()
            * vax["antigen"].nunique())
print(f"{expected} expected rows, {len(vax)} in the file")
```

Category combinations: the breakdown built into the element — In R

```
c(expected = n_distinct(vax$facility_id) * n_distinct(vax$period) *
             n_distinct(vax$antigen),
  actual = nrow(vax))
```

Data element versus indicator

- A **data element** is stored. Somebody typed it.
- An **indicator** is calculated. It has a numerator, a denominator and a factor, and DHIS2 evaluates it at whatever...

Data element versus indicator — Example

```
Indicator:   Penta3 coverage
Numerator:   #{Penta3 doses administered}
Denominator: #{Surviving infants}
Factor:      100
```

Data element versus indicator

- An indicator is not stored, so it cannot be wrong in the database — If a coverage figure is wrong, either a data...
- The indicator is evaluated at the level you request — and the arithmetic is sum-then-divide, not divide-then-average

Data element versus indicator — In Python

```
district = vax[vax["report_submitted"] == True]
correct = (district["doses_administered"].sum()
           / district["target_population"].sum())
wrong = (district["doses_administered"] / district["target_population"]).mean()
print(f"ratio of sums {correct:.3f}, mean of ratios {wrong:.3f}")
```

Data element versus indicator — In R

```
vax |>
  filter(report_submitted) |>
  summarise(ratio_of_sums = sum(doses_administered) / sum(target_population),
            mean_of_ratios = mean(doses_administered / target_population))
```

Data element versus indicator

- An indicator can be requested for a period the data does not support — Ask for annual coverage and DHIS2 will sum...

Datasets: the form, and what "expected" means — In Python

```
reported = vax["report_submitted"] == True
print(f"{reported.sum()} of {len(vax)} facility-month-antigen values reported")
print(f"reporting rate {reported.mean():.1%}")
```

Datasets: the form, and what "expected" means — In R

```r
mean(vax$report_submitted)
```

Datasets: the form, and what "expected" means — In Python

```
partial = (
    vax[reported].groupby(["facility_id", "period"])["antigen"].nunique()
)
print(f"{(partial < 6).sum()} facility-months with only some antigens reported")
```

Datasets: the form, and what "expected" means — In R

```
vax |>
  filter(report_submitted) |>
  summarise(antigens = n_distinct(antigen), .by = c(facility_id, period)) |>
  filter(antigens < 6) |>
  nrow()
```

Ask for the metadata, not just the data

- The **data element definitions**, including how each is aggregated over periods (summed, averaged, or last value — this...
- The **category combination** for each element, so you know what was collapsed.
- The **dataset assignment**, so you know how many units were expected.
- The **indicator definitions** for anything calculated, with numerator and denominator expressions.

What comes next

- You know what a value is and where it sits.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/routine-data-and-dhis2/dhis2-01-data-elements